Frontend

* HTML

Basics- Elements, Attribiutes,Tags

Forms and Validation

What are sematic tags

Structuring a website

* CSS (Cascading Style Sheet)

Basics- Selectors, properties

Advance concepts (Flexbox, Grid)

Making a Layout with CSS

Responsive design and media query

Designing a mega project

* JAVA SCRIPT

Syntax and basic concepts

Learn DOM Manipulation

What is ES6

Events in Javascript

Functionalities in Megaproject

* FRAMEWORK

Bootstrap, Material CSS

React JS

Angular

HTML, CSS, Java Script, php

Framework- Codeigniter

Bootstrap

Example

< ! DOCTYPE html >

<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>Helo</h1>

<p>A normal text</p>

</body>

</html>

* 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
* 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>

HTML paragraphs are defined with the <p> tag:

Example

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

HTML paragraphs are defined with the <p> tag:

Example

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

HTML links are defined with the <a> tag:

Example

<a href="https://www.w3schools.com">This is a link</a>

## 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 a slow connection, or an error in the src attribute, or if the user uses a screen reader.

### Example

<img src="img\_girl.jpg" alt="Girl with a jacket">

HTML images are defined with the <img> tag.

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

Example

<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">

* alternative text (alt)- It provides alternate information of the image if the use is not able to view the image because of an error or slow connection.

## HTML Tag Reference

W3Schools' tag reference contains additional information about these tags and their attributes.

## 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"**

## 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>

## 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.

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

## 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>

he HTML standard does not require quotes around attribute values.

However, W3C **recommends** quotes in HTML, and **demands** quotes for stricter document types like XHTML.

Good:

<a href="https://www.w3schools.com/html/">Visit our HTML tutorial</a>

Bad:

<a href=https://www.w3schools.com/html/>Visit our HTML tutorial</a>

## 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">

## 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.

## 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>

## 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>

## 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>

## 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>

## 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 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>

## HTML Tag Reference

W3Schools' tag reference contains additional information about HTML elements and their attributes.

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

Example

I am Red

I am Blue

I am Big

## 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.

## 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>

## 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>

## 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>

## 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>

## 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>

## 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

## 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

## 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>

## 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>

## 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>

## 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 60 years, WWF has worked to help people and nature thrive. As the world's leading conservation organization, WWF works in nearly 100 countries. At every level, we collaborate with people around the world to develop and deliver innovative solutions that protect communities, wildlife, and the places in which they live.  
</blockquote>

## 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>

## 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>

## 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 John Doe.<br>  
Visit us at:<br>  
Example.com<br>  
Box 564, Disneyland<br>  
USA  
</address>

## 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>

## 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>

## HTML Quotation and Citation Elements

## HTML Comment Tag

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

<!-- Write your comments here -->

## Hide Content

Comments can be used to hide content.

This can be helpful if you hide content temporarily:

### Example

<p>This is a paragraph.</p>  
  
<!-- <p>This is another paragraph </p> -->

## 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>

# **HTML Colors**

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

## Color Names

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

Tomato

Orange

DodgerBlue

MediumSeaGreen

Gray

SlateBlue

Violet

## Background Color

You can set the background color for HTML elements:

Hello World

Lorem ipsum dolor sit amet, consectetuer 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>

## Text Color

You can set the color of text:

### Hello World

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

### Example

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

## Border Color

You can set the color of borders:

## Hello World

## Hello World

### Example

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

## 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 add 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>

# **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).

## 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 x 256 x 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).

## 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)**

## 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):

# **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.

## HEX Color Values

The hexadecimal number system is a type of number system, that has a base value equal to 16. It is also pronounced sometimes as 'hex'. Hexadecimal numbers are represented by only 16 symbols. These symbols or values are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.

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

**#rrggbb**

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.

# **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).

## 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.

## 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):

# **HTML Styles - CSS**

CSS stands for Cascading Style Sheets.

## 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)!

## 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>

## 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>

## 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;  
}

## 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>

## 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;  
}

## 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;  
}

## 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;  
}

## 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.w3schools.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">

## 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

## HTML 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 W3Schools.com:

<a href="https://www.w3schools.com/">Visit W3Schools.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

## 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

## 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.w3.org/">W3C</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>

## 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>

## 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>

## 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.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>

## 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

## 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>

## 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

## 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

<img src="*url*" alt="alternatetext">

## 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

<img src="img\_chania.jpg" alt="Flowers in Chania">

## Image Size - Width and Height

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

### Example

<img src="img\_girl.jpg" alt="Girl in a jacket" style="width:500px;height:600px;">

## 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>  
  
<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">  
  
<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">  
  
</body>  
</html>

## Images in Another Folder

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

### Example

<img src="/images/html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

## 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

<img src="https://www.w3schools.com/images/w3schools\_green.jpg" alt="W3Schools.com">

## Animated Images

HTML allows animated GIFs:

### Example

<img src="programming.gif" alt="Computer Man" style="width:48px;height:48px;">

## Image as a Link

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

### Example

<a href="default.asp">  
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">

## Image Floating

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

### Example

<p><img src="smiley.gif" alt="Smiley face" style="float:right;width:42px;height:42px;">  
The image will float to the right of the text.</p>  
  
<p><img src="smiley.gif" alt="Smiley face" style="float:left;width:42px;height:42px;">  
The image will float to the left of the text.</p>

## Common Image Formats

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

## 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

## 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.

Example

<img src="workplace.jpg" alt="Workplace" usemap="#workmap">  
  
<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>

## The Image

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

<img src="workplace.jpg" alt="Workplace" usemap="#workmap">

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.

## 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 .

## 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">

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">

### 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">

## 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>

## 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

## 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>

## 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>

## 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:

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>

## 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;  
}  
</style>

## Background Stretch

If you want the background image to stretch to fit the entire element, you can set the background-size property to 100% 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%;  
}  
</style>

## 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.

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<body>

<h2>The picture Element</h2>

<picture>

<source media="(min-width: 650px)" srcset="img\_food.jpg">

<source media="(min-width: 465px)" srcset="img\_car.jpg">

<img src="img\_girl.jpg" style="width:auto;">

</picture>

<p>Resize the browser to see different versions of the picture loading at different viewport sizes.

The browser looks for the first source element where the media query matches the user's current viewport width,

and fetches the image specified in the srcset attribute.</p>

<p>The img element is required as the last child tag of the picture declaration block.

The img element is used to provide backward compatibility for browsers that do not support the picture element, or if none of the source tags matched.

</p>

</body>

</html>

**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.

## 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">  
  <img src="img\_beatles.gif" alt="Beatles" style="width:auto;">  
</picture>

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

You can use any image you like as your favicon. You can also create your own favicon on sites like [https://www.favicon.cc](https://www.favicon.cc/).

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".

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

<!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>

## Chapter Summary

* Use the HTML <link> element to insert a favicon

## 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>

## 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.

## **Note:** A table cell can contain all sorts of HTML elements: text, images, lists, links, other tables, etc.

## Table Rows

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

tr stands for table row.

## Table Headers

Sometimes you want your cells to be table header cells. In those cases use the <th> tag instead of the <td> tag:

th stands for table header.

## How To Add a Border

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

### Example

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

## 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:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

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

## 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:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

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

## Round Table Borders

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

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

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

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

## 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;  
}

## Border Color

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

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

# **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.

## 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>

## HTML Table Column Width

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

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

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>

## HTML Table Headers

Table headers are defined with th elements. Each th element represents a table cell.

## Vertical Table Headers

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

## Align Table Headers

By default, table headers are bold and centered:

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

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

Example

th {  
  text-align: left;  
}

## Header for Multiple Columns

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

|  |  |  |
| --- | --- | --- |
| **Name** | | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |

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

<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>

## Table Caption

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

|  |  |
| --- | --- |
| Monthly savings | |
| **Month** | **Savings** |
| January | $100 |
| February | $50 |

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

<table style="width:100%">  
  <caption>Monthly savings</caption>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>

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

# **HTML Table Padding & Spacing**

## 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:

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:

th, td {  
  padding-top: 10px;  
  padding-bottom: 20px;  
  padding-left: 30px;  
  padding-right: 40px;  
}

## 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:

table {  
  border-spacing: 30px;  
}

## HTML Table - Colspan

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

<table>  
  <tr>  
    <th colspan="2">Name</th>  
    <th>Age</th>  
  </tr>  
  <tr>

……

## HTML Table - Rowspan

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

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

…..

## 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:

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

## 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:

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

## 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);  
}

## Horizontal Dividers

| **First Name** | **Last Name** | **Savings** |
| --- | --- | --- |
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| Joe | Swanson | $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:

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

## Hoverable Table

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

tr:hover {background-color: #D6EEEE;}

## HTML Table Colgroup

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

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

Each group is specified with a <col> element.

The span attribute specifies how many columns that get 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>  
...

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

[width](https://www.w3schools.com/cssref/pr_dim_width.php) property  
[visibility](https://www.w3schools.com/cssref/pr_class_visibility.php) property  
[background](https://www.w3schools.com/cssref/css3_pr_background.php) properties  
[border](https://www.w3schools.com/cssref/pr_border.php) properties

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

## Multiple Col Elements

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

<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>  
...

## 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:

<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>  
...

## Hide Columns

You can hide columns with the visibility: collapse property:

<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>  
...

## 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:

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

## 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:

|  |  |
| --- | --- |
| **Value** | **Description** |
| disc | Sets the list item marker to a bullet (default) |
| circle | Sets the list item marker to a circle |
| square | Sets the list item marker to a square |
| none | The list items will not be marked |

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

## 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:

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

## Ordered HTML List - The Type Attribute

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

|  |  |
| --- | --- |
| **Type** | **Description** |
| type="1" | The list items will be numbered with numbers (default) |
| type="A" | The list items will be numbered with uppercase letters |
| type="a" | The list items will be numbered with lowercase letters |
| type="I" | The list items will be numbered with uppercase roman numbers |
| type="i" | The list items will be numbered with lowercase roman numbers |

### Numbers:

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

## 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:

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

## 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:

<!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>

## 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

## 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:

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

## Nested HTML Lists

Lists can be nested (list inside list):

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

## 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

## 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>](https://www.w3schools.com/tags/tag_address.asp)[<article>](https://www.w3schools.com/tags/tag_article.asp)[<aside>](https://www.w3schools.com/tags/tag_aside.asp)[<blockquote>](https://www.w3schools.com/tags/tag_blockquote.asp)[<canvas>](https://www.w3schools.com/tags/tag_canvas.asp)[<dd>](https://www.w3schools.com/tags/tag_dd.asp)[<div>](https://www.w3schools.com/tags/tag_div.asp)[<dl>](https://www.w3schools.com/tags/tag_dl.asp)[<dt>](https://www.w3schools.com/tags/tag_dt.asp)[<fieldset>](https://www.w3schools.com/tags/tag_fieldset.asp)[<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp)

[<figure>](https://www.w3schools.com/tags/tag_figure.asp)[<footer>](https://www.w3schools.com/tags/tag_footer.asp)[<form>](https://www.w3schools.com/tags/tag_form.asp)[<h1>-<h6>](https://www.w3schools.com/tags/tag_hn.asp)[<header>](https://www.w3schools.com/tags/tag_header.asp)[<hr>](https://www.w3schools.com/tags/tag_hr.asp)[<li>](https://www.w3schools.com/tags/tag_li.asp)[<main>](https://www.w3schools.com/tags/tag_main.asp)[<nav>](https://www.w3schools.com/tags/tag_nav.asp)[<noscript>](https://www.w3schools.com/tags/tag_noscript.asp)[<ol>](https://www.w3schools.com/tags/tag_ol.asp)[<p>](https://www.w3schools.com/tags/tag_p.asp)[<pre>](https://www.w3schools.com/tags/tag_pre.asp)

[<section>](https://www.w3schools.com/tags/tag_section.asp)[<table>](https://www.w3schools.com/tags/tag_table.asp)[<tfoot>](https://www.w3schools.com/tags/tag_tfoot.asp)[<ul>](https://www.w3schools.com/tags/tag_ul.asp)[<video>](https://www.w3schools.com/tags/tag_video.asp)

## 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.

<span>Hello World</span>

Here are the inline elements in HTML:

[<a>](https://www.w3schools.com/tags/tag_a.asp)[<abbr>](https://www.w3schools.com/tags/tag_abbr.asp)[<acronym>](https://www.w3schools.com/tags/tag_acronym.asp)[<b>](https://www.w3schools.com/tags/tag_b.asp)[<bdo>](https://www.w3schools.com/tags/tag_bdo.asp)[<big>](https://www.w3schools.com/tags/tag_big.asp)[<br>](https://www.w3schools.com/tags/tag_br.asp)[<button>](https://www.w3schools.com/tags/tag_button.asp)[<cite>](https://www.w3schools.com/tags/tag_cite.asp)[<code>](https://www.w3schools.com/tags/tag_code.asp)[<dfn>](https://www.w3schools.com/tags/tag_dfn.asp)[<em>](https://www.w3schools.com/tags/tag_em.asp)[<i>](https://www.w3schools.com/tags/tag_i.asp)[<img>](https://www.w3schools.com/tags/tag_img.asp)[<input>](https://www.w3schools.com/tags/tag_input.asp)

[<kbd>](https://www.w3schools.com/tags/tag_kbd.asp)[<label>](https://www.w3schools.com/tags/tag_label.asp)[<map>](https://www.w3schools.com/tags/tag_map.asp)[<object>](https://www.w3schools.com/tags/tag_object.asp)[<output>](https://www.w3schools.com/tags/tag_output.asp)[<q>](https://www.w3schools.com/tags/tag_q.asp)[<samp>](https://www.w3schools.com/tags/tag_samp.asp)[<script>](https://www.w3schools.com/tags/tag_script.asp)[<select>](https://www.w3schools.com/tags/tag_select.asp)[<small>](https://www.w3schools.com/tags/tag_small.asp)[<span>](https://www.w3schools.com/tags/tag_span.asp)[<strong>](https://www.w3schools.com/tags/tag_strong.asp)

[<sub>](https://www.w3schools.com/tags/tag_sub.asp)[<sup>](https://www.w3schools.com/tags/tag_sup.asp)[<textarea>](https://www.w3schools.com/tags/tag_textarea.asp)[<time>](https://www.w3schools.com/tags/tag_time.asp)[<tt>](https://www.w3schools.com/tags/tag_tt.asp)[<var>](https://www.w3schools.com/tags/tag_var.asp)

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

## 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:

<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>

## 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:

<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>

## Chapter Summary

* 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

## <div> as a container

The <div> element is often used to group sections of a web page together.

## Center align a <div> element

If you have a <div> element that is not 100% wide, and you want to center-align it, set the CSS margin property to auto.

<style>  
div {  
  width:300px;  
  margin:auto;  
}  
</style>

## You can have many <div> containers on the same page.

## Float

The CSS float property was not originally meant to align <div> elements side-by-side, but has been used for this purpose for many years.

The CSS float property is used for positioning and formatting content and allow elements float next to each other instead of on top of each other.

How to use float to align div elements side by side:

<style>  
.mycontainer {  
  width:100%;  
  overflow:auto;  
}  
.mycontainer div {  
  width:33%;  
  float:left;  
}  
</style>

## Inline-block

If you change the <div> element's display property from block to inline-block, the <div> elements will no longer add a line break before and after, and will be displayed side by side instead of on top of each other.

How to use display: inline-block to align div elements side by side:

<style>  
div {  
  width: 30%;  
  display: inline-block;  
}  
</style>

## Flex

The CSS Flexbox Layout Module was introduced to make it easier to design flexible responsive layout structure without using float or positioning.

To make the CSS flex method work, surround the <div> elements with another <div> element and give it the status as a flex container.

How to use flex to align div elements side by side:

<style>  
.mycontainer {  
  display: flex;  
}  
.mycontainer > div {  
  width:33%;  
}  
</style>

## Grid

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.

Sounds almost the same as flex, but has the ability to define more than one row and position each row individually.

The CSS grid method requires that you surround the <div> elements with another <div> element and give the status as a grid container, and you must specify the width of each column.

How to use grid to align <div> elements side by side:

<style>  
.grid-container {  
  display: grid;  
  grid-template-columns: 33% 33% 33%;  
}  
</style>

## 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:

<!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:

<!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!

## 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 {}

## 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:

<style>

.city {

background-color: tomato;

color: white;

padding: 10px;

}

.main {

text-align: center;

}

</style>

</head>

<body>

<h2>Multiple Classes</h2>

<p>Here, all three h2 elements belongs to the "city" class. In addition, London also belongs to the "main" class, which center-aligns the text.</p>

<h2 class="city main">London</h2>

<h2 class="city">Paris</h2>

<h2 class="city">Tokyo</h2>

</body>

</html>

## 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:

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>

## 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

## 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.).

## 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

## 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>

<a href="#C4">Jump to Chapter 4</a>

## 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:

Use the id attribute to manipulate text with JavaScript:

<h2>Using The id Attribute in JavaScript</h2>

<p>JavaScript can access an element with a specified id by using the getElementById() method:</p>

<h1 id="myHeader">Hello World!</h1>

<button onclick="displayResult()">Change text</button>

<script>

function displayResult() {

document.getElementById("myHeader").innerHTML = "Have a nice day!";

}

</script>

## 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

# **HTML Iframes**

## An HTML iframe is used to display a web page within a web page.

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

<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.

Use the height and width attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

<iframe src="demo\_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>

## 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:

<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:

<iframe src="demo\_iframe.htm" style="border:2px solid red;" title="Iframe Example"></iframe>

## 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:

<iframe src="demo\_iframe.htm" name="iframe\_a" title="Iframe Example"></iframe>  
  
<p><a href="https://www.w3schools.com" target="iframe\_a">W3Schools.com</a></p>

## 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 specify the size of the iframe
* Use border:none; to remove the border around the iframe

## 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":

<script>  
document.getElementById("demo").innerHTML = "Hello JavaScript!";  
</script>

<p id=”demo”></p>

## 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:

<script>  
document.getElementById("demo").innerHTML = "Hello JavaScript!";  
</script>  
<noscript>Sorry, your browser does not support JavaScript!</noscript>

## File Path Examples

|  |  |
| --- | --- |
| **Path** | **Description** |
| <img src="picture.jpg"> | The "picture.jpg" file is located in the same folder as the current page |
| <img src="images/picture.jpg"> | The "picture.jpg" file is located in the images folder in the current folder |
| <img src="/images/picture.jpg"> | The "picture.jpg" file is located in the images folder at the root of the current web |
| <img src="../picture.jpg"> | The "picture.jpg" file is located in the folder one level up from the current folder |

## 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

## Absolute File Paths

<img src="https://www.w3schools.com/images/picture.jpg" alt="Mountain">

## 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:

<img src="/images/picture.jpg" alt="Mountain">

## 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.

## 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 content 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!

## The HTML <style> Element

The <style> element is used to define style information for a single HTML page:

<style>  
  body {background-color: powderblue;}  
  h1 {color: red;}  
  p {color: blue;}  
</style>

## 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:

<link rel="stylesheet" href="mystyle.css">

## 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 is used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

**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:

<meta charset="UTF-8">  
<meta name="description" content="Free Web tutorials">  
<meta name="keywords" content="HTML, CSS, JavaScript">  
<meta name="author" content="John Doe">

## 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":

<script>  
function myFunction() {  
  document.getElementById("demo").innerHTML = "Hello JavaScript!";  
}  
</script>

## 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!

<!DOCTYPE html>

<html>

<head>

<base href="https://www.w3schools.com/" target="\_blank">

</head>

<body>

<h1>The base element</h1>

<p><img src="images/stickman.gif" width="24" height="39" alt="Stickman"> - Notice that we have only specified a relative address for the image. Since we have specified a base URL in the head section, the browser will look for the image at "https://www.w3schools.com/images/stickman.gif".</p>

<p><a href="tags/tag\_base.asp">HTML base tag</a> - Notice that the link opens in a new window, even if it has no target="\_blank" attribute. This is because the target attribute of the base element is set to "\_blank".</p>

</body>

</html>

## 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

## HTML Layout Elements

HTML has several semantic elements that define the different parts of a web page:

|  |  |
| --- | --- |
| HTML5 Semantic Elements | * <header> - Defines a header for a document or a section * <nav> - Defines a set of navigation links * <section> - Defines a section in a document * <article> - Defines an independent, self-contained content * <aside> - Defines content aside from the content (like a sidebar) * <footer> - Defines a footer for a document or a section * <details> - Defines additional details that the user can open and close on demand * <summary> - Defines a heading for the <details> element |

## 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

## CSS Frameworks

If you want to create your layout fast, you can use a CSS framework, like [W3.CSS](https://www.w3schools.com/w3css/default.asp) or [Bootstrap](https://www.w3schools.com/bootstrap/default.asp).

## CSS Flexbox Layout

Use of flexbox ensures that elements behave predictably when the page layout must accommodate different screen sizes and different display devices.

## 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.

## 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):

### 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

## Responsive Text Size

The text size can be set with a "vw" unit, which means the "viewport width".

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.

## 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.

<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>

## Responsive Web Design - Frameworks

All popular CSS Frameworks offer responsive design.

They are free, and easy to use.

## W3.CSS

W3.CSS is a modern CSS framework with support for desktop, tablet, and mobile design by default.

W3.CSS is smaller and faster than similar CSS frameworks.

W3.CSS is designed to be independent of jQuery or any other JavaScript library.

## Bootstrap

Another popular CSS framework is Bootstrap

HTML contains several elements for defining user input and computer code.

<code>  
x = 5;  
y = 6;  
z = x + y;  
</code>

## 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.

Define some text as keyboard input in a document:

<p>Save the document by pressing <kbd>Ctrl + S</kbd></p>

## 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.

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>

## 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.

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:

<pre>  
<code>  
x = 5;  
y = 6;  
z = x + y;  
</code>  
</pre>

## 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.

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 x b x h, where b is the base, and h is the vertical height.

## 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

# **HTML Semantic Elements**

## Semantic elements = elements with a meaning. 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.

## 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>

## HTML <section> Element

The <section> element defines a section in a document.

According to W3C's 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>

## 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:

* Forum posts
* Blog posts
* User comments
* Product cards
* Newspaper articles

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>

## 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.

## 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.

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>

## 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.

## 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 blocks 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.

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>

## 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.

### 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>

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;  
}

## 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.

<figure>  
  <img src="pic\_trulli.jpg" alt="Trulli">  
  <figcaption>Fig1. - Trulli, Puglia, Italy.</figcaption>  
</figure>

## Why Semantic Elements?

According to the W3C: "A semantic Web allows data to be shared and reused across applications, enterprises, and communities."

## Semantic Elements in HTML

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<article>](https://www.w3schools.com/tags/tag_article.asp) | Defines independent, self-contained content |
| [<aside>](https://www.w3schools.com/tags/tag_aside.asp) | Defines content aside from the page content |
| [<details>](https://www.w3schools.com/tags/tag_details.asp) | Defines additional details that the user can view or hide |
| [<figcaption>](https://www.w3schools.com/tags/tag_figcaption.asp) | Defines a caption for a <figure> element |
| [<figure>](https://www.w3schools.com/tags/tag_figure.asp) | Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. |
| [<footer>](https://www.w3schools.com/tags/tag_footer.asp) | Defines a footer for a document or section |
| [<header>](https://www.w3schools.com/tags/tag_header.asp) | Specifies a header for a document or section |
| [<main>](https://www.w3schools.com/tags/tag_main.asp) | Specifies the main content of a document |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |
| [<nav>](https://www.w3schools.com/tags/tag_nav.asp) | Defines navigation links |
| [<section>](https://www.w3schools.com/tags/tag_section.asp) | Defines a section in a document |
| [<summary>](https://www.w3schools.com/tags/tag_summary.asp) | Defines a visible heading for a <details> element |
| [<time>](https://www.w3schools.com/tags/tag_time.asp) | Defines a date/time |

## 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.

<!DOCTYPE html>  
<html lang="en-us">  
<head>

## 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>

## 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.

## 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

## Loading JavaScript in HTML

Use simple syntax for loading external scripts (the type attribute is not necessary):

<script src="myscript.js">

## Accessing HTML Elements with JavaScript

Using "untidy" HTML code can result in JavaScript errors.

These two JavaScript statements will produce different results:

getElementById("Demo").innerHTML = "Hello";  
  
getElementById("demo").innerHTML = "Hello";

# **HTML Entities**

Reserved characters in HTML must be replaced with entities:

* < (less than) = **&lt;**
* > (greather than) = **&gt;**

## HTML Character Entities

Some characters are reserved in HTML.

If you use the less than (<) or greater than (>) signs in your HTML text, the browser might mix them with tags.

Entity names or entity numbers can be used to display reserved HTML characters.

Entity names look like this:

&*entity\_name*;

Entity numbers look like this:

&#*entity\_number*;

To display a less than sign (<) we must write: **&lt;** or **&#60;**

**Entity names** are easier to remember than entity numbers.

## Non-breaking Space

A commonly used HTML entity 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.

The non-breaking hyphen ([&#8209;](https://www.w3schools.com/charsets/ref_utf_punctuation.asp)) is used to define a hyphen character (‑) that does not break into a new line.

## Some Useful HTML Character Entities

|  |  |  |  |
| --- | --- | --- | --- |
| **Result** | **Description** | **Name** | **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 | &apos; | &#39; |
| ¢ | cent | &cent; | &#162; |
| £ | pound | &pound; | &#163; |
| ¥ | yen | &yen; | &#165; |
| € | euro | &euro; | &#8364; |
| © | copyright | &copy; | &#169; |
| ® | trademark | &reg; | &#174; |

## 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 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 |

## 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:

Display the euro sign:

<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 €

## Some Mathematical Symbols Supported by HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **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 |
| ∑ | &#8721; | &sum; | N-ary summation |

## Some Greek Letters Supported by HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| Α | &#913; | &Alpha; | GREEK ALPHA |
| Β | &#914; | &Beta; | GREEK BETA |
| Γ | &#915; | &Gamma; | GREEK GAMMA |
| Δ | &#916; | &Delta; | GREEK DELTA |
| Ε | &#917; | &Epsilon; | GREEK EPSILON |
| Ζ | &#918; | &Zeta; | GREEK ZETA |

## Some Other Entities Supported by HTML

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | | **Description** |
| © | &#169; | &copy; | | COPYRIGHT |
| ® | &#174; | &reg; | | REGISTERED |
| € | &#8364; | &euro; | | EURO SIGN |
| ™ | &#8482; | &trade; | | TRADEMARK |
| ← | &#8592; | &larr; | | LEFT ARROW |
| ↑ | &#8593; | &uarr; | | UP ARROW |
| → | &#8594; | &rarr; | | RIGHT ARROW |
| ↓ | &#8595; | &darr; | | DOWN ARROW |
| ♠ | &#9824; | &spades; | | SPADE |
| ♣ | &#9827; | &clubs; | | CLUB |
| ♥ | &#9829; | &hearts; | | HEART |
| ♦ | &#9830; | &diams; | | DIAMOND |
| **Emoji** | | | **Value** | |
| 🗻 | | | &#128507; | |
| 🗼 | | | &#128508; | |
| 🗽 | | | &#128509; | |
| 🗾 | | | &#128510; | |
| 🗿 | | | &#128511; | |
| 😀 | | | &#128512; | |
| 😁 | | | &#128513; | |
| 😂 | | | &#128514; | |
| 😃 | | | &#128515; | |
| 😄 | | | &#128516; | |
| 😅 | | | &#128517; | |

## 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

<!DOCTYPE html>  
<html>  
<meta charset="UTF-8">  
<body>  
  
<p>I will display A B C</p>  
<p>I will display &#65; &#66; &#67;</p>  
  
</body>  
</html>

## Emoji Characters

Emojis are also characters from the UTF-8 alphabet:

* 😄 is 128516
* 😍 is 128525
* 💗 is 128151

## The ASCII Character Set

ASCII was the first character encoding standard for the web. It defined 128 different characters that could be used on the internet:

* English letters (A-Z)
* Numbers (0-9)
* Special characters like ! $ + - ( ) @ < >.

## The ANSI Character Set

ANSI (Windows-1252) was the original Windows character set:

* Identical to ASCII for the first 127 characters
* Special characters from 128 to 159
* Identical to UTF-8 from 160 to 255

<meta charset="Windows-1252">

## The ISO-8859-1 Character Set

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.

* Identical to ASCII for the first 127 characters
* Does not use the characters from 128 to 159
* Identical to ANSI and UTF-8 from 160 to 255

### HTML 4 Example

<meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">

### HTML 5 Example

<meta charset="ISO-8859-1">

## The UTF-8 Character Set

* is identical to ASCII for the values from 0 to 127
* Does not use the characters from 128 to 159
* Identical to ANSI and 8859-1 from 160 to 255
* Continues from the value 256 to 10 000 characters

<meta charset="UTF-8">

* A URL can be composed of words (e.g. w3schools.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.

## 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.w3schools.com/html/default.asp> 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 w3schools.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

## 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 |

## URL Encoding

URLs can only be sent over the Internet using the [ASCII character-set](https://www.w3schools.com/charsets/ref_html_ascii.asp). If a URL contains characters outside the ASCII set, the URL has to be converted.

URL encoding converts non-ASCII characters into a format that can be transmitted over the Internet.

URL encoding replaces non-ASCII characters with a "%" followed by hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign, or %20.

## 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 |

## What is XHTML?

* XHTML stands for E**X**tensible **H**yper**T**ext **M**arkup **L**anguage
* 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

## 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**

## 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: <img src="happy.gif" alt="Happy face" />

## 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.

## 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** |
| <input type="text"> | Displays a single-line text input field |
| <input type="radio"> | Displays a radio button (for selecting one of many choices) |
| <input type="checkbox"> | Displays a checkbox (for selecting zero or more of many choices) |
| <input type="submit"> | Displays a submit button (for submitting the form) |
| <input type="button"> | Displays a clickable button |

## Text Fields

The <input type="text"> defines a single-line input field for text input.

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>

## 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 focuses on the input element.

The <label> element also helps 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.

## Radio Buttons

The <input type="radio"> defines a radio button.

Radio buttons let a user select ONE of a limited number of choices.

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>

## Checkboxes

The <input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

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>

## 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.

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>

## 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:

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>

## 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** |
| \_blank | The response is displayed in a new window or tab |
| \_self | The response is displayed in the current window |
| \_parent | The response is displayed in the parent frame |
| \_top | The response is displayed in the full body of the window |
| *framename* | The response is displayed in a named iframe |

The default value is \_self which means that the response will open in the current window.

<form action="/action\_page.php" target="\_blank">

## 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.

This example uses the GET method when submitting the form data:

<form action="/action\_page.php" method="get">

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!

## 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.

A form with autocomplete on:

<form action="/action\_page.php" autocomplete="on">

## 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.

A form with a novalidate attribute:

<form action="/action\_page.php" novalidate>

## List of All <form> Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| [accept-charset](https://www.w3schools.com/tags/att_form_accept_charset.asp) | Specifies the character encodings used for form submission |
| [action](https://www.w3schools.com/tags/att_form_action.asp) | Specifies where to send the form-data when a form is submitted |
| [autocomplete](https://www.w3schools.com/tags/att_form_autocomplete.asp) | Specifies whether a form should have autocomplete on or off |
| [enctype](https://www.w3schools.com/tags/att_form_enctype.asp) | Specifies how the form-data should be encoded when submitting it to the server (only for method="post") |
| [method](https://www.w3schools.com/tags/att_form_method.asp) | Specifies the HTTP method to use when sending form-data |
| [name](https://www.w3schools.com/tags/att_form_name.asp) | Specifies the name of the form |
| [novalidate](https://www.w3schools.com/tags/att_form_novalidate.asp) | Specifies that the form should not be validated when submitted |
| [rel](https://www.w3schools.com/tags/att_form_rel.asp) | Specifies the relationship between a linked resource and the current document |
| [target](https://www.w3schools.com/tags/att_form_target.asp) | Specifies where to display the response that is received after submitting the form |

## 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>

## The <input> Element

One of the most used form elements is the <input> element.

The <input> element can be displayed in several ways, depending on the type attribute.

## <label for="fname">First name:</label> <input type="text" id="fname" name="fname">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.

## The <select> Element

The <select> element defines a drop-down list:

<label for="cars">Choose a car:</label>  
<select id="cars" name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

The <option> element 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:

### <option value="fiat" selected>Fiat</option>

### Visible Values:

Use the size attribute to specify the number of visible values:

### <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:

## <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>

## The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

<form action="/action\_page.php">

<textarea name="message" rows="10" cols="30">The cat was playing in the garden.</textarea>

<br><br>

<input type="submit">

</form>

The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

You can also define the size of the text area by using CSS: <textarea name="message" style="width:200px; height:600px;">  
The cat was playing in the garden.  
</textarea>

## The <button> Element

The <button> element defines a clickable button:

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

**Note:** Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

## 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.

<form action="/action\_page.php">  
  <fieldset>  
    <legend>Personalia:</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>

## 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.

<form action="/action\_page.php">  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Edge">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>

## The <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

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>

## 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".

## Input Type Text

<input type="text"> defines a **single-line text input field**:

## <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>

## Input Type Password

<input type="password"> defines a **password field**:

<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>

## 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:

<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>

## Input Type Reset

<input type="reset"> defines a **reset button** that will reset all form values to their default values:

<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" value="Reset">  
</form>

## Input Type Radio

<input type="radio"> defines a **radio button**.

Radio buttons let a user select ONLY ONE of a limited number of choices:

<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>

## Input Type Checkbox

<input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

## <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>

## Input Type Button

<input type="button"> defines a **button**:

## <input type="button" onclick="alert('Hello World!')" value="Click Me!"> 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

<form>  
  <label for="favcolor">Select your favorite color:</label>  
  <input type="color" id="favcolor" name="favcolor">  
</form>

## 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.

<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: <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>

## 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.

<form>  
  <label for="birthdaytime">Birthday (date and time):</label>  
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">  
</form>

## 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.

<form>  
  <label for="email">Enter your email:</label>  
  <input type="email" id="email" name="email">  
</form>

## Input Type Image

The <input type="image"> defines an image as a submit button.

The path to the image is specified in the src attribute.

<form>  
<input type="image" src="img\_submit.gif" alt="Submit" width="48" height="48">  
</form>

## Input Type File

The <input type="file"> defines a file-select field and a "Browse" button for file uploads.

<form>  
  <label for="myfile">Select a file:</label>  
  <input type="file" id="myfile" name="myfile">  
</form>

## 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!

<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>

## 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.

<form>  
  <label for="bdaymonth">Birthday (month and year):</label>  
  <input type="month" id="bdaymonth" name="bdaymonth">  
</form>

## 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:

<form>  
  <label for="quantity">Quantity (between 1 and 5):</label>  
  <input type="number" id="quantity" name="quantity" min="1" max="5">  
</form>

## Input Restrictions

Here is a list of some common input restrictions:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| checked | Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio") |
| disabled | Specifies that an input field should be disabled |
| max | Specifies the maximum value for an input field |
| maxlength | Specifies the maximum number of character for an input field |
| min | Specifies the minimum value for an input field |
| pattern | Specifies a regular expression to check the input value against |
| readonly | Specifies that an input field is read only (cannot be changed) |
| required | Specifies that an input field is required (must be filled out) |
| size | Specifies the width (in characters) of an input field |
| step | Specifies the legal number intervals for an input field |
| value | Specifies the default value for an input field |

## 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:

<form>  
  <label for="vol">Volume (between 0 and 50):</label>  
  <input type="range" id="vol" name="vol" min="0" max="50">  
</form>

## Input Type Search

The <input type="search"> is used for search fields (a search field behaves like a regular text field).

<form>  
  <label for="gsearch">Search Google:</label>  
  <input type="search" id="gsearch" name="gsearch">  
</form>

## Input Type Tel

The <input type="tel"> is used for input fields that should contain a telephone number.

<form action="/action\_page.php">

<label for="phone">Enter a phone number:</label><br><br>

<input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required><br><br>

<small>Format: 123-45-678</small><br><br>

<input type="submit" value="Submit">

</form>

## 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.

<form>  
  <label for="appt">Select a time:</label>  
  <input type="time" id="appt" name="appt">  
</form>

## 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.

<form>  
  <label for="homepage">Add your homepage:</label>  
  <input type="url" id="homepage" name="homepage">  
</form>

## 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.

<form>  
  <label for="week">Select a week:</label>  
  <input type="week" id="week" name="week">  
</form>

## The value Attribute

The input value attribute specifies an initial value for an input field:

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>

## 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!

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>

## 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!

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>

## 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.

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>

## 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.

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">

## 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.

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>

## 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.

<form>  
  <label for="files">Select files:</label>  
  <input type="file" id="files" name="files" multiple>  
</form>

## 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](https://www.w3schools.com/tags/att_global_title.asp) attribute to describe the pattern to help the user.

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>

## 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, number, tel, email, and password.

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>

## 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.

A required input field:

<form>  
  <label for="username">Username:</label>  
  <input type="text" id="username" name="username" required>  
</form>

## 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.

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)!

## The autofocus Attribute

The input autofocus attribute specifies that an input field should automatically get focus when the page loads.

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>

## 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).

## The list Attribute

The input list attribute refers to a <datalist> element that contains pre-defined options for an <input> element.

An <input> element with pre-defined values in a <datalist>:

<form>  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Edge">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>

## 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.

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>

## 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.

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">

## 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.

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>

## 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.

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>

## 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
* 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>

## 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.

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>

## 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.

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>

## 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.

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>

## 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.

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">  
</canvas>

**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.

**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:

<script>  
const 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>

**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:

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** |
| coords.latitude | The latitude as a decimal number (always returned) |
| coords.longitude | The longitude as a decimal number (always returned) |
| coords.accuracy | The accuracy of position (always returned) |
| coords.altitude | The altitude in meters above the mean sea level (returned if available) |
| coords.altitudeAccuracy | The altitude accuracy of position (returned if available) |
| coords.heading | The heading as degrees clockwise from North (returned if available) |
| coords.speed | The speed in meters per second (returned if available) |
| timestamp | The date/time of the response (returned if available) |

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>  
const 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>

## 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>  
  
<img id="drag1" src="img\_logo.gif" draggable="true" ondragstart="drag(event)" width="336" height="69">  
  
</body>  
</html>

## 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..  
}