

1.1 Introduction to HTML:

HTML means Hypertext Markup Language. In 1960 Ted Nelson introduced Hypertext. HTML is a markup language which is used to create static web pages. If you are thinking of creating your own web pages, you need to know at least basic HTML. These HTML documents are plain text files; user can create these documents using text editor like Notepad or gedit.

HTML is a hypertext Language because it supports font styled text, pictures, graphics and animations and also it provides hyper links that used to browse the Internet easily. Text becomes hypertext with the addition of links that connects other hypertext documents. Hypertext is a text augmented with links-pointers to other pieces of text, possible else where in the same document (internal linking) or in another document (external linking).

Note: Hypertext is non-linear text which contains links to other pieces of text Markup comprises a set of symbols that can be used to provide instructions for viewing a web page

History of HTML

HTML was originally developed by Tim Berners-Lee while at CERN, and popularized by the Mosaic browser developed at NCSA. HTML standards are created by group of interested organizations and individuals called **W3C (World Wide Web Consortium)**. There have now been three official HTML standards:

1. HTML 2.0 was released in 1994 and remains the baseline for backwards compatibility and should be supported by all browsers.
2. HTML 3.2 was released in 1996 with many useful additions
3. HTML 4.0 was released in 1997 and slightly amended in 1999
4. Now, The current version is HTML 5.0 was released in 2012

Advantages of HTML:

1. First advantage it is widely used.
2. Easy to learn and use.
3. It is error-free language
4. Every browser supports HTML language.
5. It is by default in every **OS**. so you don't need to purchase extra software.

Disadvantages of HTML:

1. It can create only static and plain pages so if we need dynamic pages then HTML is not useful.
2. Need to write lot of code for making simple webpage.
3. Security features are not good in HTML.

1.2 Tags:

A tag is a format name surrounded by angular brackets which tells when to switch on a piece of formatting and when to switch it off.

- ❖ A tag is made up of left operator (<), a right operator (>) and a tag name between these two operators.
- ❖ If you forget to mention the right operator (>) or if you give any space between left operator and tag name browser will not consider it as tag.
- ❖ At the same time if browser not understands the tag name it just ignores it, browser won't generate any errors.
- ❖ HTML language is not case sensitive; hence user can write the code in either upper case or lower case. No difference between <HTML>, <html> and <hTmL>

- ❖ Every HTML document begins with start tag is <HTML> terminates with an ending tag is </HTML>
 - ❖ HTML documents should be saved with the extension **.html**.
- Syntax of a tag:

< tagname [parameter=value]>

Ex: HR is a tag name that displays a horizontal ruler line.

<HR> - ---- (No parameters, no value)
 <HR ALIGN=CENTER> ----- (Tag with parameter and value for the parameter)
 <HR WIDTH="30%" SIZE="100" ALIGN="RIGHT"> ----- (Tag with more parameters with their values)

Different types of Tags:

1. Container/Paired tags
 2. Empty/Singleton tags
- ❖ Singleton tag does not require an ending tag but they end with slash. (Ex: <HR/>)
 - ❖ Container tag required an ending tag, which is similar to opening tag except backslash before the tag name (Ex: <HTML> is opening tag, then ending tag is </HTML>)

1.3 Structure of an HTML document:

All HTML documents follow the same basic structure. They have the root tag as <html>, which contains <head> tag and <body> tag. The <head> tag is used to place control information used by the browser and server. The <body> tag contains the actual user information that is to be displayed on the computer screen. The basic document is shown below:

```
<html>
  <head>
    <title> Basic HTML document </title>
  </head>
  <body>
    <h1> Welcome to the world of Web Technologies</h1>
    <p> A Sample HTML Document </p>
  </body>
</html>
```

Besides <head> and <body> tag, there are some other tags like title, which is a sub tag of head, that displays the information in the title bar of the browser. <h1> is used to display the line in its own format i.e., bold with some big font size. <p> is used to write the content in the form of paragraph.

Comments in HTML

An HTML comment begins with "<!--" and ends with "-->". There should not be a space between angular bracket and exclamation mark. Each comment can contain as many lines of text as you like. If comment is having more lines, then each line must start and end with -- and must not contain -- within its body.

<!-- this is a single line comment line -->

```
<!-- this is a multiline comment --  
-- spawned over --  
-- three line -->
```

The major purpose of comments to improve the readability of a program.

The Document Head

The only tag that most authors insert in their head sections is the title:

```
<title>... </title>
```

All HTML documents have just one title which is displayed at the top of the browser window. The title is used as the name in the bookmark fields and on search engines

The Document Body

In HTML, Documents are structured as blocks of text, each of which can be formatted independently. The two major block of text in HTML documents are the paragraph and heading.

In HTML paragraphs are created by <p> tag and headings are created by <h_n> tag where n starts from 1 to 6

Creating HTML Page

The Following steps are needed to create a HTML page

Step 1: Open any text editor like Notepad, gedit, Wordpad etc.

Step 2: Use the File menu to create a new document (File→ New) and type the HTML code

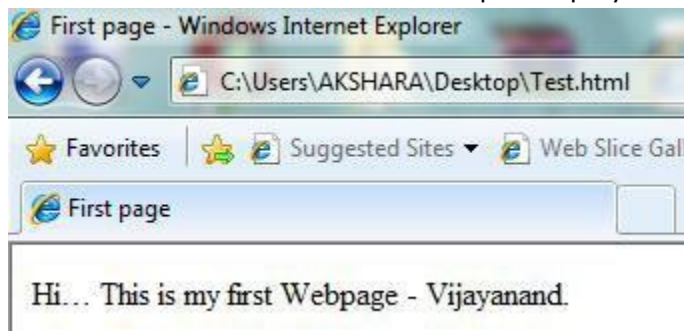
```
<HTML>  
  <HEAD>  
    <TITLE>First page </TITLE>  
  </HEAD>  
  <BODY>  
    Hi... This is my first Webpage - Vijayanand.  
  </BODY>  
</HTML>
```

Step 3: Go to the file menu and choose "Save as" option (File->Save as) and give the name of the file as "example.html" under root directory(C:)(or any valid path)

Step 4: After saving, an internet explorer icon will be displayed as shown below



Step 5: Double click to execute it. The output displayed as follows



1.4 Basic HTML tags

Character-formatting tags:

You can use character formatting tags to format a text block that is as small as a single character or as large as an entire document. Some of the most frequently used Character-formatting tags are :

1. Boldface tag

This tag is used for implement bold effect on the text

` `

2. Italic tag

This implements italic effects on the text.

`<i>.....</i>`

3. Underline tag

This is used to specify that the selected text be displayed with underline.

`<u>. . . </u>`

4. Big tag

This is to specify that the selected text be displayed as bigger font size as compared to the font for the rest of the text.

`<big>. . .</big>`

5. Small tag

This is to specify that the selected text be displayed as smaller font size as compared to the font for the rest of the text.

`<small>. . .</small>`

6. strong tag

This tag is used to always emphasized the text

`. . . `

7. sub and sup tag

These tags are used for subscript and superscript effects on the text.

`_{. . .}`

`^{. . .}`

Block-formatting tags:

You can use following tags to format blocks of text within HTML document. Some of the most frequently used Block-formatting tags are :

1. Body tag (<body>)

Body tag contains some attributes such as bgcolor, background etc. *bgcolor* is used for setting the background color of a webpage which takes background color name or hexadecimal number such as #000000 to #FFFFFF and background attribute used for setting image as a background for webpage and it will take the path of the image which you can place as the background image in the browser.

Syntax:

`<body bgcolor="name/#rrggbb" background="image name"> . . . </body>`

Example:

`<body bgcolor="#F2F3F4" background="c:\vijay\image.jpg">...</body>`

2. Paragraph tag (<p>)

Most text is part of a paragraph of information. Each paragraph is aligned to the left, right or center of the page by using an attribute called as align.

Syntax:

```
<p [align="left" | "right" | "center"] > . . . </p>
```

3. Heading tag(<H_n>)

HTML is having six levels of heading that are commonly used. The largest heading tag is <h1>. The different levels of heading tag besides <h1> are <h2>, <h3>, <h4>, <h5> and <h6>. Each heading tag has an attribute called as align which can be set to left, center, or right. By default all headings align left.

Syntax:

```
<h1 [align="left" | "right" | "center"]> . . . </h1>
```

```
<h2 [align="left" | "right" | "center"]> . . . </h2>
```

```
.
```

```
<h6 [align="left" | "right" | "center"]> . . . </h6>
```

4. <hr> tag

This tag places a horizontal line across the screen. These lines are used to break up the page. This tag does not require an end tag. This tag also contains attributes which determines how the rule will be displayed. It can be aligned but by default is centered on the screen. The *size* attribute specifies the thickness of the rule in pixels. *noshade* draws the rule as a single thick line rather than giving it's default 3D appearance.

Syntax:

```
<hr align="left" | "right" | "center" width="nn%" size="n" [noshade]/>.
```

5. base font

This specify the minimum text size for the entire webpage but not the headings.

Syntax:

```
<basefont size="n">
```

font tag

This sets font type, size, color and relative values for a particular text. Absolute font sizes are can be set from 1 to 7. Relative font sizes are set by using +/- 1 to 7. The color of the text is set by *color* attribute. This takes hex value which represents the amounts of red, green and blue in a chosen color.

Syntax:

```
<font face="font name "size="["+|-]n" color="#rrggbb">
```

6. tt tag

This tag is used to give teletype effect on the text

```
<tt>. . .</tt>
```

7. Line break tag

This tag is used to the break the line and start from the next line. It is an empty tag.

```
<br/>
```

8. Pre-formatted text tag

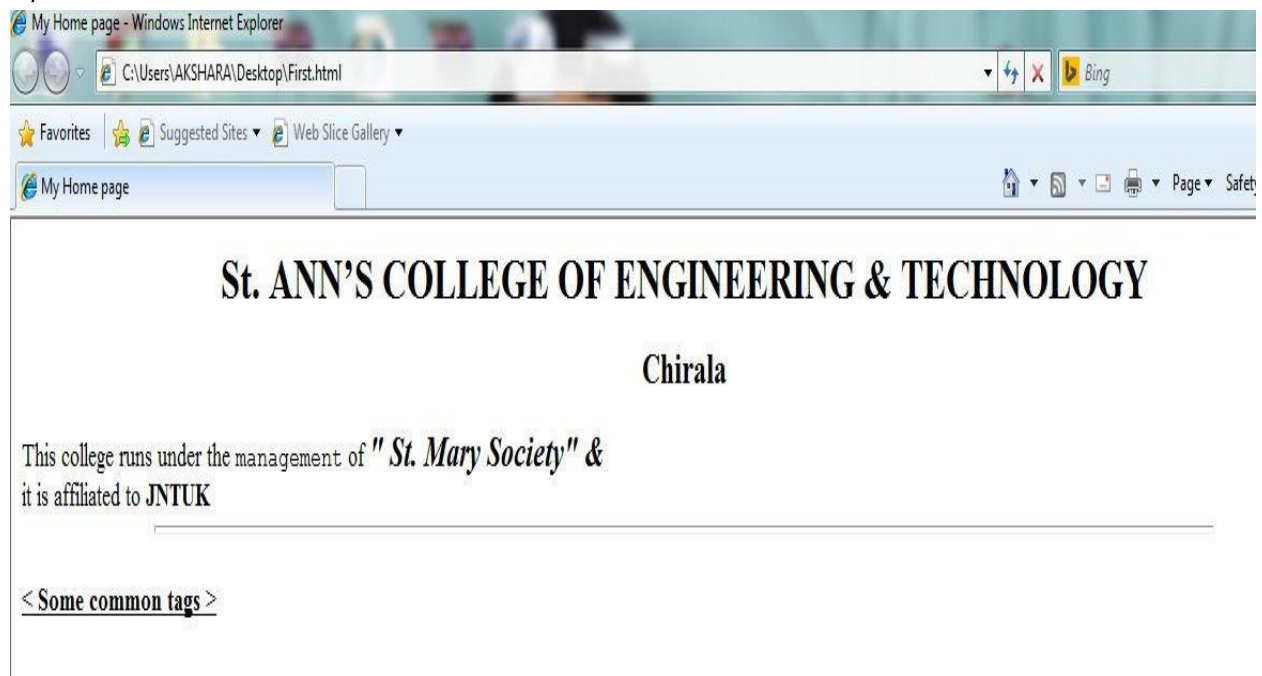
It Considers spaces, new lines etc. and as it is prints the information.

```
<PRE>. . .</PRE>
```

Example : HTML code to implement common tags.

mypage.html

```
<html>
<head> <!-- This page implements common html tags -->
<title> My Home page </title>
</head>
<body >
<h1 align="center"> St. ANN"S COLLEGE OF ENGINEERING & TECHNOLOGY</h1>
<h2 align="center"> Chirala</h2>
<basefont size=4>
<p> This college runs under the <tt>management</tt> of <font size=5> <b><i>&quot;
St. Mary Society&quot; & amp </i></b></font><b>
it is affiliated to <strong> JNTUK</strong>
<hr size=5 width=80%>
<h3> <u>&lt Some common tags &gt</u> </h3><br>
</body>
</html>
```



9. MARQUEE Tag:

This is to create scrolling effect for the selected text in an HTML Page. When you use this tag, the selected text is highlighted and scrolls from right to left and vice versa.

Syntax:

```
<marquee> . . .</marquee>
```

Marquee tag attributes:

- align: sets the alignment of the text relative to the marquee. Set to top(default), middle or bottom.
- behavior: Sets how the text in the marquee should move, It can be scroll(default), slide(text enters from one side and stops at the other end), or alternate(text seems to bounce from one side to other)
- bgcolor: sets the background color for the marquee box
- direction: sets the direction of the text for scrolling. It can be left(default), right, down or up.

UNIT-I***HTML AND JAVASCRIPT***

- e) height and width: to specify the height and width of a marquee text either in pixels or in % of the screen height.
- f) Loop: to specify the number of times a marquee will loop. -1 for infinite times.

- g) `scrollDelay`: to control the speed at which marquee text is drawn on the screen. This attribute specifies the number of milliseconds between each successive draw of the marquee text.
- h) `scrollAmount`: to specify the number of pixels that marquee text moves in every movement.

Example:

```
<marquee align="middle" behavior="slide" bgcolor="red" direction="down">
  Vijayanand
</marquee>
```

10. Blinking text Tag

`<blink>.....</blink>` displays enclosed text as blinking on and off approximately once a second. This tag only works in Netscape navigator

11. Escape sequences

These are character escape sequence which are required if you want to display characters that HTML uses as control sequences. All of these characters starts with an ampersand (&) and are terminated with a semicolon (;)

&amp;	&lt;	&gt;	&nbsp;	&quot;	&copy;
&	<	>		'	©

1.5 HYPERLINKS

The real power and flexibility of HTML is in Hyperlinks. Hyperlinks are created with anchor tag(`<a>`)

You can use hyperlink to

1. create links with HTML pages
2. link different HTML pages
3. access services at other Internet sites

Anchor tag:

The anchor tag is created by `<a> . . . ` tags. The tag has three sections: the address of the referenced document, a piece of text displayed as link, and the closing tag.

Syntax:

```
<a href="address" name="id" target="name" title="description"> Text </a>
```

- ✓ *href* means Hypertext references that can be used for giving the path of a file which you want to link.
- ✓ The "Text" between the `<a> . . . ` tags acted as a hyperlink. This text is called "*hypertext*". When you click on this text, the linked page or file will be displayed. You can also replace this text with image. In that case, the image will act as a hyperlink.

Attributes	Description
<i>href</i>	Used to specify the path and file name of the HTML page that you need to access by using a hyperlink
<i>name</i>	The <i>name</i> attribute enable you to create anchor with in HTML page. This anchor tag is used to bookmark a location in an HTML page.
<i>title</i>	The <i>title</i> attribute is used to provide information about a link. The value specified for this attribute appears as a small tip when you place the mouse pointer over the hyperlink.

Target	The <i>target</i> attribute is used to specify the target window where the linked HTML page is displayed. You can assign the name of a frame as the value for this attribute.
---------------	---

Creating hyperlinks to HTML Pages

To create hyperlinks to an HTML page, you specify the address of the HTML pages as the value of *href* attribute of the `<a> . . . ` tags

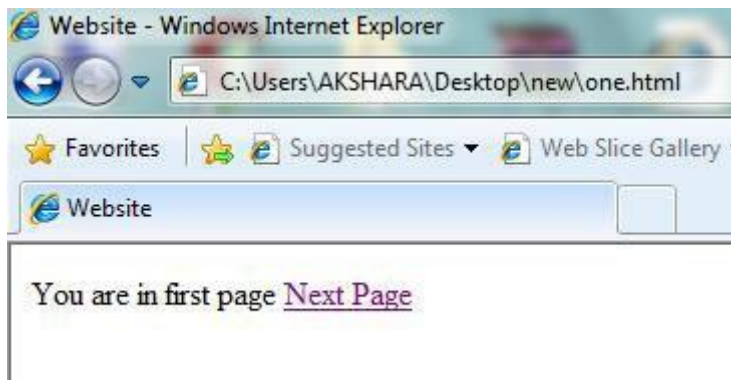
Consider the example of simple HTML page that displays „next“ as hyperlink, Create a HTML page that connect web page created through hyperlink.

first.html

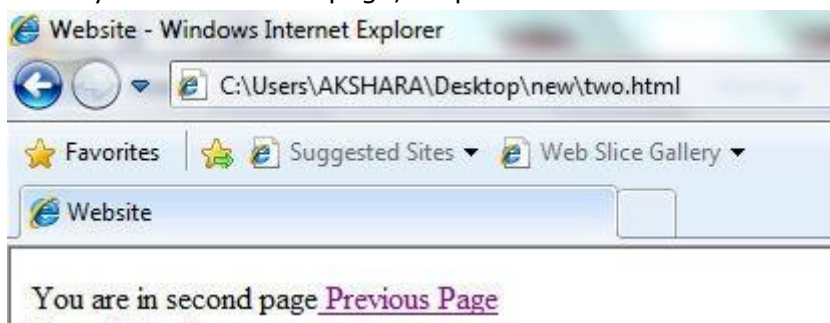
```
<html>
<head> <title> Navigation </title> </head>
<body>
<p align="left">You are in first page<a href="second.html">Next Page</a> </p>
</body>
</html>
```

second.html

```
<html>
<head> <title> Navigation </title> </head>
<body>
<p align="left">You are in second page<a href="first.html">Previous Page</a> </p>
</body>
</html>
```



When you click on Next page, it opens



Creating Hyperlinks within an HTML Page

HRFF and NAME are the most important attributes of anchor tag that are required to create hyperlinks within an HTML page. The *name* attribute is used to create anchors

with in a page and the href attribute is used to create links that to these anchors.

To Create anchors in an HTML page, you can use the following syntax:

```
<a name="Myname"> This is anchor</a>
```

After you create an in HTML page, you create a hyperlink to refer that anchor, To do this , use the following syntax:

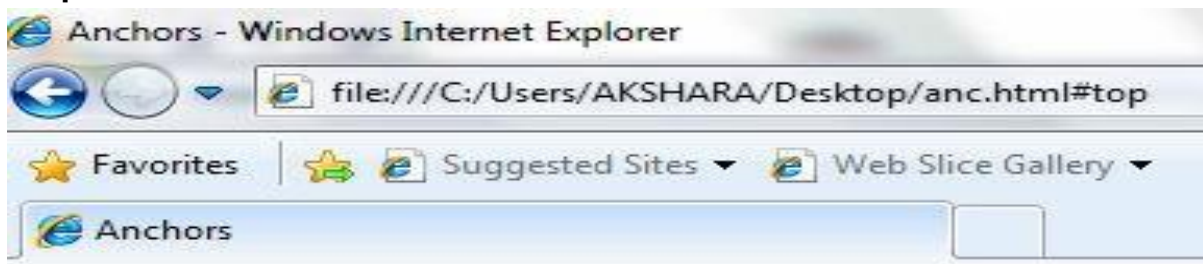
```
<a href="#Myname"> This is anchor</a>
```

The hash (#) symbol before Myname specifies that it is the name of an anchor. When you click the text within the <a> . . . tags, The browser searches for the Myname anchor within the HTML page. When the anchor is found, the Webpage is scrolled up or down to display the text at the anchor position.

Consider the following code:

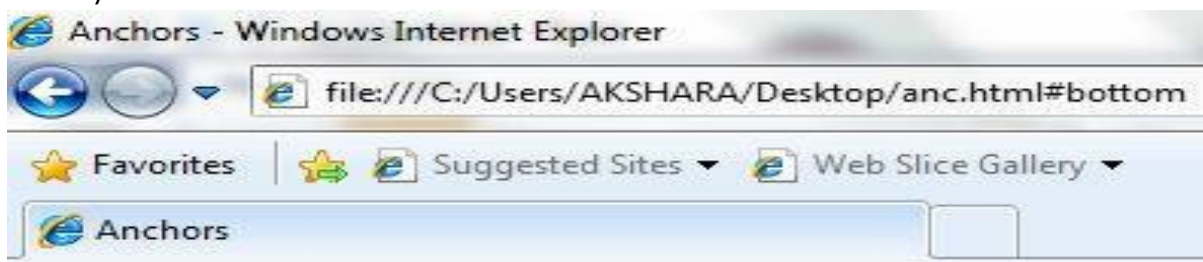
```
<html>
  <head> <title> Anchors </title> </head>
  <body>
    <p align="left">Top of the page<a name="top" href="#bottom"> Bottom
    </a> </p>
    <br/><br/><br/><br/><br/><br/><br/><br/><br/>
    <p align="left">Bottom of the page<a name="bottom" href="#top"> Back to Top
    </a> </p>
  </body>
</html>
```

Output:



Top of the page [Bottom](#)

When you click on bottom



Bottom of the page [Back to Top](#)

1.6 Color and Images

Color is essential to the Web experience; it brings life to pages and makes them exciting. Color can be used in number of places on a Webpage: The background can be colored, individual elements can be altered, and links which already colored can be changed

To change the colors of links or of the page background hexadecimal values are placed in the <body> tag:

Syntax:

```
<body bgcolor = "#rrggbb" text = "#rrggbb" link = "#rrggbb" vlink = "#rrggbb" alink = "#rrggbb">
```

The *vlink* attribute sets the color of links visited recently, *alink* the color of a currently active link. The six figure hexadecimal values must be enclosed in double quotes and preceded by a hash(#).

Images

Images are second aspect of pleasant Web experience. You can add images to an HTML page to either improve its appearance or present important information. To add images to an HTML page, you can use several image formats. These formats include „gif“, „jpg“ and „png“.

Images can be added in two different ways:

1. By using „background“ attribute of <body> tag
2. By using tag

The <BODY> Tag: background attribute

You can use the background attribute of the <BODY> tag to add image as a background in an HTML page.

Syntax:

```
<body background = "URL"> . . .</body>
```

This tag will set a background image present in the URL.

The Image tag

You can use image tag in an HTML page to add images along with text. An image added using the image tag occupies space within the HTML page. To add image to HTML page, use tag.

Syntax:

```

```

Attributes:

- a) src : used to specify the name of the file. When using tag, it is mandatory to specify a value for the **SRC** attribute
- b) align : used to specify the vertical alignment of an image
- c) height : used to specify the vertical area that an image will occupy in HTML page
- d) Width : used to specify the horizontal area that an image will occupy in HTML page
- e) alt : used to specify the text when browser unable to display the image or image not available.

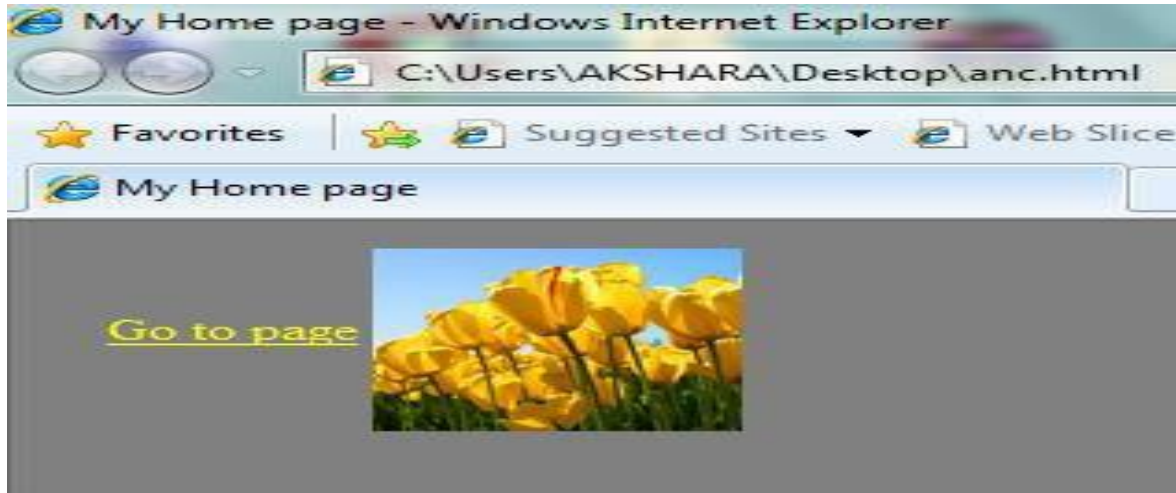
Example : HTML code that implements color and image

```
<html>
<head>
  <title> Colors and images </title>
</head>
<body bgcolor="gray" text = "silver" link = "white" vlink = "yellow" alink = "brown">
  <a href="first.html">Go to First Page</a>
  <img src= "C:\Users\Public\Pictures\Sample Pictures \Tulips.jpg" align = "middle"
    height = "100" width="100" alt="Tulips"/>
```

UNIT-I
</body> </html>

HTML AND JAVASCRIPT

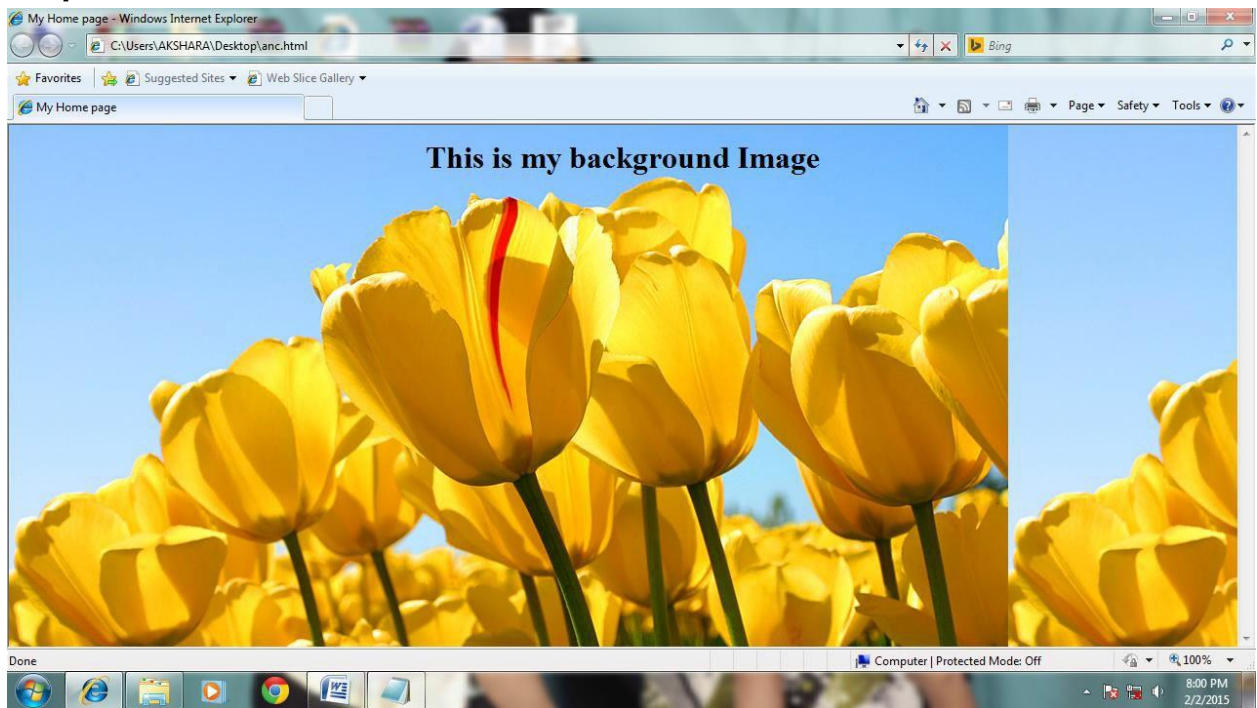
Output:



Example : HTML code that implements background image

```
<html>
<head> <title> My Home page </title> </head>
<body background="C:\Users\Public\Pictures\Sample Pictures\Tulips.jpg">
<h1 align="center"> This is my background Image</h1>
</body> </html>
```

Output:



1.7 Lists

One of the most effective ways of structuring a web site is to use lists. Lists provides straight forward index in the website. HTML provides three types of lists.

1. Unordered / Bulleted list
2. Ordered / Numbered list and
3. Definition/Description list

Lists can be easily embedded within other lists to provide complex yet readable structures.

...:

The Ordered / Unordered lists are each made up of set of list items. These list

items are added with **** tag. Elements of a list item may format with any of the usual formatting tags and may be images or hyperlinks.

Unordered Lists

Unordered lists are also called Un-numbered lists. The Unordered list elements are used to represent a list of items, which are typically separated by white space and/or marked by bullets. Using **** tag is used to create unordered lists in HTML. Which is paired tag, so it requires ending tag that is ****. The list of items is included in between **.....**. The *TYPE* attribute can also be added to the **** tag that indicates the displayed bullet along with list of item is **square, disc** or **circle**. By default it is disc.

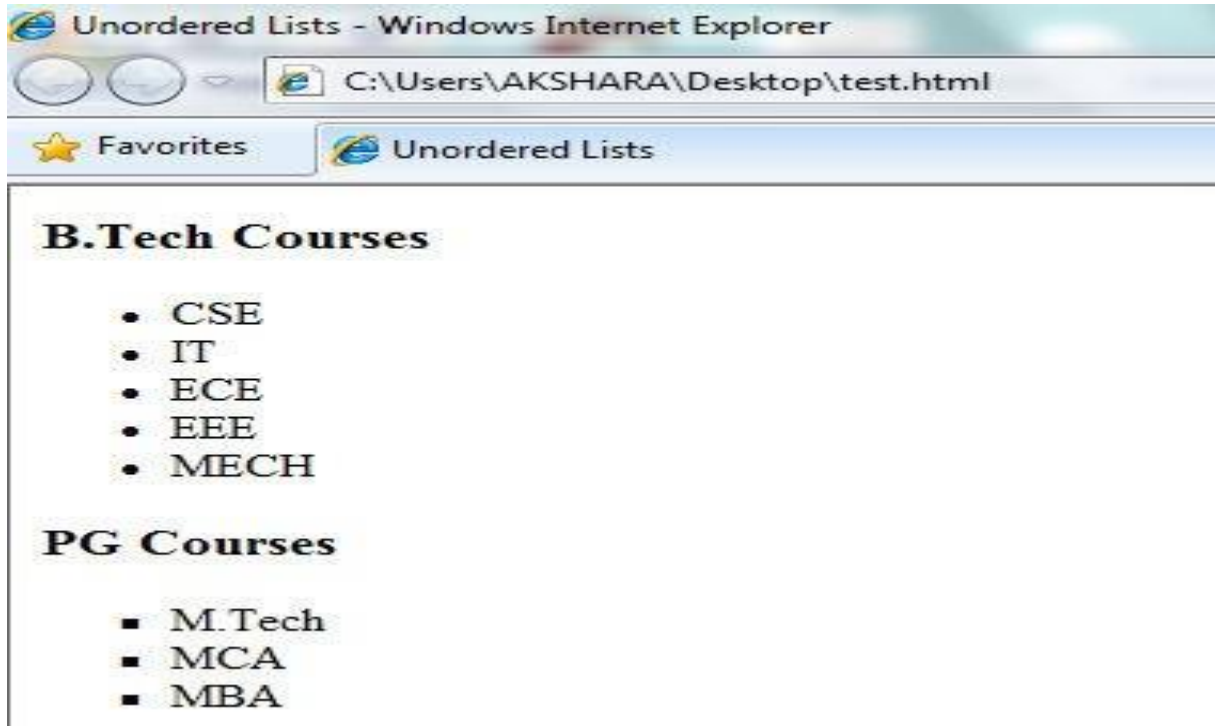
Syntax:- <UL [TYPE="square | disc |circle "] >
 item name1
 item name2

 item name n

Example: Write a HTML program for displaying names of B.Tech Courses with default bullets and names of PG Courses with square bullets.

```
<html>
<head>
  <title>Unordered Lists</title>
</head>
<body>
  <h3>B.Tech Courses </h3>
  <ul type="disc">
    <li>CSE </li>
    <li>IT </li>
    <li>ECE</li>
    <li>EEE</li>
    <li>MECH</li>
  </ul>
  <h3>PG Courses </h3>
  <ul type="square">
    <li>M.Tech</li>
    <li>MCA</li>
    <li>MBA</li>
  </ul>
</body>
</html>
```

Output:



Ordered Lists:-

Ordered lists are also called **Numbered** or **Sequenced lists**. In the ordered list the list of items have an order that is signified by numbers, hence it some times called as number lists. Elements used to present a list of items, which are typically separated by white space and/or marked by numbers or alphabets. An orders list should start with the element, which is immediately followed by a element which is same as in unordered list. End of ordered lists is specified with ending tag .

Different Ordered list types like roman numeral list, alphabet list etc. can be specified with *TYPE* attribute. Another optional parameter with tag is *START* attribute, which indicates the starting number or alphabet of the ordered list. For example TYPE="A" and START=5 will give list start with letter E. The TYPE attribute used in , changes the list type for particular item. To give more flexibility to list, we can use VALUE parameter with tag that helps us to change the count for the list item and subsequence items.

Syntax:-

```
<OL [TYPE="1 | A | a | I | i] start="n">
```

```
<LI>item name1
```

```
<LI>item name2
```

```
-----
```

```
-----
```

```
<LI>item name n
```

```
</OL>
```

Different Ordered list types

Type="1" (default) e.g.1,2,3,4.....

Type="A" Capital letters e.g.A,B,C...

Type="a" Small letters e.g. a,b,c.....

Type="I" Large roman letters e.g. I, II, III,...

Example: Write a HTML program for displaying Colors of VIBGYOR with numbers and names of Fruits with alphabets from "E".

```
<html>
```



```

<head> <title>Ordered Lists</title> </head>
<body text="maroon">
<h3>Colors in VIBGYOR</h3>
<OL>
    <LI>Violet</LI>
    <LI>Indigo</LI>
    <LI>Blue</LI>
    <LI>Green</LI>
    <LI>Yellow</LI>
    <LI>Orange</LI>
    <LI>Red</LI>
</OL>
<h3>Types of Fruits</h3>
<OL type="A" start="5">
    <LI>Apple</LI>
    <LI>Banana</LI>
    <LI>Grapes</LI>
</OL>
</body>
</html>

```

Output:**Definition/Description List:** <DL>.....</DL>

A Definition list is a list of definition terms <DT> and corresponding Definition Description <DD> on a new line. To create a definition it must start with <DL> and immediately followed by the first definition term <DT>

Example:

```

<html>
    <head><title>Definition List</title> </head>
    <body>
        <DL>

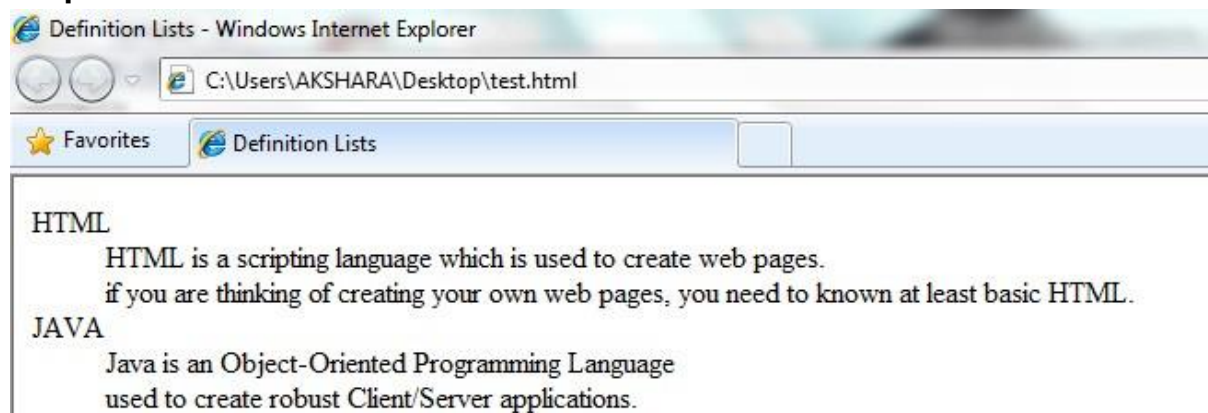
```



```

        <DT>HTML</DT>
        <DD>
            HTML is a scripting language which is used to create web
            pages.<br/>if you are thinking of creating your own web
            pages, you need to know at least basic HTML.
        </DD>
        <DT>JAVA</DT>
        <DD>
            Java is an Object-Oriented Programming Language<br/>
            used to create robust Client/Server applications.
        </DD>
    </DL>
</body>
</html>

```

Output:

Nested Lists: Lists can be nested that is Nested Lists is list with in another list.

1.8 Tables

Table is one of the most useful HTML constructs. Tables are finding all over the web applications. The main uses of table are that they are used to structure the pieces of information and to structure the whole web page.

- ❖ Tables allow you to present data across rows and columns, making it easy for reader to correlate connected pieces of information.
- ❖ A Table is collection of rows and columns or collection of cells
- ❖ The basic element in a table is *cell* or the *grid*. A cell is formed when a row crosses a column.
- ❖ You can specify the amount of space between cells in a table. This space is called *cell spacing*
- ❖ You can specify the amount of space between the contents of a cell and the cell wall. This space is called *cell padding*

You can use various elements to specify the details of a table. Many table elements also take attributes, which allows you to further specify the look of the table.

- ❖ TABLE
- ❖ Table row
- ❖ Table data
- ❖ Table Heading

The TABLE Element

The TABLE element is the container element for table and uses the <TABLE>...

</TABLE> tags to enclose all the other table tags. If the <TABLE> tag is omitted or not closed, the browser ignores all the other tags that you specify for the table. Everything that we write between these two tags will be within a table. The attributes of the table will control in formatting of the table.

The <TABLE> tag has the following attributes:

- align: used to specify the alignment of a table in a HTML page.
- border: used to specify the thickness of the table border in pixels.
- bgcolor: used to specify the background color for the table.
- frame: used to specify the which side of the outer border is visible in the browser. You must specify the border attribute before you specify the FRAME attribute.

Values for the FRAME Attribute

Value	Description
VOID	Removes all External borders
BOX	Displays a box around the table i.e. all four sides of the table
HSIDES	Displays an external border at the top and bottom of the table
VSIDES	Displays an external border at the right and left of the table
LHS	Displays an external border only on the left hand side of the table
RHS	Displays an external border only on the right hand side of the table
ABOVE	Displays an external border only on the top of the table
BELOW	Displays an external border only on the bottom of the table

- rules: used to specify the borders between cells. You must specify the border attribute before you specify the RULES attribute.

Values for the RULES Attribute

Value	Description
NONE	Removes all internal rules from the table
ROWS	Displays rules between rows of the table
COLS	Displays rules between columns of the table
ALL	Displays rules between all rows and columns

- cellspacing: used to specify the spacing between cells in pixels.
- cellpadding: used to specify the spacing between cell content and cell wall in pixels.
- height: used to specify the height of table in pixels or %.
- width: used to specify the width of table in pixels or %.

Syntax:

```
<TABLE align="left | right | center" border="n" bgcolor="#rrggbb" cellspacing="n"
cellpadding="n" frame="value" rules="value" height="n | %" width="n | %">
...
</TABLE>
```

The Table row element

You use table row element to create rows in a table. The Table row element uses the <TR> tag to create a row. You can use <TR> tag within <TABLE> tags. <TR> tag used as container for the row.

The <TR> tag has the following attributes:

- align: used to specify the horizontal alignment of the contents for cells of a row.
- valign: used to specify the vertical alignment of the cell content for all cells of the row.
- bgcolor: used to specify the background color of the row.

Syntax:

```
<TR align="left | right | center" align="top | bottom | middle" bgcolor="#rrggbb"
> ..... </TR>
```

The Table data element

You use the Table data element to create data cells. The Table data element uses <TD> tag within the <TABLE> and <TR> tags to create data cells. A data cell only appears with in a table row.

The <TD> tag has the following **attributes**:

- colspan: used to specify the number of columns the cell can span.
- rowspan: used to specify the number of rows the cell can span
- align: used to specify the horizontal alignment of the data within a cell.
- valign: used to specify the vertical alignment of data within the cell.
- bgcolor: used to specify the background color of the cell.

Syntax:

```
<TD align="left | right | center" align="top | bottom | middle" bgcolor="#rrggbb"
colspan="n" rowspan="n"> ..... </TD>
```

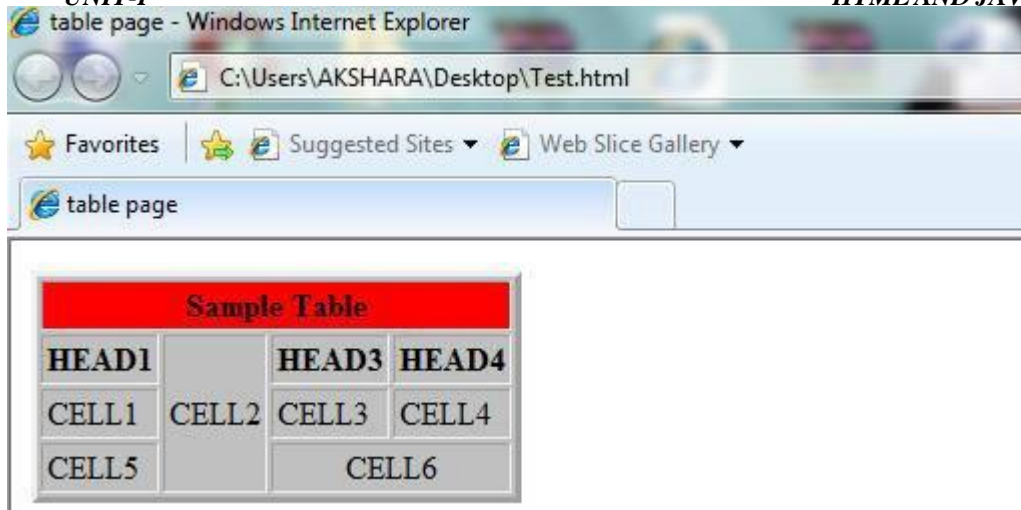
The Table heading element

You use the table heading element to create heading cells for table rows, columns, or both. The table heading element uses <TH> tag to create heading cells. The <TH> tag is exactly same as <TD> tag, except that the text displayed in bold. <TH> tag and <TD> tag share the same properties.

Example: HTML Program that prints table

```
<html>
<head> <title> table page</title> </head>
<body>
<table align="left" cellpadding="2" cellspacing="2" border="3" bgcolor="red">
<tr>
  <th colspan="4"> Sample Table</th>
</tr>
<tr>
  <th>HEAD1</th>
  <td rowspan="3">CELL2</td>
  <th>HEAD3</th>
  <th>HEAD4</th>
</tr>
<tr>
  <td>CELL1</td>
  <td> CELL3</td>
  <td> CELL4</td>
</tr>
<tr>
  <td>CELL5</td>
  <td colspan="2" align="center"> CELL6</td>
</tr>
</table>
</body> </html>
```

Output:



1.9 Frames:

Frames provide a pleasing interface which makes your web site easy to navigate. You can use frames in HTML page to manage the layout of the page. Frames divide the browser window into horizontal or vertical sections. Each section can be used to display a separate HTML page.

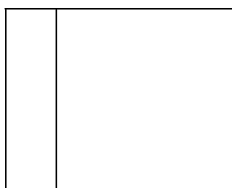
Each frame within a webpage has the following properties:

- ❖ It has unique name
- ❖ It displays an HTML document independent of other frames
- ❖ Its size can be dynamically changed according to the size of the content in HTML page

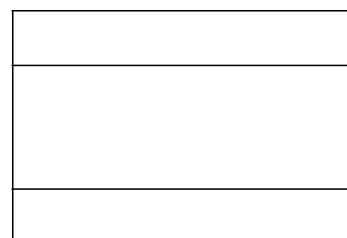
Creating frames

When we talk about frames actually we are referring to frameset, which is a special type of web page. Simply frameset is nothing but collection of frames. Web page that contains frame element is called framed page. Framed page begins with `<frameset>` tag and ends with `</frameset>`. Each individual frame is identified through `<frame>` tag. Creation of framed page is very simple. You can nest the framesets. First you decide how you want to divide your webpage and accordingly define frame elements.

Consider the following diagrams, first page divides into two columns and the second form divides into three rows.



Two columns frameset



Three rows frameset

The `<FRAMESET>` . . . `</FRAMESET>` Tags

The `<FRAMESET>` . . . `</FRAMESET>` tags are the container tags for all other tags that are used to create frames. These tags are used to specify the size and number of the frames to be created in an HTML Page. The frames within these tags are jointly referred to as a *frameset*.

Syntax:

```
<FRAMESET cols="n,n" rows="n,n">
```

```
<FRAME> Tags
```

```
</FRAMESET>
```

Attributes

COLS: The COLS attribute is used to specify the number of vertical frames within HTML Page.

ROWS: The ROWS attribute is used to specify the number of horizontal frames within HTML Page. To specify the values of the attributes, use a comma-separated list of values. The values can be of the following two types:

- **Value:** This numeric value is used to specify number of pixels that a frame will occupy in the browser window.
- **Value%:** This value is used to specify percentage of space that each frame will occupy in the browser window. The total of all the values specified in the comma-separated list is 100%.

In order to divide into two columns we can use the following syntax

```
<FRAMESET COLS="25%,75%">
<frame> tags
</FRAMESET>
```

In the second diagram we have three rows so by using rows parameter of frameset, we can divide logically the window into three rows.

```
<FRAMESET ROWS="20%,70%,10%">
<frame> tags
</FRAMESET>
```

According to above code, first row occupies 20% of the window, third row occupies 10% of the window, second row occupies 70% of the window.

<FRAME> Tag

You use the <FRAME> Tag to create frame with in *frameset*. You can use the following attributes to specify the name of the frame and the HTML page that you need to display in the frame.

- ❖ **SRC:** SRC attribute to specify the HTML page that you want to display in a frame.
- ❖ **NAME:** NAME attribute to specify the name of the frame. This *NAME* is used by the anchor element to display the linked HTML Pages within frame.
- ❖ **SCROLLING:** attribute used to add scrollbars to a frame. The SCROLLING attribute takes three value: YES, NO, AUTO.
 - The value *YES* specifies that a scrollbar should always be visible on the frame
 - The value *NO* specifies that the scrollbar should never be visible on the frame
 - The value *AUTO* specifies the browser to automatically display or remove scrollbars from a frame
- ❖ **FRAMEBORDER:** attribute to specify whether a frame should have a border. The value 0(zero) specifies no border. If you specify any other numeric value, the border is displayed.
- ❖ **NORESIZE:** By default, You can resize a frame by dragging its borders. You can restrict this by specifying the NORESIZE attribute.

Syntax:

```
<FRAME SRC = "URL" NAME =" myframe" SCROLLING = "yes | no | auto"
FRAMEBORDER = "0|1" [NORESIZE]/>
```

Nested Framesets

Some times it is required to divide your window into rows and columns, and then there is requirement of nested framesets. *Frameset with in another frameset is known as nested frameset.*

Example:

home.html

```
<frameset rows="20%,*">
  <frame name="fr1" src="top.html">
  <frameset cols="25%,*">
    <frame name="fr2" src="dept.html">
    <frame name="fr3" src="desc.html">
  </frameset>
</frameset>
```

top.html

```
<html>
  <body text="magenta">
    <center>
      <br/><h1> SACET::CHIRALA</h1>
    </center>
  </body></html>
```

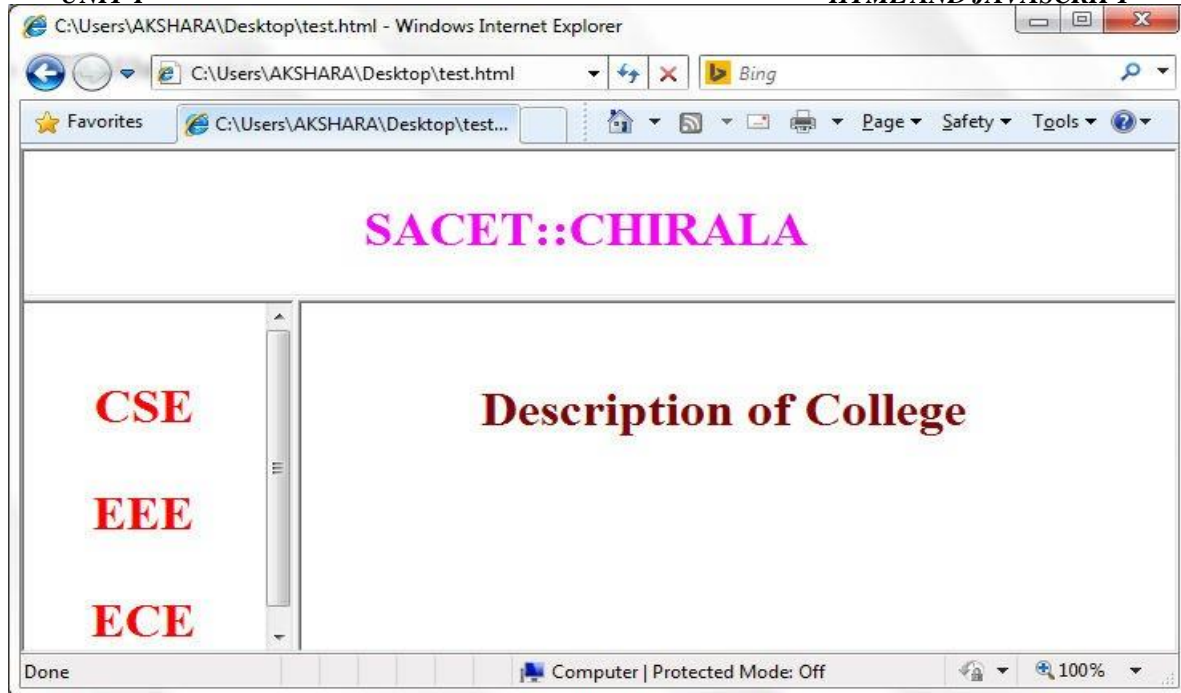
dept.html

```
<html>
  <body text="red">
    <center>
      <h1>
        <br/>CSE<br/>
        <br/>EEE<br/>
        <br/>ECE<br/>
      </h1>
    </body></html>
```

desc.html

```
<html>
  <body text="maroon">
    <center>
      <br/> <br/>
      <h1>Description of College</h1>
    </center>
  </body></html>
```

Output:



1.10 Forms

Forms are the best way of adding interactivity to a web page. A form is a collection of fields that you can use to gather information from people visiting your site. Forms acts as a means of user interactions on the Web.

Working of a Form

A Form contains certain text fields, radio buttons, check boxes and buttons for entering data. When data entry is complete, the user submits the form for processing using SUBMIT button. This communicates to the browser that the user has completed data entry. The browser sends data to the server. The server process the details and sends response to browser, which displayed on the browser.

The FORM Tag

A form is defined using the `<FORM>`. . . `</FORM>` tags. The FORM tag has four attributes:

1. NAME
2. ACTION
3. METHOD
4. ENCTYPE

Syntax:

```
<FORM action="URL" method = "POST" | "GET" enctype="encoding"> . . . </FORM>
```

NAME: used to specify the name of the form

ACTION: The ACTION attribute of the form tag is used to specify the name, and location of a script that will be used to process the data.

METHOD: The METHOD attribute of the form tag is used to specify the method by which browser sends the data to the server for processing. The type method can be either GET or POST. When *get* is used, the data is included as part of the URL. The *post* method encodes the data within the body of the message. Post can be used to send large amount of data, and it is more secure than *get*.

ENCTYPE: The ENCTYPE attribute is used to specify the content type for encoding the data. There are two content types:

1. application/x-www-form-urlencoded(default)
2. multipart/form-data.

ENCTYPE is used only when you need to upload a file.

The INPUT Element

The INPUT Element is used to specify input fields, such as single line text fields, passwords fields, check boxes, radio buttons etc.,. The INPUT element is defined by using <INPUT> tag.

You can create the following controls using the INPUT tag:

- ❖ Text
- ❖ Password
- ❖ Radio
- ❖ Checkbox
- ❖ Button
- ❖ Submit
- ❖ Reset

Syntax:

```
<INPUT type = "text" | "password" | "checkbox" | "radio" | "submit"  
name="string" value="string" size="n" maxlength="n" />
```

The tags used inside the form tag are:

In the above tag, the attribute *type* is used to implement text, password, checkbox, radio, submit, reset button and button.

Text: It is used to input the characters of the size n and if the value is given than it is used as a default value. It uses single line of text. Each component can be given a separate name using the name attribute.

Password: It works exactly as text, but the content is not displayed to the screen, instead * is displayed.

Radio: This creates a radio button that accepts single value from a set of alternatives. They are always grouped together with a same name but different values.

Checkbox: This creates a checkbox that enables you select multiple values from a set of alternatives. They are always grouped together with a same name but different values.

Submit: This creates a Submit button which displays the value attribute as its text. It is used to send the data to the server.

Reset: This creates a Reset button which displays the value attribute as its text. It is used to clear the form data.

Button: This creates a user-defined button which displays the value attribute as its text. It works according to user requirement.

The SELECT Element

The SELECT Element is a container element. This element is used to create a list box or drop-down list box.

Syntax:

```
<SELECT NAME="string" SIZE="n" MULTIPLE> . . .</SELECT>
```

Where

MULTIPLE: Indicates that the user is allowed to make several selections from list box

NAME: represents name of the list box

SIZE: Specifies the number of visible items in a list box.

The OPTION Element

It represents a choice in the list box and only occurs in SELECT element.

Syntax:

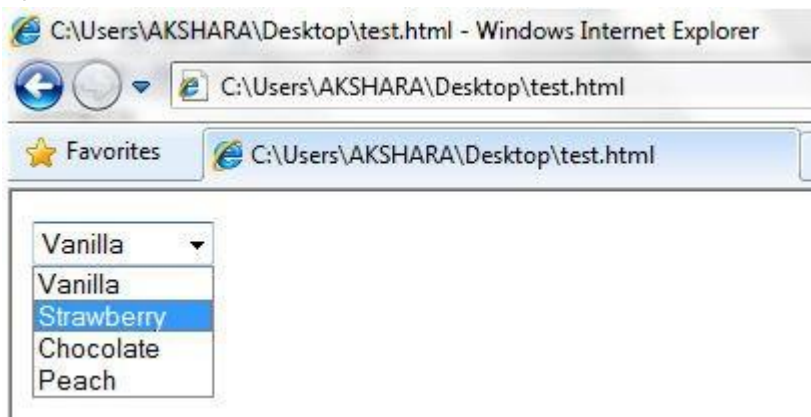
```
<OPTION VALUE="String" SELECTED>.....</OPTION>
```

SELECTED: indicates the option is initially selected

VALUE: value that indicates a particular option selected.

Example:

```
<html>
<body>
<form>
<select name="flavour">
  <option value="1" selected>Vanilla</option>
  <option value="2">Strawberry</option>
  <option value="3">Chocolate</option>
  <option value="4">Peach</option>
</select>
</body>
</html>
```



The TEXTAREA Element

This creates a multi-line plain text field into which the user can enter anything they like.

The area will be sized at rows by cols but supports automatic scrolling.

Syntax:

```
<TEXTAREA NAME="string" ROWS="n" COLS="n"> . . . </TEXTAREA>
```

Where

NAME: Specify the name of the text field

ROWS: Set the number of rows of text that will be visible

COLS: Set the number of columns of text that will be visible

Example : HTML code that implements forms

```
<html>
<head> <title>Student Data </title> </head>
```

```

<body>
<form>
<p><td>Name: <input type="text" maxlength="30" size="15"> <br/>
    Password: <input type="password" maxlenght=10 size=15> <br/>
    Qualification: <input type="checkbox" name="q" value="bt">B.Tech
    <input type="checkbox" name="q" value="mt">M.Tech <br/>
    Gender: <input type="radio" name="g" value="m">Male
        <input type="radio" name="g" value="f">Female <br/>
    Course: <select name="course" size="1">
        <option value="1" selected>CSE
        <option value="2">ECE
    </select> <br/>
    Address: <textarea name="addr" rows="5" cols="15"> </textarea> <br/>
    <input type="submit" name="s" value="Accept">
    <input type="reset" name="c" value="Ignore"> </p>
</form>
</body>
</html>

```

Output:

Student Data - Windows Internet Explorer

C:\Users\AKSHARA\Desktop\test.html

Student Data

Name: ch Vijay

Password: ●●●●●●

Qualification: ☒ B.Tech ☒ M.Tech

Gender: ☒ Male ☐ Female

Course: CSE

Address: CC

Accept Ignore

1.1.1 Cascading Style Sheets

CSS stands for Cascading Style Sheets. A style sheet is document that allows you to control the rendering, such as fonts, colors, typefaces and other aspects of style, of a web document. You can define style sheet properties within HTML document or in an external file.

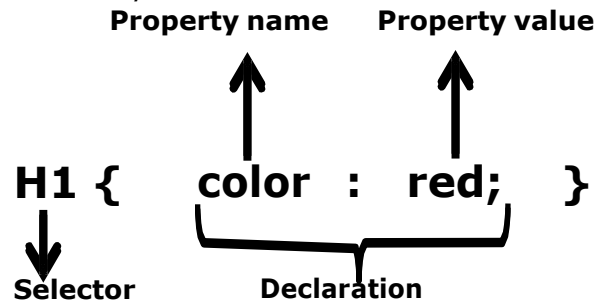
— A Style sheet is a collection of rules

Advantages:

1. Enables you to separate content from formatting.
2. Reduces download time by removing formatting information from the document.
3. Allows you to ensure a consistent appearance across a site
4. It supports reusability (Inheritance).

Defining a style:

A rule is a statement that defines a style.



A rule consists of two parts:

1. Selector
2. Declaration

A selector is an HTML statement that is linked to a specific style. A declaration defines the style for selector. A declaration again has two parts:

1. Property name
2. Property value

Property name is name of the property for which you need to define a style. Property value is the value assigned to property name.

Types of style sheets

You can define three types of style sheets:

1. Inline style sheets
2. Internal/Embedded style sheets
3. External style sheets

An inline sheet applies style to a particular element in a webpage. Embedding a style sheet is for defining styles to a single webpage. An external style sheet can apply styles to multiple web pages.

Inline style sheets

The style included within the tag is known as *inline style sheet*. To include style definition within tag, you use "STYLE" property.

Syntax:

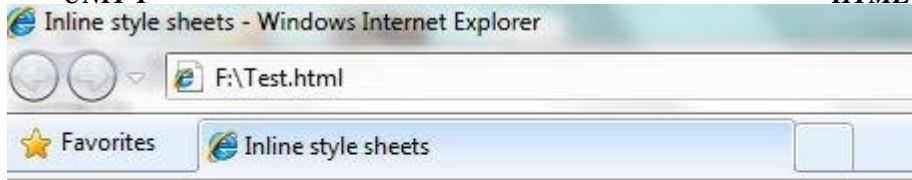
```
<TAG STYLE="propertyname: property value;". . . </TAG>
```

An Inline style affects only the element for which the style is defined. The style does not affect any other element even if the element is of the same type.

Consider the following code:

```
<html>
<head>
    <title> Inline style sheets </title>
</head>
<body>
    <P STYLE="font-family: Arial; font-style: italic; color: red;">This is paragraph with
inline style </P>
    <P>This is paragraph without style </P>
</body>
</html>
```

Output



This is paragraph with inline style

This is paragraph without style

Internal style sheets

In this, the style definitions are enclosed in `<STYLE> . . . </STYLE>` tags. In turn, this STYLE tag pairs must be placed in between `<HEAD> . . . </HEAD>` tags. The style defined in `<HEAD>` section is applied to whole document. The styles defined for the tags within `<STYLE>` tag will reflect for every instance of the tag in the entire document.

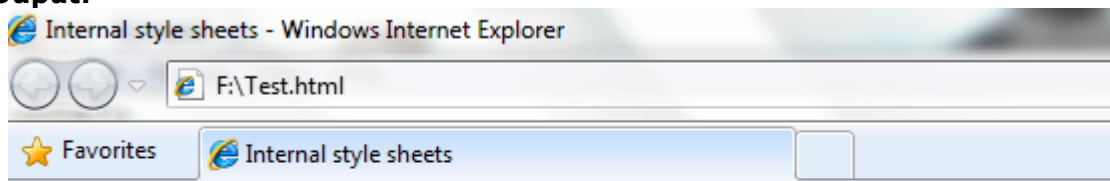
Syntax:

```
<HTML>
<HEAD>
  <STYLE TYPE="TEXT/CSS">
    <!-- STYLE DEFINITIONS -->
  </STYLE>
</HEAD>
<BODY> . . . </BODY>
</HTML>
```

Consider the following code that illustrates internal style sheets:

```
<html>
<head>
  <title> Internal style sheets </title>
  <STYLE>
    h1 {
      font-style:italic;
      color:red;
      background:yellow;
    }
  </STYLE>
</head>
<body>
  <h1>An Embedded style sheet</h1>
  <p> Embedded style sheet is a part of HTML document</p>
</body>
</html>
```

Output:



An Embedded style sheet

Embedded style sheet is a part of HTML document

External Style Sheets

Styles defined in the external files are called external style sheets which can be

used in any document by including them via URL. Use an external style sheet when you want to apply one style to many pages. If you made any change in an external style sheet, the change is universal on all the pages where the style sheet is used. An external style sheet is declared in an external file with a .css extension. It is called by pages whose interface it will affect.

Advantages of External CSS

1. The same style can be used by all of the WebPages in your website.
2. You can change the appearance of several pages by just altering the style sheet rather each individual page.
3. A Style sheet can acts as style template
4. A style sheet can import and use styles from other style sheets, making modular development and good reuse.
5. Reduces the download time by removing formatting information from the document.

<LINK> Tag

External style sheets are called using the `<link>` tag which should be placed in the head section of an HTML document. This tag takes three attributes.

Attributes of the <link> tag:

- ❖ **rel** - When using an external stylesheet on a webpage, this attribute takes the value "stylesheet"
- ❖ **type** - When using an external stylesheet on a webpage, this attribute takes the value "text/css"
- ❖ **href** - Denotes the name and location of the external style sheet to be used.

Syntax:

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

Consider the following code that illustrates external style sheets:

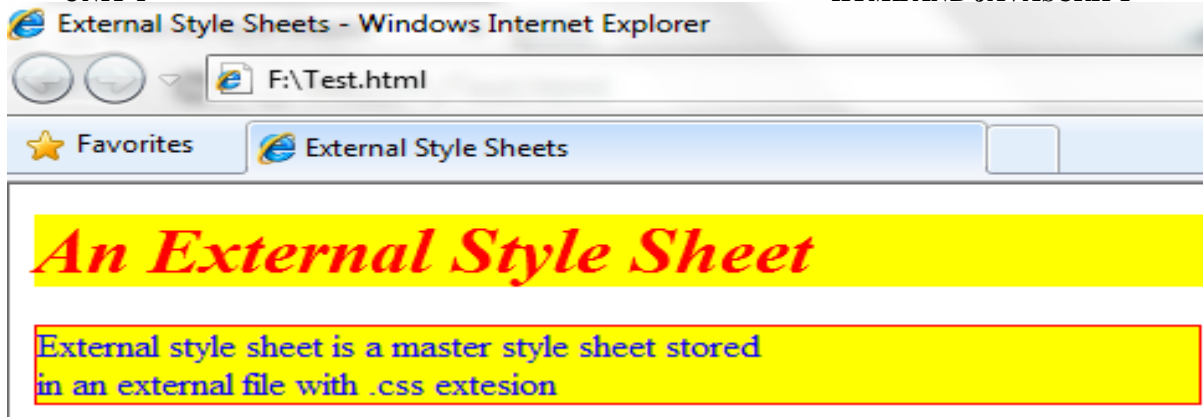
styles.css

```
h1
{
    font-style:italic;
    color:red;
    background:yellow;
}
p
{
    color:blue;
    background:yellow;
    font-size:12pt;
    border:1px solid red;
}
```

Test.html

```
<html>
<head>
<title> External style sheets </title>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<h1>An External style sheet</h1>
<p> External style sheet is a master style sheet stored <br/>in an external file
with .css extension</p>
</body>
</html>
```

Output



Class selectors

Class selectors are used, when you want different elements to share the same format. A class selector definition starts with a period (.) followed by user defined and then style definition.

Syntax:

```
<style>
  .myname
  {
    Style definitions
  }
</style>
```

1.11 Introduction JavaScript

Script means small piece of code. Scripting languages are two kinds one is client-side other one is servers-side scripting. In general client-side scripting is used for verifying simple validation at client side, server-side scripting is used for database verifications. VBScript, java script and J script are examples for client-side scripting and ASP, JSP, Servlets, PHP etc. are examples of server-side scripting.

JavaScript (originally known as "LiveScript") is a scripting language that runs inside the browser to manipulate and enhance the contents of Web pages. Java Script is designed to add interactivity to HTML pages. Web pages are two types

5. Static web page
 6. Dynamic webpage
- ❖ Static web page where there is no specific interaction with the client
 - ❖ Dynamic web page which is having interactions with client and as well as validations can be added.

Simple HTML script is called static web page, if you add script to HTML page it is called dynamic page. Netscape navigator developed java script. Microsoft's version of JavaScript is Jscript.

- ❖ Java script code as written between <script>-----</script>tags
- ❖ All java script statements end with a semicolon
- ❖ Java script ignores white space
- ❖ Java script is case sensitive language
- ❖ Script program can save as either. Js or. html

Benefits of JavaScript

- ❖ It is widely supported by web browsers;
- ❖ It gives easy access to the document objects and can manipulate most of them.
- ❖ Java Script gives interesting animations with long download times associated with many multimedia data types;
- ❖ Web surfers don't need a special plug-in to use your scripts
- ❖ Java Script relatively secure - you can't get a virus infection directly from Java Script.
- ❖ JavaScript code resembles the code of C Language; the syntax of both the language is very close to each other. The set of tokens and constructs are same in both the language.

Problems with JavaScript

- ❖ Most scripts rely upon manipulating the elements of DOM;
- ❖ Your script does not work then your page is useless
- ❖ Because of the problems of broken scripts many web surfers disable java script support in their browsers
- ❖ Script can run slowly and complex scripts can take long time to start up

Similarities between java script and java:

1. Both java script and java having same kind of operators
2. Java script uses similar control structures of java
3. Nowadays both are used as languages for use on internet.
4. Labeled break and labeled continue both are similar
5. Both are case-sensitive languages

Difference between java script and java:

Java Script	Java
Java Script is scripting language	Java is a programming language
Java Script is object-based programming language.	Java is object –oriented programming language
Java script code is not compiled, only interpreted.	Java is compiled as well as interpreted language
Java Script is Weekly-typed language	Java is Strongly-typed language
JavaScript code is run on a browser only.	Java creates applications that run in a virtual machine or browser

The syntax of the script tag is as follows:

```
<script language=""scripting language name"">
```

```
</script>
```

The language attribute specifies the scripting language used in the script. Both Microsoft internet explorer and Netscape navigator use java script as the default scripting language. The script tag may be placed in either the head or the body or the body of an HTML document.

Ex: <script language=""javascript"">

```
</script>
```

Comments in JavaScript:

Single line comment- //
Multi-line comment- <!-- comment -->

Operators in JavaScript:

- ❖ Arithmetic operators (+, -, *, /, %)
- ❖ Relational operators (<, >, !=, <=, >=)
- ❖ Logical operators(&&, ||, !)
- ❖ Assignment operator(=)
- ❖ Increment decrement operators(++ , - -)
- ❖ Conditional/Ternary operator(?:)
- ❖ Bitwise operators(&, |, !)

Control structures:

- ❖ If statement
- ❖ Switch
- ❖ While
- ❖ Do-while
- ❖ For
- ❖ Break

❖ Continue

Control structures syntax and working as same as java language.

1.12 Variables

Variables are like storage units/place holders to hold values. A variable is a memory location to hold certain different types of data. In Javascript, A variable can store all kinds of data. It is important to know the proper syntax to which variables must conform:

- ❖ They must start with a letter or underscore (" _")
- ❖ Subsequent characters can also be digits (0-9) or letters (A-Z and/or a-z). Remember, JavaScript is case-sensitive. (That means that MyVar and myVar are two different names to JavaScript, because they have different capitalization.)
- ❖ You cannot use reserved words as variable names.
- ❖ You cannot use spaces in names.
- ❖ Names are case-sensitive.

Syntax:

```
var v_name = value;
```

Examples of legal variable names are fname, temp99, and _name.

When you declare a variable by assignment outside of a function, it is called global variable, because it is available everywhere in the document. When you declare a variable within a function, it is called local variable, because it is available only within the function. To assign a value to a variable, you use the following notation:

```
var num = 8;
var real= 4.5;
var myString = "Web Technologies";
```

Values of Variables(Data types)

JavaScript recognizes the following types of values:

- ❖ Numbers, such as 42 or 3.14159
- ❖ Boolean values, either true or false
- ❖ Strings, such as "Howdy!"
- ❖ NULL, a special keyword which refers to nothing.

1.13 Functions

Functions are one of the fundamental building blocks in JavaScript. A function is a JavaScript procedure -a set of statements that performs a specific task when called. A function definition has these basic parts:

- ❖ The *function* keyword
- ❖ A function name
- ❖ A comma-separated list of arguments to the function in parentheses
- ❖ The statements in the function in curly braces: { }

Defining a Function

Defining the function means, name the function and specifies what to do when the function is called. You define a function within the <SCRIPT>...</SCRIPT> tags within the <HEAD> ... </HEAD> tags. While defining a function, you can also declare the variables which you will be calling in that function. Here's an example of *defining* a function:

```
function msg()
{
    window.alert("This is an alert box.");
}
```

Here's an example of a function that takes a parameter:

```
function welcome(string)
{
    window.alert("Hi"+string);
}
```

When you call this function, you need to pass a parameter (such as the word that the user clicked on) into the function.

Calling a Function

Calling the function actually performs the specified actions. When you call a

function, this is usually within the BODY of the HTML page, and you usually pass a parameter into the function on which the function will act.

Here's an example of calling the same function:

```
msg();
```

For the other example, this is how you may call it:

```
<input type="button" name="welcome" onClick="msg1("Vijay")"/>
```

1.14 OBJECTS IN JAVA SCRIPT

When you load a document in your Web browser, it creates a number of JavaScript objects with properties and capabilities based on the HTML in the document and other information. These objects exist in a hierarchy that reflects the structure of the HTML page itself. The pre-defined objects that are most commonly used are the window and document objects. Some of the useful Objects are:

1. Document
2. Window
3. Browser
4. Form
5. Math
6. Date

1.15.1 The Document Object

A document is a web page that is being either displayed or created. The document has a number of properties that can be accessed by JavaScript programs and used to manipulate the content of the page.

write or writeln

Html pages can be created on the fly using JavaScript. This is done by using the write or writeln methods of the document object.

Syntax:

```
document.write("String");
document.writeln("String");
```

In this document is object name and write () or writeln () are methods. Symbol period is used as connector between object name and method name. The difference between these two methods is carriage form feed character that is new line character automatically added into the document.

Exmaple: document.write("<body>");
 document.write("<h1> Hello </h1>");

bgcolor and fgcolor

These are used to set background and foreground(text) color to webpage. The methods accept either hexadecimal values or common names for colors.

Syntax:

```
document.bgcolor="#1f9de1";
document.fgcolor="silver";
```

anchors

The anchors property is an array of anchor names in the order in which they appear in the HTML Document. Anchors can be accessed like this:

Syntax:

```
document.anchors[0];
:
document.anchors[n-1];
```

Links

Another array holding all links in the order in which they were appeared on the Webpage

Forms

Another array, this one contains all of the HTML forms. By combining this array with the individual form objects each form item can be accessed.

1.15.2 The Window Object

The window object is used to create a new window and to control the properties of window.

Methods:

1. `open("URL","name")` : This method opens a new window which contains the document specified by URL and the new window is identified by its name.
2. `close()`: this shutdowns the current window.

Properties:

`toolbar = [1|0]` `location = [1|0]` `menubar = [1|0]` `scrollbars = [1|0]` `status = [1|0]` `resizable = [1|0]`

where as 1 means *on* and 0 means *off*

`height=pixels`, `width=pixels` : These properties can be used to set the window size.

The following code shows how to open a new window

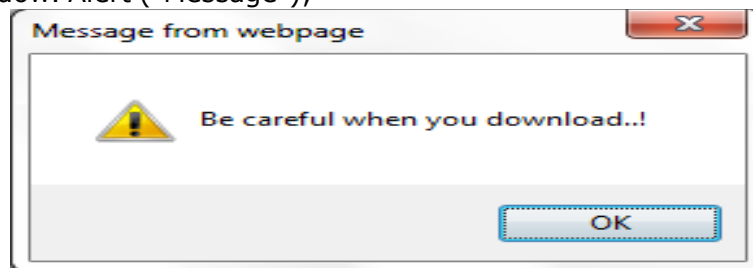
```
newWin = open("first.html","newWin","status=0,toolbar=0,width=100,height=100");
```

Window object supports three types of message boxes.

1. Alert box
2. Confirm box
3. Prompt box

Alert box is used to display warning/error messages to user. It displays a text string with *OK* button.

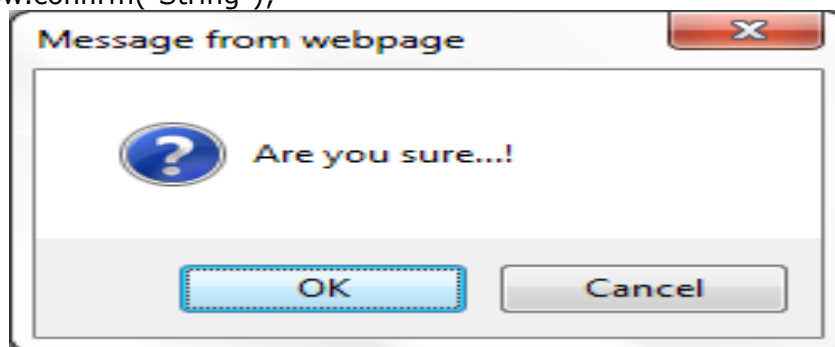
Syntax: `window.Alert("Message");`



Confirm Box is useful when submitting form data. This displays a window containing message with two buttons: *OK* and *Cancel*. Selecting *Cancel* will abort the any pending action, while *OK* will let the action proceed.

Syntax

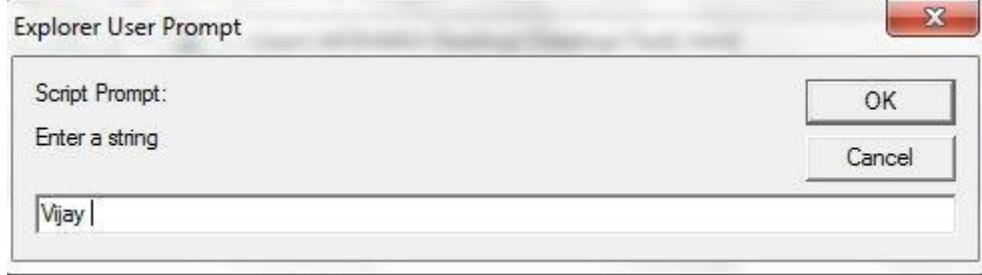
`window.confirm("String");`



Prompt box used for accepting data from user through keyboard. This displays simple window that contains a prompt and a text field in which user can enter data. This method has two parameters: a text string to be used as a prompt and a string to use as the default value. If you don't want to display a default then simply use an empty string.

Syntax

`Variable=window.prompt("string","default value");`



1.15.3 The Form Object

Two aspects of the form can be manipulated through JavaScript. First, most commonly and probably most usefully, the data that is entered onto your form can be checked at submission. Second you can actually build forms through JavaScript. Form object supports three events to validate the form

onClick = "method()"

This can be applied to all form elements. This event is triggered when the user clicks on the element.

onSubmit = "method()"

This event can only be triggered by form itself and occurs when a form is submitted.

onReset = "method()"

This event can only be triggered by form itself and occurs when a form is reset.

Example: HTML program that applies a random background color when you click on button

```
<html>
<head>
<script language = "javascript">
    function change()
    {
        var clr = document.bgColor=parseInt(Math.random()*999999);
        document.f1.color.value=clr;
    }
</script>
</head>
<body>
<form name="f1">
    <input type="text" name="color"/>
    <input type="button" value="Click me" onclick="change()"/>
</form> </body> </html>
```

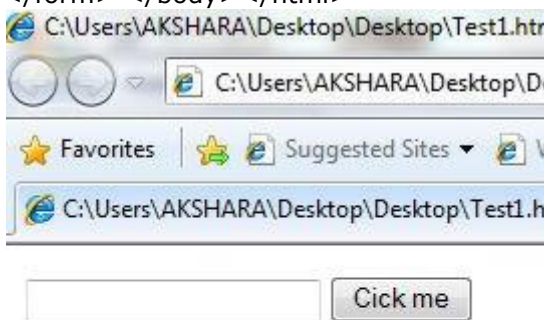


Fig.1: On first run background is white

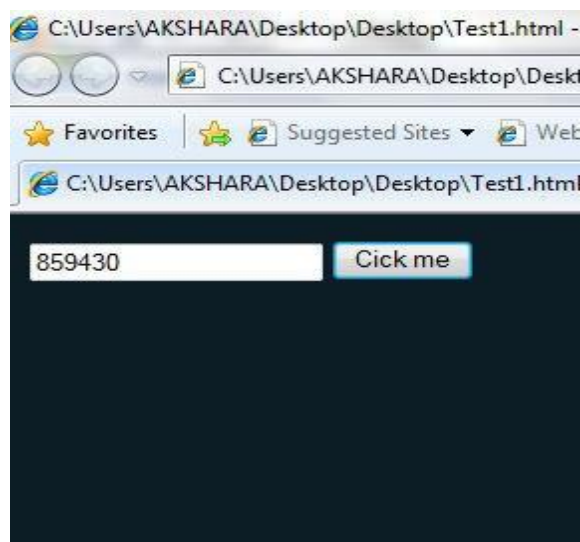


Fig.2: After clicking on button "Click me": background changed to "black"

1.15.4 The browser Object

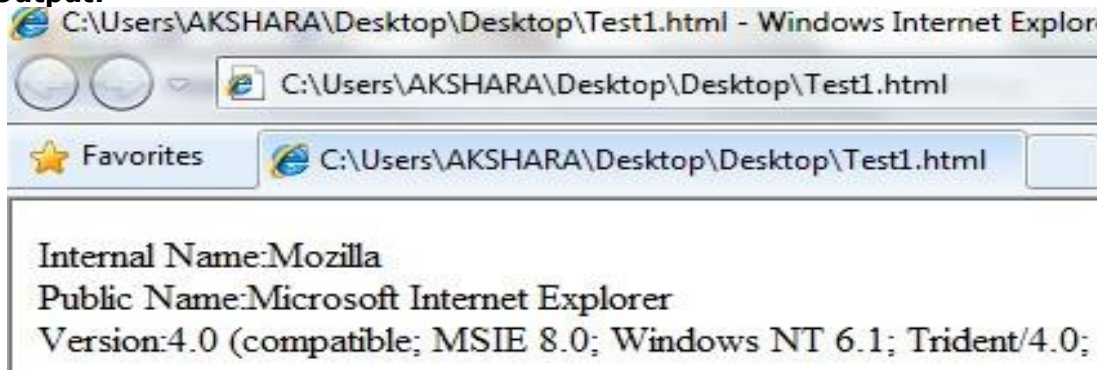
The browser is JavaScript object that can be used to know the details of browser. Some of the properties of the browser object is as follows:

Property	Description
<code>navigator.appCodeName</code>	It returns the internal name for the browser. For major browsers it is <i>Mozilla</i>
<code>navigator.appName</code>	It returns the public name of the browser – navigator or Internet Explorer
<code>navigator.appVersion</code>	It returns the version number, platform on which the browser is running.
<code>navigator.userAgent</code>	The strings <code>appCodeName</code> and <code>appVersion</code> concatenated together
<code>navigator.plugins</code>	An array containing details of all installed plug-ins
<code>Navigator.mimeTypes</code>	An array of all supported MIME Types

Example: Write javascript to display internal details of a browser (**Test.html**)

```
<script language = "javascript">
<!--
    document.writeln("Internal    Name:"+navigator.appCodeName);
    document.writeln("<br/>Public    Name:"+navigator.appName);
    document.writeln("<br/>Version:"+navigator.appVersion);
-->
</script>
```

Output:



1.15.5 The Math Object

The *Math* object holds all mathematical functions and values. All the functions and attributes used in complex mathematics must be accessed via this object.

Syntax:

```
Math.methodname();
Math.value;
```

Method	Description	Example
<code>Math.abs(x)</code>	Returns the absolute value	<code>Math.abs(-20)</code> is 20
<code>Math.ceil(x)</code>	Returns the ceil value	<code>Math.ceil(5.8)</code> is 6 <code>Math.ceil(2.2)</code> is 3
<code>Math.floor(x)</code>	Returns the floor value	<code>Math.floor(5.8)</code> is 5 <code>Math.floor(2.2)</code> is 2
<code>Math.round(x)</code>	Returns the round value, nearest integer value	<code>Math.round(5.8)</code> is 6 <code>Math.round(2.2)</code> is 2
<code>Math.trunc(x)</code>	Removes the decimal places it returns only integer value	<code>Math.trunc(5.8)</code> is 5 <code>Math.trunc(2.2)</code> is 2
<code>Math.max(x,y)</code>	Returns the maximum value	<code>Math.max(2,3)</code> is 3 <code>Math.max(5,2)</code> is 5
<code>Math.min(x,y)</code>	Returns the minimum value	<code>Math.min(2,3)</code> is 2 <code>Math.min(5,2)</code> is 2
<code>Math.sqrt(x)</code>	Returns the square root of x	<code>Math.sqrt(4)</code> is 2

UNIT-I		HTML AND JAVASCRIPT
Math.pow(a,b)	This method will compute the a^b	Math.pow(2,4) is 16
Math.sin(x)	Returns the sine value of x	Math.sin(0.0) is 0.0
Math.cos(x)	Returns cosine value of x	Math.cos(0.0) is 1.0
Math.tan(x)	Returns tangent value of x	Math.tan(0.0) is 0
Math.exp(x)	Returns exponential value i.e e^x	Math.exp(0) is 1
Math.random(x)	Generates a random number in between 0 and 1	Math.random()
Math.log(x)	Display logarithmic value	Math.log(2.7) is 1
Math.PI	Returns a π value	a = Math.PI; a = 3.141592653589793

1.15.6 The Date Object

This object is used for obtaining the date and time. In JavaScript, dates and times represent in milliseconds since 1st January 1970 UTC. JavaScript supports two time zones: UTC and local. UTC is Universal Time, also known as Greenwich Mean Time(GMT), which is standard time throughout the world. Local time is the time on your System. A JavaScript *Date* represents date from -1,000,000,000 to 1,000,000,000 days relative to 01/01/1970.

Date Object Constructors:

new Date(); Constructs an empty date object.

new Date("String"); Creates a Date object based upon the contents of a text string.

new Date(year, month, day[,hour, minute, second]); Creates a Date object based upon the numerical values for the year, month and day.

```
var dt=new Date();
```

```
document.write(dt); // Tue Dec 23 11:23:45 UTC+0530 2015
```

Methods in Date object:

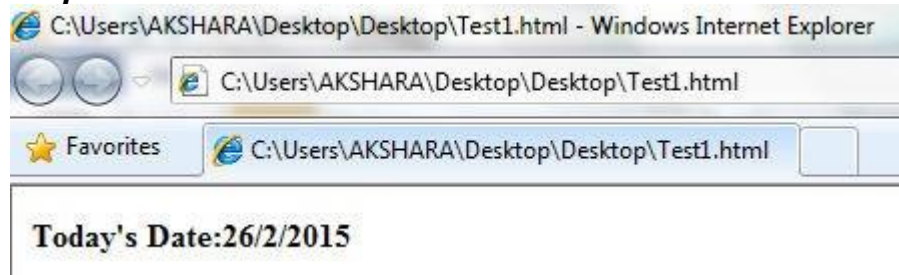
JavaScript date object provides several methods, they can be classified as string form, get methods and set methods. All these methods are provided in the following table.

Method	Description
getDate()	Returns day of the month i.e. 1 to 31
getDay()	Returns an integer representing day of the week(0 - 6), Sunday to Saturday respectively.
getMonth()	Returns month of the year from 0 to 11, January to December respectively.
getFullYear()	Returns four-digit year number
getHours()	Returns hour field of the Date Object in 24 hours time format (0 to 23)
getMinutes()	Returns minute field of the Date Object from 0 to 59
getSeconds()	Returns seconds field of the Date Object 0 to 59
setDate(v)	To set the date, day, month and full year
setDay(v)	
setMonth(v)	
setFullYear(y,m,d)	
setHours(v)	To set the hours, minutes, seconds of time
setMinutes(v)	
setSeconds(v)	
toString()	Returns the Date as a string

Example: Write javascript to display internal details of a browser (**Test.html**)

```
<script language = "javascript">
<!--
var dt = new Date(); var day = dt.getDate();
var month = dt.getMonth()+1; var year = dt.getFullYear();
document.writeln("<h4>Today's Date:"+day+"/"+month+"/"+year);-
-->
```

</script>

Output:**1.16 DYNAMIC HTML**

DHTML is combination of HTML, CSS and JavaScript. It gives pleasant appearance to web page.

Difference between HTML and DHTML

HTML	DHTML
HTML is used to create static web pages.	DHTML is used to create dynamic web pages.
HTML is consists of simple html tags.	DHTML is made up of HTML tags+cascading style sheets+javascript.
Creation of html web pages is simplest but less interactive.	Creation of DHTML is complex but more interactive.