HTML
Hypertext Markup Language
Notes - Part I

Introduction

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

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

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

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

Basic HTML Document

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

```
<!DOCTYPE html>
<html>
        <head>
                <meta charset="UTF-8">
                <title>Hello World!!</title>
        </head>

        <body>
                <h1>HTML Hello World</h1>
                <p>Content of the document.. </p>
        </body>
</html>
```

Let's save it in an HTML file test.html using your favorite text editor (notepad++). Finally open it using a web browser like Google Chrome, or Firefox etc. It must show the following output:

HTML Hello World
Content of the document

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 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>, , <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 students have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

A complete list of standard tags available in HTML5 is given below. All the tags are ordered alphabetically along with an indication if they have been introduced newly or they have been deprecated in HTML5.

Tag	Description
<!--...-->	Specifies a comment
<!DOCTYPE>	Specifies the document type
<a>	Specifies an anchor
<abbr>	Specifies an abbreviation
<acronym>	Deprecated: Specifies an acronym
<address>	Specifies an address element
<applet>	Deprecated: Specifies an applet
<area>	Specifies an area inside an image map
<article>	New Tag: Specifies an independent piece of content of a document, such as a blog entry or newspaper article
<aside>	New Tag: Specifies a piece of content that is only slightly related to the rest of the page.
<audio>	New Tag: Specifies an audio file.
<base>	Specifies a base URL for all the links in a page
<basefont>	Deprecated: Specifies a base font
<bdo>	Specifies the direction of text display
<bgsound>	Specifies the background music
<blink>	Specifies a text which blinks
<blockquote>	Specifies a long quotation
<body>	Specifies the body element
** **	Inserts a single line break
<button>	Specifies a push button
<canvas>	New Tag: This is used for rendering dynamic bitmap graphics on the fly, such as graphs or games.
<caption>	Specifies a table caption
** **	Deprecated: Specifies centered text
<col>	Specifies attributes for table columns
<colgroup>	Specifies groups of table columns
<command>	New Tag: Specifies a command the user can invoke.
<comment>	Puts a comment in the document
<datalist>	New Tag: Together with the a new list attribute for input can be used to make comboboxes
<dd>	Specifies a definition description
<del>	Specifies deleted text
<details>	New Tag: Specifies additional information or controls which the user can obtain on demand.
<dir>	Deprecated: Specifies a directory list
** **	Specifies a section in a document

<dl>	Specifies a definition list
<dt>	Specifies a definition term
<embed>	New Tag: Defines external interactive content or plugin.
<fieldset>	Specifies a fieldset
<figure>	New Tag: Specifies a piece of self-contained flow content, typically referenced as a single unit from the main flow of the document.
<b>	Specifies bold text
<big>	Deprecated: Specifies big text
<i>	Specifies italic text
<small>	Specifies small text
<tt>	Deprecated: Specifies teletype text
<font>	Deprecated: Specifies text font, size, and color
<footer>	New Tag: Specifies a footer for a section and can contain information about the author, copyright information, et cetera.
<form>	Specifies a form
<frame>	Deprecated: Specifies a sub window (a frame)
<frameset>	Deprecated: Specifies a set of frames
<head>	Specifies information about the document
<header>	New Tag: Specifies a group of introductory or navigational aids.
<hgroup>	New Tag: Specifies the header of a section.
<h1> to <h6>	Specifies header 1 to header 6
<hr>	Specifies a horizontal rule
<html>	Specifies an html document
<isindex>	Deprecated: Specifies a single-line input field
<iframe>	Specifies an inline sub window (frame)
<ilayer>	Specifies an inline layer
<img>	Specifies an image
<input>	Specifies an input field
<ins>	Specifies inserted text
<keygen>	New Tag: Specifies control for key pair generation.
<keygen>	Generate key information in a form
<label>	Specifies a label for a form control
<layer>	Specifies a layer
<legend>	Specifies a title in a fieldset
** **	Specifies a list item
<link>	Specifies a resource reference
<map>	Specifies an image map
<mark>	New Tag: Specifies a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.
<marquee>	Create a scrolling-text marquee
<menu>	Deprecated: Specifies a menu list
<meta>	Specifies meta information
<meter>	New Tag: Specifies a measurement, such as disk usage.
<multicol>	Specifies a multicolumn text flow
<nav>	New Tag: Specifies a section of the document intended for navigation.
<nobr>	No breaks allowed in the enclosed text
<noembed>	Specifies content to be presented by browsers that do not support the <embed>tag
<noframes>	Deprecated: Specifies a noframe section
<noscript>	Specifies a noscript section
<object>	Specifies an embedded object
** **	Specifies an ordered list
<optgroup>	Specifies an option group
<option>	Specifies an option in a drop-down list
<output>	New Tag: Specifies some type of output, such as from a calculation done through scripting.
<p>	Specifies a paragraph
<param>	Specifies a parameter for an object
<cite>	Specifies a citation
<code>	Specifies computer code text
<dfn>	Specifies a definition term
****	Specifies emphasized text

<kbd>	Specifies keyboard text
<samp>	Specifies sample computer code
****	Specifies strong text
<var>	Specifies a variable
<plaintext>	Deprecated: Render the text content of the document as preformatted plain text
<pre>	Specifies preformatted text
<progress>	New Tag: Specifies a completion of a task, such as downloading or when performing a series of expensive operations.
<q>	Specifies a short quotation
<ruby>	New Tag: Together with <rt> and <rp> allow for marking up ruby annotations.
<script>	Specifies a script
<section>	New Tag: Represents a generic document or application section.
<select>	Specifies a selectable list
<spacer>	Specifies a white space
****	Specifies a section in a document
<s>	Deprecated: Specifies strikethrough text
<strike>	Deprecated: Specifies strikethrough text
<style>	Specifies a style definition
<sub>	Specifies subscripted text
<sup>	Specifies superscripted text
<table>	Specifies a table
<tbody>	Specifies a table body
<td>	Specifies a table cell
<textarea>	Specifies a text area
<tfoot>	Specifies a table footer
<th>	Specifies a table header
<thead>	Specifies a table header
<time>	New Tag: Specifies a date and/or time.
<title>	Specifies the document title
<tr>	Specifies a table row
<u>	Deprecated: Specifies underlined text
** **	Specifies an unordered list
<video>	New Tag: Specifies a video file.
<wbr>	New Tag: Specifies a line break opportunity.
<wbr>	Indicate a potential word break point within a <nobr> section
<xmp>	Deprecated: Specifies preformatted text

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

HTML Document Structure

A typical HTML document will have following structure:

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

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

The <!DOCTYPE> Declaration

The **<!DOCTYPE>** declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 (HTML5) and it makes use of the following declaration:

<!DOCTYPE html>

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

Heading Tags

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

Example:

```
<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>
```

This will produce following result:

This is heading 1
This is heading 2
This is heading 3
This is heading 4
This is heading 5
This is heading 6

Paragraph Tag

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

Example:

```
<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>
```

This will produce following result:

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

Line Break Tag

Whenever you use the **
** 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 **
** 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 **
** it is not valid in XHTML

Example:

```
<body>
    <p>Hello<br />
    I am writing to confirm that you delivered your assignment in good time.<br />
    Kind regards,<br />
    Raymond</p>
</body>
```

This will produce following result:

Hello
I am writing to confirm that you delivered your assignment in good time.
Kind regards,
Raymond

Centering Content

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

Example:

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

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

This will produce following result:

This text is not in the center.

This text is in the center.

Horizontal Lines

Horizontal lines are used to visually break up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.
For example you may want to give a line between two paragraphs as in the given example below:

```
<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>
```

This will produce following result:

This is paragraph one and should be on top

This is paragraph two and should be at bottom

Note here that **<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 those cases, you can use the preformatted tag **<pre>**.

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

```
<body>
    <pre>
        function testFunction( strText ){
            alert (strText)
        }
    </pre>
</body>
```

This will produce following result:

```
function testFunction( strText ){
    alert (strText)
}
```

HTML Elements

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

Start Tag	Content	End Tag
	This is paragraph content.	
<h1>	This is heading content.	</h1>
	This is division content.	

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

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

HTML Tag vs. Element

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

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

Nested HTML Elements

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

Example:

```
<body>
    <h1>This is <i>italic</i> heading</h1>
    <p>This is <u>underlined</u> paragraph</p>
</body>
```

This will display following result:

This is *italic* heading

This is underlined paragraph

HTML Attributes

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

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

- The **name** is the property you want to set. For example, the paragraph 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:

```
<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>
```

This will display following result:

This is left aligned

This is center aligned

This is right aligned

Core Attributes

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

- id
- title
- class
- style

The id Attribute

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

i. If an element carries an id attribute as a unique identifier it is possible to identify just that element and its content.
ii. 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:

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

The title Attribute

The **title** attribute gives a suggested title for the element. The 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:

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

This will produce following result:

Titled Heading Tag Example

On the web page 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.

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

This will produce following result:

Some text...

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

Internationalization Attributes

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

- dir
- lang
- xml:lang

The dir Attribute

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

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

Example:

```
<!DOCTYPE html>
<html dir = "rtl">
    <head>
        <title>Display Directions</title>
    </head>
    <body>
        This is how IE 5 renders right-to-left directed text.
    </body>
</html>
```

This will produce following result:

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

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

The lang Attribute

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

The xml:lang Attribute

The **xml:lang** attribute is the XHTML replacement for the **lang** attribute.

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.

Standard Attributes

The attributes listed below are supported by almost all the HTML 5 tags.

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.
contextmenu	Menu id	Specifies the context menu for an element.
data-XXXX	User Defined	Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data-".
draggable	true,false, auto	Specifies whether or not a user is allowed to drag an element.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.
id	User Defined	Names an element for use with Cascading Style Sheets.
item	List of elements	Used to group elements.
itemprop	List of items	Used to group items.
spellcheck	true, false	Specifies if the element must have its spelling or grammar checked.
style	CSS Style sheet	Specifies an inline style for an element.
subject	User define id	Specifies the element's corresponding item.
tabindex	Tab number	Specifies the tab order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

Bold Text

Anything that appears within ... (or ...) element, is displayed in bold as shown below:

Example:

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

This will produce following result:

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

Italic Text

Anything that appears within ... element is displayed in italicized as shown below:

Example:

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

This will produce 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:

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

This will produce 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:

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

This will produce following result:

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

Superscript Text

The content of a ^{...} 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:

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

This will produce following result:

The following word uses a $^{\text{superscript}}$ typeface.

Subscript Text

The content of a _{...} 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:

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

This will produce following result:

The following word uses a $_{\text{subscript}}$ typeface.

Grouping Content

The and 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 element to indicate that all of the elements within that element relate to the footnotes. You might then attach a style to this element so that they appear using a special set of style rules.
Example:

```
<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>
```

HOME | CONTACT | ABOUT

Content Articles

Actual content goes here.....

Span

The 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 element as follows
Example

```
<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>
```

This will produce 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 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:

```
<head>  <!-- Document Header Starts -->
        <title>This is document title</title>
</head> <!-- Document Header Ends -->
```

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

```
<body>
    <!--  This is valid comment -->
    <p>Document content goes here.....</p>
</body>
```

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.

```
<body>
    < !--  This is not a valid comment -->
    <p>Document content goes here.....</p>
</body>
```

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:

```
<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 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> or (<image>) 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 about_me.html and image file my_photo.jpg in the images directory:

```
<body>
    <p>Simple Image Insert</p>
    <img src="/images/ my_photo.jpg" alt="Test Image" />
</body>
```

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:

```
</body>
        <p>Setting image Border</p>
        < img src="/images/ my_photo.jpg" alt="Test Image" border="3"/>
</body>
```

Special Characters in HTML

Special Characters in HTML			
left single quote		‘	‘
right single quote		’	’
single low-9 quote		‚	‚
left double quote		“	“
right double quote		”	”
double low-9 quote		„	„
dagger		†	†
double dagger		‡	‡
per mill sign		‰	‰
single left-pointing angle quote		‹	‹
single right-pointing angle quote		›	›
black spade suit		♠	♠
black club suit		♣	♣
black heart suit		♥	♥
black diamond suit		♦	♦
overline, = spacing overscore		‾	‾
leftward arrow		←	←
upward arrow		↑	↑
rightward arrow		→	→
downward arrow		↓	↓
trademark sign	™	™	™
unused	&#00;- &#08;		
horizontal tab			
line feed
unused	&#11;		
space			
exclamation mark	!		!
double quotation mark	"	"	"
number sign	#		#
dollar sign	$		$
percent sign	%		%
ampersand	&	&	&
apostrophe	'		'
left parenthesis	(		(
right parenthesis	)		)
asterisk	*		*
plus sign	+		+
comma	,		,
hyphen	-		-
period	.		.
slash	/	⁄	/
digits 0-9	0- 9		
colon	:		:
semicolon	;		;

less-than sign	&#60;	&lt;	<
equals sign	&#61;		=
greater-than sign	&#62;	&gt;	>
question mark	&#63;		**?**
at sign	&#64;		**@**
uppercase letters A-Z	A- Z		
left square bracket	&#91;		**[**
backslash	&#92;		\
right square bracket	&#93;		**]**
caret	&#94;		^
horizontal bar (underscore)	&#95;		_
grave accent	&#96;		`
lowercase letters a-z	a- z		
left curly brace	&#123;		**{**
vertical bar	&#124;		|
right curly brace	&#125;		**}**
tilde	&#126;		~
ellipses	&#133;	&hellip;	…
en dash	&#150;	&ndash;	–
em dash	&#151;	&mdash;	—
unused	˜- Ÿ		
nonbreaking space			
inverted exclamation	&#161;	&iexcl;	¡
cent sign	&#162;	&cent;	¢
pound sterling	&#163;	&pound;	£
general currency sign	&#164;	&curren;	¤
yen sign	&#165;	&yen;	¥
broken vertical bar	&#166;	¦ or &brkbar;	¦
section sign	&#167;	&sect;	§
umlaut	&#168;	¨ or ¨	¨
copyright	&#169;	&copy;	©
feminine ordinal	&#170;	&ordf;	ª
left angle quote	&#171;	&laquo;	«
not sign	&#172;	&not;	¬
soft hyphen	&#173;	&shy;	
registered trademark	&#174;	&reg;	®
macron accent	&#175;	¯ or &hibar;	¯
degree sign	&#176;	&deg;	°
plus or minus	&#177;	&plusmn;	±
superscript two	&#178;	&sup2;	²
superscript three	&#179;	&sup3;	³
acute accent	&#180;	&acute;	´
micro sign	&#181;	&micro;	µ
paragraph sign	&#182;	&para;	¶
middle dot	&#183;	&middot;	·

cedilla	¸	¸	¸
superscript one	¹	¹	¹
masculine ordinal	º	º	º
right angle quote	»	»	»
one-fourth	¼	¼	¼
one-half	½	½	½
three-fourths	¾	¾	¾
inverted question mark	¿	¿	¿
uppercase A, grave accent	À	À	À
uppercase A, acute accent	Á	Á	Á
uppercase A, circumflex accent	Â	Â	Â
uppercase A, tilde	Ã	Ã	Ã
uppercase A, umlaut	Ä	Ä	Ä
uppercase A, ring	Å	Å	Å
uppercase AE	Æ	Æ	Æ
uppercase C, cedilla	Ç	Ç	Ç
uppercase E, grave accent	È	È	È
uppercase E, acute accent	É	É	É
uppercase E, circumflex accent	Ê	Ê	Ê
uppercase E, umlaut	Ë	Ë	Ë
uppercase I, grave accent	Ì	Ì	Ì
uppercase I, acute accent	Í	Í	Í
uppercase I, circumflex accent	Î	Î	Î
uppercase I, umlaut	Ï	Ï	Ï
uppercase Eth, Icelandic	Ð	Ð	Ð
uppercase N, tilde	Ñ	Ñ	Ñ
uppercase O, grave accent	Ò	Ò	Ò
uppercase O, acute accent	Ó	Ó	Ó
uppercase O, circumflex accent	Ô	Ô	Ô
uppercase O, tilde	Õ	Õ	Õ
uppercase O, umlaut	Ö	Ö	Ö
multiplication sign	×	×	×
uppercase O, slash	Ø	Ø	Ø
uppercase U, grave accent	Ù	Ù	Ù
uppercase U, acute accent	Ú	Ú	Ú
uppercase U, circumflex accent	Û	Û	Û
uppercase U, umlaut	Ü	Ü	Ü
uppercase Y, acute accent	Ý	Ý	Ý
uppercase THORN, Icelandic	Þ	Þ	Þ
lowercase sharps, German	ß	ß	ß
lowercase a, grave accent	à	à	à
lowercase a, acute accent	á	á	á
lowercase a, circumflex accent	â	â	â
lowercase a, tilde	ã	ã	ã
lowercase a, umlaut	ä	ä	ä
lowercase a, ring	å	å	å
lowercase ae	æ	æ	æ
lowercase c, cedilla	ç	ç	ç
lowercase e, grave accent	è	è	è
lowercase e, acute accent	é	é	é

lowercase e, circumflex accent	&#234;	&ecirc;	**ê**
lowercase e, umlaut	&#235;	&euml;	**ë**
lowercase i, grave accent	&#236;	&igrave;	**ì**
lowercase i, acute accent	&#237;	&iacute;	**í**
lowercase i, circumflex accent	&#238;	&icirc;	**î**
lowercase i, umlaut	&#239;	&iuml;	**ï**
lowercase eth, Icelandic	&#240;	&eth;	**ð**
lowercase n, tilde	&#241;	&ntilde;	**ñ**
lowercase o, grave accent	&#242;	&ograve;	**ò**
lowercase o, acute accent	&#243;	&oacute;	**ó**
lowercase o, circumflex accent	&#244;	&ocirc;	**ô**
lowercase o, tilde	&#245;	&otilde;	**õ**
lowercase o, umlaut	&#246;	&ouml;	**ö**
division sign	&#247;	&divide;	**÷**
lowercase o, slash	&#248;	&oslash;	**ø**
lowercase u, grave accent	&#249;	&ugrave;	**ù**
lowercase u, acute accent	&#250;	&uacute;	**ú**
lowercase u, circumflex accent	&#251;	&ucirc;	**û**
lowercase u, umlaut	&#252;	&uuml;	**ü**
lowercase y, acute accent	&#253;	&yacute;	**ý**
lowercase thorn, Icelandic	&#254;	&thorn;	**þ**
lowercase y, umlaut	&#255;	&yuml;	**ÿ**
Alpha	&Alpha;		**Α**
alpha	&alpha;		**α**
Beta	&Beta;		**Β**
beta	&beta;		**β**
Gamma	&Gamma;		**Γ**
gamma	&gamma;		**γ**
Delta	&Delta;		**Δ**
delta	&delta;		**δ**
Epsilon	&Epsilon;		**Ε**
epsilon	&epsilon;		**ε**
Zeta	&Zeta;		**Ζ**
zeta	&zeta;		**ζ**
Eta	&Eta;		**Η**
eta	&eta;		**η**
Theta	&Theta;		**Θ**
theta	&theta;		**θ**
Iota	&Iota;		**Ι**
iota	&iota;		**ι**
Kappa	&Kappa;		**Κ**
kappa	&kappa;		**κ**
Lambda	&Lambda;		**Λ**
lambda	&lambda;		**λ**
Mu	&Mu;		**Μ**
mu	&mu;		**μ**
Nu	&Nu;		**Ν**
nu	&nu;		**ν**
Xi	&Xi;		**Ξ**
xi	&xi;		**ξ**

Omicron	Ο		**Ο**
omicron	ο		**ο**
Pi	Π		**Π**
pi	π		**π**
Rho	Ρ		**Ρ**
rho	ρ		**ρ**
Sigma	Σ		**Σ**
sigma	σ		**σ**
Tau	Τ		**Τ**
tau	τ		**τ**
Upsilon	Υ		**Υ**
upsilon	υ		**υ**
Phi	Φ		**Φ**
phi	φ		**φ**
Chi	Χ		**Χ**
chi	χ		**χ**
Psi	Ψ		**Ψ**
psi	ψ		**ψ**
Omega	Ω		**Ω**
omega	ω		**ω**
password dot	●		●
bullet	•		•

Color codes

http://html-color-codes.info/