**HTML Coding**

**P1. First program**

<!doctype html>

<html>

<head>

<title>HTML website</title>

</head>

<body>

<!-- it is use for comment in html-->

<h1>This is a HTML site heading</h1>

<p>this is paragraph for content writing</p>

<p>Line Break Example<br />

Thanks for this program <br />

</p>

<!-- example of center paragraph-->

<center>

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

</center>

<!—example of Horizontal lines break-up sections of a document-->

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

<hr />

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

<pre>

function testFunction( strText ){

alert (strText) }

</pre>

<!--(&nbsp) Nonbreaking Spaces Example (12 Angry Men)--> <p>An example nonbreaking spaces a movie name "12&nbsp;Angry&nbsp;Men."</p>

</body>

</html>

**Note🡪 HTML (Hyper Text Markup Language)**

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Also it is a case-insensitive.

**Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is 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.

**Basic 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 program of HTML uses the following tags: -

1. **<!DOCTYPE...>** This tag defines the document type and HTML version. The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. There are many other declaration types which can be used in HTML document depending on what version of HTML is being used.
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>** The is where you put in all the information you want to include about your document. This tag represents the document's header which can keep other HTML tags like <title>, <link> etc. it has opening and closing write as <head> ………</head>
4. **<title>** The <title> tag is used inside the <head> tag to mention the document title. it has opening and closing write as <title>…….</title>.
5. **<body>** The <body> is basically the content of your HTML document. Put in another way, *the <body> section is what people see when they view your HTML document*. This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc. it has opening and closing write as <body>…… </body>.
6. **<h1>** This tag represents the heading. 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. it has opening and closing write as <h1> ……. </h1>.
7. **<p>** This tag represents a paragraph. 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. it has opening and closing write as <p> ….. </p>.
8. **<br />** 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 also known as *Line Break 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 HTML. This Tag is an example of the **empty element**
9. **<center>** You can use **<center>** tag to put any content in the center of the page or any table cell. it has opening and closing write as <center> ….. </center>.
10. **<hr />** 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.
11. **<pre>** 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.
12. **&nbsp;** 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, 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.

To learn HTML, you will need to study various tags and understand how 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 Element**

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, 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 Element vs HTML Tag* - 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.

<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>

**HTML Attributes->**

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.

Another term you would need to know when it comes to HTML elements are attributes. Attributes modify the tags where they appear. Attributes are name­value pairs, which are separated by the equal sign.

The HTML element <p align=”center”>align this paragraph in center</p> contains the attribute align=”center". Attributes usually use the following syntax:

<tag attribute="value of attribute">content</tag>

A lot of people confuse the alt attribute as a tag. It is not. The alt attribute modifies the <img>, <area>, <input> and <applet> tags, which makes it an attribute.

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.** Attribute names and attribute values are case-insensitive.

* 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***.**

<!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>

**Core Attributes: -**

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

* Id
* Title
* Class
* Style

**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.

The ID attribute of an element is an identifier which must be unique in the whole document. Its purpose is to uniquely identify the element when linking (using an anchor), scripting, or styling (with CSS).

You should not have two elements with the same ID in the same document, even if the attributes are attached to two different kinds of elements. For example, the following code is incorrect:

<div id="example-id"></div>

<span id="example-id"></span>

Browsers will do their best to render this code, but unexpected behavior may occur when styling with CSS or adding functionality with JavaScrip

To reference elements by their ID in CSS, prefix the ID with #.

#example-id { color: green; }

To jump to an element with an ID on a given page, append # with the element name in the URL.

http://example.com/about#example-id

This feature is supported in most browsers and does not require additional JavaScript or CSS to work.

The only restrictions on the value of an id are:

1. it must be unique in the document

2. it must not contain any space characters

3. it must contain at least one character

So the value can be all digits, just one digit, just punctuation characters, include special characters, whatever. Just no whitespace

An id value must begin with a letter, which can then be followed only by:

letters (A-Z/a-z)

digits (0-9)

hyphens ("-")

underscores ("\_")

colons (":")

periods (".")

So these are valid:

<div id="container"> ... </div>

<div id="999"> ... </div>

<div id="#%LV-||"> ... </div>

<div id="\_\_\_\_V"> ... </div>

<div id="⌘⌥"> ... </div>

<div id="♥"> ... </div>

<div id="{}"> ... </div>

<div id="©"> ... </div>

<div id="♤₩¤☆€~¥"> ... </div>

This is invalid:

<div id=" "> ... </div>

This is also invalid, when included in the same document:

<div id="results"> ... </div>

<div id="results"> ... </div>

**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 Example</title>

</head>

<body>

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

This will produce the 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.

**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).

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

class="className1 className2 className3

**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>

This will produce the following result:

Some text...

**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) |

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 the 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.

<html lang="en">

<!—it Mean this page is using English Language Page-->

**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 |
| align | 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 width of tables, images, or table cells |
| title | User Defined | "Pop-up" title of the elements |
| src |  |  |
| href |  |  |
| hspace |  |  |

* **HTML FORMATTING->**

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 in box.

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

*This will produce the following result:* The following word uses a **bold** typeface

***Italic Text*** Anything that appears within **<i>...</i>** element is displayed in italicized as shown in box.

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

*This will produce the following result:* The following word uses an *italicized* typeface

***Underlined Text*** Anything that appears within **<u>...</u>** element, is displayed with underline as shown in box.

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

*This will produce the following result:* The following word uses an 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 in box.

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the following result:* The following word uses a subscript typeface

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

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

*This will produce the following result:* I want to drink wine

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

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

*This will produce the following result:* I want to drink ~~cola~~

***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 in box.

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

*This will produce the 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 in box.

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

*This will produce the 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.

The difference between the div tag and the span tag is that the div tag is used with block level elements whilst the span tag is used with inline elements.

The HTML <div> tag is used for defining a section of your document. With the div tag, you can group large sections of HTML elements together and format them with CSS

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 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 the following result

HOME | CONTACT | ABOUT

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 the following result:

This is the example of span tag and the div tag along with CSS

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

* **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.

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

*This will produce the following result:*

The following word uses an *emphasized* typeface

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

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

*This will produce the following result:*

The following word has been marked with yellow.

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

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the following result:* This text will go left to right.

.Thgir ot tfel og lliw txet siht

***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.

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

*This will produce the 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.

<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>

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the 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.

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

*This will produce the 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.

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

This will produce the following result:

Result produced by the program is Hello World!

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

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

This will produce the following result:

*388A, Road No 22, Jubilee Hills - Hyderabad*

* **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 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 tags inside the header of the document which is represented by and 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 the 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.

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 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 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.

**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 Example</title>

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

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

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

</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 Example</title>

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

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

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

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

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

* **HTML COMMENTS->**

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 the 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 cannot 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.

Here, the 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 the 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.

<!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 the 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.

<!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.

<!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.

<!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 the 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 Cascading 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.

<!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 the following result:

Hello, World!

* **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** 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.

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="test.png" alt="Test Image" />

</body>

</html>

*This will produce the 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 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.

Assuming our image location is "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="images/test.png" alt="Test Image" />

</body>

</html>

*This will produce the 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.

<!DOCTYPE html>

<html>

<head>

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

<body>

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

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

</body>

</html>

*This will produce the 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

<!DOCTYPE html>

<html>

<head>

<title>Set Image Border</title>

</head>

<body>

<p>Setting image Border</p>

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

</body>

</html>

*This will produce the following result:*

Setting image Border

****

**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="test.png" alt="Test Image" border="3" align="right"/>

</body>

</html>

*This will produce the following result:*

Setting image Alignment

****

* **HTML TABLES->**

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.

<!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 the following result:*

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

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".