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

HTML

1. HTML – OVERVIEW

* + **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.

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

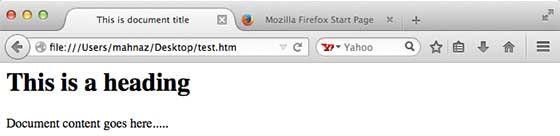
Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

# Basic HTML Document

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

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>This is document title</title>  </head>  <body>  <h1>This is a heading</h1>  <p>Document content goes here </p>  </body>  </html> |

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



# HTML Tags

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

Above example of HTML document uses the following tags:

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

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

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

# HTML Document Structure

A typical HTML document will have the following structure:

|  |
| --- |
| Document declaration tag  <html>  <head>  Document header related tags  </head>  <body>  Document body related tags  </body>  </html> |

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

# The <!DOCTYPE> Declaration

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

|  |
| --- |
| <!DOCTYPE html> |

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing

<!DOCTYPE...> tag along with other HTML tags.

HTML

# Heading Tags

2. HTML – BASIC TAGS

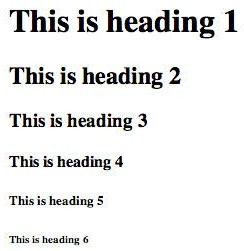
Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>,**

**<h5>, and <h6>**. While displaying any heading, browser adds one line before and one line after that heading.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Heading Example</title>  </head>  <body>  <h1>This is heading 1</h1>  <h2>This is heading 2</h2>  <h3>This is heading 3</h3>  <h4>This is heading 4</h4>  <h5>This is heading 5</h5>  <h6>This is heading 6</h6>  </body>  </html> |

This will produce the following result:



# Paragraph Tag

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Paragraph Example</title>  </head>  <body>  <p>Here is a first paragraph of text.</p>  <p>Here is a second paragraph of text.</p>  <p>Here is a third paragraph of text.</p>  </body>  </html> |

This will produce the following result:

|  |
| --- |
| Here is a first paragraph of text.  Here is a second paragraph of text. Here is a third paragraph of text. |

# Line Break Tag

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Line Break Example</title>  </head>  <body> |

|  |
| --- |
| <p>Hello<br />  You delivered your assignment on time.<br /> Thanks<br />  Mahnaz</p>  </body>  </html> |

This will produce the following result:

|  |
| --- |
| Hello  You delivered your assignment on time. Thanks  Mahnaz |

# Centering Content

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Centring Content Example</title>  </head>  <body>  <p>This text is not in the center.</p>  <center>  <p>This text is in the center.</p>  </center>  </body>  </html> |

This will produce the following result:

This text is not in the center.

This text is in the center.

# Horizontal Lines

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Horizontal Line Example</title>  </head>  <body>  <p>This is paragraph one and should be on top</p>  <hr />  <p>This is paragraph two and should be at bottom</p>  </body>  </html> |

This will produce the following result:

This is paragraph one and should be on top

This is paragraph two and should be at bottom

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

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

# Preserve Formatting

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head> |

|  |
| --- |
| <title>Preserve Formatting Example</title> |

</head>

|  |
| --- |
| <body>  <pre>  function testFunction( strText ){ alert (strText)  }  </pre>  </body>  </html> |

This will produce the following result: function testFunction( strText ){

alert (strText)

}

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

# Nonbreaking Spaces

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

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

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

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Nonbreaking Spaces Example</title>

</head>

<body>

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

</body>

|  |
| --- |
| </html> |

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

HTML

3. HTML – ELEMENTS

|  |  |  |
| --- | --- | --- |
| **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 Tag vs. Element

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

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

# Nested HTML Elements

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head> |
| <title>Nested Elements Example</title>  </head>  <body>  <h1>This is <i>italic</i> heading</h1>  <p>This is <u>underlined</u> paragraph</p>  </body>  </html> |

This will display the following result:

**This is *italic* heading**

This is underlined paragraph

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

HTML

4. 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Align Attribute Example</title>  </head>  <body>  <p align="left">This is left aligned</p>  <p align="center">This is center aligned</p>  <p align="right">This is right aligned</p>  </body>  </html> |

This will display the following result:

This is left aligned

This is center aligned

This is right aligned

# Core Attributes

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

* + Id
  + Title
  + Class
  + Style

**The Id Attribute**

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

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

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

**Example**

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

**The title Attribute**

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>The title Attribute Example</title>  </head>  <body>  <h3 title="Hello HTML!">Titled Heading Tag Example</h3> |
| </body>  </html> |

This will produce 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.

**The class Attribute**

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

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

|  |
| --- |
| class="className1 className2 className3" |

**The style Attribute**

The style attribute allows you to specify 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...

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

# Internationalization Attributes

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

* + - dir
    - lang
    - xml:lang

**The dir Attribute**

The **dir** attribute allows you to indicate to the browser 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. Check for a complete list of language codes.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>English Language Page</title>  </head>  <body>  This page is using English Language  </body>  </html> |

# The xml:lang Attribute

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

**Generic Attributes**

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

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

We will see related examples as we will proceed to study other HTML tags. For a complete list of HTML Tags and related attributes please check reference to.

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.

HTML

5. HTML – FORMATTING

# Bold Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Bold Text Example</title>  </head>  <body>  <p>The following word uses a <b>bold</b> typeface.</p>  </body>  </html> |

This will produce 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 below:

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Italic Text Example</title>  </head>  <body>  <p>The following word uses a <i>italicized</i> typeface.</p> |

|  |
| --- |
| </body>  </html> |

This will produce 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 below:

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Underlined Text Example</title>  </head>  <body>  <p>The following word uses a <u>underlined</u> typeface.</p>  </body>  </html> |

This will produce 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 below:

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Strike Text Example</title>  </head>  <body>  <p>The following word uses a <strike>strikethrough</strike> typeface.</p>  </body> |
| </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Monospaced Font Example</title>  </head>  <body>  <p>The following word uses a <tt>monospaced</tt> typeface.</p>  </body>  </html> |

This will produce 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.

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Superscript Text Example</title>

</head>

<body>

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

|  |
| --- |
| </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Subscript Text Example</title>  </head>  <body>  <p>The following word uses a <sub>subscript</sub> typeface.</p>  </body>  </html> |

This will produce the following result:

The following word uses a subscript typeface.

# Inserted Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Inserted Text Example</title>  </head>  <body>  <p>I want to drink <del>cola</del> <ins>wine</ins></p>  </body> |



# Deleted Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Deleted Text Example</title>  </head>  <body>  <p>I want to drink <del>cola</del> <ins>wine</ins></p>  </body>  </html> |

This will produce the following result:



# Larger Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Larger Text Example</title>  </head>  <body>  <p>The following word uses a <big>big</big> typeface.</p>  </body> |
| The following word uses a **big** typeface. |

# Smaller Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Smaller Text Example</title>  </head>  <body>  <p>The following word uses a <small>small</small> typeface.</p>  </body>  </html> |

This will produce 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.

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Div Tag Example</title>  </head> |

|  |
| --- |
| <body>  <div id="menu" align="middle" >  <a href="/index.htm">HOME</a> |  <a href="/about/contact\_us.htm">CONTACT</a> |  <a href="/about/index.htm">ABOUT</a>  </div>  <div id="content" align="left" bgcolor="white">  <h5>Content Articles</h5>  <p>Actual content goes here </p>  </div>  </body>  </html> |

This will produce the following result:

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

CONTENT ARTICLES

Actual content goes here.....

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Span Tag Example</title>  </head>  <body>  <p>This is the example of <span style="color:green">span tag</span> and the  <span style="color:red">div tag</span> alongwith CSS</p>  </body>  </html> |

This will produce 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.

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.

HTML

6. HTML – PHRASE TAGS

# Emphasized Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Emphasized Text Example</title>  </head>  <body>  <p>The following word uses a <em>emphasized</em> typeface.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Marked Text Example</title>  </head>  <body>  <p>The following word has been <mark>marked</mark> with yellow</p> |
| </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Strong Text Example</title>  </head>  <body>  <p>The following word uses a <strong>strong</strong> typeface.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Text Abbreviation</title>  </head>  <body>  <p>My best friend's name is <abbr title="Abhishek">Abhy</abbr>.</p>  </body> |
| </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Acronym Example</title>  </head>  <body>  <p>This chapter covers marking up text in <acronym>XHTML</acronym>.</p>  </body>  </html> |

This will produce 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.

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Text Direction Example</title>

</head>

<body>

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

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

|  |
| --- |
| </body>  </html> |

This will produce the following result:

This text will go left to right. This text will go right to left.

# Special Terms

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Special Terms Example</title>  </head>  <body>  <p>The following word is a <dfn>special</dfn> term.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head> |
| <title>Blockquote Example</title>  </head> |

|  |
| --- |
| <body>  <p>The following description of XHTML is taken from the W3C Web site:</p>  <blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Double Quote Example</title>  </head>  <body>  <p>Amit is in Spain, <q>I think I am wrong</q>.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Citations Example</title>  </head>  <body>  <p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Computer Code Example</title>  </head>  <body>  <p>Regular text. <code>This is code.</code> Regular text.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Keyboard Text Example</title>  </head>  <body>  <p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Variable Text Example</title>  </head>  <body>  <p><code>document.write("<var>user-name</var>")</code></p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html> |

|  |
| --- |
| <html>  <head>  <title>Program Output Example</title>  </head>  <body>  <p>Result produced by the program is <samp>Hello World!</samp></p>  </body>  </html> |

This will produce the following result:

Result produced by the program is Hello World!

# Address Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Address Example</title>  </head>  <body>  <address>388A, Road No 22, Jubilee Hills - Hyderabad</address>  </body>  </html> |

This will produce the following result:

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

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

HTML

* + **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.

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

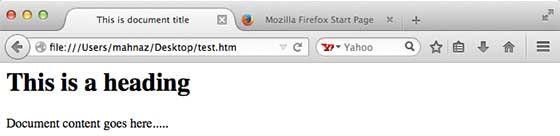
Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

# Basic HTML Document

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

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>This is document title</title>  </head>  <body>  <h1>This is a heading</h1>  <p>Document content goes here </p>  </body>  </html> |

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



# HTML Tags

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

Above example of HTML document uses the following tags:

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

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

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

# HTML Document Structure

A typical HTML document will have the following structure:

|  |
| --- |
| Document declaration tag  <html>  <head>  Document header related tags  </head>  <body>  Document body related tags  </body>  </html> |

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

# The <!DOCTYPE> Declaration

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

|  |
| --- |
| <!DOCTYPE html> |

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing

<!DOCTYPE...> tag along with other HTML tags.

HTML

# Heading Tags

2. HTML – BASIC TAGS

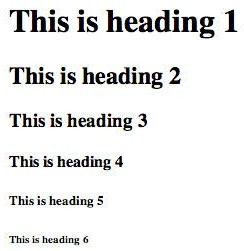
Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>,**

**<h5>, and <h6>**. While displaying any heading, browser adds one line before and one line after that heading.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Heading Example</title>  </head>  <body>  <h1>This is heading 1</h1>  <h2>This is heading 2</h2>  <h3>This is heading 3</h3>  <h4>This is heading 4</h4>  <h5>This is heading 5</h5>  <h6>This is heading 6</h6>  </body>  </html> |

This will produce the following result:



# Paragraph Tag

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Paragraph Example</title>  </head>  <body>  <p>Here is a first paragraph of text.</p>  <p>Here is a second paragraph of text.</p>  <p>Here is a third paragraph of text.</p>  </body>  </html> |

This will produce the following result:

|  |
| --- |
| Here is a first paragraph of text.  Here is a second paragraph of text. Here is a third paragraph of text. |

# Line Break Tag

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Line Break Example</title>  </head>  <body> |
| <p>Hello<br />  You delivered your assignment on time.<br /> Thanks<br />  Mahnaz</p>  </body>  </html> |

This will produce the following result:

|  |
| --- |
| Hello  You delivered your assignment on time. Thanks  Mahnaz |

# Centering Content

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Centring Content Example</title>  </head>  <body>  <p>This text is not in the center.</p>  <center>  <p>This text is in the center.</p>  </center>  </body>  </html> |

This will produce the following result:

This text is not in the center.

This text is in the center.

# Horizontal Lines

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Horizontal Line Example</title>  </head>  <body>  <p>This is paragraph one and should be on top</p>  <hr />  <p>This is paragraph two and should be at bottom</p>  </body>  </html> |

This will produce the following result:

This is paragraph one and should be on top

This is paragraph two and should be at bottom

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

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

# Preserve Formatting

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head> |

|  |
| --- |
| <title>Preserve Formatting Example</title> |

</head>

|  |
| --- |
| <body>  <pre>  function testFunction( strText ){ alert (strText)  }  </pre>  </body>  </html> |

This will produce the following result: function testFunction( strText ){

alert (strText)

}

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

# Nonbreaking Spaces

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

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

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

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Nonbreaking Spaces Example</title>

</head>

<body>

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

</body>

|  |
| --- |
| </html> |

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

HTML

3. HTML – ELEMENTS

|  |  |  |
| --- | --- | --- |
| **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 Tag vs. Element

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

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

# Nested HTML Elements

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head> |
| <title>Nested Elements Example</title>  </head>  <body>  <h1>This is <i>italic</i> heading</h1>  <p>This is <u>underlined</u> paragraph</p>  </body>  </html> |

This will display the following result:

**This is *italic* heading**

This is underlined paragraph

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

HTML

4. 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Align Attribute Example</title>  </head>  <body>  <p align="left">This is left aligned</p>  <p align="center">This is center aligned</p>  <p align="right">This is right aligned</p>  </body>  </html> |

This will display the following result:

This is left aligned

This is center aligned

This is right aligned

# Core Attributes

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

* + Id
  + Title
  + Class
  + Style

**The Id Attribute**

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

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

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

**Example**

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

**The title Attribute**

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>The title Attribute Example</title>  </head>  <body>  <h3 title="Hello HTML!">Titled Heading Tag Example</h3> |
| </body>  </html> |

This will produce 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.

**The class Attribute**

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

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

|  |
| --- |
| class="className1 className2 className3" |

**The style Attribute**

The style attribute allows you to specify 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...

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

# Internationalization Attributes

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

* + - dir
    - lang
    - xml:lang

**The dir Attribute**

The **dir** attribute allows you to indicate to the browser 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. Check [**HTML Language Codes: ISO 639**](http://www.tutorialspoint.com/html/language_iso_codes.htm) for a complete list of language codes.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <title>English Language Page</title>  </head>  <body>  This page is using English Language  </body>  </html> |

# The xml:lang Attribute

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

**Generic Attributes**

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

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

We will see related examples as we will proceed to study other HTML tags. For a complete TML Tags and related attributes please check reference to [**HTML Tags List**](http://www.tutorialspoint.com/html/html_tags_ref.htm).

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.

HTML

5. HTML – FORMATTING

# Bold Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Bold Text Example</title>  </head>  <body>  <p>The following word uses a <b>bold</b> typeface.</p>  </body>  </html> |

This will produce 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 below:

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Italic Text Example</title>  </head>  <body>  <p>The following word uses a <i>italicized</i> typeface.</p> |
| </body>  </html> |

This will produce 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 below:

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Underlined Text Example</title>  </head>  <body>  <p>The following word uses a <u>underlined</u> typeface.</p>  </body>  </html> |

This will produce 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 below:

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Strike Text Example</title>  </head>  <body>  <p>The following word uses a <strike>strikethrough</strike> typeface.</p>  </body> |
| </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Monospaced Font Example</title>  </head>  <body>  <p>The following word uses a <tt>monospaced</tt> typeface.</p>  </body>  </html> |

This will produce 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.

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Superscript Text Example</title>

</head>

<body>

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

|  |
| --- |
| </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Subscript Text Example</title>  </head>  <body>  <p>The following word uses a <sub>subscript</sub> typeface.</p>  </body>  </html> |

This will produce the following result:

The following word uses a subscript typeface.

# Inserted Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Inserted Text Example</title>  </head>  <body>  <p>I want to drink <del>cola</del> <ins>wine</ins></p>  </body> |



# Deleted Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Deleted Text Example</title>  </head>  <body>  <p>I want to drink <del>cola</del> <ins>wine</ins></p>  </body>  </html> |

This will produce the following result:



# Larger Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Larger Text Example</title>  </head>  <body>  <p>The following word uses a <big>big</big> typeface.</p>  </body> |
| The following word uses a **big** typeface. |

# Smaller Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Smaller Text Example</title>  </head>  <body>  <p>The following word uses a <small>small</small> typeface.</p>  </body>  </html> |

This will produce 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.

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Div Tag Example</title>  </head> |

|  |
| --- |
| <body>  <div id="menu" align="middle" >  <a href="/index.htm">HOME</a> |  <a href="/about/contact\_us.htm">CONTACT</a> |  <a href="/about/index.htm">ABOUT</a>  </div>  <div id="content" align="left" bgcolor="white">  <h5>Content Articles</h5>  <p>Actual content goes here </p>  </div>  </body>  </html> |

This will produce the following result:

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

CONTENT ARTICLES

Actual content goes here.....

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Span Tag Example</title>  </head>  <body>  <p>This is the example of <span style="color:green">span tag</span> and the  <span style="color:red">div tag</span> alongwith CSS</p>  </body>  </html> |

This will produce 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.

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.

HTML

6. HTML – PHRASE TAGS

# Emphasized Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Emphasized Text Example</title>  </head>  <body>  <p>The following word uses a <em>emphasized</em> typeface.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Marked Text Example</title>  </head>  <body>  <p>The following word has been <mark>marked</mark> with yellow</p> |
| </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Strong Text Example</title>  </head>  <body>  <p>The following word uses a <strong>strong</strong> typeface.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Text Abbreviation</title>  </head>  <body>  <p>My best friend's name is <abbr title="Abhishek">Abhy</abbr>.</p>  </body> |
| </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Acronym Example</title>  </head>  <body>  <p>This chapter covers marking up text in <acronym>XHTML</acronym>.</p>  </body>  </html> |

This will produce 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.

**Example**

<!DOCTYPE html>

<html>

<head>

<title>Text Direction Example</title>

</head>

<body>

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

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

|  |
| --- |
| </body>  </html> |

This will produce the following result:

This text will go left to right. This text will go right to left.

# Special Terms

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

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Special Terms Example</title>  </head>  <body>  <p>The following word is a <dfn>special</dfn> term.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head> |

|  |
| --- |
| <title>Blockquote Example</title>  </head> |

|  |
| --- |
| <body>  <p>The following description of XHTML is taken from the W3C Web site:</p>  <blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Double Quote Example</title>  </head>  <body>  <p>Amit is in Spain, <q>I think I am wrong</q>.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Citations Example</title>  </head>  <body>  <p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Computer Code Example</title>  </head>  <body>  <p>Regular text. <code>This is code.</code> Regular text.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Keyboard Text Example</title>  </head>  <body>  <p>Regular text. <kbd>This is inside kbd element</kbd> Regular text.</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Variable Text Example</title>  </head>  <body>  <p><code>document.write("<var>user-name</var>")</code></p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html> |

|  |
| --- |
| <html>  <head>  <title>Program Output Example</title>  </head>  <body>  <p>Result produced by the program is <samp>Hello World!</samp></p>  </body>  </html> |

This will produce the following result:

Result produced by the program is Hello World!

# Address Text

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

**Example**

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Address Example</title>  </head>  <body>  <address>388A, Road No 22, Jubilee Hills - Hyderabad</address>  </body>  </html> |

This will produce the following result:

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

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

8. 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.

**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.

**Example**

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.

**Example**

<!DOCTYPE html><html>

<head>

<title>Multiline Comments</title>

</head>

<body>

<!--

This is a multiline comment and it can span through as many as lines you like.

|  |
| --- |
| -->  <p>Document content goes here </p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html><html>  <head>  <title>Conditional Comments</title>  <!--[if IE 6]>  Special instructions for IE 6 here  <![endif]-->  </head>  <body>  <p>Document content goes here </p>  </body>  </html> |

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

# Using Comment Tag

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

**Example**

|  |
| --- |
| <!DOCTYPE html><html>  <head>  <title>Using Comment Tag</title> |

|  |
| --- |
| </head>  <body>  <p>This is <comment>not</comment> Internet Explorer.</p>  </body>  </html> |

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

This is Internet Explorer.

But if you are not using IE, then it will produce following result: This is Internet Explorer.

# Commenting Script Code

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

**Example**

|  |
| --- |
| <!DOCTYPE html><html>  <head>  <title>Commenting Script Code</title>  <script>  <!--  document.write("Hello World!")  //-->  </script>  </head>  <body>  <p>Hello , World!</p>  </body>  </html> |

This will produce 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.

**Example**

|  |
| --- |
| <!DOCTYPE html><html>  <head>  <title>Commenting Style Sheets</title>  <style>  <!--  .example {  border:1px solid #4a7d49;  }  //-->  </style>  </head>  <body>  <div class="example">Hello , World!</div>  </body>  </html> |

This will produce the following result:

|  |
| --- |
| Hello, World! |

9. 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.

HTML

# Insert Image

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

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

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

**Example**

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

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Using Image in Webpage</title>  </head>  <body>  <p>Simple Image Insert</p>  <img src="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.

**Example**

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.

**Example**

|  |
| --- |
| <!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.

**Example**

|  |
| --- |
| <!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.

**Example**

|  |
| --- |
| <!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