**HTML**

HTML is the standard markup language for Web pages.With HTML you can create your own Website.

# HTML Introduction

HTML is the standard markup language for creating Web pages.

## **What is HTML?**

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a Heading

This is a paragraph.

### **Example Explained**

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

## **What is an HTML Element?**

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

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

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

<h1>My First Heading</h1>

<p>My first paragraph.</p>

|  |  |  |
| --- | --- | --- |
| **Start tag** | **Element content** | **End tag** |
| <h1> | My First Heading | </h1> |
| <p> | My first paragraph. | </p> |
| <br> | *none* | *none* |

## **Web Browsers**

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

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



## **HTML Page Structure**

Below is a visualization of an HTML page structure:

<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

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

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

</body>

</html>

## **HTML History**

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

|  |  |
| --- | --- |
| **Year** | **Version** |
| 1989 | Tim Berners-Lee invented www |
| 1991 | Tim Berners-Lee invented HTML |
| 1993 | Dave Raggett drafted HTML+ |
| 1995 | HTML Working Group defined HTML 2.0 |
| 1997 | W3C Recommendation: HTML 3.2 |
| 1999 | W3C Recommendation: HTML 4.01 |
| 2000 | W3C Recommendation: XHTML 1.0 |
| 2008 | WHATWG HTML5 First Public Draft |
| 2012 | [WHATWG HTML5 Living Standard](http://whatwg.org/html/) |
| 2014 | [W3C Recommendation: HTML5](http://www.w3.org/TR/html5/) |
| 2016 | W3C Candidate Recommendation: HTML 5.1 |
| 2017 | [W3C Recommendation: HTML5.1 2nd Edition](http://www.w3.org/TR/html51/) |
| 2017 | [W3C Recommendation: HTML5.2](http://www.w3.org/TR/html52/) |

# HTML Editors

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

## **Learn HTML Using Notepad or TextEdit**

Web pages can be created and modified by using professional HTML editors.

However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac).

We believe that using a simple text editor is a good way to learn HTML.

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

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

**Windows 8 or later:**

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

**Windows 7 or earlier:**

Open **Start** >**Programs >** **Accessories >** **Notepad**

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

Open **Finder > Applications > TextEdit**

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

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

**Then open a new document to place the code.**

## **Step 2: Write Some HTML**

Write or copy the following HTML code into Notepad:

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



## **Step 3: Save the HTML Page**

Save the file on your computer. Select **File > Save as** in the Notepad menu.

Name the file **"index.htm"** and set the encoding to **UTF-8** (which is the preferred encoding for HTML files).



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

Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

The result will look much like this:



## **HTML Documents**

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

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

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

<!DOCTYPE html>

<html>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>

</html>

**OUTPUT**

# My First Heading

My first paragraph.

## **The <!DOCTYPE> Declaration**

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

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

The <!DOCTYPE> declaration is not case sensitive.

The <!DOCTYPE> declaration for HTML5 is:

<!DOCTYPE html>

## **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**

<!DOCTYPE html>

<html>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>

<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

**Output**

# This is heading 1

## This is heading 2

### This is heading 3

#### This is heading 4

##### This is heading 5

###### This is heading 6

## **HTML Paragraphs**

HTML paragraphs are defined with the <p> tag:

**Example**

<!DOCTYPE html>

<html>

<body>

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

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

</body>

</html>

**Output**

This is a paragraph.

This is another paragraph.

## **HTML Links**

HTML links are defined with the <a> tag:

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>HTML Links</h2>

<p>HTML links are defined with the a tag:</p>

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

</body>

</html>

**Output**

## HTML Links

HTML links are defined with the a tag:

[This is a link](https://www.w3schools.com/)

## **HTML Images**

HTML images are defined with the <img> tag.

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>HTML Images</h2>

<p>HTML images are defined with the img tag:</p>

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

</body>

</html>

**Output**

## HTML Images

HTML images are defined with the img tag:



## **How to View HTML Source**

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

### **View HTML Source Code:**

Click CTRL + U in an HTML page, or right-click on the page and select "View Page Source". This will open a new tab containing the HTML source code of the page.

### **Inspect an HTML Element:**

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

## **HTML Elements**

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

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

Examples of some HTML elements:

<h1>My First Heading</h1>

<p>My first paragraph.</p>

|  |  |  |
| --- | --- | --- |
| **Start tag** | **Element content** | **End tag** |
| <h1> | My First Heading | </h1> |
| <p> | My first paragraph. | </p> |
| <br> | none | none |

## **Nested HTML Elements**

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

All HTML documents consist of nested HTML elements.

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>

</html>

**Output**

# My First Heading

My first paragraph.

### **Example Explained**

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

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

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

The <body> element defines the document's body.

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

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

The <h1> element defines a heading.

It has a start tag <h1> and an end tag </h1>:

<h1>My First Heading</h1>

The <p> element defines a paragraph.

It has a start tag <p> and an end tag </p>:

<p>My first paragraph.</p>

## **Never Skip the End Tag**

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

**Example**

<!DOCTYPE html>

<html>

<body>

<p>This is a paragraph.

<p>This is a paragraph.

</body>

</html>

**Output**

This is a paragraph.

This is a paragraph.

## **Empty HTML Elements**

HTML elements with no content are called empty elements.

The <br> tag defines a line break, and is an empty element without a closing tag:

**Example**

<!DOCTYPE html>

<html>

<body>

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

</body>

</html>

**Output**

This is a  
paragraph with a line break.

## **HTML is Not Case Sensitive**

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

# The HTML standard does not require lowercase tags, and demands lowercase for stricter document types like XHTML. HTML Attributes

HTML attributes provide additional information about HTML elements.

* 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 href Attribute**

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>The href Attribute</h2>

<p>HTML links are defined with the a tag. The link address is specified in the href attribute:</p>

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

</body>

</html>

## **The href Attribute**

HTML links are defined with the a tag. The link address is specified in the href attribute:

[Visit W3Schools](https://www.w3schools.com/)

## **The src Attribute**

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

<!DOCTYPE html>

<html>

<body>

<h2>The src Attribute</h2>

<p>HTML images are defined with the img tag, and the filename of the image source is specified in the src attribute:</p>

<img src="img\_girl.jpg" width="500" height="600">

</body>

</html>

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

**1. Absolute URL** - Links to an external image that is hosted on another website. Example: src="https://www.w3schools.com/images/img\_girl.jpg".

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

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

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

## **The width and height Attributes**

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>Width and Height Attributes</h2>

<p>The width and height attributes of the img tag, defines the width and height of the image:</p>

<img src="img\_girl.jpg" width="500" height="600">

</body>

</html>

## **Output**

## Width and Height Attributes

The width and height attributes of the img tag, defines the width and height of the image:



## **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.

## **The style Attribute**

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>The style Attribute</h2>

<p>The style attribute is used to add styles to an element, such as color:</p>

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

</body>

</html>

## **Output**

## The style Attribute

The style attribute is used to add styles to an element, such as color:

This is a red paragraph.

## **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-US">

## **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**

<!DOCTYPE html>

<html>

<body>

<h2 title="I'm a header">The title Attribute</h2>

<p title="I'm a tooltip">Mouse over this paragraph, to display the title attribute as a tooltip.</p>

</body>

</html>

**Output**

## The title Attribute

Mouse over this paragraph, to display the title attribute as a tooltip.

## **Single or Double Quotes?**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h2>Single or Double Quotes?</h2>

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

<p>Move your mouse over the paragraphs below to see the effect:</p>

<p title='John "ShotGun" Nelson'>John with double quotes</p>

<p title="John 'ShotGun' Nelson">John with single quotes</p>

</body>

</html>

## **Single or Double Quotes?**

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

Move your mouse over the paragraphs below to see the effect:

John with double quotes

John with single quotes

# HTML Headings

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

# EXAMPLE

# <!DOCTYPE html>

# <html>

# <body>

# <h1>Heading 1</h1>

# <h2>Heading 2</h2>

# <h3>Heading 3</h3>

# <h4>Heading 4</h4>

# <h5>Heading 5</h5>

# <h6>Heading 6</h6>

# </body>

# </html>

**Output**

# Heading 1

# Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

## **Headings Are Important**

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

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

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

## **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

<!DOCTYPE html>

<html>

<body>

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

<p>You can change the size of a heading with the style attribute, using the font-size property.</p>

</body>

</html>

# Heading 1

You can change the size of a heading with the style attribute, using the font-size property.

# EXAMPLE

<!DOCTYPE html>

<html>

<body>

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

<p>You can change the size of a heading with the style attribute, using the font-size property.</p>

</body>

</html>

**Output**

# Heading 1

You can change the size of a heading with the style attribute, using the font-size property.

## **HTML Paragraphs**

The HTML <p> element defines a paragraph.

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

# EXAMPLE

<!DOCTYPE html>

<html>

<body>

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

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

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

</body>

</html>

**Output**

This is a paragraph.

This is a paragraph.

This is a paragraph.

## **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**

<!DOCTYPE html>

<html>

<body>

<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>

<p>

The number of lines in a paragraph depends on the size of the browser window. If you resize the browser window, the number of lines in this paragraph will change.

</p>

</body>

</html>

**Output**

This paragraph contains a lot of lines in the source code, but the browser ignores it.

This paragraph contains a lot of spaces in the source code, but the browser ignores it.

The number of lines in a paragraph depends on the size of the browser window. If you resize the browser window, the number of lines in this paragraph will change.

## **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**

<!DOCTYPE html>

<html>

<body>

<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>

<h2>This is heading 2</h2>

<p>This is some other text.</p>

</body>

</html>

**OUTPUT**

# This is heading 1

This is some text.

## This is heading 2

This is some other text.

## This is heading 2

This is some other text.

## **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**

<!DOCTYPE html>

<html>

<body>

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

</body>

</html>

**OUTPUT**

This is  
a paragraph  
with line breaks.

## **Solution - The HTML <pre> Element**

The HTML <pre> element defines preformatted text.

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<p>The pre tag preserves both spaces and line breaks:</p>

<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>

</body>

</html>

**OUTPUT**

The pre tag preserves both spaces and line breaks:

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.

# HTML Styles

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<p>I am normal</p>

<p style="color:red;">I am red</p>

<p style="color:blue;">I am blue</p>

<p style="font-size:50px;">I am big</p>

</body>

</html>

**OUTPUT**

I am normal

I am red

I am blue

I am big

## **The HTML Style Background ColorAttribute**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body style="background-color:powderblue;">

<h1>This is a heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

## **Text Color**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="color:blue;">This is a heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

## **Fonts**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="font-family:verdana;">This is a heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

## **Text Size**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="font-size:300%;">This is a heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

## **Text Alignment**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="text-align:center;">Centered Heading</h1>

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

</body>

</html>

**OUTPUT**

# Centered Heading

Centered paragraph.

## **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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<p>This text is normal.</p>

<p><b>This text is bold.</b></p>

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

<p><i>This text is italic.</i></p>

<p><em>This text is emphasized.</em></p>

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

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

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

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

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

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

</body>

</html>

**OUTPUT**

This text is normal.

**This text is bold.**

**This text is important!**

*This text is italic.*

This text is emphasized.

This is some smaller text.

Do not forget to buy milk today.

My favorite color is  red.

My favorite color is  red.

This is subscripted text.

This is superscripted text.

# HTML Quotation and Citation Elements

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<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>

</body>

</html>

**OUTPUT**

Here is a quote from WWF's website:

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.

## **HTML <blockquote> for Quotations**

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

Browsers usually indent <blockquote> elements.

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<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>

</body>

</html>

**OUTPUT**

Here is a quote from WWF's website:

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.

## **HTML <q> for Short Quotations**

The HTML <q> tag defines a short quotation.

Browsers normally insert quotation marks around the quotation.

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<p>Browsers usually insert quotation marks around the q element.</p>

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

</body>

</html>

**OUTPUT**

Browsers usually insert quotation marks around the q element.

WWF's goal is to: Build a future where people live in harmony with nature.

## **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**

<!DOCTYPE html>

<html>

<body>

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

<p>Marking up abbreviations can give useful information to browsers, translation systems and search-engines.</p>

</body>

</html>

**OUTPUT**

The WHO was founded in 1948.

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

## **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**

<!DOCTYPE html>

<html>

<body>

<p>The HTML address element defines contact information (author/owner) of a document or article.</p>

<address>

Written by John Doe.<br>

Visit us at:<br>

Example.com<br>

Box 564, Disneyland<br>

USA

</address>

</body>

</html>

**OUTPUT**

The HTML address element defines contact information (author/owner) of a document or article.

*Written by John Doe.  
Visit us at:  
Example.com  
Box 564, Disneyland  
USA*

## **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**

<!DOCTYPE html>

<html>

<body>

<p>The HTML cite element defines the title of a work.</p>

<p>Browsers usually display cite elements in italic.</p>

<img src="img\_the\_scream.jpg" width="220" height="277" alt="The Scream">

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

</body>

</html>

**OUTPUT**

The HTML cite element defines the title of a work.

Browsers usually display cite elements in italic.

## **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**

<!DOCTYPE html>

<html>

<body>

<p>If your browser supports bi-directional override (bdo), the next line will be written from right to left (rtl):</p>

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

</body>

</html>

**OUTPUT**



*The Scream* by Edvard Munch. Painted in 1893.

## **HTML Comment Tag**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

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

<!--

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

<img border="0" src="pic\_trulli.jpg" alt="Trulli">

-->

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

</body>

</html>

**OUTPUT**

This is a paragraph.

This is a paragraph too.

# HTML Colors

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h3 style="color:Tomato;">Hello World</h3>

<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>

</body>

</html>

**OUTPUT**

### 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.

## **Border Color**

You can set the color of borders:

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

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

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

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

</body>

</html>

**OUTPUT**

# Hello World

# Hello World

# Hello World

## **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:

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<p>Same as color name "Tomato":</p>

<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>

<h1 style="background-color:#ff6347;">#ff6347</h1>

<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>

<p>Same as color name "Tomato", but 50% transparent:</p>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>

<p>In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.</p>

</body>

</html>

Same as color name "Tomato":

# rgb(255, 99, 71)

# #ff6347

# hsl(9, 100%, 64%)

**OUTPUT**

Same as color name "Tomato", but 50% transparent:

# rgba(255, 99, 71, 0.5)

# hsla(9, 100%, 64%, 0.5)

In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>

<h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255)</h1>

<h1 style="background-color:rgb(60, 179, 113);">rgb(60, 179, 113)</h1>

<h1 style="background-color:rgb(238, 130, 238);">rgb(238, 130, 238)</h1>

<h1 style="background-color:rgb(255, 165, 0);">rgb(255, 165, 0)</h1>

<h1 style="background-color:rgb(106, 90, 205);">rgb(106, 90, 205)</h1>

</body>

</html>

**OUTPUT**

# rgb(255, 0, 0)

# rgb(0, 0, 255)

# rgb(60, 179, 113)

# rgb(238, 130, 238)

# rgb(255, 165, 0)

# rgb(106, 90, 205)

## **Shades of Gray**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="background-color:rgb(60, 60, 60);">rgb(60, 60, 60)</h1>

<h1 style="background-color:rgb(100, 100, 100);">rgb(100, 100, 100)</h1>

<h1 style="background-color:rgb(140, 140, 140);">rgb(140, 140, 140)</h1>

<h1 style="background-color:rgb(180, 180, 180);">rgb(180, 180, 180)</h1>

<h1 style="background-color:rgb(200, 200, 200);">rgb(200, 200, 200)</h1>

<h1 style="background-color:rgb(240, 240, 240);">rgb(240, 240, 240)</h1>

</body>

</html>

**OUTPUT**

# 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):

Experiment by mixing the RGBA values below:**ba(255, 99, 71, 0.5)**

RED

255

GREEN

99

BLUE

71

ALPHA

0.5

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="background-color:rgba(255, 99, 71, 0);">rgba(255, 99, 71, 0)</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.2);">rgba(255, 99, 71, 0.2)</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.4);">rgba(255, 99, 71, 0.4)</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.6);">rgba(255, 99, 71, 0.6)</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.8);">rgba(255, 99, 71, 0.8)</h1>

<h1 style="background-color:rgba(255, 99, 71, 1);">rgba(255, 99, 71, 1)</h1>

</body>

</html>

**OUTPUT**

# rgba(255, 99, 71, 0)

# rgba(255, 99, 71, 0.2)

# rgba(255, 99, 71, 0.4)

# rgba(255, 99, 71, 0.6)

# rgba(255, 99, 71, 0.8)

# rgba(255, 99, 71, 1)

# 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**

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.

**EXAMPLE**

# <!DOCTYPE html>

# <html>

# <body>

# <h1 style="background-color:#ff0000;">#ff0000</h1>

# <h1 style="background-color:#0000ff;">#0000ff</h1>

# <h1 style="background-color:#3cb371;">#3cb371</h1>

# <h1 style="background-color:#ee82ee;">#ee82ee</h1>

# <h1 style="background-color:#ffa500;">#ffa500</h1>

# <h1 style="background-color:#6a5acd;">#6a5acd</h1>

# </body>

# </html>

**OUTPUT**

# #ff0000

# #0000ff

# #3cb371

# #ee82ee

# #ffa500

# #6a5acd

## **HSL Color Values**

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

**hsl(hue, saturation, lightness)**

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

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

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

Experiment by mixing the HSL values below:

**hsl(0, 100%, 50%)**

HUE

0

SATURATION

100%

LIGHTNESS

50%

### **Saturation**

Saturation can be described as the intensity of a color.

100% is pure color, no shades of gray.

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

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

### **Lightness**

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

**EXAMPLE**

<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>

<h1 style="background-color:hsl(0, 80%, 50%);">hsl(0, 80%, 50%)</h1>

<h1 style="background-color:hsl(0, 60%, 50%);">hsl(0, 60%, 50%)</h1>

<h1 style="background-color:hsl(0, 40%, 50%);">hsl(0, 40%, 50%)</h1>

<h1 style="background-color:hsl(0, 20%, 50%);">hsl(0, 20%, 50%)</h1>

<h1 style="background-color:hsl(0, 0%, 50%);">hsl(0, 0%, 50%)</h1>

<p>With HSL colors, less saturation mean less color. 0% is completely gray.</p>

</body>

</html>

**OUTPUT**

# hsl(0, 100%, 50%)

# hsl(0, 80%, 50%)

# hsl(0, 60%, 50%)

# hsl(0, 40%, 50%)

# hsl(0, 20%, 50%)

# hsl(0, 0%, 50%)

With HSL colors, less saturation mean less color. 0% is completely gray.

## **HSLA Color Values**

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

An HSLA color value is specified with:

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

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

Experiment by mixing the HSLA values below: **(0, 100%, 50%, 0.5)**

HUE

0

SATURATION

100%

LIGHTNESS

50%

ALPHA

0.5

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h1 style="background-color:hsla(9, 100%, 64%, 0);">hsla(9, 100%, 64%, 0)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.2);">hsla(9, 100%, 64%, 0.2)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.4);">hsla(9, 100%, 64%, 0.4)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.6);">hsla(9, 100%, 64%, 0.6)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 0.8);">hsla(9, 100%, 64%, 0.8)</h1>

<h1 style="background-color:hsla(9, 100%, 64%, 1);">hsla(9, 100%, 64%, 1)</h1>

</body>

</html>

**OUTPUT**

# hsla(9, 100%, 64%, 0)

# hsla(9, 100%, 64%, 0.2)

# hsla(9, 100%, 64%, 0.4)

# hsla(9, 100%, 64%, 0.6)

# hsla(9, 100%, 64%, 0.8)

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

# HTML Styles - CSS

CSS stands for Cascading Style Sheets.

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

## **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)!

## **Using CSS**

CSS can be added to HTML documents in 3 ways:

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

The most common way to add CSS, is to keep the styles in external CSS files.

## **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**

<!DOCTYPE html>

<html>

<body>

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

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

</body>

</html>

**OUTPUT**

# A Blue Heading

A red paragraph.

## **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>

**OUTPUT**

# This is a heading

This is a paragraph.

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**

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**

<!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>

**OUTPUT**

# This is a heading

This is a paragraph.

## **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**

<!DOCTYPE html>

<html>

<head>

<style>

p {

border: 2px solid powderblue;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

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

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

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

## **CSS Padding**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<head>

<style>

p {

border: 2px solid powderblue;

padding: 30px;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

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

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

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

## **CSS Margin**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<head>

<style>

p {

border: 2px solid powderblue;

margin: 50px;

}

</style>

</head>

<body>

<h1>This is a heading</h1>

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

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

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

This is a paragraph.

This is a paragraph.

## **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**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://www.w3schools.com/html/styles.css">

</head>

<body>

<h1>This is a heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

### **Example**

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

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<h1>This is a heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a heading

This is a paragraph.

### **Example**

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

<!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>

**OUTPUT**

# This is a heading

This is a paragraph.

## **HTML Links - Hyperlinks**

HTML links are hyperlinks.

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

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

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

## **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**

<!DOCTYPE html>

<html>

<body>

<h1>HTML Links</h1>

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

</body>

</html>

**OUTPUT**

# HTML Links

[Visit W3Schools.com!](https://www.w3schools.com/)

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

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

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

## **HTML Links - The target Attribute**

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

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>The target Attribute</h2>

<a href="https://www.w3schools.com/" target="\_blank">Visit W3Schools!</a>

<p>If target="\_blank", the link will open in a new browser window or tab.</p>

</body>

</html>

**OUTPUT**

## The target Attribute

[Visit W3Schools!](https://www.w3schools.com/)

If target="\_blank", the link will open in a new browser window or tab.

## **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**

<!DOCTYPE html>

<html>

<body>

<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>

</body>

</html>

**OUTPUT**

## Absolute URLs

[W3C](https://www.w3.org/)

[Google](https://www.google.com/)

## Relative URLs

[HTML Images](https://www.w3schools.com/html/html_images.asp)

[CSS Tutorial](https://www.w3schools.com/css/default.asp)

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

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

**Example**

<!DOCTYPE html>

<html>

<body>

<h2>Image as a Link</h2>

<p>The image below is a link. Try to click on it.</p>

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

</body>

</html>

**OUTPUT**

## Image as a Link

The image below is a link. Try to click on it.

[HTML tutorial](https://www.w3schools.com/html/default.asp)

## **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**

<!DOCTYPE html>

<html>

<body>

<h2>Link to an Email Address</h2>

<p>To create a link that opens in the user's email program (to let them send a new email), use mailto: inside the href attribute:</p>

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

</body>

</html>

**OUTPUT**

## Link to an Email Address

To create a link that opens in the user's email program (to let them send a new email), use mailto: inside the href attribute:

[Send email](mailto:someone@example.com)

## **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**

<!DOCTYPE html>

<html>

<body>

<h2>Button as a Links</h2>

<p>Click the button to go to the HTML tutorial.</p>

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

</body>

</html>

**OUTPUT**

## Button as a Links

Click the button to go to the HTML tutorial.

## **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**

<!DOCTYPE html>

<html lang="en-US">

<body>

<h2>Link Titles</h2>

<p>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.</p>

<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>

</body>

</html>

**OUTPUT**

## 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.

[Visit our HTML Tutorial](https://www.w3schools.com/html/)

## **More on Absolute URLs and Relative URLs**

<!DOCTYPE html>

<html>

<body>

<h2>External Paths</h2>

<p>This example uses a full URL to link to a web page:</p>

<p><a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a></p>

</body>

</html>

**OUTPUT**

## External Paths

This example uses a full URL to link to a web page:

[HTML tutorial](https://www.w3schools.com/html/default.asp)

### **Example**

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

<!DOCTYPE html>

<html>

<body>

<h2>External Paths</h2>

<p>This example links to a page located in the html folder on the current web site:</p>

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

</body>

</html>

**OUTPUT**

## **External Paths**

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

### **Example**

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

<!DOCTYPE html>

<html>

<body>

<h2>External Paths</h2>

<p>This example links to a page located in the same folder as the current page:</p>

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

</body>

</html>

**OUTPUT**

## External Paths

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

# HTML Links - Different Colors

### **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:

<!DOCTYPE html>

<html>

<head>

<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>

</head>

<body>

<h2>Link Colors</h2>

<p>You can change the default colors of links</p>

<a href="html\_images.asp" target="\_blank">HTML Images</a>

</body>

</html>

**OUTPUT**

## Link Colors

You can change the default colors of links

[HTML Images](https://www.w3schools.com/html/html_images.asp)

## **Link Buttons**

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

[This is a link](javascript:void(0))

<!DOCTYPE html>

<html>

<head>

<style>

a:link, a:visited {

background-color: #f44336;

color: white;

padding: 15px 25px;

text-align: center;

text-decoration: none;

display: inline-block;

}

a:hover, a:active {

background-color: red;

}

</style>

</head>

<body>

<h2>Link Button</h2>

<p>A link styled as a button:</p>

<a href="default.asp" target="\_blank">This is a link</a>

</body>

</html>

**OUTPUT**

## Link Button

A link styled as a button:

[This is a link](https://www.w3schools.com/html/default.asp)

# HTML Links - Create Bookmarks

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

## **Create a Bookmark in HTML**

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

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

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

## **Example**

First, use the id attribute to create a bookmark:

<!DOCTYPE html>

<html>

<body>

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

<p><a href="#C10">Jump to Chapter 10</a></p>

<h2>Chapter 1</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 2</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 3</h2>

<p>This chapter explains ba bla bla</p>

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

<p>This chapter explains ba bla bla</p>

<h2>Chapter 5</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 6</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 7</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 8</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 9</h2>

<p>This chapter explains ba bla bla</p>

<h2 id="C10">Chapter 10</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 11</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 12</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 13</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 14</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 15</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 16</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 17</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 18</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 19</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 20</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 21</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 22</h2>

<p>This chapter explains ba bla bla</p>

<h2>Chapter 23</h2>

<p>This chapter explains ba bla bla</p>

</body>

</html>

**OUTPUT**

[Jump to Chapter 4](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_bookmark#C4)

[Jump to Chapter 10](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_bookmark#C10)

## Chapter 1

This chapter explains ba bla bla

## Chapter 2

This chapter explains ba bla bla

## Chapter 3

This chapter explains ba bla bla

## Chapter 4

This chapter explains ba bla bla

## Chapter 5

This chapter explains ba bla bla

## Chapter 6

This chapter explains ba bla bla

## Chapter 7

This chapter explains ba bla bla

## Chapter 8

This chapter explains ba bla bla

## Chapter 9

This chapter explains ba bla bla

## Chapter 10

This chapter explains ba bla bla

## Chapter 11

This chapter explains ba bla bla

## Chapter 12

This chapter explains ba bla bla

## Chapter 13

This chapter explains ba bla bla

## Chapter 14

This chapter explains ba bla bla

## Chapter 15

This chapter explains ba bla bla

## Chapter 16

This chapter explains ba bla bla

## Chapter 17

This chapter explains ba bla bla

## Chapter 18

This chapter explains ba bla bla

## Chapter 19

This chapter explains ba bla bla

## Chapter 20

This chapter explains ba bla bla

## Chapter 21

This chapter explains ba bla bla

## Chapter 22

This chapter explains ba bla bla

## Chapter 23

This chapter explains ba bla bla

# HTML Images

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

### **Example**

<img src="pic\_trulli.jpg" alt="Italian Trulli">

### **Example**

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

### **Example**

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

## **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"

## **The alt Attribute**

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

The value of the alt attribute should describe the image:

### **Example**

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

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

### **Example**

<img src="wrongname.gif" alt="Flowers in Chania">

**Tip:** A screen reader is a software program that reads the HTML code, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.

## **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;">

Alternatively, you can use the width and height attributes:

### **Example**

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

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

**Note:** Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

## **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

## **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">

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

## **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;">  
</a>

## **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>

**Tip:** To learn more about CSS Float, read our [CSS Float Tutorial](https://www.w3schools.com/css/css_float.asp).

## **Common Image Formats**

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

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

# HTML Image Maps

## **Image Maps**

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

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



<!DOCTYPE html>

<html>

<body>

<h2>Image Maps</h2>

<p>Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:</p>

<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">

<map name="workmap">

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

<area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">

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

</map>

</body>

</html>

## **Image Maps**

Click on the computer, the phone, or the cup of coffee to go to a new page and read more about the topic:



## **How Does it Work?**

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

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

## **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.

**Tip:** You can use any image as an 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">

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



### **Shape="circle"**

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

337,300



Then specify the radius of the circle:

44 pixels



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

### **Example**

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

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



### **Shape="poly"**

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

This can be used to create any shape.

Like maybe a croissant shape!

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



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



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

### **Example**

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

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



## **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:

# HTML Background Images

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h2>Background Image</h2>

<p>A background image for a p element:</p>

<p style="background-image: url('img\_girl.jpg');">

You can specify background images<br>

for any visible HTML element.<br>

In this example, the background image<br>

is specified for a p element.<br>

By default, the background-image<br>

will repeat itself in the direction(s)<br>

where it is smaller than the element<br>

where it is specified. (Try resizing the<br>

browser window to see how the<br>

background image behaves.

</p>

</body>

</html>

**OUTPUT**

## Background Image

A background image for a p element:

You can specify background images  
for any visible HTML element.  
In this example, the background image  
is specified for a p element.  
By default, the background-image  
will repeat itself in the direction(s)  
where it is smaller than the element  
where it is specified. (Try resizing the  
browser window to see how the  
background image behaves.

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<head>

<style>

p {

background-image: url('img\_girl.jpg');

}

</style>

</head>

<body>

<h2>Background Image</h2>

<p>You can specify background images<br>

for any visible HTML element.<br>

In this example, the background image<br>

is specified for a div element.<br>

By default, the background-image<br>

will repeat itself in the direction(s)<br>

where it is smaller than the element<br>

where it is specified. (Try resizing the<br>

browser window to see how the<br>

background image behaves.</p>

</body>

</html

**OUTPUT**

## Background Image

You can specify background images  
for any visible HTML element.  
In this example, the background image  
is specified for a div element.  
By default, the background-image  
will repeat itself in the direction(s)  
where it is smaller than the element  
where it is specified. (Try resizing the  
browser window to see how the  
background image behaves.

## **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:

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url('example\_img\_girl.jpg');

}

</style>

</head>

<body>

<h2>Background Repeat</h2>

<p>By default, the background image will repeat itself if it is smaller than the element where it is specified, in this case the body element.</p>

</body>

</html>

## Background Repeat

By default, the background image will repeat itself if it is smaller than the element where it is specified, in this case the body element.

## **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>

# HTML <picture> Element

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



## **The HTML <picture> Element**

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

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

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

### **Example**

<!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>

## The picture Element



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.

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.

## **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**

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<h2>The picture Element</h2>

<picture>

<source srcset="img\_avatar.png">

<source srcset="img\_girl.jpg">

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

</picture>

<p>The picture element can be used when the image format is not supported by all devices.</p>

<p>The device will use the first image format it supports, and ignore the rest of the images.</p>

</body>

</html>

## The picture Element

The picture element can be used when the image format is not supported by all devices.

The device will use the first image format it supports, and ignore the rest of the images.

# HTML Favicon

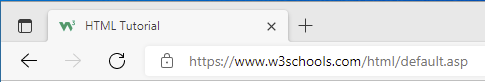
A favicon is a small image displayed next to the page title in the browser tab.

## **How To Add a Favicon in HTML**

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

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

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



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

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

### **Example**

<!DOCTYPE html>  
<html>  
<head>  
  <title>My Page Title</title>  
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">  
</head>  
<body>  
<h1>This is a Heading</h1>  
<p>This is a paragraph.</p>  
</body>  
</html>

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

## **Favicon File Format Support**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Browser** | **ICO** | **PNG** | **GIF** | **JPEG** | **SVG** |
| Edge | Yes | Yes | Yes | Yes | Yes |
| Chrome | Yes | Yes | Yes | Yes | Yes |
| Firefox | Yes | Yes | Yes | Yes | Yes |
| Opera | Yes | Yes | Yes | Yes | Yes |
| Safari | Yes | Yes | Yes | Yes | Ye |

# HTML Page Title

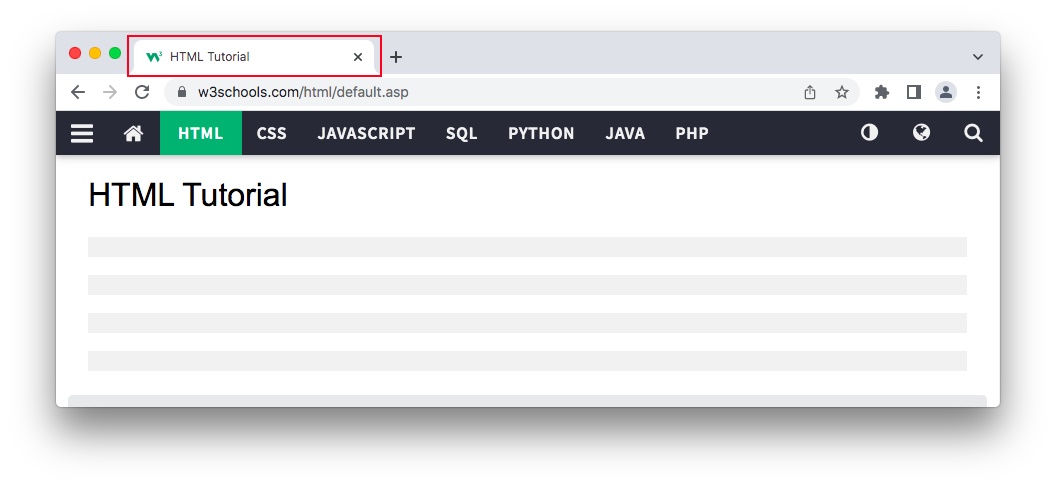
Every web page should have a page title to describe the meaning of the page.

The <title> element adds a title to your page:

### **Example**

<!DOCTYPE html>  
<html>  
<head>  
  <title>HTML Tutorial</title>  
</head>  
<body>  
  
The content of the document......  
  
</body>  
</html>

The title is shown in the browser's title bar:



The title should describe the content and the meaning of the page.

The page title is very important for search engine optimization (SEO). The text 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

# HTML Tables

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

### **Example**

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

## **Define an HTML Table**

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

### **Example**

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

}

</style>

<body>

<h2>A basic HTML table</h2>

<table style="width:100%">

<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>

<p>To understand the example better, we have added borders to the table.</p>

</body>

</html>

## A basic HTML table

|  |  |  |
| --- | --- | --- |
| **Company** | **Contact** | **Country** |
| Alfreds Futterkiste | Maria Anders | Germany |
| Centro comercial Moctezuma | Francisco Chang | Mexico |

To understand the example better, we have added borders to the 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.

### **Example**

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

}

</style>

<body>

<h2>TD elements define table cells</h2>

<table style="width:100%">

<tr>

<td>Emil</td>

<td>Tobias</td>

<td>Linus</td>

</tr>

</table>

<p>To understand the example better, we have added borders to the table.</p>

</body>

</html>

**OUTPUT**

## TD elements define table cells

|  |  |  |
| --- | --- | --- |
| Emil | Tobias | Linus |

To understand the example better, we have added borders to the table.

## **Table Rows**

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

tr stands for table row.

**Example**

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

}

</style>

<body>

<h2>TR elements define table rows</h2>

<table style="width:100%">

<tr>

<td>Emil</td>

<td>Tobias</td>

<td>Linus</td>

</tr>

<tr>

<td>16</td>

<td>14</td>

<td>10</td>

</tr>

</table>

<p>To understand the example better, we have added borders to the table.</p>

</body>

</html>

**OUTPUT**

## TR elements define table rows

|  |  |  |
| --- | --- | --- |
| Emil | Tobias | Linus |
| 16 | 14 | 10 |

To understand the example better, we have added borders to the table.

## **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.

### **Example**

Let the first row be table header cells:

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

}

</style>

<body>

<h2>TH elements define table headers</h2>

<table style="width:100%">

<tr>

<th>Person 1</th>

<th>Person 2</th>

<th>Person 3</th>

</tr>

<tr>

<td>Emil</td>

<td>Tobias</td>

<td>Linus</td>

</tr>

<tr>

<td>16</td>

<td>14</td>

<td>10</td>

</tr>

</table>

<p>To understand the example better, we have added borders to the table.</p>

</body>

</html>

**OUTPUT**

## TH elements define table headers

|  |  |  |
| --- | --- | --- |
| **Person 1** | **Person 2** | **Person 3** |
| Emil | Tobias | Linus |
| 16 | 14 | 10 |

To understand the example better, we have added borders to the table.

# HTML Table Borders

HTML tables can have borders of different styles and shapes.

## **How To Add a Border**

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

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

</style>

</head>

<body>

<h2>Table With Border</h2>

<p>Use the CSS border property to add a border to the table.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

## Table With Border

Use the CSS border property to add a border to the table.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **Collapsed Table Borders**

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

This will make the borders collapse into a single border:

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Collapsed Borders</h2>

<p>If you want the borders to collapse into one border, add the CSS border-collapse property.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

## Collapsed Borders

If you want the borders to collapse into one border, add the CSS border-collapse property.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **Style Table Borders**

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

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid white;

border-collapse: collapse;

}

th, td {

background-color: #96D4D4;

}

</style>

</head>

<body>

<h2>Table With Invisible Borders</h2>

<p>Style the table with white borders and a background color of the cells to make the impression of invisible borders.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Table With Invisible Borders

Style the table with white borders and a background color of the cells to make the impression of invisible borders.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **Round Table Borders**

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

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-radius: 10px;

}

</style>

</head>

<body>

<h2>Table With Rounded Borders</h2>

<p>Use the CSS border-radius property to add rounded corners to the borders.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Table With Rounded Borders

Use the CSS border-radius property to add rounded corners to the borders.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **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**

<!DOCTYPE html>

<html>

<head>

<style>

th, td {

border-style: dotted;

}

</style>

</head>

<body>

<h2>Table With Dotted Borders</h2>

<p>Use the CSS border-style property to set the style of the borders.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Table With Dotted Borders

Use the CSS border-style property to set the style of the borders.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **Border Color**

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

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<head>

<style>

th, td {

border-style:solid;

border-color: #96D4D4;

}

</style>

</head>

<body>

<h2>Table With Border Color</h2>

<p>Use the CSS border-color property to set the color of the borders.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

## Table With Border Color

Use the CSS border-color property to set the color of the borders.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

# 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**

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

border-collapse: collapse;

}

</style>

<body>

<h2>100% wide HTML Table</h2>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## 100% wide HTML Table

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **HTML Table Column Width**

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

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

### **Example**

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

border-collapse: collapse;

}

</style>

<body>

<h2>Set the first column to 70% of the table width</h2>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Set the first column to 70% of the table width

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

## **HTML Table Row Height**

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

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

### **Example**

Set the height of the second row to 200 pixels:

<!DOCTYPE html>

<html>

<style>

table, th, td {

border:1px solid black;

border-collapse: collapse;

}

</style>

<body>

<h2>Set the height of the second row to 200 pixels</h2>

<table style="width:100%">

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Age</th>

</tr>

<tr style="height:200px">

<td>Jill</td>

<td>Smith</td>

<td>50</td>

</tr>

<tr>

<td>Eve</td>

<td>Jackson</td>

<td>94</td>

</tr>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Set the height of the second row to 200 pixels

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

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

|  |  |  |
| --- | --- | --- |
| **EMIL** | **TOBIAS** | **LINUS** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **8:00** |  |  |
| **9:00** |  |  |
| **10:00** |  |  |
| **11:00** |  |  |
| **12:00** |  |  |
| **13:00** |  |  |

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

|  |  |  |
| --- | --- | --- |
| **DECEMBER** | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## **HTML Table Headers**

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

**EXAMPLE**

# <!DOCTYPE html>

# <html>

# <head>

# <style>

# table, th, td {

# border: 1px solid black;

# border-collapse: collapse;

# }

# </style>

# </head>

# <body>

# <h2>Table Headers</h2>

# <p>Use the TH element to define table headers.</p>

# <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>

# </body>

# </html>

# OUTPUT

## Table Headers

Use the TH element to define table headers.

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

## **Vertical Table Headers**

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

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Vertical Table Headers</h2>

<p>The first column becomes table headers if you set the first table cell in each table row to a TH element:</p>

<table style="width:100%">

<tr>

<th>Firstname</th>

<td>Jill</td>

<td>Eve</td>

</tr>

<tr>

<th>Lastname</th>

<td>Smith</td>

<td>Jackson</td>

</tr>

<tr>

<th>Age</th>

<td>50</td>

<td>94</td>

</tr>

</table>

</body>

</html>

## Vertical Table Headers

The first column becomes table headers if you set the first table cell in each table row to a TH element:

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

## **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**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th {

text-align: left;

}

</style>

</head>

<body>

<h2>Left-align Headers</h2>

<p>To left-align the table headers, use the CSS text-align property.</p>

<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>

</body>

</html>

**OUTPUT**

## Left-align Headers

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

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

## **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:

**EXAMPLE**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>A header that spans two columns</h2>

<p>Use the colspan attribute to have a header span over multiple columns.</p>

<table style="width:100%">

<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>

</body>

</html>

**OUTPUT**

## A header that spans two columns

Use the colspan attribute to have a header span over multiple columns.

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

## **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:

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 5px;

text-align: left;

}

</style>

</head>

<body>

<h2>Table Caption</h2>

<p>To add a caption to a table, use the caption tag.</p>

<table style="width:100%">

<caption>Monthly savings</caption>

<tr>

<th>Month</th>

<th>Savings</th>

</tr>

<tr>

<td>January</td>

<td>$100</td>

</tr>

<tr>

<td>February</td>

<td>$50</td>

</tr>

</table>

</body>

</html>

OUTPUT

## Table Caption

To add a caption to a table, use the caption tag.

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

# HTML Table Padding & Spacing

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

|  |  |  |
| --- | --- | --- |
| With Padding | | |
| hello | hello | hello |
| hello | hello | hello |
| hello | hello | hello |

|  |  |  |
| --- | --- | --- |
| With Spacing | | |
| hello | hello | hello |
| hello | hello | hello |
| hello | hello | hello |

## **HTML Table - Cell Padding**

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

By default the padding is set to 0.

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 15px;

}

</style>

</head>

<body>

<h2>Cellpadding</h2>

<p>Cell padding specifies the space between the cell content and its borders.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

<p><strong>Tip:</strong> Try to change the padding to 5px.</p>

</body>

</html>

**OUTPUT**

## Cellpadding

Cell padding specifies the space between the cell content and its borders.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

**Tip:** Try to change the padding to 5px.

## **HTML Table - Cell Spacing**

Cell spacing is the space between each cell.

By default the space is set to 2 pixels.

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

table {

border-spacing: 30px;

}

</style>

</head>

<body>

<h2>Cellspacing</h2>

<p>Change the space between the cells with the border-spacing property.</p>

<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>

<tr>

<td>John</td>

<td>Doe</td>

<td>80</td>

</tr>

</table>

</body>

</html>

## Cellspacing

Change the space between the cells with the border-spacing property.

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Age** |
| Jill | Smith | 50 |
| Eve | Jackson | 94 |
| John | Doe | 80 |

# HTML Table Colspan & Rowspan

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

|  |  |  |
| --- | --- | --- |
| **NAME** | |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **APRIL** |  |  |
|  |  |
|  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **2022** | | |
|  |  |  |
| **FIESTA** | |  |
|  |
|  |  |  |

## **HTML Table - Colspan**

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Cell that spans two columns</h2>

<p>To make a cell span more than one column, use the colspan attribute.</p>

<table style="width:100%">

<tr>

<th colspan="2">Name</th>

<th>Age</th>

</tr>

<tr>

<td>Jill</td>

<td>Smith</td>

<td>43</td>

</tr>

<tr>

<td>Eve</td>

<td>Jackson</td>

<td>57</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

|  |  |  |
| --- | --- | --- |
| **Name** | | **Age** |
| Jill | Smith | 43 |
| Eve | Jackson | 57 |

## **HTML Table - Rowspan**

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Cell that spans two rows</h2>

<p>To make a cell span more than one row, use the rowspan attribute.</p>

<table style="width:100%">

<tr>

<th>Name</th>

<td>Jill</td>

</tr>

<tr>

<th rowspan="2">Phone</th>

<td>555-1234</td>

</tr>

<tr>

<td>555-8745</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

|  |  |
| --- | --- |
| **Name** | Jill |
| **Phone** | 555-1234 |
| 555-8745 |

# HTML Table Styling

Use CSS to make your tables look better.

## **HTML Table - Zebra Stripes**

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

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

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

text-align: left;

padding: 8px;

}

tr:nth-child(even) {

background-color: #D6EEEE;

}

</style>

</head>

<body>

<h2>Zebra Striped Table</h2>

<p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p>

<table>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Points</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

## Zebra Striped Table

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

|  |  |  |
| --- | --- | --- |
| **First Name** | **Last Name** | **Points** |
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| Joe | Swanson | $300 |
| Cleveland | Brown | $250 |

## **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:

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th:nth-child(even),td:nth-child(even) {

background-color: #D6EEEE;

}

</style>

</head>

<body>

<h2>Striped Table</h2>

<p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p>

<table style="width:100%">

<tr>

<th>MON</th>

<th>TUE</th>

<th>WED</th>

<th>THU</th>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Striped Table

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MON** | **TUE** | **WED** | **THU** | **FRI** | **SAT** | **SUN** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## **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**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

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);

}

</style>

</head>

<body>

<h2>Striped Table</h2>

<p>For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:</p>

<table style="width:100%">

<tr>

<th>MON</th>

<th>TUE</th>

<th>WED</th>

<th>THU</th>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Striped Table

For zebra-striped tables, use the nth-child() selector and add a background-color to all even (or odd) table rows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MON** | **TUE** | **WED** | **THU** | **FRI** | **SAT** | **SUN** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## **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:

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 100%;

}

tr {

border-bottom: 1px solid #ddd;

}

</style>

</head>

<body>

<h2>Bordered Table Dividers</h2>

<p>Add the border-bottom property to the tr elements for horizontal dividers:</p>

<table>

<tr>

<th>Firstname</th>

<th>Lastname</th>

<th>Savings</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

**OUTPUT**

## Bordered Table Dividers

Add the border-bottom property to the tr elements for horizontal dividers:

|  |  |  |
| --- | --- | --- |
| **Firstname** | **Lastname** | **Savings** |
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| Joe | Swanson | $300 |
| Cleveland | Brown | $250 |

## **Hoverable Table**

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

|  |  |  |
| --- | --- | --- |
| **First Name** | **Last Name** | **Savings** |
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| Joe | Swanson | $300 |

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table {

border-collapse: collapse;

width: 100%;

}

th, td {

padding: 8px;

text-align: left;

border-bottom: 1px solid #DDD;

}

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

</style>

</head>

<body>

<h2>Hoverable Table</h2>

<p>Move the mouse over the table rows to see the effect.</p>

<table>

<tr>

<th>First Name</th>

<th>Last Name</th>

<th>Points</th>

</tr>

<tr>

<td>Peter</td>

<td>Griffin</td>

<td>$100</td>

</tr>

<tr>

<td>Lois</td>

<td>Griffin</td>

<td>$150</td>

</tr>

<tr>

<td>Joe</td>

<td>Swanson</td>

<td>$300</td>

</tr>

<tr>

<td>Cleveland</td>

<td>Brown</td>

<td>$250</td>

</tr>

</table>

</body>

</html>

## **Hoverable Table**

Move the mouse over the table rows to see the effect.

|  |  |  |
| --- | --- | --- |
| **First Name** | **Last Name** | **Points** |
| Peter | Griffin | $100 |
| Lois | Griffin | $150 |
| Joe | Swanson | $300 |
| Cleveland | Brown | $250 |

# HTML Table Colgroup

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

## **HTML Table Colgroup**

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

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

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

Each group 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.

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Colgroup</h2>

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

<table style="width: 100%;">

<colgroup>

<col span="2" style="background-color: #D6EEEE">

</colgroup>

<tr>

<th>MON</th>

<th>TUE</th>

<th>WED</th>

<th>THU</th>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

</tr>

<tr>

<td>8</td>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

<td>13</td>

<td>14</td>

</tr>

<tr>

<td>15</td>

<td>16</td>

<td>17</td>

<td>18</td>

<td>19</td>

<td>20</td>

<td>21</td>

</tr>

<tr>

<td>22</td>

<td>23</td>

<td>24</td>

<td>25</td>

<td>26</td>

<td>27</td>

<td>28</td>

</tr>

</table>

</body>

</html>

## Colgroup

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

**OUTPUT**

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

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Colgroup</h2>

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

<table style="width: 100%;">

<colgroup>

<col span="2" style="background-color: #D6EEEE">

</colgroup>

<tr>

<th>MON</th>

<th>TUE</th>

<th>WED</th>

<th>THU</th>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

</tr>

<tr>

<td>8</td>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

<td>13</td>

<td>14</td>

</tr>

<tr>

<td>15</td>

<td>16</td>

<td>17</td>

<td>18</td>

<td>19</td>

<td>20</td>

<td>21</td>

</tr>

<tr>

<td>22</td>

<td>23</td>

<td>24</td>

<td>25</td>

<td>26</td>

<td>27</td>

<td>28</td>

</tr>

</table>

</body>

</html>

## Colgroup

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

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

## **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>:

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Multiple Col Elements</h2>

<p>Add multiple col elements in the colgroup:</p>

<table style="width: 100%;">

<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>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

</tr>

<tr>

<td>8</td>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

<td>13</td>

<td>14</td>

</tr>

<tr>

<td>15</td>

<td>16</td>

<td>17</td>

<td>18</td>

<td>19</td>

<td>20</td>

<td>21</td>

</tr>

<tr>

<td>22</td>

<td>23</td>

<td>24</td>

<td>25</td>

<td>26</td>

<td>27</td>

<td>28</td>

</tr>

</table>

</body>

</html>

## Multiple Col Elements

Add multiple col elements in the colgroup:

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

## **Empty Colgroups**

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Empty Colgroups</h2>

<p>Add "empty" col elements that represents the columns before the columns you want to style:</p>

<table style="width: 100%;">

<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>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

</tr>

<tr>

<td>8</td>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

<td>13</td>

<td>14</td>

</tr>

<tr>

<td>15</td>

<td>16</td>

<td>17</td>

<td>18</td>

<td>19</td>

<td>20</td>

<td>21</td>

</tr>

<tr>

<td>22</td>

<td>23</td>

<td>24</td>

<td>25</td>

<td>26</td>

<td>27</td>

<td>28</td>

</tr>

</table>

</body>

</html>

## Empty Colgroups

Add "empty" col elements that represents the columns before the columns you want to style:

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

## **Hide Columns**

You can hide columns with the visibility: collapse property:

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

</style>

</head>

<body>

<h2>Hide Columns</h2>

<p>You can hide specific columns with the visibility property:</p>

<table style="width: 100%;">

<colgroup>

<col span="2">

<col span="3" style="visibility: collapse">

</colgroup>

<tr>

<th>MON</th>

<th>TUE</th>

<th>WED</th>

<th>THU</th>

<th>FRI</th>

<th>SAT</th>

<th>SUN</th>

</tr>

<tr>

<td>1</td>

<td>2</td>

<td>3</td>

<td>4</td>

<td>5</td>

<td>6</td>

<td>7</td>

</tr>

<tr>

<td>8</td>

<td>9</td>

<td>10</td>

<td>11</td>

<td>12</td>

<td>13</td>

<td>14</td>

</tr>

<tr>

<td>15</td>

<td>16</td>

<td>17</td>

<td>18</td>

<td>19</td>

<td>20</td>

<td>21</td>

</tr>

<tr>

<td>22</td>

<td>23</td>

<td>24</td>

<td>25</td>

<td>26</td>

<td>27</td>

<td>28</td>

</tr>

</table>

<p><b>Note:</b> The table columns does not collapse properly in Safari browsers.</p>

</body>

</html>

## Hide Columns

You can hide specific columns with the visibility property:

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

**Note:** The table columns does not collapse properly in Safari browsers.

# HTML Lists

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

### **Example**

An unordered HTML list:

* Item
* Item
* Item
* Item

An ordered HTML list:

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h2>An Unordered HTML List</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

<h2>An Ordered HTML List</h2>

<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

**OUTPUT**

## An Unordered HTML List

* Coffee
* Tea
* Milk

## An Ordered HTML List

1. Coffee
2. Tea
3. Milk

## **Unordered HTML List**

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

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

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>An unordered HTML list</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

**OUTPUT**

## An unordered HTML list

* Coffee
* Tea
* Milk

## **HTML Description Lists**

HTML also supports description lists.

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

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

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>A Description List</h2>

<dl>

<dt>Coffee</dt>

<dd>- black hot drink</dd>

<dt>Milk</dt>

<dd>- white cold drink</dd>

</dl>

</body>

</html>

## A Description List

Coffee

- black hot drink

Milk

- white cold drink

# HTML Unordered Lists

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

## **Unordered HTML List**

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

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

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>An unordered HTML list</h2>

<ul>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

## An unordered HTML list

* Coffee
* Tea
* Milk

## **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 |

### **Example - Disc**

<!DOCTYPE html>

<html>

<body>

<h2>Unordered List with Disc Bullets</h2>

<ul style="list-style-type:disc;">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

## Unordered List with Disc Bullets

* Coffee
* Tea
* Milk<!DOCTYPE html>

<html>

<body>

<h2>Unordered List with Circle Bullets</h2>

<ul style="list-style-type:circle;">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

## Unordered List with Circle Bullets

* Coffee
* Tea
* Milk

### **Example - Circle**

<!DOCTYPE html>

<html>

<body>

<h2>Unordered List with Circle Bullets</h2>

<ul style="list-style-type:circle;">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

## Unordered List with Circle Bullets

* Coffee
* Tea
* Milk

### **Example - Square**

<!DOCTYPE html>

<html>

<body>

<h2>Unordered List with Square Bullets</h2>

<ul style="list-style-type:square;">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

## Unordered List with Square Bullets

* Coffee
* Tea
* Milk

### **Example - None**

<!DOCTYPE html>

<html>

<body>

<h2>Unordered List without Bullets</h2>

<ul style="list-style-type:none;">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ul>

</body>

</html>

## Unordered List without Bullets

* Coffee
* Tea
* Milk

## **Nested HTML Lists**

Lists can be nested (list inside list):

<!DOCTYPE html>

<html>

<body>

<h2>A Nested List</h2>

<p>Lists can be nested (list inside list):</p>

<ul>

<li>Coffee</li>

<li>Tea

<ul>

<li>Black tea</li>

<li>Green tea</li>

</ul>

</li>

<li>Milk</li>

</ul>

</body>

</html>

**EXAMPLE**

## A Nested List

Lists can be nested (list inside list):

* Coffee
* Tea
  + Black tea
  + Green tea
* Milk

## **Horizontal List with CSS**

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

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333333;

}

li {

float: left;

}

li a {

display: block;

color: white;

text-align: center;

padding: 16px;

text-decoration: none;

}

li a:hover {

background-color: #111111;

}

</style>

</head>

<body>

<h2>Navigation Menu</h2>

<p>In this example, we use CSS to style the list horizontally, to create a navigation menu:</p>

<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>

## **Navigation Menu**

In this example, we use CSS to style the list horizontally, to create a navigation menu:

# HTML Ordered Lists

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

## **Ordered HTML List**

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

The list items will be marked with numbers by default:

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>An ordered HTML list</h2>

<ol>

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

## An ordered HTML list

1. Coffee
2. Tea
3. Milk

## **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:**

<!DOCTYPE html>

<html>

<body>

<h2>Ordered List with Numbers</h2>

<ol type="1">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

## Ordered List with Numbers

1. Coffee
2. Tea
3. Milk

### **Uppercase Letters:**

<!DOCTYPE html>

<html>

<body>

<h2>Ordered List with Letters</h2>

<ol type="A">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

**EXAMPLE**

## Ordered List with Letters

1. Coffee
2. Tea
3. Milk

### **Lowercase Letters:**

<!DOCTYPE html>

<html>

<body>

<h2>Ordered List with Lowercase Letters</h2>

<ol type="a">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

## Ordered List with Lowercase Letters

1. Coffee
2. Tea
3. Milk

### **Uppercase Roman Numbers:**

<!DOCTYPE html>

<html>

<body>

<h2>Ordered List with Roman Numbers</h2>

<ol type="I">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

**OUTPUT**

## Ordered List with Roman Numbers

1. Coffee
2. Tea
3. Milk

### **Lowercase Roman Numbers:**

<!DOCTYPE html>

<html>

<body>

<h2>Ordered List with Lowercase Roman Numbers</h2>

<ol type="i">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

## **Ordered List with Lowercase Roman Numbers**

1. Coffee
2. Tea
3. Milk

## **Control List Counting**

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

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h2>The start attribute</h2>

<p>By default, an ordered list will start counting from 1. Use the start attribute to start counting from a specified number:</p>

<ol start="50">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

<ol type="I" start="50">

<li>Coffee</li>

<li>Tea</li>

<li>Milk</li>

</ol>

</body>

</html>

## **The start attribute**

By default, an ordered list will start counting from 1. Use the start attribute to start counting from a specified number:

1. Coffee
2. Tea
3. Milk
4. Coffee
5. Tea
6. Milk

## **Nested HTML Lists**

Lists can be nested (list inside list):

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>A Nested List</h2>

<p>Lists can be nested (list inside list):</p>

<ol>

<li>Coffee</li>

<li>Tea

<ol>

<li>Black tea</li>

<li>Green tea</li>

</ol>

</li>

<li>Milk</li>

</ol>

</body>

</html>

## **OUTPUT**

## A Nested List

Lists can be nested (list inside list):

1. Coffee
2. Tea
   1. Black tea
   2. Green tea
3. Milk

# HTML Other Lists

HTML also supports description lists.

## **HTML Description Lists**

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

The [<dl>](https://www.w3schools.com/tags/tag_dl.asp) tag defines the description list, the [<dt>](https://www.w3schools.com/tags/tag_dt.asp) tag defines the term (name), and the [<dd>](https://www.w3schools.com/tags/tag_dd.asp) tag describes each term:

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>A Description List</h2>

<dl>

<dt>Coffee</dt>

<dd>- black hot drink</dd>

<dt>Milk</dt>

<dd>- white cold drink</dd>

</dl>

</body>

</html>

## **A Description List**

Coffee

- black hot drink

Milk

- white cold drink

# HTML Block and Inline Elements

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

The two most common display values are block and inline.

## **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**

# <!DOCTYPE html>

# <html>

# <body>

# <p style="border: 1px solid black">Hello World</p>

# <div style="border: 1px solid black">Hello World</div>

# <p>The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).</p>

# </body>

# </html>

# OUTPUT

Hello World

Hello World

The P and the DIV elements are both block elements, and they will always start on a new line and take up the full width available (stretches out to the left and right as far as it can).

## **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.

# <!DOCTYPE html>

# <html>

# <body>

# <p>This is an inline span <span style="border: 1px solid black">Hello World</span> element inside a paragraph.</p>

# <p>The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.</p>

# </body>

# </html>

# OUTPUT

This is an inline span Hello World element inside a paragraph.

The SPAN element is an inline element, and will not start on a new line and only takes up as much width as necessary.

## **The <div> Element**

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

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

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

### **Example**

<!DOCTYPE html>

<html>

<body>

<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>

<p>Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.</p>

</div>

</body>

</html>

## **The <span> Element**

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

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

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

### **Example**

<!DOCTYPE html>

<html>

<body>

<h1>The span element</h1>

<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>

</body>

</html>

# The span element

My mother has **blue** eyes and my father has **dark green** eyes.

# HTML Div Element

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

## **The <div> Element**

The <div> element is by default a block element, meaning that it takes all available width, and comes with line breaks before and after.

<!DOCTYPE html>

<html>

<style>

div {

background-color: #FFF4A3;

}

</style>

<body>

<h1>HTML DIV Example</h1>

Lorem Ipsum <div>I am a div</div> dolor sit amet.

<p>The yellow background is added to demonstrate the footprint of the DIV element.</p>

</body>

</html>

# HTML DIV Example

Lorem Ipsum

I am a div

dolor sit amet.

The yellow background is added to demonstrate the footprint of the DIV element.

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

## **<div> as a container**

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

### **Example**

A <div> element with HTML elements:

<!DOCTYPE html>

<html>

<style>

div {

background-color: #FFF4A3;

}

</style>

<body>

<h1>HTML DIV Example</h1>

<div>

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

<p>The yellow background is added to demonstrate the footprint of the DIV element.</p>

</body>

</html>

# HTML DIV Example

## London

London is the capital city of England.

London has over 13 million inhabitants.

The yellow background is added to demonstrate the footprint of the DIV element.

## **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.

### **Example**

<!DOCTYPE html>

<html>

<style>

div {

width: 300px;

margin: auto;

background-color: #FFF4A3;

}

</style>

<body>

<h1>Center align a DIV element</h1>

<div>

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

</body>

</html>

**OUTPUT**

# Center align a DIV element

## London

London is the capital city of England.

London has over 13 million inhabitants.

## **Multiple <div> elements**

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

### **Example**

<div>  
  <h2>London</h2>  
  <p>London is the capital city of England.</p>  
  <p>London has over 13 million inhabitants.</p>  
</div>  
  
<div>  
  <h2>Oslo</h2>  
  <p>Oslo is the capital city of Norway.</p>  
  <p>Oslo has over 600.000 inhabitants.</p>  
</div>  
  
<div>  
  <h2>Rome</h2>  
  <p>Rome is the capital city of Italy.</p>  
  <p>Rome has almost 3 million inhabitants.</p>  
</div>

<!DOCTYPE html>

<html>

<body>

<h1>Multiple DIV Elements</h1>

<div style="background-color:#FFF4A3;">

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

<div style="background-color:#FFC0C7;">

<h2>Oslo</h2>

<p>Oslo is the capital city of Norway.</p>

<p>Oslo has over 600.000 inhabitants.</p>

</div>

<div style="background-color:#D9EEE1;">

<h2>Rome</h2>

<p>Rome is the capital city of Italy.</p>

<p>Rome has almost 3 million inhabitants.</p>

</div>

<p>CSS styles are added to make it easier to separate the divs, and to make them more pretty:)</p>

</body>

</html>

# Multiple DIV Elements

## London

London is the capital city of England.

London has over 13 million inhabitants.

## Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

## Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

CSS styles are added to make it easier to separate the divs, and to meake them more pretty:)

## **Aligning <div> elements side by side**

When building web pages, you often want to have two or more <div> elements side by side, like this:

## **London**

London is the capital city of England.

London has over 13 million inhabitants.

## **Oslo**

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

## **Rome**

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

There are different methods for aligning elements side by side, all include some CSS styling. We will look at the most common methods:

## **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.

### **Example**

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

<!DOCTYPE html>

<html>

<style>

div.mycontainer {

width:100%;

overflow:auto;

}

div.mycontainer div {

width:33%;

float:left;

}

</style>

<body>

<div class="mycontainer">

<div style="background-color:#FFF4A3;">

<h2>London</h2>

<p>London is the capital city of England.</p>

<p>London has over 13 million inhabitants.</p>

</div>

<div style="background-color:#FFC0C7;">

<h2>Oslo</h2>

<p>Oslo is the capital city of Norway.</p>

<p>Oslo has over 600.000 inhabitants.</p>

</div>

<div style="background-color:#D9EEE1;">

<h2>Rome</h2>

<p>Rome is the capital city of Italy.</p>

<p>Rome has almost 3 million inhabitants.</p>

</div>

</div>

</body>

</html>

**OUTPUT**

## London

London is the capital city of England.

London has over 13 million inhabitants.

## Oslo

Oslo is the capital city of Norway.

Oslo has over 600.000 inhabitants.

## Rome

Rome is the capital city of Italy.

Rome has almost 3 million inhabitants.

# HTML class Attribute

The HTML class attribute is used to specify a class for an HTML element.

Multiple HTML elements can share the same class.

## **Using The class Attribute**

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

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<style>

.city {

background-color: tomato;

color: white;

border: 2px solid black;

margin: 20px;

padding: 20px;

}

</style>

</head>

<body>

<div class="city">

<h2>London</h2>

<p>London is the capital of England.</p>

</div>

<div class="city">

<h2>Paris</h2>

<p>Paris is the capital of France.</p>

</div>

<div class="city">

<h2>Tokyo</h2>

<p>Tokyo is the capital of Japan.</p>

</div>

</body>

</html>

**OUTPUT**

## London

London is the capital of England.

## Paris

Paris is the capital of France.

## Tokyo

Tokyo is the capital of Japan.

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

### **Example**

<!DOCTYPE html>  
<html>  
<head>  
<style>  
.note {  
  font-size: 120%;  
  color: red;  
}  
</style>  
</head>  
<body>  
  
<h1>My <span class="note">Important</span> Heading</h1>  
<p>This is some <span class="note">important</span> text.</p>  
  
</body>  
</html>

**OUTPUT**

# My Important Heading

This is some important text.

## **The Syntax For Class**

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

### **Example**

Create a class named "city":

<!DOCTYPE html>  
<html>  
<head>  
<style>  
.city {  
  background-color: tomato;  
  color: white;  
  padding: 10px;  
}  
</style>  
</head>  
<body>  
  
<h2 class="city">London</h2>  
<p>London is the capital of England.</p>  
  
<h2 class="city">Paris</h2>  
<p>Paris is the capital of France.</p>  
  
<h2 class="city">Tokyo</h2>  
<p>Tokyo is the capital of Japan.</p>  
  
</body>  
</html>

## **Multiple Classes**

HTML elements can belong to more than one class.

To define multiple classes, separate the class names with a space, e.g. <div class="city main">. The element will be styled according to all the classes specified.

In the following example, the first <h2> element belongs to both the city class and also to the main class, and will get the CSS styles from both of the classes:

### **Example**

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

## **Different Elements Can Share Same Class**

Different HTML elements can point to the same class name.

In the following example, both <h2> and <p> point to the "city" class and will share the same style:

### **Example**

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

## **Use of The class Attribute in JavaScript**

The class name can also be used by JavaScript to perform certain tasks for specific elements.

JavaScript can access elements with a specific class name with the getElementsByClassName() method:

### **Example**

Click on a button to hide all elements with the class name "city":

<script>  
function myFunction() {  
  var x = **document.getElementsByClassName("city")**;  
  for (var i = 0; i < x.length; i++) {  
    x[i].style.display = "none";  
  }  
}  
</script>

# HTML id Attribute

The HTML id attribute is used to specify a unique id for an HTML element.

You cannot have more than one element with the same id in an HTML document.

## **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:

### **Example**

<style>  
/\* Style the element with the id "myHeader" \*/  
**#myHeader** {  
  background-color: lightblue;  
  color: black;  
  padding: 40px;  
  text-align: center;  
}  
  
/\* Style all elements with the class name "city" \*/  
**.city**{  
  background-color: tomato;  
  color: white;  
  padding: 10px;  
}  
</style>  
<!-- An element with a unique id -->  
<h1 id="myHeader">My Cities</h1>  
  
<!-- Multiple elements with same class -->  
<h2 class="city">London</h2>  
<p>London is the capital of England.</p>  
<h2 class="city">Paris</h2>  
<p>Paris is the capital of France.</p>  
  
<h2 class="city">Tokyo</h2>  
<p>Tokyo is the capital of Japan.</p>

## **HTML Bookmarks with ID and Links**

HTML bookmarks are used to allow readers to jump to specific parts of a webpage.

Bookmarks can be useful if your page is very long.

To use a bookmark, you must first create it, and then add a link to it.

Then, when the link is clicked, the page will scroll to the location with the bookmark.

## **Example**

First, create a bookmark with the id attribute:

<h2 id="C4">Chapter 4</h2>

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

### **Example**

<a href="#C4">Jump to Chapter 4</a>

Or, add a link to the bookmark ("Jump to Chapter 4"), from another page:

<a href="html\_demo.html#C4">Jump to Chapter 4</a>

## **Using The id Attribute in JavaScript**

The id attribute can also be used by JavaScript to perform some tasks for that specific element.

JavaScript can access an element with a specific id with the getElementById() method:

### **Example**

Use the id attribute to manipulate text with JavaScript:

<script>  
function displayResult() {  
  document.getElementById("myHeader").innerHTML = "Have a nice day!";  
}  
</script>

# HTML Iframes

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

## **HTML Iframe Syntax**

The HTML <iframe> tag specifies an inline frame.

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

### **Syntax**

<iframe src="*url*" title="description"></iframe>

**Tip:** It is a good practice to always include a title attribute for the <iframe>. This is used by screen readers to read out what the content of the iframe is.

## **Iframe - Set Height and Width**

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

The height and width are specified in pixels by default:

### **Example**

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

Or you can add the style attribute and use the CSS height and width properties:

### **Example**

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

## **Iframe - Remove the Border**

By default, an iframe has a border around it.

To remove the border, add the style attribute and use the CSS border property:

### **Example**

<iframe src="demo\_iframe.htm" style="border:none;" title="Iframe Example"></iframe>

With CSS, you can also change the size, style and color of the iframe's border:

### **Example**

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

## **Iframe - Target for a Link**

An iframe can be used as the target frame for a link.

The target attribute of the link must refer to the name attribute of the iframe:

### **Example**

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

## **HTML iframe Tag**

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<iframe>](https://www.w3schools.com/tags/tag_iframe.asp) | Defines an inline frame |

# HTML JavaScript

JavaScript makes HTML pages more dynamic and interactive.

### **Example**

## **My First JavaScript**

Click me to display Date and Time

## **The HTML <script> Tag**

The HTML <script> tag is used to define a client-side script (JavaScript).

The <script> element either contains script statements, or it points to an external script file through the src attribute.

Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

To select an HTML element, JavaScript most often uses the document.getElementById() method.

This JavaScript example writes "Hello JavaScript!" into an HTML element with id="demo":

### **Example**

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

## **A Taste of JavaScript**

Here are some examples of what JavaScript can do:

### **Example**

JavaScript can change content:

document.getElementById("demo").innerHTML = "Hello JavaScript!";

### **Example**

JavaScript can change styles:

document.getElementById("demo").style.fontSize = "25px";  
document.getElementById("demo").style.color = "red";  
document.getElementById("demo").style.backgroundColor = "yellow";

### **Example**

JavaScript can change attributes:

document.getElementById("image").src = "picture.gif";

## **The HTML <noscript> Tag**

The HTML <noscript> tag defines an alternate content to be displayed to users that have disabled scripts in their browser or have a browser that doesn't support scripts:

### **Example**

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

p> element to "Hello World!".

## **HTML Script Tags**

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<script>](https://www.w3schools.com/tags/tag_script.asp) | Defines a client-side script |
| [<noscript>](https://www.w3schools.com/tags/tag_noscript.asp) | Defines an alternate content for users that do not support client-side scripts |

# HTML File Paths

A file path describes the location of a file in a web site's folder structure.

## **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**

An absolute file path is the full URL to a file:

### **Example**

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

The <img> tag is explained in the chapter: [HTML Images](https://www.w3schools.com/html/html_images.asp)

## **Relative File Paths**

A relative file path points to a file relative to the current page.

In the following example, the file path points to a file in the images folder located at the root of the current web:

### **Example**

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

In the following example, the file path points to a file in the images folder located in the current folder:

### **Example**

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

In the following example, the file path points to a file in the images folder located in the folder one level up from the current folder:

### **Example**

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

# HTML - The Head Element

The HTML <head> element is a container for the following elements: <title>, <style>, <meta>, <link>, <script>, and <base>.

## **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

A simple HTML document:

<!DOCTYPE html>

<html>

<head>

<title>A Meaningful Page Title</title>

</head>

<body>

<p>The content of the body element is displayed in the browser window.</p>

<p>The content of the title element is displayed in the browser tab, in favorites and in search-engine results.</p>

</body>

</html>

**OUTPUT**

The content of the body element is displayed in the browser window.

The content of the title element is displayed in the browser tab, in favorites and in search-engine results.

## **The HTML <style> Element**

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

### **Example**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

<style>

body {background-color: powderblue;}

h1 {color: red;}

p {color: blue;}

</style>

</head>

<body>

<h1>This is a Heading</h1>

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

<p>The content of the body element is displayed in the browser window.</p>

<p>The content of the title element is displayed in the browser tab, in favorites and in search-engine results.</p>

</body>

</html>

**OUTPUT**

# This is a Heading

This is a paragraph.

The content of the body element is displayed in the browser window.

The content of the title element is displayed in the browser tab, in favorites and in search-engine results.

## **The HTML <link> Element**

The <link> element defines the relationship between the current document and an external resource.  
  
The <link> tag is most often used to link to external style sheets:

### **Example**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

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

</head>

<body>

<h1>This is a Heading</h1>

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

</body>

</html>

**OUTPUT**

# This is a Heading

This is a paragraph.

## **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.

## **Examples**

**Define the character set used:**

<meta charset="UTF-8">

**Define keywords for search engines:**

<meta name="keywords" content="HTML, CSS, JavaScript">

**Define a description of your web page:**

<meta name="description" content="Free Web tutorials">

**Define the author of a page:**

<meta name="author" content="John Doe">

**Refresh document every 30 seconds:**

<meta http-equiv="refresh" content="30">

**Setting the viewport to make your website look good on all devices:**

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

**Example of <meta> tags:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="description" content="Free Web tutorials">

<meta name="keywords" content="HTML, CSS, JavaScript">

<meta name="author" content="John Doe">

</head>

<body>

<p>All meta information goes inside the head section.</p>

</body>

</html>

**OUTPUT**

All meta information goes inside the head section.

## **Setting The Viewport**

The viewport is the user's visible area of a web page. It varies with the device - it will be smaller on a mobile phone than on a computer screen.

You should include the following <meta> element in all your web pages:

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

This gives the browser instructions on how to control the page's dimensions and scaling.

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page without the viewport meta tag, and the same web page with the viewport meta tag:

[[](https://www.w3schools.com/html/example_withoutviewport.htm)  
  
**Without the viewport meta tag**](https://www.w3schools.com/html/example_withoutviewport.htm)

[[](https://www.w3schools.com/html/example_withviewport.htm)  
  
**With the viewport meta tag**](https://www.w3schools.com/html/example_withviewport.htm)

## **The HTML <script> Element**

The <script> element is used to define client-side JavaScripts.

The following JavaScript writes "Hello JavaScript!" into an HTML element with id="demo":

### **Example**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

<script>

function myFunction() {

document.getElementById("demo").innerHTML = "Hello JavaScript!";

}

</script>

</head>

<body>

<h1>My Web Page</h1>

<p id="demo">A Paragraph</p>

<button type="button" onclick="myFunction()">Try it</button>

</body>

</html>

**OUTPUT**

# My Web Page

A Paragraph

Try it

## **The HTML <base> Element**

The <base> element specifies the base URL and/or target for all relative URLs in a page.

The <base> tag must have either an href or a target attribute present, or both.

There can only be one single <base> element in a document!

### **Example**

Specify a default URL and a default target for all links on a page:

<!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>

# The base element

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".

[HTML base tag](https://www.w3schools.com/tags/tag_base.asp) - 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".

# HTML Symbols

Symbols or letters that are not present on your keyboard can be added to HTML using entities.

## HTML Symbol Entities

HTML entities were described in the previous chapter.

Many mathematical, technical, and currency symbols, are not present on a normal keyboard.

To add such symbols to an HTML page, you can use the entity name or the entity number (a decimal or a hexadecimal reference) for the symbol:

### **Example**

Display the euro sign:

<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 |

# Using Emojis in HTML

Emojis are characters from the UTF-8 character set: 😄 😍 💗

|  |  |
| --- | --- |
| **Emoji** | **Value** |
| 🗻 | &#128507; |
| 🗼 | &#128508; |
| 🗽 | &#128509; |
| 🗾 | &#128510; |
| 🗿 | &#128511; |
| 😀 | &#128512; |
| 😁 | &#128513; |
| 😂 | &#128514; |
| 😃 | &#128515; |
| 😄 | &#128516; |
| 😅 | &#128517; |

## HTML Emojis Examples

🚀🚁🚂🚃🚄

[HTML Emoji Transport Symbols](https://www.w3schools.com/charsets/ref_emoji_transport.asp)

💺💻💼💽💾

[HTML Emoji Office Symbols](https://www.w3schools.com/charsets/ref_emoji_office.asp)

👮👯👰👱👲

[HTML Emoji People Symbols](https://www.w3schools.com/charsets/ref_emoji_body.asp)

🐂🐃🐄🐅🐆

[HTML Emoji Animals Symbols](https://www.w3schools.com/charsets/ref_emoji_animals.asp)

## What are Emojis?

Emojis look like images, or icons, but they are not.

They are letters (characters) from the UTF-8 (Unicode) character set.

UTF-8 covers almost all of the characters and symbols in the world.

## The HTML charset Attribute

To display an HTML page correctly, a web browser must know the character set used in the page.

This is specified in the <meta> tag:

<meta charset="UTF-8">

If not specified, UTF-8 is the default character set in HTML.

## UTF-8 Characters

Many UTF-8 characters cannot be typed on a keyboard, but they can always be displayed using numbers (called entity numbers):

* A is 65
* B is 66
* C is 67

### Example

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

### Example Explained

The <meta charset="UTF-8"> element defines the character set.

The characters A, B, and C, are displayed by the numbers 65, 66, and 67.

To let the browser understand that you are displaying a character, you must start the entity number with &# and end it with ; (semicolon).

## Emoji Characters

Emojis are also characters from the UTF-8 alphabet:

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

### Example

<!DOCTYPE html>  
<html>  
<meta charset="UTF-8">  
<body>  
  
<h1>My First Emoji</h1>  
  
<p>&#128512;</p>  
  
</body>  
</html>

Since Emojis are characters, they can be copied, displayed, and sized just like any other character in HTML.

### Example

<!DOCTYPE html>  
<html>  
<meta charset="UTF-8">  
<body>  
  
<h1>Sized Emojis</h1>  
  
<p style="font-size:48px">  
&#128512; &#128516; &#128525; &#128151;  
</p>  
  
</body>  
</html>

# HTML Encoding (Character Sets)

To display an HTML page correctly, a web browser must know which character set to use.

## The HTML charset Attribute

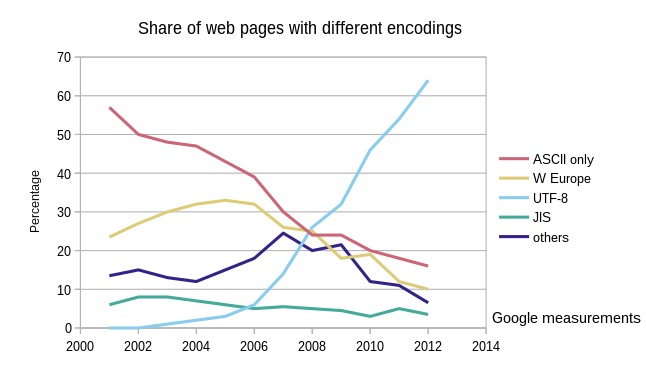
The character set is specified in the <meta> tag:

### Example

<meta charset="UTF-8">

The HTML5 specification encourages web developers to use the UTF-8 character set.

UTF-8 covers almost all of the characters and symbols in the world!

[](https://commons.wikimedia.org/wiki/File:Unicode_Web_growth.svg)

[Full UTF-8 Reference](https://www.w3schools.com/charsets/ref_utf_basic_latin.asp)

## **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">

## Differences Between Character Sets

The following table displays the differences between the character sets described above:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Numb** | **ASCII** | **ANSI** | **8859** | **UTF‑8** | **Description** |
| 32 |  |  |  |  | space |
| 33 | ! | ! | ! | ! | exclamation mark |
| 34 | " | " | " | " | quotation mark |
| 35 | # | # | # | # | number sign |
| 36 | $ | $ | $ | $ | dollar sign |
| 37 | % | % | % | % | percent sign |
| 38 | & | & | & | & | ampersand |
| 39 | ' | ' | ' | ' | apostrophe |
| 40 | ( | ( | ( | ( | left parenthesis |
| 41 | ) | ) | ) | ) | right parenthesis |
| 42 | \* | \* | \* | \* | asterisk |
| 43 | + | + | + | + | plus sign |
| 44 | , | , | , | , | comma |
| 45 | - | - | - | - | hyphen-minus |
| 46 | . | . | . | . | full stop |
| 47 | / | / | / | / | solidus |
| 48 | 0 | 0 | 0 | 0 | digit zero |
| 49 | 1 | 1 | 1 | 1 | digit one |
| 50 | 2 | 2 | 2 | 2 | digit two |
| 51 | 3 | 3 | 3 | 3 | digit three |
| 52 | 4 | 4 | 4 | 4 | digit four |
| 53 | 5 | 5 | 5 | 5 | digit five |
| 54 | 6 | 6 | 6 | 6 | digit six |
| 55 | 7 | 7 | 7 | 7 | digit seven |
| 56 | 8 | 8 | 8 | 8 | digit eight |
| 57 | 9 | 9 | 9 | 9 | digit nine |
| 58 | : | : | : | : | colon |
| 59 | ; | ; | ; | ; | semicolon |
| 60 | < | < | < | < | less than |
| 61 | = | = | = | = | equals sign |
| 62 | > | > | > | > | greater than |
| 63 | ? | ? | ? | ? | question mark |
| 64 | @ | @ | @ | @ | commercial at |
| 65 | A | A | A | A | Latin A |
| 66 | B | B | B | B | Latin B |
| 67 | C | C | C | C | Latin C |
| 68 | D | D | D | D | Latin D |
| 69 | E | E | E | E | Latin E |
| 70 | F | F | F | F | Latin F |
| 71 | G | G | G | G | Latin G |
| 72 | H | H | H | H | Latin H |
| 73 | I | I | I | I | Latin I |
| 74 | J | J | J | J | Latin J |
| 75 | K | K | K | K | Latin K |
| 76 | L | L | L | L | Latin L |
| 77 | M | M | M | M | Latin M |
| 78 | N | N | N | N | Latin N |
| 79 | O | O | O | O | Latin O |
| 80 | P | P | P | P | Latin P |
| 81 | Q | Q | Q | Q | Latin Q |
| 82 | R | R | R | R | Latin R |
| 83 | S | S | S | S | Latin S |
| 84 | T | T | T | T | Latin T |
| 85 | U | U | U | U | Latin U |
| 86 | V | V | V | V | Latin V |
| 87 | W | W | W | W | Latin W |
| 88 | X | X | X | X | Latin X |
| 89 | Y | Y | Y | Y | Latin Y |
| 90 | Z | Z | Z | Z | Latin Z |
| 91 | [ | [ | [ | [ | left square bracket |
| 92 | \ | \ | \ | \ | reverse solidus |
| 93 | ] | ] | ] | ] | right square bracket |
| 94 | ^ | ^ | ^ | ^ | circumflex accent |
| 95 | \_ | \_ | \_ | \_ | low line |
| 96 | ` | ` | ` | ` | grave accent |
| 97 | a | a | a | a | Latin small a |
| 98 | b | b | b | b | Latin small b |
| 99 | c | c | c | c | Latin small c |
| 100 | d | d | d | d | Latin small d |
| 101 | e | e | e | e | Latin small e |
| 102 | f | f | f | f | Latin small f |
| 103 | g | g | g | g | Latin small g |
| 104 | h | h | h | h | Latin small h |
| 105 | i | i | i | i | Latin small i |
| 106 | j | j | j | j | Latin small j |
| 107 | k | k | k | k | Latin small k |
| 108 | l | l | l | l | Latin small l |
| 109 | m | m | m | m | Latin small m |
| 110 | n | n | n | n | Latin small n |
| 111 | o | o | o | o | Latin small o |
| 112 | p | p | p | p | Latin small p |
| 113 | q | q | q | q | Latin small q |
| 114 | r | r | r | r | Latin small r |
| 115 | s | s | s | s | Latin small s |
| 116 | t | t | t | t | Latin small t |
| 117 | u | u | u | u | Latin small u |
| 118 | v | v | v | v | Latin small v |
| 119 | w | w | w | w | Latin small w |
| 120 | x | x | x | x | Latin small x |
| 121 | y | y | y | y | Latin small y |
| 122 | z | z | z | z | Latin small z |
| 123 | { | { | { | { | left curly bracket |
| 124 | | | | | | | | | vertical line |
| 125 | } | } | } | } | right curly bracket |
| 126 | ~ | ~ | ~ | ~ | tilde |
| 127 | DEL |  |  |  |  |
| 128 |  | € |  |  | euro sign |
| 129 |  |  |  |  | NOT USED |
| 130 |  | ‚ |  |  | single low-9 quotation mark |
| 131 |  | ƒ |  |  | Latin small f with hook |
| 132 |  | „ |  |  | double low-9 quotation mark |
| 133 |  | … |  |  | horizontal ellipsis |
| 134 |  | † |  |  | dagger |
| 135 |  | ‡ |  |  | double dagger |
| 136 |  | ˆ |  |  | modifier letter circumflex accent |
| 137 |  | ‰ |  |  | per mille sign |
| 138 |  | Š |  |  | Latin S with caron |
| 139 |  | ‹ |  |  | single left-pointing angle quotation mark |
| 140 |  | Œ |  |  | Latin capital ligature OE |
| 141 |  |  |  |  | NOT USED |
| 142 |  | Ž |  |  | Latin Z with caron |
| 143 |  |  |  |  | NOT USED |
| 144 |  |  |  |  | NOT USED |
| 145 |  | ‘ |  |  | left single quotation mark |
| 146 |  | ’ |  |  | right single quotation mark |
| 147 |  | “ |  |  | left double quotation mark |
| 148 |  | ” |  |  | right double quotation mark |
| 149 |  | • |  |  | Bullet |
| 150 |  | – |  |  | en dash |
| 151 |  | — |  |  | em dash |
| 152 |  | ˜ |  |  | small tilde |
| 153 |  | ™ |  |  | trade mark sign |
| 154 |  | š |  |  | Latin small s with caron |
| 155 |  | › |  |  | single right-pointing angle quotation mark |
| 156 |  | œ |  |  | Latin small ligature oe |
| 157 |  |  |  |  | NOT USED |
| 158 |  | ž |  |  | Latin small z with caron |
| 159 |  | Ÿ |  |  | Latin Y with diaeresis |
| 160 |  |  |  |  | no-break space |
| 161 |  | ¡ | ¡ | ¡ | inverted exclamation mark |
| 162 |  | ¢ | ¢ | ¢ | cent sign |
| 163 |  | £ | £ | £ | pound sign |
| 164 |  | ¤ | ¤ | ¤ | currency sign |
| 165 |  | ¥ | ¥ | ¥ | yen sign |
| 166 |  | ¦ | ¦ | ¦ | broken bar |
| 167 |  | § | § | § | section sign |
| 168 |  | ¨ | ¨ | ¨ | Diaeresis |
| 169 |  | © | © | © | copyright sign |
| 170 |  | ª | ª | ª | feminine ordinal indicator |
| 171 |  | « | « | « | left-pointing double angle quotation mark |
| 172 |  | ¬ | ¬ | ¬ | not sign |
| 173 |  | ­ | ­ | ­ | soft hyphen |
| 174 |  | ® | ® | ® | registered sign |
| 175 |  | ¯ | ¯ | ¯ | Macron |
| 176 |  | ° | ° | ° | degree sign |
| 177 |  | ± | ± | ± | plus-minus sign |
| 178 |  | ² | ² | ² | superscript two |
| 179 |  | ³ | ³ | ³ | superscript three |
| 180 |  | ´ | ´ | ´ | acute accent |
| 181 |  | µ | µ | µ | micro sign |
| 182 |  | ¶ | ¶ | ¶ | pilcrow sign |
| 183 |  | · | · | · | middle dot |
| 184 |  | ¸ | ¸ | ¸ | Cedilla |
| 185 |  | ¹ | ¹ | ¹ | superscript one |
| 186 |  | º | º | º | masculine ordinal indicator |
| 187 |  | » | » | » | right-pointing double angle quotation mark |
| 188 |  | ¼ | ¼ | ¼ | vulgar fraction one quarter |
| 189 |  | ½ | ½ | ½ | vulgar fraction one half |
| 190 |  | ¾ | ¾ | ¾ | vulgar fraction three quarters |
| 191 |  | ¿ | ¿ | ¿ | inverted question mark |
| 192 |  | À | À | À | Latin A with grave |
| 193 |  | Á | Á | Á | Latin A with acute |
| 194 |  | Â | Â | Â | Latin A with circumflex |
| 195 |  | Ã | Ã | Ã | Latin A with tilde |
| 196 |  | Ä | Ä | Ä | Latin A with diaeresis |
| 197 |  | Å | Å | Å | Latin A with ring above |
| 198 |  | Æ | Æ | Æ | Latin AE |
| 199 |  | Ç | Ç | Ç | Latin C with cedilla |
| 200 |  | È | È | È | Latin E with grave |
| 201 |  | É | É | É | Latin E with acute |
| 202 |  | Ê | Ê | Ê | Latin E with circumflex |
| 203 |  | Ë | Ë | Ë | Latin E with diaeresis |
| 204 |  | Ì | Ì | Ì | Latin I with grave |
| 205 |  | Í | Í | Í | Latin I with acute |
| 206 |  | Î | Î | Î | Latin I with circumflex |
| 207 |  | Ï | Ï | Ï | Latin I with diaeresis |
| 208 |  | Ð | Ð | Ð | Latin Eth |
| 209 |  | Ñ | Ñ | Ñ | Latin N with tilde |
| 210 |  | Ò | Ò | Ò | Latin O with grave |
| 211 |  | Ó | Ó | Ó | Latin O with acute |
| 212 |  | Ô | Ô | Ô | Latin O with circumflex |
| 213 |  | Õ | Õ | Õ | Latin O with tilde |
| 214 |  | Ö | Ö | Ö | Latin O with diaeresis |
| 215 |  | × | × | × | multiplication sign |
| 216 |  | Ø | Ø | Ø | Latin O with stroke |
| 217 |  | Ù | Ù | Ù | Latin U with grave |
| 218 |  | Ú | Ú | Ú | Latin U with acute |
| 219 |  | Û | Û | Û | Latin U with circumflex |
| 220 |  | Ü | Ü | Ü | Latin U with diaeresis |
| 221 |  | Ý | Ý | Ý | Latin Y with acute |
| 222 |  | Þ | Þ | Þ | Latin Thorn |
| 223 |  | ß | ß | ß | Latin small sharp s |
| 224 |  | à | à | à | Latin small a with grave |
| 225 |  | á | á | á | Latin small a with acute |
| 226 |  | â | â | â | Latin small a with circumflex |
| 227 |  | ã | ã | ã | Latin small a with tilde |
| 228 |  | ä | ä | ä | Latin small a with diaeresis |
| 229 |  | å | å | å | Latin small a with ring above |
| 230 |  | æ | æ | æ | Latin small ae |
| 231 |  | ç | ç | ç | Latin small c with cedilla |
| 232 |  | è | è | è | Latin small e with grave |
| 233 |  | é | é | é | Latin small e with acute |
| 234 |  | ê | ê | ê | Latin small e with circumflex |
| 235 |  | ë | ë | ë | Latin small e with diaeresis |
| 236 |  | ì | ì | ì | Latin small i with grave |
| 237 |  | í | í | í | Latin small i with acute |
| 238 |  | î | î | î | Latin small i with circumflex |
| 239 |  | ï | ï | ï | Latin small i with diaeresis |
| 240 |  | ð | ð | ð | Latin small eth |
| 241 |  | ñ | ñ | ñ | Latin small n with tilde |
| 242 |  | ò | ò | ò | Latin small o with grave |
| 243 |  | ó | ó | ó | Latin small o with acute |
| 244 |  | ô | ô | ô | Latin small o with circumflex |
| 245 |  | õ | õ | õ | Latin small o with tilde |
| 246 |  | ö | ö | ö | Latin small o with diaeresis |
| 247 |  | ÷ | ÷ | ÷ | division sign |
| 248 |  | ø | ø | ø | Latin small o with stroke |
| 249 |  | ù | ù | ù | Latin small u with grave |
| 250 |  | ú | ú | ú | Latin small u with acute |
| 251 |  | û | û | û | Latin small with circumflex |
| 252 |  | ü | ü | ü | Latin small u with diaeresis |
| 253 |  | ý | ý | ý | Latin small y with acute |
| 254 |  | þ | þ | þ | Latin small thorn |
| 255 |  | ÿ | ÿ | ÿ | Latin small y with diaeresis |

# HTML Uniform Resource Locators

A URL is another word for a web address.

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.

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.

Top of Form

Bottom of Form

If you click "Submit", the browser will URL encode the input before it is sent to the server.

A page at the server will display the received input.

Try some other input and click Submit again.

## 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 |

# HTML Versus XHTML

XHTML is a stricter, more XML-based version of HTML.

## 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

## Why XHTML?

XML is a markup language where all documents must be marked up correctly (be "well-formed").

XHTML was developed to make HTML more extensible and flexible to work with other data formats (such as XML). In addition, browsers ignore errors in HTML pages, and try to display the website even if it has some errors in the markup. So XHTML comes with a much stricter error handling.

If you want to study XML, please read our [XML Tutorial](https://www.w3schools.com/xml/default.asp).

## 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 - <!DOCTYPE ....> Is Mandatory

An XHTML document must have an XHTML <!DOCTYPE> declaration.

The <html>, <head>, <title>, and <body> elements must also be present, and the xmlns attribute in <html> must specify the xml namespace for the document.

### Example

Here is an XHTML document with a minimum of required tags:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
  <title>Title of document</title>  
</head>  
<body>  
  some content here...  
  
</body>  
</html>

## XHTML Elements Must be Properly Nested

In XHTML, elements must always be properly nested within each other, like this:

### Correct:

<b><i>Some text</i></b>

### Wrong:

<b><i>Some text</b></i>

## XHTML Elements Must Always be Closed

In XHTML, elements must always be closed, like this:

### Correct:

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

### Wrong:

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

## 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" />

### Wrong:

A break: <br>  
A horizontal rule: <hr>  
An image: <img src="happy.gif" alt="Happy face">

## XHTML Elements Must be in Lowercase

In XHTML, element names must always be in lowercase, like this:

### Correct:

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

### Wrong:

<BODY>  
<P>This is a paragraph</P>  
</BODY>

## XHTML Attribute Names Must be in Lowercase

In XHTML, attribute names must always be in lowercase, like this:

### Correct:

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

### Wrong:

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

## XHTML Attribute Values Must be Quoted

In XHTML, attribute values must always be quoted, like this:

### Correct:

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

### Wrong:

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

## XHTML Attribute Minimization is Forbidden

In XHTML, attribute minimization is forbidden:

### Correct:

<input type="checkbox" name="vehicle" value="car" checked="checked" />  
<input type="text" name="lastname" disabled="disabled" />

### Wrong:

<input type="checkbox" name="vehicle" value="car" checked />  
<input type="text" name="lastname" disabled />

# HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>HTML Forms</h2>

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

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

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

</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".</p>

</body>

</html>

## HTML Forms

Top of Form

First name:  
  
Last name:

Bottom of Form

If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".

## **The <form> Element**

The HTML <form> element is used to create an HTML form for user input:

<form>  
.  
*form elements*  
.  
</form>

The <form> element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

All the different form elements are covered in this chapter: [HTML Form Elements](https://www.w3schools.com/html/html_form_elements.asp).

## **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.

### **Example**

A form with input fields for text:

<!DOCTYPE html>

<html>

<body>

<h2>Text input fields</h2>

<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>

<p>Note that the form itself is not visible.</p>

<p>Also note that the default width of text input fields is 20 characters.</p>

</body>

</html>

**OUTPUT**

## Text input fields

Top of Form

First name:  
  
Last name:

Bottom of Form

Note that the form itself is not visible.

Also note that the default width of text input fields is 20 characters.

## **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.

### **Example**

A form with radio buttons:

<!DOCTYPE html>

<html>

<body>

<h2>Radio Buttons</h2>

<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>

</body>

</html>

**OUTPUT**

## Radio Buttons

Choose your favorite Web language:

Top of Form

     HTML  
     CSS  
     JavaScript

## **Checkboxes**

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

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

### **Example**

A form with checkboxes:

<!DOCTYPE html>

<html>

<body>

<h2>Checkboxes</h2>

<p>The <strong>input type="checkbox"</strong> defines a checkbox:</p>

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

<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><br><br>

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

</form>

</body>

</html>

**OUTPUT**

## Checkboxes

The **input type="checkbox"** defines a checkbox:

Top of Form

## I have a bike  I have a car  I have a boat **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.

<!DOCTYPE html>

<html>

<body>

<h2>HTML Forms</h2>

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

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

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

</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".</p>

</body>

</html>

**OUTPUT**

## HTML Forms

Top of Form

First name:  
  
Last name:

Bottom of Form

If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".

## **The Name Attribute for <input>**

Notice that each input field must have a name attribute to be submitted.

If the name attribute is omitted, the value of the input field will not be sent at all.

### **Example**

This example will not submit the value of the "First name" input field:

<!DOCTYPE html>

<html>

<body>

<h2>HTML Forms</h2>

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

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

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

</form>

<p>If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".</p>

</body>

</html>

## HTML Forms

Top of Form

First name:  
  
Last name:

Bottom of Form

If you click the "Submit" button, the form-data will be sent to a page called "/action\_page.php".

## **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.

### **Example**

Here, the submitted result will open in a new browser tab:

<!DOCTYPE html>

<html>

<body>

<h2>The form target attribute</h2>

<p>When submitting this form, the result will be opened in a new browser tab:</p>

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

<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>

</body>

</html>

**OUTPUT**

## The form target attribute

When submitting this form, the result will be opened in a new browser tab:

Top of Form

## First name: Last name: **The Method Attribute**

The method attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

### **Example**

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

## **The Method Attribute**

The method attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

### **Example**

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

## The method Attribute

This form will be submitted using the GET method:

Top of Form

First name:  
  
Last name:

Bottom of Form

After you submit, notice that the form values is visible in the address bar of the new browser tab.

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h2>The method Attribute</h2>

<p>This form will be submitted using the POST method:</p>

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

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname" value="John"><br>

<label for="lname">Last name:</label><br>

<input type="text" id="lname" name="lname" value="Doe"><br><br>

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

</form>

<p>After you submit, notice that, unlike the GET method, the form values is NOT visible in the address bar of the new browser tab.</p>

</body>

</html>

**OUTPUT**

## The method Attribute

This form will be submitted using the POST method:

Top of Form

First name:  
  
Last name:

Bottom of Form

After you submit, notice that, unlike the GET method, the form values is NOT visible in the address bar of the new browser tab.

**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

## **The Autocomplete Attribute**

The autocomplete attribute specifies whether a form should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

### **Example**

A form with autocomplete on:

<!DOCTYPE html>

<html>

<body>

<h1>The form autocomplete attribute</h1>

<p>Fill in and submit the form, then reload the page, start to fill in the form again - and see how autocomplete works.</p>

<p>Then, try to set autocomplete to "off".</p>

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

<label for="fname">First name:</label>

<input type="text" id="fname" name="fname"><br><br>

<label for="email">Email:</label>

<input type="text" id="email" name="email"><br><br>

<input type="submit">

</form>

</body>

</html>

# The form autocomplete attribute

Fill in and submit the form, then reload the page, start to fill in the form again - and see how autocomplete works.

Then, try to set autocomplete to "off".

Top of Form

First name:   
  
Email:

## **The Novalidate Attribute**

The novalidate attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

### **Example**

A form with a novalidate attribute:

<!DOCTYPE html>

<html>

<body>

<h1>The form novalidate attribute</h1>

<p>The novalidate attribute indicates that the form input is not to be validated on submit:</p>

<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">

</form>

</body>

</html>

# The form novalidate attribute

The novalidate attribute indicates that the form input is not to be validated on submit:

Top of Form

Enter your email:

# HTML Form Elements

## **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.

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>The input Element</h2>

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

<label for="fname">First name:</label><br>

<input type="text" id="fname" name="fname"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

## The input Element

Top of Form

First name:

## **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:

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>The select Element</h2>

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

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

<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>

<input type="submit">

</form>

</body>

</html>

**OUTPUT**

## The select Element

The select element defines a drop-down list:

Top of Form

Choose a car:      Volvo     Saab     Fiat     Audi

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:

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>Pre-selected Option</h2>

<p>You can preselect an option with the selected attribute:</p>

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

<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" selected>Fiat</option>

<option value="audi">Audi</option>

</select>

<input type="submit">

</form>

</body>

</html>

**OUTPUT**

## Pre-selected Option

You can preselect an option with the selected attribute:

Top of Form

Choose a car:      Volvo     Saab     Fiat     Audi

### **Visible Values:**

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

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>Visible Option Values</h2>

<p>Use the size attribute to specify the number of visible values.</p>

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

<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><br><br>

<input type="submit">

</form>

</body>

</html>

**OUTPUT**

## Visible Option Values

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

Top of Form

### Choose a car:      Volvo     Saab     Fiat     Audi    **Allow Multiple Selections:**

Use the multiple attribute to allow the user to select more than one value:

**Example**

<!DOCTYPE html>

## <html>

## <body>

## <h2>Allow Multiple Selections</h2>

## <p>Use the multiple attribute to allow the user to select more than one value.</p>

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

## <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><br><br>

## <input type="submit">

## </form>

## <p>Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.</p>

## </body>

## </html>

## **OUTPUT**

## Allow Multiple Selections

Use the multiple attribute to allow the user to select more than one value.

Top of Form

Choose a car:      Volvo     Saab     Fiat     Audi

Bottom of Form

Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.

## **The <textarea> Element**

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

### **Example**

### <!DOCTYPE html>

### <html>

### <body>

### <h2>Textarea</h2>

### <p>The textarea element defines a multi-line input field.</p>

### <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>

### </body>

**OUTPUT**

## Textarea

The textarea element defines a multi-line input field.

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

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

This is how the HTML code above will be displayed in a browser:

You can also define the size of the text area by using CSS:

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>Styling Textarea</h2>

<p>Use CSS to change the size of the textarea:</p>

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

<textarea name="message" style="width:200px; height:600px;">The cat was playing in the garden.</textarea>

<br>

<input type="submit">

</form>

</body>

</html>

**OUTPUT**

## Styling Textarea

Use CSS to change the size of the textarea:

## **The <button> Element**

The <button> element defines a clickable button:

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<h2>The button Element</h2>

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

</body>

</html>

### **Example**

## The button Element

Click Me!

## **The <fieldset> and <legend> Elements**

The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>Grouping Form Data with Fieldset</h2>

<p>The fieldset element is used to group related data in a form, and the legend element defines a caption for the fieldset element.</p>

<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>

</body>

</html>

**OUTPUT**

## Grouping Form Data with Fieldset

The fieldset element is used to group related data in a form, and the legend element defines a caption for the fieldset element.

Top of Form

Personalia:First name:  
  
Last name:  
This is how the HTML code above will be displayed in a browser:

Top of Form

Personalia:First name:  
  
Last name:

Bottom of 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.

### **Example**

<!DOCTYPE html>

<html>

<body>

<h2>The datalist Element</h2>

<p>The datalist element specifies a list of pre-defined options for an input element.</p>

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

<input list="browsers" name="browser">

<datalist id="browsers">

<option value="Edge">

<option value="Firefox">

<option value="Chrome">

<option value="Opera">

<option value="Safari">

</datalist>

<input type="submit">

</form>

</body>

</html>

**OUTPUT**

## The datalist Element

The datalist element specifies a list of pre-defined options for an input element.

## **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**:

### **Example**

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
</form>

This is how the HTML code above will be displayed in a browser:

First name:  
  
Last name:

## **Input Type Password**

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

### **Example**

<form>  
  <label for="username">Username:</label><br>  
  <input type="text" id="username" name="username"><br>  
  <label for="pwd">Password:</label><br>  
  <input type="password" id="pwd" name="pwd">  
</form>

This is how the HTML code above will be displayed in a browser:

Username:  
  
Password:

The characters in a password field are masked (shown as asterisks or circles).

## **Input Type Submit**

<input type="submit"> defines a button for **submitting** form data to a **form-handler**.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

### **Example**

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

First name:  
  
Last name:

Bottom of Form

If you omit the submit button's value attribute, the button will get a default text:

### **Example**

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit">  
</form>

## **Input Type Reset**

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

### **Example**

<form action="/action\_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
  <input type="reset" value="Reset">  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

First name:  
  
Last name:

Bottom of Form

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

## **Input Type Radio**

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

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

### **Example**

<p>Choose your favorite Web language:</p>  
  
<form>  
  <input type="radio" id="html" name="fav\_language" value="HTML">  
  <label for="html">HTML</label><br>  
  <input type="radio" id="css" name="fav\_language" value="CSS">  
  <label for="css">CSS</label><br>  
  <input type="radio" id="javascript" name="fav\_language" value="JavaScript">  
  <label for="javascript">JavaScript</label>  
</form>

This is how the HTML code above will be displayed in a browser:

 HTML  
 CSS  
 JavaScript

## **Input Type Checkbox**

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

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

### **Example**

<form>  
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">  
  <label for="vehicle1"> I have a bike</label><br>  
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">  
  <label for="vehicle2"> I have a car</label><br>  
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">  
  <label for="vehicle3"> I have a boat</label>  
</form>

This is how the HTML code above will be displayed in a browser:

 I have a bike  
 I have a car  
 I have a boat

## **Input Type Button**

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

### **Example**

<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.

### **Example**

<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.

### **Example**

<form>  
  <label for="birthday">Birthday:</label>  
  <input type="date" id="birthday" name="birthday">  
</form>

You can also use the min and max attributes to add restrictions to dates:

### **Example**

<form>  
  <label for="datemax">Enter a date before 1980-01-01:</label>  
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>  
  <label for="datemin">Enter a date after 2000-01-01:</label>  
  <input type="date" id="datemin" name="datemin" min="2000-01-02">  
</form>

## **Input Type Datetime-local**

The <input type="datetime-local"> specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

### **Example**

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

## **Input Type Email**

The <input type="email"> is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

### **Example**

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

## **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.

### **Example**

<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.

### **Example**

<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!

### **Example**

<form>  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <input type="hidden" id="custId" name="custId" value="3487">  
  <input type="submit" value="Submit">  
</form>

## **Input Type Month**

The <input type="month"> allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

### **Example**

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

## **Input Type Number**

The <input type="number"> defines a **numeric** input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

### **Example**

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

## **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 |

You will learn more about input restrictions in the next chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

### **Example**

<form>  
  <label for="quantity">Quantity:</label>  
  <input type="number" id="quantity" name="quantity" min="0" max="100" step="10" value="30">  
</form>

## **Input Type Range**

The <input type="range"> defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

### **Example**

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

## **Input Type Search**

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

### **Example**

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

## **Input Type Tel**

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

### **Example**

<form>  
  <label for="phone">Enter your phone number:</label>  
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">  
</form>

## **Input Type Time**

The <input type="time"> allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

### **Example**

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

## **Input Type Url**

The <input type="url"> is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

### **Example**

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

## **Input Type Week**

The <input type="week"> allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

### **Example**

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

## **Exercise:**

In the form below, add an input field for text, with the name "username" .

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

**Bottom of Form**

# HTML Input Attributes

## **The value Attribute**

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

### **Example**

Input fields with initial (default) values:

# <!DOCTYPE html>

# <html>

# <body>

# <h1>The input value attribute</h1>

# <p>The value attribute specifies an initial value for an input field:</p>

# <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>

# </body>

# </html>

# OUTPUT

# The input value attribute

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

Top of Form

## First name: Last name: **The readonly Attribute**

The input readonly attribute specifies that an input field is read-only.

A read-only input field cannot be modified (however, a user can tab to it, highlight it, and copy the text from it).

The value of a read-only input field will be sent when submitting the form!

### **Example**

A read-only input field:

<!DOCTYPE html>

<html>

<body>

<h1>The input readonly attribute</h1>

<p>The readonly attribute specifies that an input field should be read-only (cannot be changed):</p>

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

<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"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The input readonly attribute

The readonly attribute specifies that an input field should be read-only (cannot be changed):

Top of Form

## First name: Last name: **The disabled Attribute**

The input disabled attribute specifies that an input field should be disabled.

A disabled input field is unusable and un-clickable.

The value of a disabled input field will not be sent when submitting the form!

### **Example**

A disabled input field:

<!DOCTYPE html>

<html>

<body>

<h1>The input disabled attribute</h1>

<p>The disabled attribute specifies that an input field should be disabled (unusable and un-clickable):</p>

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

<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"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The input disabled attribute

The disabled attribute specifies that an input field should be disabled (unusable and un-clickable):

Top of Form

First name:  
  
Last name:

## **The size Attribute**

The input size attribute specifies the visible width, in characters, of an input field.

The default value for size is 20.

**Note:** The size attribute works with the following input types: text, search, tel, url, email, and password.

### **Example**

Set a width for an input field:

<!DOCTYPE html>

<html>

<body>

<h1>The input size attribute</h1>

<p>The size attribute specifies the width (in characters) of an input field:</p

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

<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"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The input size attribute

The size attribute specifies the width (in characters) of an input field:

Top of Form

## First name: PIN: **The maxlength Attribute**

The input maxlength attribute specifies the maximum number of characters allowed in an input field.

**Note:** When a maxlength is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

### **Example**

Set a maximum length for an input field:

<!DOCTYPE html>

<html>

<body>

<h1>The input maxlength attribute</h1>

<p>The maxlength attribute specifies the maximum number of characters allowed in an input field:</p>

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

<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"><br><br>

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

</form>

</body>

</html>  
**OUTPUT**

# The input maxlength attribute

The maxlength attribute specifies the maximum number of characters allowed in an input field:

Top of Form

## First name: PIN: **The min and max Attributes**

The input min and max attributes specify the minimum and maximum values for an input field.

The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

**Tip:** Use the max and min attributes together to create a range of legal values.

### **Example**

Set a max date, a min date, and a range of legal values:

<!DOCTYPE html>

<html>

<body>

<h1>The input min and max attributes</h1>

<p>The min and max attributes specify the minimum and maximum values for an input element.</p>

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

<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"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The input min and max attributes

The min and max attributes specify the minimum and maximum values for an input element.

Top of Form

## Enter a date before 1980-01-01:  Enter a date after 2000-01-01:  Quantity (between 1 and 5):  **The multiple Attribute**

The input multiple attribute specifies that the user is allowed to enter more than one value in an input field.

The multiple attribute works with the following input types: email, and file.

### **Example**

A file upload field that accepts multiple values:

<!DOCTYPE html>

<html>

<body>

<h1>The input multiple attributes</h1>

<p>The multiple attribute specifies that the user is allowed to enter more than one value in an input field.</p>

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

<label for="files">Select files:</label>

<input type="file" id="files" name="files" multiple><br><br>

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

</form>

<p>To select multiple files, hold down the CTRL or SHIFT key while selecting.</p>

</body>

# </html> The input multiple attributes

The multiple attribute specifies that the user is allowed to enter more than one value in an input field.

Top of Form

Select files: 

Bottom of Form

To select multiple files, hold down the CTRL or SHIFT key while selecting.

## **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.

**Tip:** Learn more about [regular expressions](https://www.w3schools.com/js/js_regexp.asp) in our JavaScript tutorial.

### **Example**

An input field that can contain only three letters (no numbers or special characters):

<!DOCTYPE html>

<html>

<body>

<h1>The input pattern attribute</h1>

<p>The pattern attribute specifies a regular expression that the input element's value is checked against.</p>

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

<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"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The input pattern attribute

The pattern attribute specifies a regular expression that the input element's value is checked against.

Top of Form

## Country code:

## **The placeholder Attribute**

The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).

The short hint is displayed in the input field before the user enters a value.

The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

### **Example**

An input field with a placeholder text:

<!DOCTYPE html>

<html>

<body>

<h1>The input placeholder attribute</h1>

<p>The placeholder attribute specifies a short hint that describes the expected value of an input field.</p>

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

<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}"><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The input placeholder attribute

The placeholder attribute specifies a short hint that describes the expected value of an input field.

Top of Form

Enter a phone number:

## **The step Attribute**

The input step attribute specifies the legal number intervals for an input field.

Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

**Tip:** This attribute can be used together with the max and min attributes to create a range of legal values.

The step attribute works with the following input types: number, range, date, datetime-local, month, time and week.

### **Example**

An input field with a specified legal number intervals:

<form>  
  <label for="points">Points:</label>  
  <input type="number" id="points" name="points" step="3">  
</form>

**Note:** Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must also be checked by the receiver (the server)!

## **The autofocus Attribute**

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

### **Example**

Let the "First name" input field automatically get focus when the page loads:

<form>  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" autofocus><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname">  
</form>

## **The height and width Attributes**

The input height and width attributes specify the height and width of an <input type="image"> element.

**Tip:** Always specify both the height and width attributes for images. If height and width are set, the space required for the image is reserved when the page is loaded. Without these attributes, the browser does not know the size of the image, and cannot reserve the appropriate space to it. The effect will be that the page layout will change during loading (while the images load).

### **Example**

Define an image as the submit button, with height and width attributes:

<form>  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="image" src="img\_submit.gif" alt="Submit" width="48" height="48">  
</form>

## **The list Attribute**

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

### **Example**

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

<form>  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="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.

### **Example**

An HTML form with autocomplete on, and off for one input field:

<form action="/action\_page.php" autocomplete="on">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <label for="email">Email:</label>  
  <input type="email" id="email" name="email" autocomplete="off"><br><br>  
  <input type="submit" value="Submit">  
</form>

**OUTPUT**

# The autocomplete attribute

The autocomplete attribute specifies whether or not an input field should have autocomplete enabled.

Fill in and submit the form, then reload the page to see how autocomplete works.

Notice that autocomplete is "on" for the form, but "off" for the e-mail field!

Top of Form

# First name:  Last name:  Email:  HTML Input form\* Attributes

## **The form Attribute**

The input form attribute specifies the form the <input> element belongs to.

The value of this attribute must be equal to the id attribute of the <form> element it belongs to.

### **Example**

An input field located outside of the HTML form (but still a part of the form):

<!DOCTYPE html>

<html>

<body>

<h1>The input form attribute</h1>

<p>The form attribute specifies the form an input element belongs to.</p>

<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>

<p>The "Last name" field below is outside the form element, but still part of the form.</p>

<label for="lname">Last name:</label>

<input type="text" id="lname" name="lname" form="form1">

</body>

</html>

**OUTPUT**

# The input form attribute

The form attribute specifies the form an input element belongs to.

Top of Form

First name: 

Bottom of Form

The "Last name" field below is outside the form element, but still part of the form.

Last name:

## **The formenctype Attribute**

The input formenctype attribute specifies how the form-data should be encoded when submitted (only for forms with method="post").

**Note:** This attribute overrides the enctype attribute of the <form> element.

The formenctype attribute works with the following input types: submit and image.

### **Example**

A form with two submit buttons. The first sends the form-data with default encoding, the second sends the form-data encoded as "multipart/form-data":

<!DOCTYPE html>

<html>

<body>

<h1>The input formenctype attribute</h1>

<p>The formenctype attribute specifies how the form data should be encoded when submitted.</p>

<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>

</body>

</html>

**OUTPUT**

# The input formenctype attribute

The formenctype attribute specifies how the form data should be encoded when submitted.

Top of Form

First name:

## **The formmethod Attribute**

The input formmethod attribute defines the HTTP method for sending form-data to the action URL.

**Note:** This attribute overrides the method attribute of the <form> element.

The formmethod attribute works with the following input types: submit and image.

The form-data can be sent as URL variables (method="get") or as an HTTP post transaction (method="post").

**Notes on the "get" method:**

* This method appends the form-data to the URL in name/value pairs
* This method is useful for form submissions where a user want to bookmark the result
* There is a limit to how much data you can place in a URL (varies between browsers), therefore, you cannot be sure that all of the form-data will be correctly transferred
* Never use the "get" method to pass sensitive information! (password or other sensitive information will be visible in the browser's address bar)

**Notes on the "post" method:**

* This method sends the form-data as an HTTP post transaction
* Form submissions with the "post" method cannot be bookmarked
* The "post" method is more robust and secure than "get", and "post" does not have size limitations

### **Example**

A form with two submit buttons. The first sends the form-data with method="get". The second sends the form-data with method="post":  
<!DOCTYPE html>

<html>

<body>

<h1>The input formmethod Attribute</h1>

<p>The formmethod attribute defines the HTTP method for sending form-data to the action URL.</p>

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

<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>

</body>

</html>

**OUTPUT**

# The input formmethod Attribute

The formmethod attribute defines the HTTP method for sending form-data to the action URL.

Top of Form

## First name:  Last name:   **The formnovalidate Attribute**

The input formnovalidate attribute specifies that an <input> element should not be validated when submitted.

**Note:** This attribute overrides the novalidate attribute of the <form> element.

The formnovalidate attribute works with the following input types: submit.

### **Example**

A form with two submit buttons (with and without validation):

<!DOCTYPE html>

<html>

<body>

<h1>The input formnovalidate attribute</h1>

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

<label for="email">Enter your email:</label>

<input type="email" id="email" name="email" required><br><br>

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

<input type="submit" formnovalidate="formnovalidate" value="Submit without validation">

</form>

</body>

</html>

**OUTPUT**

# The input formnovalidate attribute

Top of Form

Enter your email:

## **The novalidate Attribute**

The novalidate attribute is a <form> attribute.

When present, novalidate specifies that all of the form-data should not be validated when submitted.

### **Example**

Specify that no form-data should be validated on submit:

<!DOCTYPE html>

<html>

<body>

<h1>The form novalidate attribute</h1>

<p>The novalidate attribute specifies that the form data should not be validated when submitted.</p>

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

<label for="email">Enter your email:</label>

<input type="email" id="email" name="email" required><br><br>

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

</form>

</body>

</html>

**OUTPUT**

# The form novalidate attribute

The novalidate attribute specifies that the form data should not be validated when submitted.

Top of Form

# Enter your email:

# HTML Canvas Graphics:

The HTML <canvas> element is used to draw graphics on a web page.

The graphic to the left is created with <canvas>. It shows four elements: a red rectangle, a gradient rectangle, a multicolor rectangle, and a multicolor text.

## **What is HTML Canvas?**

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

## **Browser Support**

The numbers in the table specify the first browser version that fully supports the <canvas> element.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element |  |  |  |  |  |
| <canvas> | 4.0 | 9.0 | 2.0 | 3.1 | 9.0 |

## **Canvas Examples**

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

The markup looks like this:

<canvas id="myCanvas" width="200" height="100"></canvas>

**Note:** Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas. To add a border, use the style attribute.

Here is an example of a basic, empty canvas:

### **Example**

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

## **Add a JavaScript:**

After creating the rectangular canvas area, you must add a JavaScript to do the drawing.

Here are some examples:

### Draw a Line

### Example

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.moveTo(0, 0);  
ctx.lineTo(200, 100);  
ctx.stroke();  
</script>

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

Your browser does not support the HTML canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

ctx.moveTo(0,0);

ctx.lineTo(200,100);

ctx.stroke();

</script>

</body>

</html>

### Example

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.beginPath();  
ctx.arc(95, 50, 40, 0, 2 \* Math.PI);  
ctx.stroke();  
</script>

<!DOCTYPE html>

<html>

<body

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

Your browser does not support the HTML canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

ctx.beginPath();

ctx.arc(95,50,40,0,2\*Math.PI);

ctx.stroke();

</script>

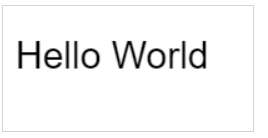
</body>

</html>

### Draw a Text

### Example

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.fillText("Hello World", 10, 50);  
</script>



### Example

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
ctx.font = "30px Arial";  
ctx.strokeText("Hello World", 10, 50);  
</script>

Output:



<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">

Your browser does not support the HTML canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

// Create gradient

var grd = ctx.createLinearGradient(0,0,200,0);

grd.addColorStop(0,"red");

grd.addColorStop(1,"white");

// Fill with gradient

ctx.fillStyle = grd;

ctx.fillRect(10,10,150,80);

</script>

</body>

</html>

Output:



### Example

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
  
// Create gradient  
var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10, 10, 150, 80);  
</script>



### Example

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
  
// Create gradient  
var grd = ctx.createRadialGradient(75, 50, 5, 90, 60, 100);  
grd.addColorStop(0, "red");  
grd.addColorStop(1, "white");  
  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10, 10, 150, 80);  
</script>

Output:



### Draw Image

<script>  
var c = document.getElementById("myCanvas");  
var ctx = c.getContext("2d");  
var img = document.getElementById("scream");  
ctx.drawImage(img, 10, 10);  
</script>

Program:

<!DOCTYPE html>

<html>

<body>

<p>Image to use:</p>

<img id="scream" src="img\_the\_scream.jpg" alt="The Scream" width="220" height="277">

<p>Canvas to fill:</p>

<canvas id="myCanvas" width="250" height="300"

style="border:1px solid #d3d3d3;">

Your browser does not support the HTML canvas tag.</canvas>

<p><button onclick="myCanvas()">Try it</button></p>

<script>

function myCanvas() {

var c = document.getElementById("myCanvas");

var ctx = c.getContext("2d");

var img = document.getElementById("scream");

ctx.drawImage(img,10,10);

}

</script>

</body>

</html>

Output:



# HTML SVG Graphics

SVG defines vector-based graphics in XML format.

## **What is SVG?**

* SVG stands for Scalable Vector Graphics
* SVG is used to define graphics for the Web

## **The HTML <svg> Element**

The HTML <svg> element is a container for SVG graphics.

SVG has several methods for drawing paths, boxes, circles, text, and graphic images.

**EXAMPLE**

<!DOCTYPE html>

<html>

<body>

<svg width="100" height="100">

<circle cx="50" cy="50" r="40"

stroke="green" stroke-width="4" fill="yellow" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

## **Browser Support**

The numbers in the table specify the first browser version that fully supports the <svg> element.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element |  |  |  |  |  |
| <svg> | 4.0 | 9.0 | 3.0 | 3.2 | 10.1 |

## **SVG Rectangle**

### **Example**

## **SVG Rectangle**

### **Example**

## SVG Rounded Rectangle

### Example

# <!DOCTYPE html>

# <html>

# <body>

# <svg width="400" height="180">

# <rect x="50" y="20" rx="20" ry="20" width="150" height="150"

# style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />

# Sorry, your browser does not support inline SVG.

# </svg>

# </body>

# </html>

## **SVG Star**

### **Example**

<!DOCTYPE html>

<html>

<body>

<svg width="300" height="200">

<polygon points="100,10 40,198 190,78 10,78 160,198"

style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

## **SVG Logo**

SVG

### **Example**

<!DOCTYPE html>

<html>

<body>

<svg height="130" width="500">

<defs>

<linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">

<stop offset="0%"

style="stop-color:rgb(255,255,0);stop-opacity:1" />

<stop offset="100%"

style="stop-color:rgb(255,0,0);stop-opacity:1" />

</linearGradient>

</defs>

<ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />

<text fill="#ffffff" font-size="45" font-family="Verdana"

x="50" y="86">SVG</text>

Sorry, your browser does not support inline SVG.

</svg>

</body>

</html>

## **Differences Between SVG and Canvas**

SVG is a language for describing 2D graphics in XML.

Canvas draws 2D graphics, on the fly (with JavaScript).

SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

## **Comparison of Canvas and SVG**

The table below shows some important differences between Canvas and SVG:

|  |  |
| --- | --- |
| **Canvas** | **SVG** |
| * Resolution dependent * No support for event handlers * Poor text rendering capabilities * You can save the resulting image as .png or .jpg * Well suited for graphic-intensive games | * Resolution independent * Support for event handlers * Best suited for applications with large rendering areas (Google Maps) * Slow rendering if complex (anything that uses the DOM a lot will be slow) * Not suited for game applications |

# HTML Multimedia

Multimedia on the web is sound, music, videos, movies, and animations.

## **What is Multimedia?**

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

## **Browser Support**

The first web browsers had support for text only, limited to a single font in a single color.

Later came browsers with support for colors, fonts, images, and multimedia!

## **Multimedia Formats**

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

## **Common Video Formats**

|  |  |
| --- | --- |
| Videoformats | There are many video formats out there.  The MP4, WebM, and Ogg formats are supported by HTML.  The MP4 format is recommended by YouTube. |

|  |  |  |
| --- | --- | --- |
| **Format** | **File** | **Description** |
| MPEG | .mpg .mpeg | MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML. |
| AVI | .avi | AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| WMV | .wmv | WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| QuickTime | .mov | QuickTime. Developed by Apple. Commonly used in video cameras and TV hardware. Plays well on Apple computers, but not in web browsers. |
| RealVideo | .rm .ram | RealVideo. Developed by Real Media to allow video streaming with low bandwidths. Does not play in web browsers. |
| Flash | .swf .flv | Flash. Developed by Macromedia. Often requires an extra component (plug-in) to play in web browsers. |
| Ogg | .ogg | Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML. |
| WebM | .webm | WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML. |
| MPEG-4 or MP4 | .mp4 | MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and  recommended by YouTube. |

**Note:** Only MP4, WebM, and Ogg video are supported by the HTML standard.

## **Common Audio Formats**

MP3 is the best format for compressed recorded music. The term MP3 has become synonymous with digital music.

If your website is about recorded music, MP3 is the choice.

|  |  |  |
| --- | --- | --- |
| **Format** | **File** | **Description** |
| MIDI | .mid .midi | MIDI (Musical Instrument Digital Interface). Main format for all electronic music devices like synthesizers and PC sound cards. MIDI files do not contain sound, but digital notes that can be played by electronics. Plays well on all computers and music hardware, but not in web browsers. |
| RealAudio | .rm .ram | RealAudio. Developed by Real Media to allow streaming of audio with low bandwidths. Does not play in web browsers. |
| WMA | .wma | WMA (Windows Media Audio). Developed by Microsoft. Plays well on Windows computers, but not in web browsers. |
| AAC | .aac | AAC (Advanced Audio Coding). Developed by Apple as the default format for iTunes. Plays well on Apple computers, but not in web browsers. |
| WAV | .wav | WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML. |
| Ogg | .ogg | Ogg. Developed by the Xiph.Org Foundation. Supported by HTML. |
| MP3 | .mp3 | MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers. |
| MP4 | .mp4 | MP4 is a video format, but can also be used for audio. Supported by all browsers. |

**Note:** Only MP3, WAV, and Ogg audio are supported by the HTML standard.

# HTML Video

The HTML <video> element is used to show a video on a web page.

### **Example**

Courtesy of [Big Buck Bunny](https://www.bigbuckbunny.org/):

<!DOCTYPE html>

<html>

<body>

<video width="400" controls>

<source src="mov\_bbb.mp4" type="video/mp4">

<source src="mov\_bbb.ogg" type="video/ogg">

Your browser does not support HTML video.

</video>

<p>

Video courtesy of

<a href="https://www.bigbuckbunny.org/" target="\_blank">Big Buck Bunny</a>.

</p>

</body>

</html>

**OUTPUT**

Video courtesy of [Big Buck Bunny](https://www.bigbuckbunny.org/).

## **The HTML <video> Element**

To show a video in HTML, use the <video> element:

<!DOCTYPE html>

<html>

<body>

<video width="320" height="240" controls>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

</body>

</html>

## **How it Works**

The controls attribute adds video controls, like play, pause, and volume.

It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.

The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

## **HTML <video> Autoplay**

To start a video automatically, use the autoplay attribute:

### **Example**

<!DOCTYPE html>

<html>

<body>

<video width="320" height="240" autoplay>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

</body>

</html>

**Note:** Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

Add muted after autoplay to let your video start playing automatically (but muted):

### **Example**

<!DOCTYPE html>

<html>

<body

<video width="320" height="240" autoplay muted>

<source src="movie.mp4" type="video/mp4">

<source src="movie.ogg" type="video/ogg">

Your browser does not support the video tag.

</video>

</body>

</html>

## **Browser Support**

The numbers in the table specify the first browser version that fully supports the <video> element.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element |  |  |  |  |  |
| <video> | 4.0 | 9.0 | 3.5 | 4.0 | 10.5 |

## **HTML Video Formats**

There are three supported video formats: MP4, WebM, and Ogg. The browser support for the different formats is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser** | **MP4** | **WebM** | **Ogg** |
| Edge | YES | YES | YES |
| Chrome | YES | YES | YES |
| Firefox | YES | YES | YES |
| Safari | YES | YES | NO |
| Opera | YES | YES | YES |

## **HTML Video - Media Types**

|  |  |
| --- | --- |
| **File Format** | **Media Type** |
| MP4 | video/mp4 |
| WebM | video/webm |
| Ogg | video/ogg |

## **HTML Video - Methods, Properties, and Events**

The HTML DOM defines methods, properties, and events for the <video> element.

This allows you to load, play, and pause videos, as well as setting duration and volume.

There are also DOM events that can notify you when a video begins to play, is paused, etc.

### **Example: Using JavaScript**

For a full DOM reference, go to our [HTML Audio/Video DOM Reference](https://www.w3schools.com/tags/ref_av_dom.asp).

## **HTML Video Tags**

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<video>](https://www.w3schools.com/tags/tag_video.asp) | Defines a video or movie |
| [<source>](https://www.w3schools.com/tags/tag_source.asp) | Defines multiple media resources for media elements, such as <video> and <audio> |
| [<track>](https://www.w3schools.com/tags/tag_track.asp) | Defines text tracks in media players |

# HTML Audio

The HTML <audio> element is used to play an audio file on a web page.

## **The HTML <audio> Element**

To play an audio file in HTML, use the <audio> element:

### **Example**

# <!DOCTYPE html>

# <html>

# <body>

# <audio controls>

# <source src="horse.ogg" type="audio/ogg">

# <source src="horse.mp3" type="audio/mpeg">

# Your browser does not support the audio element.

# </audio>

# </body>

# </html>

## **HTML Audio - How It Works**

The controls attribute adds audio controls, like play, pause, and volume.

The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

## **HTML <audio> Autoplay**

To start an audio file automatically, use the autoplay attribute:

### **Example**

## **HTML Audio - How It Works**

The controls attribute adds audio controls, like play, pause, and volume.

The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

## **HTML <audio> Autoplay**

To start an audio file automatically, use the autoplay attribute:

### **Example**

<!DOCTYPE html>

<html>

<body>

<audio controls autoplay>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

</body>

</html>

## **Browser Support**

The numbers in the table specify the first browser version that fully supports the <audio> element.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element |  |  |  |  |  |
| <audio> | 4.0 | 9.0 | 3.5 | 4.0 | 10.5 |

## **HTML Audio Formats**

There are three supported audio formats: MP3, WAV, and OGG. The browser support for the different formats is:

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser** | **MP3** | **WAV** | **OGG** |
| Edge/IE | YES | YES\* | YES\* |
| Chrome | YES | YES | YES |
| Firefox | YES | YES | YES |
| Safari | YES | YES | NO |
| Opera | YES | YES | YES |

\*From Edge 79

## **HTML Audio - Media Types**

|  |  |
| --- | --- |
| **File Format** | **Media Type** |
| MP3 | audio/mpeg |
| OGG | audio/ogg |
| WAV | audio/wav |

## HTML Audio - Methods, Properties, and Events

The HTML DOM defines methods, properties, and events for the <audio> element.

This allows you to load, play, and pause audios, as well as set duration and volume.

There are also DOM events that can notify you when an audio begins to play, is paused, etc.

## **HTML Audio Tags**

|  |  |
| --- | --- |
| Tag | Description |
| [<audio>](https://www.w3schools.com/tags/tag_audio.asp) | Defines sound content |
| [<source>](https://www.w3schools.com/tags/tag_source.asp) | Defines multiple media resources for media elements, such as <video> and <audio> |

# HTML Plug-ins

Plug-ins are computer programs that extend the standard functionality of the browser.

## **Plug-ins**

Plug-ins were designed to be used for many different purposes:

* To run Java applets
* To run Microsoft ActiveX controls
* To display Flash movies
* To display maps
* To scan for viruses
* To verify a bank id

**Warning !**

Most browsers no longer support Java Applets and Plug-ins.

ActiveX controls are no longer supported in any browsers.

The support for Shockwave Flash has also been turned off in modern browsers.

## **The <object> Element**

The <object> element is supported by all browsers.

The <object> element defines an embedded object within an HTML document.

It was designed to embed plug-ins (like Java applets, PDF readers, and Flash Players) in web pages, but can also be used to include HTML in HTML:

### **Example**

<!DOCTYPE html>

<html>

<body>

<object width="100%" height="500px" data="snippet.html"></object>

</body>

</html>

**OUTPUT**

|  |  |  |
| --- | --- | --- |
| Alfreds Futterkiste | Berlin | Germany |
| Berglunds snabbköp | Luleå | Sweden |
| Centro comercial Moctezuma | México D.F. | Mexico |
| Ernst Handel | Graz | Austria |
| FISSA Fabrica Inter. Salchichas S.A. | Madrid | Spain |
| Galería del gastrónomo | Barcelona | Spain |
| Island Trading | Cowes | UK |
| Königlich Essen | Brandenburg | Germany |
| Laughing Bacchus Wine Cellars | Vancouver | Canada |
| Magazzini Alimentari Riuniti | Bergamo | Italy |
| North/South | London | UK |
| Paris spécialités | Paris | France |
| Rattlesnake Canyon Grocery | Albuquerque | USA |
| Simons bistro | København | Denmark |

**EXAMPLE**  
<!DOCTYPE html>

<html>

<body>

<object data="audi.jpeg"></object>

</body>

</html>

## **The <embed> Element**

The <embed> element is supported in all major browsers.

The <embed> element also defines an embedded object within an HTML document.

Web browsers have supported the <embed> element for a long time. However, it has not been a part of the HTML specification before HTML5.

### **Example**

<!DOCTYPE html>

<html>

<body>

<embed src="audi.jpeg">

</body>

</html>

Note that the <embed> element does not have a closing tag. It can not contain alternative text.

The <embed> element can also be used to include HTML in HTML:

**Example**

### <!DOCTYPE html>

### <html>

### <body>

### <embed width="100%" height="500px" src="snippet.html">

### </body>

### </html>

### **OUTPUT**

|  |  |  |
| --- | --- | --- |
| Alfreds Futterkiste | Berlin | Germany |
| Berglunds snabbköp | Luleå | Sweden |
| Centro comercial Moctezuma | México D.F. | Mexico |
| Ernst Handel | Graz | Austria |
| FISSA Fabrica Inter. Salchichas S.A. | Madrid | Spain |
| Galería del gastrónomo | Barcelona | Spain |
| Island Trading | Cowes | UK |
| Königlich Essen | Brandenburg | Germany |
| Laughing Bacchus Wine Cellars | Vancouver | Canada |
| Magazzini Alimentari Riuniti | Bergamo | Italy |
| North/South | London | UK |
| Paris spécialités | Paris | France |
| Rattlesnake Canyon Grocery | Albuquerque | USA |
| Simons bistro | København | Denmark |
| The Big Cheese | Portland | USA |
| Vaffeljernet | Århus | Denmark |
| Wolski Zajazd | Warszawa | Poland |

Bottom of Form

Bottom of Form

# HTML YouTube Videos

The easiest way to play videos in HTML, is to use YouTube.

## **Struggling with Video Formats?**

Converting videos to different formats can be difficult and time-consuming.

An easier solution is to let YouTube play the videos in your web page.

## **YouTube Video Id**

YouTube will display an id (like tgbNymZ7vqY), when you save (or play) a video.

You can use this id, and refer to your video in the HTML code.

## **Playing a YouTube Video in HTML**

To play your video on a web page, do the following:

* Upload the video to YouTube
* Take a note of the video id
* Define an <iframe> element in your web page
* Let the src attribute point to the video URL
* Use the width and height attributes to specify the dimension of the player
* Add any other parameters to the URL (see below)

### **Example**

### <!DOCTYPE html>

### <html>

### <body>

### <iframe width="420" height="345" src="https://www.youtube.com/embed/tgbNymZ7vqY">

### </iframe>

### </body>

## **YouTube Autoplay + Mute**

You can let your video start playing automatically when a user visits the page, by adding autoplay=1 to the YouTube URL. However, automatically starting a video is annoying for your visitors!

**Note:** Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

Add mute=1 after autoplay=1 to let your video start playing automatically (but muted).

### **YouTube - Autoplay + Muted**

## **YouTube Autoplay + Mute**

You can let your video start playing automatically when a user visits the page, by adding autoplay=1 to the YouTube URL. However, automatically starting a video is annoying for your visitors!

**Note:** Chromium browsers do not allow autoplay in most cases. However, muted autoplay is always allowed.

Add mute=1 after autoplay=1 to let your video start playing automatically (but muted).

### **YouTube - Autoplay + Muted**

## **YouTube Playlist**

A comma separated list of videos to play (in addition to the original URL).

## **YouTube Loop**

Add loop=1 to let your video loop forever.

Value 0 (default): The video will play only once.

Value 1: The video will loop (forever).

### **YouTube - Loop**

## **YouTube Controls**

Add controls=0 to not display controls in the video player.

Value 0: Player controls does not display.

Value 1 (default): Player controls display.

### **YouTube - Controls**

<iframe width="420" height="315"  
src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">  
</iframe>

**OUTPUT**

<!DOCTYPE html>

<html>

<body>

<iframe width="420" height="345" src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=0">

</iframe>

</body>

</html>

# HTML Geolocation API

The HTML Geolocation API is used to locate a user's position.

## **Locate the User's Position**

The HTML Geolocation API is used to get the geographical position of a user.

Since this can compromise privacy, the position is not available unless the user approves it.

Try It

**Note:**Geolocation is most accurate for devices with GPS, like smartphones.

## **Browser Support**

The numbers in the table specify the first browser version that fully supports Geolocation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| Geolocation | 5.0 - 49.0 (http) 50.0 (https) | 9.0 | 3.5 | 5.0 | 16.0 |

**Note:** As of Chrome 50, the Geolocation API will only work on secure contexts such as HTTPS. If your site is hosted on an non-secure origin (such as HTTP) the requests to get the users location will no longer function.

## **Using HTML Geolocation**

The getCurrentPosition() method is used to return the user's position.

The example below returns the latitude and longitude of the user's position:

### **Example**

<script>  
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>

Example explained:

* Check if Geolocation is supported
* If supported, run the getCurrentPosition() method. If not, display a message to the user
* If the getCurrentPosition() method is successful, it returns a coordinates object to the function specified in the parameter (showPosition)
* The showPosition() function outputs the Latitude and Longitude

The example above is a very basic Geolocation script, with no error handling.

## **Handling Errors and Rejections**

The second parameter of the getCurrentPosition() method is used to handle errors. It specifies a function to run if it fails to get the user's location:

### **Example**

function showError(error) {  
  switch(error.code) {  
    case error.PERMISSION\_DENIED:  
      x.innerHTML = "User denied the request for Geolocation."  
      break;  
    case error.POSITION\_UNAVAILABLE:  
      x.innerHTML = "Location information is unavailable."  
      break;  
    case error.TIMEOUT:  
      x.innerHTML = "The request to get user location timed out."  
      break;  
    case error.UNKNOWN\_ERROR:  
      x.innerHTML = "An unknown error occurred."  
      break;  
  }  
}

## **Location-specific Information**

This page has demonstrated how to show a user's position on a map.

Geolocation is also very useful for location-specific information, like:

* Up-to-date local information
* Showing Points-of-interest near the user

|  |  |
| --- | --- |
| **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) |

* Turn-by-turn navigation (GPS)

## **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:

## **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 API

In HTML, any element can be dragged and dropped.

## **Example**

W3Schools

Drag the W3Schools image into the rectangle.

## **Drag and Drop**

Drag and drop is a very common feature. It is when you "grab" an object and drag it to a different location.

## **Browser Support**

The numbers in the table specify the first browser version that fully supports Drag and Drop.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| Drag and Drop | 4.0 | 9.0 | 3.5 | 6.0 | 12.0 |

## **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>

It might seem complicated, but lets go through all the different parts of a drag and drop event.

## **Make an Element Draggable**

First of all: To make an element draggable, set the draggable attribute to true:

<img draggable="true">

## **What to Drag - ondragstart and setData()**

Then, specify what should happen when the element is dragged.

In the example above, the ondragstart attribute calls a function, drag(event), that specifies what data to be dragged.

The dataTransfer.setData() method sets the data type and the value of the dragged data:

function drag(ev) {  
  ev.dataTransfer.setData("text", ev.target.id);  
}

In this case, the data type is "text" and the value is the id of the draggable element ("drag1").

## **Where to Drop - ondragover**

The ondragover event specifies where the dragged data can be dropped.

By default, data/elements cannot be dropped in other elements. To allow a drop, we must prevent the default handling of the element.

This is done by calling the event.preventDefault() method for the ondragover event:

*event*.preventDefault()

## **Do the Drop - ondrop**

When the dragged data is dropped, a drop event occurs.

In the example above, the ondrop attribute calls a function, drop(event):

function drop(ev) {  
  ev.preventDefault();  
  var data = ev.dataTransfer.getData("text");  
  ev.target.appendChild(document.getElementById(data));  
}

Code explained:

* Call preventDefault() to prevent the browser default handling of the data (default is open as link on drop)
* Get the dragged data with the dataTransfer.getData() method. This method will return any data that was set to the same type in the setData() method
* The dragged data is the id of the dragged element ("drag1")
* Append the dragged element into the drop element

## **More Examples**

### **Example**

How to drag (and drop) an image back and forth between two <div> elements:

# HTML Web Storage API

HTML web storage; better than cookies.

## **What is HTML Web Storage?**

With web storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

## **Browser Support**

The numbers in the table specify the first browser version that fully supports Web Storage.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| Web Storage | 4.0 | 8.0 | 3.5 | 4.0 | 11.5 |

## **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..  
}

## **The localStorage Object**

The localStorage object stores the data with no expiration date. The data will not be deleted when the browser is closed, and will be available the next day, week, or year.

### **Example**

// Store  
localStorage.setItem("lastname", "Smith");  
  
// Retrieve  
document.getElementById("result").innerHTML = localStorage.getItem("lastname");

Example explained:

* Create a localStorage name/value pair with name="lastname" and value="Smith"
* Retrieve the value of "lastname" and insert it into the element with id="result"

The example above could also be written like this:

// Store  
localStorage.lastname = "Smith";  
// Retrieve  
document.getElementById("result").innerHTML = localStorage.lastname;

The syntax for removing the "lastname" localStorage item is as follows:

localStorage.removeItem("lastname");

**Note:** Name/value pairs are always stored as strings. Remember to convert them to another format when needed!

The following example counts the number of times a user has clicked a button. In this code the value string is converted to a number to be able to increase the counter:

### **Example**

if (localStorage.clickcount) {  
  localStorage.clickcount = Number(localStorage.clickcount) + 1;  
} else {  
  localStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
localStorage.clickcount + " time(s).";

## **The sessionStorage Object**

The sessionStorage object is equal to the localStorage object, **except** that it stores the data for only one session. The data is deleted when the user closes the specific browser tab.

The following example counts the number of times a user has clicked a button, in the current session:

### **Example**

if (sessionStorage.clickcount) {  
  sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  
} else {  
  sessionStorage.clickcount = 1;  
}  
document.getElementById("result").innerHTML = "You have clicked the button " +  
sessionStorage.clickcount + " time(s) in this session.";

# HTML Web Workers API

A web worker is a JavaScript running in the background, without affecting the performance of the page.

## **What is a Web Worker?**

When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.

A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

## **Browser Support**

The numbers in the table specify the first browser version that fully support Web Workers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| Web Workers | 4.0 | 10.0 | 3.5 | 4.0 | 11.5 |

## **HTML Web Workers Example**

The example below creates a simple web worker that count numbers in the background:

### **Example**

Count numbers:

Start Worker Stop Worker

## **Check Web Worker Support**

Before creating a web worker, check whether the user's browser supports it:

if (typeof(Worker) !== "undefined") {  
  // Yes! Web worker support!  
  // *Some code.....*  
} else {  
  // Sorry! No Web Worker support..  
}

## **Create a Web Worker File**

Now, let's create our web worker in an external JavaScript.

Here, we create a script that counts. The script is stored in the "demo\_workers.js" file:

var i = 0;  
  
function timedCount() {  
  i = i + 1;  
  postMessage(i);  
  setTimeout("timedCount()",500);  
}  
  
timedCount();

The important part of the code above is the postMessage() method - which is used to post a message back to the HTML page.

**Note:** Normally web workers are not used for such simple scripts, but for more CPU intensive tasks.

## **Create a Web Worker Object**

Now that we have the web worker file, we need to call it from an HTML page.

The following lines checks if the worker already exists, if not - it creates a new web worker object and runs the code in "demo\_workers.js":

if (typeof(w) == "undefined") {  
  w = new Worker("demo\_workers.js");  
}

Then we can send and receive messages from the web worker.

Add an "onmessage" event listener to the web worker.

w.onmessage = function(event){  
  document.getElementById("result").innerHTML = event.data;  
};

When the web worker posts a message, the code within the event listener is executed. The data from the web worker is stored in event.data.

## **Terminate a Web Worker**

When a web worker object is created, it will continue to listen for messages (even after the external script is finished) until it is terminated.

To terminate a web worker, and free browser/computer resources, use the terminate() method:

w.terminate();

## **Reuse the Web Worker**

If you set the worker variable to undefined, after it has been terminated, you can reuse the code:

w = undefined;

## **Full Web Worker Example Code**

We have already seen the Worker code in the .js file. Below is the code for the HTML page:

### **Example**

<!DOCTYPE html>  
<html>  
<body>  
<p>Count numbers: <output id="result"></output></p>  
<button onclick="startWorker()">Start Worker</button>  
<button onclick="stopWorker()">Stop Worker</button>  
<script>  
var w;  
function startWorker() {  
  if (typeof(Worker) !== "undefined") {  
    if (typeof(w) == "undefined") {  
      w = new Worker("demo\_workers.js");  
    }  
    w.onmessage = function(event) {  
      document.getElementById("result").innerHTML = event.data;  
    };  
  } else {  
    document.getElementById("result").innerHTML = "Sorry! No Web Worker support.";  
  }  
}  
  
function stopWorker() {  
  w.terminate();  
  w = undefined;  
}  
</script>  
  
</body>  
</html>

## **Web Workers and the DOM**

Since web workers are in external files, they do not have access to the following JavaScript objects:

* The window object
* The document object
* The parent object

# HTML SSE API

Server-Sent Events (SSE) allow a web page to get updates from a server.

## **Server-Sent Events - One Way Messaging**

A server-sent event is when a web page automatically gets updates from a server.

This was also possible before, but the web page would have to ask if any updates were available. With server-sent events, the updates come automatically.

Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.

## **Browser Support**

The numbers in the table specify the first browser version that fully support server-sent events.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| API |  |  |  |  |  |
| SSE | 6.0 | 79.0 | 6.0 | 5.0 | 11.5 |

## **Receive Server-Sent Event Notifications**

The EventSource object is used to receive server-sent event notifications:

### **Example**

var source = new EventSource("demo\_sse.php");  
source.onmessage = function(event) {  
  document.getElementById("result").innerHTML += event.data + "<br>";  
};

Example explained:

* Create a new EventSource object, and specify the URL of the page sending the updates (in this example "demo\_sse.php")
* Each time an update is received, the onmessage event occurs
* When an onmessage event occurs, put the received data into the element with id="result"

## **Check Server-Sent Events Support**

In the tryit example above there were some extra lines of code to check browser support for server-sent events:

if(typeof(EventSource) !== "undefined") {  
  // Yes! Server-sent events support!  
  // *Some code.....*  
} else {  
  // Sorry! No server-sent events support..  
}

## **Server-Side Code Example**

For the example above to work, you need a server capable of sending data updates (like PHP or ASP).

The server-side event stream syntax is simple. Set the "Content-Type" header to "text/event-stream". Now you can start sending event streams.

Code in PHP (demo\_sse.php):

<?php  
header('Content-Type: text/event-stream');  
header('Cache-Control: no-cache');  
  
$time = date('r');  
echo "data: The server time is: {$time}\n\n";  
flush();  
?>

Code in ASP (VB) (demo\_sse.asp):

<%  
Response.ContentType = "text/event-stream"  
Response.Expires = -1  
Response.Write("data: The server time is: " & now())  
Response.Flush()  
%>

Code explained:

* Set the "Content-Type" header to "text/event-stream"
* Specify that the page should not cache
* Output the data to send (**Always** start with "data: ")
* Flush the output data back to the web page

## **The EventSource Object**

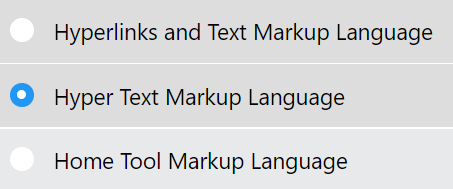
In the examples above we used the onmessage event to get messages. But other events are also available:

|  |  |
| --- | --- |
| **Events** | **Description** |
| onopen | When a connection to the server is opened |
| onmessage | When a message is received |
| onerror | When an error occurs |

# HTML Quiz

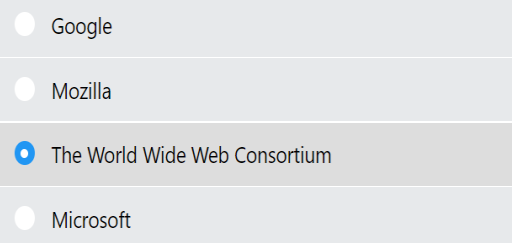
### **Question 1:**

What does HTML stand for?



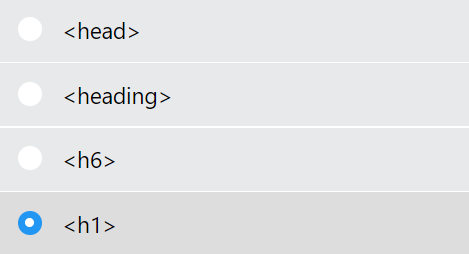
### **Question 2:**

Who is making the Web standards?



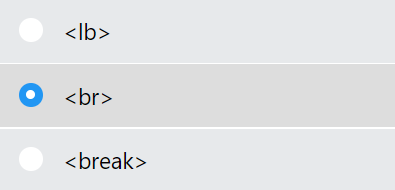
Question 3:

Choose the correct HTML element for the largest heading:Top of Form



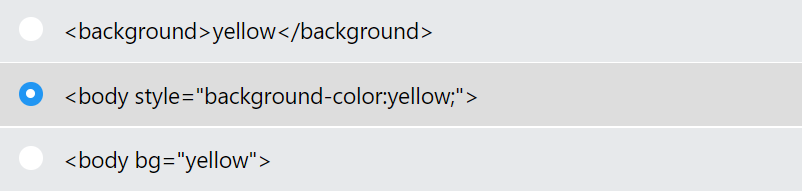
Question 4:

What is the correct HTML element for inserting a line break?



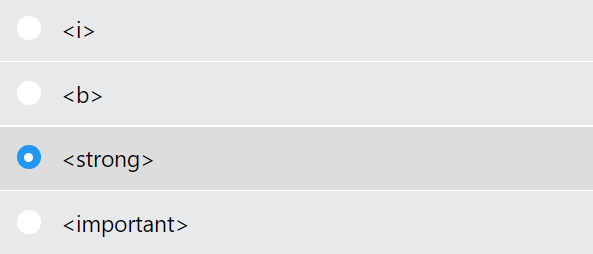
### **Question 5:**

What is the correct HTML for adding a background color?



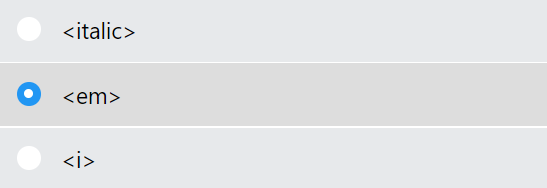
### **Question 6:**

Choose the correct HTML element to define important text



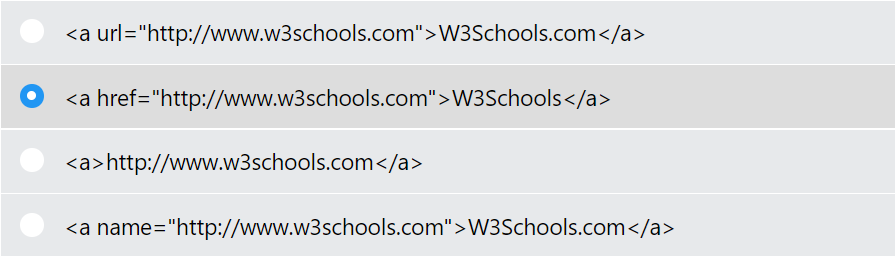
### **Question 7:**

Choose the correct HTML element to define emphasized text



### **Question 8:**

What is the correct HTML for creating a hyperlink?



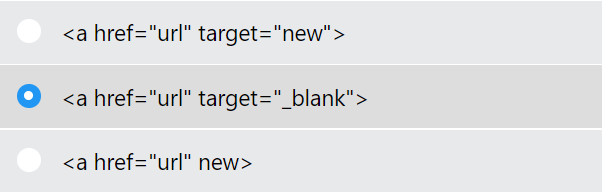
### **Question 9:**

Which character is used to indicate an end tag?



Question 10:

How can you open a link in a new tab/browser window?



Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form