HTML

HTML stands for **H**yper**t**ext **M**arkup **L**anguage 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.

# Basic 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>

## HTML Tags

|  |  |
| --- | --- |
| **Sr.No** | **Tag & Description** |
| 1 | **<!DOCTYPE...>**This tag defines the document type and HTML version. |
| 2 | **<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. |
| 3 | **<head>**This tag represents the document's header which can keep other HTML tags like <title>, <link> etc. |
| 4 | **<title>**The <title> tag is used inside the <head> tag to mention the document title. |
| 5 | **<body>**This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc. |
| 6 | **<h1>**This tag represents the heading. |
| 7 | **<p>**This tag represents a paragraph. |

# 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>

HTML - Basic 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.

<!DOCTYPE html>

<html>

<head>

<title>Heading </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>

# 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 −

<!DOCTYPE html>

<html>

<head>

<title>Paragraph </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>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Line Break </title>

</head>

<body>

<p>Hello<br />

You delivered your assignment ontime.<br />

Thanks<br />

Mahnaz</p>

</body>

</html>

# Centering Content

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

<!DOCTYPE html>

<html>

<head>

<title>Centring Content </title>

</head>

<body>

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

<center>

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

</center>

</body>

</html>

# 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.

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 horizontal line, while if you miss the forward slash character and just use **<hr>** it is not valid in XHTML

<!DOCTYPE html>

<html>

<head>

<title>Horizontal Line </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>

# Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these 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.

<!DOCTYPE html>

<html>

<head>

<title>Preserve Formatting </title>

</head>

<body>

<pre>

function testFunction( strText ){

alert (strText)

}

</pre>

</body>

</html>

# 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

<!DOCTYPE html>

<html>

<head>

<title>Nonbreaking Spaces </title>

</head>

<body>

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

</body>

</html>

HTML - Elements

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 consists 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 - Attributes

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.

<!DOCTYPE html>

<html>

<head>

<title>Align Attribute </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>

# 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.

<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.

<!DOCTYPE html>

<html>

<head>

<title>The title Attribute </title>

</head>

<body>

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

</body>

</html>

## The class Attribute

The **class** attribute is used to associate an element with a style sheet, and specifies the class of element.

class = "className1 className2 className3"

**The style Attribute**

The style attribute allows you to specify Cascading 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>

# 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 about 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) |

<!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>

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**](https://www.tutorialspoint.com/html/language_iso_codes.htm) for a complete list of language codes.

<!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. |

HTML - Formatting

# Bold Text

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

<!DOCTYPE html>

<html>

<head>

<title>Bold Text </title>

</head>

<body>

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

</body>

</html>

# Italic Text

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

<!DOCTYPE html>

<html>

<head>

<title>Italic Text </title>

</head>

<body>

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

</body>

</html>

# Underlined Text

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

<!DOCTYPE html>

<html>

<head>

<title>Underlined Text </title>

</head>

<body>

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

</body>

</html>

# Strike Text

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

<!DOCTYPE html>

<html>

<head>

<title>Strike Text </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Monospaced Font </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Superscript Text </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Subscript Text </title>

</head>

<body>

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

</body>

</html>

# Inserted Text

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

<!DOCTYPE html>

<html>

<head>

<title>Inserted Text </title>

</head>

<body>

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

</body>

</html>

# Deleted Text

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

<!DOCTYPE html>

<html>

<head>

<title>Deleted Text </title>

</head>

<body>

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

</body>

</html>

# 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 −

<!DOCTYPE html>

<html>

<head>

<title>Larger Text </title>

</head>

<body>

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

</body>

</html>

# 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 −

<!DOCTYPE html>

<html>

<head>

<title>Smaller Text </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Div Tag </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>

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.

<!DOCTYPE html>

<html>

<head>

<title>Span Tag </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>

HTML - Phrase Tags

The phrase tags have been desicolgned 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.

<!DOCTYPE html>

<html>

<head>

<title>Emphasized Text </title>

</head>

<body>

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

</body>

</html>

# Marked Text

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

<!DOCTYPE html>

<html>

<head>

<title>Marked Text </title>

</head>

<body>

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

</body>

</html>

# Strong Text

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

<!DOCTYPE html>

<html>

<head>

<title>Strong Text </title>

</head>

<body>

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

</body>

</html>

# 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.

<!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>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Acronym </title>

</head>

<body>

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

</body>

</html>

# Text Direction

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

<!DOCTYPE html>

<html>

<head>

<title>Text Direction </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>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Special Terms </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Blockquote </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>

# Short Quotations

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

<!DOCTYPE html>

<html>

<head>

<title>Double Quote </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Citations </title>

</head>

<body>

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

</body>

</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.

<!DOCTYPE html>

<html>

<head>

<title>Computer Code </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Keyboard Text </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Variable Text </title>

</head>

<body>

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

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Program Output </title>

</head>

<body>

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

</body>

</html>

# Address Text

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

<!DOCTYPE html>

<html>

<head>

<title>Address </title>

</head>

<body>

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

</body>

</html>

HTML - Meta Tags

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 −

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **Name** Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc. |
| 2 | **content** Specifies the property's value. |
| 3 | **scheme** Specifies a scheme to interpret the property's value (as declared in the content attribute). |
| 4 | **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

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags </title>

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

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

# 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.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags </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.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags </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.

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

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags </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.

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 </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 information 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.

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 </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>

# Setting Author Name

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

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags </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.

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 </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 </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>

HTML - Comments

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.

<!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>

# Multiline Comments

<!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>

# 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.

<!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>

HTML - Images

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

<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.

<!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>

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 Width/Height

<!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>

# 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.

<!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.

<!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>

HTML - Tables

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. The elements under <td> are regular and left aligned by default

<!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>

Here, the **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. Headings, which are defined in <th> tag are centered and bold by default.

<!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>

# Cellpadding and Cellspacing Attributes

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

<!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>

# 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.

<!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>

# Table Height and Width

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

<!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>

# 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>

<!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>

Nested Tables

<!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>

HTML - Lists

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.

<!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>

# 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">

# 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>.

<!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>

# 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"> - Upper-Case Letters.

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

# 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.

# HTML Definition Lists

HTML and XHTML supports 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

<!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>

HTML - Text Links

Linking Documents

A link is specified using HTML tag <a>. This tag is called **anchor tag** and anything between the opening <a> tag and the closing </a> tag becomes part of the link and a user can click that part to reach to the linked document. Following is the simple syntax to use <a> tag.

<a href = "Document URL" ... attributes-list>Link Text</a>

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink </title>

</head>

<body>

<p>Click following link</p>

<a href = "https://www.tutorialspoint.com" target = "\_self">Tutorials Point</a>

</body>

</html>

# The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are the possible options −

|  |  |
| --- | --- |
| **Sr.No** | **Option & Description** |
| 1 | **\_blank** Opens the linked document in a new window or tab. |
| 2 | **\_self** Opens the linked document in the same frame. |
| 3 | **\_parent** Opens the linked document in the parent frame. |
| 4 | **\_top** Opens the linked document in the full body of the window. |
| 5 | **targetframe** Opens the linked document in a named *targetframe*. |

Try following example to understand basic difference in few options given for target attribute.

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink </title>

<base href = "https://www.tutorialspoint.com/">

</head>

<body>

<p>Click any of the following links</p>

<a href = "/html/index.htm" target = "\_blank">Opens in New</a> |

<a href = "/html/index.htm" target = "\_self">Opens in Self</a> |

<a href = "/html/index.htm" target = "\_parent">Opens in Parent</a> |

<a href = "/html/index.htm" target = "\_top">Opens in Body</a>

</body>

</html>

## Use of Base Path

When you link HTML documents related to the same website, it is not required to give a complete URL for every link. You can get rid of it if you use **<base>**tag in your HTML document header. This tag is used to give a base path for all the links. So your browser will concatenate given relative path to this base path and will make a complete URL.

<!DOCTYPE html>

<html>

<head>

<title>Hyperlink </title>

<base href = "https://www.tutorialspoint.com/">

</head>

<body>

<p>Click following link</p>

<a href = "/html/index.htm" target = "\_blank">HTML Tutorial</a>

</body>

</html>

HTML - Image Links

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now, we will learn how to use images to create hyperlinks.

<!DOCTYPE html>

<html>

<head>

<title>Image Hyperlink </title>

</head>

<body>

<p>Click following link</p>

<a href = "https://www.tutorialspoint.com" target = "\_self">

<img src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>

</a>

</body>

</html>

HTML - Email Links

# HTML Email Tag

HTML **<a>** tag provides you option to specify an email address to send an email. While using <a> tag as an email tag, you will use **mailto: email address** along with *href* attribute. Following is the syntax of using **mailto**instead of using http.

<a href = "mailto: abc@example.com">Send Email</a>

This code will generate the following link which you can use to send email.

[Send Email](mailto:abc@example.com)

Now, if a user clicks this link, it launches one Email Client (like Lotus Notes, Outlook Express etc. ) installed on your user's computer. There is another risk to use this option to send email because if user do not have email client installed on their computer then it would not be possible to send email.

## Default Settings

You can specify a default *email subject* and *email body* along with your email address. Following is the example to use default subject and body.

<a href = "mailto:abc@example.com?subject = Feedback&body = Message">

Send Feedback

</a>

This code will generate the following link which you can use to send email.

[Send Feedback](mailto:abc@example.com?subject=Feedback&body=Message)

HTML - Iframes

The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document.

The **src** attribute is used to specify the URL of the document that occupies the inline frame.

<!DOCTYPE html>

<html>

<head>

<title>HTML Iframes</title>

</head>

<body>

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

<iframe src = "/html/menu.htm" width = "555" height = "200">

Sorry your browser does not support inline frames.

</iframe>

<p>Document content also go here...</p>

</body>

</html>

HTML - Blocks

All the HTML elements can be categorized into two categories **(a)** Block Level Elements **(b)** Inline Elements.

# Block Elements

Block elements appear on the screen as if they have a line break before and after them. For example, the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

# Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, <em>, <strong>, <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

# Grouping HTML Elements

There are two important tags which we use very frequently to group various other HTML tags (i) <div> tag and (ii) <span> tag

## The <div> tag

<!DOCTYPE html>

<html>

<head>

<title>HTML div Tag</title>

</head>

<body>

<!-- First group of tags -->

<div style = "color:red">

<h4>This is first group</h4>

<p>Following is a list of vegetables</p>

<ul>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</div>

<!-- Second group of tags -->

<div style = "color:green">

<h4>This is second group</h4>

<p>Following is a list of fruits</p>

<ul>

<li>Apple</li>

<li>Banana</li>

<li>Mango</li>

<li>Strawberry</li>

</ul>

</div>

</body>

</html>

## The <span> tag

The HTML <span> is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the <span> tag and the <div> tag is that the <span> tag is used with inline elements whereas the <div> tag is used with block-level elements.

<!DOCTYPE html>

<html>

<head>

<title>HTML span Tag</title>

</head>

<body>

<p>This is <span style = "color:red">red</span> and this is

<span style = "color:green">green</span></p>

</body>

</html>

HTML - Fonts

The font tag is having three attributes called **size, color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the <font> tag. The text that follows will remain changed until you close with the </font> tag.

# Set Font Size

<!DOCTYPE html>

<html>

<head>

<title>Setting Font Size</title>

</head>

<body>

<font size = "1">Font size = "1"</font><br />

<font size = "2">Font size = "2"</font><br />

<font size = "3">Font size = "3"</font><br />

<font size = "4">Font size = "4"</font><br />

<font size = "5">Font size = "5"</font><br />

<font size = "6">Font size = "6"</font><br />

<font size = "7">Font size = "7"</font>

</body>

</html>

# Relative Font Size

You can specify how many sizes larger or how many sizes smaller than the preset font size should be. You can specify it like **<font size = "+n">** or **<font size = "−n">**

<!DOCTYPE html>

<html>

<head>

<title>Relative Font Size</title>

</head>

<body>

<font size = "-1">Font size = "-1"</font><br />

<font size = "+1">Font size = "+1"</font><br />

<font size = "+2">Font size = "+2"</font><br />

<font size = "+3">Font size = "+3"</font><br />

<font size = "+4">Font size = "+4"</font>

</body>

</html>

# Setting Font Face

<!DOCTYPE html>

<html>

<head>

<title>Font Face</title>

</head>

<body>

<font face = "Times New Roman" size = "5">Times New Roman</font><br />

<font face = "Verdana" size = "5">Verdana</font><br />

<font face = "Comic sans MS" size =" 5">Comic Sans MS</font><br />

<font face = "WildWest" size = "5">WildWest</font><br />

<font face = "Bedrock" size = "5">Bedrock</font><br />

</body>

</html>

# Specify alternate font faces

<font face = "arial,helvetica">

<font face = "Lucida Calligraphy,Comic Sans MS,Lucida Console">

# Setting Font Color

<!DOCTYPE html>

<html>

<head>

<title>Setting Font Color</title>

</head>

<body>

<font color = "#FF00FF">This text is in pink</font><br />

<font color = "red">This text is red</font>

</body>

</html>

# The <basefont> Element

The <basefont> element is supposed to set a default font size, color, and typeface for any parts of the document that are not otherwise contained within a <font> tag. You can use the <font> elements to override the <basefont> settings.

The <basefont> tag also takes color, size and face attributes and it will support relative font setting by giving size a value of +1 for a size larger or −2 for two sizes smaller.

<!DOCTYPE html>

<html>

<head>

<title>Setting Basefont Color</title>

</head>

<body>

<basefont face = "arial, verdana, sans-serif" size = "2" color = "#ff0000">

<p>This is the page's default font.</p>

<h2> of the &lt;basefont&gt; Element</h2>

<p><font size = "+2" color = "darkgray">

This is darkgray text with two sizes larger

</font>

</p>

<p><font face = "courier" size = "-1" color = "#000000">

It is a courier font, a size smaller and black in color.

</font>

</p>

</body>

</html>

HTML - Forms

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax −

<form action = "Script URL" method = "GET|POST">

form elements like input, textarea etc.

</form>

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **action** Backend script ready to process your passed data. |
| 2 | **method** Method to be used to upload data. The most frequently used are GET and POST methods. |
| 3 | **target** Specify the target window or frame where the result of the script will be displayed. It takes values like \_blank, \_self, \_parent etc. |
| 4 | **enctype** You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are −  **application/x-www-form-urlencoded** − This is the standard method most forms use in simple scenarios.  **mutlipart/form-data** − This is used when you want to upload binary data in the form of files like image, word file etc. |

# HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

* Text Input Controls
* Checkboxes Controls
* Radio Box Controls
* Select Box Controls
* File Select boxes
* Hidden Controls
* Clickable Buttons
* Submit and Reset Button

# Text Input Controls

There are three types of text input used on forms −

* **Single-line text input controls** − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.
* **Password input controls** − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
* **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

## Single-line text input controls

<!DOCTYPE html>

<html>

<head>

<title>Text Input Control</title>

</head>

<body>

<form >

First name: <input type = "text" name = "first\_name" />

<br>

Last name: <input type = "text" name = "last\_name" />

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **type** Indicates the type of input control and for text input control it will be set to **text**. |
| 2 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 3 | **value** This can be used to provide an initial value inside the control. |
| 4 | **size** Allows to specify the width of the text-input control in terms of characters. |
| 5 | **maxlength** Allows to specify the maximum number of characters a user can enter into the text box. |

## Password input controls

<!DOCTYPE html>

<html>

<head>

<title>Password Input Control</title>

</head>

<body>

<form >

User ID : <input type = "text" name = "user\_id" />

<br>

Password: <input type = "password" name = "password" />

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **type** Indicates the type of input control and for password input control it will be set to **password**. |
| 2 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 3 | **value** This can be used to provide an initial value inside the control. |
| 4 | **size** Allows to specify the width of the text-input control in terms of characters. |
| 5 | **maxlength** Allows to specify the maximum number of characters a user can enter into the text box. |

## Multiple-Line Text Input Controls

<!DOCTYPE html>

<html>

<head>

<title>Multiple-Line Input Control</title>

</head>

<body>

<form>

Description : <br />

<textarea rows = "5" cols = "50" name = "description">

Enter description here...

</textarea>

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 2 | **rows** Indicates the number of rows of text area box. |
| 3 | **cols** Indicates the number of columns of text area box |

# Checkbox Control

<!DOCTYPE html>

<html>

<head>

<title>Checkbox Control</title>

</head>

<body>

<form>

<input type = "checkbox" name = "maths" value = "on"> Maths

<input type = "checkbox" name = "physics" value = "on"> Physics

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **type** Indicates the type of input control and for checkbox input control it will be set to **checkbox.**. |
| 2 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 3 | **value** The value that will be used if the checkbox is selected. |
| 4 | **checked** Set to *checked* if you want to select it by default. |

# Radio Button Control

<!DOCTYPE html>

<html>

<head>

<title>Radio Box Control</title>

</head>

<body>

<form>

<input type = "radio" name = "subject" value = "maths"> Maths

<input type = "radio" name = "subject" value = "physics"> Physics

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **type** Indicates the type of input control and for checkbox input control it will be set to radio. |
| 2 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 3 | **value** The value that will be used if the radio box is selected. |
| 4 | **checked** Set to *checked* if you want to select it by default. |

Select Box Control

<!DOCTYPE html>

<html>

<head>

<title>Select Box Control</title>

</head>

<body>

<form>

<select name = "dropdown">

<option value = "Maths" selected>Maths</option>

<option value = "Physics">Physics</option>

</select>

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 2 | **size** This can be used to present a scrolling list box. |
| 3 | **multiple** If set to "multiple" then allows a user to select multiple items from the menu. |

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **value** The value that will be used if an option in the select box box is selected. |
| 2 | **selected** Specifies that this option should be the initially selected value when the page loads. |
| 3 | **label** An alternative way of labeling options |

# File Upload Box

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<input type = "file" name = "fileupload" accept = "image/\*" />

</form>

</body>

</html>

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **name** Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 2 | **accept** Specifies the types of files that the server accepts. |

# Button Controls

|  |  |
| --- | --- |
| **Sr.No** | **Type & Description** |
| 1 | **submit** This creates a button that automatically submits a form. |
| 2 | **reset** This creates a button that automatically resets form controls to their initial values. |
| 3 | **button** This creates a button that is used to trigger a client-side script when the user clicks that button. |
| 4 | **image** This creates a clickable button but we can use an image as background of the button. |

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

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

<input type = "reset" name = "reset" value = "Reset" />

<input type = "button" name = "ok" value = "OK" />

<input type = "image" name = "imagebutton" src = "/html/images/logo.png" />

</form>

</body>

</html>

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<p>This is page 10</p>

<input type = "hidden" name = "pagename" value = "10" />

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

<input type = "reset" name = "reset" value = "Reset" />

</form>

</body>

</html>

HTML - Embed Multimedia

Sometimes you need to add music or video into your web page. The easiest way to add video or sound to your web site is to include the special HTML tag called **<embed>**. This tag causes the browser itself to include controls for the multimedia automatically provided browser supports <embed> tag and given media type.

You can also include a **<noembed>** tag for the browsers which don't recognize the <embed> tag. You could, for example, use <embed> to display a movie of your choice, and **<noembed>** to display a single JPG image if browser does not support <embed> tag.

<!DOCTYPE html>

<html>

<head>

<title>HTML embed Tag</title>

</head>

<body>

<embed src = "/html/yourfile.mid" width = "100%" height = "60" >

<noembed><img src = "yourimage.gif" alt = "Alternative Media" ></noembed>

</embed>

</body>

</html>

HTML - Header

Document declaration tag

<html>

<head>

Document header related tags

</head>

<body>

Document body related tags

</body>

</html>

# The HTML <title> Tag

<!DOCTYPE html>

<html>

<head>

<title>HTML Title Tag </title>

</head>

<body>

<p>Hello, World!</p>

</body>

</html>

# The HTML <meta> Tag

<!DOCTYPE html>

<html>

<head>

<title>HTML Meta Tag </title>

<!-- Provide list of keywords -->

<meta name = "keywords" content = "C, C++, Java, PHP, Perl, Python">

<!-- Provide description of the page -->

<meta name = "description" content = "Simply Easy Learning by Tutorials Point">

<!-- Author information -->

<meta name = "author" content = "Tutorials Point">

<!-- Page content type -->

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

<!-- Page refreshing delay -->

<meta http-equiv = "refresh" content = "30">

<!-- Page expiry -->

<meta http-equiv = "expires" content = "Wed, 21 June 2006 14:25:27 GMT">

<!-- Tag to tell robots not to index the content of a page -->

<meta name = "robots" content = "noindex, nofollow">

</head>

<body>

<p>Hello, World!</p>

</body>

</html>

# The HTML <base> Tag

The HTML <base> tag is used for specifying the base URL for all relative URLs in a page, which means all the other URLs will be concatenated into base URL while locating for the given item.

<!DOCTYPE html>

<html>

<head>

<title>HTML Base Tag </title>

<base href = "https://www.tutorialspoint.com/" />

</head>

<body>

<img src = "/images/logo.png" alt = "Logo Image"/>

<a href = "/html/index.htm" title = "HTML Tutorial"/>HTML Tutorial</a>

</body>

</html>

# The HTML <link> Tag

<!DOCTYPE html>

<html>

<head>

<title>HTML link Tag </title>

<base href = "https://www.tutorialspoint.com/" />

<link rel = "stylesheet" type = "text/css" href = "/css/style.css">

</head>

<body>

<p>Hello, World!</p>

</body>

</html>

# The HTML <style> Tag

<!DOCTYPE html>

<html>

<head>

<title>HTML style Tag </title>

<base href = "https://www.tutorialspoint.com/" />

<style type = "text/css">

.myclass {

background-color: #aaa;

padding: 10px;

}

</style>

</head>

<body>

<p class = "myclass">Hello, World!</p>

</body>

</html>

# The HTML <script> Tag

<!DOCTYPE html>

<html>

<head>

<title>HTML script Tag </title>

<base href = "http://www.tutorialspoint.com/" />

<script type = "text/JavaScript">

function Hello() {

alert("Hello, World");

}

</script>

</head>

<body>

<input type = "button" onclick = "Hello();" name = "ok" value = "OK" />

</body>

</html>

HTML - Style Sheet

Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced.

<!DOCTYPE html>

<html>

<head>

<title>HTML CSS</title>

</head>

<body>

<p><font color = "green" size = "5">Hello, World!</font></p>

</body>

</html>

We can re-write above example with the help of Style Sheet as follows −

<!DOCTYPE html>

<html>

<head>

<title>HTML CSS</title>

</head>

<body>

<p style = "color:green; font-size:24px;" >Hello, World!</p>

</body>

</html>

You can use CSS in three ways in your HTML document −

* **External Style Sheet** − Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML <link> tag.
* **Internal Style Sheet** − Define style sheet rules in header section of the HTML document using <style> tag.
* **Inline Style Sheet** − Define style sheet rules directly along-with the HTML elements using **style** attribute.

# External Style Sheet

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using <link> tag.

Consider we define a style sheet file **style.css** which has following rules −

.red {

color: red;

}

.thick {

font-size:20px;

}

.green {

color:green;

}

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document −

<!DOCTYPE html>

<html>

<head>

<title>HTML External CSS</title>

<link rel = "stylesheet" type = "text/css" href = "/html/style.css">

</head>

<body>

<p class = "red">This is red</p>

<p class = "thick">This is thick</p>

<p class = "green">This is green</p>

<p class = "thick green">This is thick and green</p>

</body>

</html>

# Internal Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

<!DOCTYPE html>

<html>

<head>

<title>HTML Internal CSS</title>

<style type = "text/css">

.red {

color: red;

}

.thick{

font-size:20px;

}

.green {

color:green;

}

</style>

</head>

<body>

<p class = "red">This is red</p>

<p class = "thick">This is thick</p>

<p class = "green">This is green</p>

<p class = "thick green">This is thick and green</p>

</body>

</html>

Inline Style Sheet

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

<!DOCTYPE html>

<html>

<head>

<title>HTML Inline CSS</title>

</head>

<body>

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

<p style = "font-size:20px;">This is thick</p>

<p style = "color:green;">This is green</p>

<p style = "color:green;font-size:20px;">This is thick and green</p>

</body>

</html>

HTML - JavaScript

A **script** is a small piece of program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu. This script could be written using JavaScript or VBScript.

# External JavaScript

Consider we define a small function using JavaScript in **script.js** which has following code −

function Hello() {

alert("Hello, World");

}

Now let's make use of the above external JavaScript file in our following HTML document −

<!DOCTYPE html>

<html>

<head>

<title>Javascript External Script</title>

<script src = "/html/script.js" type = "text/javascript"/></script>

</head>

<body>

<input type = "button" onclick = "Hello();" name = "ok" value = "Click Me" />

</body>

</html>

# Internal Script

You can write your script code directly into your HTML document. Usually we keep script code in header of the document using <script> tag, otherwise there is no restriction and you can put your source code anywhere in the document but inside <script> tag.

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Internal Script</title>

<base href = "https://www.tutorialspoint.com/" />

<script type = "text/JavaScript">

function Hello() {

alert("Hello, World");

}

</script>

</head>

<body>

<input type = "button" onclick = "Hello();" name = "ok" value = "Click Me" />

</body>

</html>

# Event Handlers

Event handlers are nothing but simply defined functions which can be called against any mouse or keyboard event. You can define your business logic inside your event handler which can vary from a single to 1000s of line code.

Following example explains how to write an event handler. Let's write one simple function *EventHandler()* in the header of the document. We will call this function when any user brings mouse over a paragraph.

<!DOCTYPE html>

<html>

<head>

<title>Event Handlers </title>

<base href = "https://www.tutorialspoint.com/" />

<script type = "text/JavaScript">

function EventHandler() {

alert("I'm event handler!!");

}

</script>

</head>

<body>

<p onmouseover = "EventHandler();">Bring your mouse here to see an alert</p>

</body>

</html>

# Hide Scripts from Older Browsers

Although most (if not all) browsers these days support JavaScript, but still some older browsers don't. If a browser doesn't support JavaScript, instead of running your script, it would display the code to the user. To prevent this, you can simply place HTML comments around the script as shown below.

**JavaScript :**

<script type = "text/JavaScript">

<!--

document.write("Hello JavaScript!");

//-->

</script>

**VBScript :**

<script type = "text/vbscript">

<!--

document.write("Hello VBScript!")

'-->

</script>

# The <noscript> Element

You can also provide alternative info to the users whose browsers don't support scripts and for those users who have disabled script option their browsers. You can do this using the **<noscript>** tag.

**JavaScript :**

<script type = "text/JavaScript">

<!--

document.write("Hello JavaScript!");

//-->

</script>

<noscript>Your browser does not support JavaScript!</noscript>

**VBScript :**

<script type = "text/vbscript">

<!--

document.write("Hello VBScript!")

'-->

</script>

<noscript>Your browser does not support VBScript!</noscript>

# Default Scripting Language

There may be a situation when you will include multiple script files and ultimately using multiple <script> tags. You can specify a default scripting language for all your *script* tags. This saves you from specifying the language every time you use a script tag within the page. Below is the example −

<meta http-equiv = "Content-Script-Type" content = "text/JavaScript" />

Note that you can still override the default by specifying a language within the script tag.

HTML - Layouts.

# HTML Layout - Using Tables

The simplest and most popular way of creating layouts is using HTML <table> tag. These tables are arranged in columns and rows, so you can utilize these rows and columns in whatever way you like.

<!DOCTYPE html>

<html>

<head>

<title>HTML Layout using Tables</title>

</head>

<body>

<table width = "100%" border = "0">

<tr>

<td colspan = "2" bgcolor = "#b5dcb3">

<h1>This is Web Page Main title</h1>

</td>

</tr>

<tr valign = "top">

<td bgcolor = "#aaa" width = "50">

<b>Main Menu</b><br />

HTML<br />

PHP<br />

PERL...

</td>

<td bgcolor = "#eee" width = "100" height = "200">

Technical and Managerial Tutorials

</td>

</tr>

<tr>

<td colspan = "2" bgcolor = "#b5dcb3">

<center>

Copyright © 2007 Tutorialspoint.com

</center>

</td>

</tr>

</table>

</body>

</html>

# Multiple Columns Layout - Using Tables

You can design your webpage to put your web content in multiple pages. You can keep your content in middle column and you can use left column to use menu and right column can be used to put advertisement or some other stuff.

<!DOCTYPE html>

<html>

<head>

<title>Three Column HTML Layout</title>

</head>

<body>

<table width = "100%" border = "0">

<tr valign = "top">

<td bgcolor = "#aaa" width = "20%">

<b>Main Menu</b><br />

HTML<br />

PHP<br />

PERL...

</td>

<td bgcolor = "#b5dcb3" height = "200" width = "60%">

Technical and Managerial Tutorials

</td>

<td bgcolor = "#aaa" width = "20%">

<b>Right Menu</b><br />

HTML<br />

PHP<br />

PERL...

</td>

</tr>

<table>

</body>

</html>

HTML Layouts - Using DIV, SPAN

The <div> element is a block level element used for grouping HTML elements. While the <div> tag is a block-level element, the HTML <span> element is used for grouping elements at an inline level.

Although we can achieve pretty nice layouts with HTML tables, but tables weren't really designed as a layout tool. Tables are more suited to presenting tabular data.

**Note** − This example makes use of Cascading Style Sheet (CSS), so before understanding this example you need to have a better understanding on how CSS works.

<!DOCTYPE html>

<html>

<head>

<title>HTML Layouts using DIV, SPAN</title>

</head>

<body>

<div style = "width:100%">

<div style = "background-color:#b5dcb3; width:100%">

<h1>This is Web Page Main title</h1>

</div>

<div style = "background-color:#aaa; height:200px; width:100px; float:left;">

<div><b>Main Menu</b></div>

HTML<br />

PHP<br />

PERL...

</div>

<div style = "background-color:#eee; height:200px; width:350px; float:left;" >

<p>Technical and Managerial Tutorials</p>

</div>

<div style = "background-color:#aaa; height:200px; width:100px; float:right;">

<div><b>Right Menu</b></div>

HTML<br />

PHP<br />

PERL...

</div>

<div style = "background-color:#b5dcb3; clear:both">

<center>

Copyright © 2007 Tutorialspoint.com

</center>

</div>

</div>

</body>

</html>