HTML stands for **H**yper**t**ext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

* **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus the link available on a webpage are called Hypertext.
* As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Basic HTML Document

In its simplest form, following is an example of an HTML document:

<!DOCTYPE html>

<html>

<head>

<title>This is document title</title>

</head>

<body>

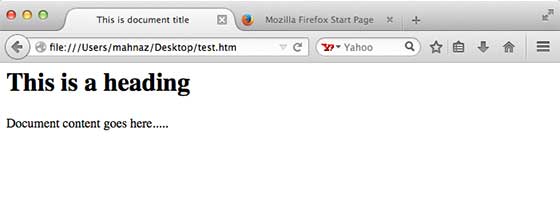
<h1>This is a heading</h1>

<p>Document content goes here.....</p>

</body>

</html>

Either you can use **Try it** option available at the top right corner of the code box to check the result of this HTML code, or let's save it in an HTML file **test.htm** using your favorite text editor. Finally open it using a web browser like Internet Explorer or Google Chrome, or Firefox etc. It must show the following output:



HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example **<html>** has its closing tag**</html>** and **<body>** tag has its closing tag **</body>** tag etc.

Above example of HTML document uses folloiwng tags:

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <!DOCTYPE...> | This tag defines the document type and HTML version. |
| <html> | This tag encloses the complete HTML document and mainly comprises of document header which is represented by **<head>...</head>** and document body which is represented by **<body>...</body>** tags. |
| <head> | This tag represents the document's header which can keep other HTML tags like <title>, <link> etc. |
| <title> | The **<title>** tag is used inside the <head> tag to mention the document title. |
| <body> | This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc. |
| <h1> | This tag represents the heading. |
| <p> | This tag represents a paragraph. |

To learn HTML, you will need to study various tags and understand how do they behave while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

HTML Document Structure

A typical HTML document will have following structure:

Document declaration tag

<html>

<head>

Document header related tags

</head>

<body>

Document body related tags

</body>

</html>

We will study all the header and body tags in subsequent chapters, for now let's see what is document declaration tag.

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

<!DOCTYPE html>

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing <!DOCTYPE...> tag along with other HTML tags.

## Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>**. While displaying any heading, browser adds one line before and one line after that heading.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Heading Example</title>

</head>

<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>

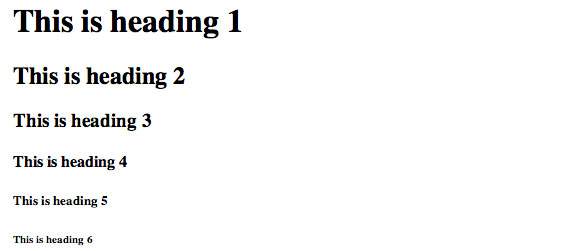
<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

This will produce following result:



## Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag as shown below in the example:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Paragraph Example</title>

</head>

<body>

<p>Here is a first paragraph of text.</p>

<p>Here is a second paragraph of text.</p>

<p>Here is a third paragraph of text.</p>

</body>

</html>

This will produce following result:

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

## Line Break Tag

Whenever you use the **<br />** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <br /> tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid in XHTML

### Example

<!DOCTYPE html>

<html>

<head>

<title>Line Break Example</title>

</head>

<body>

<p>Hello<br />

You delivered your assignment ontime.<br />

Thanks<br />

Mahnaz</p>

</body>

</html>

This will produce following result:

Hello  
You delivered your assignment ontime.  
Thanks  
Mahnaz

## Centering Content

You can use **<center>** tag to put any content in the center of the page or any table cell.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Centring Content Example</title>

</head>

<body>

<p>This text is not in the center.</p>

<center>

<p>This text is in the center.</p>

</center>

</body>

</html>

This will produce following result:

This text is not in the center.

This text is in the center.

## Horizontal Lines

Horizontal lines are used to visually break up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example you may want to give a line between two paragraphs as in the given example below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Horizontal Line Example</title>

</head>

<body>

<p>This is paragraph one and should be on top</p>

<hr />

<p>This is paragraph two and should be at bottom</p>

</body>

</html>

This will produce following result:

This is paragraph one and should be on top

This is paragraph two and should be at bottom

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <hr /> element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontak line, while if you miss the forward slash character and just use <hr> it is not valid in XHTML

## Preserve Formatting

Sometimes you want your text to follow the exact format of how it is written in the HTML document. In those cases, you can use the preformatted tag <pre>.

Any text between the opening <pre> tag and the closing </pre> tag will preserve the formatting of the source document.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Preserve Formatting Example</title>

</head>

<body>

<pre>

function testFunction( strText ){

alert (strText)

}

</pre>

</body>

</html>

This will produce following result:

function testFunction( strText ){

alert (strText)

}

Try using same code without keeping it inside <pre>...</pre> tags

## Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here you would not want a browser to split the "12, Angry" and "Men" across two lines:

An example of this technique appears in the movie "12 Angry Men."

In cases where you do not want the client browser to break text, you should use a nonbreaking space entity **&nbsp;** instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Nonbreaking Spaces Example</title>

</head>

<body>

<p>An example of this technique appears in the movie "12&nbsp;Angry&nbsp;Men."</p>

</body>

</html>

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags:

|  |  |  |
| --- | --- | --- |
| **Start Tag** | **Content** | **End Tag** |
| <p> | This is paragraph content. | </p> |
| <h1> | This is heading content. | </h1> |
| <div> | This is division content. | </div> |
| <br /> |  |  |

So here <p>....</p> is an HTML element, <h1>...</h1> is another HTML element. There are some HTML elements which don't need to be closed, such as <img.../>, <hr /> and <br /> elements. These are known as **void elements**.

HTML documents consist of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

## HTML Tag vs. Element

An HTML element is defined by a *starting tag*. If the element contains other content, it ends with a *closing tag*.

For example <p> is starting tag of a paragraph and </p> is closing tag of the same paragraph but **<p>This is paragraph</p>** is a paragraph element.

## Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Nested Elements Example</title>

</head>

<body>

<h1>This is <i>italic</i> heading</h1>

<p>This is <u>underlined</u> paragraph</p>

</body>

</html>

This will display following result:

# This is *italic* heading

This is underlined paragraph

We have seen few HTML tags and their usage like heading tags <h1>, <h2>, paragraph tag <p> and other tags. We used them so far in their simplest form, but most of the HTML tags can also have attributes, which are extra bits of information.

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts: a **name** and a **value**:

* The **name** is the property you want to set. For example, the paragraph <p> element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
* The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left, center** and **right**.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Align Attribute Example</title>

</head>

<body>

<p align="left">This is left aligned</p>

<p align="center">This is center aligned</p>

<p align="right">This is right aligned</p>

</body>

</html>

This will display following result:

This is left aligned

This is center aligned

This is right aligned

## Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are:

* id
* title
* class
* style

### The id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element:

* If an element carries an id attribute as a unique identifier it is possible to identify just that element and its content.
* If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

We will discuss style sheet in separate tutorial. For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

#### EXAMPLE

<p id="html">This para explains what is HTML</p>

<p id="css">This para explains what is Cascading Style Sheet</p>

### The title Attribute

The **title** attribute gives a suggested title for the element. They syntax for the **title** attribute is similar as explained for **id** attribute:

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

#### EXAMPLE

<!DOCTYPE html>

<html>

<head>

<title>The title Attribute Example</title>

</head>

<body>

<h3 title="Hello HTML!">Titled Heading Tag Example</h3>

</body>

</html>

This will produce following result:

### Titled Heading Tag Example

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

### The class Attribute

The **class** attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the use of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it.

The value of the attribute may also be a space-separated list of class names. For example:

class="className1 className2 className3"

### The style Attribute

The style attribute allows you to specify Casecading Style Sheet (CSS) rules within the element.

<!DOCTYPE html>

<html>

<head>

<title>The style Attribute</title>

</head>

<body>

<p style="font-family:arial; color:#FF0000;">Some text...</p>

</body>

</html>

This will produce following result:

Some text...

At this point of time, we are not learning CSS, so just let's proceed without bothering much about CSS. Here you need to understand what are HTML attributes and how they can be used while formatting content.

## Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

* dir
* lang
* xml:lang

### The dir Attribute

The **dir** attribute allows you to indicate to the browser the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows:

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| ltr | Left to right (the default value) |
| rtl | Right to left (for languages such as Hebrew or Arabic that are read right to left) |

#### EXAMPLE

<!DOCTYPE html>

<html dir="rtl">

<head>

<title>Display Directions</title>

</head>

<body>

This is how IE 5 renders right-to-left directed text.

</body>

</html>

This will produce following result:

This is how IE 5 renders right-to-left directed text.

When *dir* attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

### The lang Attribute

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the **xml:lang** attribute in new XHTML documents.

The values of the *lang* attribute are ISO-639 standard two-character language codes. Check[**HTML Language Codes: ISO 639**](http://www.tutorialspoint.com/html/language_iso_codes.htm) for a complete list of language codes.

#### EXAMPLE

<!DOCTYPE html>

<html lang="en">

<head>

<title>English Language Page</title>

</head>

<body>

This page is using English Language

</body>

</html>

### The xml:lang Attribute

The *xml:lang* attribute is the XHTML replacement for the *lang* attribute. The value of the*xml:lang* attribute should be an ISO-639 country code as mentioned in previous section.

## Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Options** | **Function** |
| align | right, left, center | Horizontally aligns tags |
| valign | top, middle, bottom | Vertically aligns tags within an HTML element. |
| bgcolor | numeric, hexidecimal, RGB values | Places a background color behind an element |
| background | URL | Places a background image behind an element |
| Id | User Defined | Names an element for use with Cascading Style Sheets. |
| class | User Defined | Classifies an element for use with Cascading Style Sheets. |
| width | Numeric Value | Specifies the width of tables, images, or table cells. |
| height | Numeric Value | Specifies the height of tables, images, or table cells. |
| title | User Defined | "Pop-up" title of the elements. |

We will see related examples as we will proceed to study other HTML tags.

A complete list of standard tags from HTML 4.01 / XHTML 1.0 is given here. All the tags are ordered alphabetically.

We have used the Netscape and Internet Explorer indications to the far right of each item to indicate tags that are extensions to the HTML 4.01 and XHTML 1.0 standards. If no indication is shown, then the tag is part of the HTML 4.01 and XHTML 1.0 standards.

* **NS**: indicates Netscape Extension
* **IE**: indicates Internet Explorer Extension

There are many modifications have been done in HTML 4. For a complete detail please refer to[What is new in HTML 4?](http://www.tutorialspoint.com/html/what_is_new_in_html4.htm)

|  |  |  |
| --- | --- | --- |
| **Tag** | **Description** | **Browser** |
| [<!--...-->](http://www.tutorialspoint.com/html/html_comment_tag.htm) | Specifies a comment |  |
| [<!DOCTYPE>](http://www.tutorialspoint.com/html/html_doctype_tag.htm) | Specifies the document type |  |
| [<a>](http://www.tutorialspoint.com/html/html_a_tag.htm) | Specifies an anchor |  |
| [<abbr>](http://www.tutorialspoint.com/html/html_abbr_tag.htm) | Specifies an abbreviation |  |
| [<acronym>](http://www.tutorialspoint.com/html/html_acronym_tag.htm) | Specifies an acronym |  |
| [<address>](http://www.tutorialspoint.com/html/html_address_tag.htm) | Specifies an address element |  |
| [<applet>](http://www.tutorialspoint.com/html/html_applet_tag.htm) | **Deprecated.** Specifies an applet |  |
| [<area>](http://www.tutorialspoint.com/html/html_area_tag.htm) | Specifies an area inside an image map |  |
| [<b>](http://www.tutorialspoint.com/html/html_font_style_tag.htm) | Specifies bold text |  |
| [<base>](http://www.tutorialspoint.com/html/html_base_tag.htm) | Specifies a base URL for all the links in a page |  |
| [<basefont>](http://www.tutorialspoint.com/html/html_basefont_tag.htm) | **Deprecated.** Specifies a base font |  |
| [<bdo>](http://www.tutorialspoint.com/html/html_bdo_tag.htm) | Specifies the direction of text display |  |
| [<bgsound>](http://www.tutorialspoint.com/html/html_bgsound_tag.htm) | Specifies the background music | IE |
| [<big>](http://www.tutorialspoint.com/html/html_font_style_tag.htm) | Specifies big text |  |
| [<blink>](http://www.tutorialspoint.com/html/html_blink_tag.htm) | Specifies a text which blinks | NS |
| [<blockquote>](http://www.tutorialspoint.com/html/html_blockquote_tag.htm) | Specifies a long quotation |  |
| [<body>](http://www.tutorialspoint.com/html/html_body_tag.htm) | Specifies the body element |  |
| [<br>](http://www.tutorialspoint.com/html/html_br_tag.htm) | Inserts a single line break |  |
| [<button>](http://www.tutorialspoint.com/html/html_button_tag.htm) | Specifies a push button |  |
| [<caption>](http://www.tutorialspoint.com/html/html_caption_tag.htm) | Specifies a table caption |  |
| [<center>](http://www.tutorialspoint.com/html/html_center_tag.htm) | **Deprecated.** Specifies centered text |  |
| [<cite>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies a citation |  |
| [<code>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies computer code text |  |
| [<col>](http://www.tutorialspoint.com/html/html_col_tag.htm) | Specifies attributes for table columns |  |
| [<colgroup>](http://www.tutorialspoint.com/html/html_colgroup_tag.htm) | Specifies groups of table columns |  |
| [<comment>](http://www.tutorialspoint.com/html/html_comment_tag.htm) | Puts a comment in the document | IE |
| [<dd>](http://www.tutorialspoint.com/html/html_dd_tag.htm) | Specifies a definition description |  |
| [<del>](http://www.tutorialspoint.com/html/html_del_tag.htm) | Specifies deleted text |  |
| [<dfn>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies a definition term |  |
| [<dir>](http://www.tutorialspoint.com/html/html_dir_tag.htm) | **Deprecated.** Specifies a directory list |  |
| [<div>](http://www.tutorialspoint.com/html/html_div_tag.htm) | Specifies a section in a document |  |
| [<dl>](http://www.tutorialspoint.com/html/html_dl_tag.htm) | Specifies a definition list |  |
| [<dt>](http://www.tutorialspoint.com/html/html_dt_tag.htm) | Specifies a definition term |  |
| [<em>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies emphasized text |  |
| [<embed>](http://www.tutorialspoint.com/html/html_embed_tag.htm) | **Deprecated.** Embeds an application in a document | IE & NS |
| [<fieldset>](http://www.tutorialspoint.com/html/html_fieldset_tag.htm) | Specifies a fieldset |  |
| [<font>](http://www.tutorialspoint.com/html/html_font_tag.htm) | **Deprecated.** Specifies text font, size, and color |  |
| [<form>](http://www.tutorialspoint.com/html/html_form_tag.htm) | Specifies a form |  |
| [<frame>](http://www.tutorialspoint.com/html/html_frame_tag.htm) | Specifies a sub window (a frame) |  |
| [<frameset>](http://www.tutorialspoint.com/html/html_frameset_tag.htm) | Specifies a set of frames |  |
| [<h1> to <h6>](http://www.tutorialspoint.com/html/html_hn_tag.htm) | Specifies header 1 to header 6 |  |
| [<head>](http://www.tutorialspoint.com/html/html_head_tag.htm) | Specifies information about the document |  |
| [<hr>](http://www.tutorialspoint.com/html/html_hr_tag.htm) | Specifies a horizontal rule |  |
| [<html>](http://www.tutorialspoint.com/html/html_html_tag.htm) | Specifies an html document |  |
| [<i>](http://www.tutorialspoint.com/html/html_font_style_tag.htm) | Specifies italic text |  |
| [<iframe>](http://www.tutorialspoint.com/html/html_iframe_tag.htm) | Specifies an inline sub window (frame) |  |
| [<ilayer>](http://www.tutorialspoint.com/html/html_ilayer_tag.htm) | Specifies an inline layer | NS |
| [<img>](http://www.tutorialspoint.com/html/html_img_tag.htm) | Specifies an image |  |
| [<input>](http://www.tutorialspoint.com/html/html_input_tag.htm) | Specifies an input field |  |
| [<ins>](http://www.tutorialspoint.com/html/html_ins_tag.htm) | Specifies inserted text |  |
| <isindex> | **Deprecated.** Specifies a single-line input field |  |
| [<kbd>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies keyboard text |  |
| [<keygen>](http://www.tutorialspoint.com/html/html_keygen_tag.htm) | Generate key information in a form |  |
| [<label>](http://www.tutorialspoint.com/html/html_label_tag.htm) | Specifies a label for a form control |  |
| [<layer>](http://www.tutorialspoint.com/html/html_layer_tag.htm) | Specifies a layer | NS |
| [<legend>](http://www.tutorialspoint.com/html/html_legend_tag.htm) | Specifies a title in a fieldset |  |
| [<li>](http://www.tutorialspoint.com/html/html_li_tag.htm) | Specifies a list item |  |
| [<link>](http://www.tutorialspoint.com/html/html_link_tag.htm) | Specifies a resource reference |  |
| [<map>](http://www.tutorialspoint.com/html/html_map_tag.htm) | Specifies an image map |  |
| [<marquee>](http://www.tutorialspoint.com/html/html_marquee_tag.htm) | Create a scrolling-text marquee | IE |
| [<menu>](http://www.tutorialspoint.com/html/html_menu_tag.htm) | **Deprecated.** Specifies a menu list |  |
| [<meta>](http://www.tutorialspoint.com/html/html_meta_tag.htm) | Specifies meta information |  |
| [<multicol>](http://www.tutorialspoint.com/html/html_multicol_tag.htm) | Specifies a multicolumn text flow | NS |
| [<nobr>](http://www.tutorialspoint.com/html/html_nobr_tag.htm) | No breaks allowed in the enclosed text |  |
| [<noembed>](http://www.tutorialspoint.com/html/html_noembed_tag.htm) | Specifies content to be presented by browsers that do not support the <embed> tag | NS |
| [<noframes>](http://www.tutorialspoint.com/html/html_noframes_tag.htm) | Specifies a noframe section |  |
| [<noscript>](http://www.tutorialspoint.com/html/html_noscript_tag.htm) | Specifies a noscript section |  |
| [<object>](http://www.tutorialspoint.com/html/html_object_tag.htm) | Specifies an embedded object |  |
| [<ol>](http://www.tutorialspoint.com/html/html_ol_tag.htm) | Specifies an ordered list |  |
| [<optgroup>](http://www.tutorialspoint.com/html/html_optgroup_tag.htm) | Specifies an option group |  |
| [<option>](http://www.tutorialspoint.com/html/html_option_tag.htm) | Specifies an option in a drop-down list |  |
| [<p>](http://www.tutorialspoint.com/html/html_p_tag.htm) | Specifies a paragraph |  |
| [<param>](http://www.tutorialspoint.com/html/html_param_tag.htm) | Specifies a parameter for an object |  |
| [<plaintext>](http://www.tutorialspoint.com/html/html_plaintext_tag.htm) | **Deprecated.** Render the raminder of the document as preformatted plain text |  |
| [<pre>](http://www.tutorialspoint.com/html/html_pre_tag.htm) | Specifies preformatted text |  |
| [<q>](http://www.tutorialspoint.com/html/html_q_tag.htm) | Specifies a short quotation |  |
| [<s>](http://www.tutorialspoint.com/html/html_strike_tag.htm) | **Deprecated.** Specifies strikethrough text |  |
| [<samp>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies sample computer code |  |
| [<script>](http://www.tutorialspoint.com/html/html_script_tag.htm) | Specifies a script |  |
| [<select>](http://www.tutorialspoint.com/html/html_select_tag.htm) | Specifies a selectable list |  |
| [<spacer>](http://www.tutorialspoint.com/html/html_spacer_tag.htm) | Specifies a white space | NS |
| [<small>](http://www.tutorialspoint.com/html/html_font_style_tag.htm) | Specifies small text |  |
| [<span>](http://www.tutorialspoint.com/html/html_span_tag.htm) | Specifies a section in a document |  |
| [<strike>](http://www.tutorialspoint.com/html/html_strike_tag.htm) | **Deprecated.** Specifies strikethrough text |  |
| [<strong>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies strong text |  |
| [<style>](http://www.tutorialspoint.com/html/html_style_tag.htm) | Specifies a style definition |  |
| [<sub>](http://www.tutorialspoint.com/html/html_sup_tag.htm) | Specifies subscripted text |  |
| [<sup>](http://www.tutorialspoint.com/html/html_sup_tag.htm) | Specifies superscripted text |  |
| [<table>](http://www.tutorialspoint.com/html/html_table_tag.htm) | Specifies a table |  |
| [<tbody>](http://www.tutorialspoint.com/html/html_tbody_tag.htm) | Specifies a table body |  |
| [<td>](http://www.tutorialspoint.com/html/html_td_tag.htm) | Specifies a table cell |  |
| [<textarea>](http://www.tutorialspoint.com/html/html_textarea_tag.htm) | Specifies a text area |  |
| [<tfoot>](http://www.tutorialspoint.com/html/html_tfoot_tag.htm) | Specifies a table footer |  |
| [<th>](http://www.tutorialspoint.com/html/html_th_tag.htm) | Specifies a table header |  |
| [<thead>](http://www.tutorialspoint.com/html/html_thead_tag.htm) | Specifies a table header |  |
| [<title>](http://www.tutorialspoint.com/html/html_title_tag.htm) | Specifies the document title |  |
| [<tr>](http://www.tutorialspoint.com/html/html_tr_tag.htm) | Specifies a table row |  |
| [<tt>](http://www.tutorialspoint.com/html/html_font_style_tag.htm) | Specifies teletype text |  |
| [<u>](http://www.tutorialspoint.com/html/html_u_tag.htm) | **Deprecated.** Specifies underlined text |  |
| [<ul>](http://www.tutorialspoint.com/html/html_ul_tag.htm) | Specifies an unordered list |  |
| [<var>](http://www.tutorialspoint.com/html/html_phrase_elements_tag.htm) | Specifies a variable |  |
| [<wbr>](http://www.tutorialspoint.com/html/html_wbr_tag.htm) | Indicate a potential word break point within a <nobr> section | IE & NS |
| <xmp> | **Deprecated.** Specifies preformatted text |  |

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

## Bold Text

Anything that appears within **<b>...</b>** element, is displayed in bold as shown below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Bold Text Example</title>

</head>

<body>

<p>The following word uses a <b>bold</b> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a **bold** typeface.

## Italic Text

Anything that appears within **<i>...</i>** element is displayed in italicized as shown below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Italic Text Example</title>

</head>

<body>

<p>The following word uses a <i>italicized</i> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a *italicized* typeface.

## Underlined Text

Anything that appears within **<u>...</u>** element, is displayed with underline as shown below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Underlined Text Example</title>

</head>

<body>

<p>The following word uses a <u>underlined</u> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a underlined typeface.

## Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Strike Text Example</title>

</head>

<body>

<p>The following word uses a <strike>strikethrough</strike> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a ~~strikethrough~~ typeface.

## Monospaced Font

The content of a **<tt>...</tt>** element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Monospaced Font Example</title>

</head>

<body>

<p>The following word uses a <tt>monospaced</tt> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a monospaced typeface.

## Superscript Text

The content of a **<sup>...</sup>** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Superscript Text Example</title>

</head>

<body>

<p>The following word uses a <sup>superscript</sup> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a superscript typeface.

## Subscript Text

The content of a **<sub>...</sub>** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Subscript Text Example</title>

</head>

<body>

<p>The following word uses a <sub>subscript</sub> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a subscript typeface.

## Inserted Text

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Inserted Text Example</title>

</head>

<body>

<p>I want to drink <del>cola</del> <ins>wine</ins></p>

</body>

</html>

This will produce following result:

I want to drink  wine

## Deleted Text

Anything that appears within **<del>...</del>** element, is displayed as deleted text.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Deleted Text Example</title>

</head>

<body>

<p>I want to drink <del>cola</del> <ins>wine</ins></p>

</body>

</html>

This will produce following result:

I want to drink  wine

## Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it as shown below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Larger Text Example</title>

</head>

<body>

<p>The following word uses a <big>big</big> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a big typeface.

## Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it as shown below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Smaller Text Example</title>

</head>

<body>

<p>The following word uses a <small>small</small> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a small typeface.

## Grouping Content

The **<div>** and **<span>** elements allow you to group together several elements to create sections or subsections of a page.

For example, you might want to put all of the footnotes on a page within a <div> element to indicate that all of the elements within that <div> element relate to the footnotes. You might then attach a style to this <div> element so that they appear using a special set of style rules.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Div Tag Example</title>

</head>

<body>

<div id="menu" align="middle" >

<a href="/index.htm">HOME</a> |

<a href="/about/contact\_us.htm">CONTACT</a> |

<a href="/about/index.htm">ABOUT</a>

</div>

<div id="content" align="left" bgcolor="white">

<h5>Content Articles</h5>

<p>Actual content goes here.....</p>

</div>

</body>

</html>

This will produce following result:

[HOME](http://www.tutorialspoint.com/index.htm) | [CONTACT](http://www.tutorialspoint.com/about/contact_us.htm) | [ABOUT](http://www.tutorialspoint.com/about/index.htm)

##### CONTENT ARTICLES

Actual content goes here.....

The <span> element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the <span> element as follows

### Example

<!DOCTYPE html>

<html>

<head>

<title>Span Tag Example</title>

</head>

<body>

<p>This is the example of <span style="color:green">span tag</span> and the <span style="color:red">div tag</span> alongwith CSS</p>

</body>

</html>

This will produce following result:

This is the example of span tag and the div tag alongwith CSS

These tags are commonly used with CSS to allow you to attach a style to a section of a page.

The phrase tags have been designed for specific purposes, though they are displayed in a similar way as other basic tags like <b>, <i>, <pre>, and <tt>, you have seen in previous chapter. This chapter will take you through all the important phrase tags, so let's start seeing them one by one.

## Emphasized Text

Anything that appears within **<em>...</em>** element is displayed as emphasized text.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Emphasized Text Example</title>

</head>

<body>

<p>The following word uses a <em>emphasized</em> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a emphasized typeface.

## Marked Text

Anything that appears with-in **<mark>...</mark>** element, is displayed as marked with yellow ink.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Marked Text Example</title>

</head>

<body>

<p>The following word has been <mark>marked</mark> with yellow</p>

</body>

</html>

This will produce following result:

The following word has been marked with yellow.

## Strong Text

Anything that appears within **<strong>...</strong>** element is displayed as important text.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Strong Text Example</title>

</head>

<body>

<p>The following word uses a <strong>strong</strong> typeface.</p>

</body>

</html>

This will produce following result:

The following word uses a **strong** typeface.

## Text Abbreviation

You can abbreviate a text by putting it inside opening **<abbr>** and closing **</abbr>** tags. If present, the title attribute must contain this full description and nothing else.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Text Abbreviation</title>

</head>

<body>

<p>My best friend's name is <abbr title="Abhishek">Abhy</abbr>.</p>

</body>

</html>

This will produce following result:

My best friend's name is Abhy.

## Acronym Element

The **<acronym>** element allows you to indicate that the text between <acronym> and </acronym> tags is an acronym.

At present, the major browsers do not change the appearance of the content of the <acronym> element.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Acronym Example</title>

</head>

<body>

<p>This chapter covers marking up text in <acronym>XHTML</acronym>.</p>

</body>

</html>

This will produce following result:

This chapter covers marking up text in XHTML.

## Text Direction

The **<bdo>...</bdo>** element stands for Bi-Directional Override and it is used to override the current text direction.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Text Direction Example</title>

</head>

<body>

<p>This text will go left to right.</p>

<p><bdo dir="rtl">This text will go right to left.</bdo></p>

</body>

</html>

This will produce following result:

This text will go left to right.

This text will go right to left.

## Special Terms

The **<dfn>...</dfn>** element (or HTML Definition Element) allows you to specify that you are introducing a special term. It's usage is similar to italic words in the midst of a paragraph.

Typically, you would use the <dfn> element the first time you introduce a key term. Most recent browsers render the content of a <dfn> element in an italic font.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Special Terms Example</title>

</head>

<body>

<p>The following word is a <dfn>special</dfn> term.</p>

</body>

</html>

This will produce following result:

The following word is a special term.

## Quoting Text

When you want to quote a passage from another source, you should put it in between**<blockquote>...</blockquote>** tags.

Text inside a <blockquote> element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Blockquote Example</title>

</head>

<body>

<p>The following description of XHTML is taken from the W3C Web site:</p>

<blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>

</body>

</html>

This will produce following result:

The following description of XHTML is taken from the W3C Web site:

XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.

## Short Quotations

The **<q>...</q>** element is used when you want to add a double quote within a sentence.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Double Quote Example</title>

</head>

<body>

<p>Amit is in Spain, <q>I think I am wrong</q>.</p>

</body>

</html>

This will produce following result:

Amit is in Spain, I think I am wrong.

## Text Citations

If you are quoting a text, you can indicate the source placing it between an opening **<cite>** tag and closing **</cite>** tag

As you would expect in a print publication, the content of the <cite> element is rendered in italicized text by default.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Citations Example</title>

</head>

<body>

<p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>

</body>

</html>

This will produce following result:

This HTML tutorial is derived from W3 Standard for HTML.

## Computer Code

Any programming code to appear on a Web page should be placed inside **<code>...</code>**tags. Usually the content of the <code> element is presented in a monospaced font, just like the code in most programming books.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Computer Code Example</title>

</head>

<body>

<p>Regular text. <code>This is code.</code> Regular text.</p>

</body>

</html>

This will produce following result:

Regular text. This is code. Regular text.

## Keyboard Text

When you are talking about computers, if you want to tell a reader to enter some text, you can use the **<kbd>...</kbd>** element to indicate what should be typed in, as in this example.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Keyboard Text Example</title>

</head>

<body>

<p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>

</body>

</html>

This will produce following result:

Regular text. This is inside kbd element Regular text.

## Programming Variables

This element is usually used in conjunction with the **<pre>** and **<code>** elements to indicate that the content of that element is a variable.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Variable Text Example</title>

</head>

<body>

<p><code>document.write("<var>user-name</var>")</code></p>

</body>

</html>

This will produce following result:

document.write("user-name")

## Program Output

The **<samp>...</samp>** element indicates sample output from a program, and script etc. Again, it is mainly used when documenting programming or coding concepts.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Program Output Example</title>

</head>

<body>

<p>Result produced by the program is <samp>Hello World!</samp></p>

</body>

</html>

This will produce following result:

Result produced by the program is Hello World!

## Address Text

The **<address>...</address>** element is used to contain any address.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Address Example</title>

</head>

<body>

<address>388A, Road No 22, Jubilee Hills - Hyderabad</address>

</body>

</html>

This will produce following result:

388A, Road No 22, Jubilee Hills - Hyderabad

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

## Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by **<head>** and **</head>** tags. A meta tag can have following attributes in addition to core attributes:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Name | Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc. |
| content | Specifies the property's value. |
| scheme | Specifies a scheme to interpret the property's value (as declared in the content attribute). |
| http-equiv | Used for http response message headers. For example http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie. |

## Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

### Example

Following is an example where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

This will produce following result:

Hello HTML5!

## Document Description

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

## Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="revised" content="Tutorialspoint, 3/7/2014" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

## Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

### Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="revised" content="Tutorialspoint, 3/7/2014" />

<meta http-equiv="refresh" content="5" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

## Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

### Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="revised" content="Tutorialspoint, 3/7/2014" />

<meta http-equiv="refresh" content="5; url=http://www.tutorialspoint.com" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

## Setting Cookies

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various infromation based on your web application need.

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

### Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="revised" content="Tutorialspoint, 3/7/2014" />

<meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15 23:59:59 GMT;" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

If you do not include the expiration date and time, the cookie is considered a session cookie and will be deleted when the user exits the browser.

**Note:**You can check [**PHP and Cookies**](http://www.tutorialspoint.com/php/php_cookies.htm) tutorial for a complete detail on Cookies.

## Setting Author Name

You can set an author name in a web page using meta tag. See an example below:

### Example

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="author" content="Mahnaz Mohtashim" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

## Specify Character Set

You can use <meta> tag to specify character set used within the webpage.

### Example

By default, Web servers and Web browsers use ISO-8859-1 (Latin1) encoding to process Web pages. Following is an example to set UTF-8 encoding:

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="author" content="Mahnaz Mohtashim" />

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

To serve the static page with traditional Chinese characters, the webpage must contain a <meta> tag to set Big5 encoding:

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

<meta name="description" content="Learning about Meta Tags." />

<meta name="author" content="Mahnaz Mohtashim" />

<meta http-equiv="Content-Type" content="text/html; charset=Big5" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code and increases code readability.

HTML comments are placed in between **<!-- ... -->** tags. So any content placed with-in <!-- ... --> tags will be treated as comment and will be completely ignored by the browser.

### Example

<!DOCTYPE html>

<html>

<head> <!-- Document Header Starts -->

<title>This is document title</title>

</head> <!-- Document Header Ends -->

<body>

<p>Document content goes here.....</p>

</body>

</html>

This will produce following result without displaying the content given as a part of comments:

Document content goes here.....

## Valid vs Invalid Comments

Comments do not nest which means a comment can not be put inside another comment. Second the double-dash sequence "--" may not appear inside a comment except as part of the closing --> tag. You must also make sure that there are no spaces in the start-of-comment string.

### Example

Here given comment is a valid comment and will be wiped off by the browser.

<!DOCTYPE html>

<html>

<head>

<title>Valid Comment Example</title>

</head>

<body>

<!-- This is valid comment -->

<p>Document content goes here.....</p>

</body>

</html>

But following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

<!DOCTYPE html>

<html>

<head>

<title>Invalid Comment Example</title>

</head>

<body>

< !-- This is not a valid comment -->

<p>Document content goes here.....</p>

</body>

</html>

This will produce following result:

< !-- This is not a valid comment -->

Document content goes here.....

## Multiline Comments

So far we have seen single line comments, but HTML supports multi-line comments as well.

You can comment multiple lines by the special beginning tag <!-- and ending tag --> placed before the first line and end of the last line as shown in the given example below.

### Example

<!DOCTYPE html><html>

<head>

<title>Multiline Comments</title>

</head>

<body>

<!--

This is a multiline comment and it can

span through as many as lines you like.

-->

<p>Document content goes here.....</p>

</body>

</html>

This will produce following result:

Document content goes here.....

## Conditional Comments

Conditional comments only work in Internet Explorer (IE) on Windows but they are ignored by other browsers. They are supported from Explorer 5 onwards, and you can use them to give conditional instructions to different versions of IE.

### Example

<!DOCTYPE html><html>

<head>

<title>Conditional Comments</title>

<!--[if IE 6]>

Special instructions for IE 6 here

<![endif]-->

</head>

<body>

<p>Document content goes here.....</p>

</body>

</html>

You will come across a situation where you will need to apply a different style sheet based on different versions of Internet Explorer, in such situation conditional comments will be helpful.

## Using Comment Tag

There are few browsers that support <comment> tag to comment a part of HTML code.

### Example

<!DOCTYPE html><html>

<head>

<title>Using Comment Tag</title>

</head>

<body>

<p>This is <comment>not</comment> Internet Explorer.</p>

</body>

</html>

If you are using IE then it will produce following result:

This is Internet Explorer.

But if you are not using IE, then it will produce following result:

This is  Internet Explorer.

## Commenting Script Code

Though you will learn Javascript with HTML, in a separate tutorial, but here you must make a note that if you are using Java Script or VB Script in your HTML code then it is recommended to put that script code inside proper HTML comments so that old browsers can work properly.

### Example

<!DOCTYPE html><html>

<head>

<title>Commenting Script Code</title>

<script>

<!--

document.write("Hello World!")

//-->

</script>

</head>

<body>

<p>Hello , World!</p>

</body>

</html>

This will produce following result:

Hello World!

Hello , World!

## Commenting Style Sheets

Though you will learn using style sheets with HTML in a separate tutorial, but here you must make a note that if you are using Casecading Style Sheet (CSS) in your HTML code then it is recommended to put that style sheet code inside proper HTML comments so that old browsers can work properly.

### Example

<!DOCTYPE html><html>

<head>

<title>Commenting Style Sheets</title>

<style>

<!--

.example {

border:1px solid #4a7d49;

}

//-->

</style>

</head>

<body>

<div class="example">Hello , World!</div>

</body>

</html>

This will produce following result:

Hello , World!

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

## Insert Image

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

<img src="Image URL" ... attributes-list/>

The <img> tag is an empty tag, which means that it can contain only list of attributes and it has no closing tag.

### Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory:

<!DOCTYPE html>

<html>

<head>

<title>Using Image in Webpage</title>

</head>

<body>

<p>Simple Image Insert</p>

<img src="/html/images/test.png" alt="Test Image" />

</body>

</html>

This will produce following result:

Simple Image Insert

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

## Set Image Location

Usually we keep our all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory **images** inside the home directory where we will keep our image test.png.

### Example

Assuming our image location is "/html/image/test.png", try the following example:

<!DOCTYPE html>

<html>

<head>

<title>Using Image in Webpage</title>

</head>

<body>

<p>Simple Image Insert</p>

<img src="/html/images/test.png" alt="Test Image" />

</body>

</html>

This will produce following result:



Simple Image Insert

## Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height**attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Set Image Width and Height</title>

</head>

<body>

<p>Setting image width and height</p>

<img src="/html/images/test.png" alt="Test Image" width="150" height="100"/>

</body>

</html>

This will produce following result:



Setting image width and height

## Set Image Border

By default image will have a border around it, you can specify border thickness in terms of pixels using **border** attribute. A thickness of 0 means, no border around the picture.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Set Image Border</title>

</head>

<body>

<p>Setting image Border</p>

<img src="/html/images/test.png" alt="Test Image" border="3"/>

</body>

</html>

## Set Image Alignment

By default image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

### Example

<!DOCTYPE html>

<html>

<head>

<title>Set Image Alignment</title>

</head>

<body>

<p>Setting image Alignment</p>

<img src="/html/images/test.png" alt="Test Image" border="3" align="right"/>

</body>

</html>

This will produce following result:

Setting image Alignment 

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Tables</title>

</head>

<body>

<table border="1">

<tr>

<td>Row 1, Column 1</td>

<td>Row 1, Column 2</td>

</tr>

<tr>

<td>Row 2, Column 1</td>

<td>Row 2, Column 2</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| Row 1, Column 1 | Row 1, Column 2 |
| Row 2, Column 1 | Row 2, Column 2 |

Here **border** is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border then you can use border="0".

## Table Heading

Table heading can be defined using **<th>** tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Header</title>

</head>

<body>

<table border="1">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| **Name** | **Salary** |
| Ramesh Raman | 5000 |
| Shabbir Hussein | 7000 |

## Cellpadding and Cellspacing Attributes

There are two attribiutes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines the width of the border, while cellpadding represents the distance between cell borders and the content within a cell.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Cellpadding</title>

</head>

<body>

<table border="1" cellpadding="5" cellspacing="5">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| **Name** | **Salary** |
| Ramesh Raman | 5000 |
| Shabbir Hussein | 7000 |

## Colspan and Rowspan Attributes

You will use **colspan** attribute if you want to merge two or more columns into a single column. Similar way you will use **rowspan** if you want to merge two or more rows.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Colspan/Rowspan</title>

</head>

<body>

<table border="1">

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |  |  |
| --- | --- | --- |
| **Column 1** | **Column 2** | **Column 3** |
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
| Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 | | |

## Tables Backgrounds

You can set table background using one of the following two ways:

* **bgcolor** attribute - You can set background color for whole table or just for one cell.
* **background** attribute - You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Background</title>

</head>

<body>

<table border="1" bordercolor="green" bgcolor="yellow">

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>

This will produce following result:

|  |  |  |
| --- | --- | --- |
| **Column 1** | **Column 2** | **Column 3** |
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
| Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 | | |

Here is an example of using **background** attribute. Here we will use an image available in /images directory.

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Background</title>

</head>

<body>

<table border="1" bordercolor="green" background="/images/test.png">

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>

<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>

<tr><td colspan="3">Row 3 Cell 1</td></tr>

</table>

</body>

</html>

This will produce following result. Here background image did not apply to table's header.

|  |  |  |
| --- | --- | --- |
| **Column 1** | **Column 2** | **Column 3** |
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
| Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 | | |

## Table Height and Width

You can set a table width and height using **width** and **height** attrubutes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Width/Height</title>

</head>

<body>

<table border="1" width="400" height="150">

<tr>

<td>Row 1, Column 1</td>

<td>Row 1, Column 2</td>

</tr>

<tr>

<td>Row 2, Column 1</td>

<td>Row 2, Column 2</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| Row 1, Column 1 | Row 1, Column 2 |
| Row 2, Column 1 | Row 2, Column 2 |

## Table Caption

The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table Caption</title>

</head>

<body>

<table border="1" width="100%">

<caption>This is the caption</caption>

<tr>

<td>row 1, column 1</td><td>row 1, columnn 2</td>

</tr>

<tr>

<td>row 2, column 1</td><td>row 2, columnn 2</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |
| --- | --- |
| This is the caption | |
| row 1, column 1 | row 1, columnn 2 |
| row 2, column 1 | row 2, columnn 2 |

## Table Header, Body, and Footer

Tables can be divided into three portions: a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

The three elements for separating the head, body, and foot of a table are:

* **<thead> -**to create a separate table header.
* **<tbody> -**to indicate the main body of the table.
* **<tfoot> -**to create a separate table footer.

A table may contain several <tbody> elements to indicate different *pages* or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Table</title>

</head>

<body>

<table border="1" width="100%">

<thead>

<tr>

<td colspan="4">This is the head of the table</td>

</tr>

</thead>

<tfoot>

<tr>

<td colspan="4">This is the foot of the table</td>

</tr>

</tfoot>

<tbody>

<tr>

<td>Cell 1</td>

<td>Cell 2</td>

<td>Cell 3</td>

<td>Cell 4</td>

</tr>

</tbody>

</table>

</body>

</html>

This will produce following result:

| This is the head of the table | | | |
| --- | --- | --- | --- |
| This is the foot of the table | | | |
| Cell 1 | Cell 2 | Cell 3 | Cell 4 |

## Nested Tables

You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

## Example

Following is the example of using another table and other tags inside a table cell.

<!DOCTYPE html>

<html>

<head>

<title>HTML Table</title>

</head>

<body>

<table border="1" width="100%">

<tr>

<td>

<table border="1" width="100%">

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Ramesh Raman</td>

<td>5000</td>

</tr>

<tr>

<td>Shabbir Hussein</td>

<td>7000</td>

</tr>

</table>

</td>

</tr>

</table>

</body>

</html>

This will produce following result:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Name** | **Salary** | | Ramesh Raman | 5000 | | Shabbir Hussein | 7000 | |

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain:

* **<ul>** - An unordered list. This will list items using plain bullets.
* **<ol>** - An ordered list. This will use different schemes of numbers to list your items.
* **<dl>** - A definition list. This arranges your items in the same way as they are arranged in a dictionary.

## HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **<ul>** tag. Each item in the list is marked with a bullet.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

## The type Attribute

You can use **type** attribute for <ul> tag to specify the type of bullet you like. By default it is a disc. Following are the possible options:

<ul type="square">

<ul type="disc">

<ul type="circle">

### Example

Following is an example where we used <ul type="square">

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul type="square">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

### Example

Following is an example where we used <ul type="disc"> :

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul type="disc">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

### Example

Following is an example where we used <ul type="circle"> :

<!DOCTYPE html>

<html>

<head>

<title>HTML Unordered List</title>

</head>

<body>

<ul type="circle">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</body>

</html>

This will produce following result:

* Beetroot
* Ginger
* Potato
* Radish

## HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted then HTML ordered list will be used. This list is created by using **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

## The type Attribute

You can use **type** attribute for <ol> tag to specify the type of numbering you like. By default it is a number. Following are the possible options:

<ol type="1"> - Default-Case Numerals.

<ol type="I"> - Upper-Case Numerals.

<ol type="i"> - Lower-Case Numerals.

<ol type="a"> - Lower-Case Letters.

<ol type="A"> - Upper-Case Letters.

### Example

Following is an example where we used <ol type="1">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="1">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

### Example

Following is an example where we used <ol type="I">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="I">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

### Example

Following is an example where we used <ol type="i">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="i">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

### Example

Following is an example where we used <ol type="A">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="A">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

### Example

Following is an example where we used <ol type="a">

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="a">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

## The start Attribute

You can use **start** attribute for <ol> tag to specify the starting point of numbering you need. Following are the possible options:

<ol type="1" start="4"> - Numerals starts with 4.

<ol type="I" start="4"> - Numerals starts with IV.

<ol type="i" start="4"> - Numerals starts with iv.

<ol type="a" start="4"> - Letters starts with d.

<ol type="A" start="4"> - Letters starts with D.

### Example

Following is an example where we used <ol type="i" start="4" >

<!DOCTYPE html>

<html>

<head>

<title>HTML Ordered List</title>

</head>

<body>

<ol type="i" start="4">

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ol>

</body>

</html>

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

## HTML Definition Lists

HTML and XHTML support a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

* <dl> - Defines the start of the list
* <dt> - A term
* <dd> - Term definition
* </dl> - Defines the end of the list

### Example

<!DOCTYPE html>

<html>

<head>

<title>HTML Definition List</title>

</head>

<body>

<dl>

<dt><b>HTML</b></dt>

<dd>This stands for Hyper Text Markup Language</dd>

<dt><b>HTTP</b></dt>

<dd>This stands for Hyper Text Transfer Protocol</dd>

</dl>

</body>

</html>

This will produce following result:

**HTML**

This stands for Hyper Text Markup Language

**HTTP**

This stands for Hyper Text Transfer Protocol

HTML Computer Code Elements

[« Previous](http://www.w3schools.com/html/html_quotation_elements.asp)

[Next Chapter »](http://www.w3schools.com/html/html_comments.asp)

Computer Code

var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
}

HTML Computer Code Formatting

Normally, HTML uses **variable** letter size, and variable letter spacing.

This is not wanted when displaying examples of **computer code**.

The **<kbd>**, **<samp>**, and **<code>** elements all support **fixed** letter size and spacing.

HTML Keyboard Formatting

The HTML **<kbd>** element defines **keyboard input**:

Example

<p>To open a file, select:</p>  
  
<p><kbd>File | Open...</kbd></p>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_kbd)

HTML Sample Formatting

The HTML **<samp>** element defines a **computer output**:

Example

<samp>  
demo.example.com login: Apr 12 09:10:17  
Linux 2.6.10-grsec+gg3+e+fhs6b+nfs+gr0501+++p3+c4a+gr2b-reslog-v6.189  
</samp>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_samp)

HTML Code Formatting

The HTML **<code>** element defines **programming code**:

Example

<code>  
var person = { firstName:"John", lastName:"Doe", age:50, eyeColor:"blue" }  
</code>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_code)

The <code> element does **not** preserve extra **whitespace** and **line-breaks**:

 Example

<p>Coding Example:</p>  
  
<code>  
var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
}  
</code>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_codelines)

To fix this, you must wrap the code in a <pre> element:

 Example

<p>Coding Example:</p>  
  
<code>  
<pre>  
var person = {  
    firstName:"John",  
    lastName:"Doe",  
    age:50,  
    eyeColor:"blue"  
}  
</pre>  
</code>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_codepre)

HTML *Variable* Formatting

The HTML **<var>** element defines a **mathematical variable**:

Example

<p>Einstein wrote:</p>  
  
<p><var>E = m c<sup>2</sup></var></p>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_formatting_var)

HTML Computer Code Elements

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<code>](http://www.w3schools.com/tags/tag_code.asp) | Defines programming code |
| [<kbd>](http://www.w3schools.com/tags/tag_kbd.asp) | Defines keyboard input |
| [<samp>](http://www.w3schools.com/tags/tag_samp.asp) | Defines computer output |
| [<var>](http://www.w3schools.com/tags/tag_var.asp) | Defines a mathematical variable |
| [<pre>](http://www.w3schools.com/tags/tag_pre.asp) | Defines preformatted text |

# HTML Styles - CSS

[« Previous](http://www.w3schools.com/html/html_comments.asp)

[Next Chapter »](http://www.w3schools.com/html/html_links.asp)

# CSS = Styles and Colors

Manipulate Text

Colors,  Boxes

## Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_internal)

## Styling HTML with CSS

CSS stands for **C**ascading **S**tyle **S**heets

Styling can be added to HTML elements in 3 ways:

* Inline - using a **style attribute** in HTML elements
* Internal - using a **<style> element** in the HTML <head> section
* External - using one or more **external CSS files**

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

|  |  |
| --- | --- |
| **Note** | You can learn much more about CSS in our [CSS Tutorial](http://www.w3schools.com/css/default.asp). |

## CSS Syntax

CSS styling has the following syntax:

element {property:value; property:value }

The **element** is an HTML element name. The ***property*** is a CSS property. The ***value*** is a CSS value.

Multiple styles are separated with semicolon.

## Inline Styling (Inline CSS)

**Inline styling** is useful for applying a unique style to a single HTML element:

Inline styling uses the **style attribute**.

This inline styling changes the text color of a single heading:

## Example

<h1 style="color:blue">This is a Blue Heading</h1>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_inline)

## Internal Styling (Internal CSS)

An internal style sheet can be used to define a common style for all HTML elements on a page.

**Internal styling** is defined in the **<head>** section of an HTML page, using a **<style>** element:

## Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_internal)

## External Styling (External CSS)

External style sheet are ideal when the style is applied to many pages.

With external style sheets, you can change the look of an entire web site by changing one file.

**External styles** are defined in an external CSS file, and then linked to in the **<head>** section of an 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>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_external)

## CSS Fonts

The CSS **color** property defines the text color to be used for the HTML element.

The CSS **font-family** property defines the font to be used for the HTML element.

The CSS **font-size**property defines the text size to be used for the HTML element.

## 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>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_fonts)

## The CSS Box Model

Every HTML element has a box around it, even if you cannot see it.

The CSS **border** property defines a visible border around an HTML element:

## Example

p {  
    border:1px solid black;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_borders)

The CSS **padding** property defines a padding (space) inside the border:

## Example

p {  
    border:1px solid black;  
    padding:10px;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_padding)

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

## Example

p {  
    border:1px solid black;  
    padding:10px;  
    margin:30px;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_margin)

|  |  |
| --- | --- |
| **Note** | The CSS examples above use px to define sizes in pixels. |

## The id Attribute

All the examples above use CSS to style HTML elements in a general way.

To define a special style for one special element, first add an id attribute to the element:

## Example

<p id="p01">I am different</p>

then define a different style for the (identified) element:

## Example

p#p01 {  
    color:blue;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_id)

## The class Attribute

To define a style for a special type (class) of elements, add a class attribute to the element:

## Example

<p class="error">I am different</p>

Now you can define a different style for all elements with the specified class:

## Example

p.error {  
    color:red;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_css_class)

|  |  |
| --- | --- |
| **Note** | Use **id** to address **single** elements. Use **class** to address **groups** of elements. |

## Deprecated Tags and Attributes in HTML5

In older HTML versions, several tags and attributes were used to style documents.

These tags and attributes are not supported in HTML5!

Avoid using the <font>, <center>, and <strike> elements.

Avoid using the color and bgcolor attributes.

## Chapter Summary

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

HTML Links

[« Previous](http://www.w3schools.com/html/html_css.asp)

[Next Chapter »](http://www.w3schools.com/html/html_images.asp)

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

HTML Links - Hyperlinks

HTML links are hyperlinks.

A hyperlink is a text or an image you can click on, and jump to another document.

HTML Links - Syntax

In HTML, links are defined with the **<a>** tag:

<a href="*url*">*link text*</a>

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_w3schools)

The **href** attribute specifies the destination address (http://www.w3schools.com/html/)

The **link text** is the visible part (Visit our HTML tutorial).

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

|  |  |
| --- | --- |
| **Note** | The link text does not have to be text. It can be an HTML image or any other HTML element. |

|  |  |
| --- | --- |
| **Note** | Without a trailing slash on subfolder addresses, you might generate two requests to the server. Many servers will automatically add a trailing slash to the address, and then create a new request. |

Local Links

The example above used an absolute URL (A full web address).

A local link (link to the same web site) is specified with a relative URL (without http://www....).

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links)

HTML Links - Colors

When you move the mouse over a link, two things will normally happen:

* The mouse arrow will turn into a little hand
* The color of the link element will change

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

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

You can change the default colors, by using styles:

Example

<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>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_colors)

HTML Links - The target Attribute

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

This example will open the linked document in a new browser window or in a new tab:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_target)

|  |  |
| --- | --- |
| **Target Value** | **Description** |
| \_blank | Opens the linked document in a new window or tab |
| \_self | Opens the linked document in the same frame as it was clicked (this is default) |
| \_parent | Opens the linked document in the parent frame |
| \_top | Opens the linked document in the full body of the window |
| *framename* | Opens the linked document in a named frame |

If your webpage is locked in a frame, you can use target="\_top" to break out of the frame:

Example

<a href="http://www.w3schools.com/html/" target="\_top">HTML5 tutorial!</a>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_target_top)

HTML Links - Image as Link

It is common to use images as links:

Example

<a href="default.asp">  
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0">  
</a>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_image)

|  |  |
| --- | --- |
| **Note** | border:0 is added to prevent IE9 (and earlier) from displaying a border around the image. |

HTML Links - Create a Bookmark

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

Bookmarks are practical if your website has long pages.

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

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="tips">Useful Tips Section</h2>

Then, add a link to the bookmark ("Useful Tips Section"), from within the same page:

<a href="#tips">Visit the Useful Tips Section</a>

Or, add a link to the bookmark ("Useful Tips Section"), from another page:

Example

<a href="html\_tips.htm#tips">Visit the Useful Tips Section</a>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_links_bookmark)

Chapter Summary

* Use the HTML **<a>** element to define a link
* Use the HTML **href** attribute to define the link address
* Use the HTML **target** attribute to define where to open the linked document
* Use the HTML **<img>** element (inside <a>) to use an image as a link
* Use the HTML **id** attribute (id="*value*") to define bookmarks in a page
* Use the HTML **href**attribute (href="#*value*") to link to the bookmark

HTML Images

[« Previous](http://www.w3schools.com/html/html_links.asp)

[Next Chapter »](http://www.w3schools.com/html/html_tables.asp)

Example

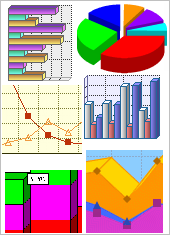
GIF Images



JPG Images



PNG Images



<!DOCTYPE html>  
<html>  
<body>  
  
<h2>Spectacular Mountain</h2>  
<img src="pic\_mountain.jpg" alt="Mountain View" style="width:304px;height:228px;">  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_mountain)

|  |  |
| --- | --- |
| **Note** | Always specify the width and height of an image. If width and height are not specified, the page will flicker while the image loads. |

HTML Images Syntax

In HTML, images are defined with the **<img>** tag.

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

The src attribute specifies the URL (web address) of the image:

<img src="*url*" alt="*some\_text*">

The alt Attribute

The alt attribute specifies an alternate text for an image, if the image cannot be displayed.

The alt attribute provides alternative information for an image if a 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).

If a browser cannot find an image, it will display the alt text:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_wrong)

The alt attribute is required. A web page will not validate correctly without it.

HTML Screen Readers

A screen reader is a software programs that can read what is displayed on a screen.

Screen readers are useful to people who are blind, visually impaired, or learning disabled.

|  |  |
| --- | --- |
| **Note** | Screen readers can read the **alt** attribute. |

Image Size - Width and Height

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

The values are specified in pixels (use px after the value):

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_size)

Alternatively, you can use **width** and **height** attributes. Here, the values are specified in pixels by default:

Example

<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_attributes)

Width and Height or Style?

Both the width, height, and style attributes are valid in the latest HTML5 standard.

We suggest you use the style attribute. It prevents styles sheets from changing the original size of images:

Example

<!DOCTYPE html>  
<html>  
<head>  
<style>  
img {   
    width:100%;   
}  
</style>  
</head>  
<body>  
  
<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">  
<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_style)

Images in Another Folder

If not specified, the browser expects to find the image in the same folder as the web page.

However, it is common to store images in a sub-folder. You must then include the folder name in the src attribute:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_folder)

Images on Another Server

Some web sites store their images on image servers.

Actually, you can access images from any web address in the world:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_w3schools)

Animated Images

The GIF standard allows animated images:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_hackman)

Note that the syntax of inserting animated images is no different from non-animated images.

Using an Image as a Link

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

Example

<a href="default.asp">  
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;border:0;">  
</a>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_link)

|  |  |
| --- | --- |
| **Note** | Add "border:0;" to prevent IE9 (and earlier) from displaying a border around the image. |

Image Floating

Use the CSS float property to let the image float.

The image can 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>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_float)

Image Maps

Use the <map> tag to define an image-map. An image-map is an image with clickable areas.

The name attribute of the <map> tag is associated with the <img>'s usemap attribute and creates a relationship between the image and the map.

The <map> tag contains a number of <area> tags, that defines the clickable areas in the image-map:

Example

<img src="planets.gif" alt="Planets" usemap="#planetmap" style="width:145px;height:126px;">  
  
<map name="planetmap">  
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">  
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">  
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">  
</map>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_images_map)

Chapter Summary

* Use the HTML **<img>** element to define an image
* Use the HTML **src** attribute to define the URL of the image
* Use the HTML **alt** attribute to define an alternate text for an image, if it cannot be displayed
* Use the HTML **width** and **height** attributes to define the size of the image
* Use the CSS **width** and **height** properties to define the size of the image (alternatively)
* Use the CSS **float** property to let the image float
* Use the HTML **<map>** element to define an image-map
* Use the HTML **<area>** element to define the clickable areas in the image-map
* Use the HTML <img>'s element **usemap** attribute to point to an image-map

|  |  |
| --- | --- |
| **Note** | Loading images takes time. Large images can slow down your page. Use images carefully. |

HTML Tables

[« Previous](http://www.w3schools.com/html/html_images.asp)

[Next Chapter »](http://www.w3schools.com/html/html_lists.asp)

HTML Table Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **First Name** | **Last Name** | **Points** |
| 1 | Eve | Jackson | 94 |
| 2 | John | Doe | 80 |
| 3 | Adam | Johnson | 67 |
| 4 | Jill | Smith | 50 |

Defining HTML Tables

Example

<table style="width:100%">  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>   
    <td>50</td>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>   
    <td>94</td>  
  </tr>  
</table>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table)

Example explained:

Tables are defined with the **<table>** tag.

Tables are divided into **table rows** with the **<tr>** tag.

Table rows are divided into **table data** with the **<td>** tag.

A table row can also be divided into **table headings** with the **<th>** tag.

|  |  |
| --- | --- |
| **Note** | Table data <td> are the data containers of the table. They can contain all sorts of HTML elements like text, images, lists, other tables, etc. |

An HTML Table with a Border Attribute

If you do not specify a border for the table, it will be displayed without borders.

A border can be added using the border attribute:

Example

<table border="1" style="width:100%">  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>   
    <td>50</td>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>   
    <td>94</td>  
  </tr>  
</table>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_border_attribute)

|  |  |
| --- | --- |
| **Note** | The border attribute is on its way out of the HTML standard! It is better to use CSS. |

To add borders, use the **CSS border** property:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_border)

Remember to define borders for both the table and the table cells.

An HTML Table with Collapsed Borders

If you want the borders to collapse into one border, add **CSS border-collapse**:

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_collapse)

An HTML Table with Cell Padding

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

If you do not specify a padding, the table cells will be displayed without padding.

To set the padding, use the **CSS padding** property:

Example

table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}  
th, td {  
    padding: 15px;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_cellpadding)

HTML Table Headings

Table headings are defined with the **<th>** tag.

By default, all major browsers display table headings as bold and centered:

Example

<table style="width:100%">  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>   
    <th>Points</th>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>   
    <td>94</td>  
  </tr>  
</table>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_headings)

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

Example

th {  
    text-align: left;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_headings_left)

An HTML Table with Border Spacing

Border spacing specifies the space between the cells.

To set the border spacing for a table, use the **CSS border-spacing** property:

Example

table {  
    border-spacing: 5px;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_cellspacing)

|  |  |
| --- | --- |
| **Note** | If the table has collapsed borders, border-spacing has no effect. |

Table Cells that Span Many Columns

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

Example

<table style="width:100%">  
  <tr>  
    <th>Name</th>  
    <th colspan="2">Telephone</th>  
  </tr>  
  <tr>  
    <td>Bill Gates</td>  
    <td>555 77 854</td>  
    <td>555 77 855</td>  
  </tr>  
</table>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_colspan)

Table Cells that Span Many Rows

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

Example

<table style="width:100%">  
  <tr>  
    <th>Name:</th>  
    <td>Bill Gates</td>  
  </tr>  
  <tr>  
    <th rowspan="2">Telephone:</th>  
    <td>555 77 854</td>  
  </tr>  
  <tr>  
    <td>555 77 855</td>  
  </tr>  
</table>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_rowspan)

An HTML Table With a Caption

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

Example

<table style="width:100%">  
  <caption>Monthly savings</caption>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>$100</td>  
  </tr>  
  <tr>  
    <td>February</td>  
    <td>$50</td>  
  </tr>  
</table>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_tables2)

|  |  |
| --- | --- |
| **Note** | The <caption> tag must be inserted immediately after the <table> tag. |

A Special Style for One Table

To define a special style for a special table, add an **id attribute** to the table:

Example

<table id="t01">  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>   
    <th>Points</th>  
  </tr>  
  <tr>  
    <td>Eve</td>  
    <td>Jackson</td>   
    <td>94</td>  
  </tr>  
</table>

Now you can define a special style for this table:

table#t01 {  
    width: 100%;   
    background-color: #f1f1c1;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_id1)

And add more styles:

table#t01 tr:nth-child(even) {  
    background-color: #eee;  
}  
table#t01 tr:nth-child(odd) {  
    background-color: #fff;  
}  
table#t01 th {  
    color: white;  
    background-color: black;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_id2)

Chapter Summary

* Use the HTML **<table>** element to define a table
* Use the HTML **<tr>** element to define a table row
* Use the HTML **<td>** element to define a table data
* Use the HTML **<th>** element to define a table heading
* Use the HTML **<caption>** element to define a table caption
* Use the CSS **border** property to define a border
* Use the CSS **border-collapse** property to collapse cell borders
* Use the CSS **padding** property to add padding to cells
* Use the CSS **text-align** property to align cell text
* Use the CSS **border-spacing** property to set the spacing between cells
* Use the **colspan** attribute to make a cell span many columns
* Use the **rowspan** attribute to make a cell span many rows
* Use the **id** attribute to uniquely define one table

# HTML Lists

[« Previous](http://www.w3schools.com/html/html_tables.asp)

[Next Chapter »](http://www.w3schools.com/html/html_blocks.asp)

## Unordered lists and ordered lists are commonly used in HTML:

### Unordered List

* The first item
* The second item
* The third item
* The fourth item

### Ordered List

1. The first item
2. The second item
3. The third item
4. The fourth item

## Unordered HTML Lists

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):

## Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_unordered)

## Unordered HTML Lists - The Style Attribute

A **style** attribute can be added to an **unordered list**, to define the style of the marker:

|  |  |
| --- | --- |
| **Style** | **Description** |
| list-style-type:disc | The list items will be marked with bullets (default) |
| list-style-type:circle | The list items will be marked with circles |
| list-style-type:square | The list items will be marked with squares |
| list-style-type:none | The list items will not be marked |

## Disc:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_unordered_disc)

## Circle:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_unordered_circle)

## Square:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_unordered_square)

## None:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_unordered_none)

## Ordered HTML Lists

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

The list items will be marked with numbers:

## Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_ordered)

## Ordered HTML Lists - The Type Attribute

A **type** attribute can be added to an **ordered list**, to define the type of the marker:

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

## Numbers:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_ordered_numbers)

## Uppercase Letters:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_ordered_ucase)

## Lowercase Letters:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_ordered_lcase)

## Uppercase Roman Numbers:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_ordered_roman_ucase)

## Lowercase Roman Numbers:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_ordered_roman_lcase)

## HTML Description Lists

HTML also supports description lists.

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

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

## Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_description)

## Nested HTML Lists

List can be nested (lists inside lists):

## Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_nested)

|  |  |
| --- | --- |
| **Note** | List items can contain new list, and other HTML elements, like images and links, etc. |

## Horizontal Lists

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

One popular way, is to style a list to be displayed horizontally:

## Example

<!DOCTYPE html>  
<html>  
  
<head>  
<style>  
ul#menu li {  
    display:inline;  
}  
</style>  
</head>  
  
<body>  
  
<h2>Horizontal List</h2>  
  
<ul id="menu">  
  <li>HTML</li>  
  <li>CSS</li>  
  <li>JavaScript</li>  
  <li>PHP</li>  
</ul>   
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_horizontal)

With a little extra style, you can make it look like a menu:

## Example

ul#menu {  
    padding: 0;  
}  
  
ul#menu li {  
    display: inline;  
}  
  
ul#menu li a {  
    background-color: black;  
    color: white;  
    padding: 10px 20px;  
    text-decoration: none;  
    border-radius: 4px 4px 0 0;  
}  
  
ul#menu li a:hover {  
    background-color: orange;  
}

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_menu)

## Chapter Summary

* Use the HTML **<ul>** element to define an unordered list
* Use the HTML **style** attribute to define the bullet style
* Use the HTML **<ol>** element to define an ordered list
* Use the HTML **type** attribute to define the numbering type
* Use the HTML **<li>** element to define a list item
* Use the HTML **<dl>** element to define a description list
* Use the HTML **<dt>** element to define the description term
* Use the HTML **<dd>** element to define the description data
* Lists can be nested inside lists
* List items can contain other HTML elements
* Use the CSS property **display:inline** to display a list horizontally

## The <form> Element

HTML forms are used to collect user input.

The **<form>** element defines an HTML form:

<form>  
.  
*form elements*  
.  
</form>

HTML forms contain **form elements**.

Form elements are different types of input elements, checkboxes, radio buttons, submit buttons, and more.

## The <input> Element

The **<input>** element is the most important **form element**.

The <input> element has many variations, depending on the **type** attribute.

Here are the types used in this chapter:

|  |  |
| --- | --- |
| **Type** | **Description** |
| Text | Defines normal text input |
| Radio | Defines radio button input (for selecting one of many choices) |
| Submit | Defines a submit button (for submitting the form) |

## Text Input

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

### Example

<form>  
  First name:<br>  
  <input type="text" name="firstname">  
  <br>  
  Last name:<br>  
  <input type="text" name="lastname">  
</form>

This is how it will look like in a browser:

First name:  
  
Last name:  


**Note:** The form itself is not visible. Also note that the default width of a text field is 20 characters.

## Radio Button Input

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

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

### Example

<form>  
  <input type="radio" name="sex" value="male" checked>Male  
  <br>  
  <input type="radio" name="sex" value="female">Female  
</form>

This is how the HTML code above will be displayed in a browser:

 Male   
 Female

## The Submit Button

**<input type="submit">** defines a button for **submitting** a form 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">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey">  
  <br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse">  
  <br><br>  
  <input type="submit" value="Submit">  
</form>

O/P

Top of Form

First name:  
   
Last name:  
   
  


Bottom of Form

## The Action Attribute

The **action attribute** defines the action to be performed when the form is submitted.

The common way to submit a form to a server, is by using a submit button.

Normally, the form is submitted to a web page on a web server.

In the example above, a server-side script is specified to handle the submitted form:

<form **action="action\_page.php**">

If the action attribute is omitted, the action is set to the current page.

## The Method Attribute

The **method attribute** specifies the HTTP method (**GET**or **POST**) to be used when submitting the forms:

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

or:

<form action="action\_page.php" **method="post"**>

## When to Use GET?

You can use GET (the default method):

If the form submission is passive (like a search engine query), and without sensitive information.

When you use GET, the form data will be visible in the page address:

action\_page.php?firstname=Mickey&lastname=Mouse

|  |  |
| --- | --- |
| **Note** | GET is best suited to short amounts of data. Size limitations are set in your browser. |

## When to Use POST?

You should use POST:

If the form is updating data, or includes sensitive information (password).

POST offers better security because the submitted data is not visible in the page address.

## The Name Attribute

To be submitted correctly, each input field must have a name attribute.

This example will only submit the "Last name" input field:

### Example

<form action="action\_page.php">  
  First name:<br>  
  <input type="text" value="Mickey">  
  <br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse">  
  <br><br>  
  <input type="submit" value="Submit">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_submit_id)

## Grouping Form Data with <fieldset>

The **<fieldset>** element groups related data in a form.

The **<legend>** element defines a caption for the <fieldset> element.

### Example

<form action="action\_page.php">  
  <fieldset>  
    <legend>Personal information:</legend>  
    First name:<br>  
    <input type="text" name="firstname" value="Mickey">  
    <br>  
    Last name:<br>  
    <input type="text" name="lastname" value="Mouse">  
    <br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_legend)

This is how the HTML code above will be displayed in a browser:

Top of Form

Personal information:First name:  
   
Last name:  
   
  


Bottom of Form

## HTML Form Attributes

An HTML <form> element, with all possible attributes set, will look like this:

<form action="action\_page.php" method="GET" target="\_blank" accept-charset="UTF-8"  
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>  
.  
*form elements*  
.  
</form>

Here is the list of <form> attributes:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| accept-charset | Specifies the charset used in the submitted form (default: the page charset). |
| Action | Specifies an address (url) where to submit the form (default: the submitting page). |
| autocomplete | Specifies if the browser should autocomplete the form (default: on). |
| Enctype | Specifies the encoding of the submitted data (default: is url-encoded). |
| method | Specifies the HTTP method used when submitting the form (default: GET). |
| name | Specifies a name used to identify the form (for DOM usage: document.forms.name). |
| novalidate | Specifies that the browser should not validate the form. |
| target | Specifies the target of the address in the action attribute (default: \_self). |

## The <input> Element

The most important form element is the **<input>** element.

The <input> element can vary in many ways, depending on the **type** attribute.

## The <select> Element (Drop-Down List)

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

### Example

<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_select)

The **<option>** elements defines the options to select.

The list will normally show the first item as selected.

You can add a selected attribute to define a predefined option.

### Example

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

## The <textarea> Element

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

### Example

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

This is how the HTML code above will be displayed in a browser:



## The <button> Element

The **<button>** element defines a clickable **button**:

### Example

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

This is how the HTML code above will be displayed in a browser:

Click Me!

## HTML5 Form Elements

HTML5 added the following form elements:

* <datalist>
* <keygen>
* <output>

|  |  |
| --- | --- |
|  | By default, browsers do not display unknown elements. New elements will not destroy your page. |

## HTML5 <datalist> Element

The **<datalist>** element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of pre-defined options as they input data.

The **list** attribute of the <input> element, must refer to the **id** attribute of the <datalist> element.

OperaSafariChromeFirefoxInternet Explorer

### Example

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

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_datalist)

## HTML5 <keygen> Element

The purpose of the **<keygen>** element is to provide a secure way to authenticate users.

The <keygen> element specifies a key-pair generator field in a form.

When the form is submitted, two keys are generated, one private and one public.

The private key is stored locally, and the public key is sent to the server.

The public key could be used to generate a client certificate to authenticate the user in the future.

OperaSafariChromeFirefoxInternet Explorer

### Example

A form with a keygen field:

<form action="action\_page.php">  
  Username: <input type="text" name="user">  
  Encryption: <keygen name="security">  
  <input type="submit">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_elem_keygen)

## HTML5 <output> Element

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

OperaSafariChromeFirefoxInternet Explorer

### Example

Perform a calculation and show the result in an <output> element:

<form action="action\_page.asp"  
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">  
  0  
  <input type="range"  id="a" name="a" value="50">  
  100 +  
  <input type="number" id="b" name="b" value="50">  
  =  
  <output name="x" for="a b"></output>  
  <br><br>  
  <input type="submit">  
</form>

## HTML Form Elements

= new in HTML5.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<form>](http://www.w3schools.com/tags/tag_form.asp) | Defines an HTML form for user input |
| [<input>](http://www.w3schools.com/tags/tag_input.asp) | Defines an input control |
| [<textarea>](http://www.w3schools.com/tags/tag_textarea.asp) | Defines a multiline input control (text area) |
| [<label>](http://www.w3schools.com/tags/tag_label.asp) | Defines a label for an <input> element |
| [<fieldset>](http://www.w3schools.com/tags/tag_fieldset.asp) | Groups related elements in a form |
| [<legend>](http://www.w3schools.com/tags/tag_legend.asp) | Defines a caption for a <fieldset> element |
| [<select>](http://www.w3schools.com/tags/tag_select.asp) | Defines a drop-down list |
| [<optgroup>](http://www.w3schools.com/tags/tag_optgroup.asp) | Defines a group of related options in a drop-down list |
| [<option>](http://www.w3schools.com/tags/tag_option.asp) | Defines an option in a drop-down list |
| [<button>](http://www.w3schools.com/tags/tag_button.asp) | Defines a clickable button |
| [<datalist>](http://www.w3schools.com/tags/tag_datalist.asp) | Specifies a list of pre-defined options for input controls |
| [<keygen>](http://www.w3schools.com/tags/tag_keygen.asp) | Defines a key-pair generator field (for forms) |
| [<output>](http://www.w3schools.com/tags/tag_output.asp) | Defines the result of a calculation |

## Input Types

This chapter describes the input types of the <input> element.

## Input Type: text

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

### Example

<form>  
First name:<br>  
<input type="text" name="firstname">  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</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>  
User name:<br>  
<input type="text" name="username">  
<br>  
User password:<br>  
<input type="password" name="psw">  
</form>

This is how the HTML code above will be displayed in a browser:

User name:  
  
User password:  


|  |  |
| --- | --- |
| **Note** | 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 input 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">  
First name:<br>  
<input type="text" name="firstname" value="Mickey">  
<br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse">  
<br><br>  
<input type="submit" value="Submit">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_submit)

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">  
First name:<br>  
<input type="text" name="firstname" value="Mickey">  
<br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse">  
<br><br>  
<input type="submit">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_submit_nn)

## 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

<form>  
<input type="radio" name="sex" value="male" checked> Male  
<br>  
<input type="radio" name="sex" value="female"> Female  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_radio)

This is how the HTML code above will be displayed in a browser:

 Male   
 Female

## 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" name="vehicle1" value="Bike"> I have a bike  
<br>  
<input type="checkbox" name="vehicle2" value="Car"> I have a car   
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_checkbox)

This is how the HTML code above will be displayed in a browser:

 I have a bike   
 I have a car

## Input Type: button

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

### Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_button)

This is how the HTML code above will be displayed in a browser:

 Click Me !

## HTML5 Input Types

HTML5 added several new input types:

* color
* date
* datetime
* datetime-local
* email
* month
* number
* range
* search
* tel
* time
* url
* week

|  |  |
| --- | --- |
| **Note** | Input types, not supported by old web browsers, will behave as input type text. |

## Input Type: number

The **<input type="number">** is used for input fields that should contain a numeric value.

You can set restrictions on the numbers.

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

## Input Restrictions

Here is a list of some common input restrictions (some are new in HTML5):

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Quantity:  
  <input type="number" name="points" min="0" max="100" step="10" value="30">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_number_step)

## 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Birthday:  
  <input type="date" name="bday">  
</form>

<form>  
  Enter a date before 1980-01-01:  
  <input type="date" name="bday" max="1979-12-31"><br>  
  Enter a date after 2000-01-01:  
  <input type="date" name="bday" min="2000-01-02"><br>  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_date_max_min)

## 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_color)

## Input Type: range

The **<input type="range">** is used for input fields that should contain a value within a range.

Depending on browser support, the input field can be displayed as a slider control.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  <input type="range" name="points" min="0" max="10">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_range)

You can use the following attributes to specify restrictions: min, max, step, value.

## 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_month)

## 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Select a week:  
  <input type="week" name="week\_year">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_week)

## 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Select a time:  
  <input type="time" name="usr\_time">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_time)

## Input Type: datetime

The **<input type="datetime">** allows the user to select a date and time (with time zone).

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Birthday (date and time):  
  <input type="datetime" name="bdaytime">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_datetime)

|  |  |
| --- | --- |
| **Note** | The input type datetime is removed from the HTML standard. Use datetime-local instead. |

## Input Type: datetime-local

The **<input type="datetime-local">** allows the user to select a date and time (no time zone).

Depending on browser support, a date picker can show up in the input field.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Birthday (date and time):  
  <input type="datetime-local" name="bdaytime">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_datetime-local)

## 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 adds ".com" to the keyboard to match email input.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  E-mail:  
  <input type="email" name="email">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_email)

## Input Type: search

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

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Search Google:  
  <input type="search" name="googlesearch">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_search)

## Input Type: tel

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

The tel type is currently supported only in Safari 8.

OperaSafariChromeFirefoxInternet Explorer

### Example

<form>  
  Telephone:  
  <input type="tel" name="usrtel">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_tel)

## 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

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

## The value Attribute

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

### Example

<form action="">  
First name:<br>  
<input type="text" name="firstname" value="John">  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_value)

## The readonly Attribute

The **readonly** attribute specifies that the input field is read only (cannot be changed):

### Example

<form action="">  
First name:<br>  
<input type="text" name="firstname" value="John" readonly>  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_readonly)

The readonly attribute does not need a value. It is the same as writing readonly="readonly".

## The disabled Attribute

The **disabled** attribute specifies that the input field is disabled.

A disabled element is un-usable and un-clickable.

Disabled elements will not be submitted.

### Example

<form action="">  
First name:<br>  
<input type="text" name="firstname" value="John" disabled>  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_disabled)

The disabled attribute does not need a value. It is the same as writing disabled="disabled".

## The size Attribute

The **size** attribute specifies the size (in characters) for the input field:

### Example

<form action="">  
First name:<br>  
<input type="text" name="firstname" value="John" size="40">  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_size)

## The maxlength Attribute

The **maxlength** attribute specifies the maximum allowed length for the input field:

### Example

<form action="">  
First name:<br>  
<input type="text" name="firstname" maxlength="10">  
<br>  
Last name:<br>  
<input type="text" name="lastname">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_input_attributes_maxlength)

With a maxlength attribute, the input control will not accept more than the allowed number of characters.

The attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code.

|  |  |
| --- | --- |
| **Note** | Input restrictions are not foolproof. JavaScript provides many ways to add illegal input. To safely restrict input, restrictions must be checked by the receiver (the server) as well. |

## HTML5 Attributes

HTML5 added the following attributes for <input>:

* autocomplete
* autofocus
* form
* formaction
* formenctype
* formmethod
* formnovalidate
* formtarget
* height and width
* list
* min and max
* multiple
* pattern (regexp)
* placeholder
* required
* step

and the following attributes for <form>:

* autocomplete
* novalidate

## The autocomplete Attribute

The autocomplete attribute specifies whether a form or input field should have autocomplete on or off.

When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

**Tip:** It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.

The autocomplete attribute works with <form> and the following <input> types: text, search, url, tel, email, password, datepickers, range, and color.

OperaSafariChromeFirefoxInternet Explorer

### Example

An HTML form with autocomplete on (and off for one input field):

<form action="action\_page.php" autocomplete="on">  
  First name:<input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  E-mail: <input type="email" name="email" autocomplete="off"><br>  
  <input type="submit">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_autocomplete)

**Tip:** In some browsers you may need to activate the autocomplete function for this to work.

## The novalidate Attribute

The novalidate attribute is a <form> attribute.

When present, novalidate specifies that form data should not be validated when submitted.

OperaSafariChromeFirefoxInternet Explorer

### Example

Indicates that the form is not to be validated on submit:

<form action="action\_page.php" novalidate>  
  E-mail: <input type="email" name="user\_email">  
  <input type="submit">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_form_novalidate)

## The autofocus Attribute

The autofocus attribute is a boolean attribute.

When present, it specifies that an <input> element should automatically get focus when the page loads.

OperaSafariChromeFirefoxInternet Explorer

### Example

Let the "First name" input field automatically get focus when the page loads:

First name:<input type="text" name="fname" autofocus>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_autofocus)

## The form Attribute

The form attribute specifies one or more forms an <input> element belongs to.

**Tip:** To refer to more than one form, use a space-separated list of form ids.

OperaSafariChromeFirefoxInternet Explorer

### Example

An input field located outside the HTML form (but still a part of the form):

<form action="action\_page.php" id="form1">  
  First name: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
</form>  
  
Last name: <input type="text" name="lname" form="form1">

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_form)

## The formaction Attribute

The formaction attribute specifies the URL of a file that will process the input control when the form is submitted.

The formaction attribute overrides the action attribute of the <form> element.

The formaction attribute is used with type="submit" and type="image".

OperaSafariChromeFirefoxInternet Explorer

### Example

An HTML form with two submit buttons, with different actions:

<form action="action\_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formaction="demo\_admin.asp"  
  value="Submit as admin">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_formaction)

## The formenctype Attribute

The formenctype attribute specifies how the form-data should be encoded when submitting it to the server (only for forms with method="post").

The formenctype attribute overrides the enctype attribute of the <form> element.

The formenctype attribute is used with type="submit" and type="image".

OperaSafariChromeFirefoxInternet Explorer

### Example

Send form-data that is default encoded (the first submit button), and encoded as "multipart/form-data" (the second submit button):

<form action="demo\_post\_enctype.asp" method="post">  
  First name: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formenctype="multipart/form-data"  
  value="Submit as Multipart/form-data">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_formenctype)

## The formmethod Attribute

The formmethod attribute defines the HTTP method for sending form-data to the action URL.

The formmethod attribute overrides the method attribute of the <form> element.

The formmethod attribute can be used with type="submit" and type="image".

OperaSafariChromeFirefoxInternet Explorer

### Example

The second submit button overrides the HTTP method of the form:

<form action="action\_page.php" method="get">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formmethod="post" formaction="demo\_post.asp"  
  value="Submit using POST">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_formmethod)

## The formnovalidate Attribute

The novalidate attribute is a boolean attribute.

When present, it specifies that the <input> element should not be validated when submitted.

The formnovalidate attribute overrides the novalidate attribute of the <form> element.

The formnovalidate attribute can be used with type="submit".

OperaSafariChromeFirefoxInternet Explorer

### Example

A form with two submit buttons (with and without validation):

<form action="action\_page.php">  
  E-mail: <input type="email" name="userid"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formnovalidate value="Submit without validation">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_formnovalidate)

## The formtarget Attribute

The formtarget attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

The formtarget attribute overrides the target attribute of the <form> element.

The formtarget attribute can be used with type="submit" and type="image".

OperaSafariChromeFirefoxInternet Explorer

### Example

A form with two submit buttons, with different target windows:

<form action="action\_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit as normal">  
  <input type="submit" formtarget="\_blank"  
  value="Submit to a new window">  
</form>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_formtarget)

## The height and width Attributes

The height and width attributes specify the height and width of an <input> element.

The height and width attributes are only used with <input type="image">.

|  |  |
| --- | --- |
| **Note** | Always specify the size of images. If the browser does not know the size, the page will flicker while images load. |

OperaSafariChromeFirefoxInternet Explorer

### Example

Define an image as the submit button, with height and width attributes:

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_height_width)

## The list Attribute

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

OperaSafariChromeFirefoxInternet Explorer

### Example

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

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_datalist)

## The min and max Attributes

The min and max attributes specify the minimum and maximum value for an <input> element.

The min and max attributes work with the following input types: number, range, date, datetime, datetime-local, month, time and week.

OperaSafariChromeFirefoxInternet Explorer

### Example

<input> elements with min and max values:

Enter a date before 1980-01-01:  
<input type="date" name="bday" max="1979-12-31">  
  
Enter a date after 2000-01-01:  
<input type="date" name="bday" min="2000-01-02">  
  
Quantity (between 1 and 5):  
<input type="number" name="quantity" min="1" max="5">

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_max_min)

## The multiple Attribute

The multiple attribute is a boolean attribute.

When present, it specifies that the user is allowed to enter more than one value in the <input> element.

The multiple attribute works with the following input types: email, and file.

OperaSafariChromeFirefoxInternet Explorer

### Example

A file upload field that accepts multiple values:

Select images: <input type="file" name="img" multiple>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_multiple)

## The pattern Attribute

The pattern attribute specifies a regular expression that the <input> element's value is checked against.

The pattern attribute works with the following input types: text, search, url, tel, email, and password.

**Tip:** Use the global [title](http://www.w3schools.com/tags/att_global_title.asp) attribute to describe the pattern to help the user.

**Tip:** Learn more about [regular expressions](http://www.w3schools.com/js/js_regexp.asp) in our JavaScript tutorial.

OperaSafariChromeFirefoxInternet Explorer

### Example

An input field that can contain only three letters (no numbers or special characters):

Country code: <input type="text" name="country\_code" pattern="[A-Za-z]{3}" title="Three letter country code">

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_pattern)

## The placeholder Attribute

The placeholder attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).

The 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.

OperaSafariChromeFirefoxInternet Explorer

### Example

An input field with a placeholder text:

<input type="text" name="fname" placeholder="First name">

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_placeholder)

## The required Attribute

The required attribute is a boolean attribute.

When present, it specifies that an input field must be filled out before submitting the form.

The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

OperaSafariChromeFirefoxInternet Explorer

### Example

A required input field:

Username: <input type="text" name="usrname" required>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml5_input_required)

## The step Attribute

The step attribute specifies the legal number intervals for an <input> element.

Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

**Tip:** The step 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, datetime-local, month, time and week.

OperaSafariChromeFirefoxInternet Explorer

### Example

An input field with a specified legal number intervals:

<input type="number" name="points" step="3">

HTML Block and Inline Elements

[« Previous](http://www.w3schools.com/html/html_lists.asp)

[Next Chapter »](http://www.w3schools.com/html/html_classes.asp)

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

Block-level Elements

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

The <div> element is a block-level element.

Examples of block-level elements:

* <div>
* <h1> - <h6>
* <p>
* <form>

Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

Examples of inline elements:

* <span>
* <a>
* <img>

The <div> Element

The <div> element is a **block-level element** that is often used as a container for other HTML elements.

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

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

Example

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

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_div_capitals)

The <span> Element

The <span> element is an **inline element** that is often used as a container for some text.

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

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

Example

<h1>My <span style="color:red">Important</span> Heading</h1>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_span_red)

HTML Grouping Tags

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<div>](http://www.w3schools.com/tags/tag_div.asp) | Defines a section in a document (block-level) |
| [<span>](http://www.w3schools.com/tags/tag_span.asp) | Defines a section in a document (inline) |

# HTML Classes

[« Previous](http://www.w3schools.com/html/html_blocks.asp)

[Next Chapter »](http://www.w3schools.com/html/html_layout.asp)

## Classing Block Elements

The HTML class attribute makes it possible to define equal styles for "equal" <div> elements:

## London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

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

## Paris

Paris is the capital and most populous city of France.

Situated on the Seine River, it is at the heart of the Île-de-France region, also known as the région parisienne.

Within its metropolitan area is one of the largest population centers in Europe, with over 12 million inhabitants.

## Tokyo

Tokyo is the capital of Japan, the center of the Greater Tokyo Area, and the most populous metropolitan area in the world.

It is the seat of the Japanese government and the Imperial Palace, and the home of the Japanese Imperial Family.

The Tokyo prefecture is part of the world's most populous metropolitan area with 38 million people and the world's largest urban economy.

## Example

<!DOCTYPE html>  
<html>  
<head>  
<style>  
div.cities {  
    background-color:black;  
    color:white;  
    margin:20px;  
    padding:20px;  
}   
</style>  
</head>  
<body>  
  
<div class="cities">  
<h2>London</h2>  
<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>  
</div>  
  
<div class="cities">  
<h2>Paris</h2>  
<p>Paris is the capital and most populous city of France.</p>  
</div>  
  
<div class="cities">  
<h2>Tokyo</h2>  
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,  
and the most populous metropolitan area in the world.</p>  
</div>  
  
</body>  
</html>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_classes_capitals)

## Classing Inline Elements

The HTML class attribute also makes it possible to define equal styles for "equal" <span> elements:

## Example

<!DOCTYPE html>  
<html>  
<head>  
<style>  
span.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>

[Try it Yourself »](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_classes_span)