UNIT 2:

Programming Concepts: Introduction to HTML, Table Handling in HTML, Creating Forms, and Techniques involved in building a local web page. Client side programming using Java script. Introduction to Client server programming. Server Side Programming concepts.

1. Overview

We consider there are two types of web sites: **static web sites and dynamic web sites**.

Static web sites: web sites that are performed solely for the languages (X) HTML and CSS. They work very well but their contents cannot be updated automatically: it requires that the web site owner (webmaster) modifies the source code for adding new content. It's not very practical when you should update his web site several times in one day! Static web sites are well suited for making web sites "window" to present such a business, but without going further. This type of web site is increasingly rare today, because as soon as we add an element of interaction (like a contact form), we no longer speak of static web site but dynamic web site.

Dynamic web sites: more complex, they use other languages in addition to (X) HTML and CSS, such as PHP and MySQL. The content of these web sites is called "**dynamic**" because it can change without the intervention of the webmaster! Most web sites you visit today are dynamic web sites. The only prerequisite for learning to create this type of web site is already known to make static web sites in XHTML and CSS.

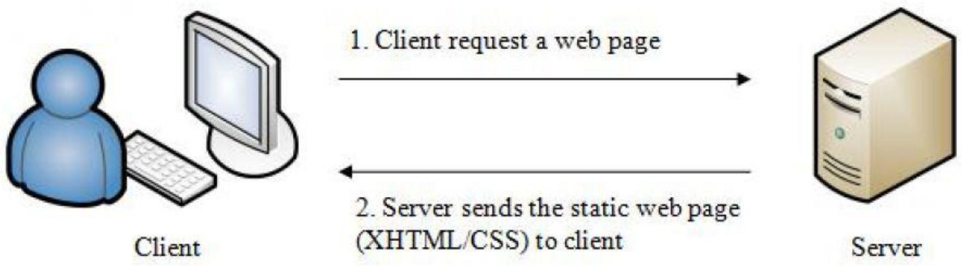
2. How does a web site work?

2.1: For a static web site

When the web site is static, the pattern is very simple. This happens in two stages:

* The client asks the server to see a web page.

* The server responds by sending the requested page.


1. Client request a web page
2. Server sends the static web page
(XHTML/CSS) to client
Client
Server

In the case of a static web site, the server stores web pages and sends back the entire web page

(Without any change in its initial content) to the client who is requesting the page them without Changing them.

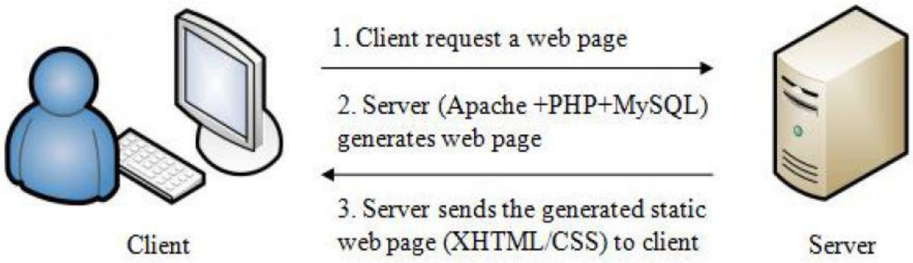
2.2: For a dynamic web site with PHP and MySQL

When the web site is dynamic, there is an intermediate step: the page is generated.

* The client asks the server to see a web page.

* The server prepares the page specifically for the client's request.

* The server sends the page it has generated.


1. Client request a web page
2. Server (Apache +PHP+MySQL) generates web page
3. Server sends the generated static web page (XHTML/CSS) to client
Client
Server

The web page is generated each time a client (web browser) demands it. This is precisely what makes dynamic web sites alive: the content of a page can change from moment to moment.

This is how some web sites manage to display some information such as your username on all pages. Since the server generates a page every time the client requests it, the can customize it to suit the tastes and preferences of the client (such as his real name).

2.3: Programming languages for web sites.

1) For a **static web site**: XHTML and CSS

Many languages have been created to produce web sites. Two of them are an essential basis for all webmasters:

- ✓ **XHTML** is the language at the base of web sites. It is very similar to HTML but imposes some rules a little stricter. Wherever possible, I recommend using XHTML rather than HTML because it forces you to treat the source code of your web site. XHTML is a simple language to learn who operates out of tags.

An example of XHTML code:

<p>This is a snapshot of a <em>paragraph</em> of text!</p>

- ✓ **CSS** is the language formatting of web sites. While XHTML is used to write the contents of your web pages and the structure, the CSS handles the formatting and layout. This CSS is that we choose the particular color, size, menus and much more.

Here is a CSS code:

```
div.banner {
text-align: center;
font-weight: bold;
font-size: 120%;
}
```

2) **For a dynamic web site:** We add **PHP and MySQL**

3. What programs do we need?

Depending on whether one creates a static web site or a dynamic web site, you need different Software. In fact, making a dynamic web site for us unfortunately requires a bit more software!

3.1: With a static web site

The webmasters who create static web sites with XHTML and CSS are lucky; they usually already have all the programs they need:

- ➢ **A text editor**: in theory a program such as Notepad shipped with Windows only, although it is Recommended to use a bit more advanced tool like Notepad + +.
- ➢ **A web browser**: it can test the web page. You can use eg Mozilla Firefox, Internet Explorer, Google Chrome, Opera, Safari, or any other browser you are accustomed to go on the web. It is advisable to regularly test your web site on different browsers.
- ➢ **A web server** (such as Apache): This is responsible for hosting and issuing web pages to visitors. However, Apache only handles static web sites (it can only handle HTML pages). However, for those who work on dynamic web sites, these tools are not enough. It is necessary to install additional programs.

4. Introduction to HTML

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

A Simple HTML Document

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

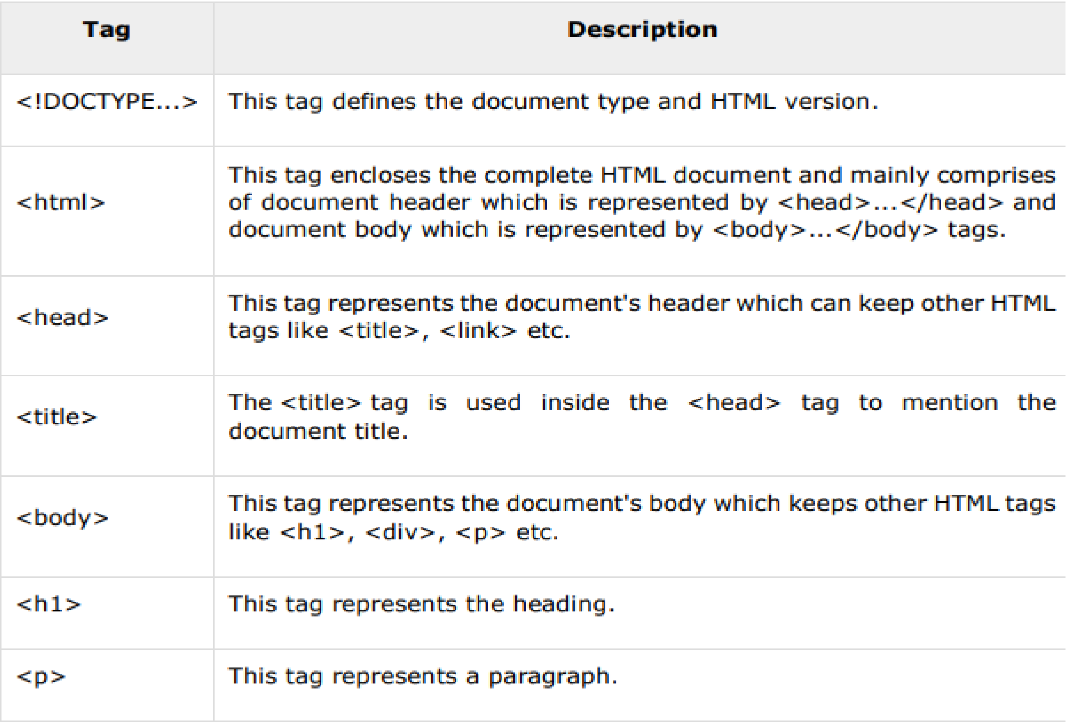
Example Explained

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document
- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

4.1: HTML Tags

HTML tags are element names surrounded by angle brackets:
<tagname>content goes here...</tagname>

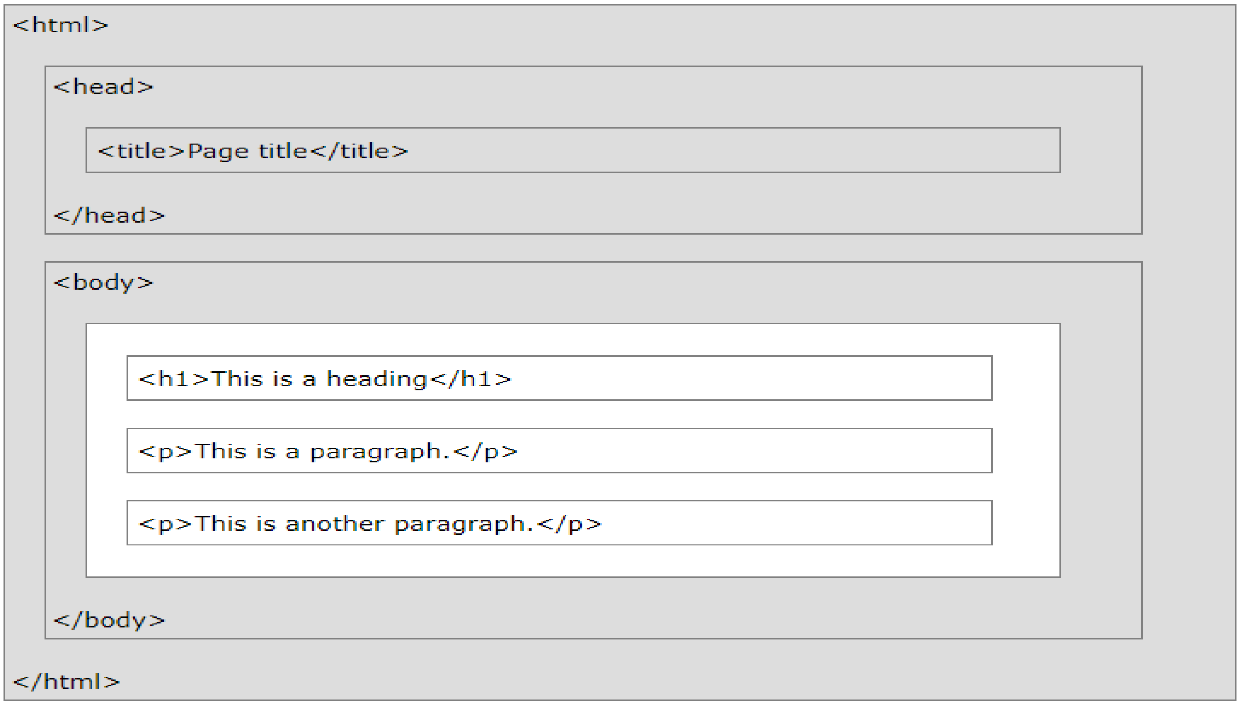
- HTML tags normally come **in pairs** like <p> and </p>
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name



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.

4.2: HTML Page Structure

Below is a visualization of an HTML page structure:



```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
      <p>This is another paragraph.</p>
  </body>
</html>
```

Note: Only the content inside the <body> section (the white area above) is displayed in a browser

4.3: HTML Editors

Web pages can be created and modified by using professional HTML editors. However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac). We believe using a simple text editor is a good way to learn HTML.

Follow the four steps below to create your first web page with Notepad or TextEdit.

Step 1: Open Notepad (PC)

Windows 8 or later:

Open the **Start Screen** (the window symbol at the bottom left on your screen). Type **Notepad**.

Windows 7 or earlier:

Open **Start** > **Programs** > **Accessories** > **Notepad**

Step 1: Open TextEdit (Mac)

Open **Finder** > **Applications** > **TextEdit**

Also change some preferences to get the application to save files correctly. In **Preferences** > **Format** > choose **"Plain Text"**

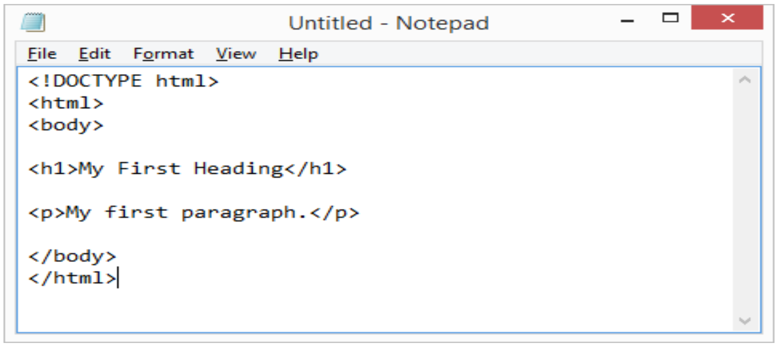
Then under "Open and Save", check the box that says "Ignore rich text commands in HTML files".

Then open a new document to place the code.

Step 2: Write Some HTML

Write or copy some HTML into Notepad.

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

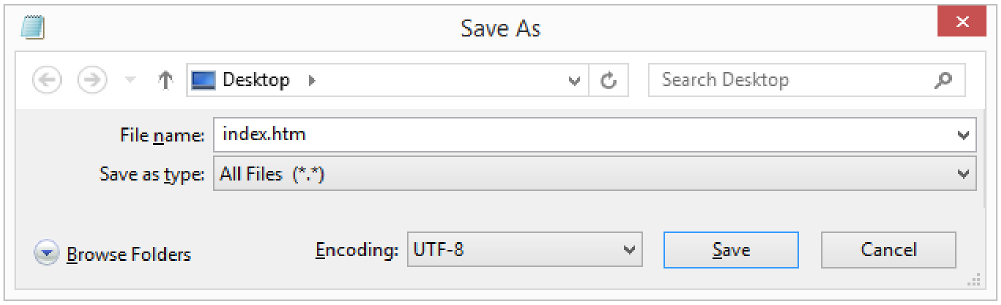


```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Step 3: Save the HTML Page

Save the file on your computer. Select File > Save as in the Notepad menu.

Name the file "index.htm" and set the encoding to UTF-8 (which is the preferred encoding for HTML files).



File name: index.htm

Save as type: All Files (*.*)

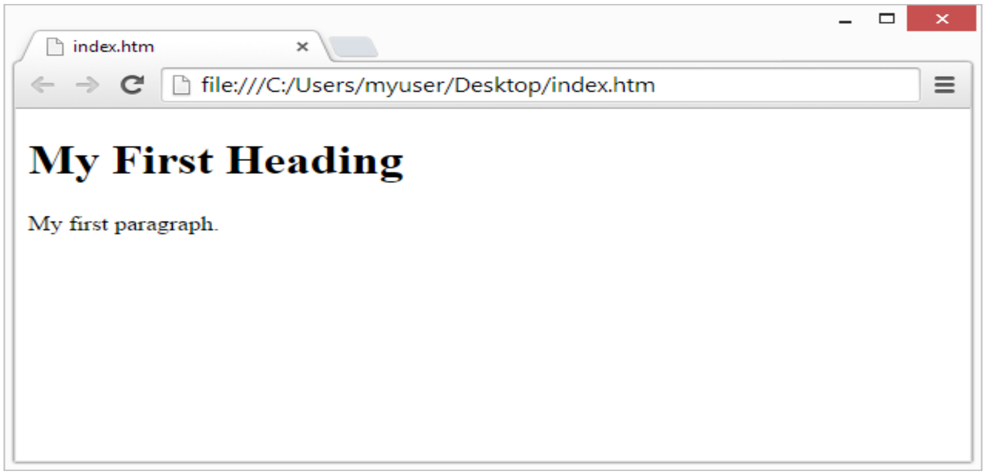
Encoding: UTF-8

You can use either .htm or .html as file extension. There is no difference, it is up to you.

Step 4: View the HTML Page in Your Browser

Open the saved HTML file in your favorite browser (double clicks on the file, or right-click - and choose "Open with").

The result will look much like this:



file:///C:/Users/myuser/Desktop/index.htm

My First Heading

My first paragraph.

4.4: HTML Basic

4.4.1: HTML Documents

All HTML documents must start with a document type declaration: <**!DOCTYPE html**>.

The HTML document itself begins with <**html**> and ends with </**html**>.

The visible part of the HTML document is between <**body**> and </**body**>.

Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

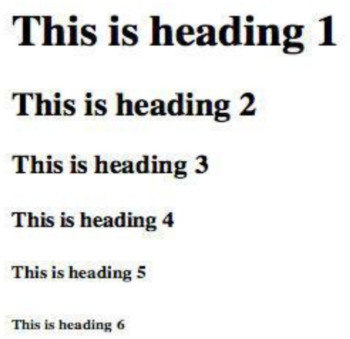
4.4.2: HTML Headings

HTML headings are defined with the <**h1**> to <**h6**> tags. <h1> defines the most important heading. <h6> defines the least important heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Output



This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

4.4.3: HTML Paragraphs

HTML paragraphs are defined with the <**p**> tag:

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

4.4.4: HTML Links

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

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

The link's destination is specified in the **href attribute**.
Attributes are used to provide additional information about HTML elements.

4.4.5: HTML Images

HTML images are defined with the **<img>** tag.
The source file (src), alternative text (alt), width, and height are provided as attributes:

Example

```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142
">
```

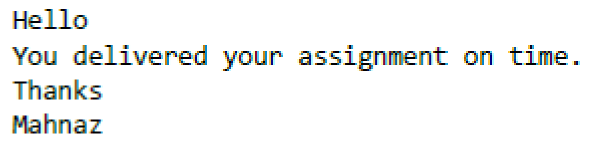
4.4.6: Line Break Tag

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
```

4.4.7: Centering Content

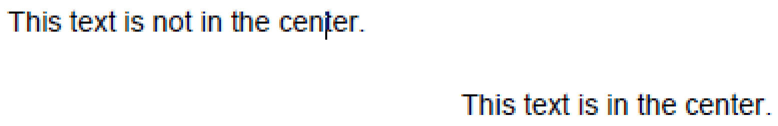
You can use **<center>** tag to put any content in the center of the page or any table cell.

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.

4.4.8: 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.

4.4.9: Nonbreaking Spaces

In HTML coding, the non-breaking space is a character entity which can:

- Create white space between words or web page elements
- Stop the browser from breaking a line in the wrong place.

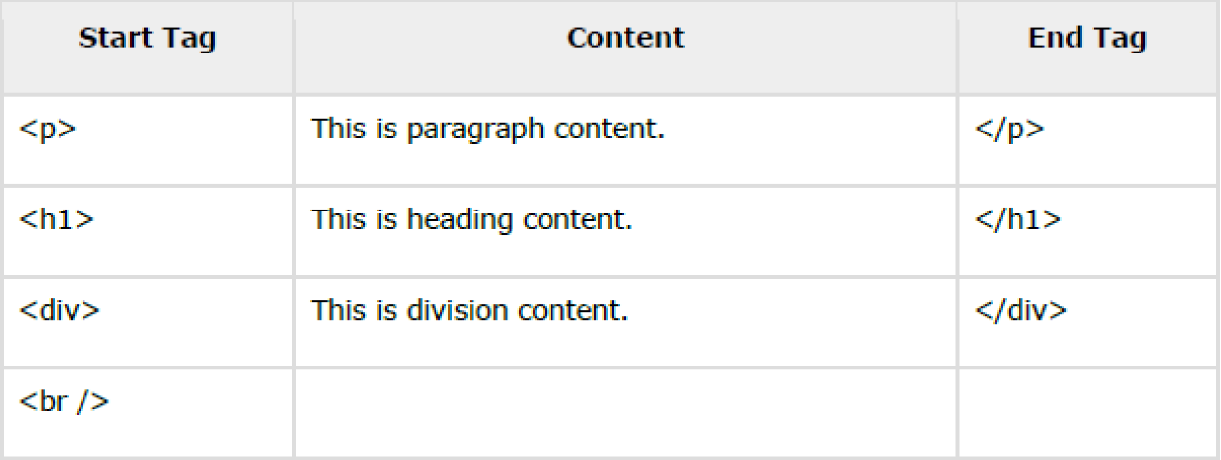
To insert a non-breaking space you would use: **&nbsp;**

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>
```

4.5: HTML Elements

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



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

So here **<p>....</p>** is an HTML element, **<h1>...</h1>** is another HTML element. There are some HTML elements which don't need to be closed, such as **<img.../>**, **<hr** /> and **<br** /> elements. These are known as **void elements**. HTML documents consist of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

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

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

Nested HTML Elements

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

Example

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

This will display the following result:

This is *italic* heading

This is <u>underlined</u> paragraph

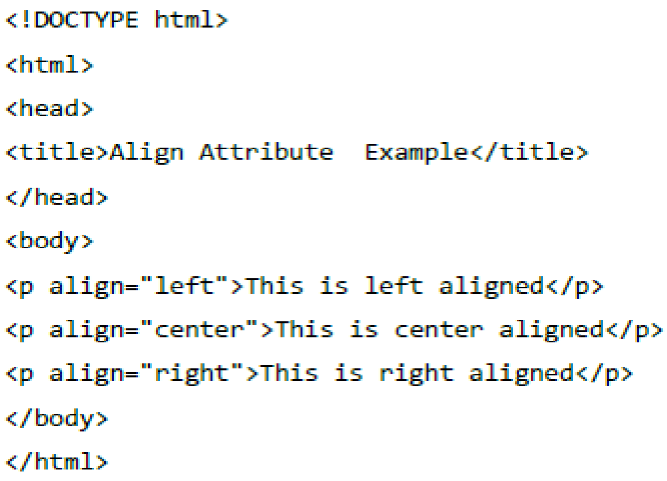
4.6: HTML – ATTRIBUTES

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**.

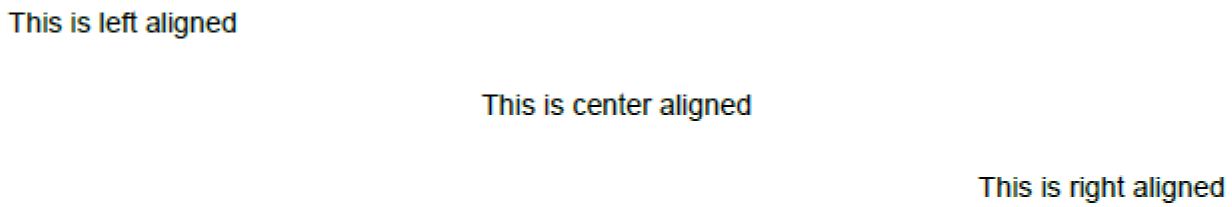
Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

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

The lang Attribute

The language of the document can be declared in the **<html>** tag.

The language is declared with the **lang** attribute.

Declaring a language is important for accessibility applications (screen readers) and search engines:

```
<!DOCTYPE html>
<html lang="en-US">
<body>
...
</body>
</html>
```

The first two letters specify the language (en). If there is a dialect, use two more letters (US).

The title Attribute

Here, a **title** attribute is added to the **<p>** element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:

Example

```
<p title="I'm a tooltip">
This is a paragraph.
</p>
```

The href Attribute

HTML links are defined with the **<a>** tag. The link address is specified in the href attribute:

Example

```
<a href="https://www.w3schools.com">This is a link</a>
```

Size Attributes

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

The filename of the source (**src**), and the size of the image (**width** and **height**) are all provided as **attributes**:

Example

```
<img src="w3schools.jpg" width="104" height="142">
```

The image size is specified in pixels: width="104" means 104 screen pixels wide.

The alt Attribute

The **alt** attribute specifies an alternative text to be used, when an image cannot be displayed.

The value of the attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a blind person, can "hear" the element.

Example

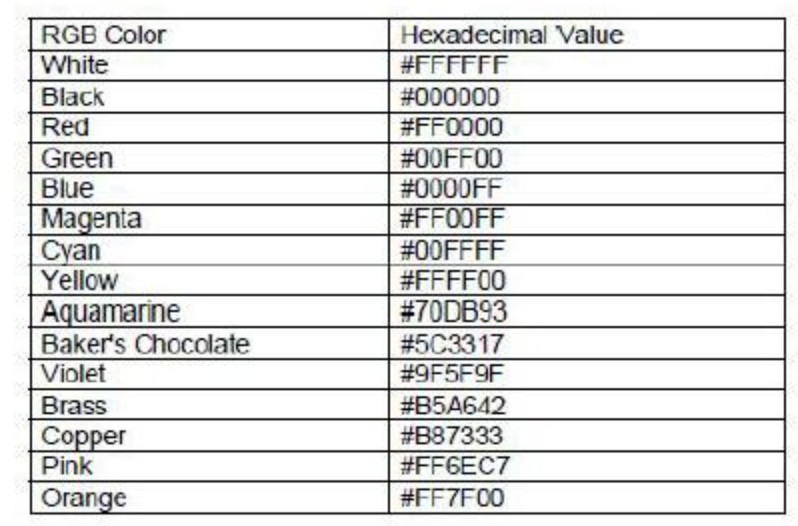
```
<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">
```

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.

Note:



RGB Color	Hexadecimal Value
White	#FFFFFF
Black	#000000
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Magenta	#FF00FF
Cyan	#00FFFF
Yellow	#FFFF00
Aquamarine	#70DB93
Baker's Chocolate	#5C3317
Violet	#9F5F9F
Brass	#B5A642
Copper	#B87333
Pink	#FF6EC7
Orange	#FF7F00

4.7: 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 an <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 an <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.

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.
```

Grouping Content

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

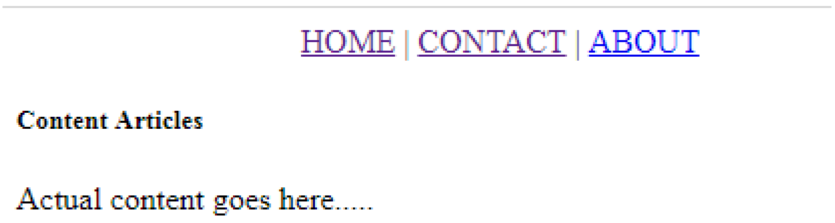
For example, you might want to put all of the footnotes on a page within a <div> element to indicate that all of the elements within that <div> element relate to the footnotes. You might then attach a style to this <div> element so that they appear using a special set of style rules.

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 | CONTACT | ABOUT

Content Articles

Actual content goes here.....

4.8: Comments in HTML

Comment is a piece of code which is ignored by any web browser. 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.....

4.9: HTML – IMAGES

In HTML, images are defined with the **<img>** tag. The <img> tag is empty, it contains attributes only, and does not have a closing tag. The src attribute specifies the URL (web address) of the image:

Example

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

```
</body>
</html>
```

The alt Attribute

The alt attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). If a browser cannot find an image, it will display the value of the alt attribute:

Example

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

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

Image Size - Width and Height

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

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

Example

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

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

Example

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

Width and Height, or Style?

Both the width, height, and style attributes are valid in HTML5.
However, we suggest using the style attribute. It prevents internal or external styles sheets from changing the original size of images:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width:100%;
}
</style>
```

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

Images in Another Folder

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

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

Example

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

Images on Another Server

Some web sites store their images on image servers.

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

Example

```
<img src="https://www.w3schools.com/images/w3schools_green.jpg"alt="
W3Schools.com">
```

Animated Images

The GIF standard allows animated images:

Example

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

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

Using an Image as a Link

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

Example

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

Maps

Use the <map> tag to define an image-map. An image-map is an image with clickable areas. The name attribute of the <map> tag is associated with the <img>'s usemap attribute and creates a relationship between the image and the map. The <map> tag contains a number of <area> tags, that defines the clickable areas in the image-map:

Example

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

Background Image

To add a background image on an HTML element, use the CSS property background-image:

Example

To add a background image on a web page, specify the background-image property on the BODY element:

```
<body style="background-image:url('skies.jpg')">
<h2>Background Image</h2>
</body>
```

Example

To add a background image on a paragraph, specify the background-image property on the P element:

```
<body>
<p style="background-image:url('skies.jpg')">
...
</p>
</body>
```

4.10: HTML Tables

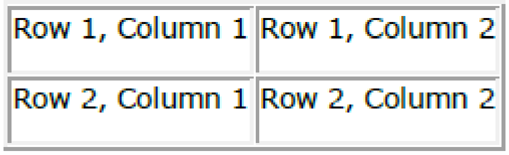
An HTML table is defined with the **<table>** tag. Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag. By default, table headings are bold and centered. A table data/cell is defined with the **<td>** tag.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Tables</title>
</head>
```

```
<body>
<table border="1">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

This will produce the following result:



Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

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

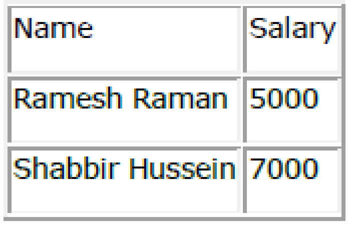
Table Heading

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Header</title>
</head>
<body>
<table border="1">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```

This will produce the following result:



Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

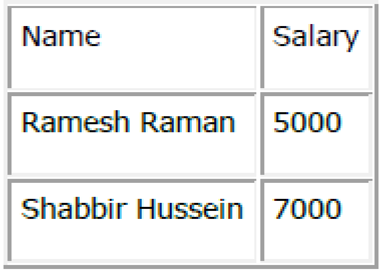
Cellpadding and Cellspacing Attributes

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Cellpadding</title>
</head>
<body>
<table border="1" cellpadding="5" cellspacing="5">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</body>
</html>
```

This will produce the following result:



Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

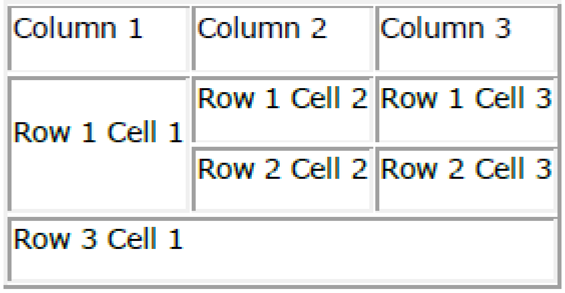
Colspan and Rowspan Attributes

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Colspan/Rowspan</title>
</head>
<body>
<table border="1">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
</body>
</html>
```

This will produce the following result:



Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

Tables Backgrounds

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

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

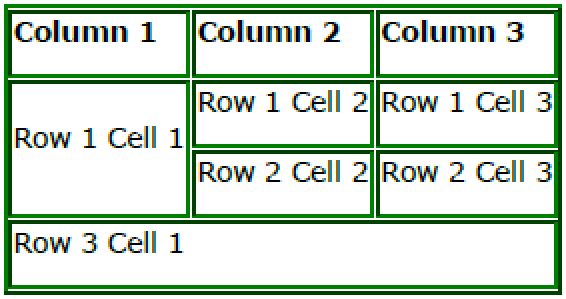
You can also set border color also using **bordercolor** attribute.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Background</title>
</head>
<body>
<table border="1" bordercolor="green" bgcolor="yellow">
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
</tr>
<tr><td rowspan="2">Row 1 Cell 1</td><td>Row 1 Cell 2</td><td>Row 1 Cell 3</td></tr>
<tr><td>Row 2 Cell 2</td><td>Row 2 Cell 3</td></tr>
<tr><td colspan="3">Row 3 Cell 1</td></tr>
</table>
```

```
</body>
</html>
```

This will produce the following result:



Column 1	Column 2	Column 3
Row 1 Cell 1	Row 1 Cell 2	Row 1 Cell 3
	Row 2 Cell 2	Row 2 Cell 3
Row 3 Cell 1		

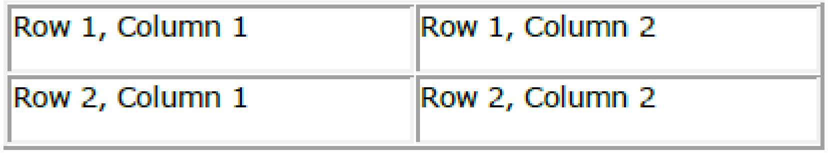
Table Height and Width

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Width/Height</title>
</head>
<body>
<table border="1" width="400" height="150">
<tr>
<td>Row 1, Column 1</td>
<td>Row 1, Column 2</td>
</tr>
<tr>
<td>Row 2, Column 1</td>
<td>Row 2, Column 2</td>
</tr>
</table>
</body>
</html>
```

This will produce the following result:



Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

Table Caption

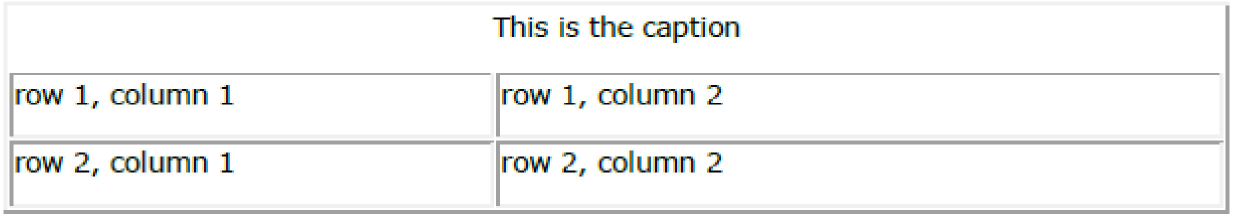
The **caption** tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table Caption</title>
</head>
<body>
```

```
<table border="1" width="100%">
<caption>This is the caption</caption>
<tr>
<td>row 1, column 1</td><td>row 1, column 2</td>
</tr>
<tr>
<td>row 2, column 1</td><td>row 2, column 2</td>
</tr>
</table>
</body>
</html>
```

This will produce the following result:



This is the caption

row 1, column 1	row 1, column 2
row 2, column 1	row 2, column 2

Nested Tables

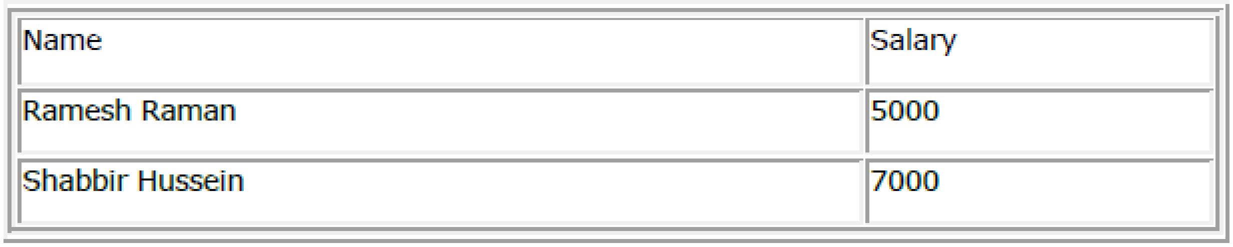
You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

Example

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

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Table</title>
</head>
<body>
<table border="1" width="100%">
<tr>
<td>
<table border="1" width="100%">
<tr>
<th>Name</th>
<th>Salary</th>
</tr>
<tr>
<td>Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

This will produce the following result:



Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

4.11: HTML – LISTS

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

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

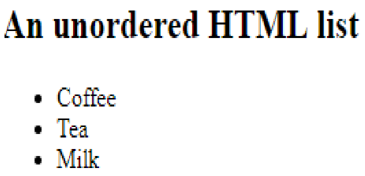
Unordered HTML List

An unordered list starts with the **<ul>** tag. Each list item starts with the **<li>** tag.

The list items will be marked with bullets (small black circles) by default:

Example

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

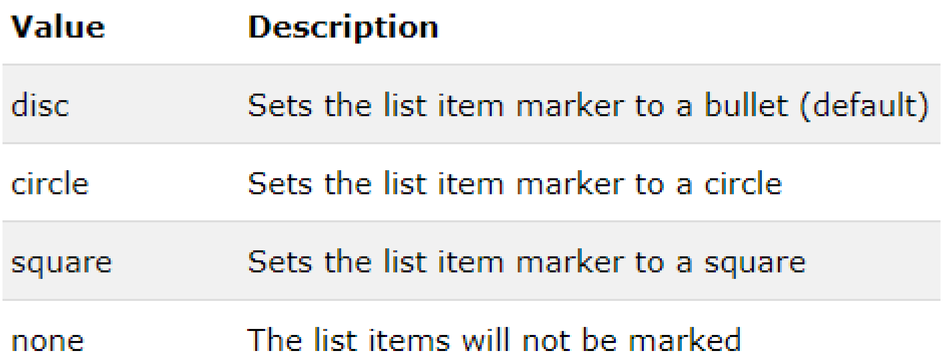


An unordered HTML list

- Coffee
- Tea
- Milk

Unordered HTML List - Choose List Item Marker

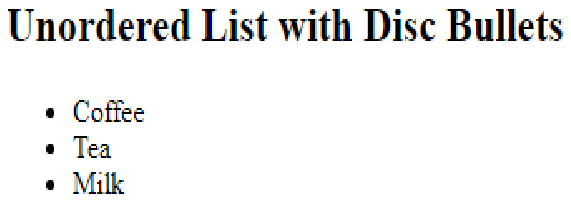
The CSS **list-style-type** property is used to define the style of the list item marker:



Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

Example - Disc

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

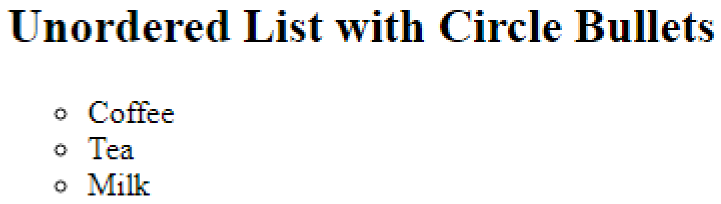


Unordered List with Disc Bullets

- Coffee
- Tea
- Milk

Example - Circle

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

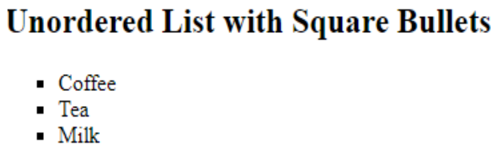


Unordered List with Circle Bullets

- Coffee
- Tea
- Milk

Example - Square

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

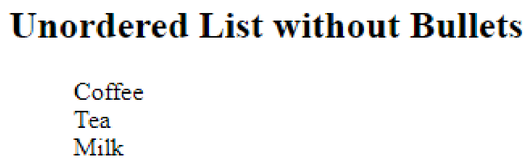


Unordered List with Square Bullets

- Coffee
- Tea
- Milk

Example - None

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



Unordered List without Bullets

Coffee
Tea
Milk

Ordered HTML List

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

The list items will be marked with numbers by default:

Example

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

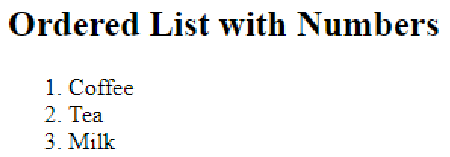
Ordered HTML List - The Type Attribute

The **type** attribute of the <ol> tag, defines the type of the list item marker:

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

Numbers:

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

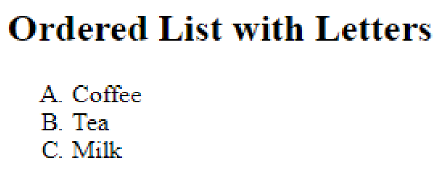


Ordered List with Numbers

1. Coffee
2. Tea
3. Milk

Uppercase Letters:

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

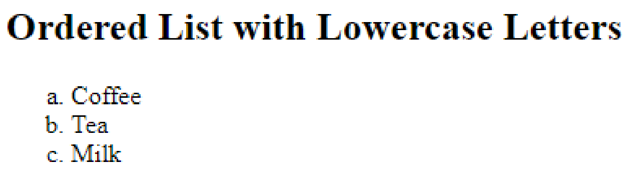


Ordered List with Letters

A. Coffee
B. Tea
C. Milk

Lowercase Letters:

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

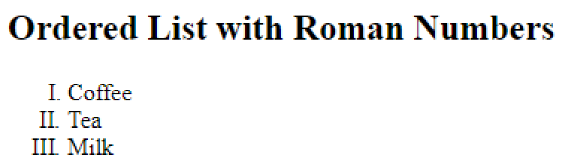


Ordered List with Lowercase Letters

a. Coffee
b. Tea
c. Milk

Uppercase Roman Numbers:

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

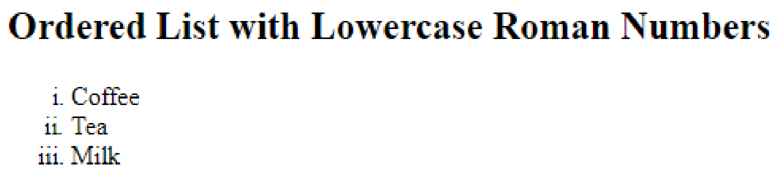


Ordered List with Roman Numbers

I. Coffee
II. Tea
III. Milk

Lowercase Roman Numbers:

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



Ordered List with Lowercase Roman Numbers

i. Coffee
ii. Tea
iii. Milk

HTML Description Lists

HTML also supports description lists. A description list is a list of terms, with a description of each term. The <**dl**> tag defines the description list, the <**dt**> tag defines the term (name), and the <**dd**> tag describes each term:

Example

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

Nested HTML Lists

List can be nested (lists inside lists):

Example

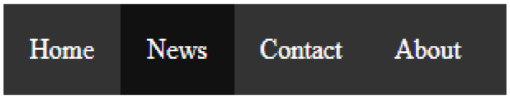
```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Horizontal Lists

HTML lists can be styled in many different ways with CSS. One popular way is to style a list horizontally, to create a menu:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333333;
}
li {
    float: left;
}
li a {
    display: block;
    color: white;
    text-align: center;
    padding: 16px;
    text-decoration: none;
}
li a:hover {
    background-color: #111111;
}
</style>
</head>
<body>
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```



4.12: HTML – TEXT LINKS

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks. Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

Linking Documents

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

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

Example

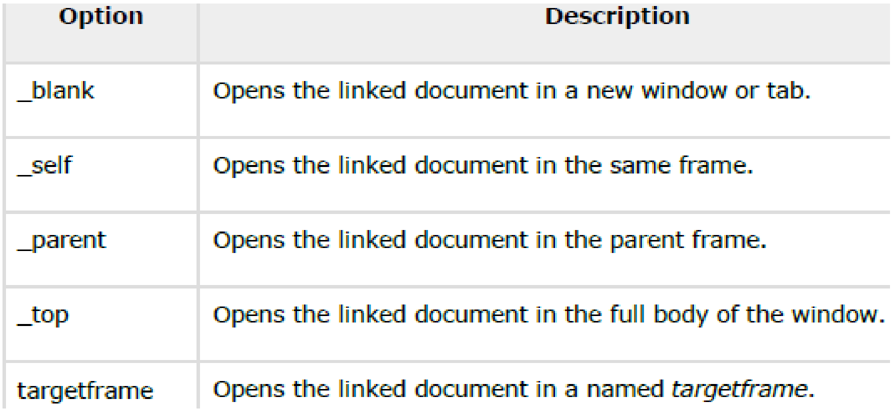
```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.tutorialspoint.com" target="_self">Tutorials Point</a>
</body>
</html>
```

This will produce the following result, where you can click on the link generated to reach to the home page of Tutorials Point (in this example).

Tutorials Point

The target Attribute

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



Option	Description
_blank	Opens the linked document in a new window or tab.
_self	Opens the linked document in the same frame.
_parent	Opens the linked document in the parent frame.
_top	Opens the linked document in the full body of the window.
targetframe	Opens the linked document in a named *targetframe*.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body>
<p>Click any of the following links</p>
<a href="/html/index.htm" target="_blank">Opens in New</a> |
<a href="/html/index.htm" target="_self">Opens in Self</a> |
<a href="/html/index.htm" target="_parent">Opens in Parent</a> |
<a href="/html/index.htm" target="_top">Opens in Body</a>
</body>
</html>
```

Opens in New | Opens in Self | Opens in Parent | Opens in Body

Use of Base Path

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

Example

Following example makes use of <base> tag to specify base URL and later we can use relative path to all the links instead of giving complete URL for every link.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body>
<p>Click following link</p>
<a href="/html/index.htm" target="_blank">HTML Tutorial</a>
</body>
</html>
```

This will produce the following result, where you can click on the link generated **HTML Tutorial** to reach to the HTML tutorial. Now given URL <a href="/html/index.htm" is being considered as <ahref="http://www.tutorialspoint.com/html/index.htm"

HTML Tutorial

Linking to a Page Section

You can create a link to a particular section of a given webpage by using **name** attribute. This is a two-step process.

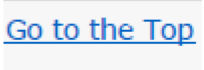
First create a link to the place where you want to reach with-in a webpage and name it using <a...> tag as follows:

<h1>HTML Text Links <a name="top"></a></h1>

Second step is to create a hyperlink to link the document and place where you want to reach:

<a href="/html/html_text_links.htm#top">Go to the Top</a>

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.



Go to the Top

Setting Link Colors

You can set colors of your links, active links and visited links using **link**, **alink** and **vlink** attributes of <body> tag.

Example

Save the following in test.htm and open it in any web browser to see how **link**, **alink** and **vlink** attributes work.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body alink="#54A250" link="#040404" vlink="#F40633">
<p>Click following link</p>
<a href="/html/index.htm" target="_blank" >HTML Tutorial</a>
</body>
</html>
```

This will produce the following result. Just check color of the link before clicking on it, next check its color when you activate it and when the link has been visited.

Click following link

HTML Tutorial

Download Links

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple; you just need to give complete URL of the downloadable file as follows:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
```

```
</head>
<body>
<a href="http://www.tutorialspoint.com/page.pdf">Download PDF File</a>
</body>
</html>
```

This will produce following link and will be used to download a file.

Download PDF File

HTML – IMAGE LINKS

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

Example

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below:

```
<!DOCTYPE html>
<html>
<head>
<title>Image Hyperlink Example</title>
</head>
<body>
<p>Click following link</p>
<a href="http://www.tutorialspoint.com" target="_self">
<img src="/images/logo.png" alt="Tutorials Point" border="0"/>
</a>
</body>
</html>
```

This will produce the following result, where you can click on the images to reach to the home page of Tutorials Point.

Click following link

Put the output here

4.13: HTML – FRAMES

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in your webpages:

- Some smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Sometimes your page will be displayed differently on different computers due to different screen resolution.
- The browser's *back button* might not work as the user hopes.

- There are still few browsers that do not support frame technology.

Creating Frames

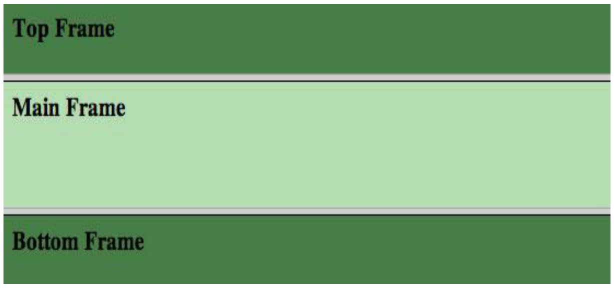
To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Example

Following is the example to create three horizontal frames:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset rows="10%,80%,10%">
<frame name="top" src="/html/top_frame.htm" />
<frame name="main" src="/html/main_frame.htm" />
<frame name="bottom" src="/html/bottom_frame.htm" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

This will produce the following result:



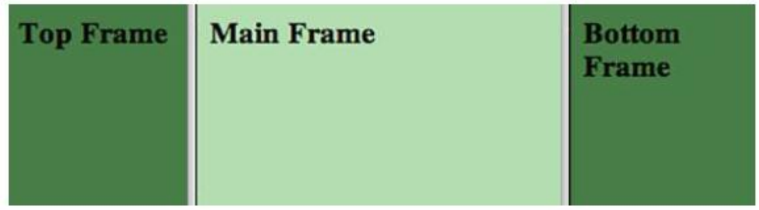
Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Frames</title>
</head>
<frameset cols="25%,50%,25%">
<frame name="left" src="/html/top_frame.htm" />
<frame name="center" src="/html/main_frame.htm" />
<frame name="right" src="/html/bottom_frame.htm" />
<noframes>
<body>
Your browser does not support frames.
</body>
```

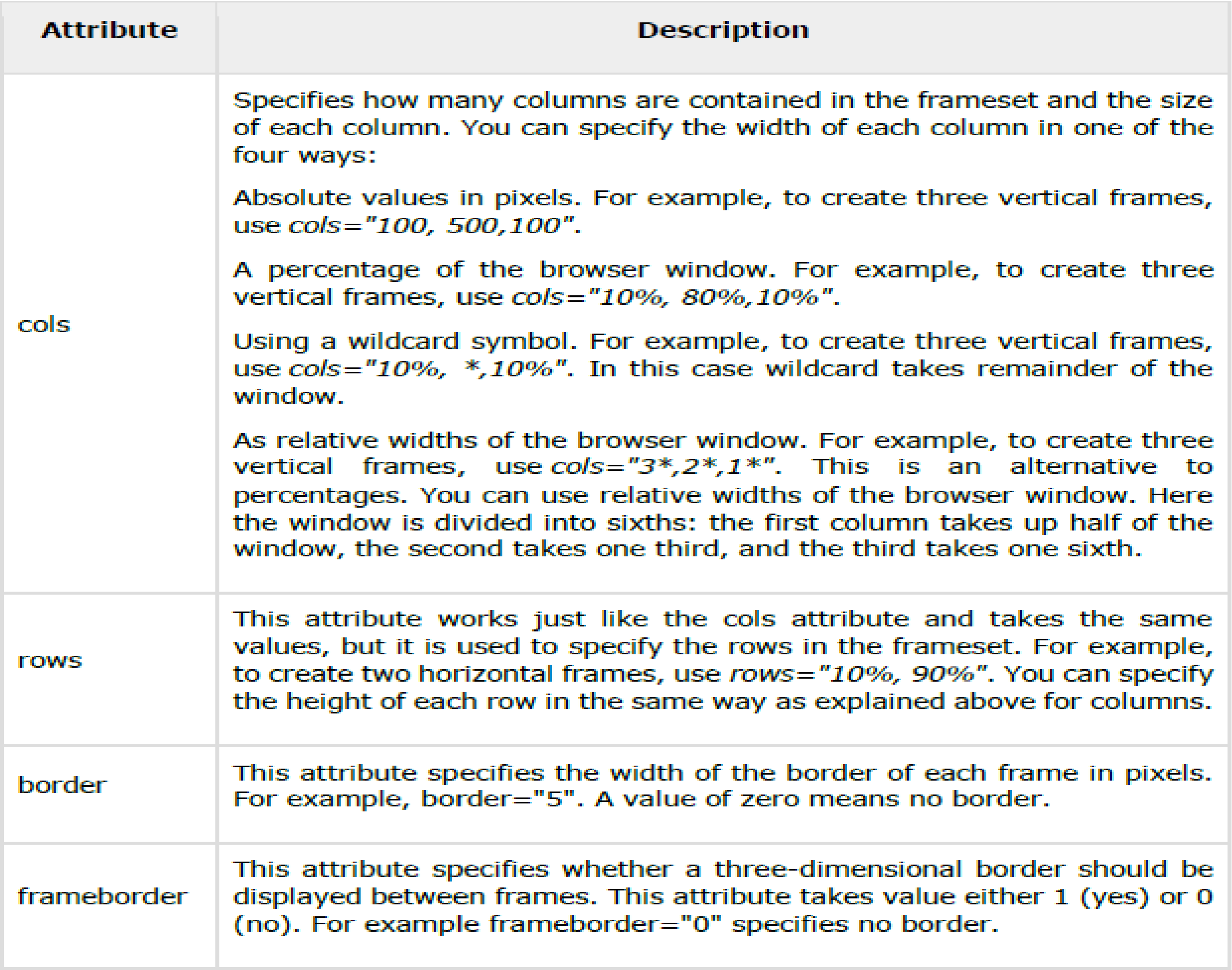
```
</noframes>
</frameset>
</html>
```

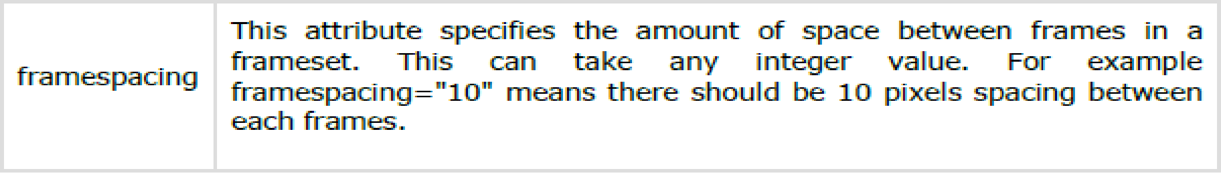
This will produce the following result:



The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag:

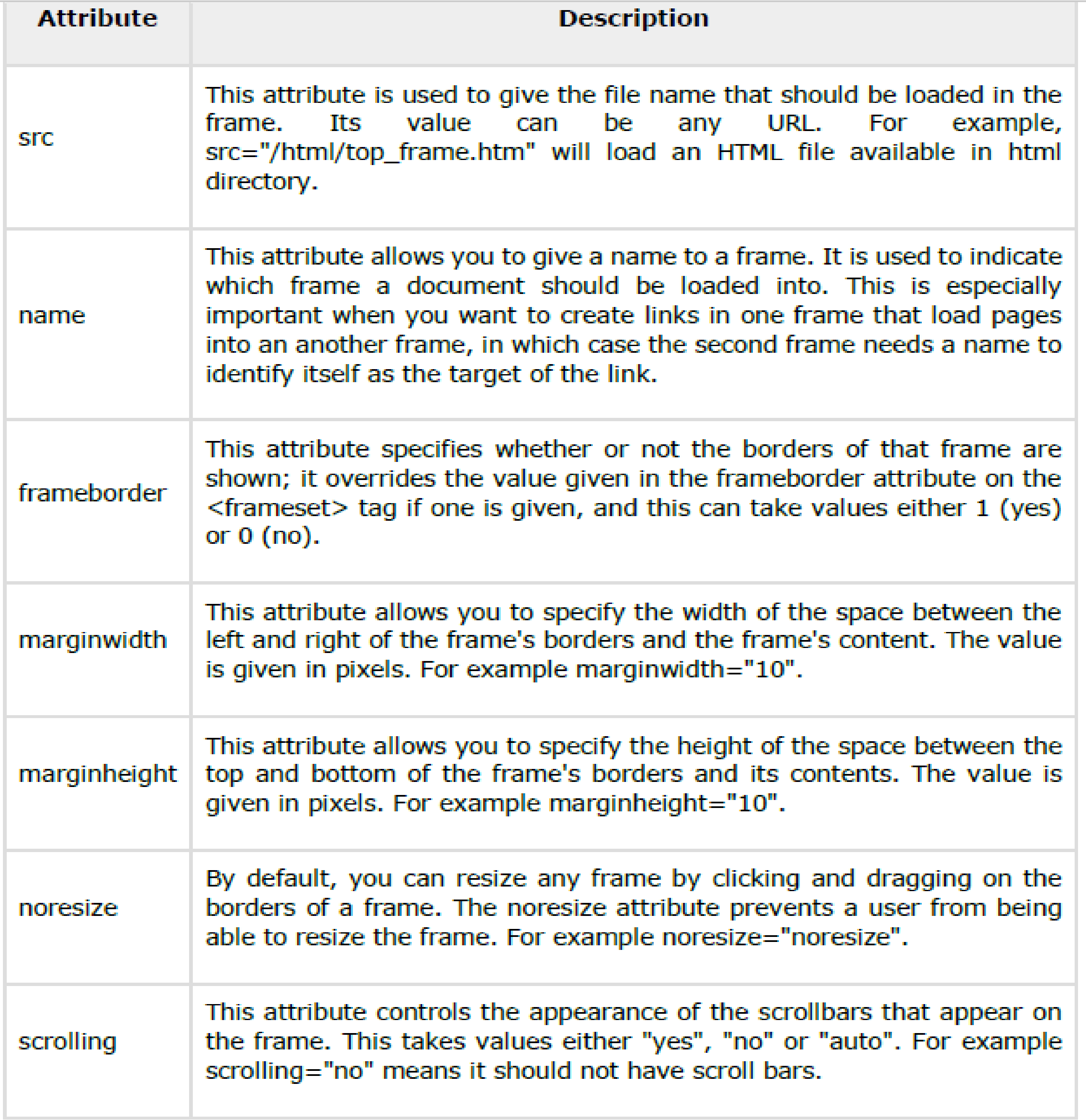
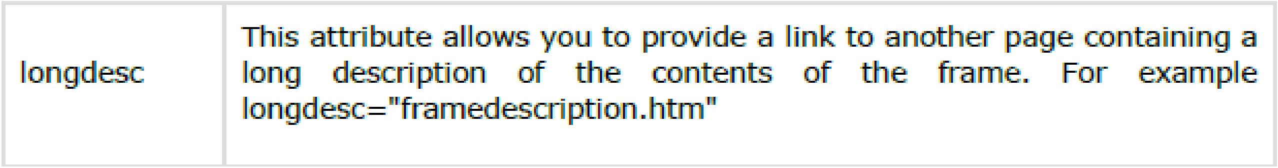




Attribute	Description
cols	Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways: Absolute values in pixels. For example, to create three vertical frames, use *cols="100, 500,100"*. A percentage of the browser window. For example, to create three vertical frames, use *cols="10%, 80%,10%"*. Using a wildcard symbol. For example, to create three vertical frames, use *cols="10%, *,10%"*. In this case wildcard takes remainder of the window. As relative widths of the browser window. For example, to create three vertical frames, use *cols="3*,2*,1*"*. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.
rows	This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use *rows="10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
border	This attribute specifies the width of the border of each frame in pixels. For example, border="5". A value of zero means no border.
frameborder	This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder="0" specifies no border.
framespacing	This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing="10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are the important attributes of <frame> tag:

Attribute	Description
src	This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src="/html/top_frame.htm" will load an HTML file available in html directory.
name	This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
frameborder	This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
marginwidth	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth="10".
marginheight	This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight="10".
noresize	By default, you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".
scrolling	This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.
longdesc	This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

Browser Support for Frames

If a user is using any old browser or any browser, which does not support frames then **<noframes>** element should be displayed to the user.

So you must place a <body> element inside the **<noframes>** element because the <frameset> element is supposed to replace the <body> element, but if a browser does not understand <frameset> element then it should understand what is inside the <body> element which is contained in a <noframes> element. .

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a test.htm file has following code:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Target Frames</title>
</head>
<frameset cols="200, *">
<frame src="/html/menu.htm" name="menu_page" />
<frame src="/html/main.htm" name="main_page" />
<noframes>
<body>
Your browser does not support frames.
</body>
</noframes>
</frameset>
</html>
```

Here, we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menu bar implemented by **menu.htm** file. The second column fills in remaining space and will contain the main part of the page and it is implemented by **main.htm** file. For all the three links available in menu bar, we have mentioned target frame as **main_page**, so whenever you click any of the links in menu bar, available link will open in main page.

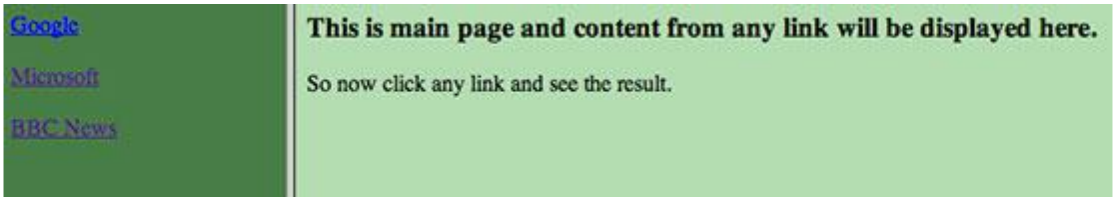
Following is the content of **menu.htm** file:

```
<!DOCTYPE html>
<html>
<body bgcolor="#4a7d49">
<a href="http://www.google.com" target="main_page">Google</a>
<br /><br />
<a href="http://www.microsoft.com" target="main_page">Microsoft</a>
<br /><br />
<a href="http://news.bbc.co.uk" target="main_page">BBC News</a>
</body>
</html>
```

Following is the content of main.htm file:

```
<!DOCTYPE html>
<html>
<body bgcolor="#b5dcb3">
<h3>This is main page and content from any link will be displayed here.</h3>
<p>So now click any link and see the result.</p>
</body>
</html>
```

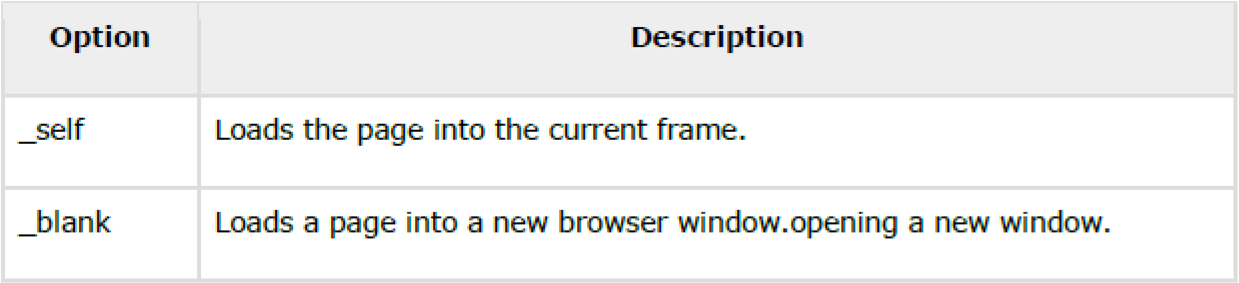
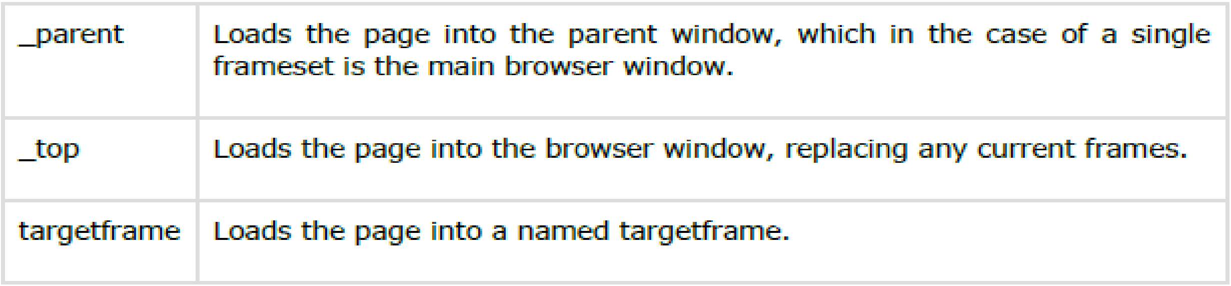
When we load **test.htm** file, it produces following result:



This is main page and content from any link will be displayed here.

So now click any link and see the result.

Now you can try to click links available in the left panel and see the result. The *target*attribute can also take one of the following values:

Option	Description
_self	Loads the page into the current frame.
_blank	Loads a page into a new browser window.opening a new window.
_parent	Loads the page into the parent window, which in the case of a single frameset is the main browser window.
_top	Loads the page into the browser window, replacing any current frames.
targetframe	Loads the page into a named targetframe.

The <Iframe> Tag Attributes

An iframe is used to display a web page within a web page.

Iframe Syntax

An HTML iframe is defined with the <**iframe**> tag:

```
<iframe src="URL"></iframe>
```

The **src** attribute specifies the URL (web address) of the inline frame page.

Iframe - Set Height and Width

Use the **height** and **width** attributes to specify the size of the iframe.

The attribute values are specified in pixels by default, but they can also be in percent (like "80%").

Example

```
<iframe src="demo_iframe.htm" height="200" width="300"></iframe>
```

Iframe - Remove the Border

By default, an iframe has a border around it. To remove the border, add the **style** attribute and use the CSS **border** property:

Example

```
<iframe src="demo_iframe.htm" style="border:none;"></iframe>
```

With CSS, you can also change the size, style and color of the iframe's border:

Example

```
<iframe src="demo_iframe.htm" style="border:2px solid grey;"></iframe>
```

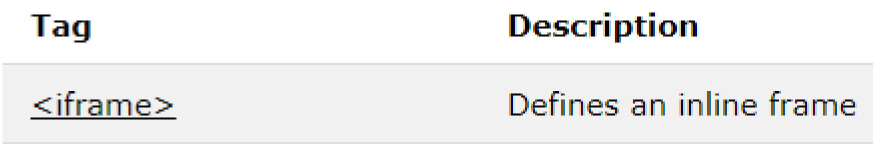
Iframe - Target for a Link

An iframe can be used as the target frame for a link. The **target** attribute of the link must refer to the **name** attribute of the iframe:

Example

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
<p><a href="https://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

HTML iframe Tag



Tag	Description
<iframe>	Defines an inline frame

4.14: HTML – BACKGROUNDS

- Html Background with Colors
- Html Background with Images

By default, your webpage background is white in color. You may not like it, but no worries. HTML provides you following two good ways to decorate your webpage background.

Html Background with Colors

The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds. Following is the syntax to use bgcolor attribute with any HTML tag.

```
<tagname bgcolor="color_value"...>
```

This color_value can be given in any of the following formats:

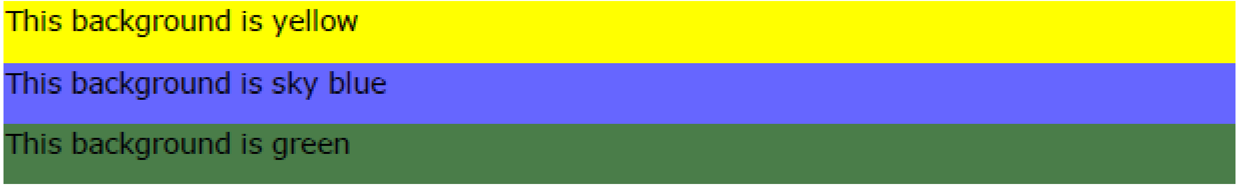
```
<!-- Format 1 - Use color name -->
<table bgcolor="lime" >
<!-- Format 2 - Use hex value -->
<table bgcolor="#f1f1f1" >
<!-- Format 3 - Use color value in RGB terms -->
<table bgcolor="rgb(0,0,120)" >
```

Example

Here are the examples to set background of an HTML tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Colors</title>
</head>
<body>
<!-- Format 1 - Use color name -->
<table bgcolor="yellow" width="100%">
<tr><td>
This background is yellow
</td></tr>
</table>
<!-- Format 2 - Use hex value -->
<table bgcolor="#6666FF" width="100%">
<tr><td>
This background is sky blue
</td></tr>
</table>
<!-- Format 3 - Use color value in RGB terms -->
<table bgcolor="rgb(255,0,255)" width="100%">
<tr><td>
This background is green
</td></tr>
</table>
</body>
</html>
```

This will produce the following result:



This background is yellow

This background is sky blue

This background is green

Html Background with Images

The **background** attribute can also be used to control the background of an HTML element, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table. Following is the syntax to use background attribute with any HTML tag.

Note: The *background* attribute is deprecated and it is recommended to use Style Sheet for background setting.

```
<tagname background="Image URL"...>
```

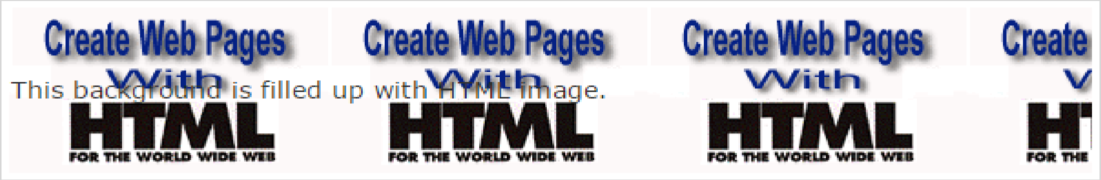
The most frequently used image formats are JPEG, GIF and PNG images.

Example

Here are the examples to set background images of a table.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Images</title>
</head>
<body>
<!-- Set table background -->
<table background="/images/html.gif" width="100%" height="100">
<tr><td>
This background is filled up with HTML image.
</td></tr>
</table>
</body>
</html>
```

This will produce the following result: This background is filled up with HTML image.



This background is filled up with HTML image.

4.15: HTML – COLORS

Colors are very important to give a good look and feel to your website. You can specify colors on page level using <body> tag or you can set colors for individual tags using **bgcolor** attribute. The <body> tag has following attributes which can be used to set different colors:

- **bgcolor -** sets a color for the background of the page.
- **text -** sets a color for the body text.
- **alink -** sets a color for active links or selected links.
- **link -** sets a color for linked text.
- **vlink -** sets a color for *visited links* - that is, for linked text that you have already clicked on.

HTML Color Coding Methods

There are following three different methods to set colors in your web page:

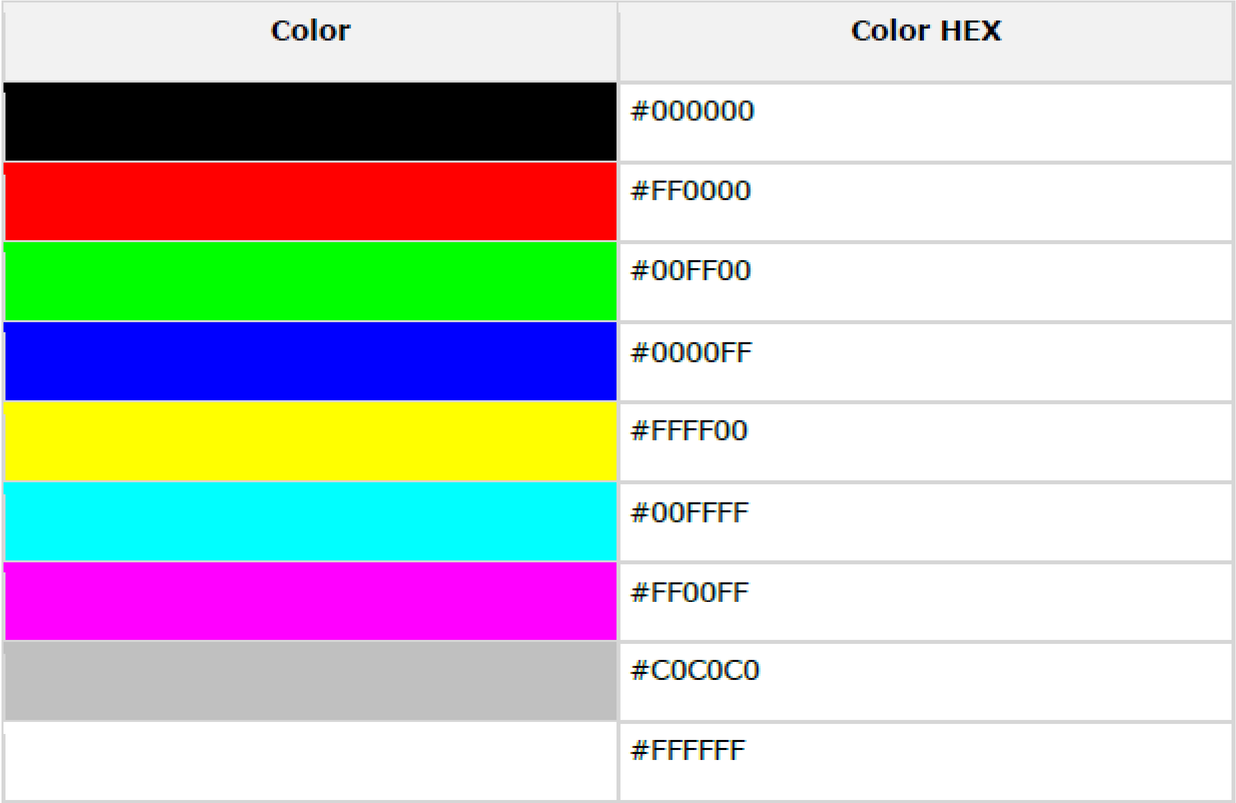
- **Color names -** You can specify color names directly like green, blue or red.
- **Hex codes -** A six-digit code representing the amount of red, green, and blue that makes up the color.
- **Color decimal or percentage values -** This value is specified using the rgb() property.

HTML Colors - Color Names

You can specify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by major browsers.

HTML Colors - Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB). Each hexadecimal code will be preceded by a pound or hash sign #. Following is a list of few colors using hexadecimal notation.



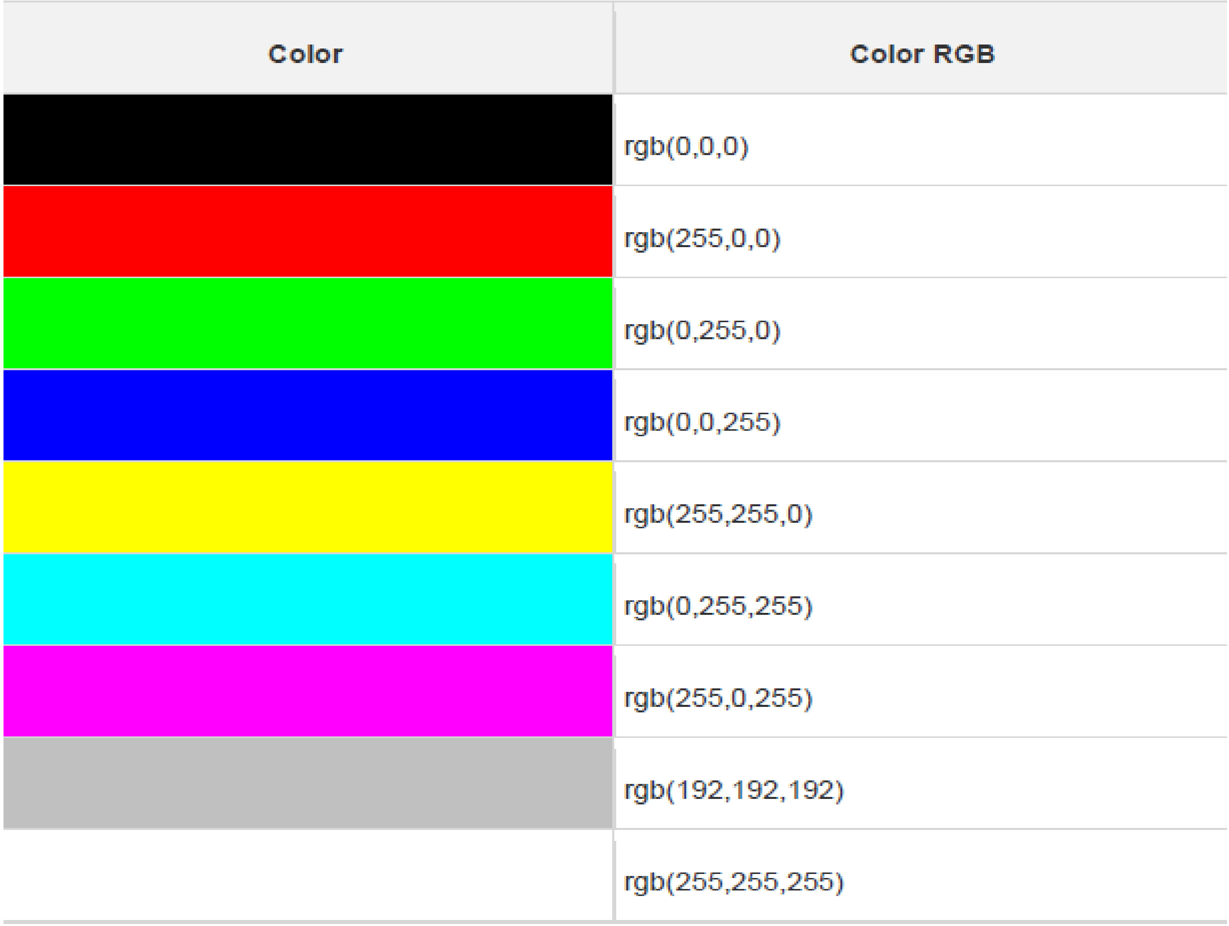
Color	Color HEX
⬛	#000000
🟥	#FF0000
🟩	#00FF00
🟦	#0000FF
🟨	#FFFF00
🟦	#00FFFF
🟪	#FF00FF
▨	#C0C0C0
⬜	#FFFFFF

HTML Colors - RGB Values

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

Note: All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is a list to show few colors using RGB values.



Color	Color RGB
⬛	rgb(0,0,0)
🟥	rgb(255,0,0)
🟩	rgb(0,255,0)
🟦	rgb(0,0,255)
🟨	rgb(255,255,0)
🟦	rgb(0,255,255)
🟪	rgb(255,0,255)
▨	rgb(192,192,192)
⬜	rgb(255,255,255)

4.16: HTML – FORMS

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

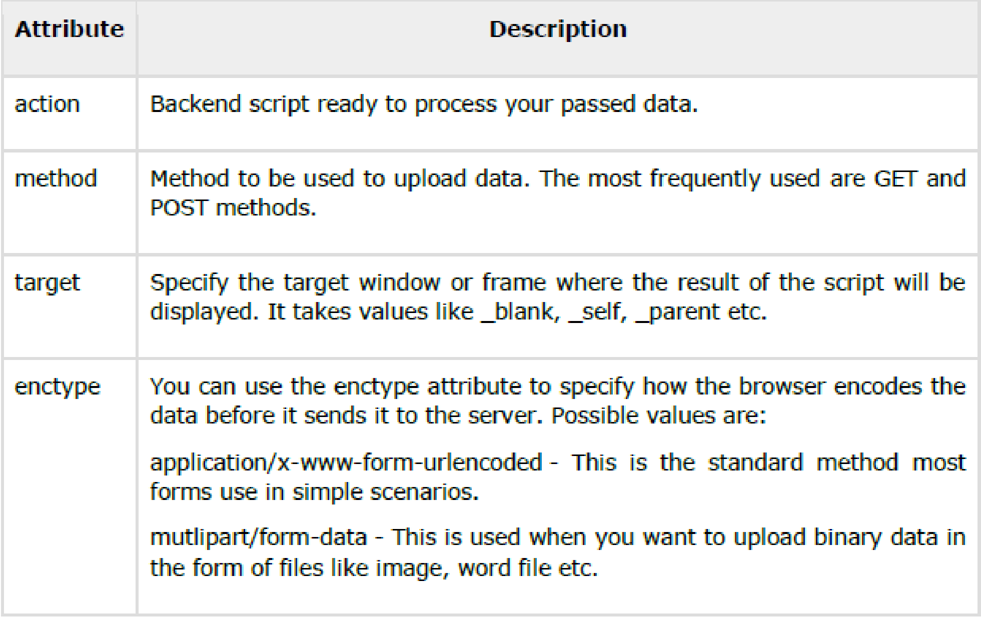
There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

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

```
<form action="Script URL" method="GET|POST">
form elements like input, textarea etc.
</form>
```

Form Attributes

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



Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are: application/x-www-form-urlencoded - This is the standard method most forms use in simple scenarios. mutlipart/form-data - This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Controls

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

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

The INPUT Tag

- ➢ The INPUT tag is a multipurpose tag that creates many different types of controls
- ➢ The type of input is controlled by the TYPE attribute – Can be **TEXT, PASSWORD, CHECKBOX, RADIO, SUBMIT, RESET, FILE, HIDDEN, IMAGE, or BUTTON**
- ➢ Almost all of these should have a NAME attribute
- ➢ Their initial state can be set with a VALUE attribute
- ➢ They can all be disabled with DISABLE
- ➢ An INPUT tag is never closed (no)

Checkboxes

• The NAME attribute names this checkbox

• The CHECKED attribute (with no value) is used to indicate a pre-checked checkbox

• The VALUE attribute specifies the value bound to name if this checkbox is submitted (default = ON)

```
Check all that apply:<br>
<input type="checkbox" name="dogs">I like dogs<br>
<input type="checkbox" name="cats">I like cats<br>
<input type="checkbox" name="pigs">I like pigs
```

Radio Boxes

- <input type="radio" ...>
- Used when user is to select one of many *mutually exclusive* options
- Radio buttons with same name form a group of mutually exclusive options

```
Select <em>one of</em> the following:<br>
<input type="radio" name="agree"
checked value="a">I agree completely<br>
<input type="radio" name="agree"
value="b">I agree a little<br>
<input type="radio" name="agree"
value="c">I disagree a little<br>
<input type="radio" name="agree"
value="d">I disagree completely<br>
```

Text Boxes

- <input type="text" ...>
- Allows entry of *one line* of text (Actually, not completely true – try cut and paste)
- Attribute SIZE specifies the width (in characters)
- Attribute MAXLENGTH specifies the maximum number of characters

```
User's full name:
<input name="fullname" type="text" size="30"
maxlength="50">
```

Passwords

- <input type="password" ...>
- Identical to a text box, but text typed into the box is not readable
- Useful for submitting sensitive information (like passwords)

Buttons

✓ <input type="submit" ...>

– Creates a button that submits the form to the server

✓ <input type="reset" ...>

– Creates a button that resets all form fields to their default state

✓ <input type="button" ...>

– Creates a button that does nothing

<input type="submit" value="Submit Form Data">

Buttons (2)

✓ The BUTTON tag provides similar functionality but with a bit more flexibility

✓ Attributes are NAME, VALUE, and TYPE

✓ TYPE can be SUBMIT, BUTTON, RESET

```
<button name="sb" value="sbData" type="submit">
Submit All Form Data Now
</button>
```

Main difference is that the button label is text

within the tag instead of the VALUE attribute

Image Buttons

- <input type="image" ...>
- Displays an image that behaves like a submit button
- The SRC attribute specifies the location of an image file
- The ALT attribute specifies some text to render if the image is not displayable

<input type="image" src="button.png" alt="Submit">

Hidden Control

- <input type="hidden" ...>
- Creates a control similar to a text control

– User does not see control

– User can not easily change the value

- Useful for keeping track of data as the user traverses a collection of pages

```
<input type="hidden" name="hiddendata" value="Hidden Data in
Here">
```

Text Areas

- The TEXTAREA tag provides a multiline text entry area
- The ROWS and COLS attributes are required and they specify the number of rows and number of columns

```
<textarea rows="30" cols="50" name="bigtext">
The preformatted initial text is sandwiched within the tag.
</textarea>
```

Menus

- Drop-down menus are created using the SELECT tag

- Attribute SIZE determines how many rows to display at once
- Each option is enclosed in an OPTION tag

```
<select name="country" size="5">
<option value="AB">Abkhazia</option>
...
<option value="ZB">Zimbabwe</option>
</select>
```

The MULTIPLE attribute of the SELECT tag creates menus that allow multiple selections

- Options can be grouped hierarchically using the OPTGROUP tag

Labels

✓ The LABEL tag specifies that the enclosed item is a label for the named form element

✓ For example, clicking the label will shift the focus or change the state of the associated form element

```
Check all that apply<br>
<input type="checkbox" name="doglover" id="dogs"
checked>
<label for="dogs">I like dogs</label>
<br>
<input type="checkbox" name="catlover" id="cats">
<label for="cats">I like cats</label>
<br>
<input type="checkbox" name="piglover"
<label for="pigs">I like pigs</label>
```

Fieldsets

➢ The FIELDSET tag is used to group together a set of related form elements

➢ The LEGEND tag assigns a caption to a field set

```
<fieldset>
<legend>Personal Information</legend>
First Name: <input type="text" name="fn" size="20">
<br>
Last Name: <input type="text" name="ln" size="20">
<br>
Date of Birth: <input type="text" name="dob"
size="10">
</fieldset>
```