



# HTML for All Learner

HTML you can create your own Webpages.

HTML stands for Hypertext Markup Language

HTML describes the structure of a web page

HTML is easy to learn



01 JANUARY 2023

[Bintr.online](http://Bintr.online)

Darani, Agia, Goalpara, Assam (IND)

# Introduction to the book

Computers have entered our life as both professional and recreational requirements. the way we interact with each other has also evolved.

The content in this book combines the various distinguishing features that help people to bring a revolutionary change in their life, through personal computers.

Congratulations on choosing this book. The content has been designed keeping in mind the needs of the current industry.

I would be glad to have your advice. Kindly send your best wishes to [rabhabinod146@gmail.com](mailto:rabhabinod146@gmail.com) or your feedback.

Wishing you all the best.

Name: Binod Rabha



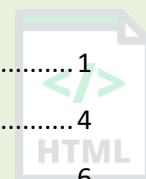
[This Books](#) is free for user from ASSAM Only . However, you have no permission to redistribute the downloadable file on any Books collection website Or Share the downloaded file in any compatible media. for example WhatsApp

[Contact us](#) for more info.

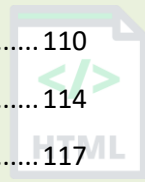
Search [bintr.online](#) in google



1	HTML Introduction.....	1
2	HTML Editors.....	4
3	HTML Basic Examples.....	6
4	HTML Elements .....	8
5	HTML Attributes.....	9
6	HTML Headings .....	12
7	HTML Styles.....	17
8	HTML Text Formatting .....	19
9	HTML Quotation and Citation Elements .....	22
10	HTML Comments.....	24
11	HTML Colors .....	26
12	HTML Colors Names.....	28
13	HTML RGB and RGBA Colors .....	37
14	HTML HEX Colors.....	40
15	HTML HSL and HSLA Colors.....	42
16	HTML Styles - CSS.....	45
17	HTML Style Tags .....	50
18	HTML - Marquees.....	54
19	HTML Link Tags .....	56
20	HTML Images.....	59
21	HTML Image Tags.....	63
22	HTML Background Images .....	69
23	HTML <picture> Element .....	72
24	HTML.....	74
25	HTML Tables.....	76
26	HTML Table Borders.....	79
27	HTML Table Sizes.....	82
28	HTML Table Headers .....	85
29	HTML Table Padding & Spacing .....	89
30	HTML Table Colspan & Rowspan .....	91
31	HTML Table Styling.....	93
32	HTML Table Colgroup.....	96
33	HTML Lists .....	99



34	HTML Ordered Lists.....	104
35	HTML Block and Inline Elements.....	107
36	HTML class Attribute.....	110
37	HTML id Attribute .....	114
38	HTML Iframes .....	117
39	HTML JavaScript.....	119
40	HTML File Paths.....	120
41	HTML - The Head Element .....	123
42	HTML Layout Elements and Techniques .....	129
43	HTML Responsive Web Design.....	131
44	Hello World .....	135
45	HTML Computer Code Elements.....	137
46	HTML Semantic Elements .....	140
47	HTML Style Guide.....	147
48	HTML Entities .....	156
49	HTML Symbols.....	158
50	Using Emojis in HTML.....	160
51	HTML Encoding (Character Sets).....	163
52	HTML Uniform Resource Locators .....	171
53	HTML ASCII Reference .....	172
54	HTML Versus XHTML.....	175
55	HTML Forms .....	179
56	HTML Form Attributes .....	184
57	HTML Form Elements.....	187
58	HTML Input Types .....	192
59	HTML Input Attributes .....	201
60	HTML Input form Attributes .....	207
61	HTML Canvas Graphics.....	211
62	HTML SVG Graphics .....	214
63	HTML Multimedia .....	217
64	HTML Video.....	219
65	HTML Audio.....	222
66	HTML Plug-ins .....	225
67	HTML YouTube Videos .....	227
68	HTML Geolocation API .....	229



69	HTML Drag and Drop API .....	232
70	HTML Web Storage API .....	235
71	HTML Web Workers API.....	237
72	HTML SSE API .....	240
73	HTML Accessibility .....	242
74	HTML Element Reference .....	245
75	HTML Element Reference - By Category .....	250
76	HTML Attribute Reference .....	256
77	HTML Global Attributes .....	264
78	HTML Event Attributes.....	266
79	HTML Canvas Reference .....	271
80	HTML Audio/Video DOM Reference .....	275
81	HTML Character Sets.....	278
82	HTML <!DOCTYPE>.....	285
83	HTML URL Encoding Reference.....	289
84	HTML Language Code Reference .....	298
85	HTML ISO Country Codes Reference.....	304
86	HTTP Status Messages .....	309
87	HTTP Request Methods.....	312
88	Pixels to Ems Conversion .....	315
89	Keyboard Shortcuts.....	317
90	HTML Tags.....	321

# 1 HTML Introduction

HTML is the standard markup language for creating Web pages.

What is HTML?

- **HTML stands for Hyper Text Markup Language**
- **HTML is the standard markup language for creating Web pages**
- **HTML describes the structure of a Web page**
- **HTML consists of a series of elements**
- **HTML elements tell the browser how to display the content**
- **HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.**



**HTML page structure**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

**Example Explained**

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

**Tags in HTML**

1. HTML tags are used to mark-up HTML elements.
2. HTML tags are surrounded by the two characters `<` and `>`.
3. The surrounding characters are called angle brackets.
4. HTML tags normally come in pairs like `<b>` and `</b>`.
5. The first tag in a pair is the start tag, the second tag is the end tag.
6. The text between the start and end tags is the element content.
7. HTML tags are not case sensitive, `<b>` means the same as `<B>`.

**HTML Tag Code:**

`<tag> Content </tag>`

`<p>This text will be rendered like a paragraph. </p>`.

Tags are represented in two ways.

- Start and end tag (paired tags).
- An empty tag.



### Start and End Tag

An HTML tag looks like: `<tag> This is come content </tag>`.

Here `<tag>` is start/opening tag and `</tag>` is closing/end tag. Both together with content inside them are called as an HTML element.

### An Empty Tag

HTML defines certain tags as empty tags, means they don't require closing tags and they do not contain any content. Such tags are closed in a start tag only. For example there is an element `<br>` which breaks the line to the next line. When we write

```
<p> This is <br> a paragraph </p>
```

Then it would produce result like:

This is

a paragraph

Means it's breaking the line and does not require closing tag `</br>`. Instead we can close it as `<br />` in itself. It can write either `<br>` or `<br />`. Both are having same meaning, but writing `<br />` is better practice as per the HTML standards.

### What is an HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:

```
<tagname> Content goes here... </tagname>
```

The HTML **element** is everything from the start tag to the end tag:

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

Start tag	Element content	End tag
<code>&lt;h1&gt;</code>	My First Heading	<code>&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	My first paragraph.	<code>&lt;/p&gt;</code>
<code>&lt;br&gt;</code>	none	none

**Note:** Some HTML elements have no content (like the `<br>` element). These elements are called empty elements. Empty elements do not have an end tag!

### Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

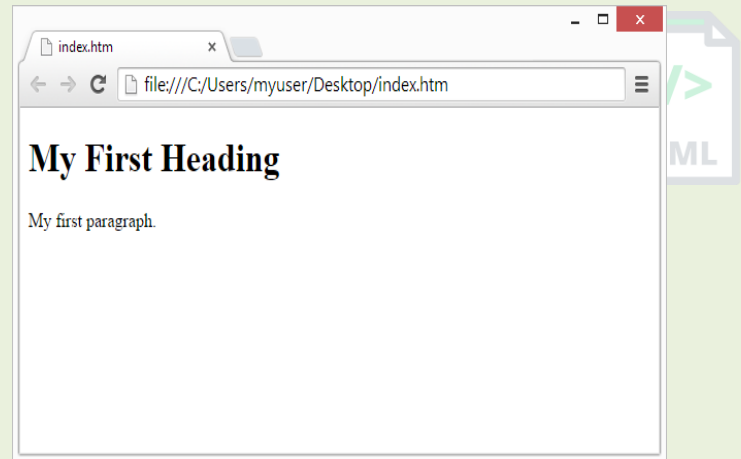


A browser does not display the HTML tags, but uses them to determine how to display the document:

## HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```



**Note:** The content inside the `<body>` section (the white area above) will be displayed in a browser. The content inside the `<title>` element will be shown in the browser's title bar or in the page's tab.

## 1.1 HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	HTML 3.2
1999	HTML 4.01
2000	XHTML 1.0
2008	HTML5 First Public Draft
2012	HTML5 Living Standard
2014	HTML5
2016	HTML 5.1
2017	HTML5.1 2nd Edition
2017	HTML5.2

This tutorial follows the latest HTML5 standard.

## 2 HTML Editors

A simple text editor is all you need to learn HTML.

### 2.1 Learn HTML Using Notepad or TextEdit

Web pages can be created and modified by using professional HTML editors. However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac). We believe in that using a simple text editor is a good way to learn HTML.



Follow the steps below to create your first web page with Notepad or TextEdit.

#### Step 1: Open Notepad (PC)

##### Windows 8 or later:

Open the Start Screen (the window symbol at the bottom left on your screen). Type Notepad.

##### Windows 7 or earlier:

Open Start > Programs > Accessories > Notepad

#### Step 1: Open TextEdit (Mac)

Open Finder > Applications > TextEdit

Also change some preferences to get the application to save files correctly. In Preferences > Format > choose "Plain Text"

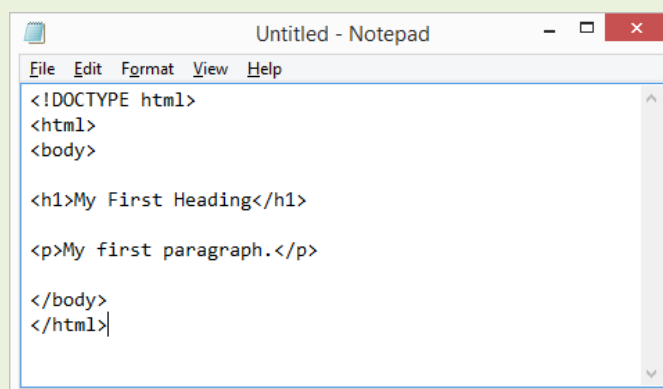
Then under "Open and Save", check the box that says "Display HTML files as HTML code instead of formatted text".

Then open a new document to place the code.

#### Step 2: Write Some HTML

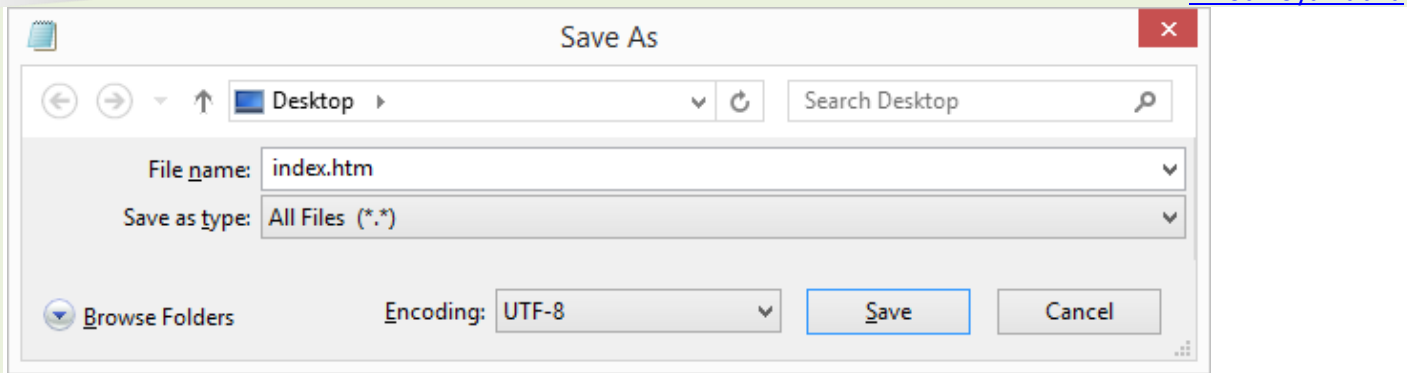
Write or copy the following HTML code into Notepad:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```



#### Step 3: Save the HTML Page

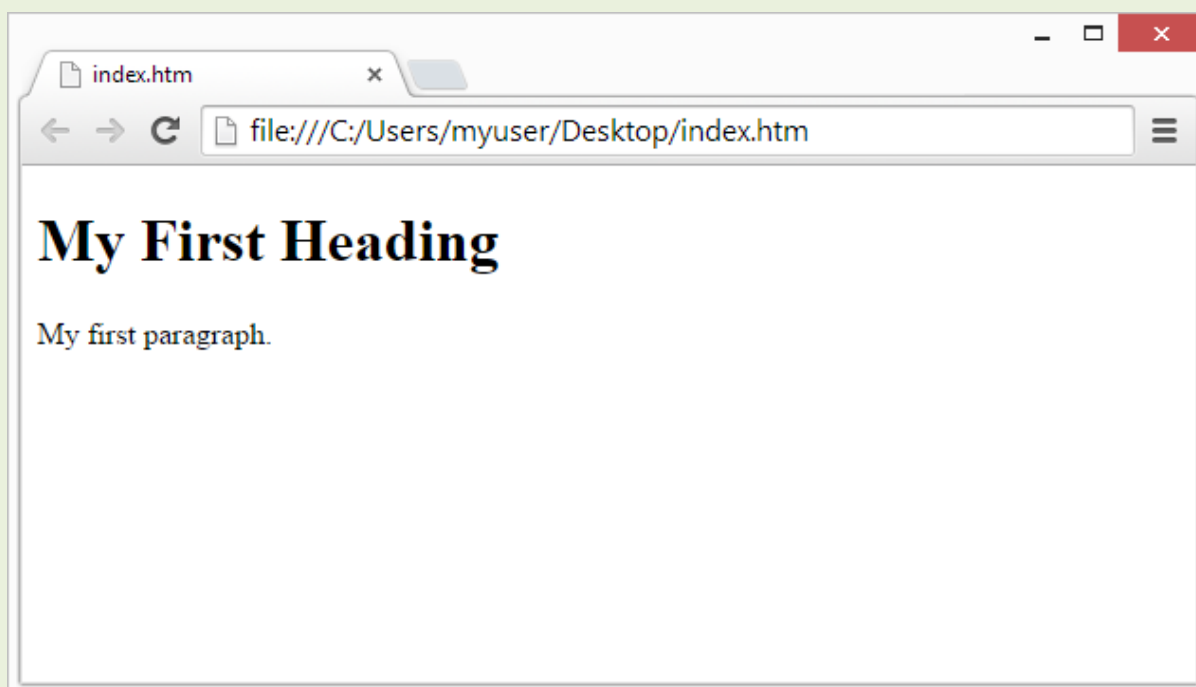
Save the file on your computer. Select File > Save as in the Notepad menu. Name the file "index.htm" and set the encoding to UTF-8 (which is the preferred encoding for HTML files).



**Tip:** You can use either .htm or .html as file extension. There is no difference, it is up to you.

#### Step 4: View the HTML Page in Your Browser

Open the saved HTML file in your favourite browser (double click on the file, or right-click - and choose "Open with"). The result will look much like this:



## 3 HTML Basic Examples

In this chapter we will show some basic HTML examples. Don't worry if we use tags you have not learned about yet.

### HTML Documents

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

The HTML document itself begins with `<html>` and ends with `</html>`.

The visible part of the HTML document is between `<body>` and `</body>`.

### Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

The `<!DOCTYPE>` Declaration

The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The `<!DOCTYPE>` declaration is not case sensitive.

The `<!DOCTYPE>` declaration for HTML5 is:

```
<!DOCTYPE html>
```

### 3.1 HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading:

#### Example:

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

### 3.2 HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

#### Example:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```



### 3.3 HTML Links

HTML links are defined with the `<a>` tag:

**Example:**

The link's destination is specified in the `href` attribute. Attributes are used to provide additional information about HTML elements.

You will learn more about attributes in a later chapter.



### 3.4 HTML Images

HTML images are defined with the `<img>` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

**Example:**

**How to View HTML Source?**

Have you ever seen a Web page and wondered "Hey! How did they do that?"

**View HTML Source Code:**

Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in Edge), or similar in other browsers. This will open a window containing the HTML source code of the page.

**Inspect an HTML Element:**

Right-click on an element (or a blank area), and choose "Inspect" or "Inspect Element" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

## 4 HTML Elements

An HTML element is defined by a start tag, some content, and an end tag.

### 4.1 HTML Elements

The HTML **element** is everything from the start tag to the end tag:

```
<tagname>Content goes here...</tagname>
```

Examples of some HTML elements:

**Note:** Some HTML elements have no content (like the `<br>` element). These elements are called empty elements. Empty elements do not have an end tag!

### 4.2 Nested HTML Elements

HTML elements can be nested (this means that elements can contain other elements).

All HTML documents consist of nested HTML elements.

The following example contains four HTML elements (`<html>`, `<body>`, `<h1>` and `<p>`):

**Example:**

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

#### Example Explained

The `<html>` element is the root element and it defines the whole HTML document.

It has a start tag `<html>` and an end tag `</html>`.

Then, inside the `<html>` element there is a `<body>` element:

```
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
```

The `<body>` element defines the document's body. It has a start tag `<body>` and an end tag `</body>`.

Then, inside the `<body>` element there are two other elements: `<h1>` and `<p>`:

```
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

The `<h1>` element defines a heading. It has a start tag `<h1>` and an end tag `</h1>`:

```
<h1>My First Heading</h1>
```



The `<p>` element defines a paragraph. It has a start tag `<p>` and an end tag `</p>`:

```
<p>My first paragraph.</p>
```



### 4.3 Never Skip the End Tag

Some HTML elements will display correctly, even if you forget the end tag:

**Example:**

```
<html>
<body>
<p>This is a paragraph
<p>This is a paragraph
</body>
</html>
```

**However, never rely on this! Unexpected results and errors may occur if you forget the end tag!**

### 4.4 Empty HTML Elements

HTML elements with no content are called empty elements. The `<br>` tag defines a line break, and is an empty element without a closing tag:

**Example**

```
<p>This is a <br> paragraph with a line break.</p>
```

### 4.5 HTML is Not Case Sensitive

HTML tags are not case sensitive: `<P>` means the same as `<p>`.

The HTML standard does not require lowercase tags, but I **recommends** use lowercase in HTML, and **demands** lowercase for stricter document types like XHTML.

I always use lowercase tag names.

### 4.6 HTML Tag Reference

Bintr' tag reference contains additional information about these tags and their attributes.

Tag	Description
<code>&lt;html&gt;</code>	Defines the root of an HTML document
<code>&lt;body&gt;</code>	Defines the document's body
<code>&lt;h1&gt;</code> to <code>&lt;h6&gt;</code>	Defines HTML headings

## 5 HTML Attributes

HTML attributes provide additional information about HTML elements.

## 5.1 HTML Attributes

All HTML elements can have **attributes**

Attributes provide **additional information** about elements

Attributes are always specified in **the start tag**

Attributes usually come in name/value pairs like: **name="value"**



## 5.2 The href Attribute

The `<a>` tag defines a hyperlink. The `href` attribute specifies the URL of the page the link goes to:

### Example

```
<a href="https://www.bintr.online"> Visit Bintr.online </a>
```

## 5.3 The src Attribute

The `<img>` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

### Example

```

```

There are two ways to specify the URL in the `src` attribute:

**1. Absolute URL** - Links to an external image that is hosted on another website.

Example: `src="https://www.bintr.com/images/img_girl.jpg"`.

**Notes:** External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

**2. Relative URL** - Links to an image that is hosted within the website. Here, the URL does not include the domain name. If the URL begins without a slash, it will be relative to the current page. Example: `src="img_girl.jpg"`. If the URL begins with a slash, it will be relative to the domain. Example: `src="/images/img_girl.jpg"`.

**Tip:** It is almost always best to use relative URLs. They will not break if you change domain.

## 5.4 The width and height Attributes

The `<img>` tag should also contain the `width` and `height` attributes, which specifies the width and height of the image (in pixels):

### Example

```

```

The alt Attribute

The required `alt` attribute for the `<img>` tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the `src` attribute, or if the user uses a screen reader.

### Example

```

```

### Example



See what happens if we try to display an image that does not exist:

```

```



## 5.5 The style Attribute

The `style` attribute is used to add styles to an element, such as color, font, size, and more.

### Example

```
<p style="color:red;">This is a red paragraph.</p>
```

## 5.6 The lang Attribute

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

Country codes can also be added to the language code in the `lang` attribute. So, the first two characters define the language of the HTML page, and the last two characters define the country.

The following example specifies English as the language and United States as the country:

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

## 5.7 The title Attribute

The `title` attribute defines some extra information about an element.

The value of the title attribute will be displayed as a tooltip when you mouse over the element:

### Example

```
<p title="I'm a tooltip">This is a paragraph.</p>
```

## 5.8 We Suggest: Always Use Lowercase Attributes

The HTML standard does not require lowercase attribute names.

The title attribute (and all other attributes) can be written with uppercase or lowercase like `title` or `TITLE`.

However, I **recommends** lowercase attributes in HTML, and **demands** lowercase attributes for stricter document types like XHTML.

At Bintr.online we always use lowercase attribute names.

## 5.9 We Suggest: Always Quote Attribute Values

The HTML standard does not require quotes around attribute values.

However, C **recommends** quotes in HTML, and **demands** quotes for stricter document types like XHTML.

### Good:

```
<a href="https://www.bintr.com/html/">Visit our HTML tutorial</a>
```

### Bad:

```
<a href=https://www.bintr.com/html/>Visit our HTML tutorial</a>
```

Sometimes you have to use quotes. This example will not display the title attribute correctly, because it contains a space:

### Example

```
<p title>About Bintr>
```

At Bintr.online we always use quotes around attribute values.

## 5.10 Single or Double Quotes?

Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

```
<p title="John 'ShotGun' Nelson">
```

## 5.11 Chapter Summary

- All HTML elements can have **attributes**
- The `href` attribute of `<a>` specifies the URL of the page the link goes to
- The `src` attribute of `<img>` specifies the path to the image to be displayed
- The `width` and `height` attributes of `<img>` provide size information for images
- The `alt` attribute of `<img>` provides an alternate text for an image
- The `style` attribute is used to add styles to an element, such as color, font, size, and more
- The `lang` attribute of the `<html>` tag declares the language of the Web page
- The `title` attribute defines some extra information about an element

---

## 6 HTML Headings

---

HTML headings are titles or subtitles that you want to display on a webpage.

### Example

# Heading 1

## Heading 2



### Heading 3

### Heading 4

### Heading 5

### Heading 6



## 6.1 HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading.

### Example

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

**Note:** Browsers automatically add some white space (a margin) before and after a heading.

## Headings Are Important

Search engines use the headings to index the structure and content of your web pages.

Users often skim a page by its headings. It is important to use headings to show the document structure.

`<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.

**Note:** Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

## 6.2 Bigger Headings

Each HTML heading has a default size. However, you can specify the size for any heading with the `style` attribute, using the CSS `font-size` property:

### Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

## 6.3 HTML Tag Reference

tag reference contains additional information about these tags and their attributes.

Tag	Description
<code>&lt;html&gt;</code>	Defines the root of an HTML document
<code>&lt;body&gt;</code>	Defines the document's body
<code>&lt;h1&gt;</code> to <code>&lt;h6&gt;</code>	Defines HTML headings



## 6.4 HTML Paragraphs

A paragraph always starts on a new line, and is usually a block of text.

### HTML Paragraphs

The HTML `<p>` element defines a paragraph.

A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph.

### Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

### HTML Display

You cannot be sure how HTML will be displayed. Large or small screens, and resized windows will create different results. With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code. The browser will automatically remove any extra spaces and lines when the page is displayed:

### Example

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>
```

```
<p>
This paragraph
contains    a lot of spaces
in the source    code,
but the    browser
ignores it.
</p>
```

## 6.5 HTML Horizontal Rules

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:

## Example

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```



The `<hr>` tag is an empty tag, which means that it has no end tag.

## 6.6 The Poem Problem

This poem will display on a single line:

### Example

```
<p>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</p>
```

## 6.7 Solution - The HTML `<pre>` Element

The HTML `<pre>` element defines preformatted text. The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

### Example

```
<pre>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</pre>
```

## 6.8 HTML Line Breaks

The HTML `<br>` element defines a line break.

Use `<br>` if you want a line break (a new line) without starting a new paragraph:

### Example

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

The `<br>` tag is an empty tag, which means that it has no end tag.

## 6.9 HTML Tag Reference

tag reference contains additional information about HTML elements and their attributes.

Tag	Description
<code>&lt;p&gt;</code>	Defines a paragraph

<code>&lt;hr&gt;</code>	Defines a thematic change in the content
<code>&lt;br&gt;</code>	Inserts a single line break
<code>&lt;pre&gt;</code>	Defines pre-formatted text



## 7 HTML Styles

The HTML `style` attribute is used to add styles to an element, such as color, font, size, and more.

```
<!DOCTYPE html>
<html>
<body>
<p>I am normal</p>
<p style="color:red;">I am red</p>
<p style="color:blue;">I am blue</p>
<p style="font-size:50px;">I am big</p>
</body>
</html>
```

### Example

I am Red

I am Blue

I am Big



### 7.1 The HTML Style Attribute

Setting the style of an HTML element, can be done with the `style` attribute. The HTML `style` attribute has the following syntax:

```
<tagname style="property:value;">
```

The **property** is a CSS property. The **value** is a CSS value.

You will learn more about CSS later in this tutorial.

### 7.2 Background Color

The CSS `background-color` property defines the background color for an HTML element.

#### Example

Set the background color for a page to powderblue:

```
<body style="background-color:powderblue;">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

#### Example

Set background color for two different elements:

```
<body>
<h1 style="background-color:powderblue;">This is a heading</h1>
```

```
<p style="background-color:tomato;">This is a paragraph.</p>
</body>
```

## 7.3 Text Color

The CSS `color` property defines the text color for an HTML element:

### Example

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```



## 7.4 Fonts

The CSS `font-family` property defines the font to be used for an HTML element:

### Example

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

## 7.5 Text Size

The CSS `font-size` property defines the text size for an HTML element:

### Example

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

## 7.6 Text Alignment

The CSS `text-align` property defines the horizontal text alignment for an HTML element:

### Example

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

## 7.7 Chapter Summary

- Use the `style` attribute for styling HTML elements
- Use `background-color` for background color
- Use `color` for text colors
- Use `font-family` for text fonts
- Use `font-size` for text sizes
- Use `text-align` for text alignment



## 8 HTML Text Formatting

HTML contains several elements for defining text with a special meaning.

### Example

This text is bold

*This text is italic*

This is <sub>subscript</sub> and <sup>superscript</sup>



### 8.1 HTML Formatting Elements

Formatting elements were designed to display special types of text:

- `<b>` - Bold text
- `<strong>` - Important text
- `<i>` - Italic text
- `<em>` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `<del>` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

### 8.2 HTML `<b>` and `<strong>` Elements

The HTML `<b>` element defines bold text, without any extra importance.

#### Example

```
<b>This text is bold</b>
```

The HTML `<strong>` element defines text with strong importance. The content inside is typically displayed in bold.

#### Example

```
<strong>This text is important!</strong>
```

### 8.3 HTML `<i>` and `<em>` Elements

The HTML `<i>` element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

**Tip:** The `<i>` tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

#### Example

```
<i>This text is italic</i>
```

The HTML `<em>` element defines emphasized text. The content inside is typically displayed in italic.

**Tip:** A screen reader will pronounce the words in `<em>` with an emphasis, using verbal stress.

#### Example

`<em>`This text is emphasized`</em>`

## 8.4 HTML `<small>` Element

The HTML `<small>` element defines smaller text:

### Example

`<small>`This is some smaller text.`</small>`



## 8.5 HTML `<mark>` Element

The HTML `<mark>` element defines text that should be marked or highlighted:

### Example

`<p>`Do not forget to buy `<mark>`milk`</mark>` today.`</p>`

## 8.6 HTML `<del>` Element

The HTML `<del>` element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text:

### Example

`<p>`My favorite color is `<del>`blue`</del>` red.`</p>`

## 8.7 HTML `<ins>` Element

The HTML `<ins>` element defines a text that has been inserted into a document. Browsers will usually underline inserted text:

### Example

`<p>`My favorite color is `<del>`blue`</del>` `<ins>`red`</ins>`.`</p>`

## 8.8 HTML `<sub>` Element

The HTML `<sub>` element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H<sub>2</sub>O:

### Example

`<p>`This is `<sub>`subscripted`</sub>` text.`</p>`

## 8.9 HTML `<sup>` Element

The HTML `<sup>` element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW<sup>[1]</sup>:

### Example

`<p>`This is `<sup>`superscripted`</sup>` text.`</p>`

## 8.10 HTML Text Formatting Elements

Tag	Description
<b>&lt;b&gt;</b>	Defines bold text
<b>&lt;em&gt;</b>	Defines emphasized text
<b>&lt;i&gt;</b>	Defines a part of text in an alternate voice or mood
<b>&lt;small&gt;</b>	Defines smaller text
<b>&lt;strong&gt;</b>	Defines important text
<b>&lt;sub&gt;</b>	Defines subscripted text
<b>&lt;sup&gt;</b>	Defines superscripted text
<b>&lt;ins&gt;</b>	Defines inserted text
<b>&lt;del&gt;</b>	Defines deleted text
<b>&lt;mark&gt;</b>	Defines marked/highlighted text



## 9 HTML Quotation and Citation Elements

In this chapter we will go through the `<blockquote>`, `<q>`, `<abbr>`, `<address>`, `<cite>`, and `<bdo>` HTML elements.

### Example

Here is a quote from WWF's website:

For nearly 60 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by more than one million members in the United States and close to five million globally.



### 9.1 HTML `<blockquote>` for Quotations

The HTML `<blockquote>` element defines a section that is quoted from another source.

Browsers usually indent `<blockquote>` elements.

#### Example

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

### 9.2 HTML `<q>` for Short Quotations

The HTML `<q>` tag defines a short quotation.

Browsers normally insert quotation marks around the quotation.

#### Example

```
<p>WWF's goal is to: <q>Build a future where people live in harmony with
nature.</q></p>
```

### 9.3 HTML `<abbr>` for Abbreviations

The HTML `<abbr>` tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

Marking abbreviations can give useful information to browsers, translation systems and search-engines.

**Tip:** Use the global title attribute to show the description for the abbreviation/acronym when you mouse over the element.

#### Example

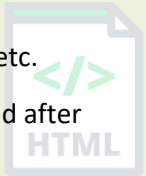
```
<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>
```

## 9.4 HTML <address> for Contact Information

The HTML <address> tag defines the contact information for the author/owner of a document or an article.

The contact information can be an email address, URL, physical address, phone number, social media handle, etc.

The text in the <address> element usually renders in *italic*, and browsers will always add a line break before and after the <address> element.



### Example

```
<address>
Written by Binod Toya Rabha.<br>
Visit us at:<br>
bintr.online<br>
Goalpara, Assam,<br>
India
</address>
```

## 9.5 HTML <cite> for Work Title

The HTML <cite> tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

**Note:** A person's name is not the title of a work.

The text in the <cite> element usually renders in *italic*.

### Example

```
<p><cite>The Scream</cite> by Edvard Munch. Painted in 1893.</p>
```

## 9.6 HTML <bdo> for Bi-Directional Override

BDO stands for Bi-Directional Override.

The HTML <bdo> tag is used to override the current text direction:

### Example

```
<bdo dir="rtl">This text will be written from right to left</bdo>
```

## 9.7 HTML Quotation and Citation Elements

Tag	Description
<abbr>	Defines an abbreviation or acronym
<address>	Defines contact information for the author/owner of a document
<bdo>	Defines the text direction
<blockquote>	Defines a section that is quoted from another source
<cite>	Defines the title of a work
<q>	Defines a short inline quotation

## 10 HTML Comments

HTML comments are not displayed in the browser, but they can help document your HTML source code.



### 10.1 HTML Comment Tag

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

Notice that there is an exclamation point (!) in the start tag, but not in the end tag.

**Note:** Comments are not displayed by the browser, but they can help document your HTML source code.

### 10.2 Add Comments

With comments you can place notifications and reminders in your HTML code:

Example

```
<!-- This is a comment -->
```

```
<p>This is a paragraph.</p>
```

```
<!-- Remember to add more information here -->
```

### 10.3 Hide Content

Comments can be used to hide content.

Which can be helpful if you hide content temporarily:

**Example**

```
<p>This is a paragraph.</p>
```

```
<!-- <p>This is another paragraph </p> -->
```

```
<p>This is a paragraph too.</p>
```

You can also hide more than one line, everything between the `<!--` and the `-->` will be hidden from the display.

**Example**

Hide a section of HTML code:

```
<p>This is a paragraph.</p>
```

```
<!--
```

```
<p>Look at this cool image:</p>
```

```

```

```
-->
```

```
<p>This is a paragraph too.</p>
```

Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors.

## 10.4 Hide Inline Content

Comments can be used to hide parts in the middle of the HTML code.

### Example

Hide a part of a paragraph:

```
<p>This <!-- great text --> is a paragraph.</p>
```



## 11 HTML Colors

HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values.



### 11.1 Color Names

In HTML, a color can be specified by using a color name:

Tomato
Orange
DodgerBlue
MediumSeaGreen
Gray
SlateBlue
Violet
LightGray

[HTML supports 140 standard color names.](#)

### 11.2 Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum color sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

**Example:**

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

### 11.3 Text Color

You can set the color of text:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

**Example:**



```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

## 11.4 Border Color

You can set the color of borders:

### Example:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

## 11.5 Color Values

In HTML, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values.

The following three <div> elements have their background color set with RGB, HEX, and HSL values:

```
rgb(255, 99, 71)
#ff6347
hsl(9, 100%, 64%)
```

The following two <div> elements have their background color set with RGBA and HSLA values, which adds an Alpha channel to the color (here we have 50% transparency):

### Example:

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

## Learn more about Color Values

You will learn more about RGB, HEX and HSL in the chapters.



## 12 HTML Colors Names

### 12.1 Color Names Supported by All Browsers

All modern browsers support the following 140 color names (click on a color name, or a hex value, to view the color as the background-color along with different text colors):



For a full overview of HTML colors, visit [our colors tutorial](#).

Color Mixer	<b>AliceBlue</b> #F0F8FF	Color Picker
Color Mixer	<b>AntiqueWhite</b> #FAEBD7	Color Picker
Color Mixer	<b>Aqua</b> #00FFFF	Color Picker
Color Mixer	<b>Aquamarine</b> #7FFFD4	Color Picker
Color Mixer	<b>Azure</b> #F0FFFF	Color Picker
Color Mixer	<b>Beige</b> #F5F5DC	Color Picker
Color Mixer	<b>Bisque</b> #FFE4C4	Color Picker
Color Mixer	<b>Black</b> #000000	Color Picker
Color Mixer	<b>BlanchedAlmond</b> #FFEBCD	Color Picker
Color Mixer	<b>Blue</b> #0000FF	Color Picker
Color Mixer	<b>BlueViolet</b> #8A2BE2	Color Picker
Color Mixer	<b>Brown</b> #A52A2A	Color Picker
Color Mixer	<b>BurlyWood</b> #DEB887	Color Picker

		<a href="#">Color Picker</a>
Color Mixer	<b>CadetBlue</b> #5F9EA0	<a href="#">Color Picker</a>
Color Mixer	<b>Chartreuse</b> #7FFF00	<a href="#">Color Picker</a>
Color Mixer	<b>Chocolate</b> #D2691E	<a href="#">Color Picker</a>
Color Mixer	<b>Coral</b> #FF7F50	<a href="#">Color Picker</a>
Color Mixer	<b>CornflowerBlue</b> #6495ED	<a href="#">Color Picker</a>
Color Mixer	<b>Cornsilk</b> #FFF8DC	<a href="#">Color Picker</a>
Color Mixer	<b>Crimson</b> #DC143C	<a href="#">Color Picker</a>
Color Mixer	<b>Cyan</b> #00FFFF	<a href="#">Color Picker</a>
Color Mixer	<b>DarkBlue</b> #00008B	<a href="#">Color Picker</a>
Color Mixer	<b>DarkCyan</b> #008B8B	<a href="#">Color Picker</a>
Color Mixer	<b>DarkGoldenRod</b> #B8860B	<a href="#">Color Picker</a>
Color Mixer	<b>DarkGray</b> #A9A9A9	<a href="#">Color Picker</a>
Color Mixer	<b>DarkGrey</b> #A9A9A9	<a href="#">Color Picker</a>
Color Mixer	<b>DarkGreen</b> #006400	<a href="#">Color Picker</a>
Color Mixer	<b>DarkKhaki</b> #BDB76B	<a href="#">Color Picker</a>
	<b>DarkMagenta</b>	<a href="#">Color Picker</a>

Color Mixer	<b>#8B008B</b>	Color Picker
Color Mixer	<b>DarkOliveGreen</b> #556B2F	Color Picker
Color Mixer	<b>DarkOrange</b> #FF8C00	Color Picker
Color Mixer	<b>DarkOrchid</b> #9932CC	Color Picker
Color Mixer	<b>DarkRed</b> #8B0000	Color Picker
Color Mixer	<b>DarkSalmon</b> #E9967A	Color Picker
Color Mixer	<b>DarkSeaGreen</b> #8FBC8F	Color Picker
Color Mixer	<b>DarkSlateBlue</b> #483D8B	Color Picker
Color Mixer	<b>DarkSlateGray</b> #2F4F4F	Color Picker
Color Mixer	<b>DarkSlateGrey</b> #2F4F4F	Color Picker
Color Mixer	<b>DarkTurquoise</b> #00CED1	Color Picker
Color Mixer	<b>DarkViolet</b> #9400D3	Color Picker
Color Mixer	<b>DeepPink</b> #FF1493	Color Picker
Color Mixer	<b>DeepSkyBlue</b> #00BFFF	Color Picker
Color Mixer	<b>DimGray</b> #696969	Color Picker
Color Mixer	<b>DimGrey</b> #696969	Color Picker

		<a href="#">Color Picker</a>
<b>DodgerBlue</b> #1E90FF		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>FireBrick</b> #B22222		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>FloralWhite</b> #FFFAF0		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>ForestGreen</b> #228B22		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>Fuchsia</b> #FF00FF		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>Gainsboro</b> #DCDCDC		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>GhostWhite</b> #F8F8FF		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>Gold</b> #FFD700		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>GoldenRod</b> #DAA520		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>Gray</b> #808080		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>Grey</b> #808080		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>Green</b> #008000		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>GreenYellow</b> #ADFF2F		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>HoneyDew</b> #F0FFF0		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>HotPink</b> #FF69B4		<a href="#">Color Picker</a>
Color Mixer		<a href="#">Color Picker</a>
<b>IndianRed</b>		<a href="#">Color Picker</a>

Color Mixer	#CD5C5C	Color Picker
Color Mixer	<b>Indigo</b> #4B0082	Color Picker
Color Mixer	<b>Ivory</b> #FFFFFF0	Color Picker
Color Mixer	<b>Khaki</b> #F0E68C	Color Picker
Color Mixer	<b>Lavender</b> #E6E6FA	Color Picker
Color Mixer	<b>LavenderBlush</b> #FFF0F5	Color Picker
Color Mixer	<b>LawnGreen</b> #7CFC00	Color Picker
Color Mixer	<b>LemonChiffon</b> #FFFACD	Color Picker
Color Mixer	<b>LightBlue</b> #ADD8E6	Color Picker
Color Mixer	<b>LightCoral</b> #F08080	Color Picker
Color Mixer	<b>LightCyan</b> #E0FFFF	Color Picker
Color Mixer	<b>LightGoldenRodYellow</b> #FAFAD2	Color Picker
Color Mixer	<b>LightGray</b> #D3D3D3	Color Picker
Color Mixer	<b>LightGrey</b> #D3D3D3	Color Picker
Color Mixer	<b>LightGreen</b> #90EE90	Color Picker
Color Mixer	<b>LightPink</b> #FFB6C1	Color Picker



		<a href="#">Color Picker</a>
Color Mixer	<b>LightSalmon</b> #FFA07A	<a href="#">Color Picker</a>
Color Mixer	<b>LightSeaGreen</b> #20B2AA	<a href="#">Color Picker</a>
Color Mixer	<b>LightSkyBlue</b> #87CEFA	<a href="#">Color Picker</a>
Color Mixer	<b>LightSlateGray</b> #778899	<a href="#">Color Picker</a>
Color Mixer	<b>LightSlateGrey</b> #778899	<a href="#">Color Picker</a>
Color Mixer	<b>LightSteelBlue</b> #B0C4DE	<a href="#">Color Picker</a>
Color Mixer	<b>LightYellow</b> #FFFFE0	<a href="#">Color Picker</a>
Color Mixer	<b>Lime</b> #00FF00	<a href="#">Color Picker</a>
Color Mixer	<b>LimeGreen</b> #32CD32	<a href="#">Color Picker</a>
Color Mixer	<b>Linen</b> #FAF0E6	<a href="#">Color Picker</a>
Color Mixer	<b>Magenta</b> #FF00FF	<a href="#">Color Picker</a>
Color Mixer	<b>Maroon</b> #800000	<a href="#">Color Picker</a>
Color Mixer	<b>MediumAquaMarine</b> #66CDAA	<a href="#">Color Picker</a>
Color Mixer	<b>MediumBlue</b> #0000CD	<a href="#">Color Picker</a>
Color Mixer	<b>MediumOrchid</b> #BA55D3	<a href="#">Color Picker</a>
	<b>MediumPurple</b>	<a href="#">Color Picker</a>

Color Mixer	#9370DB	Color Picker
Color Mixer	MediumSeaGreen #3CB371	Color Picker
Color Mixer	MediumSlateBlue #7B68EE	Color Picker
Color Mixer	MediumSpringGreen #00FA9A	Color Picker
Color Mixer	MediumTurquoise #48D1CC	Color Picker
Color Mixer	MediumVioletRed #C71585	Color Picker
Color Mixer	MidnightBlue #191970	Color Picker
Color Mixer	MintCream #F5FFFA	Color Picker
Color Mixer	MistyRose #FFE4E1	Color Picker
Color Mixer	Moccasin #FFE4B5	Color Picker
Color Mixer	NavajoWhite #FFDEAD	Color Picker
Color Mixer	Navy #000080	Color Picker
Color Mixer	OldLace #FDF5E6	Color Picker
Color Mixer	Olive #808000	Color Picker
Color Mixer	OliveDrab #6B8E23	Color Picker
Color Mixer	Orange #FFA500	Color Picker



		<a href="#">Color Picker</a>
Color Mixer	<b>OrangeRed</b> #FF4500	<a href="#">Color Picker</a>
Color Mixer	<b>Orchid</b> #DA70D6	<a href="#">Color Picker</a>
Color Mixer	<b>PaleGoldenRod</b> #EEE8AA	<a href="#">Color Picker</a>
Color Mixer	<b>PaleGreen</b> #98FB98	<a href="#">Color Picker</a>
Color Mixer	<b>PaleTurquoise</b> #AFEEEE	<a href="#">Color Picker</a>
Color Mixer	<b>PaleVioletRed</b> #DB7093	<a href="#">Color Picker</a>
Color Mixer	<b>PapayaWhip</b> #FFEFD5	<a href="#">Color Picker</a>
Color Mixer	<b>PeachPuff</b> #FFDAB9	<a href="#">Color Picker</a>
Color Mixer	<b>Peru</b> #CD853F	<a href="#">Color Picker</a>
Color Mixer	<b>Pink</b> #FFC0CB	<a href="#">Color Picker</a>
Color Mixer	<b>Plum</b> #DDA0DD	<a href="#">Color Picker</a>
Color Mixer	<b>PowderBlue</b> #B0E0E6	<a href="#">Color Picker</a>
Color Mixer	<b>Purple</b> #800080	<a href="#">Color Picker</a>
Color Mixer	<b>RebeccaPurple</b> #663399	<a href="#">Color Picker</a>
Color Mixer	<b>Red</b> #FF0000	<a href="#">Color Picker</a>
	<b>RosyBrown</b>	



Color Mixer	#BC8F8F	Color Picker
Color Mixer	RoyalBlue #4169E1	Color Picker
Color Mixer	SaddleBrown #8B4513	Color Picker
Color Mixer	Salmon #FA8072	Color Picker
Color Mixer	SandyBrown #F4A460	Color Picker
Color Mixer	SeaGreen #2E8B57	Color Picker
Color Mixer	SeaShell #FFF5EE	Color Picker
Color Mixer	Sienna #A0522D	Color Picker
Color Mixer	Silver #C0C0C0	Color Picker
Color Mixer	SkyBlue #87CEEB	Color Picker
Color Mixer	SlateBlue #6A5ACD	Color Picker
Color Mixer	SlateGray #708090	Color Picker
Color Mixer	SlateGrey #708090	Color Picker
Color Mixer	Snow #FFFAFA	Color Picker
Color Mixer	SpringGreen #00FF7F	Color Picker
Color Mixer	SteelBlue #4682B4	Color Picker
	Tan	

Color Mixer	#D2B48C	Color Picker
Color Mixer	Teal #008080	Color Picker
Color Mixer	Thistle #D8BFD8	Color Picker
Color Mixer	Tomato #FF6347	Color Picker
Color Mixer	Turquoise #40E0D0	Color Picker
Color Mixer	Violet #EE82EE	Color Picker
Color Mixer	Wheat #F5DEB3	Color Picker
Color Mixer	White #FFFFFF	Color Picker
Color Mixer	WhiteSmoke #F5F5F5	Color Picker
Color Mixer	Yellow #FFFF00	Color Picker
Color Mixer	YellowGreen #9ACD32	Color Picker

## 13 HTML RGB and RGBA Colors

An RGB color value represents RED, GREEN, and BLUE light sources.

An RGBA color value is an extension of RGB with an Alpha channel (opacity).

### 13.1 RGB Color Values

In HTML, a color can be specified as an RGB value, using this formula:

***rgb(red, green, blue)***

Each parameter (red, green, and blue) defines the intensity of the color with a value between 0 and 255.

This means that there are  $256 \times 256 \times 256 = 16777216$  possible colors!

For example, `rgb(255, 0, 0)` is displayed as red, because red is set to its highest value (255), and the other two (green and blue) are set to 0.

Another example, `rgb(0, 255, 0)` is displayed as green, because green is set to its highest value (255), and the other two (red and blue) are set to 0.

To display black, set all color parameters to 0, like this: `rgb(0, 0, 0)`.

To display white, set all color parameters to 255, like this: `rgb(255, 255, 255)`.

Experiment by mixing the RGB values below:



`rgb(255, 99, 71)`

RED  
255  
GREEN  
99  
BLUE  
71

Example:

`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgb(60, 179, 113)`

`rgb(238, 130, 238)`

`rgb(255, 165, 0)`

`rgb(106, 90, 205)`

## 13.2 Shades of Gray

Shades of gray are often defined using equal values for all three parameters:

Example:

`rgb(60, 60, 60)`

`rgb(100, 100, 100)`

`rgb(140, 140, 140)`

`rgb(180, 180, 180)`

`rgb(200, 200, 200)`

`rgb(240, 240, 240)`

## 13.3 RGBA Color Values

RGBA color values are an extension of RGB color values with an Alpha channel - which specifies the opacity for a color.

An RGBA color value is specified with:

**`rgba(red, green, blue, alpha)`**

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Experiment by mixing the RGBA values below: `rgb(99, 71, 0.5)`

RED

255  
GREEN  
99  
BLUE  
71  
ALPHA  
0.5



### Example

```
rgba(255, 99, 71, 0)  
rgba(255, 99, 71, 0.2)  
rgba(255, 99, 71, 1)
```

## 14 HTML HEX Colors

A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.



### 14.1 HEX Color Values

In HTML, a color can be specified using a hexadecimal value in the form:

#### **#rrggb**

Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).

For example, #ff0000 is displayed as red, because red is set to its highest value (ff), and the other two (green and blue) are set to 00.

Another example, #00ff00 is displayed as green, because green is set to its highest value (ff), and the other two (red and blue) are set to 00.

To display black, set all color parameters to 00, like this: #000000.

To display white, set all color parameters to ff, like this: #ffffff.

Experiment by mixing the HEX values below:

	<b>#ff6347</b>
	RED
	ff
	GREEN
	63
	BLUE
	47





**Example:**

	<b>#ff0000</b>
	<b>#0000ff</b>
	<b>#3cb371</b>
	<b>#ee82ee</b>
	<b>#ffa500</b>
	<b>#6a5acd</b>

### 14.2 Shades of Gray

Shades of gray are often defined using equal values for all three parameters:

**Example:**

	<b>#404040</b>
	<b>#686868</b>
	<b>#a0a0a0</b>
	<b>#bebebe</b>



#dcdcdc

#f8f8f8



## 15 HTML HSL and HSLA Colors

HSL stands for hue, saturation, and lightness.

HSLA color values are an extension of HSL with an Alpha channel (opacity).



### 15.1 HSL Color Values

In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form:

***hsl(hue, saturation, lightness)***

Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.

Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.

Lightness is also a percentage value, 0% is black, and 100% is white.

Experiment by mixing the HSL values below:

<b>hsl(0, 100%, 50%)</b>
HUE
0
SATURATION
100%
LIGHTNESS
50%

**Example:**

<b>hsl(0, 100%, 50%)</b>
<b>hsl(240, 100%, 50%)</b>
<b>hsl(147, 50%, 47%)</b>
<b>hsl(300, 76%, 72%)</b>
<b>hsl(39, 100%, 50%)</b>
<b>hsl(248, 53%, 58%)</b>

### Saturation

Saturation can be described as the intensity of a color.

100% is pure color, no shades of gray

50% is 50% gray, but you can still see the color.

0% is completely gray, you can no longer see the color.

**Example:**

<b>hsl(0, 100%, 50%)</b>
<b>hsl(0, 80%, 50%)</b>
<b>hsl(0, 60%, 50%)</b>
<b>hsl(0, 40%, 50%)</b>
<b>hsl(0, 20%, 50%)</b>



```
hsl(0, 0%, 50%)
```

## Lightness

The lightness of a color can be described as how much light you want to give the color, where 0% means no light (black), 50% means 50% light (neither dark nor light) 100% means full lightness (white).



### Example:

<pre>hsl(0, 100%, 0%)</pre>
<pre>hsl(0, 100%, 25%)</pre>
<pre>hsl(0, 100%, 50%)</pre>
<pre>hsl(0, 100%, 75%)</pre>
<pre>hsl(0, 100%, 90%)</pre>
<pre>hsl(0, 100%, 100%)</pre>

## 15.2 Shades of Gray

Shades of gray are often defined by setting the hue and saturation to 0, and adjust the lightness from 0% to 100% to get darker/lighter shades:

### Example:

<pre>hsl(0, 0%, 20%)</pre>
<pre>hsl(0, 0%, 30%)</pre>
<pre>hsl(0, 0%, 40%)</pre>
<pre>hsl(0, 0%, 60%)</pre>
<pre>hsl(0, 0%, 70%)</pre>
<pre>hsl(0, 0%, 90%)</pre>

## 15.3 HSLA Color Values

HSLA color values are an extension of HSL color values with an Alpha channel - which specifies the opacity for a color.

An HSLA color value is specified with:

**`hsla(hue, saturation, lightness, alpha)`**

The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Experiment by mixing the HSLA values below:`100%, 50%, 0.5)`

```
HUE
0
SATURATION
100%
LIGHTNESS
50%
ALPHA
0.5
```

Example:

```
hsla(9, 100%, 64%, 0)  
hsla(9, 100%, 64%, 0.2)
```

```
hsla(9, 100%, 64%, 1)
```



# 16 HTML Styles - CSS

CSS stands for Cascading Style Sheets.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

**CSS = Styles and Colors:**

M a n i p u l a t e   T e x t  
C o l o r s ,   B o x e s



## 16.1 What is CSS?

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

**Tip:** The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

## 16.2 Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the `style` attribute inside HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using a `<link>` element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files. However, in this tutorial we will use inline and internal styles, because this is easier to demonstrate, and easier for you to try it yourself.

## 16.3 Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the `style` attribute of an HTML element.

The following example sets the text color of the `<h1>` element to blue, and the text color of the `<p>` element to red:

**Example:**

```
<h1 style="color:blue;">A Blue Heading</h1>
```

```
<p style="color:red;">A red paragraph.</p>
```

## 16.4 Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

The following example sets the text color of ALL the `<h1>` elements (on that page) to blue, and the text color of ALL the `<p>` elements to red. In addition, the page will be displayed with a "powderblue" background color:

**Example:**

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {background-color: powderblue;}
      h1 {color: blue;}
      p {color: red;}
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```



## 16.5 External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the `<head>` section of each HTML page:

### Example:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

"styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

**Tip:** With an external style sheet, you can change the look of an entire web site, by changing one file!

## 16.6 CSS Colors, Fonts and Sizes

Here, we will demonstrate some commonly used CSS properties. You will learn more about them later.

The CSS `color` property defines the text color to be used.

The CSS `font-family` property defines the font to be used.

The CSS `font-size` property defines the text size to be used.



### Example:

Use of CSS color, font-family and font-size properties:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

## 16.7 CSS Border

The CSS `border` property defines a border around an HTML element.

**Tip:** You can define a border for nearly all HTML elements.

### Example:

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
```

## 16.8 CSS Padding

The CSS `padding` property defines a padding (space) between the text and the border.

### Example:

Use of CSS border and padding properties:

```
p {  
  border: 2px solid powderblue;  
  padding: 30px;  
}
```



## 16.9 CSS Margin

The CSS `margin` property defines a margin (space) outside the border.

### Example:

Use of CSS border and margin properties:

```
p {  
  border: 2px solid powderblue;  
  margin: 50px;  
}
```

## 16.10 Link to External CSS

External style sheets can be referenced with a full URL or with a path relative to the current web page.

### Example:

This example uses a full URL to link to a style sheet:

```
<link rel="stylesheet" href="https://www.bintr.com/html/styles.css">
```

### Example:

This example links to a style sheet located in the html folder on the current web site:

```
<link rel="stylesheet" href="/html/styles.css">
```

### Example:

This example links to a style sheet located in the same folder as the current page:

```
<link rel="stylesheet" href="styles.css">
```

You can read more about file paths in the chapter [HTML File Paths](#).

## 16.11 Chapter Summary

- Use the HTML `style` attribute for inline styling
- Use the HTML `<style>` element to define internal CSS
- Use the HTML `<link>` element to refer to an external CSS file
- Use the HTML `<head>` element to store `<style>` and `<link>` elements
- Use the CSS `color` property for text colors
- Use the CSS `font-family` property for text fonts
- Use the CSS `font-size` property for text sizes
- Use the CSS `border` property for borders
- Use the CSS `padding` property for space inside the border

- Use the CSS `margin` property for space outside the border
- 

**Tip:** You can learn much more about CSS in our [Bintr CSS Tutorial](#).



## 17 HTML Style Tags



Tag	Description
<code>&lt;style&gt;</code>	Defines style information for an HTML document
<code>&lt;link&gt;</code>	Defines a link between a document and an external resource

### 17.1 HTML Links

Links are found in nearly all web pages. Links allow users to click their way from page to page.

### 17.2 Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

**Note:** A link does not have to be text. A link can be an image or any other HTML element!

### 17.3 Links - Syntax

The HTML `<a>` tag defines a hyperlink. It has the following syntax:

```
<a href="url">Link text</a>
```

The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

#### Example:

This example shows how to create a link to Bintr.online:

```
<a href="https://www.bintr.com/">Visit Bintr.com!</a>
```

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red
- 

**Tip:** Links can of course be styled with CSS, to get another look!



## 17.4 HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The `target` attribute specifies where to open the linked document.

The `target` attribute can have one of the following values:

- `_self` - Default. Opens the document in the same window/tab as it was clicked
- `_blank` - Opens the document in a new window or tab
- `_parent` - Opens the document in the parent frame
- `_top` - Opens the document in the full body of the window

### Example:

Use `target="_blank"` to open the linked document in a new browser window or tab:

```
<a href="https://www.bintr.com/" target="_blank">Visit Bintr!</a>
```



## 17.5 Absolute URLs vs. Relative URLs

Both examples above are using an **absolute URL** (a full web address) in the `href` attribute.

A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

### Example:

```
<h2>Absolute URLs</h2>
```

```
<p><a href="https://www..org/">C</a></p>
```

```
<p><a href="https://www.google.com/">Google</a></p>
```

```
<h2>Relative URLs</h2>
```

```
<p><a href="html_images.asp">HTML Images</a></p>
```

```
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

## 17.6 HTML Links - Use an Image as a Link

To use an image as a link, just put the `<img>` tag inside the `<a>` tag:

### Example:

```
<a href="default.asp">
```

```

```

```
</a>
```

## 17.7 Link to an Email Address

Use `mailto:` inside the `href` attribute to create a link that opens the user's email program (to let them send a new email):

**Example:**

```
<a href="mailto:someone@example.com">Send email</a>
```



## 17.8 Button as a Link

To use an HTML button as a link, you have to add some JavaScript code.

JavaScript allows you to specify what happens at certain events, such as a click of a button:

**Example:**

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

**Tip:** Learn more about JavaScript in our [JavaScript Tutorial](#).

## 17.9 Link Titles

The `title` attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

**Example:**

```
<a href="https://www.bintr.com/html/" title="Go to Bintr HTML section">Visit our HTML Tutorial</a>
```

## 17.10 More on Absolute URLs and Relative URLs

**Example:**

Use a full URL to link to a web page:

```
<a href="https://www.bintr.com/html/default.asp">HTML tutorial</a>
```

**Example:**

Link to a page located in the `html` folder on the current web site:

```
<a href="/html/default.asp">HTML tutorial</a>
```

**Example:**

Link to a page located in the same folder as the current page:

```
<a href="default.asp">HTML tutorial</a>
```

## 17.11 Chapter Summary

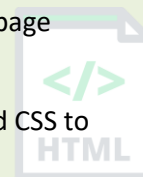
- Use the `<a>` element to define a link
- Use the `href` attribute to define the link address
- Use the `target` attribute to define where to open the linked document
- Use the `<img>` element (inside `<a>`) to use an image as a link
- Use the `mailto:` scheme inside the `href` attribute to create a link that opens the user's email program



## 18 HTML - Marquees

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML `<marquees>` tag.

**Note** – The `<marquee>` tag deprecated in HTML5. Do not use this element, instead you can use JavaScript and CSS to create such effects.



### 18.1 Syntax

A simple syntax to use HTML `<marquee>` tag is as follows –

```
<marquee attribute_name = "attribute_value" ....more attributes>
```

One or more lines or text message or image

```
</marquee>
```

### 18.2 The `<marquee>` Tag Attributes

Following is the list of important attributes which can be used with `<marquee>` tag.

Sr.No	Attribute & Description
1	<b>width</b> This specifies the width of the marquee. This can be a value like 10 or 20% etc.
2	<b>height</b> This specifies the height of the marquee. This can be a value like 10 or 20% etc.
3	<b>direction</b> This specifies the direction in which marquee should scroll. This can be a value like <i>up</i> , <i>down</i> , <i>left</i> or <i>right</i> .
4	<b>behavior</b> This specifies the type of scrolling of the marquee. This can have a value like <i>scroll</i> , <i>slide</i> and <i>alternate</i> .
5	<b>scrolldelay</b> This specifies how long to delay between each jump. This will have a value like 10 etc.
6	<b>scrollamount</b> This specifies the speed of marquee text. This can have a value like 10 etc.
7	<b>loop</b> This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
8	<b>bgcolor</b> This specifies background color in terms of color name or color hex value.
9	<b>hspace</b> This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
10	<b>vspace</b> This specifies vertical space around the marquee. This can be a value like 10 or 20% etc.

Below are few examples to demonstrate the usage of marquee tag.

## Examples - 1

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML marquee Tag</title>
  </head>
  <body>
    <marquee>This is basic example of marquee</marquee>
  </body>
</html>
```



## Examples - 2

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML marquee Tag</title>
  </head>
  <body>
    <marquee width = "50%">This example will take only 50% width</marquee>
  </body>
</html>
```

## Examples - 3

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML marquee Tag</title>
  </head>
  <body>
    <marquee direction = "right">This text will scroll from left to right</marquee>
  </body>
</html>
```

## Examples - 4

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML marquee Tag</title>
  </head>
  <body>
    <marquee direction = "up">This text will scroll from bottom to up</marquee>
  </body>
</html>
```

## 19 HTML Link Tags

Tag	Description
<code>&lt;a&gt;</code>	Defines a hyperlink



### 19.1 HTML Links - Different Colors

An HTML link is displayed in a different color depending on whether it has been visited, is unvisited, or is active.

### 19.2 HTML Link Colors

By default, a link will appear like this (in all browsers):

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

You can change the link state colors, by using CSS:

#### Example:

Here, an unvisited link will be green with no underline. A visited link will be pink with no underline. An active link will be yellow and underlined. In addition, when mousing over a link (`a:hover`) it will become red and underlined:

```
<style>
a:link {
  color: green;
  background-color: transparent;
  text-decoration: none;
}

a:visited {
  color: pink;
  background-color: transparent;
  text-decoration: none;
}

a:hover {
  color: red;
  background-color: transparent;
  text-decoration: underline;
}

a:active {
  color: yellow;
  background-color: transparent;
  text-decoration: underline;
}
</style>
```



## 19.3 Link Buttons

A link can also be styled as a button, by using CSS:

**This is a link**

**Example:**

```
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 15px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
</style>
```

To learn more about CSS, go to our [CSS Tutorial](#).

## 19.4 HTML Links - Create Bookmarks

HTML links can be used to create bookmarks, so that readers can jump to specific parts of a web page.

### 1.1.1 Create a Bookmark in HTML

Bookmarks can be useful if a web page is very long.

To create a bookmark - first create the bookmark, then add a link to it.

When the link is clicked, the page will scroll down or up to the location with the bookmark.

**Example:**

First, use the `id` attribute to create a bookmark:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

**Example:**

```
<a href="#C4">Jump to Chapter 4</a>
```

You can also add a link to a bookmark on another page:

`<a href="html_demo.html#C4">Jump to Chapter 4</a>`



## 19.5 Chapter Summary

- Use the `id` attribute (`id="value"`) to define bookmarks in a page
- Use the `href` attribute (`href="#value"`) to link to the bookmark



## 20 HTML Images

Images can improve the design and the appearance of a web page.



### Example

```

```

### Example

```

```

### Example

```

```

### 20.1 HTML Images Syntax

The HTML `<img>` tag is used to embed an image in a web page.

Images are not technically inserted into a web page; images are linked to web pages. The `<img>` tag creates a holding space for the referenced image.

The `<img>` tag is empty, it contains attributes only, and does not have a closing tag.

The `<img>` tag has two required attributes:

- `src` - Specifies the path to the image
- `alt` - Specifies an alternate text for the image

#### Syntax

```

```

### 20.2 The src Attribute

The required `src` attribute specifies the path (URL) to the image.

**Note:** When a web page loads, it is the browser, at that moment, that gets the image from a web server and inserts it into the page. Therefore, make sure that the image actually stays in the same spot in relation to the web page, otherwise your visitors will get a broken link icon. The broken link icon and the `alt` text are shown if the browser cannot find the image.

#### Example:

```

```

### 20.3 The alt Attribute

The required `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

The value of the `alt` attribute should describe the image:

**Example**

```

```

If a browser cannot find an image, it will display the value of the `alt` attribute:

**Example**

```

```



## 20.4 Image Size - Width and Height

You can use the `style` attribute to specify the width and height of an image.

**Example:**

```

```

Alternatively, you can use the `width` and `height` attributes:

**Example:**

```

```

The `width` and `height` attributes always define the width and height of the image in pixels.

## 20.5 Width and Height, or Style?

The `width`, `height`, and `style` attributes are all valid in HTML.

However, we suggest using the `style` attribute. It prevents styles sheets from changing the size of images:

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  width: 100%;
}
</style>
</head>
<body>
```

```

```

```

```

```
</body>
</html>
```

## 20.6 Images in Another Folder

If you have your images in a sub-folder, you must include the folder name in the `src` attribute:

**Example:**

```

```



## 20.7 Images on Another Server/Website

Some web sites point to an image on another server.

To point to an image on another server, you must specify an absolute (full) URL in the `src` attribute:

**Example:**

```

```

**Notes on external images:** External images might be under copyright. If you do not get permission to use it, you may be in violation of copyright laws. In addition, you cannot control external images; it can suddenly be removed or changed.

## 20.8 Animated Images

HTML allows animated GIFs:

**Example:**

```

```

## 20.9 Image as a Link

To use an image as a link, put the `<img>` tag inside the `<a>` tag:

**Example:**

```
<a href="default.asp">  
    
</a>
```

## 20.10 Image Floating

Use the CSS `float` property to let the image float to the right or to the left of a text:

**Example:**

```
<p>  
The image will float to the right of the text.</p>
```

```
<p>  
The image will float to the left of the text.</p>
```

**Tip:** To learn more about CSS Float, read our [CSS Float Tutorial](#).



## 20.11 Common Image Formats

Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
<b>APNG</b>	Animated Portable Network Graphics	.apng
<b>GIF</b>	Graphics Interchange Format	.gif
<b>ICO</b>	Microsoft Icon	.ico, .cur
<b>JPEG</b>	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .jpp
<b>PNG</b>	Portable Network Graphics	.png
<b>SVG</b>	Scalable Vector Graphics	.svg

## 20.12 Chapter Summary

- Use the HTML `<img>` element to define an image
- Use the HTML `src` attribute to define the URL of the image
- Use the HTML `alt` attribute to define an alternate text for an image, if it cannot be displayed
- Use the HTML `width` and `height` attributes or the CSS `width` and `height` properties to define the size of the image
- Use the CSS `float` property to let the image float to the left or to the right

**Note:** Loading large images takes time, and can slow down your web page. Use images carefully.

## 21 HTML Image Tags

Tag	Description
<code>&lt;img&gt;</code>	Defines an image
<code>&lt;map&gt;</code>	Defines an image map
<code>&lt;area&gt;</code>	Defines a clickable area inside an image map
<code>&lt;picture&gt;</code>	Defines a container for multiple image resources



### 21.1 HTML Image Maps

With HTML image maps, you can create clickable areas on an image.

#### Image Maps

The HTML `<map>` tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more `<area>` tags.

Try to click on the computer, phone, or the cup of coffee in the image below:

#### Example:

Here is the HTML source code for the image map above:



```

```

```
<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```



## 21.2 How Does it Work?

The idea behind an image map is that you should be able to perform different actions depending on where in the image you click.

To create an image map you need an image, and some HTML code that describes the clickable areas.

## 21.3 The Image

The image is inserted using the `<img>` tag. The only difference from other images is that you must add a `usemap` attribute:

```

```

The `usemap` value starts with a hash tag `#` followed by the name of the image map, and is used to create a relationship between the image and the image map.

**Tip:** You can use any image as an image map!

## 21.4 Create Image Map

Then, add a `<map>` element.

The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:

```
<map name="workmap">
```

The `name` attribute must have the same value as the `<img>`'s `usemap` attribute .

## 21.5 The Areas

Then, add the clickable areas.

A clickable area is defined using an `<area>` element.

### Shape

You must define the shape of the clickable area, and you can choose one of these values:

- `rect` - defines a rectangular region
- `circle` - defines a circular region
- `poly` - defines a polygonal region
- `default` - defines the entire region

You must also define some coordinates to be able to place the clickable area onto the image.

## Shape="rect"

The coordinates for `shape="rect"` come in pairs, one for the x-axis and one for the y-axis.

So, the coordinates `34,44` is located 34 pixels from the left margin and 44 pixels from the top:



The coordinates `270,350` is located 270 pixels from the left margin and 350 pixels from the top:



Now we have enough data to create a clickable rectangular area:

**Example:**



```
<area shape="rect" coords="34, 44, 270, 350" href="computer.htm">
```

This is the area that becomes clickable and will send the user to the page "computer.htm":



## Shape="circle"

To add a circle area, first locate the coordinates of the center of the circle:



337, 300



Then specify the radius of the circle:

44 pixels



Now you have enough data to create a clickable circular area:

**Example:**

```
<area shape="circle" coords="337, 300, 44" href="coffee.htm">
```

This is the area that becomes clickable and will send the user to the page "coffee.htm":



## Shape="poly"

The `shape="poly"` contains several coordinate points, which creates a shape formed with straight lines (a polygon).

This can be used to create any shape.

Like maybe a croissant shape!

How can we make the croissant in the image below become a clickable link?

We have to find the x and y coordinates for all edges of the croissant:

The coordinates come in pairs, one for the x-axis and one for the y-axis:

### Example:



```
<area shape="poly" coords="140,121,181,116,204,160,204,222,191,270,140,329,85,355,58,352,37,322,40,259,103,161,128,147" href="croissant.htm">
```

This is the area that becomes clickable and will send the user to the page "croissant.htm":



## 21.6 Image Map and JavaScript

A clickable area can also trigger a JavaScript function.

Add a `click` event to the `<area>` element to execute a JavaScript function:

### Example:

Here, we use the `onclick` attribute to execute a JavaScript function when the area is clicked:

```
<map name="workmap">
  <area shape="circle" coords="337,300,44" href="coffee.htm" onclick="myFunction()">
</map>

<script>
function myFunction() {
  alert("You clicked the coffee cup!");
}
</script>
```



## 21.7 Chapter Summary

- Use the HTML `<map>` element to define an image map
- Use the HTML `<area>` element to define the clickable areas in the image map
- Use the HTML `usemap` attribute of the `<img>` element to point to an image map

---

## 22 HTML Background Images

---

A background image can be specified for almost any HTML element.

### 22.1 Background Image on a HTML element

To add a background image on an HTML element, use the HTML `style` attribute and the CSS `background-image` property:

#### Example:

Add a background image on a HTML element:

```
<p style="background-image: url('img_girl.jpg');">
```

You can also specify the background image in the `<style>` element, in the `<head>` section:

#### Example:

Specify the background image in the `<style>` element:

```
<style>
p {
  background-image: url('img_girl.jpg');
}
</style>
```

## 22.2 Background Image on a Page

If you want the entire page to have a background image, you must specify the background image on the `<body>` element:

### Example:

Add a background image for the entire page:

```
<style>
body {
  background-image: url('img_girl.jpg');
}
</style>
```



## 22.3 Background Repeat

If the background image is smaller than the element, the image will repeat itself, horizontally and vertically, until it reaches the end of the element:

### Example:

```
<style>
body {
  background-image: url('example_img_girl.jpg');
}
</style>
```

To avoid the background image from repeating itself, set the `background-repeat` property to `no-repeat`.

### Example:

```
<style>
body {
  background-image: url('example_img_girl.jpg');
  background-repeat: no-repeat;
}
</style>
```

## 22.4 Background Cover

If you want the background image to cover the entire element, you can set the `background-size` property to `cover`.

Also, to make sure the entire element is always covered, set the `background-attachment` property to `fixed`:

This way, the background image will cover the entire element, with no stretching (the image will keep its original proportions):

### Example:

```
<style>
body {
  background-image: url('img_girl.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

```
background-size: cover;
}
```



## 22.5 Background Stretch

If you want the background image to stretch to fit the entire element, you can set the `background-size` property to `100%`:

Try resizing the browser window, and you will see that the image will stretch, but always cover the entire element.

### Example:

```
<style>
body {
background-image: url('img_girl.jpg');
background-repeat: no-repeat;
background-attachment: fixed;
background-size: 100% 100%;
}
```

## 22.6 Learn More CSS

From the examples above you have learned that background images can be styled by using the CSS background properties.

To learn more about CSS background properties, study our [CSS Background Tutorial](#).



## 23 HTML <picture> Element

The HTML <picture> element allows you to display different pictures for different devices or screen sizes.



### 23.1 The HTML <picture> Element

The HTML <picture> element gives web developers more flexibility in specifying image resources.

The <picture> element contains one or more <source> elements, each referring to different images through the *srcset* attribute. This way the browser can choose the image that best fits the current view and/or device.

Each <source> element has a *media* attribute that defines when the image is the most suitable.

#### Example:

Show different images for different screen sizes:

```
<picture>
  <source media="(min-width: 650px)" srcset="img_food.jpg">
  <source media="(min-width: 465px)" srcset="img_car.jpg">
  
</picture>
```

**Note:** Always specify an <img> element as the last child element of the <picture> element. The <img> element is used by browsers that do not support the <picture> element, or if none of the <source> tags match.

### 23.2 When to use the Picture Element

There are two main purposes for the <picture> element:

#### 1. Bandwidth

If you have a small screen or device, it is not necessary to load a large image file. The browser will use the first <source> element with matching attribute values, and ignore any of the following elements.

#### 2. Format Support

Some browsers or devices may not support all image formats. By using the <picture> element, you can add images of all formats, and the browser will use the first format it recognizes, and ignore any of the following elements.

#### Example:

The browser will use the first image format it recognizes:

```
<picture>
  <source srcset="img_avatar.png">
  <source srcset="img_girl.jpg">
  
</picture>
```



**Note:** The browser will use the first `<source>` element with matching attribute values, and ignore any following `<source>` elements.

## 24 HTML

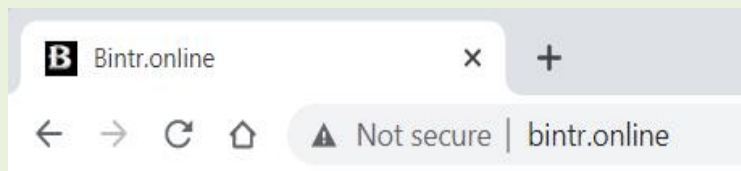
A favicon is a small image displayed to the page title in the browser tab.

### 24.1 How To Add a Favicon in HTML

You can use any image you like as your favicon. You can also create your own favicon on sites like <https://www.favicon.cc>.

**Tip:** A favicon is a small image, so it should be a simple image with high contrast.

A favicon image is displayed to the left of the page title in the browser tab, like this:



To add a favicon to your website, either save your favicon image to the root directory of your webserver, or create a folder in the root directory called images, and save your favicon image in this folder. A common name for a favicon image is "favicon.ico".

Add a `<link>` element to your "index.html" file, after the `<title>` element, like this:

**Example:**

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Now, save the "index.html" file and reload it in your browser. Your browser tab should now display your favicon image to the left of the page title.

### 24.2 Favicon File Format Support

The following table shows the file format support for a favicon image:

Browser	ICO	PNG	GIF	JPEG	SVG
Edge	Yes	Yes	Yes	Yes	Yes
Chrome	Yes	Yes	Yes	Yes	Yes
Firefox	Yes	Yes	Yes	Yes	Yes





Opera	Yes	Yes	Yes	Yes	Yes
Safari	Yes	Yes	Yes	Yes	Yes



## 24.3 Chapter Summary

- Use the HTML `<link>` element to insert a favicon

## 24.4 HTML Link Tag

Tag	Description
<code>&lt;link&gt;</code>	Defines the relationship between a document and an external resource

## 25 HTML Tables

HTML tables allow web developers to arrange data into rows and columns.



**Example:**

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

### 25.1 Define an HTML Table

A table in HTML consists of table cells inside rows and columns

**Example:**

A simple HTML table:

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

### 25.2 Table Cells

Each table cell is defined by a `<td>` and a `</td>` tag.

`td` stands for table data.

Everything between `<td>` and `</td>` are the content of the table cell.

## Example

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
</table>
```



**Note:** table data elements are the data containers of the table.

They can contain all sorts of HTML elements; text, images, lists, other tables, etc.

## 25.3 Table Rows

Each table row starts with a `<tr>` and end with a `</tr>` tag.

`tr` stands for table row.

### Example:

```
<table>
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
  <tr>
    <td>16</td>
    <td>14</td>
    <td>10</td>
  </tr>
</table>
```

You can have as many rows as you like in a table, just make sure that the number of cells are the same in each row.

**Note:** There are times where a row can have less or more cells than another. You will learn about that in a later chapter.

## 25.4 Table Headers

Sometimes you want your cells to be headers, in those cases use the `<th>` tag instead of the `<td>` tag:

### Example:

Let the first row be table headers:

```
<table>
  <tr>
    <th>Person 1</th>
    <th>Person 2</th>
    <th>Person 3</th>
  </tr>
  <tr>
```

```
<td>Emil</td>
<td>Tobias</td>
<td>Linus</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>10</td>
</tr>
</table>
```



By default, the text in `<th>` elements are bold and centred, but you can change that with CSS.

## 25.5 HTML Table Tags

Tag	Description
<code>&lt;table&gt;</code>	Defines a table
<code>&lt;th&gt;</code>	Defines a header cell in a table
<code>&lt;tr&gt;</code>	Defines a row in a table
<code>&lt;td&gt;</code>	Defines a cell in a table
<code>&lt;caption&gt;</code>	Defines a table caption
<code>&lt;colgroup&gt;</code>	Specifies a group of one or more columns in a table for formatting
<code>&lt;col&gt;</code>	Specifies column properties for each column within a <code>&lt;colgroup&gt;</code> element
<code>&lt;thead&gt;</code>	Groups the header content in a table
<code>&lt;tbody&gt;</code>	Groups the body content in a table
<code>&lt;tfoot&gt;</code>	Groups the footer content in a table

## 26 HTML Table Borders

HTML tables can have borders of different styles and shapes.



### 26.1 How To Add a Border

When you add a border to a table, you also add borders around each table cell:


To add a border, use the CSS `border` property on `table`, `th`, and `td` elements:

#### Example

```
table, th, td {  
  border: 1px solid black;  
}
```

### 26.2 Collapsed Table Borders

To avoid having double borders like in the example above, set the CSS `border-collapse` property to `collapse`.

This will make the borders collapse into a single border:


#### Example:

```
table, th, td {  
  border: 1px solid black;  
  border-collapse: collapse;  
}
```

### 26.3 Style Table Borders

If you set a background color of each cell, and give the border a white color (the same as the document background), you get the impression of an invisible border:

--	--	--


Example

```
table, th, td {  
    border: 1px solid white;  
    border-collapse: collapse;  
}  
th, td {  
    background-color: #96D4D4;  
}
```



## 26.4 Round Table Borders

With the `border-radius` property, the borders get rounded corners:


Example

```
table, th, td {  
    border: 1px solid black;  
    border-radius: 10px;  
}
```

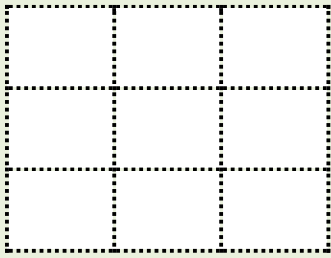
Skip the border around the table by leaving out `table` from the css selector:


Example

```
th, td {  
    border: 1px solid black;  
    border-radius: 10px;  
}
```

## 26.5 Dotted Table Borders


With the `border-style` property, you can set the appearance of the border.








The following values are allowed:

`dotted` 

`dashed` 

`solid` 

`double` 

`groove` 

`ridge` 

`inset` 

`outset` 

`none`

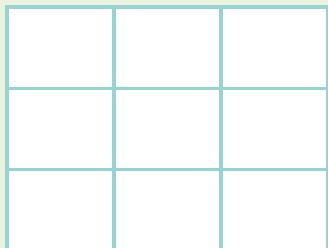
`hidden`

**Example:**

```
th, td {  
  border-style: dotted;  
}
```

## 26.6 Border Color

With the `border-color` property, you can set the color of the border.




Example

```
th, td {  
  border-color: #96D4D4;  
}
```

## 27 HTML Table Sizes

HTML tables can have different sizes for each column, row or the entire table.




Use the `style` attribute with the `width` or `height` properties to specify the size of a table, row or column.

### 27.1 HTML Table Width

To set the width of a table, add the `style` attribute to the `<table>` element:

#### Example:

Set the width of the table to 100%:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

**Note:** Using a percentage as the size unit for a width means how wide will this element be compared to its parent element, which in this case is the `<body>` element.



## 27.2 HTML Table Column Width




To set the size of a specific column, add the `style` attribute on a `<th>` or `<td>` element:

### Example:

Set the width of the first column to 70%:

```
<table style="width:100%">
  <tr>
    <th style="width:70%">Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

## 27.3 HTML Table Row Height


To set the height of a specific row, add the `style` attribute on a table row element:

### Example

Set the height of the second row to 200 pixels:

```
<table style="width:100%">
  <tr>
```

```
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr style="height:200px">
<td>Jill</td>
<td>Smith</td>
<td>50</td>
</tr>
<tr>
<td>Eve</td>
<td>Jackson</td>
<td>94</td>
</tr>
</table>
```



## 28 HTML Table Headers

EMIL	TOBIAS	LINUS

	MON	TUE	WED	THU	FRI
8:00					
9:00					
10:00					
11:00					
12:00					

. HTML tables can have headers for each column or row, or for many columns/rows

### 28.1 HTML Table Headers

Table headers are defined with `th` elements, each `th` element represents a table cell.

**Example:**

```
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

### 28.2 Vertical Table Headers

To use the first column as table headers, define the first cell in each row as a `th` element:

**Example:**

```
<table>
  <tr>
    <th>Firstname</th>
```

```
<td>Jill</td>
<td>Eve</td>
</tr>
<tr>
<th>Lastname</th>
<td>Smith</td>
<td>Jackson</td>
</tr>
<tr>
<th>Age</th>
<td>94</td>
<td>50</td>
</tr>
</table>
```



## 28.3 Align Table Headers

By default, table headers are bold and centered:

Firstname	Lastname	Age
Binod	Rabha	50
Eve	Roy	94

To left-align the table headers, use the CSS `text-align` property:

**Example:**

```
th {
  text-align: left;
}
```

## 28.4 Header for Multiple Columns

You can have a header that spans over two or more columns.

Name		Age
Jill	Rabha	50
Eve	Roy	94

To do this, use the `colspan` attribute on the `<th>` element:

**Example:**

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```



You will learn more about colspan and rowspan in the Table colspan & rowspan chapter.

## 28.5 Table Caption

You can add a caption that serves as a heading for the entire table.

Month	Savings
January	20,000 ₹
February	1,850 ₹

To add a caption to a table, use the `<caption>` tag:

**Example:**

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>20,000 ₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>1,850 ₹</td>
  </tr>
```

```
</tr>  
</table>
```

**Note:** The `<caption>` tag should be inserted immediately after the `<table>` tag.



## 29 HTML Table Padding & Spacing

HTML tables can adjust the padding inside the cells, and also the space between the cells.



With Padding		
hello	hello	hello
hello	hello	hello
hello	hello	hello

With Spacing		
hello	hello	hello
hello	hello	hello
hello	hello	hello

### 29.1 HTML Table - Cell Padding

Cell padding is the space between the cell edges and the cell content.

By default the padding is set to 0.

To add padding on table cells, use the CSS `padding` property:

**Example:**

```
th, td {  
    padding: 15px;  
}
```

To add padding only above the content, use the `padding-top` property.

And the others sides with the `padding-bottom`, `padding-left`, and `padding-right` properties:

**Example:**

```
th, td {  
    padding-top: 10px;  
    padding-bottom: 20px;  
    padding-left: 30px;
```

```
padding-right: 40px;  
}
```

## 29.2 HTML Table - Cell Spacing

Cell spacing is the space between each cell.

By default the space is set to 2 pixels.

To change the space between table cells, use the CSS `border-spacing` property on the `table` element:

### Example:

```
table {  
  border-spacing: 30px;  
}
```





## 30 HTML Table Colspan & Rowspan

HTML tables can have cells that spans over multiple rows and/or columns.



NAME		

APRIL		

2022		
FIESTA		

### 30.1 HTML Table - Colspan

To make a cell span over multiple columns, use the `colspan` attribute:

**Example:**

```
<table>
<tr>
  <th colspan="2">Name</th>
  <th>Age</th>
</tr>
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>43</td>
</tr>
<tr>
  <td>Eve</td>
  <td>Jackson</td>
  <td>57</td>
</tr>
</table>
```

**Note:** The value of the `colspan` attribute represents the number of columns to span.

### 30.2 HTML Table - Rowspan

To make a cell span over multiple rows, use the `rowspan` attribute:

**Example:**

```
<table>
<tr>
  <th>Name</th>
```

```
<td>Jill</td>
</tr>
<tr>
  <th rowspan="2">Phone</th>
  <td>555-1234</td>
</tr>
<tr>
  <td>555-8745</td>
</tr>
</table>
```



**Note:** The value of the `rowspan` attribute represents the number of rows to span.

## 31 HTML Table Styling

Use CSS to make your tables look better.



### 31.1 HTML Table - Zebra Stripes

If you add a background color on every other table row, you will get a nice zebra stripes effect.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

To style every other table row element, use the `:nth-child(even)` selector like this:

**Example:**

```
tr:nth-child(even) {  
  background-color: #D6EEEE;  
}
```

**Note:** If you use `(odd)` instead of `(even)`, the styling will occur on row 1,3,5 etc. instead of 2,4,6 etc.

### 31.2 HTML Table - Vertical Zebra Stripes

To make vertical zebra stripes, style every other *column*, instead of every other *row*.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

Set the `:nth-child(even)` for table data elements like this:

**Example:**

```
td:nth-child(even), th:nth-child(even) {
  background-color: #D6EEEE;
}
```

**Note:** Put the `:nth-child()` selector on both `th` and `td` elements if you want to have the styling on both headers and regular table cells.



### 31.3 Combine Vertical and Horizontal Zebra Stripes

You can combine the styling from the two examples above and you will have stripes on every other row and every other column.

If you use a transparent color you will get an overlapping effect.


Use an `rgba()` color to specify the transparency of the color:

**Example:**

```
tr:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}
```

```
th:nth-child(even),td:nth-child(even) {
  background-color: rgba(150, 212, 212, 0.4);
}
```

### 31.4 Horizontal Dividers

First Name	Last Name	Savings
Binod	Rabha	100 ₹
Bin	Toya	150 ₹
Joe	Das	300 ₹

If you specify borders only at the bottom of each table row, you will have a table with horizontal dividers.

Add the `border-bottom` property to all `tr` elements to get horizontal dividers:

**Example:**

```
tr {  
  border-bottom: 1px solid #ddd;  
}
```



## 31.5 Hoverable Table

Use the `:hover` selector on `tr` to highlight table rows on mouse over:

First Name	Last Name	Savings
Binod	Rabha	100 ₹
Bin	Toya	150 ₹
Joe	Das	300 ₹

**Example:**

```
tr:hover {background-color: #D6EEEE;}
```

## 32 HTML Table Colgroup

The `<colgroup>` element is used to style specific columns of a table.



### 32.1 HTML Table Colgroup

If you want to style the two first columns of a table, use the `<colgroup>` and `<col>` elements.

MON	TUE	WED	THU	FRI	SAT	SUN
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

The `<colgroup>` element should be used as a container for the column specifications.

Each group are specified with a `<col>` element.

The `span` attribute specifies how many columns that gets the style.

The `style` attribute specifies the style to give the columns.

#### Example:

```
<table>
  <colgroup>
    <col span="2" style="background-color: #D6EEEE">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
  ...
```

### 32.2 Colgroup

Add the a colgroup with a col element that spans over two columns to define a style for the two columns:

MON	TUE	WED	THU	FRI	SAT	SUN
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

22	23	24	25	26	27	28
----	----	----	----	----	----	----

**Note:** The `<colgroup>` tag must be a child of a `<table>` element and should be placed before any other table elements, like `<thead>`, `<tr>`, `<td>` etc., but after the `<caption>` element, if present.



### Legal CSS Properties

There are only a very limited selection of CSS properties that are allowed to be used in the `colgroup`:

- width property
- visibility property
- background properties
- border properties

All other CSS properties will have no effect on your tables.

## 32.3 Multiple Col Elements

If you want to style more columns with different styles, use more `<col>` elements inside the `<colgroup>`:

### Example:

```
<table>
  <colgroup>
    <col span="2" style="background-color: #D6EEEE">
    <col span="3" style="background-color: pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
```

...

## 32.4 Empty Colgroups

If you want to style columns in the middle of a table, insert a "empty" `<col>` element (with no styles) for the columns before:

### Example:

```
<table>
  <colgroup>
    <col span="3">
    <col span="2" style="background-color: pink">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
```

...

## 32.5 Hide Columns

You can hide columns with the `visibility: collapse` property

### Example:

```
<table>
  <colgroup>
    <col span="2">
    <col span="3" style="visibility: collapse">
  </colgroup>
  <tr>
    <th>MON</th>
    <th>TUE</th>
    <th>WED</th>
    <th>THU</th>
  </tr>
</table>
```

...





## 33 HTML Lists

HTML lists allow web developers to group a set of related items in lists.



### Example:

#### An unordered HTML list:

- Item
- Item
- Item
- Item

#### An ordered HTML list:

- 1 First item
- 2 Second item
- 3 Third item
- 4 Fourth item

### 33.1 Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with bullets (small black circles) by default:

#### Example:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

### 33.2 Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

#### Example:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

### 33.3 HTML Description Lists

HTML also supports description lists. A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

**Example:**

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```



## 33.4 HTML List Tags

TAG	DESCRIPTION
<code>&lt;UL&gt;</code>	Defines an unordered list
<code>&lt;OL&gt;</code>	Defines an ordered list
<code>&lt;LI&gt;</code>	Defines a list item
<code>&lt;DL&gt;</code>	Defines a description list
<code>&lt;DT&gt;</code>	Defines a term in a description list
<code>&lt;DD&gt;</code>	Describes the term in a description list

## 33.5 HTML Unordered Lists

The HTML `<ul>` tag defines an unordered (bulleted) list.

### Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.




The list items will be marked with bullets (small black circles) by default:

**Example**

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

## 33.6 Unordered HTML List - Choose List Item Marker

The CSS `list-style-type` property is used to define the style of the list item marker. It can have one of the following values:

 <p>HTML Tutorial BinTR.docx</p>	 <p>HTML Tutorial BinTR.docx</p>
 <p>HTML Tutorial BinTR.docx</p>	 <p>HTML Tutorial BinTR.docx</p>



#### Example:- Disc

```
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

#### Example: - Circle

```
<ul style="list-style-type:circle;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

#### Example:- Square

```
<ul style="list-style-type:square;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

#### Example:- None

```
<ul style="list-style-type:none;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

## 33.7 Nested HTML Lists

Lists can be nested (list inside list):

#### Example

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
</ul>
```

```
</li>
<li>Milk</li>
</ul>
```

**Note:** A list item (`<li>`) can contain a new list, and other HTML elements, like images and links, etc.



## 33.8 Horizontal List with CSS

HTML lists can be styled in many different ways with CSS.

One popular way is to style a list horizontally, to create a navigation menu:

### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}

li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>

<ul>
<li><a href="#home">Home</a></li>
<li><a href="#news">News</a></li>
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

**Tip:** You can learn much more about CSS in our [CSS Tutorial](#) (bintr).



## 33.9 Chapter Summary

- Use the HTML `<ul>` element to define an unordered list
- Use the CSS `list-style-type` property to define the list item marker
- Use the HTML `<li>` element to define a list item
- Lists can be nested
- List items can contain other HTML elements
- Use the CSS property `float:left` to display a list horizontally

## 34 HTML Ordered Lists

The HTML `<ol>` tag defines an ordered list. An ordered list can be numerical or alphabetical.



### 34.1 Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag.

The list items will be marked with numbers by default:

**Example:**

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

### 34.2 Ordered HTML List - The Type Attribute

The `type` attribute of the `<ol>` tag, defines the type of the list item marker:

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)
<code>type="A"</code>	The list items will be numbered with uppercase letters
<code>type="a"</code>	The list items will be numbered with lowercase letters
<code>type="I"</code>	The list items will be numbered with uppercase roman numbers
<code>type="i"</code>	The list items will be numbered with lowercase roman numbers

**Numbers:**

```
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

**Uppercase Letters:**

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

**Lowercase Letters:**

```
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
```

```
<li>Milk</li>
</ol>
```

#### Uppercase Roman Numbers:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

#### Lowercase Roman Numbers:

```
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



## 34.3 Control List Counting

By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the **start** attribute:

#### Example:

```
<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

## 34.4 Nested HTML Lists

Lists can be nested (list inside list):

#### Example:

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ol>
      <li>Black tea</li>
      <li>Green tea</li>
    </ol>
  </li>
  <li>Milk</li>
</ol>
```

**Note:** A list item (**<li>**) can contain a new list, and other HTML elements, like images and links, etc.

## 34.5 Chapter Summary

- Use the HTML `<ol>` element to define an ordered list
- Use the HTML `type` attribute to define the numbering type
- Use the HTML `<li>` element to define a list item
- Lists can be nested
- List items can contain other HTML elements



## 34.6 HTML Other Lists

HTML also supports description lists.

### 34.6.1 HTML Description Lists

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

**Example:**

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

## 34.7 Chapter Summary

- Use the HTML `<dl>` element to define a description list
- Use the HTML `<dt>` element to define the description term
- Use the HTML `<dd>` element to describe the term in a description list



## 35 HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.

There are two display values: block and inline.



### 35.1 Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Two commonly used block elements are: `<p>` and `<div>`.

- The `<p>` element defines a paragraph in an HTML document.
- The `<div>` element defines a division or a section in an HTML document.
  - The `<p>` element is a block-level element.
  - The `<div>` element is a block-level element.

#### Example:

```
<p>Hello World</p>
<div>Hello World</div>
```

Here are the block-level elements in HTML:

```
<address>
<article>
<aside>
<blockquote>
<canvas>
<dd>
<div>
<dl>
<dt>
<fieldset>
<figcaption>
<figure>
<footer>
<form>
<h1>-<h6>
<header>
<hr>
<li>
<main>
<nav>
<noscript>
<ol>
<p>
<pre>
<section>
<table>
<tfoot>
<ul>
<video>
```

## 35.2 Inline Elements

- An inline element does not start on a new line.
- An inline element only takes up as much width as necessary.
- This is a <span> element inside a paragraph.



### Example:

```
<span>Hello World</span>
```

Here are the inline elements in HTML:

```
<a>  
<abbr>  
<acronym>  
<b>  
<bdo>  
<big>  
<br>  
<button>  
<cite>  
<code>  
<dfn>  
<em>  
<i>  
<img>  
<input>  
<kbd>  
<label>  
<map>  
<object>  
<output>  
<q>  
<samp>  
<script>  
<select>  
<small>  
<span>  
<strong>  
<sub>  
<sup>  
<textarea>  
<time>  
<tt>  
<var>
```

**Note:** An inline element cannot contain a block-level element!

## 35.3 The <div> Element

The `<div>` element is often used as a container for other HTML elements.

The `<div>` element has no required attributes, but `style`, `class` and `id` are common.

When used together with CSS, the `<div>` element can be used to style blocks of content:

**Example:**

```
<div style="background-color:black;color:white;padding:20px;">
  <h2>London</h2>
  <p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</div>
```



## 35.4 The <span> Element

The `<span>` element is an inline container used to mark up a part of a text, or a part of a document.

The `<span>` element has no required attributes, but `style`, `class` and `id` are common.

When used together with CSS, the `<span>` element can be used to style parts of the text:

**Example:**

```
<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my father has <span style="color:darkolivegreen;font-weight:bold">dark green</span> eyes.</p>
```

## 35.5 Chapter Summary

- There are two display values: block and inline
- A block-level element always starts on a new line and takes up the full width available
- An inline element does not start on a new line and it only takes up as much width as necessary
- The `<div>` element is a block-level and is often used as a container for other HTML elements
- The `<span>` element is an inline container used to mark up a part of a text, or a part of a document

Tag	Description
<code>&lt;div&gt;</code>	Defines a section in a document (block-level)
<code>&lt;span&gt;</code>	Defines a section in a document (inline)

## 36 HTML class Attribute

The HTML `class` attribute is used to specify a class for an HTML element. Multiple HTML elements can share the same class.



### 36.1 Using The class Attribute

The `class` attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.

In the following example we have three `<div>` elements with a `class` attribute with the value of "city". All of the three `<div>` elements will be styled equally according to the `city` style definition in the head section:

#### Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  border: 2px solid black;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="city">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="city">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="city">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```

In the following example we have two `<span>` elements with a `class` attribute with the value of "note". Both `<span>` elements will be styled equally according to the `note` style definition in the head section:

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
.note {
  font-size: 120%;
  color: red;
}
</style>
</head>
<body>

<h1>My <span class="note">Important</span> Heading</h1>
<p>This is some <span class="note">important</span> text.</p>

</body>
</html>
```



**Tip:** The `class` attribute can be used on **any** HTML element.

**Note:** The class name is case sensitive!

**Tip:** You can learn much more about CSS in our [CSS Tutorial](#).

## 36.2 The Syntax For Class

To create a class; write a period (.) character, followed by a class name. Then, define the CSS properties within curly braces {}:

**Example:**

Create a class named "city":

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>
```

```
<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>

</body>
</html>
```



## 36.3 Multiple Classes

HTML elements can belong to more than one class.

To define multiple classes, separate the class names with a space, e.g. `<div class="city main">`. The element will be styled according to all the classes specified.

In the following example, the first `<h2>` element belongs to both the `city` class and also to the `main` class, and will get the CSS styles from both of the classes:

**Example:**

```
<h2 class="city main">London</h2>
<h2 class="city">Paris</h2>
<h2 class="city">Tokyo</h2>
```

## 36.4 Different Elements Can Share Same Class

Different HTML elements can point to the same class name.

In the following example, both `<h2>` and `<p>` points to the "city" class and will share the same style:

**Example:**

```
<h2 class="city">Paris</h2>
<p class="city">Paris is the capital of France</p>
```

## 36.5 Use of The class Attribute in JavaScript

The class name can also be used by JavaScript to perform certain tasks for specific elements.

JavaScript can access elements with a specific class name with the `getElementsByClassName()` method:

**Example**

Click on a button to hide all elements with the class name "city":

```
<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
```

Don't worry if you don't understand the code in the example above.

You will learn more about JavaScript in our [HTML JavaScript](#) chapter, or you can study our [JavaScript Tutorial](#).

## 36.6 Chapter Summary



- The HTML **class** attribute specifies one or more class names for an element
- Classes are used by CSS and JavaScript to select and access specific elements
- The **class** attribute can be used on any HTML element
- The class name is case sensitive
- Different HTML elements can point to the same class name
- JavaScript can access elements with a specific class name with the **getElementsByClassName()** method

## 37 HTML id Attribute

The HTML `id` attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.



### 37.1 Using The id Attribute

The `id` attribute specifies a unique id for an HTML element. The value of the `id` attribute must be unique within the HTML document.

The `id` attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

**The syntax for id is:** write a hash character (#), followed by an id name. Then, define the CSS properties within curly braces {}.

In the following example we have an `<h1>` element that points to the id name "myHeader". This `<h1>` element will be styled according to the `#myHeader` style definition in the head section:

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<style>
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}
</style>
</head>
<body>

<h1 id="myHeader">My Header</h1>

</body>
</html>
```

**Note:** The id name is case sensitive!

**Note:** The id name must contain at least one character, cannot start with a number, and must not contain whitespaces (spaces, tabs, etc.).

### 37.2 Difference Between Class and ID

A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page:

**Example:**



```
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

**Tip:** You can learn much more about CSS in our [CSS Tutorial](#).

## 37.3 HTML Bookmarks with ID and Links

HTML bookmarks are used to allow readers to jump to specific parts of a webpage.

Bookmarks can be useful if your page is very long.

To use a bookmark, you must first create it, and then add a link to it.

Then, when the link is clicked, the page will scroll to the location with the bookmark.

### Example

First, create a bookmark with the `id` attribute:

```
<h2 id="C4">Chapter 4</h2>
```

Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:

### Example:

```
<a href="#C4">Jump to Chapter 4</a>
```

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

```
<a href="html_demo.html#C4">Jump to Chapter 4</a>
```



## 37.4 Using The id Attribute in JavaScript

The `id` attribute can also be used by JavaScript to perform some tasks for that specific element.

JavaScript can access an element with a specific id with the `getElementById()` method:

### Example

Use the `id` attribute to manipulate text with JavaScript:

```
<script>
function displayResult() {
    document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

**Tip:** Study JavaScript in the [HTML JavaScript](#) chapter, or in our [JavaScript Tutorial](#).

## 37.5 Chapter Summary

- The `id` attribute is used to specify a unique id for an HTML element
- The value of the `id` attribute must be unique within the HTML document
- The `id` attribute is used by CSS and JavaScript to style/select a specific element
- The value of the `id` attribute is case sensitive
- The `id` attribute is also used to create HTML bookmarks
- JavaScript can access an element with a specific id with the `getElementById()` method



## 38 HTML Iframes

An HTML iframe is used to display a web page within a web page.



Tag	Description
<code>&lt;iframe&gt;</code>	Defines an inline frame

### 38.1 HTML Iframe Syntax

The HTML `<iframe>` tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

#### Syntax

```
<iframe src="url" title="description"></iframe>
```

**Tip:** It is a good practice to always include a `title` attribute for the `<iframe>`. This is used by screen readers to read out what the content of the iframe is.

### 38.2 Iframe - Set Height and Width

Use the `height` and `width` attributes to specify the size of the iframe. The height and width are specified in pixels by default:

#### Example:

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>
```

Or you can add the `style` attribute and use the CSS `height` and `width` properties:

#### Example:

```
<iframe src="demo_iframe.htm" style="height:200px;width:300px;" title="Iframe Example"></iframe>
```

### 38.3 Iframe - Remove the Border

By default, an iframe has a border around it. To remove the border, add the `style` attribute and use the CSS `border` property:

#### Example:

```
<iframe src="demo_iframe.htm" style="border:none;" title="Iframe Example"></iframe>
```

With CSS, you can also change the size, style and color of the iframe's border:

#### Example:

```
<iframe src="demo_iframe.htm" style="border:2px solid red;" title="Iframe Example"></iframe>
```

## 38.4 Iframe - Target for a Link

An iframe can be used as the target frame for a link. The `target` attribute of the link must refer to the `name` attribute of the iframe:

### Example:

```
<iframe src="demo_iframe.htm" name="iframe_a" title="Iframe Example"></iframe>
```

```
<p><a href="https://www.bintr.com" target="iframe_a">Bintr.com</a></p>
```



## 38.5 Chapter Summary

- The HTML `<iframe>` tag specifies an inline frame
- The `src` attribute defines the URL of the page to embed
- Always include a `title` attribute (for screen readers)
- The `height` and `width` attributes specifies the size of the iframe
- Use `border:none;` to remove the border around the iframe

## 39 HTML JavaScript

JavaScript makes HTML pages more dynamic and interactive.

### Example:

**My First JavaScript:** Click me to display Date and Time→

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First JavaScript</h1>
    <button type="button"
      onclick="document.getElementById('demo').innerHTML = Date()">
      Click me to display Date and Time.</button>
    <p id="demo"></p>
  </body>
</html>
```



### 39.1 The HTML <script> Tag

The HTML `<script>` tag is used to define a client-side script (JavaScript).

The `<script>` element either contains script statements, or it points to an external script file through the `src` attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript most often uses the `document.getElementById()` method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with `id="demo"`:

### Example:

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

**Tip:** You can learn much more about JavaScript in our JavaScript Tutorial.

### 39.2 A Taste of JavaScript

Here are some examples of what JavaScript can do:

### Example:

JavaScript can change content:

```
document.getElementById("demo").innerHTML = "Hello JavaScript!";
```

```
<!DOCTYPE html>
<html>
<body>

<h1>My First JavaScript</h1>
```

```
<p>JavaScript can change the content of an HTML element:</p>
<button type="button" onclick="myFunction()">Click Me!</button>
<p id="demo">This is a demonstration.</p>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
</body>
</html>
```



**Example:**

### JavaScript can change styles:

```
document.getElementById("demo").style.fontSize = "25px";
document.getElementById("demo").style.color = "red";
document.getElementById("demo").style.backgroundColor = "yellow";
```

**Example:**

### JavaScript can change attributes:

```
document.getElementById("image").src = "picture.gif";
```

## 39.3 The HTML `<noscript>` Tag

The HTML `<noscript>` tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts:

**Example:**

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
<noscript>Sorry, your browser does not support JavaScript!</noscript>
```

### Script Tags

Tag	Description
<code>&lt;script&gt;</code>	Defines a client-side script
<code>&lt;noscript&gt;</code>	Defines an alternate content for users that do not support client-side scripts

---

## 40 HTML File Paths

---

A file path describes the location of a file in a web site's folder structure.

### File Path Examples

Path	Description
<code>&lt;img src="picture.jpg"&gt;</code>	The "picture.jpg" file is located in the same folder as the current page
<code>&lt;img src="images/picture.jpg"&gt;</code>	The "picture.jpg" file is located in the images folder in the current folder
<code>&lt;img src="/images/picture.jpg"&gt;</code>	The "picture.jpg" file is located in the images folder at the root of the current web
<code>&lt;img src="../picture.jpg"&gt;</code>	The "picture.jpg" file is located in the folder one level up from the current folder



## 40.1 HTML File Paths

A file path describes the location of a file in a web site's folder structure.

File paths are used when linking to external files, like:

- Web pages
- Images
- Style sheets
- JavaScripts

## 40.2 Absolute File Paths

An absolute file path is the full URL to a file:

### Example

```

```

The `<img>` tag is explained in the [HTML Images](#) chapter .

## 40.3 Relative File Paths

A relative file path points to a file relative to the current page. In the following example, the file path points to a file in the images folder located at the root of the current web:

### Example:

```

```

In the following example, the file path points to a file in the images folder located in the current folder:

### Example:

```

```

In the following example, the file path points to a file in the images folder located in the folder one level up from the current folder:

**Example:**

```

```

**40.3.1 Best Practice**

It is best practice to use relative file paths (if possible).

When using relative file paths, your web pages will not be bound to your current base URL. All links will work on your own computer (localhost) as well as on your current public domain and your future public domains.





## 41 HTML - The Head Element

The HTML `<head>` element is a container for the following elements: `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, and `<base>`.



### 41.1 The HTML `<head>` Element

The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.

HTML metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

### 41.2 The HTML `<title>` Element

The `<title>` element defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The `<title>` element is required in HTML documents!

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

**The `<title>` element:**

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search engine-results

So, try to make the title as accurate and meaningful as possible!

**A simple HTML document:**

**Example:**

```
<!DOCTYPE html>
<html>
<head>
  <title>A Meaningful Page Title</title>
</head>
<body>
```

The content of the document.....

```
</body>
</html>
```

## 41.3 The HTML <style> Element

The <style> element is used to define style information for a single HTML page:

### Example:

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```



## 41.4 The HTML <link> Element

The <link> element defines the relationship between the current document and an external resource.

The <link> tag is most often used to link to external style sheets:

### Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
  <link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

**Tip:** To learn all about CSS, visit our [CSS Tutorial](#).

## 41.5 The HTML <meta> Element

The <meta> element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but are used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

## 41.6 Examples

**Define the character set used:**

```
<meta charset="UTF-8">
```

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define a description of your web page:

```
<meta name="description" content="Free Web tutorials">
```

Define the author of a page:

```
<meta name="author" content="John Doe">
```

Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

Setting the viewport to make your website look good on all devices:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```



Example of **<meta>** tags:

Example:

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="Binod Rabha">
```

## 41.7 Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following **<meta>** element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This gives the browser instructions on how to control the page's dimensions and scaling.

The **width=device-width** part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The **initial-scale=1.0** part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:



Without the viewport meta tag



With the viewport meta tag



**Tip:** If you are browsing this page with a phone or a tablet, you can click on the two links below to see the difference.

## 41.8 The HTML `<script>` Element

The `<script>` element is used to define client-side JavaScripts.

The following JavaScript writes "Hello JavaScript!" into an HTML element with id="demo":

**Example:**

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
  <script>
    function myFunction() {
      document.getElementById("demo").innerHTML = "Hello JavaScript!";
    }
  </script>
</head>
<body>
<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

```
<script>
function myFunction() {
```

```
document.getElementById("demo").innerHTML = "Hello JavaScript!";
}
</script>
```

**Tip:** To learn all about JavaScript, visit our [JavaScript Tutorial](#).



## 41.9 The HTML <base> Element

The <base> element specifies the base URL and/or target for all relative URLs in a page.

The <base> tag must have either an href or a target attribute present, or both.

There can only be one single <base> element in a document!

### Example:

Specify a default URL and a default target for all links on a page:

```
<head>
<base href="https://www.bintr.com/" target="_blank">
</head>

<body>

<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```

## 41.10 Chapter Summary

- The <head> element is a container for metadata (data about data)
- The <head> element is placed between the <html> tag and the <body> tag
- The <title> element is required and it defines the title of the document
- The <style> element is used to define style information for a single document
- The <link> tag is most often used to link to external style sheets
- The <meta> element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings
- The <script> element is used to define client-side JavaScripts
- The <base> element specifies the base URL and/or target for all relative URLs in a page

## 41.11 HTML head Elements

Tag	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<base>	Defines a default address or a default target for all links on a page

<code>&lt;link&gt;</code>	Defines the relationship between a document and an external resource
<code>&lt;meta&gt;</code>	Defines metadata about an HTML document
<code>&lt;script&gt;</code>	Defines a client-side script
<code>&lt;style&gt;</code>	Defines style information for a document



## 42 HTML Layout Elements and Techniques

Websites often display content in multiple columns (like a magazine or a newspaper).



Example:

### Cities

- London
- Paris
- Tokyo

### London

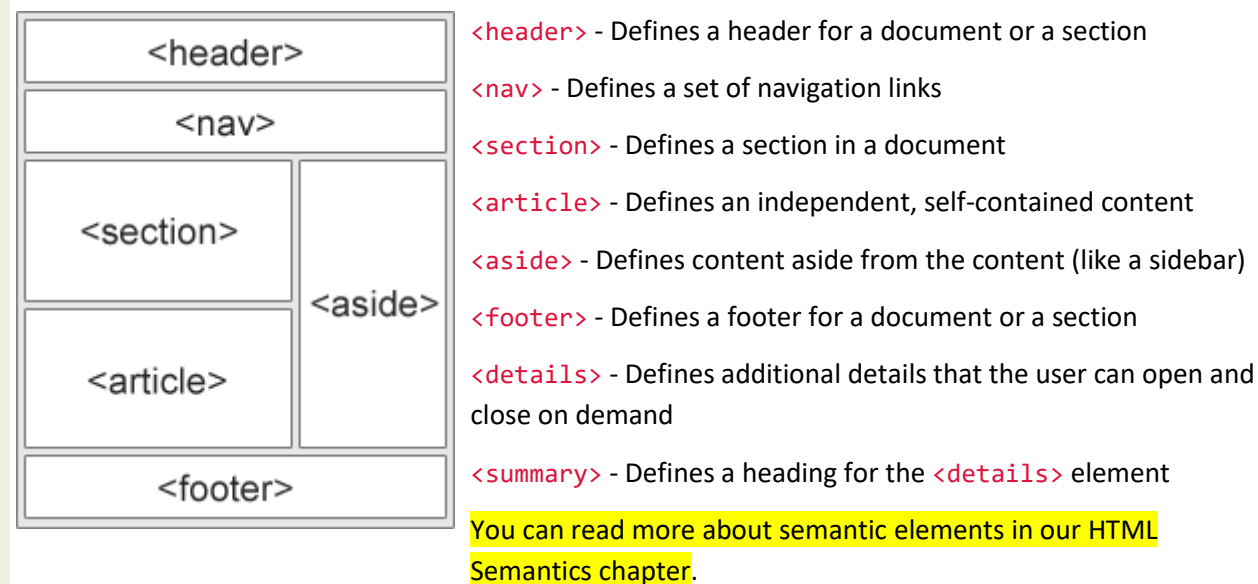
London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Footer

### 42.1 HTML Layout Elements

HTML has several semantic elements that define the different parts of a web page:



### 42.2 HTML Layout Techniques

There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:

- CSS framework
- CSS float property
- CSS flexbox
- CSS grid

## 42.3 CSS Frameworks

If you want to create your layout fast, you can use a CSS framework, like CSS or Bootstrap. Here you can create your website from scratch or use a template, and host it for free.

*\* no credit card required*



## 42.4 CSS Float Layout

It is common to do entire web layouts using the CSS `float` property. Float is easy to learn - you just need to remember how the `float` and `clear` properties work.

**Disadvantages:** Floating elements are tied to the document flow, which may harm the flexibility. Learn more about float in our CSS Float and Clear chapter.

## 42.5 CSS Flexbox Layout

Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.

Learn more about flexbox in our CSS Flexbox chapter.

## 42.6 CSS Grid Layout

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns, making it easier to design web pages without having to use floats and positioning.

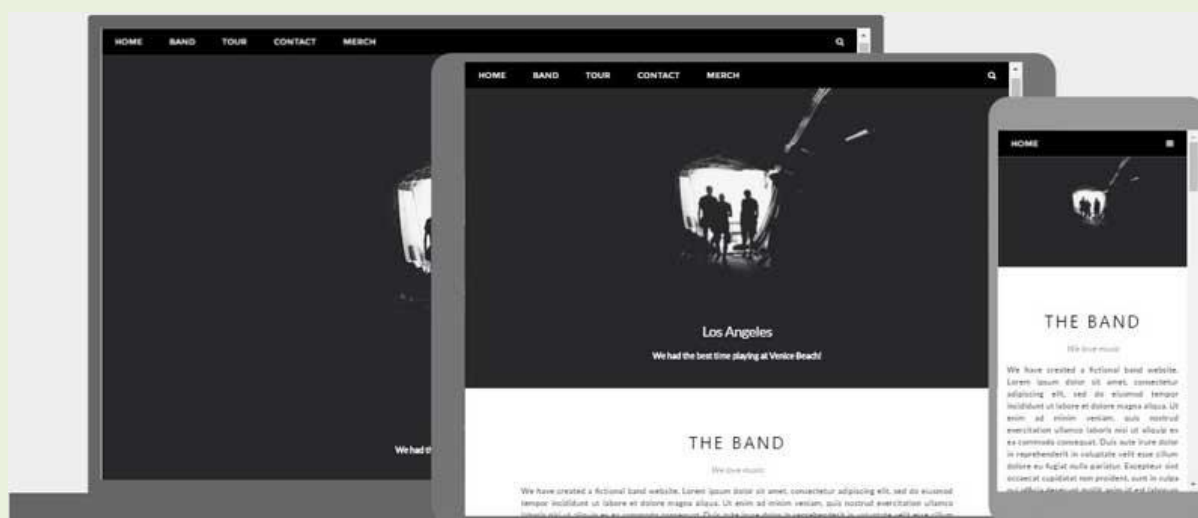
Learn more about CSS grids in our CSS Grid Intro chapter.



## 43 HTML Responsive Web Design

Responsive web design is about creating web pages that look good on all devices!

A responsive web design will automatically adjust for different screen sizes and viewports.



### 43.1 What is Responsive Web Design?

Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
* {
  box-sizing: border-box;
}
.menu {
  float: left;
  width: 20%;
}
.menuitem {
  padding: 8px;
  margin-top: 7px;
  border-bottom: 1px solid #f1f1f1;
}
.main {
  float: left;
  width: 60%;
  padding: 0 20px;
  overflow: hidden;
}
.right {
  background-color: lightblue;
  float: left;
  width: 20%;
```

```
padding: 10px 15px;
margin-top: 7px;
}
```

```
@media only screen and (max-width:800px) {
/* For tablets: */
.main {
width: 80%;
padding: 0;
}
.right {
width: 100%;
}
}
```

```
@media only screen and (max-width:500px) {
/* For mobile phones: */
.menu, .main, .right {
width: 100%;
}
}
```

```
</style>
```

```
</head>
```

```
<body style="font-family:Verdana;">
```

```
<div style="background-color:#f1f1f1;padding:15px;">
```

```
<h1>Binod Toya Rabha </h1>
```

```
<h3>Please visit website ( bintr.online )</h3>
```

```
</div>
```

```
<div style="overflow:auto">
```

```
<div class="menu">
```

```
<div class="menuitem">The Walk</div>
```

```
<div class="menuitem">Transport</div>
```

```
<div class="menuitem">History</div>
```

```
<div class="menuitem">Gallery</div>
```

```
</div>
```

```
<div class="main">
```

```
<h2>The Walk</h2>
```

```
<p>The walk from Monterosso to Riomaggiore will take you approximately two hours, give or take an hour depending on the weather conditions and your physical shape.</p>
```

```

```

```
</div>
```

```
<div class="right">
```

```
<h2>What?</h2>
```

```
<p>Cinque Terre comprises five villages: Monterosso, Vernazza, Corniglia, Manarola, and Riomaggiore.</p>
```

```
<h2>Where?</h2>
```

```
<p>On the northwest cost of the Italian Riviera, north of the city La Spezia.</p>
```

```
<h2>Price?</h2>
```

```
<p>The Walk is free!</p>
```

```
</div>
```

```
</div>
```



```
<div style="background-color:#f1f1f1;text-align:center;padding:10px;margin-top:7px;font-size:12px;"> This web page is a part of a demonstration of fluid web design made by bintr.online Resize the browser window to see the content respond to the resizing.</div>
```

```
</body>
</html>
```



## 43.2 Setting The Viewport

To create a responsive website, add the following `<meta>` tag to all your web pages:

**Example:**

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This will set the viewport of your page, which will give the browser instructions on how to control the page's dimensions and scaling.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

**Tip:** If you are browsing this page on a phone or a tablet, you can click on the two links above to see the difference.

Without the viewport meta tag



With the viewport meta tag:



## 43.3 Responsive Images

Responsive images are images that scale nicely to fit any browser size.

### Using the width Property

If the CSS `width` property is set to 100%, the image will be responsive and scale up and down:



Example:

```

```

Notice that in the example above, the image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the `max-width` property instead.

### Using the max-width Property

If the `max-width` property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:



Example:

```

```

## Show Different Images Depending on Browser Width

The HTML `<picture>` element allows you to define different images for different browser window sizes.

Resize the browser window to see how the image below change depending on the width:



Example:

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-
width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width:
1500px)">
  <source srcset="flowers.jpg">
  
</picture>
```

## Responsive Text Size

The text size can be set with a "vw" unit, which means the "viewport width".

That way the text size will follow the size of the browser window:

---

## 44 Hello World

---

Resize the browser window to see how the text size scales.

Example:

```
<h1 style="font-size:10vw">Hello World</h1>
```

Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

### 44.1 Media Queries

In addition to resize text and images, it is also common to use media queries in responsive web pages.

With media queries you can define completely different styles for different browser sizes.

**Example:** resize the browser window to see that the three div elements below will display horizontally on large screens and stacked vertically on small screens:

Left Menu

Main Content

## Right Content

### Example:

```
<style>
.left, .right {
  float: left;
  width: 20%; /* The width is 20%, by default */
}

.main {
  float: left;
  width: 60%; /* The width is 60%, by default */
}

/* Use a media query to add a breakpoint at 800px: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
  }
}
</style>
```



**Tip:** To learn more about Media Queries and Responsive Web Design, read [RWD Tutorial](#).

### Responsive Web Page - Full Example

A responsive web page should look good on large desktop screens and on small mobile phones.

### Responsive Web Design - Frameworks

All popular CSS Frameworks offer responsive design. They are free, and easy to use.

## 44.2 .CSS

.CSS is a modern CSS framework with support for desktop, tablet, and mobile design by default.

.CSS is smaller and faster than similar CSS frameworks.

.CSS is designed to be a high quality alternative to Bootstrap.

.CSS is designed to be independent of jQuery or any other JavaScript library.

## 44.3 Bootstrap

Another popular CSS framework is Bootstrap. Bootstrap uses HTML, CSS and jQuery to make responsive web pages.

### Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/boots
```



```
trap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></scrip
t>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></sc
ript>
</head>
<body>

<div class="container">
  <div class="jumbotron">
    <h1>My First Bootstrap Page</h1>
  </div>
  <div class="row">
    <div class="col-sm-4">
      ...
    </div>
    <div class="col-sm-4">
      ...
    </div>
    <div class="col-sm-4">
      ...
    </div>
  </div>
</body>
</html>
```



To learn more about Bootstrap, go to our [Bootstrap Tutorial](#).

---

## 45 HTML Computer Code Elements

---

HTML contains several elements for defining user input and computer code.

### HTML Computer Code Elements

Tag	Description
<code>	Defines programming code
<kbd>	Defines keyboard input
<samp>	Defines computer output
<var>	Defines a variable
<pre>	Defines preformatted text

#### Example:

```
<code>
x = 5;
```

```
y = 6;  
z = x + y;  
</code>
```



## 45.1 HTML <kbd> For Keyboard Input

The HTML <kbd> element is used to define keyboard input. The content inside is displayed in the browser's default monospace font.

### Example:

Define some text as keyboard input in a document:

```
<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>
```

### Result:

Save the document by pressing Ctrl + S

## 45.2 HTML <samp> For Program Output

The HTML <samp> element is used to define sample output from a computer program. The content inside is displayed in the browser's default monospace font.

### Example:

Define some text as sample output from a computer program in a document:

```
<p>Message from my computer:</p>  
<p><samp>File not found.<br>Press F1 to continue</samp></p>
```

### Result:

Message from my computer:

```
File not found.  
Press F1 to continue
```

## 45.3 HTML <code> For Computer Code

The HTML <code> element is used to define a piece of computer code. The content inside is displayed in the browser's default monospace font.

### Example:

Define some text as computer code in a document:

```
<code>  
x = 5;  
y = 6;  
z = x + y;  
</code>
```

### Result:



```
x = 5; y = 6; z = x + y;
```

Notice that the `<code>` element does not preserve extra whitespace and line-breaks.

To fix this, you can put the `<code>` element inside a `<pre>` element:

**Example:**

```
<pre>
<code>
x = 5;
y = 6;
z = x + y;
</code>
</pre>
```

**Result:**

```
x = 5;
y = 6;
z = x + y;
```



## 45.4 HTML `<var>` For Variables

The HTML `<var>` element is used to define a variable in programming or in a mathematical expression. The content inside is typically displayed in italic.

**Example:**

Define some text as variables in a document:

```
<p>The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>, where <var>b</var> is
the base, and <var>h</var> is the vertical height.</p>
```

**Result:**

The area of a triangle is:  $1/2 \times b \times h$ , where *b* is the base, and *h* is the vertical height.

## 45.5 Chapter Summary

- The `<kbd>` element defines keyboard input
- The `<samp>` element defines sample output from a computer program
- The `<code>` element defines a piece of computer code
- The `<var>` element defines a variable in programming or in a mathematical expression
- The `<pre>` element defines preformatted text

## 46 HTML Semantic Elements

Semantic elements = elements with a meaning.



### 46.1 What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

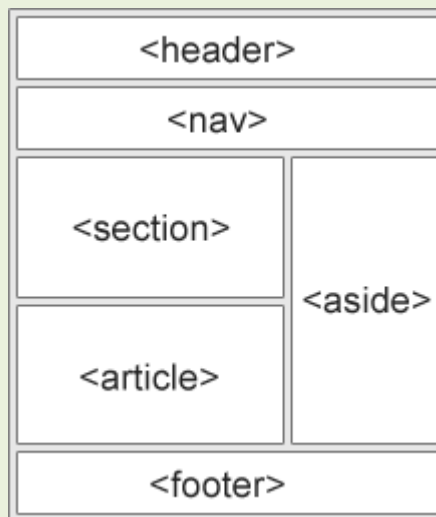
- Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.
- Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

### 46.2 Semantic Elements in HTML

Many web sites contain HTML code like: `<div id="nav">` `<div class="header">` `<div id="footer">` to indicate navigation, header, and footer.

In HTML there are some semantic elements that can be used to define different parts of a web page:

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



### 46.3 HTML `<section>` Element

The `<section>` element defines a section in a document.

According to HTML documentation: "A section is a thematic grouping of content, typically with a heading."

Examples of where a `<section>` element can be used:

- Chapters
- Introduction
- News items
- Contact information

A web page could normally be split into sections for introduction, content, and contact information.

**Example:**

Two sections in a document:

```
<section>
<h1>WWF</h1>
<p>The World Wide Fund for Nature (WWF) is an international organization working on
issues regarding the conservation, research and restoration of the environment,
formerly named the World Wildlife Fund. WWF was founded in 1961.</p>
</section>
```



```
<section>
<h1>WWF's Panda symbol</h1>
<p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated
from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo
in the same year of the establishment of WWF.</p>
</section>
```

## 46.4 HTML <article> Element

The <article> element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.

Examples of where the <article> element can be used:

- Form posts
- Blog posts
- User comments
- Product cards
- Newspaper articles

### Example:1

Three articles with independent, self-contained content:

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the
world's most popular web browser today!</p>
</article>
```

```
<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has
been the second most popular web browser since January, 2018.</p>
</article>
```

```
<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft
Edge replaced Internet Explorer.</p>
</article>
```

### Example 2

Use CSS to style the <article> element:

```
<html>
<head>
<style>
.all-browsers {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

.all-browsers > h1, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>

<article class="all-browsers">
  <h1>Most Popular Browsers</h1>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is
the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has
been the second most popular web browser since January, 2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
  </article>
</article>

</body>
</html>
```



## 46.5 Nesting <article> in <section> or Vice Versa?

The <article> element specifies independent, self-contained content.

The <section> element defines section in a document.

Can we use the definitions to decide how to nest those elements? No, we cannot!

So, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <section> elements.



## 46.6 HTML <header> Element

The <header> element represents a container for introductory content or a set of navigational links.

A <header> element typically contains:

- one or more heading elements (<h1> - <h6>)
- logo or icon
- authorship information

**Note:** You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.

### Example:

A header for an <article>:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

## 46.7 HTML <footer> Element

The <footer> element defines a footer for a document or section.

A <footer> element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents

You can have several <footer> elements in one document.

### Example:

A footer section in a document:

```
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
```



## 46.8 HTML <nav> Element

The <nav> element defines a set of navigation links.

Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

### Example:

A set of navigation links:

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/jquery/">jQuery</a>
</nav>
```

## 46.9 HTML <aside> Element

The <aside> element defines some content aside from the content it is placed in (like a sidebar).

The <aside> content should be indirectly related to the surrounding content.

### Example: 1

Display some content aside from the content it is placed in:

```
<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>
```

```
<aside>
<h4>Epcot Center</h4>
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>
```

### Example: 2

Use CSS to style the <aside> element:

```
<html>
<head>
<style>
```

```
aside {  
  width: 30%;  
  padding-left: 15px;  
  margin-left: 15px;  
  float: right;  
  font-style: italic;  
  background-color: lightgray;  
}  
</style>  
</head>  
<body>
```



```
<p>My family and I visited The Epcot center this summer. The weather was nice, and  
Epcot was amazing! I had a great summer together with my family!</p>
```

```
<aside>  
<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting  
attractions, international pavilions, award-winning fireworks and seasonal special  
events.</p>  
</aside>
```

```
<p>My family and I visited The Epcot center this summer. The weather was nice, and  
Epcot was amazing! I had a great summer together with my family!</p>  
<p>My family and I visited The Epcot center this summer. The weather was nice, and  
Epcot was amazing! I had a great summer together with my family!</p>  
  
</body>  
</html>
```

## 46.10 HTML <figure> and <figcaption> Elements

The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

The <figcaption> tag defines a caption for a <figure> element. The <figcaption> element can be placed as the first or as the last child of a <figure> element.

The <img> element defines the actual image/illustration.

### Example:

```
<figure>  
    
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>  
</figure>
```

## 46.11 Why Semantic Elements?

According to the C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."

## 46.12 Semantic Elements in HTML

Below is a list of some of the semantic elements in HTML.



Tag	Description
<code>&lt;article&gt;</code>	Defines independent, self-contained content
<code>&lt;aside&gt;</code>	Defines content aside from the page content
<code>&lt;details&gt;</code>	Defines additional details that the user can view or hide
<code>&lt;figcaption&gt;</code>	Defines a caption for a <code>&lt;figure&gt;</code> element
<code>&lt;figure&gt;</code>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<code>&lt;footer&gt;</code>	Defines a footer for a document or section
<code>&lt;header&gt;</code>	Specifies a header for a document or section
<code>&lt;main&gt;</code>	Specifies the main content of a document
<code>&lt;mark&gt;</code>	Defines marked/highlighted text
<code>&lt;nav&gt;</code>	Defines navigation links
<code>&lt;section&gt;</code>	Defines a section in a document
<code>&lt;summary&gt;</code>	Defines a visible heading for a <code>&lt;details&gt;</code> element
<code>&lt;time&gt;</code>	Defines a date/time



## 47 HTML Style Guide

A consistent, clean, and tidy HTML code makes it easier for others to read and understand your code.

Here are some guidelines and tips for creating good HTML code.



### 47.1 Always Declare Document Type

Always declare the document type as the first line in your document.

The correct document type for HTML is:

```
<!DOCTYPE html>
```

### 47.2 Use Lowercase Element Names

HTML allows mixing uppercase and lowercase letters in element names.

However, we recommend using lowercase element names, because:

- Mixing uppercase and lowercase names looks bad
- Developers normally use lowercase names
- Lowercase looks cleaner
- Lowercase is easier to write

#### Good:

```
<body>
<p>This is a paragraph.</p>
</body>
```

#### Bad:

```
<BODY>
<P>This is a paragraph.</P>
</BODY>
```

### 47.3 Close All HTML Elements

In HTML, you do not have to close all elements (for example the `<p>` element).

However, we strongly recommend closing all HTML elements, like this:

#### Good:

```
<section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>
```

#### Bad:

```
<section>
  <p>This is a paragraph.
  <p>This is a paragraph.
</section>
```



## 47.4 Use Lowercase Attribute Names

HTML allows mixing uppercase and lowercase letters in attribute names.

However, we recommend using lowercase attribute names, because:

- Mixing uppercase and lowercase names looks bad
- Developers normally use lowercase names
- Lowercase look cleaner
- Lowercase are easier to write

### Good:

```
<a href="https://www.bintr.com/html/">Visit our HTML tutorial</a>
```

### Bad:

```
<a HREF="https://www.bintr.com/html/">Visit our HTML tutorial</a>
```

## 47.5 Always Quote Attribute Values

HTML allows attribute values without quotes.

However, we recommend quoting attribute values, because:

- Developers normally quote attribute values
- Quoted values are easier to read
- You MUST use quotes if the value contains spaces

### Good:

```
<table class="striped">
```

### Bad:

```
<table class=striped>
```

### Very bad:

This will not work, because the value contains spaces:

```
<table class=table striped>
```

## 47.6 Always Specify alt, width, and height for Images

Always specify the `alt` attribute for images. This attribute is important if the image for some reason cannot be displayed.

Also, always define the `width` and `height` of images. This reduces flickering, because the browser can reserve space for the image before loading.



### Good:

```

```

### Bad:

```

```

## 47.7 Spaces and Equal Signs

HTML allows spaces around equal signs. But space-less is easier to read and groups entities better together.

### Good:

```
<link rel="stylesheet" href="styles.css">
```

### Bad:

```
<link rel = "stylesheet" href = "styles.css">
```

## 47.8 Avoid Long Code Lines

When using an HTML editor, it is NOT convenient to scroll right and left to read the HTML code.

Try to avoid too long code lines.

## 47.9 Blank Lines and Indentation

Do not add blank lines, spaces, or indentations without a reason.

For readability, add blank lines to separate large or logical code blocks.

For readability, add two spaces of indentation. Do not use the tab key.

### Good:

```
<body>
```

```
<h1>Famous Cities</h1>
```

```
<h2>Tokyo</h2>
```

```
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>
```

```
<h2>London</h2>
```

```
<p>London is the capital city of England. It is the most populous city in the United
```

```
Kingdom.</p>
```

```
<h2>Paris</h2>
```

```
<p>Paris is the capital of France. The Paris area is one of the largest population centers in Europe.</p>
```

```
</body>
```



### Bad:

```
<body>
```

```
<h1>Famous Cities</h1>
```

```
<h2>Tokyo</h2><p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.</p>
```

```
<h2>London</h2><p>London is the capital city of England. It is the most populous city in the United Kingdom.</p>
```

```
<h2>Paris</h2><p>Paris is the capital of France. The Paris area is one of the largest population centers in Europe.</p>
```

```
</body>
```

### Good Table Example:

```
<table>
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Description</th>
```

```
</tr>
```

```
<tr>
```

```
<td>A</td>
```

```
<td>Description of A</td>
```

```
</tr>
```

```
<tr>
```

```
<td>B</td>
```

```
<td>Description of B</td>
```

```
</tr>
```

```
</table>
```

### Good List Example:

```
<ul>
```

```
<li>London</li>
```

```
<li>Paris</li>
```

```
<li>Tokyo</li>
```

```
</ul>
```

## 47.10 Never Skip the <title> Element

The <title> element is required in HTML.

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The <title> element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results

So, try to make the title as accurate and meaningful as possible:

```
<title>HTML Style Guide and Coding Conventions</title>
```



## 47.11 Omitting <html> and <body>?

An HTML page will validate without the <html> and <body> tags:

**Example:**

```
<!DOCTYPE html>
<head>
  <title>Page Title</title>
</head>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

However, we strongly recommend to always add the <html> and <body> tags!

Omitting <body> can produce errors in older browsers.

Omitting <html> and <body> can also crash DOM and XML software.

## 47.12 Omitting <head>?

The HTML <head> tag can also be omitted.

Browsers will add all elements before <body>, to a default <head> element.

**Example:**

```
<!DOCTYPE html>
<html>
<title>Page Title</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

However, we recommend using the <head> tag.

## 47.13 Close Empty HTML Elements?

In HTML, it is optional to close empty elements.

### Allowed:

```
<meta charset="utf-8">
```

### Also Allowed:

```
<meta charset="utf-8" />
```

If you expect XML/XHTML software to access your page, keep the closing slash (/), because it is required in XML and XHTML.



## 47.14 Add the lang Attribute

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

### Example:

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <title>Page Title</title>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

## 47.15 Meta Data

To ensure proper interpretation and correct search engine indexing, both the language and the character encoding `<meta charset="charset">` should be defined as early as possible in an HTML document:

```
<!DOCTYPE html>
<html lang="en-us">
<head>
  <meta charset="UTF-8">
  <title>Page Title</title>
</head>
```

## 47.16 Setting The Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

**Tip:** If you are browsing this page with a phone or a tablet, you can click on the two links below to see the difference.



## 47.17 HTML Comments

Short comments should be written on one line, like this:

```
<!-- This is a comment -->
```

Comments that spans more than one line, should be written like this:

```
<!--  
  This is a long comment example. This is a long comment example.  
  This is a long comment example. This is a long comment example.  
-->
```

Long comments are easier to observe if they are indented with two spaces.

## 47.18 Using Style Sheets

Use simple syntax for linking to style sheets (the `type` attribute is not necessary):

```
<link rel="stylesheet" href="styles.css">
```

Short CSS rules can be written compressed, like this:

```
p.intro {font-family:Verdana;font-size:16em;}
```

Long CSS rules should be written over multiple lines:

```
body {  
  background-color: lightgrey;  
  font-family: "Arial Black", Helvetica, sans-serif;  
  font-size: 16em;  
  color: black;  
}
```

- Place the opening bracket on the same line as the selector
- Use one space before the opening bracket
- Use two spaces of indentation
- Use semicolon after each property-value pair, including the last
- Only use quotes around values if the value contains spaces
- Place the closing bracket on a new line, without leading spaces



## 47.19 Loading JavaScript in HTML

Use simple syntax for loading external scripts (the `type` attribute is not necessary):

```
<script src="myscript.js">
```

## 47.20 Accessing HTML Elements with JavaScript

Using "untidy" HTML code can result in JavaScript errors.

These two JavaScript statements will produce different results:

**Example:**

```
getElementById("Demo").innerHTML = "Hello";
```

```
getElementById("demo").innerHTML = "Hello";
```

Visit the JavaScript Style Guide.

## 47.21 Use Lower Case File Names

Some web servers (Apache, Unix) are case sensitive about file names: "binod.jpg" cannot be accessed as "binod.JPG".

Other web servers (Microsoft, IIS) are not case sensitive: "london.jpg" can be accessed as "London.JPG".

If you use a mix of uppercase and lowercase, you have to be aware of this.

If you move from a case-insensitive to a case-sensitive server, even small errors will break your web!

To avoid these problems, always use lowercase file names!

## 47.22 File Extensions

HTML files should have a **.html** extension (**.htm** is allowed).

CSS files should have a **.css** extension.

JavaScript files should have a **.js** extension.

## 47.23 Differences Between .htm and .html?

There is no difference between the **.htm** and **.html** file extensions!

Both will be treated as HTML by any web browser and web server.



## 47.24 Default Filenames

When a URL does not specify a filename at the end (like "https://www.bintr.com/"), the server just adds a default filename, such as "index.html", "index.htm", "default.html", or "default.htm".

If your server is configured only with "index.html" as the default filename, your file must be named "index.html", and not "default.html".

However, servers can be configured with more than one default filename; usually you can set up as many default filenames as you want.



## 48 HTML Entities

Reserved characters in HTML must be replaced with character entities.



### 48.1 HTML Entities

Some characters are reserved in HTML.

If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags.

Character entities are used to display reserved characters in HTML.

A character entity looks like this:

`&entity_name;`

OR

`&#entity_number;`

To display a less than sign (<) we must write: **&lt;** or **&#60;**

**Advantage of using an entity name:** An entity name is easy to remember.

**Disadvantage of using an entity name:** Browsers may not support all entity names, but the support for entity numbers is good.

### 48.2 Non-breaking Space

A commonly used entity in HTML is the non-breaking space: **&nbsp;**

A non-breaking space is a space that will not break into a new line.

Two words separated by a non-breaking space will stick together (not break into a new line). This is handy when breaking the words might be disruptive.

Examples:

- § 10
- 10 km/h
- 10 PM

Another common use of the non-breaking space is to prevent browsers from truncating spaces in HTML pages.

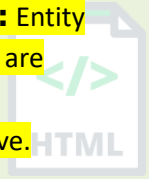
If you write 10 spaces in your text, the browser will remove 9 of them. To add real spaces to your text, you can use the **&nbsp;** character entity.

**Tip:** The non-breaking hyphen (**&#8209;**) is used to define a hyphen character (-) that does not break into a new line.

## 48.3 Some Useful HTML Character Entities

Result	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	double quotation mark	&quot;	&#34;
'	single quotation mark (apostrophe)	&apos;	&#39;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;

**Note:** Entity names are case sensitive.



## 48.4 Combining Diacritical Marks

A diacritical mark is a "glyph" added to a letter.

Some diacritical marks, like grave ( ` ) and acute ( ´ ) are called accents.

Diacritical marks can appear both above and below a letter, inside a letter, and between two letters.

Diacritical marks can be used in combination with alphanumeric characters to produce a character that is not present in the character set (encoding) used in the page.

Here are some examples:

Mark	Character	Construct	Result
`	a	a&#768;	à
'	a	a&#769;	á
^	a	a&#770;	â
~	a	a&#771;	ã
`	O	O&#768;	Ò
'	O	O&#769;	Ó
^	O	O&#770;	Ô
~	O	O&#771;	Õ

## 49 HTML Symbols

Symbols that are not present on your keyboard can also be added by using entities.



### 49.1 HTML Symbol Entities

HTML entities were described in the previous chapter.

Many mathematical, technical, and currency symbols, are not present on a normal keyboard.

To add such symbols to an HTML page, you can use the entity name or the entity number (a decimal or a hexadecimal reference) for the symbol.

#### Example:

Display the euro sign, €, with an entity name, a decimal, and a hexadecimal value:

```
<p>I will display &euro;</p>
<p>I will display &#8364;</p>
<p>I will display &#x20AC;</p>
```

#### Will display as:

I will display €  
 I will display €  
 I will display €

### 49.2 Some Mathematical Symbols Supported by HTML

Character	Number	Entity	Description
∀	&#8704;	&forall;	FOR ALL
∂	&#8706;	&part;	PARTIAL DIFFERENTIAL
∃	&#8707;	&exist;	THERE EXISTS
∅	&#8709;	&empty;	EMPTY SETS
∇	&#8711;	&nabla;	NABLA
∈	&#8712;	&isin;	ELEMENT OF
∉	&#8713;	&notin;	NOT AN ELEMENT OF
∋	&#8715;	&ni;	CONTAINS AS MEMBER
∏	&#8719;	&prod;	N-ARY PRODUCT

Σ	&#8721;	&sum;	N-ARY SUMMATION
---	---------	-------	-----------------



### 49.3 Some Greek Letters Supported by HTML

Char	Number	Entity	Description
A	&#913;	&Alpha;	GREEK CAPITAL LETTER ALPHA
B	&#914;	&Beta;	GREEK CAPITAL LETTER BETA
Γ	&#915;	&Gamma;	GREEK CAPITAL LETTER GAMMA
Δ	&#916;	&Delta;	GREEK CAPITAL LETTER DELTA
E	&#917;	&Epsilon;	GREEK CAPITAL LETTER EPSILON
Z	&#918;	&Zeta;	GREEK CAPITAL LETTER ZETA

### 49.4 Some Other Entities Supported by HTML

Char	Number	Entity	Description
©	&#169;	&copy;	COPYRIGHT SIGN
®	&#174;	&reg;	REGISTERED SIGN
€	&#8364;	&euro;	EURO SIGN
™	&#8482;	&trade;	TRADEMARK
←	&#8592;	&larr;	LEFTWARDS ARROW
↑	&#8593;	&uarr;	UPWARDS ARROW
→	&#8594;	&rarr;	RIGHTWARDS ARROW
↓	&#8595;	&darr;	DOWNWARDS ARROW
♠	&#9824;	&spades;	BLACK SPADE SUIT
♣	&#9827;	&clubs;	BLACK CLUB SUIT
♥	&#9829;	&hearts;	BLACK HEART SUIT

◆	&#9830;	&diams;	BLACK DIAMOND SUIT
---	---------	---------	--------------------



## 50 Using Emojis in HTML

Emojis are characters from the UTF-8 character set: 😊 😍 ❤️

### 50.1 What are Emojis?

Emojis look like images, or icons, but they are not.

They are letters (characters) from the UTF-8 (Unicode) character set.

UTF-8 covers almost all of the characters and symbols in the world.

### 50.2 The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page.

This is specified in the `<meta>` tag:

```
<meta charset="UTF-8">
```

If not specified, UTF-8 is the default character set in HTML.

### 50.3 UTF-8 Characters

Many UTF-8 characters cannot be typed on a keyboard, but they can always be displayed using numbers (called entity numbers):

- A is 65
- B is 66
- C is 67

**Example:**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>

<p>I will display A B C</p>
<p>I will display &#65; &#66; &#67;</p>

</body>
</html>
```

**Example Explained**

The `<meta charset="UTF-8">` element defines the character set.

The characters A, B, and C, are displayed by the numbers 65, 66, and 67.

To let the browser understand that you are displaying a character, you must start the entity number with `&#` and end it with `;` (semicolon).



## Emoji Characters

Emojis are also characters from the UTF-8 alphabet:

😊 is 128516

😍 is 128525

♥ is 128151

### Example:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>

<h1>My First Emoji</h1>

<p>&#128512;</p>

</body>
</html>
```

Since Emojis are characters, they can be copied, displayed, and sized just like any other character in HTML.

### Example:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
</head>
<body>

<h1>Sized Emojis</h1>

<p style="font-size:48px">
&#128512; &#128516; &#128525; &#128151;
</p>

</body>
</html>
```

## 50.4 Some Emoji Symbols in UTF-8

Emoji	Value
	&#128507;
	&#128508;
	&#128509;
	&#128510;
	&#128511;
	&#128512;
	&#128513;
	&#128514;
	&#128515;
	&#128516;
	&#128517;





## 51 HTML Encoding (Character Sets)

To display an HTML page correctly, a web browser must know which character set to use.



### 51.1 From ASCII to UTF-8

ASCII was the first character encoding standard. ASCII defined 128 different characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like ! \$ + - ( ) @ < > .

ISO-8859-1 was the default character set for HTML 4. This character set supported 256 different character codes. HTML 4 also supported UTF-8.

ANSI (Windows-1252) was the original Windows character set. ANSI is identical to ISO-8859-1, except that ANSI has 32 extra characters.

The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world!

### 51.2 The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page.

This is specified in the `<meta>` tag:

```
<meta charset="UTF-8">
```

### 51.3 Differences Between Character Sets

The following table displays the differences between the character sets described above:

Numb	ASCII	ANSI	8859	UTF-8	Description
32					space
33	!	!	!	!	exclamation mark
34	"	"	"	"	quotation mark
35	#	#	#	#	number sign
36	\$	\$	\$	\$	dollar sign
37	%	%	%	%	percent sign
38	&	&	&	&	ampersand
39	'	'	'	'	apostrophe
40	(	(	(	(	left parenthesis
41	)	)	)	)	right parenthesis
42	*	*	*	*	asterisk
43	+	+	+	+	plus sign
44	,	,	,	,	comma

45	-	-	-	-	hyphen-minus
46	.	.	.	.	full stop
47	/	/	/	/	solidus
48	0	0	0	0	digit zero
49	1	1	1	1	digit one
50	2	2	2	2	digit two
51	3	3	3	3	digit three
52	4	4	4	4	digit four
53	5	5	5	5	digit five
54	6	6	6	6	digit six
55	7	7	7	7	digit seven
56	8	8	8	8	digit eight
57	9	9	9	9	digit nine
58	:	:	:	:	colon
59	;	;	;	;	semicolon
60	<	<	<	<	less-than sign
61	=	=	=	=	equals sign
62	>	>	>	>	greater-than sign
63	?	?	?	?	question mark
64	@	@	@	@	commercial at
65	A	A	A	A	Latin capital letter A
66	B	B	B	B	Latin capital letter B
67	C	C	C	C	Latin capital letter C
68	D	D	D	D	Latin capital letter D
69	E	E	E	E	Latin capital letter E
70	F	F	F	F	Latin capital letter F
71	G	G	G	G	Latin capital letter G
72	H	H	H	H	Latin capital letter H
73	I	I	I	I	Latin capital letter I
74	J	J	J	J	Latin capital letter J
75	K	K	K	K	Latin capital letter K
76	L	L	L	L	Latin capital letter L
77	M	M	M	M	Latin capital letter M
78	N	N	N	N	Latin capital letter N
79	O	O	O	O	Latin capital letter O
80	P	P	P	P	Latin capital letter P
81	Q	Q	Q	Q	Latin capital letter Q



82	R	R	R	R	Latin capital letter R
83	S	S	S	S	Latin capital letter S
84	T	T	T	T	Latin capital letter T
85	U	U	U	U	Latin capital letter U
86	V	V	V	V	Latin capital letter V
87	W	W	W	W	Latin capital letter W
88	X	X	X	X	Latin capital letter X
89	Y	Y	Y	Y	Latin capital letter Y
90	Z	Z	Z	Z	Latin capital letter Z
91	[	[	[	[	left square bracket
92	\	\	\	\	reverse solidus
93	]	]	]	]	right square bracket
94	^	^	^	^	circumflex accent
95	_	_	_	_	low line
96	`	`	`	`	grave accent
97	a	a	a	a	Latin small letter a
98	b	b	b	b	Latin small letter b
99	c	c	c	c	Latin small letter c
100	d	d	d	d	Latin small letter d
101	e	e	e	e	Latin small letter e
102	f	f	f	f	Latin small letter f
103	g	g	g	g	Latin small letter g
104	h	h	h	h	Latin small letter h
105	i	i	i	i	Latin small letter i
106	j	j	j	j	Latin small letter j
107	k	k	k	k	Latin small letter k
108	l	l	l	l	Latin small letter l
109	m	m	m	m	Latin small letter m
110	n	n	n	n	Latin small letter n
111	o	o	o	o	Latin small letter o
112	p	p	p	p	Latin small letter p
113	q	q	q	q	Latin small letter q
114	r	r	r	r	Latin small letter r
115	s	s	s	s	Latin small letter s
116	t	t	t	t	Latin small letter t
117	u	u	u	u	Latin small letter u
118	v	v	v	v	Latin small letter v



119	w	w	w	W	Latin small letter w
120	x	x	x	x	Latin small letter x
121	y	y	y	y	Latin small letter y
122	z	z	z	z	Latin small letter z
123	{	{	{	{	left curly bracket
124					vertical line
125	}	}	}	}	right curly bracket
126	~	~	~	~	tilde
127	DEL				
128		€			euro sign
129		•	•	•	NOT USED
130		,			single low-9 quotation mark
131		f			Latin small letter f with hook
132		„			double low-9 quotation mark
133		...			horizontal ellipsis
134		†			dagger
135		‡			double dagger
136		^			modifier letter circumflex accent
137		‰			per mille sign
138		Š			Latin capital letter S with caron
139		‹			single left-pointing angle quotation mark
140		Œ			Latin capital ligature OE
141		•	•	•	NOT USED
142		Ž			Latin capital letter Z with caron
143		•	•	•	NOT USED
144		•	•	•	NOT USED
145		‘			left single quotation mark
146		’			right single quotation mark
147		“			left double quotation mark
148		”			right double quotation mark
149		•			bullet
150		–			en dash
151		—			em dash
152		~			small tilde
153		™			trade mark sign
154		š			Latin small letter s with caron
155		›			single right-pointing angle quotation mark



156		œ			Latin small ligature oe
157		•	•	•	NOT USED
158		ž			Latin small letter z with caron
159		ÿ			Latin capital letter Y with diaeresis
160					no-break space
161		¡	¡	¡	inverted exclamation mark
162		¢	¢	¢	cent sign
163		£	£	£	pound sign
164		¤	¤	¤	currency sign
165		¥	¥	¥	yen sign
166		¦	¦	¦	broken bar
167		§	§	§	section sign
168		¨	¨	¨	diaeresis
169		©	©	©	copyright sign
170		ª	ª	ª	feminine ordinal indicator
171		«	«	«	left-pointing double angle quotation mark
172		¬	¬	¬	not sign
173					soft hyphen
174		®	®	®	registered sign
175		¯	¯	¯	macron
176		°	°	°	degree sign
177		±	±	±	plus-minus sign
178		²	²	²	superscript two
179		³	³	³	superscript three
180		´	´	´	acute accent
181		µ	µ	µ	micro sign
182		¶	¶	¶	pilcrow sign
183		·	·	·	middle dot
184		¸	¸	¸	cedilla
185		¹	¹	¹	superscript one
186		º	º	º	masculine ordinal indicator
187		»	»	»	right-pointing double angle quotation mark
188		¼	¼	¼	vulgar fraction one quarter
189		½	½	½	vulgar fraction one half
190		¾	¾	¾	vulgar fraction three quarters
191		¿	¿	¿	inverted question mark
192		À	À	À	Latin capital letter A with grave



193	Á	Á	Á	Latin capital letter A with acute
194	Â	Â	Â	Latin capital letter A with circumflex
195	Ã	Ã	Ã	Latin capital letter A with tilde
196	Ä	Ä	Ä	Latin capital letter A with diaeresis
197	Å	Å	Å	Latin capital letter A with ring above
198	Æ	Æ	Æ	Latin capital letter AE
199	Ç	Ç	Ç	Latin capital letter C with cedilla
200	È	È	È	Latin capital letter E with grave
201	É	É	É	Latin capital letter E with acute
202	Ê	Ê	Ê	Latin capital letter E with circumflex
203	Ë	Ë	Ë	Latin capital letter E with diaeresis
204	Ì	Ì	Ì	Latin capital letter I with grave
205	Í	Í	Í	Latin capital letter I with acute
206	Î	Î	Î	Latin capital letter I with circumflex
207	Ï	Ï	Ï	Latin capital letter I with diaeresis
208	Ð	Ð	Ð	Latin capital letter Eth
209	Ñ	Ñ	Ñ	Latin capital letter N with tilde
210	Ò	Ò	Ò	Latin capital letter O with grave
211	Ó	Ó	Ó	Latin capital letter O with acute
212	Ô	Ô	Ô	Latin capital letter O with circumflex
213	Õ	Õ	Õ	Latin capital letter O with tilde
214	Ö	Ö	Ö	Latin capital letter O with diaeresis
215	×	×	×	multiplication sign
216	Ø	Ø	Ø	Latin capital letter O with stroke
217	Ù	Ù	Ù	Latin capital letter U with grave
218	Ú	Ú	Ú	Latin capital letter U with acute
219	Û	Û	Û	Latin capital letter U with circumflex
220	Ü	Ü	Ü	Latin capital letter U with diaeresis
221	Ý	Ý	Ý	Latin capital letter Y with acute
222	Þ	Þ	Þ	Latin capital letter Thorn
223	ß	ß	ß	Latin small letter sharp s
224	à	à	à	Latin small letter a with grave
225	á	á	á	Latin small letter a with acute
226	â	â	â	Latin small letter a with circumflex
227	ã	ã	ã	Latin small letter a with tilde
228	ä	ä	ä	Latin small letter a with diaeresis
229	å	å	å	Latin small letter a with ring above



230	æ	æ	æ	Latin small letter ae
231	ç	ç	ç	Latin small letter c with cedilla
232	è	è	è	Latin small letter e with grave
233	é	é	é	Latin small letter e with acute
234	ê	ê	ê	Latin small letter e with circumflex
235	ë	ë	ë	Latin small letter e with diaeresis
236	ì	ì	ì	Latin small letter i with grave
237	í	í	í	Latin small letter i with acute
238	î	î	î	Latin small letter i with circumflex
239	ï	ï	ï	Latin small letter i with diaeresis
240	ð	ð	ð	Latin small letter eth
241	ñ	ñ	ñ	Latin small letter n with tilde
242	ò	ò	ò	Latin small letter o with grave
243	ó	ó	ó	Latin small letter o with acute
244	ô	ô	ô	Latin small letter o with circumflex
245	õ	õ	õ	Latin small letter o with tilde
246	ö	ö	ö	Latin small letter o with diaeresis
247	÷	÷	÷	division sign
248	ø	ø	ø	Latin small letter o with stroke
249	ù	ù	ù	Latin small letter u with grave
250	ú	ú	ú	Latin small letter u with acute
251	û	û	û	Latin small letter with circumflex
252	ü	ü	ü	Latin small letter u with diaeresis
253	ý	ý	ý	Latin small letter y with acute
254	þ	þ	þ	Latin small letter thorn
255	ÿ	ÿ	ÿ	Latin small letter y with diaeresis



## 51.4 The ASCII Character Set

ASCII uses the values from 0 to 31 (and 127) for control characters.

ASCII uses the values from 32 to 126 for letters, digits, and symbols.

ASCII does not use the values from 128 to 255.

## 51.5 The ANSI Character Set (Windows-1252)

ANSI is identical to ASCII for the values from 0 to 127.

ANSI has a proprietary set of characters for the values from 128 to 159.

ANSI is identical to UTF-8 for the values from 160 to 255.

## 51.6 The ISO-8859-1 Character Set

ISO-8859-1 is identical to ASCII for the values from 0 to 127.

ISO-8859-1 does not use the values from 128 to 159.

ISO-8859-1 is identical to UTF-8 for the values from 160 to 255.



## 51.7 The UTF-8 Character Set

UTF-8 is identical to ASCII for the values from 0 to 127.

UTF-8 does not use the values from 128 to 159.

UTF-8 is identical to both ANSI and 8859-1 for the values from 160 to 255.

UTF-8 continues from the value 256 with more than 10 000 different characters.



## 52 HTML Uniform Resource Locators

A URL is another word for a web address.

A URL can be composed of words (e.g. bintr.com), or an Internet Protocol (IP) address (e.g. 192.68.20.50).

Most people enter the name when surfing, because names are easier to remember than numbers.



### 52.1 URL - Uniform Resource Locator

Web browsers request pages from web servers by using a URL.

A Uniform Resource Locator (URL) is used to address a document (or other data) on the web.

A web address like `https://www.bintr.com/html/binod.jpg` follows these syntax rules:

`scheme://prefix.domain:port/path/filename`

Explanation:

- **scheme** - defines the **type** of Internet service (most common is **http** or **https**)
- **prefix** - defines a domain **prefix** (default for http is **www**)
- **domain** - defines the Internet **domain name** (like bintr.com)
- **port** - defines the **port number** at the host (default for http is **80**)
- **path** - defines a **path** at the server (If omitted: the root directory of the site)
- **filename** - defines the name of a document or resource

### 52.2 Common URL Schemes

The table below lists some common schemes:

Scheme	Short for	Used for
http	HyperText Transfer Protocol	Common web pages. Not encrypted
https	Secure HyperText Transfer Protocol	Secure web pages. Encrypted
ftp	File Transfer Protocol	Downloading or uploading files
file		A file on your computer

## 53 HTML ASCII Reference

ASCII was the first character set (encoding standard) used between computers on the Internet.

Both ISO-8859-1 (default in HTML 4.01) and UTF-8 (default in HTML5), are built on ASCII.



### 53.1 The ASCII Character Set

ASCII stands for the "American Standard Code for Information Interchange".

It was designed in the early 60's, as a standard character set for computers and electronic devices.

ASCII is a 7-bit character set containing 128 characters.

It contains the numbers from 0-9, the upper and lower case English letters from A to Z, and some special characters.

The character sets used in modern computers, in HTML, and on the Internet, are all based on ASCII.

The following tables list the 128 ASCII characters and their equivalent number.

### 53.2 ASCII Printable Characters

Character	Number	Description
		Control characters (see below)
	0 – 31	space
!	33	exclamation mark
"	34	quotation mark
#	35	number sign
\$	36	dollar sign
%	37	percent sign
&	38	ampersand
'	39	apostrophe
(	40	left parenthesis
)	41	right parenthesis
*	42	asterisk
+	43	plus sign
,	44	comma
-	45	hyphen
.	46	period
/	47	slash
0	48	digit 0

1	49	digit 1
2	50	digit 2
3	51	digit 3
4	52	digit 4
5	53	digit 5
6	54	digit 6
7	55	digit 7
8	56	digit 8
9	57	digit 9
:	58	colon
;	59	semicolon
<	60	less-than
=	61	equals-to
>	62	greater-than
?	63	question mark
@	64	at sign
A	65	uppercase A
B	66	uppercase B
C	67	uppercase C
D	68	uppercase D
E	69	uppercase E
F	70	uppercase F
G	71	uppercase G
H	72	uppercase H
I	73	uppercase I
J	74	uppercase J
K	75	uppercase K
L	76	uppercase L
M	77	uppercase M
N	78	uppercase N
O	79	uppercase O
P	80	uppercase P
Q	81	uppercase Q
R	82	uppercase R
S	83	uppercase S
T	84	uppercase T
U	85	uppercase U
V	86	uppercase V
W	87	uppercase W
X	88	uppercase X



Y	89	uppercase Y
Z	90	uppercase Z
[	91	left square bracket
\	92	backslash
]	93	right square bracket
^	94	caret
_	95	underscore
`	96	grave accent



### 53.3 ASCII Encoding Examples

Your browser will encode input, according to the character-set used in your page.

The default character-set in HTML5 is UTF-8.

Character	From Windows-1252	From UTF-8
€	%80	%E2%82%AC
£	%A3	%C2%A3
©	%A9	%C2%A9
®	%AE	%C2%AE
À	%C0	%C3%80
Á	%C1	%C3%81
Â	%C2	%C3%82
Ã	%C3	%C3%83
Ä	%C4	%C3%84
Å	%C5	%C3%85

## 54 HTML Versus XHTML

XHTML is a stricter, more XML-based version of HTML.



### 54.1 What is XHTML?

- XHTML stands for **EX**tensible **HyperText Markup Language**
- XHTML is a stricter, more XML-based version of HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers

### 54.2 Why XHTML?

XML is a markup language where all documents must be marked up correctly (be "well-formed").

XHTML was developed to make HTML more extensible and flexible to work with other data formats (such as XML). In addition, browsers ignore errors in HTML pages, and try to display the website even if it has some errors in the markup. So XHTML comes with a much stricter error handling.

### 54.3 The Most Important Differences from HTML

- `<!DOCTYPE>` is **mandatory**
- The `xmlns` attribute in `<html>` is **mandatory**
- `<html>`, `<head>`, `<title>`, and `<body>` are **mandatory**
- Elements must always be **properly nested**
- Elements must always be **closed**
- Elements must always be in **lowercase**
- Attribute names must always be in **lowercase**
- Attribute values must always be **quoted**
- Attribute minimization is **forbidden**

### 54.4 XHTML - `<!DOCTYPE ....>` Is Mandatory

An XHTML document must have an XHTML `<!DOCTYPE>` declaration.

The `<html>`, `<head>`, `<title>`, and `<body>` elements must also be present, and the `xmlns` attribute in `<html>` must specify the xml namespace for the document.

#### Example:

Here is an XHTML document with a minimum of required tags:

```
<!DOCTYPE html PUBLIC "-//C//DTD XHTML 1.1//EN"
"http://www..org/TR/xhtml11/DTD/xhtml11.dtd" >
<html xmlns="http://www..org/1999/xhtml" >
<head>
  <title>Title of document</title>
```

```
</head>
<body>
  some content here...
</body>
</html>
```



## 54.5 XHTML Elements Must be Properly Nested

In XHTML, elements must always be properly nested within each other, like this:

### Correct:

```
<b><i>Some text</i></b>
```

### Wrong:

```
<b><i>Some text</b></i>
```

## 54.6 XHTML Elements Must Always be Closed

In XHTML, elements must always be closed, like this:

### Correct:

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

### Wrong:

```
<p>This is a paragraph
<p>This is another paragraph
```

## 54.7 XHTML Empty Elements Must Always be Closed

In XHTML, empty elements must always be closed, like this:

### Correct:

```
A break: <br />
A horizontal rule: <hr />
An image: 
```

### Wrong:

```
A break: <br>
A horizontal rule: <hr>
An image: 
```

XHTML Elements Must be in Lowercase

In XHTML, element names must always be in lowercase, like this:

**Correct:**

```
<body>
<p>This is a paragraph</p>
</body>
```

**Wrong:**

```
<BODY>
<P>This is a paragraph</P>
</BODY>
```

**XHTML Attribute Names Must be in Lowercase**

In XHTML, attribute names must always be in lowercase, like this:

**Correct:**

```
<a href="https://www.bintr.com/html/">Visit our HTML tutorial</a>
```

**Wrong:**

```
<a HREF="https://www.bintr.com/html/">Visit our HTML tutorial</a>
```

**XHTML Attribute Values Must be Quoted**

In XHTML, attribute values must always be quoted, like this:

**Correct:**

```
<a href="https://www.bintr.com/html/">Visit our HTML tutorial</a>
```

**Wrong:**

```
<a href=https://www.bintr.com/html/>Visit our HTML tutorial</a>
```

## 54.8 XHTML Attribute Minimization is Forbidden

In XHTML, attribute minimization is forbidden:

**Correct:**

```
<input type="checkbox" name="vehicle" value="car" checked="checked" />
<input type="text" name="lastname" disabled="disabled" />
```

**Wrong:**

```
<input type="checkbox" name="vehicle" value="car" checked />
<input type="text" name="lastname" disabled />
```

## 54.9 Validate HTML With The C Validator

Put your web address in the box below:



[https://www.w3schools.com/html/html\\_validate.html](https://www.w3schools.com/html/html_validate.html)

Validate the page





## 55 HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.



### Example:

First name:

Last name:

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Forms</h2>
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
<p>If you click the "Submit" button, the form-data will be sent to a page called "/action_page.php".</p>
</body>
</html>
```

### 55.1 The `<form>` Element

The HTML `<form>` element is used to create an HTML form for user input:

```
<form>
.
form elements
.
</form>
```

The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

All the different form elements are covered in this chapter: [HTML Form Elements](#).

## 55.2 The <input> Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the `type` attribute.

Here are some examples:



Type	Description
<code>&lt;input type="text"&gt;</code>	Displays a single-line text input field
<code>&lt;input type="radio"&gt;</code>	Displays a radio button (for selecting one of many choices)
<code>&lt;input type="checkbox"&gt;</code>	Displays a checkbox (for selecting zero or more of many choices)
<code>&lt;input type="submit"&gt;</code>	Displays a submit button (for submitting the form)
<code>&lt;input type="button"&gt;</code>	Displays a clickable button

## 55.3 Text Fields

The `<input type="text">` defines a single-line input field for text input.

### Example:

A form with input fields for text:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

**Note:** The form itself is not visible. Also note that the default width of an input field is 20 characters.

## 55.4 The <label> Element

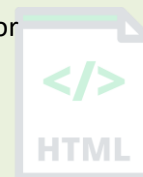
Notice the use of the `<label>` element in the example above.

The `<label>` tag defines a label for many form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The `<label>` element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the `<label>` element, it toggles the radio button/checkbox.

The `for` attribute of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.



## 55.5 Radio Buttons

The `<input type="radio">` defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

### Example:

A form with radio buttons:

```
<p>Choose your favorite Web language:</p>
```

```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

Choose your favorite Web language:

- HTML
- CSS
- JavaScript

## 55.6 Checkboxes

The `<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

### Example:

A form with checkboxes:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
```

```
<label for="vehicle2"> I have a car</label><br>
<input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
<label for="vehicle3"> I have a boat</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

- I have a bike  
 I have a car  
 I have a boat



## 55.7 The Submit Button

The `<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a file on the server with a script for processing input data.

The form-handler is specified in the form's `action` attribute.

### Example:

A form with a submit button:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

## 55.8 The Name Attribute for <input>

Notice that each input field must have a `name` attribute to be submitted.

If the `name` attribute is omitted, the value of the input field will not be sent at all.

### Example:

This example will not submit the value of the "First name" input field:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
```

```
<input type="text" id="fname" value="John"><br><br>  
<input type="submit" value="Submit">  
</form>
```



## 56 HTML Form Attributes

This chapter describes the different attributes for the HTML `<form>` element.



### 56.1 The Action Attribute

The `action` attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

In the example below, the form data is sent to a file called "action\_page.php". This file contains a server-side script that handles the form data:

#### Example:

On submit, send form data to "action\_page.php":

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

### 56.2 The Target Attribute

The `target` attribute specifies where to display the response that is received after submitting the form.

The `target` attribute can have one of the following values:

Value	Description
<code>_blank</code>	The response is displayed in a new window or tab
<code>_self</code>	The response is displayed in the current window
<code>_parent</code>	The response is displayed in the parent frame
<code>_top</code>	The response is displayed in the full body of the window
<code>framename</code>	The response is displayed in a named iframe

The default value is `_self` which means that the response will open in the current window.

#### Example:

Here, the submitted result will open in a new browser tab:

```
<form action="/action_page.php" target="_blank">
```

## 56.3 The Method Attribute

The `method` attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with `method="get"`) or as HTTP post transaction (with `method="post"`).

The default HTTP method when submitting form data is GET.

### Example:

This example uses the GET method when submitting the form data:

```
<form action="/action_page.php" method="get">
```

### Example:

This example uses the POST method when submitting the form data:

```
<form action="/action_page.php" method="post">
```

### Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

### Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

**Tip:** Always use POST if the form data contains sensitive or personal information!

## 56.4 The Autocomplete Attribute

The `autocomplete` attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

### Example:

A form with autocomplete on:

```
<form action="/action_page.php" autocomplete="on">
```

## 56.5 The Novalidate Attribute

The `novalidate` attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

### Example:



A form with a novalidate attribute:

```
<form action="/action_page.php" novalidate>
```



## 56.6 List of All <form> Attributes

Attribute	Description
<b>accept-charset</b>	Specifies the character encodings used for form submission
<b>action</b>	Specifies where to send the form-data when a form is submitted
<b>autocomplete</b>	Specifies whether a form should have autocomplete on or off
<b>enctype</b>	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
<b>method</b>	Specifies the HTTP method to use when sending form-data
<b>name</b>	Specifies the name of the form
<b>novalidate</b>	Specifies that the form should not be validated when submitted
<b>rel</b>	Specifies the relationship between a linked resource and the current document
<b>target</b>	Specifies where to display the response that is received after submitting the form



## 57 HTML Form Elements

This chapter describes all the different HTML form elements.



### 57.1 The HTML <form> Elements

The HTML <form> element can contain one or more of the following form elements:

- <input>
- <label>
- <select>
- <textarea>
- <button>
- <fieldset>
- <legend>
- <datalist>
- <output>
- <option>
- <optgroup>

### 57.2 The <input> Element

One of the most used form element is the <input> element.

The <input> element can be displayed in several ways, depending on the `type` attribute.

**Example:**

```
<label for="fname">First name:</label>
<input type="text" id="fname" name="fname">
```

All the different values of the `type` attribute are covered in the chapter.

### 57.3 The <label> Element

The <label> element defines a label for several form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.

The `for` attribute of the <label> tag should be equal to the `id` attribute of the <input> element to bind them together.

## 57.4 The <select> Element

The <select> element defines a drop-down list:

### Example:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo"> Palser </option>
  <option value="saab">honda </option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The <option> elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the `selected` attribute to the option:

### Example:

```
<option value="fiat" selected>Fiat</option>
```

### Visible Values:

Use the `size` attribute to specify the number of visible values:

### Example:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="3">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

### Allow Multiple Selections:

Use the `multiple` attribute to allow the user to select more than one value:

### Example:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

## 57.5 The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

### Example:



```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

The `rows` attribute specifies the visible number of lines in a text area.

The `cols` attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:



You can also define the size of the text area by using CSS:

**Example:**

```
<textarea name="message" style="width:200px; height:600px;">
The cat was playing in the garden.
</textarea>
```

## 57.6 The <button> Element

The `<button>` element defines a clickable button:

**Example:**

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

This is how the HTML code above will be displayed in a browser:

Click Me!

**Note:** Always specify the `type` attribute for the button element. Different browsers may use different default types for the button element.

## 57.7 The <fieldset> and <legend> Elements

The `<fieldset>` element is used to group related data in a form.

The `<legend>` element defines a caption for the `<fieldset>` element.

**Example:**

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal:</legend>
    <label for="fname">First name:</label><br>
```



```

<input type="text" id="fname" name="fname" value="John"><br>
<label for="lname">Last name:</label><br>
<input type="text" id="lname" name="lname" value="Doe"><br><br>
<input type="submit" value="Submit">
</fieldset>
</form>

```



This is how the HTML code above will be displayed in a browser:

Personal:First name:

Last name:



## 57.8 The <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

The `list` attribute of the <input> element, must refer to the `id` attribute of the <datalist> element.

**Example:**

```

<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>

```

## 57.9 The <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

**Example:**

Perform a calculation and show the result in an <output> element:

```

<form action="/action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">

```

```
=  
<output name="x" for="a b"></output>  
<br><br>  
<input type="submit">  
</form>
```



## 57.10 HTML Form Elements

Tag	Description
<b>&lt;form&gt;</b>	Defines an HTML form for user input
<b>&lt;input&gt;</b>	Defines an input control
<b>&lt;textarea&gt;</b>	Defines a multiline input control (text area)
<b>&lt;label&gt;</b>	Defines a label for an <input> element
<b>&lt;fieldset&gt;</b>	Groups related elements in a form
<b>&lt;legend&gt;</b>	Defines a caption for a <fieldset> element
<b>&lt;select&gt;</b>	Defines a drop-down list
<b>&lt;optgroup&gt;</b>	Defines a group of related options in a drop-down list
<b>&lt;option&gt;</b>	Defines an option in a drop-down list
<b>&lt;button&gt;</b>	Defines a clickable button
<b>&lt;datalist&gt;</b>	Specifies a list of pre-defined options for input controls
<b>&lt;output&gt;</b>	Defines the result of a calculation

## 58 HTML Input Types

This chapter describes the different types for the HTML `<input>` element.



### 58.1 HTML Input Types

Here are the different input types you can use in HTML:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

**Tip:** The default value of the `type` attribute is "text".

### 58.2 Input Type Text

`<input type="text">` defines a **single-line text input field**:

**Example:**

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:



## 58.3 Input Type Password

`<input type="password">` defines a **password field**:

**Example:**

```
<form>
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

This is how the HTML code above will be displayed in a browser:

Username:

Password:

The characters in a password field are masked (shown as asterisks or circles).

## 58.4 Input Type Submit

`<input type="submit">` defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's `action` attribute:

**Example:**

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:



If you omit the submit button's value attribute, the button will get a default text:

### Example:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit">
</form>
```

## 58.5 Input Type Reset

`<input type="reset">` defines a **reset button** that will reset all form values to their default values:

### Example:

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

This is how the HTML code above will be displayed in a browser:

First name:

Last name:

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

## 58.6 Input Type Radio

`<input type="radio">` defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

### Example:

```
<p>Choose your favorite Web language:</p>
```



```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```



This is how the HTML code above will be displayed in a browser:

HTML

CSS

JavaScript

## 58.7 Input Type Checkbox

`<input type="checkbox">` defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

### Example:

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

This is how the HTML code above will be displayed in a browser:

I have a bike

I have a car

I have a boat

## 58.8 Input Type Button

`<input type="button">` defines a **button**:

### Example:

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

## 58.9 Input Type Color

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

### Example:

```
<form>
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor">
</form>
```



## 58.10 Input Type Date

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

### Example:

```
<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
</form>
```

You can also use the `min` and `max` attributes to add restrictions to dates:

### Example:

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>
  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02">
</form>
```

## 58.11 Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

### Example:

```
<form>
  <label for="birthdaytime">Birthday (date and time):</label>
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">
</form>
```

## 58.12 Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

**Example:**

```
<form>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
</form>
```



## 58.13 Input Type File

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

**Example:**

```
<form>
  <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile">
</form>
```

## 58.14 Input Type Hidden

The `<input type="hidden">` defines a hidden input field (not visible to a user).

A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted.

A hidden field often stores what database record that needs to be updated when the form is submitted.

**Note:** While the value is not displayed to the user in the page's content, it is visible (and can be edited) using any browser's developer tools or "View Source" functionality. Do not use hidden inputs as a form of security!

**Example:**

```
<form>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>
```

## 58.15 Input Type Month

The `<input type="month">` allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

**Example:**

```
<form>
  <label for="bdaymonth">Birthday (month and year):</label>
  <input type="month" id="bdaymonth" name="bdaymonth">
</form>
```

## 58.16 Input Type Number

The `<input type="number">` defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

**Example:**

```
<form>
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```



## 58.17 Input Restrictions

Here is a list of some common input restrictions:

Attribute	Description
<b>checked</b>	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
<b>disabled</b>	Specifies that an input field should be disabled
<b>max</b>	Specifies the maximum value for an input field
<b>maxlength</b>	Specifies the maximum number of character for an input field
<b>min</b>	Specifies the minimum value for an input field
<b>pattern</b>	Specifies a regular expression to check the input value against
<b>readonly</b>	Specifies that an input field is read only (cannot be changed)
<b>required</b>	Specifies that an input field is required (must be filled out)
<b>size</b>	Specifies the width (in characters) of an input field
<b>step</b>	Specifies the legal number intervals for an input field
<b>value</b>	Specifies the default value for an input field

You will learn more about input restrictions in the chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

**Example:**

```
<form>
  <label for="quantity">Quantity:</label>
  <input type="number" id="quantity" name="quantity" min="0" max="100" step="10" value="30">
</form>
```

## 58.18 Input Type Range

The `<input type="range">` defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes:



### Example:

```
<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

## 58.19 Input Type Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

### Example:

```
<form>
  <label for="gsearch">Search Google:</label>
  <input type="search" id="gsearch" name="gsearch">
</form>
```

## 58.20 Input Type Tel

The `<input type="tel">` is used for input fields that should contain a telephone number.

### Example:

```
<form>
  <label for="phone">Enter your phone number:</label>
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```

## 58.21 Input Type Time

The `<input type="time">` allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

### Example:

```
<form>
  <label for="appt">Select a time:</label>
  <input type="time" id="appt" name="appt">
</form>
```

## 58.22 Input Type Url

The `<input type="url">` is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

### Example:

```
<form>
  <label for="homepage">Add your homepage:</label>
  <input type="url" id="homepage" name="homepage">
</form>
```



## 58.23 Input Type Week

The `<input type="week">` allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

### Example:

```
<form>
  <label for="week">Select a week:</label>
  <input type="week" id="week" name="week">
</form>
```

## 58.24 HTML Input Type Attribute

Tag	Description
<code>&lt;input type=""&gt;</code>	Specifies the input type to display

## 59 HTML Input Attributes

This chapter describes the different attributes for the HTML `<input>` element.



### 59.1 The value Attribute

The input `value` attribute specifies an initial value for an input field:

#### Example:

Input fields with initial (default) values:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

### 59.2 The readonly Attribute

The input `readonly` attribute specifies that an input field is read-only.

A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it).

The value of a read-only input field will be sent when submitting the form!

#### Example:

A read-only input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" readonly><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

### 59.3 The disabled Attribute

The input `disabled` attribute specifies that an input field should be disabled.

A disabled input field is unusable and un-clickable.

The value of a disabled input field will not be sent when submitting the form!

#### Example:

A disabled input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" disabled><br>
```

```
<label for="lname">Last name:</label><br>
<input type="text" id="lname" name="lname" value="Doe">
</form>
```



## 59.4 The size Attribute

The input `size` attribute specifies the visible width, in characters, of an input field.

The default value for `size` is 20.

**Note:** The `size` attribute works with the following input types: text, search, tel, url, email, and password.

### Example:

Set a width for an input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" size="4">
</form>
```

## 59.5 The maxlength Attribute

The input `maxlength` attribute specifies the maximum number of characters allowed in an input field.

**Note:** When a `maxlength` is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

### Example:

Set a maximum length for an input field:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" maxlength="4" size="4">
</form>
```

## 59.6 The min and max Attributes

The input `min` and `max` attributes specify the minimum and maximum values for an input field.

The `min` and `max` attributes work with the following input types: number, range, date, datetime-local, month, time and week.

**Tip:** Use the max and min attributes together to create a range of legal values.

### Example:



Set a max date, a min date, and a range of legal values:

```
<form>
  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```



## 59.7 The multiple Attribute

The input `multiple` attribute specifies that the user is allowed to enter more than one value in an input field.

The `multiple` attribute works with the following input types: email, and file.

### Example:

A file upload field that accepts multiple values:

```
<form>
  <label for="files">Select files:</label>
  <input type="file" id="files" name="files" multiple>
</form>
```

## 59.8 The pattern Attribute

The input `pattern` attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.

The `pattern` attribute works with the following input types: text, date, search, url, tel, email, and password.

**Tip:** Use the global `title` attribute to describe the pattern to help the user.

### Example:

An input field that can contain only three letters (no numbers or special characters):

```
<form>
  <label for="country_code">Country code:</label>
  <input type="text" id="country_code" name="country_code"
    pattern="[A-Za-z]{3}" title="Three letter country code">
</form>
```

## 59.9 The placeholder Attribute

The input `placeholder` attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).

The short hint is displayed in the input field before the user enters a value.

The `placeholder` attribute works with the following input types: text, search, url, tel, email, and password.

### Example:

An input field with a placeholder text:

```
<form>
  <label for="phone">Enter a phone number:</label>
  <input type="tel" id="phone" name="phone"
    placeholder="123-45-678"
    pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
</form>
```



## 59.10 The required Attribute

The input `required` attribute specifies that an input field must be filled out before submitting the form.

The `required` attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

### Example:

A required input field:

```
<form>
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>
</form>
```

## 59.11 The step Attribute

The input `step` attribute specifies the legal number intervals for an input field.

Example: if `step="3"`, legal numbers could be -3, 0, 3, 6, etc.

**Tip:** This attribute can be used together with the `max` and `min` attributes to create a range of legal values.

The `step` attribute works with the following input types: number, range, date, datetime-local, month, time and week.

### Example:

An input field with a specified legal number intervals:

```
<form>
  <label for="points">Points:</label>
  <input type="number" id="points" name="points" step="3">
</form>
```

**Note:** Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must also be checked by the receiver (the server)!

## 59.12 The autofocus Attribute

The input `autofocus` attribute specifies that an input field should automatically get focus when the page loads.

### Example:

Let the "First name" input field automatically get focus when the page loads:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```



## 59.13 The height and width Attributes

The input `height` and `width` attributes specify the height and width of an `<input type="image">` element.

**Tip:** Always specify both the height and width attributes for images. If height and width are set, the space required for the image is reserved when the page is loaded. Without these attributes, the browser does not know the size of the image, and cannot reserve the appropriate space to it. The effect will be that the page layout will change during loading (while the images load).

### Example:

Define an image as the submit button, with height and width attributes:

```
<form>
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
</form>
```

## 59.14 The list Attribute

The input `list` attribute refers to a `<datalist>` element that contains pre-defined options for an `<input>` element.

### Example:

An `<input>` element with pre-defined values in a `<datalist>`:

```
<form>
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

## 59.15 The autocomplete Attribute

The input `autocomplete` attribute specifies whether a form or an input field should have autocomplete on or off.

Autocomplete allows the browser to predict the value. When a user starts to type in a field, the browser should display options to fill in the field, based on earlier typed values.

The `autocomplete` attribute works with `<form>` and the following `<input>` types: text, search, url, tel, email, password, datepickers, range, and color.

### Example:

An HTML form with autocomplete on, and off for one input field:

```
<form action="/action_page.php" autocomplete="on">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" autocomplete="off"><br><br>
  <input type="submit" value="Submit">
</form>
```

**Tip:** In some browsers you may need to activate an autocomplete function for this to work (Look under "Preferences" in the browser's menu).

## 59.16 HTML Form and Input Elements

Tag	Description
<code>&lt;form&gt;</code>	Defines an HTML form for user input
<code>&lt;input&gt;</code>	Defines an input control

## 60 HTML Input form Attributes

This chapter describes the different `form` attributes for the HTML `<input>` element.



### 60.1 The form Attribute

The input `form` attribute specifies the form the `<input>` element belongs to.

The value of this attribute must be equal to the `id` attribute of the `<form>` element it belongs to.

#### Example:

An input field located outside of the HTML form (but still a part of the form):

```
<form action="/action_page.php" id="form1">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit">
</form>

<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname" form="form1">
```

### 60.2 The formaction Attribute

The input `formaction` attribute specifies the URL of the file that will process the input when the form is submitted.

**Note:** This attribute overrides the `action` attribute of the `<form>` element.

The `formaction` attribute works with the following input types: submit and image.

#### Example:

An HTML form with two submit buttons, with different actions:

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formaction="/action_page2.php" value="Submit as Admin">
</form>
```

### 60.3 The formenctype Attribute

The input `formenctype` attribute specifies how the form-data should be encoded when submitted (only for forms with `method="post"`).

**Note:** This attribute overrides the `enctype` attribute of the `<form>` element.

The `formenctype` attribute works with the following input types: submit and image.

**Example:**

A form with two submit buttons. The first sends the form-data with default encoding, the second sends the form-data encoded as "multipart/form-data":

```
<form action="/action_page_binary.asp" method="post">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formenctype="multipart/form-data"
  value="Submit as Multipart/form-data">
</form>
```



## 60.4 The formmethod Attribute

The input `formmethod` attribute defines the HTTP method for sending form-data to the action URL.

**Note:** This attribute overrides the method attribute of the `<form>` element.

The `formmethod` attribute works with the following input types: submit and image.

The form-data can be sent as URL variables (method="get") or as an HTTP post transaction (method="post").

**Notes on the "get" method:**

This method appends the form-data to the URL in name/value pairs

This method is useful for form submissions where a user want to bookmark the result

There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred

Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)

**Notes on the "post" method:**

This method sends the form-data as an HTTP post transaction

Form submissions with the "post" method cannot be bookmarked

The "post" method is more robust and secure than "get", and "post" does not have size limitations

**Example:**

A form with two submit buttons. The first sends the form-data with method="get". The second sends the form-data with method="post":

```
<form action="/action_page.php" method="get">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit using GET">
  <input type="submit" formmethod="post" value="Submit using POST">
</form>
```

## 60.5 The formtarget Attribute

The input `formtarget` attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

**Note:** This attribute overrides the target attribute of the `<form>` element.

The `formtarget` attribute works with the following input types: submit and image.

### Example:

A form with two submit buttons, with different target windows:

```
<form action="/action_page.php">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formtarget="_blank" value="Submit to a new window/tab">
</form>
```

## 60.6 The formnovalidate Attribute

The input `formnovalidate` attribute specifies that an `<input>` element should not be validated when submitted.

**Note:** This attribute overrides the novalidate attribute of the `<form>` element.

The `formnovalidate` attribute works with the following input types: submit.

### Example:

A form with two submit buttons (with and without validation):

```
<form action="/action_page.php">
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Submit">
  <input type="submit" formnovalidate="formnovalidate"
  value="Submit without validation">
</form>
```

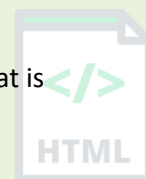
## 60.7 The novalidate Attribute

The `novalidate` attribute is a `<form>` attribute.

When present, `novalidate` specifies that all of the form-data should not be validated when submitted.

### Example:

Specify that no form-data should be validated on submit:



```
<form action="/action_page.php" novalidate>  
  <label for="email">Enter your email:</label>  
  <input type="email" id="email" name="email"><br><br>  
  <input type="submit" value="Submit">  
</form>
```



## 60.8 HTML Form and Input Elements

Tag	Description
<b>&lt;form&gt;</b>	Defines an HTML form for user input
<b>&lt;input&gt;</b>	Defines an input control



## 61 HTML Canvas Graphics

The HTML `<canvas>` element is used to draw graphics on a web page.

The graphic to the left is created with `<canvas>`. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.



### 61.1 What is HTML Canvas?

The HTML `<canvas>` element is used to draw graphics, on the fly, via JavaScript.

The `<canvas>` element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

### 61.2 Canvas Examples

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

**Note:** Always specify an `id` attribute (to be referred to in a script), and a `width` and `height` attribute to define the size of the canvas. To add a border, use the `style` attribute.

Here is an example of a basic, empty canvas:

**Example:**

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
</canvas>
```

### 61.3 Add a JavaScript

After creating the rectangular canvas area, you must add a JavaScript to do the drawing.

Here are some examples:

#### Draw a Line

**Example:**

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0, 0);
ctx.lineTo(200, 100);
ctx.stroke();
</script>
```

#### Draw a Circle

**Example:**

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95, 50, 40, 0, 2 * Math.PI);
ctx.stroke();
</script>
```



## Draw a Text

**Example:**

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillText("Hello World", 10, 50);
</script>
```

## Stroke Text

**Example:**

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.strokeText("Hello World", 10, 50);
</script>
```

## Draw Linear Gradient

**Example:**

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createLinearGradient(0, 0, 200, 0);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```

## Draw Circular Gradient

### Example:

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Create gradient
var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);
grd.addColorStop(0, "red");
grd.addColorStop(1, "white");

// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 150, 80);
</script>
```



## Draw Image

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var img = document.getElementById("scream");
ctx.drawImage(img, 10, 10);
</script>
```

## 62 HTML SVG Graphics

SVG defines vector-based graphics in XML format.



### 62.1 What is SVG?

- SVG stands for Scalable Vector Graphics
- SVG is used to define graphics for the Web

### 62.2 The HTML <svg> Element

The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

#### 1.1.1 SVG Circle

##### Example

```
<!DOCTYPE html>
<html>
<body>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

#### 1.1.2 SVG Rectangle

##### Example:

```
<svg width="400" height="100">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-
width:10;stroke:rgb(0,0,0)" />
</svg>
```

#### 1.1.3 SVG Rounded Rectangle

##### Example

```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
```

```

    style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>

```

## 62.3 SVG Star

### Example

```

<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
    style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>

```



### 1.1.4 SVG Logo

#### Example:

```

<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>

```

```

<!DOCTYPE html>
<html>
<body>
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
      <stop offset="0%"
        style="stop-color:rgb(255,255,0);stop-opacity:1" />
      <stop offset="100%"
        style="stop-color:rgb(255,0,0);stop-opacity:1" />
    </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-family="Verdana"
    x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>

```

## 62.4 Differences Between SVG and Canvas

- SVG is a language for describing 2D graphics in XML.
- Canvas draws 2D graphics, on the fly (with a JavaScript).
- SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.
- In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.
- Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.



## 62.5 Comparison of Canvas and SVG

The table below shows some important differences between Canvas and SVG:

Canvas	SVG
Resolution dependent	Resolution independent
No support for event handlers	Support for event handlers
Poor text rendering capabilities	Best suited for applications with large rendering areas (Google Maps)
You can save the resulting image as .png or .jpg	Slow rendering if complex (anything that uses the DOM a lot will be slow)
Well suited for graphic-intensive games	Not suited for game applications

## 63 HTML Multimedia

Multimedia on the web is sound, music, videos, movies, and animations.



### 63.1 What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

### 63.2 Browser Support

The first web browsers had support for text only, limited to a single font in a single color.

Later came browsers with support for colors, fonts, images, and multimedia!

### 63.3 Multimedia Formats

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

### 63.4 Common Video Formats



There are many video formats out there.

The MP4, WebM, and Ogg formats are supported by HTML.

The MP4 format is recommended by YouTube.

Format	File	Description
<b>MPEG</b>	.mpg .mpeg	MPEG Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML.
<b>AVI</b>	.avi	AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.
<b>WMV</b>	.wmv	WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers.

<b>QuickTime</b>	.mov	QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers.
<b>RealVideo</b>	.rm .ram	RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers.
<b>Flash</b>	.swf .flv	Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers.
<b>Ogg</b>	.ogg	Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.
<b>WebM</b>	.webm	WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML.
<b>MPEG-4 or MP4</b>	.mp4	MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and recommended by YouTube.



**Note:** Only MP4, WebM, and Ogg video are supported by the HTML standard.

## 63.5 Common Audio Formats

MP3 is the best format for compressed recorded music. The term MP3 has become synonymous with digital music.

If your website is about recorded music, MP3 is the choice.

Format	File	Description
MIDI	.mid .midi	MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers.
RealAudio	.rm .ram	RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers.
WMA	.wma	WMA (Windows Media Audio). Developed by Microsoft. Plays well on Windows computers, but not in web browsers.
AAC	.aac	AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers.
WAV	.wav	WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML.
Ogg	.ogg	Ogg. Developed by the Xiph.Org Foundation. Supported by HTML.
MP3	.mp3	MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.
MP4	.mp4	MP4 is a video format, but can also be used for audio. Supported by all browsers.

**Note:** Only MP3, WAV, and Ogg audio are supported by the HTML standard.



## 64 HTML Video

The HTML `<video>` element is used to show a video on a web page.



### Example:

```
<!DOCTYPE html>
<html>
<body>
<video width="400" controls>
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogv" type="video/ogg">
  Your browser does not support HTML video.
</video>
<p>
Video courtesy of
<a href="https://www.bigbuckbunny.org/" target="_blank">Big Buck Bunny</a>.
</p>
</body>
</html>
```

### 64.1 The HTML `<video>` Element

To show a video in HTML, use the `<video>` element:

#### Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogv" type="video/ogg">
Your browser does not support the video tag.
</video>
```

### 64.2 How it Works

The `controls` attribute adds video controls, like play, pause, and volume.

It is a good idea to always include `width` and `height` attributes. If height and width are not set, the page might flicker while the video loads.

The `<source>` element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the `<video>` and `</video>` tags will only be displayed in browsers that do not support the `<video>` element.

### 64.3 HTML `<video>` Autoplay

To start a video automatically, use the `autoplay` attribute:

**Example:**

```
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```



**Note:** Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

Add **muted** after **autoplay** to let your video start playing automatically (but muted):

**Example:**

```
<video width="320" height="240" autoplay muted>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

## 64.4 HTML Video Formats

There are three supported video formats: MP4, WebM, and Ogg. The browser support for the different formats is:

Browser	MP4	WebM	Ogg
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

## 64.5 HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

## 64.6 HTML Video - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the `<video>` element.

This allows you to load, play, and pause videos, as well as setting duration and volume.

There are also DOM events that can notify you when a video begins to play, is paused, etc.

**Example:** Using JavaScript

Play/Pause Big Small Normal



## 64.7 HTML Video Tags

<b>Tag</b>	<b>Description</b>
<code>&lt;video&gt;</code>	Defines a video or movie
<code>&lt;source&gt;</code>	Defines multiple media resources for media elements, such as <code>&lt;video&gt;</code> and <code>&lt;audio&gt;</code>
<code>&lt;track&gt;</code>	Defines text tracks in media players

## 65 HTML Audio

The HTML `<audio>` element is used to play an audio file on a web page.



### 65.1 The HTML `<audio>` Element

To play an audio file in HTML, use the `<audio>` element:

**Example:**

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

### 65.2 HTML Audio - How It Works

The `controls` attribute adds audio controls, like play, pause, and volume.

The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.

### 65.3 HTML `<audio>` Autoplay

To start an audio file automatically, use the `autoplay` attribute:

**Example:**

```
<audio controls autoplay>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

**Note:** Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

Add `muted` after `autoplay` to let your audio file start playing automatically (but muted):

**Example:**

```
<audio controls autoplay muted>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

## 65.4 HTML Audio Formats

There are three supported audio formats: MP3, WAV, and OGG. The browser support for the different formats is:

Browser	MP3	WAV	OGG
Edge/IE	YES	YES*	YES*
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES



## 65.5 HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

## 65.6 HTML Audio - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the `<audio>` element.

This allows you to load, play, and pause audios, as well as set duration and volume.

There are also DOM events that can notify you when an audio begins to play, is paused, etc.

For a full DOM reference, go to our [HTML Audio/Video DOM Reference](#).

## 65.7 HTML Audio Tags

Tag	Description
<code>&lt;audio&gt;</code>	Defines sound content
<code>&lt;source&gt;</code>	Defines multiple media resources for media elements, such as <code>&lt;video&gt;</code> and <code>&lt;audio&gt;</code>



## 66 HTML Plug-ins

Plug-ins are computer programs that extend the standard functionality of the browser.



### 66.1 Plug-ins

Plug-ins were designed to be used for many different purposes:

- To run Java applets
- To run Microsoft ActiveX controls
- To display Flash movies
- To display maps
- To scan for viruses
- To verify a bank id

#### Warning !

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

### 66.2 The <object> Element

The <object> element is supported by all browsers.

The <object> element defines an embedded object within an HTML document.

It was designed to embed plug-ins (like Java applets, PDF readers, and Flash Players) in web pages, but can also be used to include HTML in HTML:

#### Example:

```
<object width="100%" height="500px" data="snippet.html"></object>
```

Or images if you like:

#### Example:

```
<object data="audi.jpeg"></object>
```

### 66.3 The <embed> Element

The <embed> element is supported in all major browsers.

The <embed> element also defines an embedded object within an HTML document.

Web browsers have supported the <embed> element for a long time. However, it has not been a part of the HTML specification before HTML5.

#### Example:

```
<embed src="audi.jpeg">
```

Note that the <embed> element does not have a closing tag. It can not contain alternative text.

The <embed> element can also be used to include HTML in HTML:

**Example:**

```
<embed width="100%" height="500px" src="snippet.html">
```





## 67 HTML YouTube Videos

The easiest way to play videos in HTML, is to use YouTube.



### 67.1 Struggling with Video Formats?

Converting videos to different formats can be difficult and time-consuming.

An easier solution is to let YouTube play the videos in your web page.

### 67.2 YouTube Video Id

YouTube will display an id (like `tgbNymZ7vqY`), when you save (or play) a video.

You can use this id, and refer to your video in the HTML code.

### 67.3 Playing a YouTube Video in HTML

To play your video on a web page, do the following:

- Upload the video to YouTube
- Take a note of the video id
- Define an `<iframe>` element in your web page
- Let the `src` attribute point to the video URL
- Use the `width` and `height` attributes to specify the dimension of the player
- Add any other parameters to the URL (see below)

**Example:**

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY" >
</iframe>
```

### 67.4 YouTube Autoplay + Mute

You can let your video start playing automatically when a user visits the page, by adding `autoplay=1` to the YouTube URL. However, automatically starting a video is annoying for your visitors!

**Note:** Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

Add `mute=1` after `autoplay=1` to let your video start playing automatically (but muted).

## YouTube - Autoplay + Muted

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&mute=1">
</iframe>
```



## 67.5 YouTube Playlist

A comma separated list of videos to play (in addition to the original URL).

## 67.6 YouTube Loop

Add `loop=1` to let your video loop forever.

Value 0 (default): The video will play only once.

Value 1: The video will loop (forever).

### YouTube - Loop

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&loop=1">
</iframe>
```

## 67.7 YouTube Controls

Add `controls=0` to not display controls in the video player.

Value 0: Player controls does not display.

Value 1 (default): Player controls display.

### YouTube - Controls

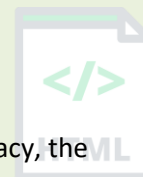
```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">
</iframe>
```

## 68 HTML Geolocation API

The HTML Geolocation API is used to locate a user's position.

### What is geolocation in HTML?

The HTML Geolocation API is **used to get the geographical position of a user**. Since this can compromise privacy, the position is not available unless the user approves it.



**Note:** Geolocation is most accurate for devices with GPS, like smartphones.

### 68.1 Locate the User's Position

The HTML Geolocation API is used to get the geographical position of a user.

Since this can compromise privacy, the position is not available unless the user approves it.

Try It

**Note:** Geolocation is most accurate for devices with GPS, like smartphones.

**Note:** As of Chrome 50, the Geolocation API will only work on secure contexts such as HTTPS. If your site is hosted on a non-secure origin (such as HTTP) the requests to get the users location will no longer function.

### 68.2 Using HTML Geolocation

The `getCurrentPosition()` method is used to return the user's position.

The example below returns the latitude and longitude of the user's position:

#### Example:

```
<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```

#### Example explained:

- Check if Geolocation is supported
- If supported, run the `getCurrentPosition()` method. If not, display a message to the user

- If the `getCurrentPosition()` method is successful, it returns a coordinates object to the function specified in the parameter (`showPosition`)
- The `showPosition()` function outputs the Latitude and Longitude

The example above is a very basic Geolocation script, with no error handling.



## 68.3 Handling Errors and Rejections

The second parameter of the `getCurrentPosition()` method is used to handle errors. It specifies a function to run if it fails to get the user's location:

### Example:

```
function showError(error) {
  switch(error.code) {
    case error.PERMISSION_DENIED:
      x.innerHTML = "User denied the request for Geolocation."
      break;
    case error.POSITION_UNAVAILABLE:
      x.innerHTML = "Location information is unavailable."
      break;
    case error.TIMEOUT:
      x.innerHTML = "The request to get user location timed out."
      break;
    case error.UNKNOWN_ERROR:
      x.innerHTML = "An unknown error occurred."
      break;
  }
}
```

## 68.4 Location-specific Information

This page has demonstrated how to show a user's position on a map.

Geolocation is also very useful for location-specific information, like:

- Up-to-date local information
- Showing Points-of-interest near the user
- Turn-by-turn navigation (GPS)

## 68.5 The `getCurrentPosition()` Method - Return Data

The `getCurrentPosition()` method returns an object on success. The latitude, longitude and accuracy properties are always returned. The other properties are returned if available:

Property	Returns
<code>coords.latitude</code>	The latitude as a decimal number (always returned)
<code>coords.longitude</code>	The longitude as a decimal number (always returned)

<code>coords.accuracy</code>	The accuracy of position (always returned)
<code>coords.altitude</code>	The altitude in meters above the mean sea level (returned if available)
<code>coords.altitudeAccuracy</code>	The altitude accuracy of position (returned if available)
<code>coords.heading</code>	The heading as degrees clockwise from North (returned if available)
<code>coords.speed</code>	The speed in meters per second (returned if available)
<code>timestamp</code>	The date/time of the response (returned if available)



## 68.6 Geolocation Object - Other interesting Methods

The Geolocation object also has other interesting methods:

- `watchPosition()` - Returns the current position of the user and continues to return updated position as the user moves (like the GPS in a car).
- `clearWatch()` - Stops the `watchPosition()` method.

The example below shows the `watchPosition()` method. You need an accurate GPS device to test this (like smartphone):

### Example:

```
<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.watchPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}
function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```

## 69 HTML Drag and Drop API

In HTML, any element can be dragged and dropped.

### Example



Drag the Bintr image into the rectangle.

### 69.1 Drag and Drop

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.

### 69.2 HTML Drag and Drop Example

The example below is a simple drag and drop example:

#### Example:

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
    ev.preventDefault();
}

function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>


```

```
</body>
</html>
```

It might seem complicated, but lets go through all the different parts of a drag and drop event.



## 69.3 Make an Element Draggable

First of all: To make an element draggable, set the `draggable` attribute to true:

```
<img draggable="true">
```

## 69.4 What to Drag - `ondragstart` and `setData()`

Then, specify what should happen when the element is dragged.

In the example above, the `ondragstart` attribute calls a function, `drag(event)`, that specifies what data to be dragged.

The `dataTransfer.setData()` method sets the data type and the value of the dragged data:

```
function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}
```

In this case, the data type is "text" and the value is the id of the draggable element ("drag1").

## 69.5 Where to Drop - `ondragover`

The `ondragover` event specifies where the dragged data can be dropped.

By default, data/elements cannot be dropped in other elements. To allow a drop, we must prevent the default handling of the element.

This is done by calling the `event.preventDefault()` method for the `ondragover` event:

```
event.preventDefault()
```

## 69.6 Do the Drop - `ondrop`

When the dragged data is dropped, a drop event occurs.

In the example above, the `ondrop` attribute calls a function, `drop(event)`:

```
function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
```

**Code explained:**

- Call `preventDefault()` to prevent the browser default handling of the data (default is open as link on drop)

- Get the dragged data with the `dataTransfer.getData()` method. This method will return any data that was set to the same type in the `setData()` method
- The dragged data is the id of the dragged element ("drag1")
- Append the dragged element into the drop element





## 70 HTML Web Storage API

HTML web storage; better than cookies.



### 70.1 What is HTML Web Storage?

With web storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

### 70.2 HTML Web Storage Objects

HTML web storage provides two objects for storing data on the client:

- `window.localStorage` - stores data with no expiration date
- `window.sessionStorage` - stores data for one session (data is lost when the browser tab is closed)

Before using web storage, check browser support for `localStorage` and `sessionStorage`:

```
if (typeof(Storage) !== "undefined") {  
    // Code for localStorage/sessionStorage.  
} else {  
    // Sorry! No Web Storage support..  
}
```

### 70.3 The localStorage Object

The `localStorage` object stores the data with no expiration date. The data will not be deleted when the browser is closed, and will be available the day, week, or year.

#### Example:

```
// Store  
localStorage.setItem("lastname", "Smith");  
  
// Retrieve  
document.getElementById("result").innerHTML = localStorage.getItem("lastname");
```

Example explained:

- Create a `localStorage` name/value pair with name="lastname" and value="Smith"
- Retrieve the value of "lastname" and insert it into the element with id="result"

The example above could also be written like this:

```
// Store
localStorage.lastname = "Smith";
// Retrieve
document.getElementById("result").innerHTML = localStorage.lastname;
```

The syntax for removing the "lastname" localStorage item is as follows:

```
localStorage.removeItem("lastname");
```



**Note:** Name/value pairs are always stored as strings. Remember to convert them to another format when needed!

The following example counts the number of times a user has clicked a button. In this code the value string is converted to a number to be able to increase the counter:

**Example:**

```
if (localStorage.clickcount) {
    localStorage.clickcount = Number(localStorage.clickcount) + 1;
} else {
    localStorage.clickcount = 1;
}
document.getElementById("result").innerHTML = "You have clicked the button " +
localStorage.clickcount + " time(s).";
```

## 70.4 The sessionStorage Object

The `sessionStorage` object is equal to the `localStorage` object, **except** that it stores the data for only one session. The data is deleted when the user closes the specific browser tab.

The following example counts the number of times a user has clicked a button, in the current session:

**Example:**

```
if (sessionStorage.clickcount) {
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;
} else {
    sessionStorage.clickcount = 1;
}
document.getElementById("result").innerHTML = "You have clicked the button " +
sessionStorage.clickcount + " time(s) in this session.";
```

## 71 HTML Web Workers API

A web worker is a JavaScript running in the background, without affecting the performance of the page.

### 71.1 What is a Web Worker?

When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.

A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.



### 71.2 HTML Web Workers Example

The example below creates a simple web worker that count numbers in the background:

#### Example:

Count numbers:

Start Worker Stop Worker

### 71.3 Check Web Worker Support

Before creating a web worker, check whether the user's browser supports it:

```
if (typeof(Worker) !== "undefined") {  
    // Yes! Web worker support!  
    // Some code.....  
} else {  
    // Sorry! No Web Worker support..  
}
```

### 71.4 Create a Web Worker File

Now, let's create our web worker in an external JavaScript.

Here, we create a script that counts. The script is stored in the "demo\_workers.js" file:

```
var i = 0;  
  
function timedCount() {  
    i = i + 1;  
    postMessage(i);  
    setTimeout("timedCount()",500);  
}  
  
timedCount();
```

The important part of the code above is the `postMessage()` method - which is used to post a message back to the HTML page.

**Note:** Normally web workers are not used for such simple scripts, but for more CPU intensive tasks.



## 71.5 Create a Web Worker Object

Now that we have the web worker file, we need to call it from an HTML page.

The following lines checks if the worker already exists, if not - it creates a new web worker object and runs the code in "demo\_workers.js":

```
if (typeof(w) == "undefined") {  
  w = new Worker("demo_workers.js");  
}
```

Then we can send and receive messages from the web worker.

Add an "onmessage" event listener to the web worker.

```
w.onmessage = function(event){  
  document.getElementById("result").innerHTML = event.data;  
};
```

When the web worker posts a message, the code within the event listener is executed. The data from the web worker is stored in event.data.

## 71.6 Terminate a Web Worker

When a web worker object is created, it will continue to listen for messages (even after the external script is finished) until it is terminated.

To terminate a web worker, and free browser/computer resources, use the `terminate()` method:

```
w.terminate();
```

## 71.7 Reuse the Web Worker

If you set the worker variable to undefined, after it has been terminated, you can reuse the code:

```
w = undefined;
```

## 71.8 Full Web Worker Example Code

We have already seen the Worker code in the .js file. Below is the code for the HTML page:

**Example:**

```
<!DOCTYPE html>  
<html>
```

```
<body>
```

```
<p>Count numbers: <output id="result"></output></p>  
<button onclick="startWorker()">Start Worker</button>  
<button onclick="stopWorker()">Stop Worker</button>
```

```
<script>
```

```
var w;
```

```
function startWorker() {  
  if (typeof(Worker) !== "undefined") {  
    if (typeof(w) == "undefined") {  
      w = new Worker("demo_workers.js");  
    }  
    w.onmessage = function(event) {  
      document.getElementById("result").innerHTML = event.data;  
    };  
  } else {  
    document.getElementById("result").innerHTML = "Sorry! No Web Worker support.";  
  }  
}
```

```
function stopWorker() {  
  w.terminate();  
  w = undefined;  
}
```

```
</script>
```

```
</body>
```

```
</html>
```



## 71.9 Web Workers and the DOM

Since web workers are in external files, they do not have access to the following JavaScript objects:

- The window object
- The document object
- The parent object

## 72 HTML SSE API

Server-Sent Events (SSE) allow a web page to get updates from a server.



### 72.1 Server-Sent Events - One Way Messaging

A server-sent event is when a web page automatically gets updates from a server.

This was also possible before, but the web page would have to ask if any updates were available. With server-sent events, the updates come automatically.

Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.

### 72.2 Receive Server-Sent Event Notifications

The EventSource object is used to receive server-sent event notifications:

#### Example:

```
var source = new EventSource("demo_sse.php");
source.onmessage = function(event) {
    document.getElementById("result").innerHTML += event.data + "<br>";
};
```

#### Example explained:

- Create a new EventSource object, and specify the URL of the page sending the updates
- Each time an update is received, the onmessage event occurs
- When an onmessage event occurs, put the received data into the element with id="result"

### 72.3 Check Server-Sent Events Support

In the example above there were some extra lines of code to check browser support for server-sent events:

```
if(typeof(EventSource) !== "undefined") {
    // Yes! Server-sent events support!
    // Some code.....
} else {
    // Sorry! No server-sent events support..
}
```

### 72.4 Server-Side Code Example

For the example above to work, you need a server capable of sending data updates (like PHP or ASP).

The server-side event stream syntax is simple. Set the "Content-Type" header to "text/event-stream". Now you can start sending event streams.

Code in PHP (demo\_sse.php):

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');

$time = date('r');
echo "data: The server time is: {$time}\n\n";
flush();
?>
```



Code in ASP (VB) (demo\_sse.asp):

```
<%
Response.ContentType = "text/event-stream"
Response.Expires = -1
Response.Write("data: The server time is: " & now())
Response.Flush()
%>
```

#### Code explained:

- Set the "Content-Type" header to "text/event-stream"
- Specify that the page should not cache
- Output the data to send (**Always** start with "data: ")
- Flush the output data back to the web page

## 72.5 The EventSource Object

In the examples above we used the onmessage event to get messages. But other events are also available:

Events	Description
onopen	When a connection to the server is opened
onmessage	When a message is received
onerror	When an error occurs

## 73 HTML Accessibility

### 73.1 HTML Accessibility

Always write HTML code with accessibility in mind!

Provide the user a good way to navigate and interact with your site. Make your HTML code as **semantic** as possible.



### 73.2 Semantic HTML

Semantic HTML means using correct HTML elements for their correct purpose as much as possible. Semantic elements are elements with a meaning; if you need a button, use the `<button>` element (and not a `<div>` element).

#### Semantic

```
<button>Report an Error</button>
```

#### Non-semantic

```
<div>Report an Error</div>
```

Semantic HTML gives context to screen readers, which read the contents of a page out loud.

button:

- buttons have more suitable styling by default
- a screen reader identifies it as a button
- focusable
- clickable

A button is also accessible for people relying on keyboard-only navigation; it can be clickable with both mouse and keys, and it can be tabbed between (using the tab key on the keyboard).

Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.

Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

### 73.3 Headings Are Important

Headings are defined with the `<h1>` to `<h6>` tags:

#### Example:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Search engines use the headings to index the structure and content of your web pages.



Users skim your pages by its headings. It is important to use headings to show the document structure and the relationships between different sections.

Screen readers also use headings as a navigational tool. The different types of heading specify the outline of the page. `<h1>` headings should be used for main headings, followed by `<h2>` headings, then the less important `<h3>`, and so on.



**Note:** Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

## 73.4 Alternative Text

The `alt` attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader).

The value of the `alt` attribute should describe the image:

### Example:

```

```

If a browser cannot find an image, it will display the value of the `alt` attribute:

### Example:

```

```

## 73.5 Declare the Language

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

```
<!DOCTYPE html>  
<html lang="en">  
<body>
```

...

```
</body>  
</html>
```

## 73.6 Use Clear Language

Always use a clear language, that is easy to understand. Also try to avoid characters that cannot be read clearly by a screen reader. For example:

- Keep sentences as short as possible
- Avoid dashes. Instead of writing 1-3, write 1 to 3
- Avoid abbreviations. Instead of writing Feb, write February
- Avoid slang words

## 73.7 Create Good Link Text

A link text should explain clearly what information the reader will get by clicking on that link.



**Note:** This page is an introduction in web accessibility. Visit our Accessibility Tutorial for more details.

## 74 HTML Element Reference

### 74.1 HTML Tags Ordered Alphabetically



Tag	Description
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink
<abbr>	Defines an abbreviation or an acronym
<acronym>	Not supported in HTML5. Use <abbr> instead. Defines an acronym
<address>	Defines contact information for the author/owner of a document
<applet>	Not supported in HTML5. Use <embed> or <object> instead. Defines an embedded applet
<area>	Defines an area inside an image map
<article>	Defines an article
<aside>	Defines content aside from the page content
<audio>	Defines embedded sound content
<b>	Defines bold text
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont>	Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<bdo>	Overrides the current text direction
<big>	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote>	Defines a section that is quoted from another source
<body>	Defines the document's body
 	Defines a single line break
<button>	Defines a clickable button
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<caption>	Defines a table caption
<center>	Not supported in HTML5. Use CSS instead. Defines centered text
<cite>	Defines the title of a work

<b>&lt;code&gt;</b>	Defines a piece of computer code
<b>&lt;col&gt;</b>	Specifies column properties for each column within a <colgroup> element
<b>&lt;colgroup&gt;</b>	Specifies a group of one or more columns in a table for formatting
<b>&lt;data&gt;</b>	Adds a machine-readable translation of a given content
<b>&lt;datalist&gt;</b>	Specifies a list of pre-defined options for input controls
<b>&lt;dd&gt;</b>	Defines a description/value of a term in a description list
<b>&lt;del&gt;</b>	Defines text that has been deleted from a document
<b>&lt;details&gt;</b>	Defines additional details that the user can view or hide
<b>&lt;dfn&gt;</b>	Specifies a term that is going to be defined within the content
<b>&lt;dialog&gt;</b>	Defines a dialog box or window
<b>&lt;dir&gt;</b>	Not supported in HTML5. Use <ul> instead. Defines a directory list
<b>&lt;div&gt;</b>	Defines a section in a document
<b>&lt;dl&gt;</b>	Defines a description list
<b>&lt;dt&gt;</b>	Defines a term/name in a description list
<b>&lt;em&gt;</b>	Defines emphasized text
<b>&lt;embed&gt;</b>	Defines a container for an external application
<b>&lt;fieldset&gt;</b>	Groups related elements in a form
<b>&lt;figcaption&gt;</b>	Defines a caption for a <figure> element
<b>&lt;figure&gt;</b>	Specifies self-contained content
<b>&lt;font&gt;</b>	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<b>&lt;footer&gt;</b>	Defines a footer for a document or section
<b>&lt;form&gt;</b>	Defines an HTML form for user input
<b>&lt;frame&gt;</b>	Not supported in HTML5. Defines a window (a frame) in a frameset
<b>&lt;frameset&gt;</b>	Not supported in HTML5. Defines a set of frames
<b>&lt;h1&gt; to &lt;h6&gt;</b>	Defines HTML headings
<b>&lt;head&gt;</b>	Contains metadata/information for the document
<b>&lt;header&gt;</b>	Defines a header for a document or section
<b>&lt;hr&gt;</b>	Defines a thematic change in the content
<b>&lt;html&gt;</b>	Defines the root of an HTML document
<b>&lt;i&gt;</b>	Defines a part of text in an alternate voice or mood
<b>&lt;iframe&gt;</b>	Defines an inline frame



<b>&lt;img&gt;</b>	Defines an image
<b>&lt;input&gt;</b>	Defines an input control
<b>&lt;ins&gt;</b>	Defines a text that has been inserted into a document
<b>&lt;kbd&gt;</b>	Defines keyboard input
<b>&lt;label&gt;</b>	Defines a label for an <input> element
<b>&lt;legend&gt;</b>	Defines a caption for a <fieldset> element
<b>&lt;li&gt;</b>	Defines a list item
<b>&lt;link&gt;</b>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<b>&lt;main&gt;</b>	Specifies the main content of a document
<b>&lt;map&gt;</b>	Defines an image map
<b>&lt;mark&gt;</b>	Defines marked/highlighted text
<b>&lt;meta&gt;</b>	Defines metadata about an HTML document
<b>&lt;meter&gt;</b>	Defines a scalar measurement within a known range (a gauge)
<b>&lt;nav&gt;</b>	Defines navigation links
<b>&lt;noframes&gt;</b>	Not supported in HTML5. Defines an alternate content for users that do not support frames
<b>&lt;noscript&gt;</b>	Defines an alternate content for users that do not support client-side scripts
<b>&lt;object&gt;</b>	Defines a container for an external application
<b>&lt;ol&gt;</b>	Defines an ordered list
<b>&lt;optgroup&gt;</b>	Defines a group of related options in a drop-down list
<b>&lt;option&gt;</b>	Defines an option in a drop-down list
<b>&lt;output&gt;</b>	Defines the result of a calculation
<b>&lt;p&gt;</b>	Defines a paragraph
<b>&lt;param&gt;</b>	Defines a parameter for an object
<b>&lt;picture&gt;</b>	Defines a container for multiple image resources
<b>&lt;pre&gt;</b>	Defines preformatted text
<b>&lt;progress&gt;</b>	Represents the progress of a task
<b>&lt;q&gt;</b>	Defines a short quotation
<b>&lt;rp&gt;</b>	Defines what to show in browsers that do not support ruby annotations
<b>&lt;rt&gt;</b>	Defines an explanation/pronunciation of characters (for East Asian typography)
<b>&lt;ruby&gt;</b>	Defines a ruby annotation (for East Asian typography)
<b>&lt;s&gt;</b>	Defines text that is no longer correct



<b>&lt;samp&gt;</b>	Defines sample output from a computer program
<b>&lt;script&gt;</b>	Defines a client-side script
<b>&lt;section&gt;</b>	Defines a section in a document
<b>&lt;select&gt;</b>	Defines a drop-down list
<b>&lt;small&gt;</b>	Defines smaller text
<b>&lt;source&gt;</b>	Defines multiple media resources for media elements (<video> and <audio>)
<b>&lt;span&gt;</b>	Defines a section in a document
<b>&lt;strike&gt;</b>	Not supported in HTML5. Use <del> or <s> instead. Defines strikethrough text
<b>&lt;strong&gt;</b>	Defines important text
<b>&lt;style&gt;</b>	Defines style information for a document
<b>&lt;sub&gt;</b>	Defines subscripted text
<b>&lt;summary&gt;</b>	Defines a visible heading for a <details> element
<b>&lt;sup&gt;</b>	Defines superscripted text
<b>&lt;svg&gt;</b>	Defines a container for SVG graphics
<b>&lt;table&gt;</b>	Defines a table
<b>&lt;tbody&gt;</b>	Groups the body content in a table
<b>&lt;td&gt;</b>	Defines a cell in a table
<b>&lt;template&gt;</b>	Defines a container for content that should be hidden when the page loads
<b>&lt;textarea&gt;</b>	Defines a multiline input control (text area)
<b>&lt;tfoot&gt;</b>	Groups the footer content in a table
<b>&lt;th&gt;</b>	Defines a header cell in a table
<b>&lt;thead&gt;</b>	Groups the header content in a table
<b>&lt;time&gt;</b>	Defines a specific time (or datetime)
<b>&lt;title&gt;</b>	Defines a title for the document
<b>&lt;tr&gt;</b>	Defines a row in a table
<b>&lt;track&gt;</b>	Defines text tracks for media elements (<video> and <audio>)
<b>&lt;tt&gt;</b>	Not supported in HTML5. Use CSS instead. Defines teletype text
<b>&lt;u&gt;</b>	Defines some text that is unarticulated and styled differently from normal text
<b>&lt;ul&gt;</b>	Defines an unordered list
<b>&lt;var&gt;</b>	Defines a variable
<b>&lt;video&gt;</b>	Defines embedded video content



**<wbr>**

Defines a possible line-break



## 75 HTML Element Reference - By Category

### 75.1 Basic HTML



Tag	Description
<!DOCTYPE>	Defines the document type
<html>	Defines an HTML document
<head>	Contains metadata/information for the document
<title>	Defines a title for the document
<body>	Defines the document's body
<h1> to <h6>	Defines HTML headings
<p>	Defines a paragraph
 	Inserts a single line break
<hr>	Defines a thematic change in the content
<!--...-->	Defines a comment

### 75.2 Formatting

Tag	Description
<acronym>	<b>Not supported in HTML5. Use &lt;abbr&gt; instead.</b> Defines an acronym
<abbr>	Defines an abbreviation or an acronym
<address>	Defines contact information for the author/owner of a document/article
<b>	Defines bold text
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<bdo>	Overrides the current text direction
<big>	<b>Not supported in HTML5. Use CSS instead.</b> Defines big text
<blockquote>	Defines a section that is quoted from another source
<center>	<b>Not supported in HTML5. Use CSS instead.</b> Defines centered text
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
<del>	Defines text that has been deleted from a document
<dfn>	Specifies a term that is going to be defined within the content



<code>&lt;em&gt;</code>	Defines emphasized text
<code>&lt;font&gt;</code>	<b>Not supported in HTML5. Use CSS instead.</b> Defines font, color, and size for text
<code>&lt;i&gt;</code>	Defines a part of text in an alternate voice or mood
<code>&lt;ins&gt;</code>	Defines a text that has been inserted into a document
<code>&lt;kbd&gt;</code>	Defines keyboard input
<code>&lt;mark&gt;</code>	Defines marked/highlighted text
<code>&lt;meter&gt;</code>	Defines a scalar measurement within a known range (a gauge)
<code>&lt;pre&gt;</code>	Defines preformatted text
<code>&lt;progress&gt;</code>	Represents the progress of a task
<code>&lt;q&gt;</code>	Defines a short quotation
<code>&lt;rp&gt;</code>	Defines what to show in browsers that do not support ruby annotations
<code>&lt;rt&gt;</code>	Defines an explanation/pronunciation of characters (for East Asian typography)
<code>&lt;ruby&gt;</code>	Defines a ruby annotation (for East Asian typography)
<code>&lt;s&gt;</code>	Defines text that is no longer correct
<code>&lt;samp&gt;</code>	Defines sample output from a computer program
<code>&lt;small&gt;</code>	Defines smaller text
<code>&lt;strike&gt;</code>	<b>Not supported in HTML5. Use <code>&lt;del&gt;</code> or <code>&lt;s&gt;</code> instead.</b> Defines strikethrough text
<code>&lt;strong&gt;</code>	Defines important text
<code>&lt;sub&gt;</code>	Defines subscripted text
<code>&lt;sup&gt;</code>	Defines superscripted text
<code>&lt;template&gt;</code>	Defines a container for content that should be hidden when the page loads
<code>&lt;time&gt;</code>	Defines a specific time (or datetime)
<code>&lt;tt&gt;</code>	<b>Not supported in HTML5. Use CSS instead.</b> Defines teletype text
<code>&lt;u&gt;</code>	Defines some text that is unarticulated and styled differently from normal text
<code>&lt;var&gt;</code>	Defines a variable
<code>&lt;wbr&gt;</code>	Defines a possible line-break



## 75.3 Forms and Input

Tag	Description
-----	-------------

<b>&lt;form&gt;</b>	Defines an HTML form for user input
<b>&lt;input&gt;</b>	Defines an input control
<b>&lt;textarea&gt;</b>	Defines a multiline input control (text area)
<b>&lt;button&gt;</b>	Defines a clickable button
<b>&lt;select&gt;</b>	Defines a drop-down list
<b>&lt;optgroup&gt;</b>	Defines a group of related options in a drop-down list
<b>&lt;option&gt;</b>	Defines an option in a drop-down list
<b>&lt;label&gt;</b>	Defines a label for an <input> element
<b>&lt;fieldset&gt;</b>	Groups related elements in a form
<b>&lt;legend&gt;</b>	Defines a caption for a <fieldset> element
<b>&lt;datalist&gt;</b>	Specifies a list of pre-defined options for input controls
<b>&lt;output&gt;</b>	Defines the result of a calculation



## 75.4 Frames

Tag	Description
<b>&lt;frame&gt;</b>	<b>Not supported in HTML5.</b> Defines a window (a frame) in a frameset
<b>&lt;frameset&gt;</b>	<b>Not supported in HTML5.</b> Defines a set of frames
<b>&lt;noframes&gt;</b>	<b>Not supported in HTML5.</b> Defines an alternate content for users that do not support frames
<b>&lt;iframe&gt;</b>	Defines an inline frame

## 75.5 Images

Tag	Description
<b>&lt;img&gt;</b>	Defines an image
<b>&lt;map&gt;</b>	Defines a client-side image map
<b>&lt;area&gt;</b>	Defines an area inside an image map
<b>&lt;canvas&gt;</b>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<b>&lt;figcaption&gt;</b>	Defines a caption for a <figure> element
<b>&lt;figure&gt;</b>	Specifies self-contained content
<b>&lt;picture&gt;</b>	Defines a container for multiple image resources
<b>&lt;svg&gt;</b>	Defines a container for SVG graphics

## 75.6 Audio / Video

Tag	Description
<code>&lt;audio&gt;</code>	Defines sound content
<code>&lt;source&gt;</code>	Defines multiple media resources for media elements ( <code>&lt;video&gt;</code> , <code>&lt;audio&gt;</code> and <code>&lt;picture&gt;</code> )
<code>&lt;track&gt;</code>	Defines text tracks for media elements ( <code>&lt;video&gt;</code> and <code>&lt;audio&gt;</code> )
<code>&lt;video&gt;</code>	Defines a video or movie



## 75.7 Links

Tag	Description
<code>&lt;a&gt;</code>	Defines a hyperlink
<code>&lt;link&gt;</code>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<code>&lt;nav&gt;</code>	Defines navigation links

## 75.8 Lists

Tag	Description
<code>&lt;ul&gt;</code>	Defines an unordered list
<code>&lt;ol&gt;</code>	Defines an ordered list
<code>&lt;li&gt;</code>	Defines a list item
<code>&lt;dir&gt;</code>	<b>Not supported in HTML5. Use <code>&lt;ul&gt;</code> instead.</b> Defines a directory list
<code>&lt;dl&gt;</code>	Defines a description list
<code>&lt;dt&gt;</code>	Defines a term/name in a description list
<code>&lt;dd&gt;</code>	Defines a description of a term/name in a description list

## 75.9 Tables

Tag	Description
<code>&lt;table&gt;</code>	Defines a table
<code>&lt;caption&gt;</code>	Defines a table caption
<code>&lt;th&gt;</code>	Defines a header cell in a table
<code>&lt;tr&gt;</code>	Defines a row in a table
<code>&lt;td&gt;</code>	Defines a cell in a table
<code>&lt;thead&gt;</code>	Groups the header content in a table
<code>&lt;tbody&gt;</code>	Groups the body content in a table

<code>&lt;tfoot&gt;</code>	Groups the footer content in a table
<code>&lt;col&gt;</code>	Specifies column properties for each column within a <code>&lt;colgroup&gt;</code> element
<code>&lt;colgroup&gt;</code>	Specifies a group of one or more columns in a table for formatting



## 75.10 Styles and Semantics

Tag	Description
<code>&lt;style&gt;</code>	Defines style information for a document
<code>&lt;div&gt;</code>	Defines a section in a document
<code>&lt;span&gt;</code>	Defines a section in a document
<code>&lt;header&gt;</code>	Defines a header for a document or section
<code>&lt;footer&gt;</code>	Defines a footer for a document or section
<code>&lt;main&gt;</code>	Specifies the main content of a document
<code>&lt;section&gt;</code>	Defines a section in a document
<code>&lt;article&gt;</code>	Defines an article
<code>&lt;aside&gt;</code>	Defines content aside from the page content
<code>&lt;details&gt;</code>	Defines additional details that the user can view or hide
<code>&lt;dialog&gt;</code>	Defines a dialog box or window
<code>&lt;summary&gt;</code>	Defines a visible heading for a <code>&lt;details&gt;</code> element
<code>&lt;data&gt;</code>	Adds a machine-readable translation of a given content

## 75.11 Meta Info

Tag	Description
<code>&lt;head&gt;</code>	Defines information about the document
<code>&lt;meta&gt;</code>	Defines metadata about an HTML document
<code>&lt;base&gt;</code>	Specifies the base URL/target for all relative URLs in a document

`<basefont>`**Not supported in HTML5. Use CSS instead.**

Specifies a default color, size, and font for all text in a document

## 75.12 Programming



HTML

Tag	Description
<code>&lt;script&gt;</code>	Defines a client-side script
<code>&lt;noscript&gt;</code>	Defines an alternate content for users that do not support client-side scripts
<code>&lt;applet&gt;</code>	<b>Not supported in HTML5. Use <code>&lt;embed&gt;</code> or <code>&lt;object&gt;</code> instead.</b> Defines an embedded applet
<code>&lt;embed&gt;</code>	Defines a container for an external (non-HTML) application
<code>&lt;object&gt;</code>	Defines an embedded object
<code>&lt;param&gt;</code>	Defines a parameter for an object

## 76 HTML Attribute Reference

### 76.1 HTML Attribute Reference

The table below lists all HTML attributes and what elements they can be used within:



Attribute	Belongs to	Description
<b>accept</b>	<input>	Specifies the types of files that the server accepts (only for type="file")
<b>accept-charset</b>	<form>	Specifies the character encodings that are to be used for the form submission
<b>accesskey</b>	Global Attributes	Specifies a shortcut key to activate/focus an element
<b>action</b>	<form>	Specifies where to send the form-data when a form is submitted
<b>align</b>	Not supported in HTML 5.	Specifies the alignment according to surrounding elements. Use CSS instead
<b>alt</b>	<area>, <img>, <input>	Specifies an alternate text when the original element fails to display
<b>async</b>	<script>	Specifies that the script is executed asynchronously (only for external scripts)
<b>autocomplete</b>	<form>, <input>	Specifies whether the <form> or the <input> element should have autocomplete enabled
<b>autofocus</b>	<button>, <input>, <select>, <textarea>	Specifies that the element should automatically get focus when the page loads
<b>autoplay</b>	<audio>, <video>	Specifies that the audio/video will start playing as soon as it is ready
<b>bgcolor</b>	Not supported in HTML 5.	Specifies the background color of an element. Use CSS instead
<b>border</b>	Not supported in HTML 5.	Specifies the width of the border of an element. Use CSS instead
<b>charset</b>	<meta>, <script>	Specifies the character encoding
<b>checked</b>	<input>	Specifies that an <input> element should be pre-selected when the page loads (for type="checkbox" or type="radio")
<b>cite</b>	<blockquote>, <del>, <ins>, <q>	Specifies a URL which explains the quote/deleted/inserted text
<b>class</b>	Global Attributes	Specifies one or more classnames for an element (refers to a class in a style sheet)
<b>color</b>	Not supported in HTML 5.	Specifies the text color of an element. Use CSS instead
<b>cols</b>	<textarea>	Specifies the visible width of a text area



<b>colspan</b>	<td>, <th>	Specifies the number of columns a table cell should span
<b>content</b>	<meta>	Gives the value associated with the http-equiv or name attribute
<b>contenteditable</b>	Global Attributes	Specifies whether the content of an element is editable or not
<b>controls</b>	<audio>, <video>	Specifies that audio/video controls should be displayed (such as a play/pause button etc)
<b>coords</b>	<area>	Specifies the coordinates of the area
<b>data</b>	<object>	Specifies the URL of the resource to be used by the object
<b>data-*</b>	Global Attributes	Used to store custom data private to the page or application
<b>datetime</b>	<del>, <ins>, <time>	Specifies the date and time
<b>default</b>	<track>	Specifies that the track is to be enabled if the user's preferences do not indicate that another track would be more appropriate
<b>defer</b>	<script>	Specifies that the script is executed when the page has finished parsing (only for external scripts)
<b>dir</b>	Global Attributes	Specifies the text direction for the content in an element
<b>dirname</b>	<input>, <textarea>	Specifies that the text direction will be submitted
<b>disabled</b>	<button>, <fieldset>, <input>, <optgroup>, <option>, <select>, <textarea>	Specifies that the specified element/group of elements should be disabled
<b>download</b>	<a>, <area>	Specifies that the target will be downloaded when a user clicks on the hyperlink
<b>draggable</b>	Global Attributes	Specifies whether an element is draggable or not
<b>enctype</b>	<form>	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
<b>for</b>	<label>, <output>	Specifies which form element(s) a label/calculation is bound to
<b>form</b>	<button>, <fieldset>, <input>, <label>, <meter>, <object>, <output>, <select>, <textarea>	Specifies the name of the form the element belongs to
<b>formaction</b>	<button>, <input>	Specifies where to send the form-data when a form is submitted. Only for type="submit"
<b>headers</b>	<td>, <th>	Specifies one or more headers cells a cell is related to
<b>height</b>	<canvas>, <embed>, <iframe>, <img>	Specifies the height of the element



	>, <input>, <object>, <video>	
<b>hidden</b>	Global Attributes	Specifies that an element is not yet, or is no longer, relevant
<b>high</b>	<meter>	Specifies the range that is considered to be a high value
<b>href</b>	<a>, <area>, <base>, <link>	Specifies the URL of the page the link goes to
<b>hreflang</b>	<a>, <area>, <link>	Specifies the language of the linked document
<b>http-equiv</b>	<meta>	Provides an HTTP header for the information/value of the content attribute
<b>id</b>	Global Attributes	Specifies a unique id for an element
<b>ismap</b>	<img>	Specifies an image as a server-side image map
<b>kind</b>	<track>	Specifies the kind of text track
<b>label</b>	<track>, <option>, <optgroup>	Specifies the title of the text track
<b>lang</b>	Global Attributes	Specifies the language of the element's content
<b>list</b>	<input>	Refers to a <datalist> element that contains pre-defined options for an <input> element
<b>loop</b>	<audio>, <video>	Specifies that the audio/video will start over again, every time it is finished
<b>low</b>	<meter>	Specifies the range that is considered to be a low value
<b>max</b>	<input>, <meter>, <progress>	Specifies the maximum value
<b>maxlength</b>	<input>, <textarea>	Specifies the maximum number of characters allowed in an element
<b>media</b>	<a>, <area>, <link>, <source>, <style>	Specifies what media/device the linked document is optimized for
<b>method</b>	<form>	Specifies the HTTP method to use when sending form-data
<b>min</b>	<input>, <meter>	Specifies a minimum value
<b>multiple</b>	<input>, <select>	Specifies that a user can enter more than one value
<b>muted</b>	<video>, <audio>	Specifies that the audio output of the video should be muted
<b>name</b>	<button>, <fieldset>, <form>, <iframe>, <input>, <map>, <meta>, <object>, <output>, <param>, <select>, <textarea>	Specifies the name of the element





<b>novalidate</b>	<form>	Specifies that the form should not be validated when submitted
<b>onabort</b>	<audio>, <embed>, <img>, <object>, <video>	Script to be run on abort
<b>onafterprint</b>	<body>	Script to be run after the document is printed
<b>onbeforeprint</b>	<body>	Script to be run before the document is printed
<b>onbeforeunload</b>	<body>	Script to be run when the document is about to be unloaded
<b>onblur</b>	All visible elements.	Script to be run when the element loses focus
<b>oncanplay</b>	<audio>, <embed>, <object>, <video>	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
<b>oncanplaythrough</b>	<audio>, <video>	Script to be run when a file can be played all the way to the end without pausing for buffering
<b>onchange</b>	All visible elements.	Script to be run when the value of the element is changed
<b>onclick</b>	All visible elements.	Script to be run when the element is being clicked
<b>oncontextmenu</b>	All visible elements.	Script to be run when a context menu is triggered
<b>oncopy</b>	All visible elements.	Script to be run when the content of the element is being copied
<b>oncuechange</b>	<track>	Script to be run when the cue changes in a <track> element
<b>oncut</b>	All visible elements.	Script to be run when the content of the element is being cut
<b>ondblclick</b>	All visible elements.	Script to be run when the element is being double-clicked
<b>ondrag</b>	All visible elements.	Script to be run when the element is being dragged
<b>ondragend</b>	All visible elements.	Script to be run at the end of a drag operation
<b>ondragenter</b>	All visible elements.	Script to be run when an element has been dragged to a valid drop target
<b>ondragleave</b>	All visible elements.	Script to be run when an element leaves a valid drop target



<b>ondragover</b>	All visible elements.	Script to be run when an element is being dragged over a valid drop target
<b>ondragstart</b>	All visible elements.	Script to be run at the start of a drag operation
<b>ondrop</b>	All visible elements.	Script to be run when dragged element is being dropped
<b>ondurationchange</b>	<audio>, <video>	Script to be run when the length of the media changes
<b>onemptied</b>	<audio>, <video>	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
<b>onended</b>	<audio>, <video>	Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")
<b>onerror</b>	<audio>, <body>, <embed>, <img>, <object>, <script>, <style>, <video>	Script to be run when an error occurs
<b>onfocus</b>	All visible elements.	Script to be run when the element gets focus
<b>onhashchange</b>	<body>	Script to be run when there has been changes to the anchor part of the a URL
<b>oninput</b>	All visible elements.	Script to be run when the element gets user input
<b>oninvalid</b>	All visible elements.	Script to be run when the element is invalid
<b>onkeydown</b>	All visible elements.	Script to be run when a user is pressing a key
<b>onkeypress</b>	All visible elements.	Script to be run when a user presses a key
<b>onkeyup</b>	All visible elements.	Script to be run when a user releases a key
<b>onload</b>	<body>, <iframe>, <img>, <input>, <link>, <script>, <style>	Script to be run when the element is finished loading
<b>onloadeddata</b>	<audio>, <video>	Script to be run when media data is loaded
<b>onloadedmetadata</b>	<audio>, <video>	Script to be run when meta data (like dimensions and duration) are loaded
<b>onloadstart</b>	<audio>, <video>	Script to be run just as the file begins to load before anything is actually loaded



<b>onmousedown</b>	All visible elements.	Script to be run when a mouse button is pressed down on an element
<b>onmousemove</b>	All visible elements.	Script to be run as long as the mouse pointer is moving over an element
<b>onmouseout</b>	All visible elements.	Script to be run when a mouse pointer moves out of an element
<b>onmouseover</b>	All visible elements.	Script to be run when a mouse pointer moves over an element
<b>onmouseup</b>	All visible elements.	Script to be run when a mouse button is released over an element
<b>onmousewheel</b>	All visible elements.	Script to be run when a mouse wheel is being scrolled over an element
<b>onoffline</b>	<body>	Script to be run when the browser starts to work offline
<b>ononline</b>	<body>	Script to be run when the browser starts to work online
<b>onpagehide</b>	<body>	Script to be run when a user navigates away from a page
<b>onpageshow</b>	<body>	Script to be run when a user navigates to a page
<b>onpaste</b>	All visible elements.	Script to be run when the user pastes some content in an element
<b>onpause</b>	<audio>, <video>	Script to be run when the media is paused either by the user or programmatically
<b>onplay</b>	<audio>, <video>	Script to be run when the media has started playing
<b>onplaying</b>	<audio>, <video>	Script to be run when the media has started playing
<b>onpopstate</b>	<body>	Script to be run when the window's history changes.
<b>onprogress</b>	<audio>, <video>	Script to be run when the browser is in the process of getting the media data
<b>onratechange</b>	<audio>, <video>	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode).
<b>onreset</b>	<form>	Script to be run when a reset button in a form is clicked.
<b>onresize</b>	<body>	Script to be run when the browser window is being resized.
<b>onscroll</b>	All visible elements.	Script to be run when an element's scrollbar is being scrolled
<b>onsearch</b>	<input>	Script to be run when the user writes something in a search field (for <input="search">)
<b>onseeked</b>	<audio>, <video>	Script to be run when the seeking attribute is set to false indicating that seeking has ended



<b>onseeking</b>	<audio>, <video>	Script to be run when the seeking attribute is set to true indicating that seeking is active
<b>onselect</b>	All visible elements.	Script to be run when the element gets selected
<b>onstalled</b>	<audio>, <video>	Script to be run when the browser is unable to fetch the media data for whatever reason
<b>onstorage</b>	<body>	Script to be run when a Web Storage area is updated
<b>onsubmit</b>	<form>	Script to be run when a form is submitted
<b>onsuspend</b>	<audio>, <video>	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
<b>ontimeupdate</b>	<audio>, <video>	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
<b>ontoggle</b>	<details>	Script to be run when the user opens or closes the <details> element
<b>onunload</b>	<body>	Script to be run when a page has unloaded (or the browser window has been closed)
<b>onvolumechange</b>	<audio>, <video>	Script to be run each time the volume of a video/audio has been changed
<b>onwaiting</b>	<audio>, <video>	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)
<b>onwheel</b>	All visible elements.	Script to be run when the mouse wheel rolls up or down over an element
<b>open</b>	<details>	Specifies that the details should be visible (open) to the user
<b>optimum</b>	<meter>	Specifies what value is the optimal value for the gauge
<b>pattern</b>	<input>	Specifies a regular expression that an <input> element's value is checked against
<b>placeholder</b>	<input>, <textarea>	Specifies a short hint that describes the expected value of the element
<b>poster</b>	<video>	Specifies an image to be shown while the video is downloading, or until the user hits the play button
<b>preload</b>	<audio>, <video>	Specifies if and how the author thinks the audio/video should be loaded when the page loads
<b>readonly</b>	<input>, <textarea>	Specifies that the element is read-only
<b>rel</b>	<a>, <area>, <form>, <link>	Specifies the relationship between the current document and the linked document
<b>required</b>	<input>, <select>, <textarea>	Specifies that the element must be filled out before submitting the form
<b>reversed</b>	<ol>	Specifies that the list order should be descending (9,8,7...)



<b>rows</b>	<textarea>	Specifies the visible number of lines in a text area
<b>rowspan</b>	<td>, <th>	Specifies the number of rows a table cell should span
<b>sandbox</b>	<iframe>	Enables an extra set of restrictions for the content in an <iframe>
<b>scope</b>	<th>	Specifies whether a header cell is a header for a column, row, or group of columns or rows
<b>selected</b>	<option>	Specifies that an option should be pre-selected when the page loads
<b>shape</b>	<area>	Specifies the shape of the area
<b>size</b>	<input>, <select>	Specifies the width, in characters (for <input>) or specifies the number of visible options (for <select>)
<b>sizes</b>	<img>, <link>, <source>	Specifies the size of the linked resource
<b>span</b>	<col>, <colgroup>	Specifies the number of columns to span
<b>spellcheck</b>	Global Attributes	Specifies whether the element is to have its spelling and grammar checked or not
<b>src</b>	<audio>, <embed>, <iframe>, <img>, <input>, <script>, <source>, <track>, <video>	Specifies the URL of the media file
<b>srcdoc</b>	<iframe>	Specifies the HTML content of the page to show in the <iframe>
<b>srlang</b>	<track>	Specifies the language of the track text data (required if kind="subtitles")
<b>srcset</b>	<img>, <source>	Specifies the URL of the image to use in different situations
<b>start</b>	<ol>	Specifies the start value of an ordered list
<b>step</b>	<input>	Specifies the legal number intervals for an input field
<b>style</b>	Global Attributes	Specifies an inline CSS style for an element
<b>tabindex</b>	Global Attributes	Specifies the tabbing order of an element
<b>target</b>	<a>, <area>, <base>, <form>	Specifies the target for where to open the linked document or where to submit the form
<b>title</b>	Global Attributes	Specifies extra information about an element
<b>translate</b>	Global Attributes	Specifies whether the content of an element should be translated or not
<b>type</b>	<a>, <button>, <embed>, <input>, <link>, <menu>, <object>, <script>, <source>, <style>	Specifies the type of element

<b>usemap</b>	<img>, <object>	Specifies an image as a client-side image map
<b>value</b>	<button>, <input> , <li>, <option>, <meter>, <progress>, <param>	Specifies the value of the element
<b>width</b>	<canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video>	Specifies the width of the element
<b>wrap</b>	<textarea>	Specifies how the text in a text area is to be wrapped when submitted in a form



## 77 HTML Global Attributes

### What are HTML Global attributes?

Global attributes are **attributes common to all HTML elements**; they can be used on all elements, though they may have no effect on some elements. Global attributes may be specified on all HTML elements, even those not specified in the standard.

### HTML Global Attributes

The global attributes are attributes that can be used with all HTML elements.

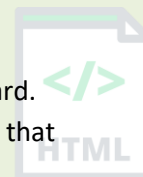
Attribute	Description
<b>accesskey</b>	Specifies a shortcut key to activate/focus an element
<b>class</b>	Specifies one or more classnames for an element (refers to a class in a style sheet)
<b>contenteditable</b>	Specifies whether the content of an element is editable or not
<b>data-*</b>	Used to store custom data private to the page or application
<b>dir</b>	Specifies the text direction for the content in an element
<b>draggable</b>	Specifies whether an element is draggable or not
<b>hidden</b>	Specifies that an element is not yet, or is no longer, relevant
<b>id</b>	Specifies a unique id for an element
<b>lang</b>	Specifies the language of the element's content
<b>spellcheck</b>	Specifies whether the element is to have its spelling and grammar checked or not
<b>style</b>	Specifies an inline CSS style for an element
<b>tabindex</b>	Specifies the tabbing order of an element
<b>title</b>	Specifies extra information about an element
<b>translate</b>	Specifies whether the content of an element should be translated or not



## 78 HTML Event Attributes

### What are HTML event attributes?

The event attributes are **applied to input controls**. Runs the script when the user presses a key on the keyboard. Runs the script when the user releases the currently pressed key. Runs the script when the user presses a key that displays a character.



### Global Event Attributes

HTML has the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

To learn more about programming events, please visit our [JavaScript tutorial](#).

Below are the global event attributes that can be added to HTML elements to define event actions.

### 78.1 Window Event Attributes

Events triggered for the window object (applies to the <body> tag):

Attribute	Value	Description
<b>onafterprint</b>	<i>script</i>	Script to be run after the document is printed
<b>onbeforeprint</b>	<i>script</i>	Script to be run before the document is printed
<b>onbeforeunload</b>	<i>script</i>	Script to be run when the document is about to be unloaded
<b>onerror</b>	<i>script</i>	Script to be run when an error occurs
<b>onhashchange</b>	<i>script</i>	Script to be run when there has been changes to the anchor part of the a URL
<b>onload</b>	<i>script</i>	Fires after the page is finished loading
<b>onmessage</b>	<i>script</i>	Script to be run when the message is triggered
<b>onoffline</b>	<i>script</i>	Script to be run when the browser starts to work offline
<b>ononline</b>	<i>script</i>	Script to be run when the browser starts to work online
<b>onpagehide</b>	<i>script</i>	Script to be run when a user navigates away from a page
<b>onpageshow</b>	<i>script</i>	Script to be run when a user navigates to a page
<b>onpopstate</b>	<i>script</i>	Script to be run when the window's history changes
<b>onresize</b>	<i>script</i>	Fires when the browser window is resized
<b>onstorage</b>	<i>script</i>	Script to be run when a Web Storage area is updated
<b>onunload</b>	<i>script</i>	Fires once a page has unloaded (or the browser window has been closed)



## 78.2 Form Events

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):



Attribute	Value	Description
onblur	script	Fires the moment that the element loses focus
onchange	script	Fires the moment when the value of the element is changed
oncontextmenu	script	Script to be run when a context menu is triggered
onfocus	script	Fires the moment when the element gets focus
oninput	script	Script to be run when an element gets user input
oninvalid	script	Script to be run when an element is invalid
onreset	script	Fires when the Reset button in a form is clicked
onsearch	script	Fires when the user writes something in a search field (for <code>&lt;input="search"&gt;</code> )
onselect	script	Fires after some text has been selected in an element
onsubmit	script	Fires when a form is submitted

## 78.3 Keyboard Events

Attribute	Value	Description
<b>onkeydown</b>	<i>script</i>	Fires when a user is pressing a key
<b>onkeypress</b>	<i>script</i>	Fires when a user presses a key
<b>onkeyup</b>	<i>script</i>	Fires when a user releases a key

## 78.4 Mouse Events

Attribute	Value	Description
<b>onclick</b>	<i>script</i>	Fires on a mouse click on the element

<b>ondblclick</b>	<i>script</i>	Fires on a mouse double-click on the element
<b>onmousedown</b>	<i>script</i>	Fires when a mouse button is pressed down on an element
<b>onmousemove</b>	<i>script</i>	Fires when the mouse pointer is moving while it is over an element
<b>onmouseout</b>	<i>script</i>	Fires when the mouse pointer moves out of an element
<b>onmouseover</b>	<i>script</i>	Fires when the mouse pointer moves over an element
<b>onmouseup</b>	<i>script</i>	Fires when a mouse button is released over an element
<b>onmousewheel</b>	<i>script</i>	<b>Deprecated.</b> Use the <b>onwheel</b> attribute instead
<b>onwheel</b>	<i>script</i>	Fires when the mouse wheel rolls up or down over an element



## 78.5 Drag Events

Attribute	Value	Description
<b>ondrag</b>	<i>script</i>	Script to be run when an element is dragged
<b>ondragend</b>	<i>script</i>	Script to be run at the end of a drag operation
<b>ondragenter</b>	<i>script</i>	Script to be run when an element has been dragged to a valid drop target
<b>ondragleave</b>	<i>script</i>	Script to be run when an element leaves a valid drop target
<b>ondragover</b>	<i>script</i>	Script to be run when an element is being dragged over a valid drop target
<b>ondragstart</b>	<i>script</i>	Script to be run at the start of a drag operation
<b>ondrop</b>	<i>script</i>	Script to be run when dragged element is being dropped
<b>onscroll</b>	<i>script</i>	Script to be run when an element's scrollbar is being scrolled

## 78.6 Clipboard Events

Attribute	Value	Description
<b>oncopy</b>	<i>script</i>	Fires when the user copies the content of an element
<b>oncut</b>	<i>script</i>	Fires when the user cuts the content of an element
<b>onpaste</b>	<i>script</i>	Fires when the user pastes some content in an element

## 78.7 Media Events

Events triggered by medias like videos, images and audio (applies to all HTML elements, but is most common in media elements, like `<audio>`, `<embed>`, `<img>`, `<object>`, and `<video>`).

**Tip:** Look at our [HTML Audio and Video DOM Reference](#) for more information.

Attribute	Value	Description
<b>onabort</b>	<i>script</i>	Script to be run on abort
<b>oncanplay</b>	<i>script</i>	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
<b>oncanplaythrough</b>	<i>script</i>	Script to be run when a file can be played all the way to the end without pausing for buffering
<b>oncuechange</b>	<i>script</i>	Script to be run when the cue changes in a <track> element
<b>ondurationchange</b>	<i>script</i>	Script to be run when the length of the media changes
<b>onemptied</b>	<i>script</i>	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
<b>onended</b>	<i>script</i>	Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")
<b>onerror</b>	<i>script</i>	Script to be run when an error occurs when the file is being loaded
<b>onloadeddata</b>	<i>script</i>	Script to be run when media data is loaded
<b>onloadedmetadata</b>	<i>script</i>	Script to be run when meta data (like dimensions and duration) are loaded
<b>onloadstart</b>	<i>script</i>	Script to be run just as the file begins to load before anything is actually loaded
<b>onpause</b>	<i>script</i>	Script to be run when the media is paused either by the user or programmatically
<b>onplay</b>	<i>script</i>	Script to be run when the media is ready to start playing
<b>onplaying</b>	<i>script</i>	Script to be run when the media actually has started playing
<b>onprogress</b>	<i>script</i>	Script to be run when the browser is in the process of getting the media data
<b>onratechange</b>	<i>script</i>	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode)
<b>onseeked</b>	<i>script</i>	Script to be run when the seeking attribute is set to false indicating that seeking has ended
<b>onseeking</b>	<i>script</i>	Script to be run when the seeking attribute is set to true indicating that seeking is active
<b>onstalled</b>	<i>script</i>	Script to be run when the browser is unable to fetch the media data for whatever reason
<b>onsuspend</b>	<i>script</i>	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
<b>ontimeupdate</b>	<i>script</i>	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
<b>onvolumechange</b>	<i>script</i>	Script to be run each time the volume is changed which (includes setting the volume to "mute")
<b>onwaiting</b>	<i>script</i>	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)



## 78.8 Misc Events

Attribute	Value	Description
<b>ontoggle</b>	<i>script</i>	Fires when the user opens or closes the <details> element



## 79 HTML Canvas Reference

### 79.1 HTML Canvas Reference

The HTML `<canvas>` tag is used to draw graphics, on the fly, via scripting (usually JavaScript).

To learn more about `<canvas>`, please read our [HTML Canvas tutorial](#).



### 79.2 Colors, Styles, and Shadows

Property	Description
<code>fillStyle</code>	Sets or returns the color, gradient, or pattern used to fill the drawing
<code>strokeStyle</code>	Sets or returns the color, gradient, or pattern used for strokes
<code>shadowColor</code>	Sets or returns the color to use for shadows
<code>shadowBlur</code>	Sets or returns the blur level for shadows
<code>shadowOffsetX</code>	Sets or returns the horizontal distance of the shadow from the shape
<code>shadowOffsetY</code>	Sets or returns the vertical distance of the shadow from the shape
Method	Description
<code>createLinearGradient()</code>	Creates a linear gradient (to use on canvas content)
<code>createPattern()</code>	Repeats a specified element in the specified direction
<code>createRadialGradient()</code>	Creates a radial/circular gradient (to use on canvas content)
<code>addColorStop()</code>	Specifies the colors and stop positions in a gradient object

### 79.3 Line Styles

Property	Description
<code>lineCap</code>	Sets or returns the style of the end caps for a line
<code>lineJoin</code>	Sets or returns the type of corner created, when two lines meet
<code>lineWidth</code>	Sets or returns the current line width
<code>miterLimit</code>	Sets or returns the maximum miter length

## 79.4 Rectangles

Method	Description
<code>rect()</code>	Creates a rectangle
<code>fillRect()</code>	Draws a "filled" rectangle
<code>strokeRect()</code>	Draws a rectangle (no fill)
<code>clearRect()</code>	Clears the specified pixels within a given rectangle



## 79.5 Paths

Method	Description
<code>fill()</code>	Fills the current drawing (path)
<code>stroke()</code>	Actually draws the path you have defined
<code>beginPath()</code>	Begins a path, or resets the current path
<code>moveTo()</code>	Moves the path to the specified point in the canvas, without creating a line
<code>closePath()</code>	Creates a path from the current point back to the starting point
<code>lineTo()</code>	Adds a new point and creates a line to that point from the last specified point in the canvas
<code>clip()</code>	Clips a region of any shape and size from the original canvas
<code>quadraticCurveTo()</code>	Creates a quadratic Bézier curve
<code>bezierCurveTo()</code>	Creates a cubic Bézier curve
<code>arc()</code>	Creates an arc/curve (used to create circles, or parts of circles)
<code>arcTo()</code>	Creates an arc/curve between two tangents
<code>isPointInPath()</code>	Returns true if the specified point is in the current path, otherwise false

## 79.6 Transformations

Method	Description
<code>scale()</code>	Scales the current drawing bigger or smaller
<code>rotate()</code>	Rotates the current drawing
<code>translate()</code>	Remaps the (0,0) position on the canvas
<code>transform()</code>	Replaces the current transformation matrix for the drawing
<code>setTransform()</code>	Resets the current transform to the identity matrix. Then runs <code>transform()</code>

## 79.7 Text

Property	Description
<code>font</code>	Sets or returns the current font properties for text content

<b>textAlign</b>	Sets or returns the current alignment for text content
<b>textBaseline</b>	Sets or returns the current text baseline used when drawing text
Method	Description
<b>fillText()</b>	Draws "filled" text on the canvas
<b>strokeText()</b>	Draws text on the canvas (no fill)
<b>measureText()</b>	Returns an object that contains the width of the specified text



## 79.8 Image Drawing

Method	Description
<b>drawImage()</b>	Draws an image, canvas, or video onto the canvas

## 79.9 Pixel Manipulation

Property	Description
<b>width</b>	Returns the width of an ImageData object
<b>height</b>	Returns the height of an ImageData object
<b>data</b>	Returns an object that contains image data of a specified ImageData object
Method	Description
<b>createImageData()</b>	Creates a new, blank ImageData object
<b>getImageData()</b>	Returns an ImageData object that copies the pixel data for the specified rectangle on a canvas
<b>putImageData()</b>	Puts the image data (from a specified ImageData object) back onto the canvas

## 79.10 Compositing

Property	Description
<b>globalAlpha</b>	Sets or returns the current alpha or transparency value of the drawing
<b>globalCompositeOperation</b>	Sets or returns how a new image is drawn onto an existing image

## 79.11 Other

Method	Description
<b>save()</b>	Saves the state of the current context
<b>restore()</b>	Returns previously saved path state and attributes
<b>createEvent()</b>	
<b>getContext()</b>	

toDataURL()





## 80 HTML Audio/Video DOM Reference

### 80.1 HTML Audio and Video DOM Reference

The HTML5 DOM has methods, properties, and events for the `<audio>` and `<video>` elements.



### 80.2 HTML Audio/Video Methods

Method	Description
<code>addTextTrack()</code>	Adds a new text track to the audio/video
<code>canPlayType()</code>	Checks if the browser can play the specified audio/video type
<code>load()</code>	Re-loads the audio/video element
<code>play()</code>	Starts playing the audio/video
<code>pause()</code>	Pauses the currently playing audio/video

### 80.3 HTML Audio/Video Properties

Property	Description
<code>audioTracks</code>	Returns an <code>AudioTrackList</code> object representing available audio tracks
<code>autoplay</code>	Sets or returns whether the audio/video should start playing as soon as it is loaded
<code>buffered</code>	Returns a <code>TimeRanges</code> object representing the buffered parts of the audio/video
<code>controller</code>	Returns the <code>MediaController</code> object representing the current media controller of the audio/video
<code>controls</code>	Sets or returns whether the audio/video should display controls (like play/pause etc.)
<code>crossOrigin</code>	Sets or returns the CORS settings of the audio/video
<code>currentSrc</code>	Returns the URL of the current audio/video
<code>currentTime</code>	Sets or returns the current playback position in the audio/video (in seconds)
<code>defaultMuted</code>	Sets or returns whether the audio/video should be muted by default
<code>defaultPlaybackRate</code>	Sets or returns the default speed of the audio/video playback
<code>duration</code>	Returns the length of the current audio/video (in seconds)
<code>ended</code>	Returns whether the playback of the audio/video has ended or not
<code>error</code>	Returns a <code>MediaError</code> object representing the error state of the audio/video
<code>loop</code>	Sets or returns whether the audio/video should start over again when finished

<b>mediaGroup</b>	Sets or returns the group the audio/video belongs to (used to link multiple audio/video elements)
<b>muted</b>	Sets or returns whether the audio/video is muted or not
<b>networkState</b>	Returns the current network state of the audio/video
<b>paused</b>	Returns whether the audio/video is paused or not
<b>playbackRate</b>	Sets or returns the speed of the audio/video playback
<b>played</b>	Returns a TimeRanges object representing the played parts of the audio/video
<b>preload</b>	Sets or returns whether the audio/video should be loaded when the page loads
<b>readyState</b>	Returns the current ready state of the audio/video
<b>seekable</b>	Returns a TimeRanges object representing the seekable parts of the audio/video
<b>seeking</b>	Returns whether the user is currently seeking in the audio/video
<b>src</b>	Sets or returns the current source of the audio/video element
<b>startDate</b>	Returns a Date object representing the current time offset
<b>textTracks</b>	Returns a TextTrackList object representing the available text tracks
<b>videoTracks</b>	Returns a VideoTrackList object representing the available video tracks
<b>volume</b>	Sets or returns the volume of the audio/video



## 80.4 HTML Audio/Video Events

Event	Description
<b>abort</b>	Fires when the loading of an audio/video is aborted
<b>canplay</b>	Fires when the browser can start playing the audio/video
<b>canplaythrough</b>	Fires when the browser can play through the audio/video without stopping for buffering
<b>durationchange</b>	Fires when the duration of the audio/video is changed
<b>emptied</b>	Fires when the current playlist is empty
<b>ended</b>	Fires when the current playlist is ended
<b>error</b>	Fires when an error occurred during the loading of an audio/video
<b>loadeddata</b>	Fires when the browser has loaded the current frame of the audio/video
<b>loadedmetadata</b>	Fires when the browser has loaded meta data for the audio/video
<b>loadstart</b>	Fires when the browser starts looking for the audio/video
<b>pause</b>	Fires when the audio/video has been paused

<b>play</b>	Fires when the audio/video has been started or is no longer paused
<b>playing</b>	Fires when the audio/video is playing after having been paused or stopped for buffering
<b>progress</b>	Fires when the browser is downloading the audio/video
<b>ratechange</b>	Fires when the playing speed of the audio/video is changed
<b>seeked</b>	Fires when the user is finished moving/skipping to a new position in the audio/video
<b>seeking</b>	Fires when the user starts moving/skipping to a new position in the audio/video
<b>stalled</b>	Fires when the browser is trying to get media data, but data is not available
<b>suspend</b>	Fires when the browser is intentionally not getting media data
<b>timeupdate</b>	Fires when the current playback position has changed
<b>volumechange</b>	Fires when the volume has been changed
<b>waiting</b>	Fires when the video stops because it needs to buffer the frame



# 81 HTML Character Sets

## 81.1 Common HTML Character Sets

The default character set in HTML5 is UTF-8.

For a closer look, visit our [Complete HTML Character Set Reference](#).



Number	ASCII	ANSI	8859-1	UTF-8	Description
32					space
33	!	!	!	!	exclamation mark
34	"	"	"	"	quotation mark
35	#	#	#	#	number sign
36	\$	\$	\$	\$	dollar sign
37	%	%	%	%	percent sign
38	&	&	&	&	ampersand
39	'	'	'	'	apostrophe
40	(	(	(	(	left parenthesis
41	)	)	)	)	right parenthesis
42	*	*	*	*	asterisk
43	+	+	+	+	plus sign
44	,	,	,	,	comma
45	-	-	-	-	hyphen-minus
46	.	.	.	.	full stop
47	/	/	/	/	solidus
48	0	0	0	0	digit zero
49	1	1	1	1	digit one
50	2	2	2	2	digit two
51	3	3	3	3	digit three
52	4	4	4	4	digit four
53	5	5	5	5	digit five
54	6	6	6	6	digit six
55	7	7	7	7	digit seven
56	8	8	8	8	digit eight
57	9	9	9	9	digit nine
58	:	:	:	:	colon
59	;	;	;	;	semicolon

60	<	<	<	<	less-than sign
61	=	=	=	=	equals sign
62	>	>	>	>	greater-than sign
63	?	?	?	?	question mark
64	@	@	@	@	commercial at
65	A	A	A	A	Latin capital letter A
66	B	B	B	B	Latin capital letter B
67	C	C	C	C	Latin capital letter C
68	D	D	D	D	Latin capital letter D
69	E	E	E	E	Latin capital letter E
70	F	F	F	F	Latin capital letter F
71	G	G	G	G	Latin capital letter G
72	H	H	H	H	Latin capital letter H
73	I	I	I	I	Latin capital letter I
74	J	J	J	J	Latin capital letter J
75	K	K	K	K	Latin capital letter K
76	L	L	L	L	Latin capital letter L
77	M	M	M	M	Latin capital letter M
78	N	N	N	N	Latin capital letter N
79	O	O	O	O	Latin capital letter O
80	P	P	P	P	Latin capital letter P
81	Q	Q	Q	Q	Latin capital letter Q
82	R	R	R	R	Latin capital letter R
83	S	S	S	S	Latin capital letter S
84	T	T	T	T	Latin capital letter T
85	U	U	U	U	Latin capital letter U
86	V	V	V	V	Latin capital letter V
87	W	W	W	W	Latin capital letter W
88	X	X	X	X	Latin capital letter X
89	Y	Y	Y	Y	Latin capital letter Y
90	Z	Z	Z	Z	Latin capital letter Z
91	[	[	[	[	left square bracket
92	\	\	\	\	reverse solidus
93	]	]	]	]	right square bracket



94	^	^	^	^	circumflex accent
95	–	–	–	–	low line
96	`	`	`	`	grave accent
97	a	a	a	a	Latin small letter a
98	b	b	b	b	Latin small letter b
99	c	c	c	c	Latin small letter c
100	d	d	d	d	Latin small letter d
101	e	e	e	e	Latin small letter e
102	f	f	f	f	Latin small letter f
103	g	g	g	g	Latin small letter g
104	h	h	h	h	Latin small letter h
105	i	i	i	i	Latin small letter i
106	j	j	j	j	Latin small letter j
107	k	k	k	k	Latin small letter k
108	l	l	l	l	Latin small letter l
109	m	m	m	m	Latin small letter m
110	n	n	n	n	Latin small letter n
111	o	o	o	o	Latin small letter o
112	p	p	p	p	Latin small letter p
113	q	q	q	q	Latin small letter q
114	r	r	r	r	Latin small letter r
115	s	s	s	s	Latin small letter s
116	t	t	t	t	Latin small letter t
117	u	u	u	u	Latin small letter u
118	v	v	v	v	Latin small letter v
119	w	w	w	w	Latin small letter w
120	x	x	x	x	Latin small letter x
121	y	y	y	y	Latin small letter y
122	z	z	z	z	Latin small letter z
123	{	{	{	{	left curly bracket
124					vertical line
125	}	}	}	}	right curly bracket
126	~	~	~	~	tilde
127	DEL				



128		€			euro sign
129		•	•	•	NOT USED
130		,			single low-9 quotation mark
131		f			Latin small letter f with hook
132		”			double low-9 quotation mark
133		…			horizontal ellipsis
134		†			dagger
135		‡			double dagger
136		^			modifier letter circumflex accent
137		‰			per mille sign
138		Š			Latin capital letter S with caron
139		‹			single left-pointing angle quotation mark
140		Œ			Latin capital ligature OE
141		•	•	•	NOT USED
142		Ž			Latin capital letter Z with caron
143		•	•	•	NOT USED
144		•	•	•	NOT USED
145		‘			left single quotation mark
146		’			right single quotation mark
147		“			left double quotation mark
148		”			right double quotation mark
149		•			bullet
150		–			en dash
151		—			em dash
152		~			small tilde
153		™			trade mark sign
154		š			Latin small letter s with caron
155		›			single right-pointing angle quotation mark
156		œ			Latin small ligature oe
157		•	•	•	NOT USED
158		ž			Latin small letter z with caron
159		ÿ			Latin capital letter Y with diaeresis
160					no-break space
161		¡	¡	¡	inverted exclamation mark



162	¢	¢	¢	cent sign
163	£	£	£	pound sign
164	¤	¤	¤	currency sign
165	¥	¥	¥	yen sign
166	‡	‡	‡	broken bar
167	§	§	§	section sign
168	¨	¨	¨	diaeresis
169	©	©	©	copyright sign
170	ª	ª	ª	feminine ordinal indicator
171	«	«	«	left-pointing double angle quotation mark
172	¬	¬	¬	not sign
173	–	–	–	soft hyphen
174	®	®	®	registered sign
175	ˉ	ˉ	ˉ	macron
176	°	°	°	degree sign
177	±	±	±	plus-minus sign
178	²	²	²	superscript two
179	³	³	³	superscript three
180	´	´	´	acute accent
181	μ	μ	μ	micro sign
182	¶	¶	¶	pilcrow sign
183	·	·	·	middle dot
184	¸	¸	¸	cedilla
185	¹	¹	¹	superscript one
186	º	º	º	masculine ordinal indicator
187	»	»	»	right-pointing double angle quotation mark
188	¼	¼	¼	vulgar fraction one quarter
189	½	½	½	vulgar fraction one half
190	¾	¾	¾	vulgar fraction three quarters
191	¿	¿	¿	inverted question mark
192	À	À	À	Latin capital letter A with grave
193	Á	Á	Á	Latin capital letter A with acute
194	Â	Â	Â	Latin capital letter A with circumflex
195	Ã	Ã	Ã	Latin capital letter A with tilde





196	Ä	Ä	Ä	Latin capital letter A with diaeresis
197	Å	Å	Å	Latin capital letter A with ring above
198	Æ	Æ	Æ	Latin capital letter AE
199	Ç	Ç	Ç	Latin capital letter C with cedilla
200	È	È	È	Latin capital letter E with grave
201	É	É	É	Latin capital letter E with acute
202	Ê	Ê	Ê	Latin capital letter E with circumflex
203	Ë	Ë	Ë	Latin capital letter E with diaeresis
204	Ì	Ì	Ì	Latin capital letter I with grave
205	Í	Í	Í	Latin capital letter I with acute
206	Î	Î	Î	Latin capital letter I with circumflex
207	Ï	Ï	Ï	Latin capital letter I with diaeresis
208	Ð	Ð	Ð	Latin capital letter Eth
209	Ñ	Ñ	Ñ	Latin capital letter N with tilde
210	Ò	Ò	Ò	Latin capital letter O with grave
211	Ó	Ó	Ó	Latin capital letter O with acute
212	Ô	Ô	Ô	Latin capital letter O with circumflex
213	Õ	Õ	Õ	Latin capital letter O with tilde
214	Ö	Ö	Ö	Latin capital letter O with diaeresis
215	×	×	×	multiplication sign
216	Ø	Ø	Ø	Latin capital letter O with stroke
217	Ù	Ù	Ù	Latin capital letter U with grave
218	Ú	Ú	Ú	Latin capital letter U with acute
219	Û	Û	Û	Latin capital letter U with circumflex
220	Ü	Ü	Ü	Latin capital letter U with diaeresis
221	Ý	Ý	Ý	Latin capital letter Y with acute
222	Þ	Þ	Þ	Latin capital letter Thorn
223	ß	ß	ß	Latin small letter sharp s
224	à	à	à	Latin small letter a with grave
225	á	á	á	Latin small letter a with acute
226	â	â	â	Latin small letter a with circumflex
227	ã	ã	ã	Latin small letter a with tilde
228	ä	ä	ä	Latin small letter a with diaeresis
229	å	å	å	Latin small letter a with ring above



230	æ	æ	æ	Latin small letter ae
231	ç	ç	ç	Latin small letter c with cedilla
232	è	è	è	Latin small letter e with grave
233	é	é	é	Latin small letter e with acute
234	ê	ê	ê	Latin small letter e with circumflex
235	ë	ë	ë	Latin small letter e with diaeresis
236	ì	ì	ì	Latin small letter i with grave
237	í	í	í	Latin small letter i with acute
238	î	î	î	Latin small letter i with circumflex
239	ï	ï	ï	Latin small letter i with diaeresis
240	ð	ð	ð	Latin small letter eth
241	ñ	ñ	ñ	Latin small letter n with tilde
242	ò	ò	ò	Latin small letter o with grave
243	ó	ó	ó	Latin small letter o with acute
244	ô	ô	ô	Latin small letter o with circumflex
245	õ	õ	õ	Latin small letter o with tilde
246	ö	ö	ö	Latin small letter o with diaeresis
247	÷	÷	÷	division sign
248	ø	ø	ø	Latin small letter o with stroke
249	ù	ù	ù	Latin small letter u with grave
250	ú	ú	ú	Latin small letter u with acute
251	û	û	û	Latin small letter u with circumflex
252	ü	ü	ü	Latin small letter u with diaeresis
253	ý	ý	ý	Latin small letter y with acute
254	þ	þ	þ	Latin small letter thorn
255	ÿ	ÿ	ÿ	Latin small letter y with diaeresis



## 82 HTML <!DOCTYPE>

### 82.1 The HTML Document Type



All HTML documents must start with a `<!DOCTYPE>` declaration.

The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

In HTML5, the `<!DOCTYPE>` declaration is simple:

```
<!DOCTYPE html>
```

In older documents (HTML 4 or XHTML), the declaration is more complicated because the declaration must refer to a DTD (Document Type Definition).

```
<!DOCTYPE HTML PUBLIC "-//C//DTD HTML 4.01 Transitional//EN"
"http://www..org/TR/html4/loose.dtd">
```

```
<!DOCTYPE html PUBLIC "-//C//DTD XHTML 1.1//EN"
"http://www..org/TR/xhtml11/DTD/xhtml11.dtd">
```

You can read more about document types in the `<!DOCTYPE>` reference.

### 82.2 Valid HTML Elements in Different DOCTYPES

Tag	HTML 5	HTML 4	XHTML
<code>&lt;a&gt;</code>	Yes	Yes	Yes
<code>&lt;abbr&gt;</code>	Yes	Yes	Yes
<code>&lt;acronym&gt;</code>	No	Yes	Yes
<code>&lt;address&gt;</code>	Yes	Yes	Yes
<code>&lt;applet&gt;</code>	No	Yes	No
<code>&lt;area&gt;</code>	Yes	Yes	No
<code>&lt;article&gt;</code>	Yes	No	No
<code>&lt;aside&gt;</code>	Yes	No	No
<code>&lt;audio&gt;</code>	Yes	No	No
<code>&lt;b&gt;</code>	Yes	Yes	Yes
<code>&lt;base&gt;</code>	Yes	Yes	Yes
<code>&lt;basefont&gt;</code>	No	Yes	No
<code>&lt;bdi&gt;</code>	Yes	No	No
<code>&lt;bdo&gt;</code>	Yes	Yes	No
<code>&lt;big&gt;</code>	No	Yes	Yes
<code>&lt;blockquote&gt;</code>	Yes	Yes	Yes
<code>&lt;body&gt;</code>	Yes	Yes	Yes

 	Yes	Yes	Yes
<button>	Yes	Yes	Yes
<canvas>	Yes	No	No
<caption>	Yes	Yes	Yes
<center>	No	Yes	No
<cite>	Yes	Yes	Yes
<code>	Yes	Yes	Yes
<col>	Yes	Yes	No
<colgroup>	Yes	Yes	No
<data>	Yes	No	No
<datalist>	Yes	No	No
<dd>	Yes	Yes	Yes
<del>	Yes	Yes	No
<details>	Yes	No	No
<dfn>	Yes	Yes	Yes
<dialog>	Yes	No	No
<dir>	No	Yes	No
<div>	Yes	Yes	Yes
<dl>	Yes	Yes	Yes
<dt>	Yes	Yes	Yes
<em>	Yes	Yes	Yes
<embed>	Yes	No	No
<fieldset>	Yes	Yes	Yes
<figcaption>	Yes	No	No
<figure>	Yes	No	No
<font>	No	Yes	No
<footer>	Yes	No	No
<form>	Yes	Yes	Yes
<frame>	No	No	No
<frameset>	No	Yes	No
<h1> to <h6>	Yes	Yes	Yes
<head>	Yes	Yes	Yes
<header>	Yes	No	No
<hr>	Yes	Yes	Yes



<html>	Yes	Yes	Yes
<i>	Yes	Yes	Yes
<iframe>	Yes	Yes	No
<img>	Yes	Yes	Yes
<input>	Yes	Yes	Yes
<ins>	Yes	Yes	No
<kbd>	Yes	Yes	Yes
<label>	Yes	Yes	Yes
<legend>	Yes	Yes	Yes
<li>	Yes	Yes	Yes
<link>	Yes	Yes	Yes
<main>	Yes	No	No
<map>	Yes	Yes	No
<mark>	Yes	No	No
<meta>	Yes	Yes	Yes
<meter>	Yes	No	No
<nav>	Yes	No	No
<noframes>	No	Yes	No
<noscript>	Yes	Yes	Yes
<object>	Yes	Yes	Yes
<ol>	Yes	Yes	Yes
<optgroup>	Yes	Yes	Yes
<option>	Yes	Yes	Yes
<output>	Yes	No	No
<p>	Yes	Yes	Yes
<param>	Yes	Yes	Yes
<picture>	Yes	No	No
<pre>	Yes	Yes	Yes
<progress>	Yes	No	No
<q>	Yes	Yes	Yes
<rp>	Yes	No	No
<rt>	Yes	No	No
<ruby>	Yes	No	No



<s>	Yes	Yes	No
<samp>	Yes	Yes	Yes
<script>	Yes	Yes	Yes
<section>	Yes	No	No
<select>	Yes	Yes	Yes
<small>	Yes	Yes	Yes
<source>	Yes	No	No
<span>	Yes	Yes	Yes
<strike>	No	Yes	No
<strong>	Yes	Yes	Yes
<style>	Yes	Yes	Yes
<sub>	Yes	Yes	Yes
<summary>	Yes	No	No
<sup>	Yes	Yes	Yes
<table>	Yes	Yes	Yes
<tbody>	Yes	Yes	No
<td>	Yes	Yes	Yes
<template>	Yes	No	No
<textarea>	Yes	Yes	Yes
<tfoot>	Yes	Yes	No
<th>	Yes	Yes	Yes
<thead>	Yes	Yes	No
<time>	Yes	No	No
<title>	Yes	Yes	Yes
<tr>	Yes	Yes	Yes
<track>	Yes	No	No
<tt>	No	Yes	Yes
<u>	Yes	Yes	No
<ul>	Yes	Yes	Yes
<var>	Yes	Yes	Yes
<video>	Yes	No	No
<wbr>	Yes	No	No



## 83 HTML URL Encoding Reference

### 83.1 URL - Uniform Resource Locator

Web browsers request pages from web servers by using a URL.

The URL is the address of a web page, like: <https://www.bintr.com>



### 83.2 URL Encoding (Percent Encoding)

URL encoding converts characters into a format that can be transmitted over the Internet.

URLs can only be sent over the Internet using the [ASCII character-set](#).

Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format.

URL encoding replaces unsafe ASCII characters with a "%" followed by two hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign or with %20.

### 83.3 Try It Yourself

If you click the "Submit" button below, the browser will URL encode the input before it is sent to the server. A page at the server will display the received input.

Try some other input and click Submit again.

### 83.4 URL Encoding Functions

In JavaScript, PHP, and ASP there are functions that can be used to URL encode a string.

PHP has the `rawurlencode()` function, and ASP has the `Server.URLEncode()` function.

In JavaScript you can use the `encodeURIComponent()` function.

Click the "URL Encode" button to see how the JavaScript function encodes the text.

**Note:** The JavaScript function encodes space as %20.

### 83.5 ASCII Encoding Reference

Your browser will encode input, according to the character-set used in your page.

The default character-set in HTML5 is UTF-8.

Character	From Windows-1252	From UTF-8
space	%20	%20
!	%21	%21
"	%22	%22
#	%23	%23
\$	%24	%24
%	%25	%25
&	%26	%26
'	%27	%27
(	%28	%28
)	%29	%29
*	%2A	%2A
+	%2B	%2B
,	%2C	%2C
-	%2D	%2D
.	%2E	%2E
/	%2F	%2F
0	%30	%30
1	%31	%31
2	%32	%32
3	%33	%33
4	%34	%34
5	%35	%35
6	%36	%36
7	%37	%37
8	%38	%38
9	%39	%39
:	%3A	%3A
;	%3B	%3B
<	%3C	%3C
=	%3D	%3D
>	%3E	%3E
?	%3F	%3F
@	%40	%40
A	%41	%41
B	%42	%42





C	%43	%43
D	%44	%44
E	%45	%45
F	%46	%46
G	%47	%47
H	%48	%48
I	%49	%49
J	%4A	%4A
K	%4B	%4B
L	%4C	%4C
M	%4D	%4D
N	%4E	%4E
O	%4F	%4F
P	%50	%50
Q	%51	%51
R	%52	%52
S	%53	%53
T	%54	%54
U	%55	%55
V	%56	%56
W	%57	%57
X	%58	%58
Y	%59	%59
Z	%5A	%5A
[	%5B	%5B
\	%5C	%5C
]	%5D	%5D
^	%5E	%5E
_	%5F	%5F
`	%60	%60
a	%61	%61
b	%62	%62
c	%63	%63
d	%64	%64
e	%65	%65
f	%66	%66
g	%67	%67



h	%68	%68
i	%69	%69
j	%6A	%6A
k	%6B	%6B
l	%6C	%6C
m	%6D	%6D
n	%6E	%6E
o	%6F	%6F
p	%70	%70
q	%71	%71
r	%72	%72
s	%73	%73
t	%74	%74
u	%75	%75
v	%76	%76
w	%77	%77
x	%78	%78
y	%79	%79
z	%7A	%7A
{	%7B	%7B
	%7C	%7C
}	%7D	%7D
~	%7E	%7E
	%7F	%7F
€	%80	%E2%82%AC
•	%81	%81
,	%82	%E2%80%9A
f	%83	%C6%92
”	%84	%E2%80%9E
...	%85	%E2%80%A6
†	%86	%E2%80%A0
‡	%87	%E2%80%A1
^	%88	%CB%86
‰	%89	%E2%80%B0
Š	%8A	%C5%A0
‹	%8B	%E2%80%B9
Œ	%8C	%C5%92



•	%8D	%C5%8D
Ž	%8E	%C5%BD
•	%8F	%8F
•	%90	%C2%90
‘	%91	%E2%80%98
’	%92	%E2%80%99
“	%93	%E2%80%9C
”	%94	%E2%80%9D
•	%95	%E2%80%A2
—	%96	%E2%80%93
—	%97	%E2%80%94
~	%98	%CB%9C
™	%99	%E2%84
š	%9A	%C5%A1
›	%9B	%E2%80
œ	%9C	%C5%93
•	%9D	%9D
ž	%9E	%C5%BE
ÿ	%9F	%C5%B8
	%A0	%C2%A0
ı	%A1	%C2%A1
ć	%A2	%C2%A2
£	%A3	%C2%A3
¤	%A4	%C2%A4
¥	%A5	%C2%A5
¡	%A6	%C2%A6
§	%A7	%C2%A7
¨	%A8	%C2%A8
©	%A9	%C2%A9
ª	%AA	%C2%AA
«	%AB	%C2%AB
¬	%AC	%C2%AC
	%AD	%C2%AD
®	%AE	%C2%AE
-	%AF	%C2%AF
°	%B0	%C2%B0
±	%B1	%C2%B1



²	%B2	%C2%B2
³	%B3	%C2%B3
´	%B4	%C2%B4
µ	%B5	%C2%B5
¶	%B6	%C2%B6
·	%B7	%C2%B7
¸	%B8	%C2%B8
¹	%B9	%C2%B9
º	%BA	%C2%BA
»	%BB	%C2%BB
¼	%BC	%C2%BC
½	%BD	%C2%BD
¾	%BE	%C2%BE
¿	%BF	%C2%BF
À	%C0	%C3%80
Á	%C1	%C3%81
Â	%C2	%C3%82
Ã	%C3	%C3%83
Ä	%C4	%C3%84
Å	%C5	%C3%85
Æ	%C6	%C3%86
Ç	%C7	%C3%87
È	%C8	%C3%88
É	%C9	%C3%89
Ê	%CA	%C3%8A
Ë	%CB	%C3%8B
Ì	%CC	%C3%8C
Í	%CD	%C3%8D
Î	%CE	%C3%8E
Ï	%CF	%C3%8F
Ð	%D0	%C3%90
Ñ	%D1	%C3%91
Ò	%D2	%C3%92
Ó	%D3	%C3%93
Ô	%D4	%C3%94
Õ	%D5	%C3%95
Ö	%D6	%C3%96



×	%D7	%C3%97
∅	%D8	%C3%98
Ù	%D9	%C3%99
Ú	%DA	%C3%9A
Û	%DB	%C3%9B
Ü	%DC	%C3%9C
Ý	%DD	%C3%9D
Þ	%DE	%C3%9E
ß	%DF	%C3%9F
à	%E0	%C3%A0
á	%E1	%C3%A1
â	%E2	%C3%A2
ã	%E3	%C3%A3
ä	%E4	%C3%A4
å	%E5	%C3%A5
æ	%E6	%C3%A6
ç	%E7	%C3%A7
è	%E8	%C3%A8
é	%E9	%C3%A9
ê	%EA	%C3%AA
ë	%EB	%C3%AB
ì	%EC	%C3%AC
í	%ED	%C3%AD
î	%EE	%C3%AE
ï	%EF	%C3%AF
ð	%F0	%C3%B0
ñ	%F1	%C3%B1
ò	%F2	%C3%B2
ó	%F3	%C3%B3
ô	%F4	%C3%B4
õ	%F5	%C3%B5
ö	%F6	%C3%B6
÷	%F7	%C3%B7
ø	%F8	%C3%B8
ù	%F9	%C3%B9
ú	%FA	%C3%BA
û	%FB	%C3%BB



ü	%FC	%C3%BC
ý	%FD	%C3%BD
þ	%FE	%C3%BE
ÿ	%FF	%C3%BF



## 83.6 URL Encoding Reference

The ASCII control characters **%00-%1F** were originally designed to control hardware devices.

Control characters have nothing to do inside a URL.

ASCII Character	Description	URL-encoding
NUL	null character	%00
SOH	start of header	%01
STX	start of text	%02
ETX	end of text	%03
EOT	end of transmission	%04
ENQ	enquiry	%05
ACK	acknowledge	%06
BEL	bell (ring)	%07
BS	backspace	%08
HT	horizontal tab	%09
LF	line feed	%0A
VT	vertical tab	%0B
FF	form feed	%0C
CR	carriage return	%0D
SO	shift out	%0E
SI	shift in	%0F
DLE	data link escape	%10
DC1	device control 1	%11
DC2	device control 2	%12
DC3	device control 3	%13
DC4	device control 4	%14
NAK	negative acknowledge	%15
SYN	synchronize	%16
ETB	end transmission block	%17
CAN	cancel	%18
EM	end of medium	%19
SUB	substitute	%1A

ESC	escape	%1B
FS	file separator	%1C
GS	group separator	%1D
RS	record separator	%1E
US	unit separator	%1F



## 84 HTML Language Code Reference

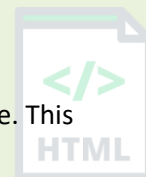
### 84.1 ISO Language Codes

You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers:

```
<html lang="en">
...
</html>
```

In XHTML, the language is declared inside the `<html>` tag as follows:

```
<html xmlns="http://www.org/1999/xhtml" lang="en" xml:lang="en">
...
</html>
```



### 84.2 ISO 639-1 Language Codes

ISO 639-1 defines abbreviations for languages:

See also: [Reference for Country Codes](#).

Language	ISO Code
Abkhazian	ab
Afar	aa
Afrikaans	af
Akan	ak
Albanian	sq
Amharic	am
Arabic	ar
Aragonese	an
Armenian	hy
Assamese	as
Avaric	av
Avestan	ae
Aymara	ay
Azerbaijani	az
Bambara	bm
Bashkir	ba
Basque	eu
Belarusian	be
Bengali (Bangla)	bn



<b>Bihari</b>	bh
<b>Bislama</b>	bi
<b>Bosnian</b>	bs
<b>Breton</b>	br
<b>Bulgarian</b>	bg
<b>Burmese</b>	my
<b>Catalan</b>	ca
<b>Chamorro</b>	ch
<b>Chechen</b>	ce
<b>Chichewa, Chewa, Nyanja</b>	ny
<b>Chinese</b>	zh
<b>Chinese (Simplified)</b>	zh-Hans
<b>Chinese (Traditional)</b>	zh-Hant
<b>Chuvash</b>	cv
<b>Cornish</b>	kw
<b>Corsican</b>	co
<b>Cree</b>	cr
<b>Croatian</b>	hr
<b>Czech</b>	cs
<b>Danish</b>	da
<b>Divehi, Dhivehi, Maldivian</b>	dv
<b>Dutch</b>	nl
<b>Dzongkha</b>	dz
<b>English</b>	en
<b>Esperanto</b>	eo
<b>Estonian</b>	et
<b>Ewe</b>	ee
<b>Faroese</b>	fo
<b>Fijian</b>	fj
<b>Finnish</b>	fi
<b>French</b>	fr
<b>Fula, Fulah, Pulaar, Pular</b>	ff
<b>Galician</b>	gl
<b>Gaelic (Scottish)</b>	gd
<b>Gaelic (Manx)</b>	gv
<b>Georgian</b>	ka
<b>German</b>	de



Greek	el
Greenlandic	kl
Guarani	gn
Gujarati	gu
Haitian Creole	ht
Hausa	ha
Hebrew	he
Herero	hz
Hindi	hi
Hiri Motu	ho
Hungarian	hu
Icelandic	is
Ido	io
Igbo	ig
Indonesian	id, in
Interlingua	ia
Interlingue	ie
Inuktitut	iu
Inupiak	ik
Irish	ga
Italian	it
Japanese	ja
Javanese	jv
Kalaallisut, Greenlandic	kl
Kannada	kn
Kanuri	kr
Kashmiri	ks
Kazakh	kk
Khmer	km
Kikuyu	ki
Kinyarwanda (Rwanda)	rw
Kirundi	rn
Kyrgyz	ky
Komi	kv
Kongo	kg
Korean	ko
Kurdish	ku



Kwanyama	kj
Lao	lo
Latin	la
Latvian (Lettish)	lv
Limburgish ( Limburger)	li
Lingala	ln
Lithuanian	lt
Luga-Katanga	lu
Luganda, Ganda	lg
Luxembourgish	lb
Manx	gv
Macedonian	mk
Malagasy	mg
Malay	ms
Malayalam	ml
Maltese	mt
Maori	mi
Marathi	mr
Marshallese	mh
Moldavian	mo
Mongolian	mn
Nauru	na
Navajo	nv
Ndonga	ng
Northern Ndebele	nd
Nepali	ne
Norwegian	no
Norwegian bokmål	nb
Norwegian nynorsk	nn
Nuosu	ii
Occitan	oc
Ojibwe	oj
Old Church Slavonic, Old Bulgarian	cu
Oriya	or
Oromo (Afaan Oromo)	om
Ossetian	os
Pāli	pi



Pashto, Pushto	ps
Persian (Farsi)	fa
Polish	pl
Portuguese	pt
Punjabi (Eastern)	pa
Quechua	qu
Romansh	rm
Romanian	ro
Russian	ru
Sami	se
Samoan	sm
Sango	sg
Sanskrit	sa
Serbian	sr
Serbo-Croatian	sh
Sesotho	st
Setswana	tn
Shona	sn
Sichuan Yi	ii
Sindhi	sd
Sinhalese	si
Siswati	ss
Slovak	sk
Slovenian	sl
Somali	so
Southern Ndebele	nr
Spanish	es
Sundanese	su
Swahili (Kiswahili)	sw
Swati	ss
Swedish	sv
Tagalog	tl
Tahitian	ty
Tajik	tg
Tamil	ta
Tatar	tt
Telugu	te



Thai	th
Tibetan	bo
Tigrinya	ti
Tonga	to
Tsonga	ts
Turkish	tr
Turkmen	tk
Twi	tw
Uyghur	ug
Ukrainian	uk
Urdu	ur
Uzbek	uz
Venda	ve
Vietnamese	vi
Volapük	vo
Wallon	wa
Welsh	cy
Wolof	wo
Western Frisian	fy
Xhosa	xh
Yiddish	yi, ji
Yoruba	yo
Zhuang, Chuang	za
Zulu	zu



## 85 HTML ISO Country Codes Reference

### 85.1 ISO Country Codes

In HTML, country codes can be used as an addition to the language code in the `lang` attribute.

The first two characters of a language code defines the language of the Web page (See [Language Code Reference](#)).

The last two characters define the country.

The following example specifies English as language and United States as country:

```
<html lang="en-US">
```

```
...
```

```
</html>
```



### 85.2 ISO Country Codes

Country	ISO Code
AFGHANISTAN	AF
ALBANIA	AL
ALGERIA	DZ
AMERICAN SAMOA	AS
ANDORRA	AD
ANGOLA	AO
ANTARCTICA	AQ
ANTIGUA AND BARBUDA	AG
ARGENTINA	AR
ARMENIA	AM
ARUBA	AW
AUSTRALIA	AU
AUSTRIA	AT
AZERBAIJAN	AZ
BAHAMAS	BS
BAHRAIN	BH
BANGLADESH	BD
BARBADOS	BB
BELARUS	BY
BELGIUM	BE
BELIZE	BZ
BENIN	BJ
BERMUDA	BM
BHUTAN	BT
BOLIVIA	BO
BOSNIA AND HERZEGOVINA	BA
BOTSWANA	BW
BOUVET ISLAND	BV
BRAZIL	BR
BRITISH INDIAN OCEAN TERRITORY	IO
BRUNEI DARUSSALAM	BN
BULGARIA	BG
BURKINA FASO	BF
BURUNDI	BI
CAMBODIA	KH
CAMEROON	CM

CANADA	CA
CAPE VERDE	CV
CAYMAN ISLANDS	KY
CENTRAL AFRICAN REPUBLIC	CF
CHAD	TD
CHILE	CL
CHINA	CN
CHRISTMAS ISLAND	CX
COCOS (KEELING) ISLANDS	CC
COLOMBIA	CO
COMOROS	KM
CONGO	CG
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD
COOK ISLANDS	CK
COSTA RICA	CR
CÔTE D'IVOIRE	CI
CROATIA	HR
CUBA	CU
CYPRUS	CY
CZECH REPUBLIC	CZ
DENMARK	DK
DJIBOUTI	DJ
DOMINICA	DM
DOMINICAN REPUBLIC	DO
ECUADOR	EC
EGYPT	EG
EL SALVADOR	SV
EQUATORIAL GUINEA	GQ
ERITREA	ER
ESTONIA	EE
ETHIOPIA	ET
FALKLAND ISLANDS (MALVINAS)	FK
FAROE ISLANDS	FO
FIJI	FJ
FINLAND	FI
FRANCE	FR
FRENCH GUIANA	GF
FRENCH POLYNESIA	PF
FRENCH SOUTHERN TERRITORIES	TF
GABON	GA
GAMBIA	GM
GEORGIA	GE
GERMANY	DE
GHANA	GH
GIBRALTAR	GI
GREECE	GR
GREENLAND	GL
GRENADA	GD
GUADELOUPE	GP
GUAM	GU
GUATEMALA	GT
GUINEA	GN
GUINEA-BISSAU	GW
GUYANA	GY



HAITI	HT
HEARD ISLAND AND MCDONALD ISLANDS	HM
HONDURAS	HN
HONG KONG	HK
HUNGARY	HU
ICELAND	IS
INDIA	IN
INDONESIA	ID
IRAN, ISLAMIC REPUBLIC OF	IR
IRAQ	IQ
IRELAND	IE
ISRAEL	IL
ITALY	IT
JAMAICA	JM
JAPAN	JP
JORDAN	JO
KAZAKHSTAN	KZ
KENYA	KE
KIRIBATI	KI
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP
KOREA, REPUBLIC OF	KR
KUWAIT	KW
KYRGYZSTAN	KG
LAO PEOPLE'S DEMOCRATIC REPUBLIC (LAOS)	LA
LATVIA	LV
LEBANON	LB
LESOTHO	LS
LIBERIA	LR
LIBYA, STATE OF	LY
LIECHTENSTEIN	LI
LITHUANIA	LT
LUXEMBOURG	LU
MACAO	MO
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK
MADAGASCAR	MG
MALAWI	MW
MALAYSIA	MY
MALDIVES	MV
MALI	ML
MALTA	MT
MARSHALL ISLANDS	MH
MARTINIQUE	MQ
MAURITANIA	MR
MAURITIUS	MU
MAYOTTE	YT
MEXICO	MX
MICRONESIA, FEDERATED STATES OF	FM
MOLDOVA, REPUBLIC OF	MD
MONACO	MC
MONGOLIA	MN
MONTENEGRO	ME
MONTSERRAT	MS
MOROCCO	MA
MOZAMBIQUE	MZ





MYANMAR	MM
NAMIBIA	NA
NAURU	NR
NEPAL, FEDERAL DEMOCRATIC REPUBLIC OF	NP
NETHERLANDS	NL
NETHERLANDS ANTILLES	AN
NEW CALEDONIA	NC
NEW ZEALAND	NZ
NICARAGUA	NI
NIGER	NE
NIGERIA	NG
NIUE	NU
NORFOLK ISLAND	NF
NORTHERN MARIANA ISLANDS	MP
NORWAY	NO
OMAN	OM
PAKISTAN	PK
PALAU	PW
PALESTINE, STATE OF	PS
PANAMA	PA
PAPUA NEW GUINEA	PG
PARAGUAY	PY
PERU	PE
PHILIPPINES	PH
PITCAIRN	PN
POLAND	PL
PORTUGAL	PT
PUERTO RICO	PR
QATAR	QA
RÉUNION	RE
ROMANIA	RO
RUSSIAN FEDERATION	RU
RWANDA	RW
SAINT HELENA	SH
SAINT KITTS AND NEVIS	KN
SAINT LUCIA	LC
SAINT PIERRE AND MIQUELON	PM
SAINT VINCENT AND THE GRENADINES	VC
SAMOA	WS
SAN MARINO	SM
SÃO TOME AND PRÍNCIPE	ST
SAUDI ARABIA	SA
SENEGAL	SN
SERBIA	RS
SEYCHELLES	SC
SIERRA LEONE	SL
SINGAPORE	SG
SLOVAKIA	SK
SLOVENIA	SI
SOLOMON ISLANDS	SB
SOMALIA	SO
SOUTH AFRICA	ZA
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS
SOUTH SUDAN	SS



SPAIN	ES
SRI LANKA	LK
SUDAN	SD
SURINAME	SR
SVALBARD AND JAN MAYEN	SJ
SWAZILAND	SZ
SWEDEN	SE
SWITZERLAND	CH
SYRIAN ARAB REPUBLIC	SY
TAIWAN	TW
TAJKISTAN	TJ
TANZANIA, UNITED REPUBLIC OF	TZ
THAILAND	TH
TIMOR-LESTE	TL
TOGO	TG
TOKELAU	TK
TONGA	TO
TRINIDAD AND TOBAGO	TT
TUNISIA	TN
TURKEY	TR
TURKMENISTAN	TM
TURKS AND CAICOS ISLANDS	TC
TUVALU	TV
UGANDA	UG
UKRAINE	UA
UNITED ARAB EMIRATES	AE
UNITED KINGDOM	GB
UNITED STATES	US
UNITED STATES MINOR OUTLYING ISLANDS	UM
URUGUAY	UY
UZBEKISTAN	UZ
VANUATU	VU
VENEZUELA	VE
VIET NAM	VN
VIRGIN ISLANDS, BRITISH	VG
VIRGIN ISLANDS, U.S.	VI
WALLIS AND FUTUNA	WF
WESTERN SAHARA	EH
YEMEN	YE
ZAMBIA	ZM
ZIMBABWE	ZW



## 86 HTTP Status Messages

### 86.1 HTML Error Messages

When a browser requests a service from a web server, an error might occur, and the server might return an error code like "404 Not Found".

It is common to name these errors HTML error messages.

But these messages are something called HTTP status messages. In fact, the server always returns a message for every request. The most common message is 200 OK.

Below is a list of HTTP status messages that might be returned:



### 86.2 1xx: Information

Message:	Description:
<b>100 Continue</b>	The server has received the request headers, and the client should proceed to send the request body
<b>101 Switching Protocols</b>	The requester has asked the server to switch protocols
<b>103 Early Hints</b>	Used with the Link header to allow the browser to start preloading resources while the server prepares a response

### 86.3 2xx: Successful

Message:	Description:
<b>200 OK</b>	The request is OK (this is the standard response for successful HTTP requests)
<b>201 Created</b>	The request has been fulfilled, and a new resource is created
<b>202 Accepted</b>	The request has been accepted for processing, but the processing has not been completed
<b>203 Non-Authoritative Information</b>	The request has been successfully processed, but is returning information that may be from another source
<b>204 No Content</b>	The request has been successfully processed, but is not returning any content
<b>205 Reset Content</b>	The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view
<b>206 Partial Content</b>	The server is delivering only part of the resource due to a range header sent by the client

### 86.4 3xx: Redirection

Message:	Description:
<b>300 Multiple Choices</b>	A link list. The user can select a link and go to that location. Maximum five addresses

<b>301 Moved Permanently</b>	The requested page has moved to a new URL
<b>302 Found</b>	The requested page has moved temporarily to a new URL
<b>303 See Other</b>	The requested page can be found under a different URL
<b>304 Not Modified</b>	Indicates the requested page has not been modified since last requested
<b>307 Temporary Redirect</b>	The requested page has moved temporarily to a new URL
<b>308 Permanent Redirect</b>	The requested page has moved permanently to a new URL



## 86.5 4xx: Client Error

Message:	Description:
<b>400 Bad Request</b>	The request cannot be fulfilled due to bad syntax
<b>401 Unauthorized</b>	The request was a legal request, but the server is refusing to respond to it. For use when authentication is possible but has failed or not yet been provided
<b>402 Payment Required</b>	<i>Reserved for future use</i>
<b>403 Forbidden</b>	The request was a legal request, but the server is refusing to respond to it
<b>404 Not Found</b>	The requested page could not be found but may be available again in the future
<b>405 Method Not Allowed</b>	A request was made of a page using a request method not supported by that page
<b>406 Not Acceptable</b>	The server can only generate a response that is not accepted by the client
<b>407 Proxy Authentication Required</b>	The client must first authenticate itself with the proxy
<b>408 Request Timeout</b>	The server timed out waiting for the request
<b>409 Conflict</b>	The request could not be completed because of a conflict in the request
<b>410 Gone</b>	The requested page is no longer available
<b>411 Length Required</b>	The "Content-Length" is not defined. The server will not accept the request without it
<b>412 Precondition Failed</b>	The precondition given in the request evaluated to false by the server
<b>413 Request Too Large</b>	The server will not accept the request, because the request entity is too large
<b>414 Request-URI Too Long</b>	The server will not accept the request, because the URI is too long. Occurs when you convert a POST request to a GET request with a long query information
<b>415 Unsupported Media Type</b>	The server will not accept the request, because the media type is not supported

<b>416 Range Not Satisfiable</b>	The client has asked for a portion of the file, but the server cannot supply that portion
<b>417 Expectation Failed</b>	The server cannot meet the requirements of the Expect request-header field



## 86.6 5xx: Server Error

Message:	Description:
<b>500 Internal Server Error</b>	A generic error message, given when no more specific message is suitable
<b>501 Not Implemented</b>	The server either does not recognize the request method, or it lacks the ability to fulfill the request
<b>502 Bad Gateway</b>	The server was acting as a gateway or proxy and received an invalid response from the upstream server
<b>503 Service Unavailable</b>	The server is currently unavailable (overloaded or down)
<b>504 Gateway Timeout</b>	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server
<b>505 HTTP Version Not Supported</b>	The server does not support the HTTP protocol version used in the request
<b>511 Network Authentication Required</b>	The client needs to authenticate to gain network access

## 87 HTTP Request Methods

### 87.1 What is HTTP?

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

**Example:** A client (browser) sends an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.



### 87.2 HTTP Methods

- GET
- POST
- **PUT**
- **HEAD**
- **DELETE**
- **PATCH**
- **OPTIONS**
- **CONNECT**
- **TRACE**

The two most common HTTP methods are: GET and POST.

### 87.3 The GET Method

GET is used to request data from a specified resource.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

```
/test/demo_form.php?name1=value1&name2=value2
```

**Some notes on GET requests:**

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests are only used to request data (not modify)

### 87.4 The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1
Host: bintr.com
```

```
name1=value1&name2=value2
```

#### Some notes on POST requests:

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length



## 87.5 Compare GET vs. POST

The following table compares the two HTTP methods: GET and POST.

	GET	POST
<b>BACK button/Reload</b>	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
<b>Bookmarked</b>	Can be bookmarked	Cannot be bookmarked
<b>Cached</b>	Can be cached	Not cached
<b>Encoding type</b>	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
<b>History</b>	Parameters remain in browser history	Parameters are not saved in browser history
<b>Restrictions on data length</b>	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
<b>Restrictions on data type</b>	Only ASCII characters allowed	No restrictions. Binary data is also allowed
<b>Security</b>	GET is less secure compared to POST because data sent is part of the URL  Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
<b>Visibility</b>	Data is visible to everyone in the URL	Data is not displayed in the URL

## 87.6 The PUT Method

PUT is used to send data to a server to create/update a resource.

The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly have side effects of creating the same resource multiple times.

## 87.7 The HEAD Method

HEAD is almost identical to GET, but without the response body.

In other words, if GET /users returns a list of users, then HEAD /users will make the same request but will not return the list of users.

HEAD requests are useful for checking what a GET request will return before actually making a GET request - like before downloading a large file or response body.



## 87.8 The DELETE Method

The DELETE method deletes the specified resource.

## 87.9 The PATCH Method

The PATCH method is used to apply partial modifications to a resource.

## 87.10The OPTIONS Method

The OPTIONS method describes the communication options for the target resource.

## 87.11The CONNECT Method

The CONNECT method is used to start a two-way communications (a tunnel) with the requested resource.

## 87.12The TRACE Method

The TRACE method method is used to perform a message loop-back test that tests the path for the target resource (useful for debugging purposes).



## 88 Pixels to Ems Conversion

### 88.1 Pixel to Em Converter



The tool below allows you to work out the em sizes from pixels (or vice versa).

- Set a default pixel size for body (usually 16px)
- Then, convert a pixel value to em, based on the default pixel size
- Or, convert an em value to pixels, based on the default pixel size

### 88.2 Body Font Size

In the table below, select a body font size in pixels (px) to display a complete "px to em and percent" conversion table.

**Tip:** The default font size is usually 16px.

px	em	percent
5px	0.3125em	31.25%
6px	0.3750em	37.50%
7px	0.4375em	43.75%
8px	0.5000em	50.00%
9px	0.5625em	56.25%
10px	0.6250em	62.50%
11px	0.6875em	68.75%
12px	0.7500em	75.00%
13px	0.8125em	81.25%
14px	0.8750em	87.50%
15px	0.9375em	93.75%
16px	1.0000em	100.00%
17px	1.0625em	106.25%
18px	1.1250em	112.50%
19px	1.1875em	118.75%
20px	1.2500em	125.00%
21px	1.3125em	131.25%
22px	1.3750em	137.50%
23px	1.4375em	143.75%
24px	1.5000em	150.00%
25px	1.5625em	156.25%

#### What is the difference between PX, EM and Percent?

Pixel is a static measurement, while percent and EM are relative measurements. The size of an EM or percent depends on its parent. If the text size of body is 16 pixels, then 150% or 1.5 EM will be 24 pixels (1.5 \* 16). Look at CSS Units for more measurement units.



## 89 Keyboard Shortcuts

### 89.1 Keyboard Shortcuts For Windows and Mac

Keyboard shortcuts are often used in modern operating systems and computer software programs.

Learning and using keyboard shortcuts can save you a lot of time.



### 89.2 Basic Shortcuts

Description	Windows	Mac OS
Edit menu	Alt + E	Ctrl + F2 + F
File menu	Alt + F	Ctrl + F2 + E
View menu	Alt + V	Ctrl + F2 + V
Select all text	Ctrl + A	Cmd + A
Copy text	Ctrl + C	Cmd + C
Find text	Ctrl + F	Cmd + F
Find and replace text	Ctrl + H	Cmd + F
New Document	Ctrl + N	Cmd + N
Open a file	Ctrl + O	Cmd + O
Print options	Ctrl + P	Cmd + P
Save file	Ctrl + S	Cmd + S
Paste text	Ctrl + V	Cmd + V
Cut text	Ctrl + X	Cmd + X
Redo text	Ctrl + Y	Shift + Cmd + Z
Undo text	Ctrl + Z	Cmd + Z

### 89.3 Text Editing

Description	Windows	Mac OS
<b>Cursor Movement</b>		
Go to the right or to the beginning of line break	Right Arrow	Right Arrow
Go to the left or to the end of previous line break	Left Arrow	Left Arrow
Go up one row	Up Arrow	Up Arrow
Go down one row	Down Arrow	Down Arrow
Go to the beginning of the current line	Home	Cmd + Left Arrow
Go to the end of the current line	End	Cmd + Right Arrow
Go to the beginning of the document	Ctrl + Home	Cmd + Up Arrow

Go to the end of the document	Ctrl + End	Cmd + Down Arrow
Move up one frame	Page Up	Fn + Up Arrow
Move down one frame	Page Down	Fn + Down Arrow
Go to beginning of previous word	Ctrl + Left Arrow	Option + Left Arrow
Go to beginning of word	Ctrl + Right Arrow	Option + Right Arrow
Go to beginning of line break	Ctrl + Up Arrow	Cmd + Left Arrow
Go to end of line break	Ctrl + Down Arrow	Cmd + Right Arrow
<b>Text Selection</b>		
Select characters to the left	Shift + Left Arrow	Shift + Left Arrow
Select characters to the right	Shift + Right Arrow	Shift + Right Arrow
Select lines upwards	Shift + Up Arrow	Shift + Up Arrow
Select lines downwards	Shift + Down Arrow	Shift + Down Arrow
Select words to the left	Shift + Ctrl + Left	Shift + Opt + Left
Select words to the right	Shift + Ctrl + Right	Shift + Opt + Right
Select paragraphs to the left	Shift + Ctrl + Up	Shift + Opt + Up
Select paragraphs to the right	Shift + Ctrl + Down	Shift + Opt + Down
Select text between the cursor and the beginning of the current line	Shift + Home	Cmd + Shift + Left Arrow
Select text between the cursor and the end of the current line	Shift + End	Cmd + Shift + Right Arrow
Select text between the cursor and the beginning of the document	Shift + Ctrl + Home	Cmd + Shift + Up Arrow or Cmd + Shift + Fn + Left Arrow
Select text between the cursor and the end of the document	Shift + Ctrl + End	Cmd + Shift + Down Arrow or Cmd + Shift + Fn + Right Arrow
Select one frame at a time of text above the cursor	Shift + Page Up	Shift + Fn + Up Arrow
Select one frame at a time of text below the cursor	Shift + Page Down	Shift + Fn + Down Arrow
Select all text	Ctrl + A	Cmd + A
Find text	Ctrl + F	Cmd + F
<b>Text Formatting</b>		
Make selected text bold	Ctrl + B	Cmd + B
Make selected text italic	Ctrl + I	Cmd + I
Underline selected text	Ctrl + U	Cmd + U
Make selected text superscript	Ctrl + Shift + =	Cmd + Shift + =



Make selected text subscript	Ctrl + =	Cmd + =
Text Editing		
Delete characters to the left	Backspace	Backspace
Delete characters to the right	Delete	Fn + Backspace
Delete words to the right	Ctrl + Del	Cmd + Backspace
Delete words to the left	Ctrl + Backspace	Cmd + Fn + Backspace
Indent	Tab	Tab
Outdent	Shift + Tab	Shift + Tab
Copy text	Ctrl + C	Cmd + C
Find and replace text	Ctrl + H	Cmd + F
Paste text	Ctrl + V	Cmd + V
Cut text	Ctrl + X	Cmd + X
Redo text	Ctrl + Y	Shift + Cmd + Z
Undo text	Ctrl + Z	Cmd + Z



## 89.4 Web Browsers

Description	Windows	Mac OS
Navigation		
Scroll down a frame	Space or Page Down	Space or Fn + Down Arrow
Scroll up a frame	Shift + Space or Page Up	Shift + Space or Fn + Up Arrow
Go to bottom of the page	End	Cmd + Down Arrow
Go to top of the page	Home	Cmd + Up Arrow
Go back	Alt + Left Arrow or Backspace	Cmd + Left Arrow
Go forward	Alt + Right Arrow or Shift + Backspace	Cmd + Right Arrow
Refresh a webpage	F5	Cmd + R
Refresh a webpage (no cache)	Ctrl + F5	Cmd + Shift + R
Stop	Esc	Esc
Toggle full-screen	F11	Cmd + Shift + F
Zoom in	Ctrl + +	Cmd + +
Zoom out	Ctrl + -	Cmd + -
Zoom 100% (default)	Ctrl + 0	Cmd + 0

Open homepage	Alt + Home	Option + Home or Option + Fn + Left Arrow
Find text	Ctrl + F	Cmd + F
<b>Tab / Window Management</b>		
Open a new tab	Ctrl + T	Cmd + T
Close current tab	Ctrl + W	Cmd + W
Close all tabs	Ctrl + Shift + W	Cmd + Q
Close all tabs except the current tab	Ctrl + Alt + F4	Cmd + Opt + W
Go to tab	Ctrl + Tab	Control + Tab or Cmd + Shift + Right Arrow
Go to previous tab	Ctrl + Shift + Tab	Shift + Control + Tab or Cmd + Shift + Left Arrow
Go to a specific tab number	Ctrl + 1-8	Cmd + 1-8
Go to the last tab	Ctrl + 9	Cmd + 9
Reopen the last closed tab	Ctrl + Shift + T	Cmd + Shift + T
Open a new window	Ctrl + N	Cmd + N
Close current window	Alt + F4	Cmd + W
Go to window	Alt + Tab	Cmd + Tab
Go to previous window	Alt + Shift + Tab	Cmd + Shift + Tab
Reopen the last closed window	Ctrl + Shift + N	
Open links in a new tab in the background	Ctrl + Click	Cmd + Click
Open links in a new tab in the foreground	Ctrl + Shift + Click	Cmd + Shift + Click
Print current webpage	Ctrl + P	Cmd + P
Save current webpage	Ctrl + S	Cmd + S
<b>Address Bar</b>		
Cycle between toolbar, search bar, and page elements	Tab	Tab
Go to browser's address bar	Ctrl + L or Alt + D	Cmd + L
Focus and select the browser's search bar	Ctrl + E	Cmd + E / Cmd + K
Open the address bar location in a new tab	Alt + Enter	Opt + Enter
Display a list of previously typed addresses	F4	
Add "www." to the beginning and ".com" to the end of the text typed in the address bar (e.g., type "bintr" and press Ctrl + Enter to open "www.bintr.com")	Ctrl + Enter	Cmd + Enter or Control + Enter
<b>Bookmarks</b>		



Open the bookmarks menu	Ctrl + B	Cmd + B
Add bookmark for current page	Ctrl + D	Cmd + Opt + B or Cmd + Shift + B
Open browsing history	Ctrl + H	Cmd + Shift + H or Cmd + Y
Open download history	Ctrl + J	Cmd + J or Cmd + Shift + J



## 89.5 Screenshots

Description	Windows	Mac OS
Save screenshot of the whole screen as file		Cmd + Shift + 3
Copy screenshot of the whole screen to the clipboard	PrtScr (Print Screen) or Ctrl + PrtScr	Cmd + Ctrl + Shift + 3
Save screenshot of window as file		Cmd + Shift + 4, then Space
Copy screenshot of window to the clipboard	Alt + PrtScr	Cmd + Ctrl + Shift + 4, then Space
Copy screenshot of wanted area to the clipboard		Cmd + Ctrl + Shift + 4
Save screenshot of wanted area as file		Cmd + Shift + 4

**Note:** Due to different keyboard setups, some shortcuts may not be compatible for all users.

# 90 HTML Tags

## 90.1 HTML `<!--...-->` Tag

### Example:

An HTML comment:

```
<!--This is a comment. Comments are not displayed in the browser-->
```

```
<p>This is a paragraph.</p>
```

### Definition and Usage

The comment tag is used to insert comments in the source code. Comments are not displayed in the browsers.

You can use comments to explain your code, which can help you when you edit the source code at a later date. This is especially useful if you have a lot of code.

## Tips and Notes

You can use the comment tag to "hide" scripts from browsers without support for scripts (so they don't show them as plain text):

```
<script type="text/javascript">
<!--
function displayMsg() {
    alert("Hello World!")
}
//-->
</script>
```



**Note:** The two forward slashes at the end of comment line (//) is the JavaScript comment symbol. This prevents JavaScript from executing the --> tag.

## Standard Attributes

The comment tag does not support any standard attributes.

## Event Attributes

The comment tag does not support any event attributes.

## 90.2 HTML <!DOCTYPE> Declaration

### Example:

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document.....
</body>
</html>
```

## Definition and Usage

All HTML documents must start with a <!DOCTYPE> declaration.

The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

In HTML 5, the declaration is simple:

```
<!DOCTYPE html>
```

## Older HTML Documents



In older documents (HTML 4 or XHTML), the declaration is more complicated because the declaration must refer to a DTD ( ).

HTML 4.01:

```
<!DOCTYPE HTML PUBLIC "-//C//DTD HTML 4.01 Transitional//EN"
"http://www..org/TR/html4/loose.dtd">
```

XHTML 1.1:

```
<!DOCTYPE html PUBLIC "-//C//DTD XHTML 1.1//EN"
"http://www..org/TR/xhtml11/DTD/xhtml11.dtd">
```



## HTML Elements and Doctypes

Look at our table of all HTML elements, and what Doctype each element appears in.

### Tips and Notes

**Tip:** The `<!DOCTYPE>` declaration is NOT case sensitive.

#### Example:

```
<!DOCTYPE html>
<!DocType html>
<!Doctype html>
<!doctype html>
```

## 90.3 HTML `<a>` Tag

#### Example:

Create a link to Bintr.com:

```
<a href="https://www.bintr.com">Visit Bintr.com!</a>
```

**"Try it Yourself"**

### Definition and Usage

The `<a>` tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

### Tips and Notes

**Tip:** If the `<a>` tag has no `href` attribute, it is only a placeholder for a hyperlink.

**Tip:** A linked page is normally displayed in the current browser window, unless you specify another target.

**Tip:** Use CSS to style links: CSS Links and CSS Buttons.



## Attributes

Attribute	Value	Description
<b>download</b>	<i>filename</i>	Specifies that the target will be downloaded when a user clicks on the hyperlink
<b>href</b>	<i>URL</i>	Specifies the URL of the page the link goes to
<b>hreflang</b>	<i>language_code</i>	Specifies the language of the linked document
<b>media</b>	<i>media_query</i>	Specifies what media/device the linked document is optimized for
<b>ping</b>	<i>list_of_URLs</i>	Specifies a space-separated list of URLs to which, when the link is followed, post requests with the body ping will be sent by the browser (in the background). Typically used for tracking.
<b>referrerpolicy</b>	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin-when-cross-origin unsafe-url	Specifies which referrer information to send with the link
<b>rel</b>	alternate author bookmark external help license  nofollow noreferrer noopener prev search tag	Specifies the relationship between the current document and the linked document
<b>target</b>	_blank _parent _self _top	Specifies where to open the linked document
<b>type</b>	<i>media_type</i>	Specifies the media type of the linked document

## Global Attributes

The `<a>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<a>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

How to use an image as a link:

```
<a href="https://www.bintr.com">

</a>
```

### Example:

How to open a link in a new browser window:

```
<a href="https://www.bintr.com" target="_blank">Visit Bintr.com!</a>
```

### Example:

How to link to an email address:

```
<a href="mailto:someone@example.com">Send email</a>
```

### Example:

How to link to a phone number:

```
<a href="tel:+4733378901">+47 333 78 901</a>
```

### Example:

How to link to another section on the same page:

```
<a href="#section2">Go to Section 2</a>
```

### Example:

How to link to a JavaScript:

```
<a href="javascript:alert('Hello World!');">Execute JavaScript</a>
```

## Default CSS Settings

Most browsers will display the `<a>` element with the following default values:

```
a:link, a:visited {
  color: (internal value);
  text-decoration: underline;
  cursor: auto;
}
```



```
a:link:active, a:visited:active {
  color: (internal value);
}
```



## 90.4 HTML <abbr> Tag

### Example:

An abbreviation is marked up as follows:

The `<abbr title="World Health Organization">WHO</abbr>` was founded in 1948.

More "Try it Yourself" examples below.

### Definition and Usage

The `<abbr>` tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM".

**Tip:** Use the global `title` attribute to show the description for the abbreviation/acronym when you mouse over the element.

### Global Attributes

The `<abbr>` tag supports the Global Attributes in HTML.

### Event Attributes

The `<abbr>` tag supports the [Event Attributes in HTML](#).

### More Examples

#### Example:

`<abbr>` can also be used with `<dfn>` to define an abbreviation:

```
<p><dfn><abbr title="Cascading Style Sheets">CSS</abbr>
</dfn> is a language that describes the style of an HTML document.</p>
```

### Default CSS Settings

Most browsers will display the `<abbr>` element with the following default values:

#### Example:

```
abbr {
  display: inline;
}
```

## 90.5 HTML <acronym> Tag

Not Supported in HTML5.

The `<acronym>` tag was used in HTML 4 to define an acronym.

## What to Use Instead?

### Example:

An acronym or abbreviation should be marked up with the `<abbr>` tag:

The `<abbr title="World Health Organization">WHO</abbr>` was founded in 1948.



## 90.6 HTML `<address>` Tag

### Example:

Contact information for Example.com:

```
<address>
Written by <a href="mailto:rabhabinod@146.com">Binod Rabha</a>.<br>
Visit us at:<br>
bintr.online <br>
Bardamal, Agia, Goalpara<br>
Assam
</address>
```

## Definition and Usage

The `<address>` tag defines the contact information for the author/owner of a document or an article.

The contact information can be an email address, URL, physical address, phone number, social media handle, etc.

The text in the `<address>` element usually renders in *italic*, and browsers will always add a line break before and after the `<address>` element.

## Global Attributes

The `<address>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<address>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<address>` element with the following default values:

### Example:

```
address {
  display: block;
```

```
font-style: italic;
}
```



## 90.7 HTML <applet> Tag

### Not Supported in HTML5.

The `<applet>` tag was used in HTML 4 to define an embedded applet (Plug-in).

### Plug-ins

Plug-ins are a computer programs that extend the standard functionality of the browser.

Plug-ins have been used for many different purposes:

- Run Java applets
- Run ActiveX controls
- Display Flash movies
- Display maps
- Scan for viruses
- Verify a bank id

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

### What to Use Instead?

If you want to embed a video, use the `<video>` tag:

#### Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

If you want to embed audio, use the `<audio>` tag:

#### Example:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

To embed objects, you can use both the `<embed>` tag and the `<object>` tags:

### Example:

Embed a document with the `<embed>` element:

```
<embed src="snippet.html">
```

### Example:

Embed a picture with the `<embed>` element:

```
<embed src="pic_trulli.jpg">
```

### Example:

Embed a document with the `<object>` element:

```
<object data="snippet.html"></object>
```

### Example:

Embed a picture with the `<object>` element:

```
<object data="pic_trulli.jpg"></object>
```

**Tip:** To embed a picture, it is better to use the `<img>` tag. To embed a document, it is better to use the `<iframe>` tag.



## 90.8 HTML `<area>` Tag

### Example:

An image map, with clickable areas:

```


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">
</map>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<area>` tag defines an area inside an image map (an image map is an image with clickable areas).

`<area>` elements are always nested inside a `<map>` tag.

**Note:** The `usemap` attribute in `<img>` is associated with the `<map>` element's `name` attribute, and creates a relationship between the image and the map.

### Attributes

Attribute	Value	Description
-----------	-------	-------------

<b>alt</b>	<i>text</i>	Specifies an alternate text for the area. Required if the href attribute is present
<b>coords</b>	<i>coordinates</i>	Specifies the coordinates of the area
<b>download</b>	<i>filename</i>	Specifies that the target will be downloaded when a user clicks on the hyperlink
<b>href</b>	<i>URL</i>	Specifies the hyperlink target for the area
<b>hreflang</b>	<i>language_code</i>	Specifies the language of the target URL
<b>media</b>	<i>media query</i>	Specifies what media/device the target URL is optimized for
<b>referrerpolicy</b>	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin-when-cross-origin unsafe-url	Specifies which referrer information to send with the link
<b>rel</b>	alternate author bookmark help license  nofollow noreferrer prefetch prev search tag	Specifies the relationship between the current document and the target URL
<b>shape</b>	default rect circle poly	Specifies the shape of the area
<b>target</b>	_blank _parent _self _top <i>framename</i>	Specifies where to open the target URL
<b>type</b>	<i>media_type</i>	Specifies the media type of the target URL



## Global Attributes

The `<area>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<area>` tag also supports the [Event Attributes in HTML](#).



## More Examples

### Example:

Another image map, with clickable areas:

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun">
  <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury">
  <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus">
</map>
```



## Default CSS Settings

Most browsers will display the `<area>` element with the following default values:

```
area {
  display: none;
}
```

## 90.9 HTML `<article>` Tag

### Example:

Three articles with independent, self-contained content:

```
<article>
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is the
world's most popular web browser today!</p>
</article>

<article>
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has
been the second most popular web browser since January, 2018.</p>
</article>

<article>
<h2>Microsoft Edge</h2>
<p>Microsoft Edge is a web browser developed by Microsoft, released in 2015. Microsoft
Edge replaced Internet Explorer.</p>
</article>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<article>` tag specifies independent, self-contained content.

An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

Potential sources for the `<article>` element:

- Forum post
- Blog post
- News story

**Note:** The `<article>` element does not render as anything special in a browser. However, you can use CSS to style the `<article>` element (see example below).

## Global Attributes

The `<article>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<article>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

Use CSS to style the `<article>` element:

```
<html>
<head>
<style>
.all-browsers {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

.all-browsers > h1, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
```



```
</head>
<body>

<article class="all-browsers">
  <h1>Most Popular Browsers</h1>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is
the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has
been the second most popular web browser since January, 2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
  </article>
</article>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<article>` element with the following default values:

```
article {
  display: block;
}
```

## 90.10HTML `<aside>` Tag

### Example:

Display some content aside from the content it is placed in:

```
<p>My family and I visited The Epcot center this summer. The weather was nice, and
Epcot was amazing! I had a great summer together with my family!</p>
```

```
<aside>
<h4>Epcot Center</h4>
<p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions,
international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<aside>` tag defines some content aside from the content it is placed in.

The aside content should be indirectly related to the surrounding content.

**Tip:** The `<aside>` content is often placed as a sidebar in a document.

**Note:** The `<aside>` element does not render as anything special in a browser. However, you can use CSS to style the `<aside>` element (see example below).



## Global Attributes

The `<aside>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<aside>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style the `<aside>` element:

```
<html>
<head>
<style>
aside {
  width: 30%;
  padding-left: 15px;
  margin-left: 15px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
</style>
</head>
<body>

<h1>The aside element</h1>

<p>My family and I visited The Epcot center this summer. The weather was nice, and Epcot was amazing! I had a great summer together with my family!</p>

<aside>
<p>The Epcot center is a theme park at Walt Disney World Resort featuring exciting attractions, international pavilions, award-winning fireworks and seasonal special events.</p>
</aside>
```

```
<p>My family and I visited The Epcot center this summer. The weather was nice, and
Epcot was amazing! I had a great summer together with my family!</p>
<p>My family and I visited The Epcot center this summer. The weather was nice, and
Epcot was amazing! I had a great summer together with my family!</p>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<aside>` element with the following default values:

```
aside {
  display: block;
}
```

## 90.11HTML <audio> Tag

### Example:

Play a sound file:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

### Definition and Usage

The `<audio>` tag is used to embed sound content in a document, such as music or other audio streams.

The `<audio>` tag contains one or more `<source>` tags with different audio sources. The browser will choose the first source it supports.

The text between the `<audio>` and `</audio>` tags will only be displayed in browsers that do not support the `<audio>` element.

There are three supported audio formats in HTML: MP3, WAV, and OGG.

### Audio Format and Browser Support

Browser	MP3	WAV	OGG
Edge / IE	YES	YES*	YES*
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO

Opera	YES	YES	YES
-------	-----	-----	-----



## Tips and Notes

**Tip:** For video files, look at the `<video>` tag.

## Attributes

Attribute	Value	Description
<code>autoplay</code>	autoplay	Specifies that the audio will start playing as soon as it is ready
<code>controls</code>	controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
<code>loop</code>	loop	Specifies that the audio will start over again, every time it is finished
<code>muted</code>	muted	Specifies that the audio output should be muted
<code>preload</code>	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
<code>src</code>	URL	Specifies the URL of the audio file

## Global Attributes

The `<audio>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<audio>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

None.

## 90.12HTML `<b>` Tag

### Example:

Make some text bold (without marking it as important):

```
<p>This is normal text - <b>and this is bold text</b>.</p>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<b>` tag specifies bold text without any extra importance.

### Tips and Notes

**Note:** According to the HTML5 specification, the `<b>` tag should be used as a LAST resort when no other tag is more appropriate. The specification states that headings should be denoted with the `<h1>` to `<h6>` tags, emphasized text should be denoted with the `<em>` tag, important text should be denoted with the `<strong>` tag, and marked/highlighted text should be denoted with the `<mark>` tag.

**Tip:** You can also use the following CSS to set bold text: "font-weight: bold;".



## Global Attributes

The `<b>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<b>` tag also supports the [Event Attributes in HTML](#).

### More Examples

#### Example:

Use CSS to set bold text:

```
<p>This is normal text - <span style="font-weight:bold;">and this is bold text</span>.</p>
```

## Default CSS Settings

Most browsers will display the `<b>` element with the following default values:

#### Example:

```
b {
  font-weight: bold;
}
```

## 90.13HTML `<base>` Tag

#### Example:

Specify a default URL and a default target for all links on a page:

```
<head>
  <base href="https://www.bintr.com/" target="_blank">
</head>

<body>

<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
```

## Definition and Usage

The `<base>` tag specifies the base URL and/or target for all relative URLs in a document.

The `<base>` tag must have either an href or a target attribute present, or both.

There can only be one single `<base>` element in a document, and it must be inside the `<head>` element.



## Attributes

Attribute	Value	Description
<code>href</code>	<i>URL</i>	Specifies the base URL for all relative URLs in the page
<code>target</code>	<code>_blank</code> <code>_parent</code> <code>_self</code> <code>_top</code>	Specifies the default target for all hyperlinks and forms in the page

## Global Attributes and Events

The `<base>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<base>` tag does not support any event attributes.

## Default CSS Settings

None.

## 90.14HTML `<basefont>` Tag

### Not Supported in HTML5.

The `<basefont>` tag was used in HTML 4 to specify a default text-color, font-size or font-family for all the text in an HTML document.

## What to Use Instead?

### Example:

Specify a default text-color for a page (with CSS):

```
<html>
<head>
<style>
body {
  color: red;
}
</style>
</head>
```



```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```



### Example:

Specify a default font-family for a page (with CSS):

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    font-family: courier, serif;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

### Example:

Specify a default font-size for a page (with CSS):

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    font-size: 50px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

**Tip:** In our [CSS tutorial](#) you can find more information about [CSS text color](#) and [CSS fonts](#).

## 90.15HTML <bdi> Tag

### Example:

Isolate the usernames from the surrounding text-direction settings:

```
<ul>
  <li>User <bdi>Binod Rabha</bdi>: 60 points</li>
  <li>User <bdi>Toya Rabha</bdi>: 80 points</li>
  <li>User <bdi>A Rabha</bdi>: 90 points</li>
</ul>
```



### Definition and Usage

BDI stands for Bi-Directional Isolation.

The <bdi> tag isolates a part of text that might be formatted in a different direction from other text outside it.

This element is useful when embedding user-generated content with an unknown text direction.

### Global Attributes

The <bdi> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <bdi> tag also supports the [Event Attributes in HTML](#).

## 90.16HTML <bdo> Tag

### Example:

Specify the text direction:

```
<bdo dir="rtl">
This text will go right-to-left.
</bdo>
```

### Definition and Usage

BDO stands for Bi-Directional Override.

The <bdo> tag is used to override the current text direction.

### Attributes

Attribute	Value	Description
<b>dir</b>	ltr rtl	Required. Specifies the text direction of the text inside the <bdo> element

## Global Attributes

The `<bdo>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<bdo>` tag also supports the [Event Attributes in HTML](#).



## Default CSS Settings

Most browsers will display the `<bdo>` element with the following default values:

```
bdo {  
  unicode-bidi: bidi-override;  
}
```

## 90.17HTML `<big>` Tag

### Not Supported in HTML5.

The `<big>` tag was used in HTML 4 to define bigger text.

### What to Use Instead?

#### Example:

Specify different font-sizes for HTML elements (with CSS):

```
<html>  
<head>  
<style>  
p.ex1 {  
  font-size: 30px;  
}  
p.ex2 {  
  font-size: 50px;  
}  
</style>  
</head>  
<body>  
  
<p>This is a normal paragraph.</p>  
<p class="ex1">This is a bigger paragraph.</p>  
<p class="ex2">This is a much bigger paragraph.</p>  
  
</body>  
</html>
```

## 90.18HTML `<blockquote>` Tag

#### Example:

A section that is quoted from another source:

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
```

```
For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.
```

```
</blockquote>
```



More "Try it Yourself" examples below.

## Definition and Usage

The `<blockquote>` tag specifies a section that is quoted from another source.

Browsers usually indent `<blockquote>` elements (look at example below to see how to remove the indentation).

## Tips and Notes

**Tip:** Use `<q>` for inline (short) quotations.

## Attributes

Attribute	Value	Description
<code>cite</code>	<i>URL</i>	Specifies the source of the quotation

## Global Attributes

The `<blockquote>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<blockquote>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to remove the indentation from the blockquote element:

```
<html>
<head>
<style>
blockquote {
  margin-left: 0;
}
</style>
</head>
<body>
```

```
<p>Here is a quote from WWF's website:</p>
```

```
<blockquote cite="http://www.worldwildlife.org/who/index.html">
```

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

```
</blockquote>
```

```
</body>
```

```
</html>
```



## Default CSS Settings

Most browsers will display the `<blockquote>` element with the following default values:

### Example:

```
blockquote {
  display: block;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-left: 40px;
  margin-right: 40px;
}
```

CSS tutorial you can find more information about [CSS Font Size](#).

## 90.19HTML <body> Tag

### Example:

A simple HTML document:

```
<html>
<head>
  <title>Title of the document</title>
</head>

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>

</html>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<body>` tag defines the document's body.

The `<body>` element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

**Note:** There can only be one `<body>` element in an HTML document.

## Global Attributes

The `<body>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<body>` tag also supports the [Event Attributes in HTML](#).



## More Examples

### Example:

Add a background image to a document (with CSS):

```
<html>
<head>
<style>
body {
  background-image: url(s.png);
}
</style>
</head>
<body>

<h1>Hello world!</h1>
<p><a href="https://www.bintr.com">Visit Bintr.com!</a></p>

</body>
```

### Example:

Set the background color of a document (with CSS):

```
<html>
<head>
<style>
body {
  background-color: #E6E6FA;
}
</style>
</head>
<body>

<h1>Hello world!</h1>
<p><a href="https://www.bintr.com">Visit Bintr.com!</a></p>

</body>
```

### Example:

Set the color of text in a document (with CSS):

```
<html>
<head>
<style>
```

```
body {
  color: green;
}
</style>
</head>
<body>

<h1>Hello world!</h1>
<p>This is some text.</p>
<p><a href="https://www.bintr.com">Visit Bintr.com!</a></p>

</body>
</html>
```



### Example:

Set the color of unvisited links in a document (with CSS):

```
<html>
<head>
<style>
a:link {
  color:#0000FF;
}
</style>
</head>
<body>

<p><a href="https://www.bintr.com">Bintr.com</a></p>
<p><a href="https://www.bintr.com/html/">HTML Tutorial</a></p>

</body>
</html>
```

### Example:

Set the color of active links in a document (with CSS):

```
<html>
<head>
<style>
a:active {
  color:#00FF00;
}
</style>
</head>
<body>

<p><a href="https://www.bintr.com">Bintr.com</a></p>
<p><a href="https://www.bintr.com/html/">HTML Tutorial</a></p>
```

```
</body>
</html>
```

### Example:

Set the color of visited links in a document (with CSS):

```
<html>
<head>
<style>
a:visited {
  color:#FF0000;
}
</style>
</head>
<body>

<p><a href="https://www.bintr.com">Bintr.com</a></p>
<p><a href="https://www.bintr.com/html/">HTML Tutorial</a></p>

</body>
</html>
```



### Default CSS Settings

Most browsers will display the `<body>` element with the following default values:

#### Example:

```
body {
  display: block;
  margin: 8px;
}

body:focus {
  outline: none;
}
```

## 90.20HTML `<br>` Tag

### Example:

Insert single line breaks in a text:

```
<p>To force<br> line breaks<br> in a text,<br> use the br<br> element.</p>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<br>` tag inserts a single line break.

The `<br>` tag is useful for writing addresses or poems.

The `<br>` tag is an empty tag which means that it has no end tag.



## Tips and Notes

**Note:** Use the `<br>` tag to enter line breaks, not to add space between paragraphs.

## Global Attributes

The `<br>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<br>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example

Use `<br>` in a poem:

```
<p>Be not afraid of greatness.<br>Some are born great,<br>some achieve greatness,<br>and others have greatness thrust upon them.</p>
```

```
<p><em>-William Shakespeare</em></p>
```

## Default CSS Settings

None.

## 90.21HTML <button> Tag

### Example:

A clickable button is marked up as follows:

```
<button type="button">Click Me!</button>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<button>` tag defines a clickable button.

Inside a `<button>` element you can put text (and tags like `<i>`, `<b>`, `<strong>`, `<br>`, `<img>`, etc.). That is not possible with a button created with the `<input>` element!

**Tip:** Always specify the `type` attribute for a `<button>` element, to tell browsers what type of button it is.

**Tip:** You can easily style buttons with CSS! Look at the examples below or visit our [CSS Buttons](#) tutorial.

## Attributes



Attribute	Value	Description
<b>autofocus</b>	autofocus	Specifies that a button should automatically get focus when the page loads
<b>disabled</b>	disabled	Specifies that a button should be disabled
<b>form</b>	<i>form_id</i>	Specifies which form the button belongs to
<b>formaction</b>	<i>URL</i>	Specifies where to send the form-data when a form is submitted. Only for type="submit"
<b>formenctype</b>	application/x-www-form-urlencoded multipart/form-data text/plain	Specifies how form-data should be encoded before sending it to a server. Only for type="submit"
<b>formmethod</b>	get post	Specifies how to send the form-data (which HTTP method to use). Only for type="submit"
<b>formnovalidate</b>	formnovalidate	Specifies that the form-data should not be validated on submission. Only for type="submit"
<b>formtarget</b>	_blank _self _parent _top <i>framename</i>	Specifies where to display the response after submitting the form. Only for type="submit"
<b>name</b>	<i>name</i>	Specifies a name for the button
<b>type</b>	button reset submit	Specifies the type of button
<b>value</b>	<i>text</i>	Specifies an initial value for the button

## Global Attributes

The `<button>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<button>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style buttons:

```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
}

.button1 {background-color: #4CAF50;} /* Green */
.button2 {background-color: #008CBA;} /* Blue */
</style>
</head>
<body>

<button class="button button1">Green</button>
<button class="button button2">Blue</button>

</body>
</html>
```

### Example:

Use CSS to style buttons (with hover effect):

```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
  border: none;
  color: white;
  padding: 16px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  transition-duration: 0.4s;
  cursor: pointer;
}

.button1 {
  background-color: white;
```



```
    color: black;
    border: 2px solid #4CAF50;
}

.button1:hover {
    background-color: #4CAF50;
    color: white;
}

.button2 {
    background-color: white;
    color: black;
    border: 2px solid #008CBA;
}

.button2:hover {
    background-color: #008CBA;
    color: white;
}

</style>
</head>
<body>

<button class="button button1">Green</button>
<button class="button button2">Blue</button>

</body>
</html>
```



## Default CSS Settings

None.

## 90.22 HTML <canvas> Tag

### Example:

Draw a red rectangle on the fly, and show it inside the <canvas> element:

```
<canvas id="myCanvas">
Your browser does not support the canvas tag.
</canvas>

<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
ctx.fillStyle = "#FF0000";
```

```
ctx.fillRect(0, 0, 80, 80);  
</script>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<canvas>` tag is used to draw graphics, on the fly, via scripting (usually JavaScript).

The `<canvas>` tag is transparent, and is only a container for graphics, you must use a script to actually draw the graphics.

Any text inside the `<canvas>` element will be displayed in browsers with JavaScript disabled and in browsers that do not support `<canvas>`.

## Tips and Notes

**Tip:** Learn more about the `<canvas>` element in our [HTML Canvas Tutorial](#).

**Tip:** For a complete reference of all the properties and methods, please visit our [HTML Canvas Reference](#).

## Attributes

Attribute	Value	Description
<code>height</code>	<i>pixels</i>	Specifies the height of the canvas. Default value is 150
<code>width</code>	<i>pixels</i>	Specifies the width of the canvas Default value is 300

## Global Attributes

The `<canvas>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<canvas>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Another `<canvas>` example:

```
<canvas id="myCanvas">
```

Your browser does not support the canvas tag.

```
</canvas>
```

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```

```
ctx.fillStyle = "red";
ctx.fillRect(20, 20, 75, 50);
//Turn transparency on
ctx.globalAlpha = 0.2;
ctx.fillStyle = "blue";
ctx.fillRect(50, 50, 75, 50);
ctx.fillStyle = "green";
ctx.fillRect(80, 80, 75, 50);
</script>
```



## Default CSS Settings

Most browsers will display the `<canvas>` element with the following default values:

### Example:

```
canvas {
  height: 150px;
  width: 300px;
}
```

## 90.23HTML <caption> Tag

### Example:

A table with a caption:

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100 ₹</td>
  </tr>
</table>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<caption>` tag defines a table caption.

The `<caption>` tag must be inserted immediately after the `<table>` tag.

**Tip:** By default, a table caption will be center-aligned above a table. However, the CSS properties `text-align` and `caption-side` can be used to align and place the caption.



## Global Attributes

The `<caption>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<caption>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Position table captions (with CSS):

```
<table>
  <caption style="text-align:right">My savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100 ₹</td>
  </tr>
</table>
<br>
```

```
<table>
  <caption style="caption-side:bottom">My savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100 ₹</td>
  </tr>
</table>
```

## Default CSS Settings

Most browsers will display the `<caption>` element with the following default values:

### Example:

```
caption {
  display: table-caption;
```

```
text-align: center;
}
```

## 90.24HTML <center> Tag

### Not Supported in HTML5.

The `<center>` tag was used in HTML4 to center-align text.



### What to Use Instead?

#### Example:

Center-align text (with CSS):

```
<html>
<head>
<style>
h1 {text-align: center;}
p {text-align: center;}
div {text-align: center;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<div>This is a div.</div>
</body>
</html>
```

## 90.25HTML <cite> Tag

#### Example:

Define the title of a work with the `<cite>` tag:

```
<p><cite>The Scream</cite> by Edward Munch. Painted in 1893.</p>
```

### Definition and Usage

The `<cite>` tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).

**Note:** A person's name is not the title of a work.

The text in the `<cite>` element usually renders in *italic*.



## Global Attributes

The `<cite>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<cite>` tag also supports the [Event Attributes in HTML](#).



## Default CSS Settings

Most browsers will display the `<cite>` element with the following default values:

Example

```
cite {
  font-style: italic;
}
```

## 90.26HTML `<code>` Tag

### Example:

Define some text as computer code in a document:

```
<p>The HTML <code>button</code>
```

 tag defines a clickable button.`</code>`

```
<p>The CSS <code>background-color</code>
```

 property defines the background color of an element.`</code>`

More "Try it Yourself" examples below.

## Definition and Usage

The `<code>` tag is used to define a piece of computer code. The content inside is displayed in the browser's default monospace font.

**Tip:** This tag is not deprecated. However, it is possible to achieve richer effect by using CSS (see example below).

Also look at:

Tag	Description
<code>&lt;samp&gt;</code>	Defines sample output from a computer program
<code>&lt;kbd&gt;</code>	Defines keyboard input
<code>&lt;var&gt;</code>	Defines a variable

```
<pre>
```

Defines preformatted text



## Global Attributes

The `<code>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<code>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style the `<code>` element:

```
<html>
<head>
<style>
code {
  font-family: Consolas,"courier new";
  color: crimson;
  background-color: #f1f1f1;
  padding: 2px;
  font-size: 105%;
}
</style>
</head>
<body>
```

```
<p>The HTML <code>button</code> tag defines a clickable button.</p>
```

```
<p>The CSS <code>background-color</code> property defines the background color of an element.</p>
```

```
</body>
</html>
```

## Default CSS Settings

Most browsers will display the `<code>` element with the following default values:

### Example:

```
code {
  font-family: monospace;
}
```

## 90.27HTML <col> Tag

### Example:

Set the background color of the three columns with the <colgroup> and <col> tags:

```
<table>
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>53 ₹</td>
  </tr>
</table>
```



More "Try it Yourself" examples below.

### Definition and Usage

The <col> tag specifies column properties for each column within a <colgroup> element.

The <col> tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.

### Attributes

Attribute	Value	Description
<b>span</b>	<i>number</i>	Specifies the number of columns a <col> element should span

### Global Attributes

The <col> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <col> tag also supports the [Event Attributes in HTML](#).

### More Examples

#### Example:

Align text in table columns (with CSS):

```
<table style="width:100%">
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td style="text-align:right">53 ₹</td>
  </tr>
  <tr>
    <td>2489604</td>
    <td>My first CSS</td>
    <td style="text-align:right">47 ₹</td>
  </tr>
</table>
```



### Example:

Vertical-align text in table columns (with CSS):

```
<table style="height:200px">
  <tr>
    <th>Month</th>
    <th style="vertical-align:bottom">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td style="vertical-align:bottom">100 ₹</td>
  </tr>
</table>
```

### Example:

Specify width of table columns (with CSS):

```
<table>
  <tr>
    <th style="width:130px">Month</th>
    <th style="width:80px">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100 ₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80 ₹</td>
  </tr>
</table>
```

## Default CSS Settings

Most browsers will display the `<col>` element with the following default values:

### Example:

```
col {  
  display: table-column;  
}
```



## 90.28HTML <colgroup> Tag

### Example:

Set the background color of the three columns with the `<colgroup>` and `<col>` tags:

```
<table>  
  <colgroup>  
    <col span="2" style="background-color:red">  
    <col style="background-color:yellow">  
  </colgroup>  
  <tr>  
    <th>ISBN</th>  
    <th>Title</th>  
    <th>Price</th>  
  </tr>  
  <tr>  
    <td>3476896</td>  
    <td>My first HTML</td>  
    <td>53₹</td>  
  </tr>  
</table>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<colgroup>` tag specifies a group of one or more columns in a table for formatting.

The `<colgroup>` tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.

**Note:** The `<colgroup>` tag must be a child of a `<table>` element, after any `<caption>` elements and before any `<thead>`, `<tbody>`, `<tfoot>`, and `<tr>` elements.

**Tip:** To define different properties to a column within a `<colgroup>`, use the `<col>` tag within the `<colgroup>` tag.

## Attributes

Attribute	Value	Description
-----------	-------	-------------

**span***number*

Specifies the number of columns a column group should span



## Global Attributes

The `<colgroup>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<colgroup>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Align text in table columns (with CSS):

```
<table style="width:100%">
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td style="text-align:right">53₹</td>
  </tr>
  <tr>
    <td>2489604</td>
    <td>My first CSS</td>
    <td style="text-align:right">47₹</td>
  </tr>
</table>
```

### Example:

Vertical-align text in table columns (with CSS):

```
<table style="height:200px">
  <tr>
    <th>Month</th>
    <th style="vertical-align:bottom">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td style="vertical-align:bottom">100₹</td>
  </tr>
</table>
```

## Example:

Specify width of a table column (with CSS):

```
<table>
  <tr>
    <th style="width:200px">Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100 ₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```



## Default CSS Settings

Most browsers will display the `<colgroup>` element with the following default values:

### Example:

```
colgroup {
  display: table-column-group;
}
```

## 90.29HTML <data> Tag

### Example:

The following example displays product names but also associates each name with a product number:

```
<ul>
  <li><data value="21053">Cherry Tomato</data></li>
  <li><data value="21054">Beef Tomato</data></li>
  <li><data value="21055">Snack Tomato</data></li>
</ul>
```

## Definition and Usage

The `<data>` tag is used to add a machine-readable translation of a given content.

This element provides both a machine-readable value for data processors, and a human-readable value for rendering in a browser.

**Tip:** If the content is time- or date-related, use the `<time>` element instead.

## Attributes

Attribute	Value	Description
<code>value</code>	<i>machine-readable format</i>	Specifies the machine-readable translation of the content of the element



## Global Attributes

The `<data>` tag also supports the [Global Attributes in HTML](#).

## 90.30HTML `<datalist>` Tag

### Example:

A datalist with pre-defined options (connected to an `<input>` element):

```
<label for="browser">Choose your browser from the list:</label>  
<input list="browsers" name="browser" id="browser">
```

```
<datalist id="browsers">  
  <option value="Edge">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>
```

## Definition and Usage

The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element.

The `<datalist>` tag is used to provide an "autocomplete" feature for `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.

The `<datalist>` element's `id` attribute must be equal to the `<input>` element's `list` attribute (this binds them together).

## Global Attributes

The `<datalist>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<datalist>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings



Most browsers will display the `<datalist>` element with the following default values:

```
datalist {  
  display: none;  
}
```



## 90.31HTML <dd> Tag

### Example:

A description list, with terms and descriptions:

```
<dl>  
  <dt>Coffee</dt>  
  <dd>Black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>White cold drink</dd>  
</dl>
```

### Definition and Usage

The `<dd>` tag is used to describe a term/name in a description list.

The `<dd>` tag is used in conjunction with `<dl>` (defines a description list) and `<dt>` (defines terms/names).

Inside a `<dd>` tag you can put paragraphs, line breaks, images, links, lists, etc.

### Global Attributes

The `<dd>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<dd>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

Most browsers will display the `<dd>` element with the following default values:

### Example:

```
dd {  
  display: block;  
  margin-left: 40px;  
}
```

## 90.32HTML <del> Tag

### Example:

A text with a deleted part, and a new, inserted part:

`<p>My favorite color is blue red!</p>`

More "Try it Yourself" examples below.

## Definition and Usage

The `<del>` tag defines text that has been deleted from a document. Browsers will usually strike a line through deleted text.



## Tips and Notes

**Tip:** Also look at the `<ins>` tag to markup inserted text.

## Attributes

Attribute	Value	Description
<code>cite</code>	URL	Specifies a URL to a document that explains the reason why the text was deleted/changed
<code>datetime</code>	YYYY-MM-DDThh:mm:ssTZD	Specifies the date and time of when the text was deleted/changed

## Global Attributes

The `<del>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<del>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style `<del>` and `<ins>`:

```
<html>
<head>
<style>
del {background-color: tomato;}
ins {background-color: yellow;}
</style>
</head>
<body>

<p>My favorite color is blue red!</p>

</body>
</html>
```

## Default CSS Settings

Most browsers will display the `<del>` element with the following default values:

**Example:**

```
del {  
  text-decoration: line-through;  
}
```



## 90.33HTML <details> Tag

**Example:**

Specify details that the user can open and close on demand:

```
<details>  
  <summary>Epcot Center</summary>  
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions,  
international pavilions, award-winning fireworks and seasonal special events.</p>  
</details>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<details>` tag specifies additional details that the user can open and close on demand.

The `<details>` tag is often used to create an interactive widget that the user can open and close. By default, the widget is closed. When open, it expands, and displays the content within.

Any sort of content can be put inside the `<details>` tag.

**Tip:** The `<summary>` tag is used in conjunction with `<details>` to specify a visible heading for the details.

### Attributes

Attribute	Value	Description
<code>open</code>	open	Specifies that the details should be visible (open) to the user

### Global Attributes

The `<details>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<details>` tag also supports the [Event Attributes in HTML](#).

### More Examples

**Example:**

Use CSS to style `<details>` and `<summary>`:

```
<html>
<style>
details > summary {
  padding: 4px;
  width: 200px;
  background-color: #eeeeee;
  border: none;
  box-shadow: 1px 1px 2px #bbbbbb;
  cursor: pointer;
}

details > p {
  background-color: #eeeeee;
  padding: 4px;
  margin: 0;
  box-shadow: 1px 1px 2px #bbbbbb;
}
</style>
<body>

<details>
  <summary>Epcot Center</summary>
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions,
international pavilions, award-winning fireworks and seasonal special events.</p>
</details>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<details>` element with the following default values:

```
details {
  display: block;
}
```

## 90.34HTML `<dfn>` Tag

### Example:

Mark up a term with `<dfn>`:

```
<p><dfn>HTML</dfn> is the standard markup language for creating web pages.</p>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<dfn>` tag stands for the "definition element", and it specifies a term that is going to be defined within the content.

The nearest parent of the `<dfn>` tag must also contain the definition/explanation for the term.

The term inside the `<dfn>` tag can be any of the following:

1. Just as the content of the `<dfn>` element:

#### Example:

```
<p><dfn>HTML</dfn> is the standard markup language for creating web pages.</p>
```

2. Or, with the title attribute added:

#### Example:

```
<p><dfn title="HyperText Markup Language">HTML</dfn> is the standard markup language for creating web pages.</p>
```

3. Or, with an `<abbr>` tag inside the `<dfn>` element:

#### Example:

```
<p><dfn><abbr title="HyperText Markup Language">HTML</abbr></dfn> is the standard markup language for creating web pages.</p>
```

4. Or, with the id attribute added. Then, whenever a term is used, it can refer back to the definition with an `<a>` tag:

#### Example:

```
<p><dfn id="html-def">HTML</dfn> is the standard markup language for creating web pages.</p>
```

```
<p>This is some text...</p>
```

```
<p>This is some text...</p>
```

```
<p>Learn <a href="#html-def">HTML</a> now.</p>
```

## Global Attributes

The `<dfn>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<dfn>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<dfn>` element with the following default values:

#### Example:

```
dfn {
  font-style: italic;
}
```



## 90.35HTML <dialog> Tag



### Example:

Using the <dialog> element:

```
<dialog open>This is an open dialog window</dialog>
```

### Definition and Usage

The <dialog> tag defines a dialog box or subwindow.

The <dialog> element makes it easy to create popup dialogs and modals on a web page.

\* Not supported by default, but can be enabled in about:config (set dom.dialog\_element.enabled to true).

### Attributes

Attribute	Value	Description
<a href="#">open</a>	open	Specifies that the dialog element is active and that the user can interact with it

### Global Attributes

The <dialog> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <dialog> tag also supports the [Event Attributes in HTML](#).

## 90.36HTML <dir> Tag

### Not Supported in HTML5.

The <dir> tag was used in HTML 4 to list directory titles.

### What to Use Instead?

#### Example:

Use <ul> to create a directory list:

```
<ul>  
  <li>html</li>  
  <li>xhtml</li>  
  <li>css</li>  
</ul>
```

### Example:

Reduce line-height in a list (with CSS):

```
<ul style="line-height:80%">
  <li>html</li>
  <li>xhtml</li>
  <li>css</li>
</ul>
```



**Tip:** In our CSS tutorial you can find more details about [styling lists](#).

## 90.37HTML <div> Tag

### Example:

A <div> section in a document that is styled with CSS:

```
<html>
<head>
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;
}
</style>
</head>
<body>
<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>
</body>
</html>
```

More "Try it Yourself" examples below.

### Definition and Usage

The <div> tag defines a division or a section in an HTML document.

The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.

The <div> tag is easily styled by using the class or id attribute.

Any sort of content can be put inside the <div> tag!

**Note:** By default, browsers always place a line break before and after the <div> element.

## Global Attributes

The `<div>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<div>` tag also supports the [Event Attributes in HTML](#).



## Default CSS Settings

Most browsers will display the `<div>` element with the following default values:

### Example:

```
div {  
  display: block;  
}
```

## 90.38HTML <dl> Tag

### Example:

A description list, with terms and descriptions:

```
<dl>  
  <dt>Coffee</dt>  
  <dd>Black hot drink</dd>  
  <dt>Milk</dt>  
  <dd>White cold drink</dd>  
</dl>
```

## Definition and Usage

The `<dl>` tag defines a description list.

The `<dl>` tag is used in conjunction with `<dt>` (defines terms/names) and `<dd>` (describes each term/name).

## Global Attributes

The `<dl>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<dl>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings



Most browsers will display the `<dl>` element with the following default values:

**Example:**

```
dl {
  display: block;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-left: 0;
  margin-right: 0;
}
```



## 90.39HTML <dt> Tag

**Example:**

A description list, with terms and descriptions:

```
<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>
```

### Definition and Usage

The `<dt>` tag defines a term/name in a description list.

The `<dt>` tag is used in conjunction with `<dl>` (defines a description list) and `<dd>` (describes each term/name).

### Global Attributes

The `<dt>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<dt>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

Most browsers will display the `<dt>` element with the following default values:

**Example:**

```
dt {
  display: block;
}
```

## 90.40HTML <em> Tag

### Example:

Mark up emphasized text in a document:

```
<p>You <em>have</em> to hurry up!</p>
<p>We <em>cannot</em> live like this.</p>
```



### Definition and Usage

The `<em>` tag is used to define emphasized text. The content inside is typically displayed in *italic*.

A screen reader will pronounce the words in `<em>` with an emphasis, using verbal stress.

### Global Attributes

The `<em>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<em>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

Most browsers will display the `<em>` element with the following default values:

#### Example:

```
em {
  font-style: italic;
}
```

## 90.41HTML <embed> Tag

### Example:

An embedded image:

```
<embed type="image/jpeg" src="pic_trulli.jpg" width="300" height="200">
```

Example

An embedded HTML page:

```
<embed type="text/html" src="snippet.html" width="500" height="200">
```

Example

An embedded video:

```
<embed type="video/webm" src="video.mp4" width="400" height="300">
```

## Definition and Usage

The `<embed>` tag defines a container for an external resource, such as a web page, a picture, a media player, or a plug-in application.

### Warning:

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

### Suggestion:

To display a picture, it is better to use the `<img>` tag.

To display HTML, it is better to use the `<iframe>` tag.

To display video or audio, it is better to use the `<video>` and `<audio>` tags.

## Attributes

Attribute	Value	Description
<code>height</code>	<i>pixels</i>	Specifies the height of the embedded content
<code>src</code>	<i>URL</i>	Specifies the address of the external file to embed
<code>type</code>	<i>media_type</i>	Specifies the media type of the embedded content
<code>width</code>	<i>pixels</i>	Specifies the width of the embedded content

## Global Attributes

The `<embed>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<embed>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<embed>` element with the following default values:

```
embed:focus {  
  outline: none;  
}
```

## 90.42HTML `<fieldset>` Tag

### Example:

Group related elements in a form:

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="birthday">Birthday:</label>
    <input type="date" id="birthday" name="birthday"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



More "Try it Yourself" examples below.

## Definition and Usage

The `<fieldset>` tag is used to group related elements in a form.

The `<fieldset>` tag draws a box around the related elements.

## Tips and Notes

**Tip:** The `<legend>` tag is used to define a caption for the `<fieldset>` element.

## Attributes

Attribute	Value	Description
<b>disabled</b>	disabled	Specifies that a group of related form elements should be disabled
<b>form</b>	<i>form_id</i>	Specifies which form the fieldset belongs to
<b>name</b>	<i>text</i>	Specifies a name for the fieldset

## Global Attributes

The `<fieldset>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<fieldset>` tag also supports the [Event Attributes in HTML](#).

## More Examples

## Example:

Use CSS to style `<fieldset>` and `<legend>`:

```
<html>
<head>
<style>
fieldset {
  background-color: #eeeeee;
}

legend {
  background-color: gray;
  color: white;
  padding: 5px 10px;
}

input {
  margin: 5px;
}
</style>
</head>
<body>

<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="birthday">Birthday:</label>
    <input type="date" id="birthday" name="birthday"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<fieldset>` element with the following default values:

```
fieldset {
  display: block;
  margin-left: 2px;
  margin-right: 2px;
  padding-top: 0.35em;
```

```
padding-bottom: 0.625em;
padding-left: 0.75em;
padding-right: 0.75em;
border: 2px groove (internal value);
}
```



## 90.43HTML <figcaption> Tag

### Example:

Use a <figure> element to mark up a photo in a document, and a <figcaption> element to define a caption for the photo:

```
<figure>
  
  <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>
```

More "Try it Yourself" examples below.

### Definition and Usage

The <figcaption> tag defines a caption for a <figure> element.

The <figcaption> element can be placed as the first or last child of the <figure> element.

### Global Attributes

The <figcaption> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <figcaption> tag also supports the [Event Attributes in HTML](#).

### More Examples

#### Example:

Use CSS to style <figure> and <figcaption>:

```
<html>
<head>
<style>
figure {
  border: 1px #cccccc solid;
  padding: 4px;
  margin: auto;
}

figcaption {
  background-color: black;
  color: white;
  font-style: italic;
}
```

```
padding: 2px;
text-align: center;
}
</style>
</head>
<body>
<figure>
  
  <figcaption>Fig.1 - Trulli, Puglia, Italy</figcaption>
</figure>
</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<figcaption>` element with the following default values:

```
figcaption {
  display: block;
}
```

## 90.44HTML `<figure>` Tag

### Example:

Use a `<figure>` element to mark up a photo in a document, and a `<figcaption>` element to define a caption for the photo:

```
<figure>
  
  <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<figure>` tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

While the content of the `<figure>` element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document.

**Tip:** The `<figcaption>` element is used to add a caption for the `<figure>` element.

## Global Attributes

The `<figure>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<figure>` tag also supports the [Event Attributes in HTML](#).

## More Examples

Example

Use CSS to style `<figure>` and `<figcaption>`:

```
<html>
<head>
<style>
figure {
  border: 1px #cccccc solid;
  padding: 4px;
  margin: auto;
}

figcaption {
  background-color: black;
  color: white;
  font-style: italic;
  padding: 2px;
  text-align: center;
}
</style>
</head>
<body>

<figure>
  
  <figcaption>Fig.1 - Trulli, Puglia, Italy</figcaption>
</figure>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<figure>` element with the following default values:

### Example:

```
figure {
  display: block;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-left: 40px;
```



```
margin-right: 40px;
}
```

## 90.45HTML <font> Tag

### Not Supported in HTML5.

The <font> tag was used in HTML 4 to specify the font face, font size, and color of text.



### What to Use Instead?

#### Example:

Set the color of text (with CSS):

```
<p style="color:red">This is a paragraph.</p>
<p style="color:blue">This is another paragraph.</p>
```

#### Example:

Set the font of text (with CSS):

```
<p style="font-family:verdana">This is a paragraph.</p>
<p style="font-family:'Courier New'">This is another paragraph.</p>
```

#### Example:

Set the size of text (with CSS):

```
<p style="font-size:30px">This is a paragraph.</p>
<p style="font-size:11px">This is another paragraph.</p>
```

In our CSS tutorial you can find more information about [CSS Text](#) and [CSS Fonts](#).

## 90.46HTML <footer> Tag

#### Example:

A footer section in a document:

```
<footer>
  <p>Author: Hege Refsnes</p>
  <p><a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<footer>` tag defines a footer for a document or section.

A `<footer>` element typically contains:

- authorship information
- copyright information
- contact information
- sitemap
- back to top links
- related documents



You can have several `<footer>` elements in one document.

## Tips and Notes

**Tip:** Contact information inside a `<footer>` element should go inside an `<address>` tag.

## Global Attributes

The `<footer>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<footer>` tag also supports the Event Attributes in HTML.

## More Examples

Example

Use CSS to style `<footer>`:

```
<html>
<head>
<style>
footer {
  text-align: center;
  padding: 3px;
  background-color: DarkSalmon;
  color: white;
}
</style>
</head>
<body>

<footer>
  <p>Author: Hege Refsnes<br>
  <a href="mailto:hege@example.com">hege@example.com</a></p>
</footer>

</body>
</html>
```

## Default CSS Settings

Most browsers will display the `<footer>` element with the following default values:

```
footer {
  display: block;
}
```



## 90.47HTML <form> Tag

### Example:

An HTML form with two input fields and one submit button:

```
<form action="/action_page.php" method="get">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<form>` tag is used to create an HTML form for user input.

The `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<textarea>`
- `<button>`
- `<select>`
- `<option>`
- `<optgroup>`
- `<fieldset>`
- `<label>`
- `<output>`

## Attributes

Attribute	Value	Description
<b>accept-charset</b>	<i>character_set</i>	Specifies the character encodings that are to be used for the form submission
<b>action</b>	<i>URL</i>	Specifies where to send the form-data when a form is submitted
<b>autocomplete</b>	on off	Specifies whether a form should have autocomplete on or off

<b>enctype</b>	application/x-www-form-urlencoded multipart/form-data text/plain	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
<b>method</b>	get post	Specifies the HTTP method to use when sending form-data
<b>name</b>	text	Specifies the name of a form
<b>novalidate</b>	novalidate	Specifies that the form should not be validated when submitted
<b>rel</b>	external help license  nofollow noopener noreferrer opener prev search	Specifies the relationship between a linked resource and the current document
<b>target</b>	_blank _self _parent _top	Specifies where to display the response that is received after submitting the form



## Global Attributes

The `<form>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<form>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

An HTML form with checkboxes:

```
<form action="/action_page.php" method="get">
  <input type="checkbox" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" name="vehicle3" value="Boat" checked>
  <label for="vehicle3"> I have a boat</label><br><br>
  <input type="submit" value="Submit">
</form>
```

### Example:

An HTML form with radiobuttons:

```
<form action="/action_page.php" method="get">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS" checked="checked">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
  <input type="submit" value="Submit">
</form>
```



## Default CSS Settings

Most browsers will display the `<form>` element with the following default values:

### Example:

```
form {
  display: block;
  margin-top: 0em;
}
```

## 90.48 HTML `<frame>` Tag

### Not Supported in HTML5.

The `<frame>` tag was used in HTML 4 to define one particular window (frame) within a `<frameset>`.

## What to Use Instead?

### Example:

Use the `<iframe>` tag to embed another document within the current HTML document:

```
<iframe src="https://www.bintr.com"></iframe>
```

## 90.49 HTML `<frameset>` Tag

### Not Supported in HTML5.

The `<frameset>` tag was used in HTML 4 to define a frameset.

## What to Use Instead?

### Example:

Use the `<iframe>` tag to embed another document within the current HTML document:

```
<iframe src="https://www.bintr.com"></iframe>
```



## 90.50HTML <h1> to <h6> Tags

### Example:

The six different HTML headings:

```
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

```
<h4>This is heading 4</h4>
```

```
<h5>This is heading 5</h5>
```

```
<h6>This is heading 6</h6>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<h1>` to `<h6>` tags are used to define HTML headings.

`<h1>` defines the most important heading. `<h6>` defines the least important heading.

**Note:** Only use one `<h1>` per page - this should represent the main heading/subject for the whole page. Also, do not skip heading levels - start with `<h1>`, then use `<h2>`, and so on.

## Global Attributes

The `<h1>` to `<h6>` tags also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<h1>` to `<h6>` tags also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Set the background color and text color of headings (with CSS):

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

```
<h2 style="color:Tomato;">Hello World</h2>
```

## Example:

Set the alignment of headings (with CSS):

```
<h1 style="text-align:center">This is heading 1</h1>
<h2 style="text-align:left">This is heading 2</h2>
<h3 style="text-align:right">This is heading 3</h3>
<h4 style="text-align:justify">This is heading 4</h4>
```



## Default CSS Settings

Most browsers will display the `<h1>` element with the following default values:

### Example:

```
h1 {
  display: block;
  font-size: 2em;
  margin-top: 0.67em;
  margin-bottom: 0.67em;
  margin-left: 0;
  margin-right: 0;
  font-weight: bold;
}
```

Most browsers will display the `<h2>` element with the following default values:

### Example:

```
h2 {
  display: block;
  font-size: 1.5em;
  margin-top: 0.83em;
  margin-bottom: 0.83em;
  margin-left: 0;
  margin-right: 0;
  font-weight: bold;
}
```

Most browsers will display the `<h3>` element with the following default values:

### Example:

```
h3 {
  display: block;
  font-size: 1.17em;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-left: 0;
  margin-right: 0;
  font-weight: bold;
}
```

Most browsers will display the `<h4>` element with the following default values:

### Example:

```
h4 {
  display: block;
  font-size: 1em;
  margin-top: 1.33em;
  margin-bottom: 1.33em;
  margin-left: 0;
  margin-right: 0;
  font-weight: bold;
}
```



Most browsers will display the `<h5>` element with the following default values:

### Example:

```
h5 {
  display: block;
  font-size: .83em;
  margin-top: 1.67em;
  margin-bottom: 1.67em;
  margin-left: 0;
  margin-right: 0;
  font-weight: bold;
}
```

Most browsers will display the `<h6>` element with the following default values:

### Example:

```
h6 {
  display: block;
  font-size: .67em;
  margin-top: 2.33em;
  margin-bottom: 2.33em;
  margin-left: 0;
  margin-right: 0;
  font-weight: bold;
}
```

## 90.51HTML `<head>` Tag

### Example:

A simple HTML document, with a `<title>` tag inside the head section:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Title of the document</title>
</head>
<body>
<h1>This is a heading</h1>
```



```
<p>This is a paragraph.</p>
</body>
</html>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<head>` element is a container for metadata (data about data) and is placed between the `<html>` tag and the `<body>` tag.

Metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

The following elements can go inside the `<head>` element:

- `<title>` (required in every HTML document)
- `<style>`
- `<base>`
- `<link>`
- `<meta>`
- `<script>`
- `<noscript>`

## Global Attributes

The `<head>` tag also supports the Global Attributes in HTML.

## More Examples

### Example:

The `<base>` tag (specifies a default URL and target for all links on a page) goes inside `<head>`:

```
<html>
<head>
  <base href="https://www.bintr.com/" target="_blank">
</head>
<body>

<a href="tags/tag_base.asp">HTML base Tag</a>
</body>
</html>
```

### Example:

The `<style>` tag (adds style information to a page) goes inside `<head>`:

```
<html>
<head>
  <style>
    h1 {color:red;}
</head>
```

```
p {color:blue;}
</style>
</head>
<body>
<h1>A heading</h1>
<p>A paragraph.</p>
</body>
</html>
```



### Example:

The <link> tag (links to an external style sheet) goes inside <head>:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<h1>I am formatted with a linked style sheet</h1>
<p>Me too!</p>
</body>
</html>
```

### Default CSS Settings

Most browsers will display the <head> element with the following default values:

```
head {
  display: none;
}
```

## 90.52HTML <header> Tag

### Example:

A header for an <article>:

```
<article>
  <header>
    <h1>A heading here</h1>
    <p>Posted by John Doe</p>
    <p>Some additional information here</p>
  </header>
  <p>Lorem Ipsum dolor set amet....</p>
</article>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<header>` element represents a container for introductory content or a set of navigational links.

A `<header>` element typically contains:

- one or more heading elements (`<h1>` - `<h6>`)
- logo or icon
- authorship information



**Note:** You can have several `<header>` elements in one HTML document. However, `<header>` cannot be placed within a `<footer>`, `<address>` or another `<header>` element.

## Global Attributes

The `<header>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<header>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

A page header:

```
<header>  
  <h1>Main page heading here</h1>  
  <p>Posted by John Doe</p>  
</header>
```

## Default CSS Settings

Most browsers will display the `<header>` element with the following default values:

```
header {  
  display: block;  
}
```

## 90.53HTML `<hr>` Tag

### Example:

Use the `<hr>` tag to define thematic changes in the content:

```
<h1>The Main Languages of the Web</h1>
```

```
<p>HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page, and consists of a series of elements. HTML elements tell the browser how to display the content.</p>
```

<hr>

<p>CSS is a language that describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work, because it can control the layout of multiple web pages all at once.</p>



<hr>

<p>JavaScript is the programming language of HTML and the Web. JavaScript can change HTML content and attribute values. JavaScript can change CSS. JavaScript can hide and show HTML elements, and more.</p>

More "Try it Yourself" examples below.

## Definition and Usage

The <hr> tag defines a thematic break in an HTML page (e.g. a shift of topic).

The <hr> element is most often displayed as a horizontal rule that is used to separate content (or define a change) in an HTML page.

## Global Attributes

The <hr> tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The <hr> tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Align a <hr> element (with CSS):

```
<hr style="width:50%;text-align:left;margin-left:0">
```

### Example:

A noshaded <hr> (with CSS):

```
<hr style="height:2px;border-width:0;color:gray;background-color:gray">
```

### Example:

Set the height of a <hr> element (with CSS):

```
<hr style="height:30px">
```

### Example:

Set the width of a <hr> element (with CSS):

```
<hr style="width:50%">
```

## Default CSS Settings

Most browsers will display the `<hr>` element with the following default values:

### Example:

```
hr {
  display: block;
  margin-top: 0.5em;
  margin-bottom: 0.5em;
  margin-left: auto;
  margin-right: auto;
  border-style: inset;
  border-width: 1px;
}
```



## 90.54HTML <html> Tag

### Example:

A simple HTML document:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Title of the document</title>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

### Definition and Usage

The `<html>` tag represents the root of an HTML document.

The `<html>` tag is the container for all other HTML elements (except for the `<!DOCTYPE>` tag).

**Note:** You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

### Attributes

Attribute	Value	Description
<code>xmlns</code>	<code>http://www.bintr.online/1999/xhtml</code>	Specifies the XML namespace attribute (If you need your content to conform to XHTML)

## Global Attributes

The `<html>` tag also supports the [Global Attributes in HTML](#).



## Default CSS Settings

Most browsers will display the `<html>` element with the following default values:

```
html {  
  display: block;  
}
```

```
html:focus {  
  outline: none;  
}
```

## 90.55HTML `<i>` Tag

### Example:

Mark up text that is set off from the normal prose in a document:

```
<p><i>Lorem ipsum</i> is the most popular filler text in history.</p>
```

```
<p>The <i>RMS Titanic</i>, a luxury steamship, sank on April 15, 1912 after striking  
an iceberg.</p>
```

## Definition and Usage

The `<i>` tag defines a part of text in an alternate voice or mood. The content inside is typically displayed in *italic*.

The `<i>` tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

Use the `<i>` element only when there is not a more appropriate semantic element, such as:

- `<em>` (emphasized text)
- `<strong>` (important text)
- `<mark>` (marked/highlighted text)
- `<cite>` (the title of a work)
- `<dfn>` (a definition term)

## Global Attributes

The `<i>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<i>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<i>` element with the following default values:

### Example:

```
i {
  font-style: italic;
}
```



## 90.56HTML <iframe> Tag

### Example:

An inline frame is marked up as follows:

```
<iframe src="https://www.bintr.com" title="Bintr Free Online Web Tutorials"></iframe>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<iframe>` tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

**Tip:** Use CSS to style the `<iframe>` (see example below).

**Tip:** It is a good practice to always include a title attribute for the `<iframe>`. This is used by screen readers to read out what the content of the `<iframe>` is.

## Attributes

Attribute	Value	Description
<b>allow</b>		Specifies a feature policy for the <code>&lt;iframe&gt;</code>
<b>allowfullscreen</b>	true false	Set to true if the <code>&lt;iframe&gt;</code> can activate fullscreen mode by calling the <code>requestFullscreen()</code> method
<b>allowpaymentrequest</b>	true false	Set to true if a cross-origin <code>&lt;iframe&gt;</code> should be allowed to invoke the Payment Request API
<b>height</b>	<i>pixels</i>	Specifies the height of an <code>&lt;iframe&gt;</code> . Default height is 150 pixels
<b>loading</b>	eager lazy	Specifies whether a browser should load an <code>iframe</code> immediately or to defer loading of <code>iframes</code> until some conditions are met
<b>name</b>	<i>text</i>	Specifies the name of an <code>&lt;iframe&gt;</code>

<b>referrerpolicy</b>	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin-when-cross-origin unsafe-url	Specifies which referrer information to send when fetching the iframe
<b>sandbox</b>	allow-forms allow-pointer-lock allow-popups allow-same-origin allow-scripts allow-top-navigation	Enables an extra set of restrictions for the content in an <iframe>
<b>src</b>	<i>URL</i>	Specifies the address of the document to embed in the <iframe>
<b>srcdoc</b>	<i>HTML_code</i>	Specifies the HTML content of the page to show in the <iframe>
<b>width</b>	<i>pixels</i>	Specifies the width of an <iframe>. Default width is 300 pixels

## Global Attributes

The <iframe> tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The <iframe> tag also supports the [Event Attributes in HTML](#).

## More Examples

### 1 Example:

Add and remove iframe borders (with CSS):

```
<iframe src="/default.asp" width="100%" height="300" style="border:1px solid black;">
</iframe>
```

```
<iframe src="/default.asp" width="100%" height="300" style="border:none;">
</iframe>
```

## Default CSS Settings

Most browsers will display the <iframe> element with the following default values:

```
iframe:focus {
  outline: none;
}
```



```
iframe[seamless] {
  display: block;
}
```



## 90.57 HTML <img> Tag

### Example:

How to insert an image:

```

```

More "Try it Yourself" examples below.

### Definition and Usage

The `<img>` tag is used to embed an image in an HTML page.

Images are not technically inserted into a web page; images are linked to web pages. The `<img>` tag creates a holding space for the referenced image.

The `<img>` tag has two required attributes:

- `src` - Specifies the path to the image
- `alt` - Specifies an alternate text for the image, if the image for some reason cannot be displayed

**Note:** Also, always specify the width and height of an image. If width and height are not specified, the page might flicker while the image loads.

**Tip:** To link an image to another document, simply nest the `<img>` tag inside an `<a>` tag (see example below).

### Attributes

Attribute	Value	Description
<b>alt</b>	<i>text</i>	Specifies an alternate text for an image
<b>crossorigin</b>	anonymous use-credentials	Allow images from third-party sites that allow cross-origin access to be used with canvas
<b>height</b>	<i>pixels</i>	Specifies the height of an image
<b>ismap</b>	ismap	Specifies an image as a server-side image map
<b>loading</b>	eager lazy	Specifies whether a browser should load an image immediately or to defer loading of images until some conditions are met
<b>longdesc</b>	<i>URL</i>	Specifies a URL to a detailed description of an image
<b>referrerpolicy</b>	no-referrer no-referrer-when-downgrade origin	Specifies which referrer information to use when fetching an image

	origin-when-cross-origin unsafe-url	
<b>sizes</b>	<i>sizes</i>	Specifies image sizes for different page layouts
<b>src</b>	<i>URL</i>	Specifies the path to the image
<b>srcset</b>	<i>URL-list</i>	Specifies a list of image files to use in different situations
<b>usemap</b>	<i>#mapname</i>	Specifies an image as a client-side image map
<b>width</b>	<i>pixels</i>	Specifies the width of an image

## Global Attributes

The `<img>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<img>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

Align image (with CSS):

```





```

### Example:

Add image border (with CSS):

```

```

### Example:

Add left and right margins to image (with CSS):

```

```

### Example:

Add top and bottom margins to image (with CSS):

```

```

### Example:

How to insert images from another folder or from another web site:

```
  

```



### Example:

How to add a hyperlink to an image:

```
<a href="https://www.bintr.com">  
  
</a>
```

### Example:

How to create an image map, with clickable regions. Each region is a hyperlink:

```
  
  
<map name="workmap">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
  <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">  
</map>
```

## Default CSS Settings

Most browsers will display the `<img>` element with the following default values:

### Example:

```
img {  
  display: inline-block;  
}
```

## 90.58HTML <input> Tag

### Example:

An HTML form with three input fields; two text fields and one submit button:

```
<form action="/action_page.php">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>
```

```
<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname"><br><br>
<input type="submit" value="Submit">
</form>
```



## Definition and Usage

The `<input>` tag specifies an input field where the user can enter data.

The `<input>` element is the most important form element.

The `<input>` element can be displayed in several ways, depending on the type attribute.

The different input types are as follows:

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">` (default value)
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

Look at the type attribute to see examples for each input type!

## Tips and Notes

Tip: Always use the `<label>` tag to define labels for `<input type="text">`, `<input type="checkbox">`, `<input type="radio">`, `<input type="file">`, and `<input type="password">`.

## Attributes

Attribute	Value	Description
<b>accept</b>	<i>file_extension</i> audio/* video/* image/* <i>media_type</i>	Specifies a filter for what file types the user can pick from the file input dialog box (only for type="file")
<b>alt</b>	<i>text</i>	Specifies an alternate text for images (only for type="image")
<b>autocomplete</b>	on off	Specifies whether an <input> element should have autocomplete enabled
<b>autofocus</b>	autofocus	Specifies that an <input> element should automatically get focus when the page loads
<b>checked</b>	checked	Specifies that an <input> element should be pre-selected when the page loads (for type="checkbox" or type="radio")
<b>dirname</b>	<i>inputname.dir</i>	Specifies that the text direction will be submitted
<b>disabled</b>	disabled	Specifies that an <input> element should be disabled
<b>form</b>	<i>form_id</i>	Specifies the form the <input> element belongs to
<b>formaction</b>	<i>URL</i>	Specifies the URL of the file that will process the input control when the form is submitted (for type="submit" and type="image")
<b>formenctype</b>	application/x-www-form-urlencoded multipart/form-data text/plain	Specifies how the form-data should be encoded when submitting it to the server (for type="submit" and type="image")
<b>formmethod</b>	get post	Defines the HTTP method for sending data to the action URL (for type="submit" and type="image")
<b>formnovalidate</b>	formnovalidate	Defines that form elements should not be validated when submitted
<b>formtarget</b>	_blank _self _parent _top <i>framename</i>	Specifies where to display the response that is received after submitting the form (for type="submit" and type="image")
<b>height</b>	<i>pixels</i>	Specifies the height of an <input> element (only for type="image")
<b>list</b>	<i>datalist_id</i>	Refers to a <datalist> element that contains pre-defined options for an <input> element
<b>max</b>	<i>number</i> <i>date</i>	Specifies the maximum value for an <input> element
<b>maxlength</b>	<i>number</i>	Specifies the maximum number of characters allowed in an <input> element
<b>min</b>	<i>number</i> <i>date</i>	Specifies a minimum value for an <input> element
<b>minlength</b>	<i>number</i>	Specifies the minimum number of characters required in an <input> element
<b>multiple</b>	multiple	Specifies that a user can enter more than one value in an <input> element

<b>name</b>	<i>text</i>	Specifies the name of an <input> element
<b>pattern</b>	<i>regexp</i>	Specifies a regular expression that an <input> element's value is checked against
<b>placeholder</b>	<i>text</i>	Specifies a short hint that describes the expected value of an <input> element
<b>readonly</b>	readonly	Specifies that an input field is read-only
<b>required</b>	required	Specifies that an input field must be filled out before submitting the form
<b>size</b>	<i>number</i>	Specifies the width, in characters, of an <input> element
<b>src</b>	<i>URL</i>	Specifies the URL of the image to use as a submit button (only for type="image")
<b>step</b>	<i>number</i> any	Specifies the interval between legal numbers in an input field
<b>type</b>	button checkbox color date datetime-local email file hidden image month number password radio range reset search submit tel text time url week	Specifies the type <input> element to display
<b>value</b>	<i>text</i>	Specifies the value of an <input> element
<b>width</b>	<i>pixels</i>	Specifies the width of an <input> element (only for type="image")

## Global Attributes

The <input> tag also supports the Global Attributes in HTML.

## Event Attributes

The <input> tag also supports the Event Attributes in HTML.

## Default CSS Settings

None.



## 90.59HTML <ins> Tag

### Example:

A text with a deleted part, and a new, inserted part:

```
<p>My favorite color is <del>blue</del> <ins>red</ins>!</p>
```

## Definition and Usage

The `<ins>` tag defines a text that has been inserted into a document. Browsers will usually underline inserted text.

**Tip:** Also look at the `<del>` tag to markup deleted text.

## Attributes

Attribute	Value	Description
<code>cite</code>	URL	Specifies a URL to a document that explains the reason why the text was inserted/changed
<code>datetime</code>	YYYY-MM-DDThh:mm:ssTZD	Specifies the date and time when the text was inserted/changed

## Global Attributes

The `<ins>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<ins>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style `<del>` and `<ins>`:

```
<html>
<head>
<style>
del {background-color: tomato;}
ins {background-color: yellow;}

```

```
</style>
</head>
<body>
<p>My favorite color is <del>blue</del> <ins>red</ins>!</p>
</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<ins>` element with the following default values:

Example

```
ins {
  text-decoration: underline;
}
```

## 90.60 HTML `<kbd>` Tag

### Example:

Define some text as keyboard input in a document:

```
<p>Press <kbd>Ctrl</kbd> + <kbd>C</kbd> to copy text (Windows).</p>
```

```
<p>Press <kbd>Cmd</kbd> + <kbd>C</kbd> to copy text (Mac OS).</p>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<kbd>` tag is used to define keyboard input. The content inside is displayed in the browser's default monospace font.

**Tip:** This tag is not deprecated. However, it is possible to achieve richer effect by using CSS (see example below).

Also look at:

Tag	Description
<code>&lt;code&gt;</code>	Defines a piece of computer code
<code>&lt;samp&gt;</code>	Defines sample output from a computer program
<code>&lt;var&gt;</code>	Defines a variable
<code>&lt;pre&gt;</code>	Defines preformatted text

## Global Attributes

The `<kbd>` tag also supports the [Global Attributes in HTML](#).



## Event Attributes

The `<kbd>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style the `<kbd>` element:

```
<html>
<head>
<style>
kbd {
  border-radius: 2px;
  padding: 2px;
  border: 1px solid black;
}
</style>
</head>
<body>

<p>Press <code>Ctrl</code> + <code>C</code> to copy text (Windows).</p>
<p>Press <code>Cmd</code> + <code>C</code> to copy text (Mac OS).</p>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<kbd>` element with the following default values:

### Example:

```
kbd {
  font-family: monospace;
}
```

## 90.61HTML <label> Tag

### Example:

Three radio buttons with labels:

```
<form action="/action_page.php">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
```

```
<input type="submit" value="Submit">
</form>
```



## Definition and Usage

The `<label>` tag defines a label for several elements:

- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="search">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`
- `<meter>`
- `<progress>`
- `<select>`
- `<textarea>`

Proper use of labels with the elements above will benefit:

- Screen reader users (will read out loud the label, when the user is focused on the element)
- Users who have difficulty clicking on very small regions (such as checkboxes) - because when a user clicks the text within the `<label>` element, it toggles the input (this increases the hit area).

## Tips and Notes

Tip: The `for` attribute of `<label>` must be equal to the `id` attribute of the related element to bind them together. A label can also be bound to an element by placing the element inside the `<label>` element.

## Attributes

Attribute	Value	Description
<b>for</b>	<i>element_id</i>	Specifies the id of the form element the label should be bound to
<b>form</b>	<i>form_id</i>	Specifies which form the label belongs to

## Global Attributes

The `<label>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<label>` tag also supports the Event Attributes in HTML.



## Default CSS Settings

Most browsers will display the `<label>` element with the following default values:

### Example:

```
label {  
  cursor: default;  
}
```

## 90.62HTML `<legend>` Tag

### Example:

Group related elements in a form:

```
<form action="/action_page.php">  
  <fieldset>  
    <legend>Personalia:</legend>  
    <label for="fname">First name:</label>  
    <input type="text" id="fname" name="fname"><br><br>  
    <label for="lname">Last name:</label>  
    <input type="text" id="lname" name="lname"><br><br>  
    <label for="email">Email:</label>  
    <input type="email" id="email" name="email"><br><br>  
    <label for="birthday">Birthday:</label>  
    <input type="date" id="birthday" name="birthday"><br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<legend>` tag defines a caption for the `<fieldset>` element.

## Global Attributes

The `<legend>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<legend>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Let the fieldset caption float to the right (with CSS):

```
<form action="/action_page.php">
  <fieldset>
    <legend style="float:right">Personalia:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="birthday">Birthday:</label>
    <input type="date" id="birthday" name="birthday"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

### Example:

Use CSS to style `<fieldset>` and `<legend>`:

```
<html>
<head>
<style>
fieldset {
  background-color: #eeeeee;
}

legend {
  background-color: gray;
  color: white;
  padding: 5px 10px;
}

input {
  margin: 5px;
}
</style>
</head>
<body>

<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label>
```



```
<input type="text" id="fname" name="fname"><br><br>
<label for="lname">Last name:</label>
<input type="text" id="lname" name="lname"><br><br>
<label for="email">Email:</label>
<input type="email" id="email" name="email"><br><br>
<label for="birthday">Birthday:</label>
<input type="date" id="birthday" name="birthday"><br><br>
<input type="submit" value="Submit">
</fieldset>
</form>

</body>
</html>
```



## Default CSS Settings

Most browsers will display the `<legend>` element with the following default values:

### Example:

```
legend {
  display: block;
  padding-left: 2px;
  padding-right: 2px;
  border: none;
}
```

## 90.63HTML `<li>` Tag

### Example:

One ordered (`<ol>`) and one unordered (`<ul>`) HTML list:

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<li>` tag defines a list item.

The `<li>` tag is used inside ordered lists(`<ol>`), unordered lists (`<ul>`), and in menu lists (`<menu>`).

In `<ul>` and `<menu>`, the list items will usually be displayed with bullet points.

In `<ol>`, the list items will usually be displayed with numbers or letters.



## Attributes

Attribute	Value	Description
<code>value</code>	<i>number</i>	Only for <code>&lt;ol&gt;</code> lists. Specifies the start value of a list item. The following list items will increment from that number

## Global Attributes

The `<li>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<li>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use of the `value` attribute in an ordered list:

```
<ol>
  <li value="100">Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
  <li>Water</li>
  <li>Juice</li>
  <li>Beer</li>
</ol>
```

### Example:

Set different list style types (with CSS):

```
<ol>
  <li>Coffee</li>
  <li style="list-style-type:lower-alpha">Tea</li>
  <li>Milk</li>
</ol>
```

```
<ul>
  <li>Coffee</li>
  <li style="list-style-type:square">Tea</li>
  <li>Milk</li>
</ul>
```



### Example:

Create a list inside a list (a nested list):

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

### Example:

Create a more complex nested list:

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea
        <ul>
          <li>China</li>
          <li>Africa</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

### Default CSS Settings

Most browsers will display the `<li>` element with the following default values:

```
li {
  display: list-item;
}
```

## 90.64HTML <link> Tag

### Example:

Link to an external style sheet:

```
<head>  
  <link rel="stylesheet" href="styles.css">  
</head>
```



### Definition and Usage

The `<link>` tag defines the relationship between the current document and an external resource.

The `<link>` tag is most often used to link to external style sheets or to add a [favicon](#) to your website.

The `<link>` element is an empty element, it contains attributes only.

### Attributes

Attribute	Value	Description
<code>crossorigin</code>	anonymous use-credentials	Specifies how the element handles cross-origin requests
<code>href</code>	<i>URL</i>	Specifies the location of the linked document
<code>hreflang</code>	<i>language_code</i>	Specifies the language of the text in the linked document
<code>media</code>	<i>media_query</i>	Specifies on what device the linked document will be displayed
<code>referrerpolicy</code>	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin unsafe-url	Specifies which referrer to use when fetching the resource



<b>rel</b>	alternate author dns-prefetch help icon license  pingback preconnect prefetch preload prerender prev search stylesheet	Required. Specifies the relationship between the current document and the linked document
<b>sizes</b>	<i>HeightxWidth</i> any	Specifies the size of the linked resource. Only for rel="icon"
<b>title</b>		Defines a preferred or an alternate stylesheet
<b>type</b>	<i>media_type</i>	Specifies the media type of the linked document



## Global Attributes

The `<link>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<link>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<link>` element with the following default values:

```
link {  
  display: none;  
}
```

## 90.65HTML <main> Tag

### Example:

Specify the main content of the document:

```
<main>
  <h1>Most Popular Browsers</h1>
  <p>Chrome, Firefox, and Edge are the most used browsers today.</p>

  <article>
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is
the world's most popular web browser today!</p>
  </article>

  <article>
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has
been the second most popular web browser since January, 2018.</p>
  </article>

  <article>
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
  </article>
</main>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<main>` tag specifies the main content of a document.

The content inside the `<main>` element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.

**Note:** There must not be more than one `<main>` element in a document. The `<main>` element must NOT be a descendant of an `<article>`, `<aside>`, `<footer>`, `<header>`, or `<nav>` element.

### Global Attributes

The `<main>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<main>` tag also supports the [Event Attributes in HTML](#).



## More Examples

### Example:

Use CSS to style the <main> element:

```
<html>
<head>
<style>
main {
  margin: 0;
  padding: 5px;
  background-color: lightgray;
}

main > h1, p, .browser {
  margin: 10px;
  padding: 5px;
}

.browser {
  background: white;
}

.browser > h2, p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>

<main>
  <h1>Most Popular Browsers</h1>
  <p>Chrome, Firefox, and Edge are the most used browsers today.</p>
  <article class="browser">
    <h2>Google Chrome</h2>
    <p>Google Chrome is a web browser developed by Google, released in 2008. Chrome is
the world's most popular web browser today!</p>
  </article>
  <article class="browser">
    <h2>Mozilla Firefox</h2>
    <p>Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has
been the second most popular web browser since January, 2018.</p>
  </article>
  <article class="browser">
    <h2>Microsoft Edge</h2>
    <p>Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.</p>
  </article>
</main>
```



```
</body>
</html>
```



## 90.66HTML <map> Tag

### Example:

An image map, with clickable areas:

```


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">
</map>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<map>` tag is used to define an image map. An image map is an image with clickable areas.

The required name attribute of the `<map>` element is associated with the `<img>`'s use map attribute and creates a relationship between the image and the map.

The `<map>` element contains a number of `<area>` elements, that defines the clickable areas in the image map.

### Attributes

Attribute	Value	Description
<code>name</code>	<i>mapname</i>	Required. Specifies the name of the image map

### Global Attributes

The `<map>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<map>` tag also supports the [Event Attributes in HTML](#).

### More Examples

#### Example:

Another image map, with clickable areas:

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun">
  <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury">
  <area shape="circle" coords="124,58,8" href="venus.htm" alt="Venus">
</map>
```



## Default CSS Settings

Most browsers will display the `<map>` element with the following default values:

```
map {
  display: inline;
}
```

## 90.67HTML `<mark>` Tag

### Example:

Highlight parts of a text:

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

## Definition and Usage

The `<mark>` tag defines text that should be marked or highlighted.

## Global Attributes

The `<mark>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<mark>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<mark>` element with the following default values:

### Example:

```
mark {
  background-color: yellow;
  color: black;
}
```

## 90.68HTML <meta> Tag

### Example:

Describe metadata within an HTML document:

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Free Web tutorials">
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```



More "Try it Yourself" examples below.

### Definition and Usage

The `<meta>` tag defines metadata about an HTML document. Metadata is data (information) about data.

`<meta>` tags always go inside the `<head>` element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.

Metadata will not be displayed on the page, but is machine parsable.

Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

There is a method to let web designers take control over the viewport (the user's visible area of a web page), through the `<meta>` tag (See "Setting The Viewport" example below).

### Attributes

Attribute	Value	Description
<b>charset</b>	<i>character_set</i>	Specifies the character encoding for the HTML document
<b>content</b>	<i>text</i>	Specifies the value associated with the http-equiv or name attribute
<b>http-equiv</b>	content-security-policy content-type default-style refresh	Provides an HTTP header for the information/value of the content attribute
<b>name</b>	application-name author description generator keywords viewport	Specifies a name for the metadata

## Global Attributes

The `<meta>` tag also supports the [Global Attributes in HTML](#).

## More Examples

Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define a description of your web page:

```
<meta name="description" content="Free Web tutorials for HTML and CSS">
```

Define the author of a page:

```
<meta name="author" content="Binod Rabha">
```

Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

Setting the viewport to make your website look good on all devices:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

## Setting the Viewport

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:





Without the viewport meta tag

With the viewport meta tag

## Default CSS Settings

None.

## 90.69HTML <meter> Tag

### Example:

Use the meter element to measure data within a given range (a gauge):

```
<label for="disk_c">Disk usage C:</label>
<meter id="disk_c" value="2" min="0" max="10">2 out of 10</meter><br>
```

```
<label for="disk_d">Disk usage D:</label>
<meter id="disk_d" value="0.6">60%</meter>
```

## Definition and Usage

The `<meter>` tag defines a scalar measurement within a known range, or a fractional value. This is also known as a gauge.

Examples: Disk usage, the relevance of a query result, etc.

**Note:** The `<meter>` tag should not be used to indicate progress (as in a progress bar). For progress bars, use the `<progress>` tag.

**Tip:** Always add the `<label>` tag for best accessibility practices!



## Attributes

Attribute	Value	Description
<b>form</b>	<i>form_id</i>	Specifies which form the <meter> element belongs to
<b>high</b>	<i>number</i>	Specifies the range that is considered to be a high value
<b>low</b>	<i>number</i>	Specifies the range that is considered to be a low value
<b>max</b>	<i>number</i>	Specifies the maximum value of the range
<b>min</b>	<i>number</i>	Specifies the minimum value of the range. Default value is 0
<b>optimum</b>	<i>number</i>	Specifies what value is the optimal value for the gauge
<b>value</b>	<i>number</i>	Required. Specifies the current value of the gauge



## Global Attributes

The <meter> tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The <meter> tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

None.

## 90.70HTML <nav> Tag

### Example:

A set of navigation links:

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/python/">Python</a>
</nav>
```

## Definition and Usage

The <nav> tag defines a set of navigation links.

Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

## Global Attributes

The `<nav>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<nav>` tag also supports the [Event Attributes in HTML](#).



## Default CSS Settings

Most browsers will display the `<nav>` element with the following default values:

```
nav {  
  display: block;  
}
```

## 90.71HTML `<noframes>` Tag

### Not Supported in HTML5.

The `<noframes>` tag was used in HTML 4 to act as a fallback tag for browsers that did not support frames.

## What to Use Instead?

### Example:

Use the `<iframe>` tag to embed another document within the current HTML document:

```
<iframe src="https://www.bintr.com"></iframe>
```

## 90.72HTML `<noscript>` Tag

### Example:

Use of the `<noscript>` tag:

```
<script>  
document.write("Hello World!")  
</script>  
<noscript>Your browser does not support JavaScript!</noscript>
```

## Definition and Usage

The `<noscript>` tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support script.

The `<noscript>` element can be used in both `<head>` and `<body>`. When used inside `<head>`, the `<noscript>` element could only contain `<link>`, `<style>`, and `<meta>` elements.



## Global Attributes

The `<noscript>` tag also supports the [Global Attributes in HTML](#).

## Default CSS Settings

None.

## 90.73HTML `<object>` Tag

### Example:

An embedded image:

```
<object data="pic_trulli.jpg" width="300" height="200"></object>
```

### Example:

An embedded HTML page:

```
<object data="snippet.html" width="500" height="200"></object>
```

### Example:

An embedded video:

```
<object data="video.mp4" width="400" height="300"></object>
```

## Definition and Usage

The `<object>` tag defines a container for an external resource.

The external resource can be a web page, a picture, a media player, or a plug-in application.

To embed a picture, it is better to use the `<img>` tag.

To embed HTML, it is better to use the `<iframe>` tag.

To embed video or audio, it is better to use the `<video>` and `<audio>` tags.

## Plug-ins

The `<object>` tag was originally designed to embed browser Plug-ins.

Plug-ins are computer programs that extend the standard functionality of the browser.

Plug-ins have been used for many different purposes:

- Run Java applets
- Run ActiveX controls
- Display Flash movies
- Display maps
- Scan for viruses
- Verify a bank id



## Warning !

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browser.

The support for Shockwave Flash has also been turned off in modern browsers.

## Attributes

Attribute	Value	Description
<b>data</b>	<i>URL</i>	Specifies the URL of the resource to be used by the object
<b>form</b>	<i>form_id</i>	Specifies which form the object belongs to
<b>height</b>	<i>pixels</i>	Specifies the height of the object
<b>name</b>	<i>name</i>	Specifies a name for the object
<b>type</b>	<i>media_type</i>	Specifies the media type of data specified in the data attribute
<b>typemustmatch</b>	<i>true/false</i>	Specifies whether the type attribute and the actual content of the resource must match to be displayed
<b>usemap</b>	<i>#mapname</i>	Specifies the name of a client-side image map to be used with the object
<b>width</b>	<i>pixels</i>	Specifies the width of the object

## Global Attributes

The `<object>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<object>` tag also supports the Event Attributes in HTML.

## Default CSS Settings

Most browsers will display the `<object>` element with the following default values:

```
object:focus {  
  outline: none;  
}
```



## 90.74HTML <ol> Tag

### Example:

Two different ordered lists (the first list starts at 1, and the second starts at 50):

```
<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

```
<ol start="50">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<ol>` tag defines an ordered list. An ordered list can be numerical or alphabetical.

The `<li>` tag is used to define each list item.

**Tip:** Use CSS to [style lists](#).

**Tip:** For unordered list, use the `<ul>` tag.

### Attributes

Attribute	Value	Description
<code>reversed</code>	reversed	Specifies that the list order should be reversed (9,8,7...)
<code>start</code>	<i>number</i>	Specifies the start value of an ordered list
<code>type</code>	1 A a I i	Specifies the kind of marker to use in the list

## Global Attributes

The `<ol>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<ol>` tag also supports the [Event Attributes in HTML](#).



## More Examples

### Example:

Set different list types (with CSS):

```
<ol style="list-style-type:upper-roman">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

```
<ol style="list-style-type:lower-alpha">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```

### Example:

Display all the different list types available with CSS:

```
<style>
ol.a {list-style-type: armenian;}
ol.b {list-style-type: cjk-ideographic;}
ol.c {list-style-type: decimal;}
ol.d {list-style-type: decimal-leading-zero;}
ol.e {list-style-type: georgian;}
ol.f {list-style-type: hebrew;}
ol.g {list-style-type: hiragana;}
ol.h {list-style-type: hiragana-iroha;}
ol.i {list-style-type: katakana;}
ol.j {list-style-type: katakana-iroha;}
ol.k {list-style-type: lower-alpha;}
ol.l {list-style-type: lower-greek;}
ol.m {list-style-type: lower-latin;}
ol.n {list-style-type: lower-roman;}
ol.o {list-style-type: upper-alpha;}
ol.p {list-style-type: upper-latin;}
ol.q {list-style-type: upper-roman;}
ol.r {list-style-type: none;}
ol.s {list-style-type: inherit;}
</style>
```

### Example:

Reduce and expand line-height in lists (with CSS):

```
<ol style="line-height:80%">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

```
<ol style="line-height:180%">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```



### Example:

Nest an unordered list inside an ordered list:

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ol>
```

### Default CSS Settings

Most browsers will display the `<ol>` element with the following default values:

### Example:

```
ol {
  display: block;
  list-style-type: decimal;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-left: 0;
  margin-right: 0;
  padding-left: 40px;
}
```

## 90.75HTML <optgroup> Tag

### Example:

Group related options with <optgroup> tags:

```
<label for="cars">Choose a car:</label>
<select name="cars" id="cars">
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```



### Definition and Usage

The <optgroup> tag is used to group related options in a <select> element (drop-down list).

If you have a long list of options, groups of related options are easier to handle for a user.

### Attributes

Attribute	Value	Description
<b>disabled</b>	disabled	Specifies that an option-group should be disabled
<b>label</b>	<i>text</i>	Specifies a label for an option-group

### Global Attributes

The <optgroup> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <optgroup> tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

None.

## 90.76HTML <option> Tag

### Example:

A drop-down list with four options:



```
<label for="cars">Choose a car:</label>
```

```
<select id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```



More "Try it Yourself" examples below.

## Definition and Usage

The `<option>` tag defines an option in a select list.

`<option>` elements go inside a `<select>`, `<optgroup>`, or `<datalist>` element.

**Note:** The `<option>` tag can be used without any attributes, but you usually need the **value** attribute, which indicates what is sent to the server on form submission.

**Tip:** If you have a long list of options, you can group related options within the `<optgroup>` tag.

## Attributes

Attribute	Value	Description
<code>disabled</code>	disabled	Specifies that an option should be disabled
<code>label</code>	<i>text</i>	Specifies a shorter label for an option
<code>selected</code>	selected	Specifies that an option should be pre-selected when the page loads
<code>value</code>	<i>text</i>	Specifies the value to be sent to a server

## Global Attributes

The `<option>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<option>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use of `<option>` in a `<datalist>` element:

```
<label for="browser">Choose your browser from the list:</label>
<input list="browsers" name="browser" id="browser">
```

```
<datalist id="browsers">
  <option value="Edge">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```



### Example:

Use of <option> in <optgroup> elements:

```
<label for="cars">Choose a car:</label>
<select id="cars">
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

### Default CSS Settings

None.

## 90.77HTML <output> Tag

### Example:

Perform a calculation and show the result in an <output> element:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  <input type="range" id="a" value="50">
  +<input type="number" id="b" value="25">
  =<output name="x" for="a b"></output>
</form>
```

### Definition and Usage

The <output> tag is used to represent the result of a calculation (like one performed by a script).

### Attributes

Attribute	Value	Description
<b>for</b>	<i>element_id</i>	Specifies the relationship between the result of the calculation, and the elements used in the calculation
<b>form</b>	<i>form_id</i>	Specifies which form the output element belongs to
<b>name</b>	<i>name</i>	Specifies a name for the output element



## Global Attributes

The `<output>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<output>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<output>` element with the following default values:

```
output {  
  display: inline;  
}
```

## 90.78HTML `<p>` Tag

### Example:

A paragraph is marked up as follows:

```
<p>This is some text in a paragraph.</p>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<p>` tag defines a paragraph.

Browsers automatically add a single blank line before and after each `<p>` element.

**Tip:** Use CSS to [style paragraphs](#).

## Global Attributes

The `<p>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<p>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Align text in a paragraph (with CSS):

```
<p style="text-align:right">This is some text in a paragraph.</p>
```



### Example:

Style paragraphs with CSS:

```
<html>
<head>
<style>
p {
  color: navy;
  text-indent: 30px;
  text-transform: uppercase;
}
</style>
</head>
<body>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt
mollit anim id est laborum.</p>
```

```
</body>
</html>
```

### Example:

More on paragraphs:

```
<p>
This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.
</p>
```

### Example:

Poem problems in HTML:

```
<p>
My Bonnie lies over the ocean.
My Bonnie lies over the sea.
```

```
My Bonnie lies over the ocean.  
Oh, bring back my Bonnie to me.  
</p>
```



## Default CSS Settings

Most browsers will display the `<p>` element with the following default values:

### Example:

```
p {  
  display: block;  
  margin-top: 1em;  
  margin-bottom: 1em;  
  margin-left: 0;  
  margin-right: 0;  
}
```

## 90.79HTML `<param>` Tag

### Example:

Set the "autoplay" parameter to "true", so the sound will start playing as soon as the page loads:

```
<object data="horse.wav">  
  <param name="autoplay" value="true">  
</object>
```

## Definition and Usage

The `<param>` tag is used to define parameters for an `<object>` element.

## Browser Support

The `<param>` tag is supported in all major browsers. However, the file format defined in `<object>` may not be supported in all browsers.

## Attributes

Attribute	Value	Description
-----------	-------	-------------

<b>name</b>	<i>name</i>	Specifies the name of a parameter
<b>value</b>	<i>value</i>	Specifies the value of the parameter



## Global Attributes

The `<param>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<param>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<param>` element with the following default values:

```
param {  
  display: none;  
}
```

## 90.80HTML `<picture>` Tag

### Example:

How to use the `<picture>` tag:

```
<picture>  
  <source media="(min-width:650px)" srcset="img_pink_flowers.jpg">  
  <source media="(min-width:465px)" srcset="img_white_flower.jpg">  
    
</picture>
```

## Definition and Usage

The `<picture>` tag gives web developers more flexibility in specifying image resources.

The most common use of the `<picture>` element will be for art direction in responsive designs. Instead of having one image that is scaled up or down based on the viewport width, multiple images can be designed to more nicely fill the browser viewport.

The `<picture>` element contains two tags: one or more `<source>` tags and one `<img>` tag.

The browser will look for the first `<source>` element where the media query matches the current viewport width, and then it will display the proper image (specified in the `srcset` attribute). The `<img>` element is required as the last child of the `<picture>` element, as a fallback option if none of the source tags matches.

**Tip:** The `<picture>` element works "similar" to `<video>` and `<audio>`. You set up different sources, and the first source that fits the preferences is the one being used.

## Global Attributes

The `<picture>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<picture>` tag also supports the [Event Attributes in HTML](#).



## 90.81HTML `<pre>` Tag

### Example:

Preformatted text:

```
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both spaces and
line breaks
</pre>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<pre>` tag defines preformatted text.

Text in a `<pre>` element is displayed in a fixed-width font, and the text preserves both spaces and line breaks. The text will be displayed exactly as written in the HTML source code.

Also look at:

Tag	Description
<code>&lt;code&gt;</code>	Defines a piece of computer code
<code>&lt;samp&gt;</code>	Defines sample output from a computer program
<code>&lt;kbd&gt;</code>	Defines keyboard input
<code>&lt;var&gt;</code>	Defines a variable

## Global Attributes

The `<pre>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<pre>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

How to create a pre-formatted text with a fixed width (with CSS):

```
<div style="width:200px;overflow:auto">
<pre>This is a pre with a fixed width. It will use as much space as specified.</pre>
</div>
```



## Default CSS Settings

Most browsers will display the `<pre>` element with the following default values:

### Example:

```
pre {
  display: block;
  font-family: monospace;
  white-space: pre;
  margin: 1em 0;
}
```

## 90.82HTML `<progress>` Tag

### Example:

Show a progress bar:

```
<label for="file">Downloading progress:</label>
<progress id="file" value="32" max="100"> 32% </progress>
```

## Definition and Usage

The `<progress>` tag represents the completion progress of a task.

**Tip:** Always add the `<label>` tag for best accessibility practices!

## Tips and Notes

**Tip:** Use the `<progress>` tag in conjunction with JavaScript to display the progress of a task.

**Note:** The `<progress>` tag is not suitable for representing a gauge (e.g. disk space usage or relevance of a query result). To represent a gauge, use the `<meter>` tag instead.



## Attributes

Attribute	Value	Description
max	number	Specifies how much work the task requires in total. Default value is 1
value	number	Specifies how much of the task has been completed



## Global Attributes

The `<progress>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<progress>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

None.

## 90.83HTML `<q>` Tag

### Example:

Mark up a short quotation:

```
<p>WWF's goal is to:
<q>Build a future where people live in harmony with nature.</q>
We hope they succeed.</p>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<q>` tag defines a short quotation.

Browsers normally insert quotation marks around the quotation.

**Tip:** Use `<blockquote>` for long quotations.

## Attributes

Attribute	Value	Description
-----------	-------	-------------

<b>cite</b>	<i>URL</i>	Specifies the source URL of the quote
-------------	------------	---------------------------------------



## Global Attributes

The `<q>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<q>` tag also supports the [Event Attributes in HTML](#).

## More Examples

Example

Use CSS to style the `<q>` element:

```
<html>
<head>
<style>
q {
  color: gray;
  font-style: italic;
}
</style>
</head>
<body>
```

```
<p>WWF's goal is to:
<q>Build a future where people live in harmony with nature.</q>
We hope they succeed.</p>
```

```
</body>
</html>
```

## Default CSS Settings

Most browsers will display the `<q>` element with the following default values:

**Example:**

```
q {
  display: inline;
}

q:before {
  content: open-quote;
}

q:after {
```

```
content: close-quote;  
}
```

## 90.84HTML <rp> Tag

### Example:

A ruby annotation:

```
<ruby>  
<rp></rp><rt>厂 马` </rt><rp></rp>  
</ruby>
```



### Definition and Usage

The `<rp>` tag can be used to provide parentheses around a ruby text, to be shown by browsers that do not support ruby annotations.

Use `<rp>` together with `<ruby>` and `<rt>`: The `<ruby>` element consists of one or more characters that needs an explanation/pronunciation, and an `<rt>` element that gives that information, and an optional `<rp>` element that defines what to show for browsers that not support ruby annotations.

### Global Attributes

The `<rp>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<rp>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

None.

## 90.85HTML <rt> Tag

### Example:

A ruby annotation:

```
<ruby>  
<rt> বিনোদ রাভা </rt>  
</ruby>
```

### Definition and Usage

The `<rt>` tag defines an explanation or pronunciation of characters (for East Asian typography) in a ruby annotation.

Use `<rt>` together with `<ruby>` and `<rp>`: The `<ruby>` element consists of one or more characters that needs an explanation/pronunciation, and an `<rt>` element that gives that information, and an optional `<rp>` element that defines what to show for browsers that not support ruby annotations.



## Global Attributes

The `<rt>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<rt>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<rt>` element with the following default values:

```
rt {  
  line-height: normal;  
}
```

## 90.86HTML `<ruby>` Tag

### Example

A ruby annotation:

```
<ruby>  
<rt> bbbb </rt>  
</ruby>
```

## Definition and Usage

The `<ruby>` tag specifies a ruby annotation.

A ruby annotation is a small extra text, attached to the main text to indicate the pronunciation or meaning of the corresponding characters. This kind of annotation is often used in Japanese publications.

Use `<ruby>` together with `<rt>` and `<rp>`: The `<ruby>` element consists of one or more characters that needs an explanation/pronunciation, and an `<rt>` element that gives that information, and an optional `<rp>` element that defines what to show for browsers that do not support ruby annotations.

## Global Attributes

The `<ruby>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<ruby>` tag also supports the [Event Attributes in HTML](#).



## 90.87HTML `<s>` Tag

### Example:

Mark up text that is no longer correct:

```
<p><s>Only 50 tickets left!</s></p>
<p>SOLD OUT!</p>
```

### Definition and Usage

The `<s>` tag specifies text that is no longer correct, accurate or relevant. The text will be displayed with a line through it.

The `<s>` tag should not be used to define deleted text in a document, use the `<del>` tag for that.

### Global Attributes

The `<s>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<s>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

Most browsers will display the `<s>` element with the following default values:

### Example:

```
s {
  text-decoration: line-through;
}
```

## 90.88HTML `<samp>` Tag

### Example:

Define some text as sample output from a computer program in a document:

```
<p>Message from my computer:</p>
<p><samp>File not found.<br>Press F1 to continue</samp></p>
```

## Definition and Usage

The `<samp>` tag is used to define sample output from a computer program. The content inside is displayed in the browser's default monospace font.

**Tip:** This tag is not deprecated. However, it is possible to achieve richer effect by using CSS.

Also look at:



Tag	Description
<code>&lt;code&gt;</code>	Defines a piece of computer code
<code>&lt;kbd&gt;</code>	Defines keyboard input
<code>&lt;var&gt;</code>	Defines a variable
<code>&lt;pre&gt;</code>	Defines preformatted text

## Global Attributes

The `<samp>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<samp>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<samp>` element with the following default values:

### Example:

```
samp {  
  font-family: monospace;  
}
```

## 90.89HTML `<script>` Tag

### Example:

Write "Hello JavaScript!" with JavaScript:

```
<script>  
document.getElementById("demo").innerHTML = "Hello JavaScript!";  
</script>
```

## Definition and Usage

The `<script>` tag is used to embed a client-side script (JavaScript).

The `<script>` element either contains scripting statements, or it points to an external script file through the `src` attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.



## Tips and Notes

**Tip:** Also look at the `<noscript>` element for users that have disabled scripts in their browser, or have a browser that doesn't support client-side scripting.

**Tip:** If you want to learn more about JavaScript, visit our [JavaScript Tutorial](#).

## Attributes

Attribute	Value	Description
<b>async</b>	async	Specifies that the script is downloaded in parallel to parsing the page, and executed as soon as it is available (before parsing completes) (only for external scripts)
<b>crossorigin</b>	anonymous use-credentials	Sets the mode of the request to an HTTP CORS Request
<b>defer</b>	defer	Specifies that the script is downloaded in parallel to parsing the page, and executed after the page has finished parsing (only for external scripts)
<b>integrity</b>	<i>filehash</i>	Allows a browser to check the fetched script to ensure that the code is never loaded if the source has been manipulated
<b>nomodule</b>	True False	Specifies that the script should not be executed in browsers supporting <a href="#">ES2015</a> modules
<b>referrerpolicy</b>	no-referrer no-referrer-when-downgrade origin origin-when-cross-origin same-origin strict-origin strict-origin-when-cross-origin unsafe-url	Specifies which referrer information to send when fetching a script
<b>src</b>	<i>URL</i>	Specifies the URL of an external script file
<b>type</b>	<i>scripttype</i>	Specifies the media type of the script

## Differences Between HTML and XHTML

In XHTML, the content inside scripts is declared as #PCDATA (instead of CDATA), which means that entities will be parsed.

This means that in XHTML, all special characters should be encoded, or all content should be wrapped inside a CDATA section:

```
<script type="text/javascript">
//
var i = 10;
if (i &lt; 5) {
    // some code
}
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="887 96 979 174" data-label="Image"><img alt="Icon representing HTML code, showing a document with a code symbol and the text 'HTML' below it."/></div><div data-bbox="56 312 230 331" data-label="Section-Header"><h2>Global Attributes</h2></div><div data-bbox="56 342 526 359" data-label="Text"><p>The <code>&lt;script&gt;</code> tag also supports the <a href="#">Global Attributes in HTML</a>.</p></div><div data-bbox="56 399 258 419" data-label="Section-Header"><h2>Default CSS Settings</h2></div><div data-bbox="56 429 683 445" data-label="Text"><p>Most browsers will display the <code>&lt;script&gt;</code> element with the following default values:</p></div><div data-bbox="56 454 223 505" data-label="Text"><pre>script {
    display: none;
}</pre></div><div data-bbox="56 547 348 568" data-label="Section-Header"><h2>90.90HTML &lt;section&gt; Tag</h2></div><div data-bbox="56 580 152 599" data-label="Section-Header"><h3>Example:</h3></div><div data-bbox="56 608 328 626" data-label="Text"><p>Two sections in a document:</p></div><div data-bbox="56 635 943 861" data-label="Text"><pre>&lt;section&gt;
&lt;h2&gt;WWF History&lt;/h2&gt;
&lt;p&gt;The World Wide Fund for Nature (WWF) is an international organization working on
issues regarding the conservation, research and restoration of the environment,
formerly named the World Wildlife Fund. WWF was founded in 1961.&lt;/p&gt;
&lt;/section&gt;

&lt;section&gt;
&lt;h2&gt;WWF's Symbol&lt;/h2&gt;
&lt;p&gt;The Panda has become the symbol of WWF. The well-known panda logo of WWF originated
from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo
in the same year of the establishment of WWF.&lt;/p&gt;
&lt;/section&gt;</pre></div><div data-bbox="56 913 267 934" data-label="Section-Header"><h2>Definition and Usage</h2></div><div data-bbox="434 954 561 972" data-label="Page-Footer">Page 442 of 499</div>
```



The `<section>` tag defines a section in a document.

## Global Attributes

The `<section>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<section>` tag also supports the [Event Attributes in HTML](#).



## Default CSS Settings

Most browsers will display the `<section>` element with the following default values:

### Example:

```
section {  
  display: block;  
}
```

## 90.91HTML `<select>` Tag

### Example:

Create a drop-down list with four options:

```
<label for="cars">Choose a car:</label>  
  
<select name="cars" id="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="mercedes">Mercedes</option>  
  <option value="audi">Audi</option>  
</select>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<select>` element is used to create a drop-down list.

The `<select>` element is most often used in a form, to collect user input.

The `name` attribute is needed to reference the form data after the form is submitted (if you omit the `name` attribute, no data from the drop-down list will be submitted).

The `id` attribute is needed to associate the drop-down list with a label.

The `<option>` tags inside the `<select>` element define the available options in the drop-down list.

**Tip:** Always add the `<label>` tag for best accessibility practices!

## Attributes

Attribute	Value	Description
<b>autofocus</b>	autofocus	Specifies that the drop-down list should automatically get focus when the page loads
<b>disabled</b>	disabled	Specifies that a drop-down list should be disabled
<b>form</b>	<i>form_id</i>	Defines which form the drop-down list belongs to
<b>multiple</b>	multiple	Specifies that multiple options can be selected at once
<b>name</b>	<i>name</i>	Defines a name for the drop-down list
<b>required</b>	required	Specifies that the user is required to select a value before submitting the form
<b>size</b>	<i>number</i>	Defines the number of visible options in a drop-down list



## Global Attributes

The `<select>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<select>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use `<select>` with `<optgroup>` tags:

```
<label for="cars">Choose a car:</label>
<select name="cars" id="cars">
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

## Default CSS Settings

None.

## 90.92HTML <small> Tag

### Example:

Define a smaller text:

```
<p>This is some normal text.</p>
<p><small>This is some smaller text.</small></p>
```

More "Try it Yourself" examples below.



### Definition and Usage

The `<small>` tag defines smaller text (like copyright and other side-comments).

**Tip:** This tag is not deprecated, but it is possible to achieve richer (or the same) effect with CSS.

### Global Attributes

The `<small>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<small>` tag also supports the [Event Attributes in HTML](#).

### More Examples

#### Example:

Use CSS to define smaller text:

```
<html>
<head>
<style>
span.small {
  font-size: smaller;
}
</style>
</head>
<body>

<p>This is some normal text.</p>
<p><span class="small">This is some smaller text.</span></p>

</body>
</html>
```

### Default CSS Settings

Most browsers will display the `<small>` element with the following default values:

#### Example:

```
small {  
  font-size: smaller;  
}
```



## 90.93HTML <source> Tag

### Example:

An audio player with two source files. The browser will choose the first <source> it supports:

```
<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
  Your browser does not support the audio element.  
</audio>
```

More "Try it Yourself" examples below.

### Definition and Usage

The <source> tag is used to specify multiple media resources for media elements, such as <video>, <audio>, and <picture>.

The <source> tag allows you to specify alternative video/audio/image files which the browser may choose from, based on browser support or viewport width. The browser will choose the first <source> it supports.

### Attributes

Attribute	Value	Description
<b>media</b>	<i>media_query</i>	Accepts any valid media query that would normally be defined in a CSS
<b>sizes</b>		Specifies image sizes for different page layouts
<b>src</b>	<i>URL</i>	Required when <source> is used in <audio> and <video>. Specifies the URL of the media file
<b>srcset</b>	<i>URL</i>	Required when <source> is used in <picture>. Specifies the URL of the image to use in different situations
<b>type</b>	<i>MIME-type</i>	Specifies the MIME-type of the resource

### Global Attributes

The <source> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <source> tag also supports the [Event Attributes in HTML](#).

### More Examples

### Example:

Use <source> within <video> to play a video:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```



### Example:

Use <source> within <picture> to define different images based on the viewport width:

```
<picture>
  <source media="(min-width:650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width:465px)" srcset="img_white_flower.jpg">
  
</picture>
```

### Default CSS Settings

None.

## 90.94HTML <span> Tag

### Example:

A <span> element which is used to color a part of a text:

```
<p>My mother has <span style="color:blue">blue</span> eyes.</p>
```

### Definition and Usage

The <span> tag is an inline container used to mark up a part of a text, or a part of a document.

The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

### Global Attributes

The <span> tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The <span> tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

None.



## 90.95HTML <strike> Tag

### Not Supported in HTML5.

The `<strike>` tag was used in HTML 4 to define strikethrough text.

## What to Use Instead?

### Example:

Use the `<del>` tag to define deleted text:

```
<p>My favorite color is <del>blue</del> <ins>red</ins>!</p>
```

### Example:

Use the `<s>` tag to mark up text that is no longer correct:

```
<p><s>My car is blue.</s></p>
```

## 90.96HTML <strong> Tag

### Example:

Define important text in a document:

```
<strong>This text is important!</strong>
```

## Definition and Usage

The `<strong>` tag is used to define text with strong importance. The content inside is typically displayed in **bold**.

**Tip:** Use the `<b>` tag to specify bold text without any extra importance!

## Global Attributes

The `<strong>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<strong>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<strong>` element with the following default values:

**Example:**

```
strong {  
  font-weight: bold;  
}
```



## 90.97HTML <style> Tag

**Example:**

Use of the <style> element to apply a simple style sheet to an HTML document:

```
<html>  
<head>  
<style>  
  h1 {color:red;}  
  p {color:blue;}  
</style>  
</head>  
<body>  
  
<h1>A heading</h1>  
<p>A paragraph.</p>  
  
</body>  
</html>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<style>` tag is used to define style information (CSS) for a document.

Inside the `<style>` element you specify how HTML elements should render in a browser.

### Tips and Notes

**Note:** When a browser reads a style sheet, it will format the HTML document according to the information in the style sheet. If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used (see example below)!

**Tip:** To link to an external style sheet, use the `<link>` tag.

**Tip:** To learn more about style sheets, please read our [CSS Tutorial](#).

### Attributes

Attribute	Value	Description
-----------	-------	-------------

<b>media</b>	<i>media_query</i>	Specifies what media/device the media resource is optimized for
<b>type</b>	text/css	Specifies the media type of the <style> tag



## Global Attributes

The <style> tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The <style> tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Multiple styles for the same elements:

```
<html>
<head>
<style>
  h1 {color:red;}
  p {color:blue;}
</style>
<style>
  h1 {color:green;}
  p {color:pink;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

HTML DOM reference: [Style Object](#)

## Default CSS Settings

Most browsers will display the <style> element with the following default values:

```
style {
  display: none;
}
```



## 90.98HTML <sub> Tag

### Example:

Subscript text:

```
<p>This text contains <sub>subscript</sub> text.</p>
```



### Definition and Usage

The `<sub>` tag defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H<sub>2</sub>O.

**Tip:** Use the `<sup>` tag to define superscripted text.

### Global Attributes

The `<sub>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<sub>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

Most browsers will display the `<sub>` element with the following default values:

### Example:

```
sub {  
  vertical-align: sub;  
  font-size: smaller;  
}
```

## 90.99HTML <summary> Tag

### Example:

Using the `<summary>` element:

```
<details>  
  <summary>Epcot Center</summary>  
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions,  
international pavilions, award-winning fireworks and seasonal special events.</p>  
</details>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<summary>` tag defines a visible heading for the `<details>` element. The heading can be clicked to view/hide the details.

**Note:** The `<summary>` element should be the first child element of the `<details>` element.



## Global Attributes

The `<summary>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<summary>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style `<details>` and `<summary>`:

```
<html>
<style>
details > summary {
  padding: 4px;
  width: 200px;
  background-color: #eeeeee;
  border: none;
  box-shadow: 1px 1px 2px #bbbbbb;
  cursor: pointer;
}

details > p {
  background-color: #eeeeee;
  padding: 4px;
  margin: 0;
  box-shadow: 1px 1px 2px #bbbbbb;
}
</style>
<body>

<details>
  <summary>Epcot Center</summary>
  <p>Epcot is a theme park at Walt Disney World Resort featuring exciting attractions,
international pavilions, award-winning fireworks and seasonal special events.</p>
</details>

</body>
</html>
```

## Default CSS Settings

Most browsers will display the `<summary>` element with the following default values:

```
summary {  
  display: block;  
}
```



## 90.100 HTML `<sup>` Tag

### Example:

Superscript text:

```
<p>This text contains <sup>superscript</sup> text.</p>
```

### Definition and Usage

The `<sup>` tag defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW<sup>[1]</sup>.

**Tip:** Use the `<sub>` tag to define subscript text.

### Global Attributes

The `<sup>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<sup>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

Most browsers will display the `<sup>` element with the following default values:

### Example:

```
sup {  
  vertical-align: super;  
  font-size: smaller;  
}
```

## 90.101 HTML `<svg>` Tag

### Example:

Draw a circle:

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<svg>` tag defines a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

To learn more about SVG, please read our [SVG Tutorial](#).

## More Examples

### Example:

Draw a rectangle:

```
<svg width="400" height="100">
  <rect width="400" height="100" style="fill:rgb(0,0,255);stroke-
width:10;stroke:rgb(0,0,0)" />
</svg>
```

### Example:

Draw a square with rounded corners:

```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150" style="fill:red;stroke:
black;stroke-width:5;opacity:0.5" />
</svg>
```

### Example:

Draw a star:

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>
```

### Example:

Draw an SVG logo:

```
<svg height="130" width="500">
<defs>
<linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
  <stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" />
```

```
<stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1" />
</linearGradient>
</defs>

<ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />

<text fill="#ffffff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>
</svg>
```



## 90.102 HTML <table> Tag

### Example:

A simple HTML table, containing two columns and two rows:

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<table>` tag defines an HTML table.

An HTML table consists of one `<table>` element and one or more `<tr>`, `<th>`, and `<td>` elements.

The `<tr>` element defines a table row, the `<th>` element defines a table header, and the `<td>` element defines a table cell.

An HTML table may also include `<caption>`, `<colgroup>`, `<thead>`, `<tfoot>`, and `<tbody>` elements.

### Global Attributes

The `<table>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<table>` tag also supports the [Event Attributes in HTML](#).

### More Examples

### Example:

How to add collapsed borders to a table (with CSS):

```
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>

<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>

</body>
</html>
```



### Example:

How to right-align a table (with CSS):

```
<table style="float:right">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```

**Example:**

How to center-align a table (with CSS):

```
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
}
table.center {
  margin-left: auto;
  margin-right: auto;
}
</style>
</head>
<body>
```

```
<table class="center">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100 ₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```

**Example:**

How to add background-color to a table (with CSS):

```
<table style="background-color:#00FF00">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```



**Example:**

How to add padding to a table (with CSS):

```
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
}

th, td {
  padding: 10px;
}
</style>
</head>
<body>

<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
</body>
</html>
```

**Example:**

How to set table width (with CSS):

```
<table style="width:400px">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
```



```
<td>80₹</td>
</tr>
</table>
```

### Example:

How to create table headers:

```
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Phone</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>123-45-678</td>
  </tr>
</table>
```



### Example:

How to create a table with a caption:

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```

### Example:

How to define table cells that span more than one row or one column:

```
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th colspan="2">Phone</th>
  </tr>
  <tr>
```

```
<td>Binod Rabha</td>
<td>binod.rabha@example.com</td>
<td>123-45-678</td>
<td>212-00-546</td>
</tr>
</table>
```



## Default CSS Settings

Most browsers will display the `<table>` element with the following default values:

### Example:

```
table {
  display: table;
  border-collapse: separate;
  border-spacing: 2px;
  border-color: gray;
}
```

## 90.103 HTML `<tbody>` Tag

### Example:

An HTML table with a `<thead>`, `<tbody>`, and a `<tfoot>` element:

```
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>180 ₹</td>
    </tr>
  </tfoot>
</table>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<tbody>` tag is used to group the body content in an HTML table.

The `<tbody>` element is used in conjunction with the `<thead>` and `<tfoot>` elements to specify each part of a table (body, header, footer).

Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.

**Note:** The `<tbody>` element must have one or more `<tr>` tags inside.

The `<tbody>` tag must be used in the following context: As a child of a `<table>` element, after any `<caption>`, `<colgroup>`, and `<thead>` elements.

**Tip:** The `<thead>`, `<tbody>`, and `<tfoot>` elements will not affect the layout of the table by default. However, you can use CSS to style these elements (see example below)!

## Global Attributes

The `<tbody>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<tbody>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Style `<thead>`, `<tbody>`, and `<tfoot>` with CSS:

```
<html>
<head>
<style>
thead {color: green;}
tbody {color: blue;}
tfoot {color: red;}

table, th, td {
  border: 1px solid black;
}
</style>
</head>
<body>

<table>
  <thead>
    <tr>
```

```
<th>Month</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>100₹</td>
</tr>
<tr>
<td>February</td>
<td>80₹</td>
</tr>
</tbody>
<tfoot>
<tr>
<td>Sum</td>
<td>180 ₹</td>
</tr>
</tfoot>
</table>
```



### Example:

How to align content inside <tbody> (with CSS):

```
<table style="width:100%">
<thead>
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
</thead>
<tbody style="text-align:right">
<tr>
<td>January</td>
<td>100₹</td>
</tr>
<tr>
<td>February</td>
<td>80₹</td>
</tr>
</tbody>
</table>
```

### Example:

How to vertical align content inside <tbody> (with CSS):

```
<table style="width:50%;">
<tr>
```

```
<th>Month</th>
<th>Savings</th>
</tr>
<tbody style="vertical-align:bottom">
  <tr style="height:100px">
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr style="height:100px">
    <td>February</td>
    <td>80₹</td>
  </tr>
</tbody>
</table>
```



## Default CSS Settings

Most browsers will display the `<tbody>` element with the following default values:

```
tbody {
  display: table-row-group;
  vertical-align: middle;
  border-color: inherit;
}
```

## 90.104 HTML `<td>` Tag

### Example:

A simple HTML table, with two rows and four table cells:

```
<table>
  <tr>
    <td>Cell A</td>
    <td>Cell B</td>
  </tr>
  <tr>
    <td>Cell C</td>
    <td>Cell D</td>
  </tr>
</table>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<td>` tag defines a standard data cell in an HTML table.

An HTML table has two kinds of cells:

- Header cells - contains header information (created with the `<th>` element)
- Data cells - contains data (created with the `<td>` element)

The text in `<td>` elements are regular and left-aligned by default.

The text in `<th>` elements are bold and centered by default.



## Attributes

Attribute	Value	Description
<code>colspan</code>	<i>number</i>	Specifies the number of columns a cell should span
<code>headers</code>	<i>header_id</i>	Specifies one or more header cells a cell is related to
<code>rowspan</code>	<i>number</i>	Sets the number of rows a cell should span

## Global Attributes

The `<td>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<td>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

How to align content inside `<td>` (with CSS):

```
<table style="width:100%">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td style="text-align:right">100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td style="text-align:right">80₹</td>
  </tr>
</table>
```

### Example:

How to add background-color to table cell (with CSS):

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td style="background-color:#FF0000">January</td>
    <td style="background-color:#00FF00">100₹</td>
  </tr>
</table>
```



### Example:

How to set the height of a table cell (with CSS):

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td style="height:100px">January</td>
    <td style="height:100px">100₹</td>
  </tr>
</table>
```

### Example:

How to specify no word-wrapping in table cell (with CSS):

```
<table>
  <tr>
    <th>Poem</th>
  </tr>
  <tr>
    <td style="white-space:nowrap">Never increase, beyond what is necessary, the
number of entities required to explain anything</td>
  </tr>
</table>
```

### Example:

How to vertical align content inside <td> (with CSS):

```
<table style="width:50%;">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr style="height:100px">
    <td style="vertical-align:bottom">January</td>
```

```
<td style="vertical-align:bottom">100₹</td>
</tr>
</table>
```

### Example:

How to set the width of a table cell (with CSS):

```
<table style="width:100%">
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
<tr>
<td style="width:70%">January</td>
<td style="width:30%">100₹</td>
</tr>
</table>
```

### Example:

How to create table headers:

```
<table>
<tr>
<th>Name</th>
<th>Email</th>
<th>Phone</th>
</tr>
<tr>
<td>John Doe</td>
<td>john.doe@example.com</td>
<td>123-45-678</td>
</tr>
</table>
```

### Example:

How to create a table with a caption:

```
<table>
<caption>Monthly savings</caption>
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
<tr>
<td>January</td>
<td>100₹</td>
</tr>
<tr>
```





```
<td>February</td>
<td>80₹</td>
</tr>
</table>
```



### Example:

How to define table cells that span more than one row or one column:

```
<table>
<tr>
<th>Name</th>
<th>Email</th>
<th colspan="2">Phone</th>
</tr>
<tr>
<td>John Doe</td>
<td>john.doe@example.com</td>
<td>123-45-678</td>
<td>212-00-546</td>
</tr>
</table>
```

### Default CSS Settings

Most browsers will display the `<td>` element with the following default values:

```
td {
  display: table-cell;
  vertical-align: inherit;
}
```

## 90.105 HTML `<template>` Tag

### Example:

Use `<template>` to hold some content that will be hidden when the page loads. Use JavaScript to display it:

```
<button onclick="showContent()">Show hidden content</button>

<template>
  <h2>Flower</h2>
  
</template>

<script>
function showContent() {
  var temp = document.getElementsByTagName("template")[0];
  var clon = temp.content.cloneNode(true);
```

```
document.body.appendChild(clon);
}
</script>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<template>` tag is used as a container to hold some HTML content hidden from the user when the page loads.

The content inside `<template>` can be rendered later with a JavaScript.

You can use the `<template>` tag if you have some HTML code you want to use over and over again, but not until you ask for it. To do this *without* the `<template>` tag, you have to create the HTML code with JavaScript to prevent the browser from rendering the code.

## Global Attributes

The `<template>` tag supports the Global Attributes in HTML.

## More Examples

### Example:

Fill the web page with one new div element for each item in an array. The HTML code of each div element is inside the template element:

```
<template>
  <div class="myClass">I like: </div>
</template>

<script>
var myArr = ["Audi", "BMW", "Ford", "Honda", "Jaguar", "Nissan"];
function showContent() {
  var temp, item, a, i;
  temp = document.getElementsByTagName("template")[0];
  item = temp.content.querySelector("div");
  for (i = 0; i < myArr.length; i++) {
    a = document.importNode(item, true);
    a.textContent += myArr[i];
    document.body.appendChild(a);
  }
}
</script>
```

### Example:

Check browser support for `<template>`:

```
<script>
if (document.createElement("template").content) {
```

```
document.write("Your browser supports template!");
} else {
document.write("Your browser does not supports template!");
}
</script>
```



## 90.106 HTML <textarea> Tag

### Example:

A multi-line text input control (text area):

```
<label for="review">Review of Bintr:</label>
```

```
<textarea id="review" name="review" rows="4" cols="50">
```

At bintr.com you will learn how to make a website. They offer free tutorials in all web development technologies.

```
</textarea>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<textarea>` tag defines a multi-line text input control.

The `<textarea>` element is often used in a form, to collect user inputs like comments or reviews.

A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).

The size of a text area is specified by the `<cols>` and `<rows>` attributes (or with CSS).

The `name` attribute is needed to reference the form data after the form is submitted (if you omit the `name` attribute, no data from the text area will be submitted).

The `id` attribute is needed to associate the text area with a label.

**Tip:** Always add the `<label>` tag for best accessibility practices!

### Attributes

Attribute	Value	Description
<code>autofocus</code>	autofocus	Specifies that a text area should automatically get focus when the page loads
<code>cols</code>	<i>number</i>	Specifies the visible width of a text area
<code>dirname</code>	<i>textarea.name.dir</i>	Specifies that the text direction of the textarea will be submitted
<code>disabled</code>	disabled	Specifies that a text area should be disabled
<code>form</code>	<i>form_id</i>	Specifies which form the text area belongs to
<code>maxlength</code>	<i>number</i>	Specifies the maximum number of characters allowed in the text area
<code>name</code>	<i>text</i>	Specifies a name for a text area

<b>placeholder</b>	<i>text</i>	Specifies a short hint that describes the expected value of a text area
<b>readonly</b>	readonly	Specifies that a text area should be read-only
<b>required</b>	required	Specifies that a text area is required/must be filled out
<b>rows</b>	<i>number</i>	Specifies the visible number of lines in a text area
<b>wrap</b>	hard soft	Specifies how the text in a text area is to be wrapped when submitted in a form



## Global Attributes

The `<textarea>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<textarea>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Disable default resize option:

```
<html>
<head>
<style>
textarea {
  resize: none;
}
</style>
</head>
<body>

<label for="review">Review of Bintr:</label>

<textarea id="review" name="review" rows="4" cols="50">
At bintr.com you will learn how to make a website. They offer free tutorials in all
web development technologies.
</textarea>

</body>
</html>
```

## Default CSS Settings

None.

## 90.107 HTML `<tfoot>` Tag

### Example:

An HTML table with a `<thead>`, `<tbody>`, and a `<tfoot>` element:

```
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>180 ₹</td>
    </tr>
  </tfoot>
</table>
```



More "Try it Yourself" examples below.

## Definition and Usage

The `<tfoot>` tag is used to group footer content in an HTML table.

The `<tfoot>` element is used in conjunction with the `<thead>` and `<tbody>` elements to specify each part of a table (footer, header, body).

Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.

**Note:** The `<tfoot>` element must have one or more `<tr>` tags inside.

The `<tfoot>` tag must be used in the following context: As a child of a `<table>` element, after any `<caption>`, `<colgroup>`, `<thead>`, and `<tbody>` elements.

**Tip:** The `<thead>`, `<tbody>`, and `<tfoot>` elements will not affect the layout of the table by default. However, you can use CSS to style these elements (see example below)!

## Global Attributes

The `<tfoot>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<tfoot>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Style `<thead>`, `<tbody>`, and `<tfoot>` with CSS:

```
<html>
<head>
<style>
thead {color: green;}
tbody {color: blue;}
tfoot {color: red;}

table, th, td {
  border: 1px solid black;
}
</style>
</head>
<body>
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>180 ₹</td>
    </tr>
  </tfoot>
</table>
```

### Example:

How to align content inside `<tfoot>` (with CSS):



```
<table style="width:100%">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
  <tfoot style="text-align:center">
    <tr>
      <td>Sum</td>
      <td>180 ₹</td>
    </tr>
  </tfoot>
</table>
```



### Example:

How to vertical align content inside <tfoot> (with CSS):

```
<table style="width:100%">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
  <tfoot style="vertical-align:bottom">
    <tr style="height:100px">
      <td>Sum</td>
      <td>180 ₹</td>
    </tr>
  </tfoot>
</table>
```

### Default CSS Settings

Most browsers will display the <tfoot> element with the following default values:

```
tfoot {  
  display: table-footer-group;  
  vertical-align: middle;  
  border-color: inherit;  
}
```



## 90.108 HTML <th> Tag

### Example:

A simple HTML table with three rows, two header cells and four data cells:

```
<table>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>100₹</td>  
  </tr>  
  <tr>  
    <td>February</td>  
    <td>80₹</td>  
  </tr>  
</table>
```

More "Try it Yourself" examples below.

### Definition and Usage

The `<th>` tag defines a header cell in an HTML table.

An HTML table has two kinds of cells:

- Header cells - contains header information (created with the `<th>` element)
- Data cells - contains data (created with the `<td>` element)

The text in `<th>` elements are bold and centered by default.

The text in `<td>` elements are regular and left-aligned by default.

### Attributes

Attribute	Value	Description
<b>abbr</b>	<i>text</i>	Specifies an abbreviated version of the content in a header cell
<b>colspan</b>	<i>number</i>	Specifies the number of columns a header cell should span



<b>headers</b>	<i>header_id</i>	Specifies one or more header cells a cell is related to
<b>rowspan</b>	<i>number</i>	Specifies the number of rows a header cell should span
<b>scope</b>	col colgroup row rowgroup	Specifies whether a header cell is a header for a column, row, or group of columns or rows



## Global Attributes

The `<th>` tag also supports the Global Attributes in HTML.

## Event Attributes

The `<th>` tag also supports the Event Attributes in HTML.

## More Examples

### Example:

How to align content inside `<th>` (with CSS):

```
<table style="width:100%">
  <tr>
    <th style="text-align:left">Month</th>
    <th style="text-align:left">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```

### Example:

How to add background-color to table header cell (with CSS):

```
<table>
  <tr>
    <th style="background-color:#FF0000">Month</th>
    <th style="background-color:#00FF00">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

**Example:**

How to set the height of a table header cell (with CSS):

```
<table>
  <tr>
    <th style="height:100px">Month</th>
    <th style="height:100px">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

**Example:**

How to specify no word-wrapping in table header cell (with CSS):

```
<table>
  <tr>
    <th>Month</th>
    <th style="white-space:nowrap">My Savings for a new car</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

**Example:**

How to vertical align content inside <th> (with CSS):

```
<table style="width:50%;">
  <tr style="height:100px">
    <th style="vertical-align:bottom">Month</th>
    <th style="vertical-align:bottom">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

**Example:**

How to set the width of a table header cell (with CSS):

```
<table style="width:100%">
  <tr>
    <th style="width:70%">Month</th>
    <th style="width:30%">Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```



### Example:

How to create table headers:

```
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Phone</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>123-45-678</td>
  </tr>
</table>
```

### Example:

How to create a table with a caption:

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```

### Example:

How to define table cells that span more than one row or one column:

```
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th colspan="2">Phone</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>123-45-678</td>
    <td>212-00-546</td>
  </tr>
</table>
```



## Default CSS Settings

Most browsers will display the `<th>` element with the following default values:

```
th {
  display: table-cell;
  vertical-align: inherit;
  font-weight: bold;
  text-align: center;
}
```

## 90.109 HTML `<thead>` Tag

### Example:

An HTML table with a `<thead>`, `<tbody>`, and a `<tfoot>` element:

```
<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
```

```
<tfoot>
  <tr>
    <td>Sum</td>
    <td>180 ₹</td>
  </tr>
</tfoot>
</table>
```



"Try it Yourself".

## Definition and Usage

The `<thead>` tag is used to group header content in an HTML table.

The `<thead>` element is used in conjunction with the `<tbody>` and `<tfoot>` elements to specify each part of a table (header, body, footer).

Browsers can use these elements to enable scrolling of the table body independently of the header and footer. Also, when printing a large table that spans multiple pages, these elements can enable the table header and footer to be printed at the top and bottom of each page.

**Note:** The `<thead>` element must have one or more `<tr>` tags inside.

The `<thead>` tag must be used in the following context: As a child of a `<table>` element, after any `<caption>` and `<colgroup>` elements, and before any `<tbody>`, `<tfoot>`, and `<tr>` elements.

**Tip:** The `<thead>`, `<tbody>`, and `<tfoot>` elements will not affect the layout of the table by default. However, you can use CSS to style these elements (see example below)!

## Global Attributes

The `<thead>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<thead>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Style `<thead>`, `<tbody>`, and `<tfoot>` with CSS:

```
<html>
<head>
<style>
thead {color: green;}
tbody {color: blue;}
tfoot {color: red;}

table, th, td {
  border: 1px solid black;
}
```

```
</style>
</head>
<body>

<table>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>180 ₹</td>
    </tr>
  </tfoot>
</table>
```



### Example:

How to align content inside <thead> (with CSS):

```
<table style="width:100%">
  <thead style="text-align:left">
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
</table>
```

## Example:

How to vertical align content inside <thead> (with CSS):

```
<table style="width:50%;">
  <thead style="vertical-align:bottom">
    <tr style="height:100px">
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100₹</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80₹</td>
    </tr>
  </tbody>
</table>
```



## Default CSS Settings

Most browsers will display the <thead> element with the following default values:

```
thead {
  display: table-header-group;
  vertical-align: middle;
  border-color: inherit;
}
```

## 90.110 HTML <time> Tag

### Example:

How to define a time and a date:

```
<p>Open from <time>10:00</time> to <time>21:00</time> every weekday.</p>
```

```
<p>I have a date on <time datetime="2008-02-14 20:00">Valentines day</time>.</p>
```

## Definition and Usage

The <time> tag defines a specific time (or datetime).

The `datetime` attribute of this element is used to translate the time into a machine-readable format so that browsers can offer to add date reminders through the user's calendar, and search engines can produce smarter search results.



## Attributes

Attribute	Value	Description
<code>datetime</code>	<code>datetime</code>	Represent a machine-readable format of the <code>&lt;time&gt;</code> element

## Global Attributes

The `<time>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<time>` tag also supports the [Event Attributes in HTML](#).

## 90.111 HTML `<title>` Tag

### Example:

Define a title for your HTML document:

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Elements Reference</title>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

## Definition and Usage

The `<title>` tag defines the title of the document. The title must be text-only, and it is shown in the browser's title bar or in the page's tab.

The `<title>` tag is required in HTML documents!

The contents of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

The `<title>` element:

- defines a title in the browser toolbar
- provides a title for the page when it is added to favorites
- displays a title for the page in search-engine results



Here are some tips for creating good titles:

- Go for a longer, descriptive title (avoid one- or two-word titles)
- Search engines will display about 50-60 characters of the title, so try not to have titles longer than that
- Do not use just a list of words as the title (this may reduce the page's position in search results)

So, try to make the title as accurate and meaningful as possible!

**Note:** You can NOT have more than one `<title>` element in an HTML document.



## Global Attributes

The `<title>` tag also supports the Global Attributes in HTML.

## Default CSS Settings

Most browsers will display the `<title>` element with the following default values:

```
title {  
  display: none;  
}
```

## 90.112 HTML `<tr>` Tag

### Example:

A simple HTML table with three rows; one header row and two data rows:

```
<table>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>100₹</td>  
  </tr>  
  <tr>  
    <td>February</td>  
    <td>80₹</td>  
  </tr>  
</table>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<tr>` tag defines a row in an HTML table.

A `<tr>` element contains one or more `<th>` or `<td>` elements.

## Global Attributes

The `<tr>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<tr>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example

How to align content inside `<tr>` (with CSS):

```
<table style="width:100%">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr style="text-align:right">
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

### Example:

How to add background-color to a table row (with CSS):

```
<table>
  <tr style="background-color:#FF0000">
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```

### Example:

How to vertical align content inside `<tr>` (with CSS):

```
<table style="height:200px">
  <tr style="vertical-align:top">
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr style="vertical-align:bottom">
    <td>January</td>
    <td>100₹</td>
  </tr>
</table>
```



**Example:**

How to create table headers:

```
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th>Phone</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>123-45-678</td>
  </tr>
</table>
```

**Example:**

How to create a table with a caption:

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>100₹</td>
  </tr>
  <tr>
    <td>February</td>
    <td>80₹</td>
  </tr>
</table>
```

**Example:**

How to define table cells that span more than one row or one column:

```
<table>
  <tr>
    <th>Name</th>
    <th>Email</th>
    <th colspan="2">Phone</th>
  </tr>
  <tr>
    <td>John Doe</td>
    <td>john.doe@example.com</td>
    <td>123-45-678</td>
  </tr>
```

```
<td>212-00-546</td>
</tr>
</table>
```



## Default CSS Settings

Most browsers will display the `<tr>` element with the following default values:

```
tr {
  display: table-row;
  vertical-align: inherit;
  border-color: inherit;
}
```

## 90.113 HTML `<track>` Tag

### Example:

A video with subtitle tracks for two languages:

```
<video width="320" height="240" controls>
  <source src="forrest_gump.mp4" type="video/mp4">
  <source src="forrest_gump.ogg" type="video/ogg">
  <track src="fgsubtitles_en.vtt" kind="subtitles" srclang="en" label="English">
  <track src="fgsubtitles_no.vtt" kind="subtitles" srclang="no" label="Norwegian">
</video>
```

## Definition and Usage

The `<track>` tag specifies text tracks for `<audio>` or `<video>` elements.

This element is used to specify subtitles, caption files or other files containing text, that should be visible when the media is playing.

Tracks are formatted in WebVTT format (.vtt files).

## Optional Attributes

Attribute	Value	Description
<b>default</b>	default	Specifies that the track is to be enabled if the user's preferences do not indicate that another track would be more appropriate
<b>kind</b>	captions chapters descriptions metadata subtitles	Specifies the kind of text track
<b>label</b>	<i>text</i>	Specifies the title of the text track
<b>src</b>	<i>URL</i>	Required. Specifies the URL of the track file

<code>srclang</code>	<code>language_code</code>	Specifies the language of the track text data (required if <code>kind="subtitles"</code> )
----------------------	----------------------------	--



## Global Attributes

The `<track>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<track>` tag also supports the [Event Attributes in HTML](#).

## Default CSS Settings

None.

## 90.114 HTML `<tt>` Tag

### Not Supported in HTML5.

The `<tt>` tag was used in HTML 4 to define teletype text.

## What to Use Instead?

Consider the `<kbd>` element (for keyboard input), the `<var>` element (for variables), the `<code>` element (for computer code), the `<samp>` element (for computer output), or use CSS instead.

### Example:

Define a teletype/monospace font for a `<p>` element (with CSS):

```
<p style="font-family:'Lucida Console', monospace">This text is monospace text.</p>
```

In our CSS tutorial you can find more details about specifying the [type of font](#) in a document.

## 90.115 HTML `<u>` Tag

### Example:

Mark up a misspelled word with the `<u>` tag:

```
<p>This is some <u>mispeled</u> text.</p>
```

More "Try it Yourself" examples below.

## Definition and Usage

The `<u>` tag represents some text that is unarticulated and styled differently from normal text, such as misspelled words or proper names in Chinese text. The content inside is typically displayed with an underline. You can change this with CSS (see example below).

**Tip:** Avoid using the `<u>` element where it could be confused for a hyperlink!



## Global Attributes

The `<u>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<u>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Use CSS to style misspelled text:

```
<html>
<head>
<style>
.spelling-error {
  text-decoration-line: underline;
  text-decoration-style: wavy;
  text-decoration-color: red;
}
</style>
</head>
<body>

<p>This is some <u class="spelling-error">mispeled</u> text.</p>

</body>
</html>
```

## Default CSS Settings

Most browsers will display the `<u>` element with the following default values:

### Example:

```
u {
  text-decoration: underline;
}
```

## 90.116 HTML `<ul>` Tag

### Example:

An unordered HTML list:

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

More "Try it Yourself" examples below.



## Definition and Usage

The `<ul>` tag defines an unordered (bulleted) list.

Use the `<ul>` tag together with the `<li>` tag to create unordered lists.

**Tip:** Use CSS to [style lists](#).

**Tip:** For ordered lists, use the `<ol>` tag.

## Global Attributes

The `<ul>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<ul>` tag also supports the [Event Attributes in HTML](#).

## More Examples

### Example:

Set the different list style types (with CSS):

```
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

```
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

```
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

### Example:

Expand and reduce line-height in lists (with CSS):

```
<ul style="line-height:180%">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

```
<ul style="line-height:80%">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



### Example:

Create a list inside a list (a nested list):

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

### Example:

Create a more complex nested list:

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea
        <ul>
          <li>China</li>
          <li>Africa</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

### Default CSS Settings

Most browsers will display the `<ul>` element with the following default values:



Example

```
ul {
  display: block;
  list-style-type: disc;
  margin-top: 1em;
  margin-bottom: 1 em;
  margin-left: 0;
  margin-right: 0;
  padding-left: 40px;
}
```



## 90.117 HTML <var> Tag

### Example

Define some text as variables in a document:

`<p>`The area of a triangle is:  $1/2 \times \text{<var>b</var>} \times \text{<var>h</var>}$ , where `<var>b</var>` is the base, and `<var>h</var>` is the vertical height.`</p>`

### Definition and Usage

The `<var>` tag is used to defines a variable in programming or in a mathematical expression. The content inside is typically displayed in *italic*.

**Tip:** This tag is not deprecated. However, it is possible to achieve richer effect by using CSS.

Also look at:

Tag	Description
<code>&lt;code&gt;</code>	Defines a piece of computer code
<code>&lt;samp&gt;</code>	Defines sample output from a computer program
<code>&lt;kbd&gt;</code>	Defines keyboard input
<code>&lt;pre&gt;</code>	Defines preformatted text

### Global Attributes

The `<var>` tag also supports the [Global Attributes in HTML](#).

### Event Attributes

The `<var>` tag also supports the [Event Attributes in HTML](#).

### Default CSS Settings

Most browsers will display the `<var>` element with the following default values:

### Example:

```
var {
  font-style: italic;
}
```

## 90.118 HTML <video> Tag



### Example:

Play a video:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

### Definition and Usage

The `<video>` tag is used to embed video content in a document, such as a movie clip or other video streams.

The `<video>` tag contains one or more `<source>` tags with different video sources. The browser will choose the first source it supports.

The text between the `<video>` and `</video>` tags will only be displayed in browsers that do not support the `<video>` element.

There are three supported video formats in HTML: MP4, WebM, and OGG.

Browser	MP4	WebM	Ogg
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

### Tips and Notes

**Tip:** For audio files, look at the `<audio>` tag.

### Optional Attributes

Attribute	Value	Description
<code>autoplay</code>	autoplay	Specifies that the video will start playing as soon as it is ready
<code>controls</code>	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
<code>height</code>	<i>pixels</i>	Sets the height of the video player

<b>loop</b>	loop	Specifies that the video will start over again, every time it is finished
<b>muted</b>	muted	Specifies that the audio output of the video should be muted
<b>poster</b>	URL	Specifies an image to be shown while the video is downloading, or until the user hits the play button
<b>preload</b>	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
<b>src</b>	URL	Specifies the URL of the video file
<b>width</b>	pixels	Sets the width of the video player



## Global Attributes

The `<video>` tag also supports the [Global Attributes in HTML](#).

## Event Attributes

The `<video>` tag also supports the [Event Attributes in HTML](#).

Default CSS Settings

None.

## 90.119 HTML `<wbr>` Tag

### Example:

A text with word break opportunities:

```
<p>To learn AJAX, you must be familiar with the XML<wbr>Http<wbr>Request Object.</p>
```

## Definition and Usage

The `<wbr>` (Word Break Opportunity) tag specifies where in a text it would be ok to add a line-break.

Tip: When a word is too long, the browser might break it at the

-----End-----

