UNIT-2 DESIGN AND DEVELOP WEB PAGES

2.1 Basic web technologies

- **Browser**
- **Web–Server**
- **Client-Server Model**
- **URL**
- **SEO techniques**
- **Domain names and domain name system.**

2.2 Creating Web-pages with HTML5 - Static web pages.

- **Introduction, Editors**
- **Tags, Attributes, Elements, Headings**
- **Links, Images, List, Tables, Forms**
- **Formatting, Layout, Iframes.**

2.3 Formatting web pages with style sheets (CSS3).

- **Introduction to CSS**
- **Inline CSS, Internal CSS, Classes and IDs**
- **Div, Color, Floating, Positioning**
- **Margins, Padding, Borders**
- **Fonts, Aligning Text, Styling Links**

2.4 Creating a webpage dynamic using JavaScript.

- **Dynamic web page and Introduction to JS**
- **Basic syntax**
- **Functions**
- **Events**

2.6 Creating dashboards in websites

Basic web technologies

Web browser:

- Web browser is program (Application software) that "requests" document from web server.
- Web browser is a software application for retrieving, presenting and traversing information resources on the www.
- Web browser runs on the client machine.
- They are called browsers because they allow the user to browse the resource available on the server.
- Server locates the document and sends it to the browser which displays it for the user.
- Mosaic was the browser with a GUI (1983)

  Ex: Mozilla, Chrome, Internet Explore...etc

Web server:

- Web server is a program that provides or serves documents to the requesting browsers.
- Web server runs on server machines.
- Ex: Apache (Open source), IIS (Microsoft)
- All communication between browser and server uses HTTP
- A web client (or) browser opens a network communication to a web server, sends information requests to the server, receives information from the server and closes the connections.

Client-Server Model:

Client- A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

Server- A server is a program that runs on the remote machine providing services to the clients. When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.

URL:

- URL is used to identify different kinds of resources or documents on the internet.
- If the web browser wants some document from the web server just giving domain name is not sufficient because domain name can only be used for locating the server. Therefore URL's should be provided.
- The general format of URL is given below:-

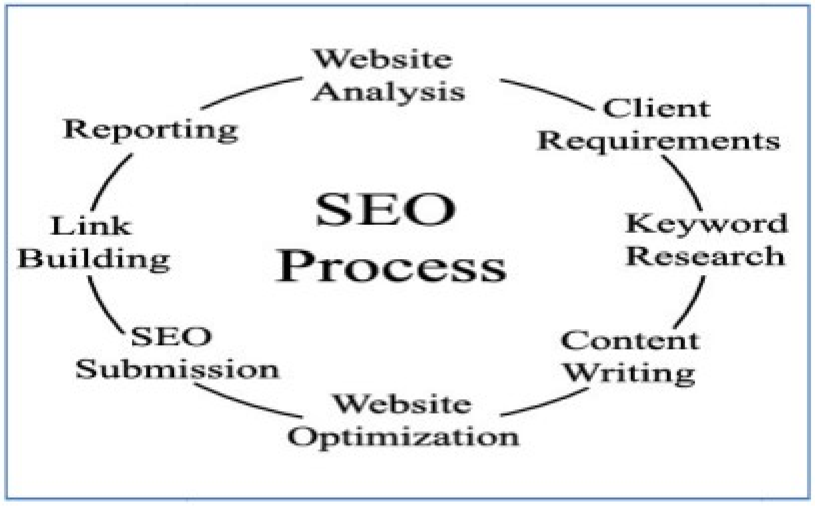
  Scheme: Object-address

  Ex: http://www.dte.kar.nic.in/college.php

- Ex: http://www.dte.kar.nic.in/index.html

SEO Techniques:

- SEO stands for Search Engine Optimization.
- It is a process designed to optimize a website for search engines.
- It helps the websites to achieve a higher ranking in search engine results when people search for keywords related to their products and services.



WEBSITE
RANKING
BACK LINKS
SEO
DATA ANALYSIS
SEARCH ENGINE OPTIMIZATION
KEYWORDS
TRAFFIC
CONTENT
SITE ARCHITECTURE

Domain Names and Domain Name System:

IP Address:

- IP address is a unique logical address assigned to a machine over the network.
- An IP address exhibits the following properties:
  - IP address is the unique address assigned to each host present on Internet.
  - IP address is 32 bits 4bytes long.
  - IP address consists of two components: network component and host component.

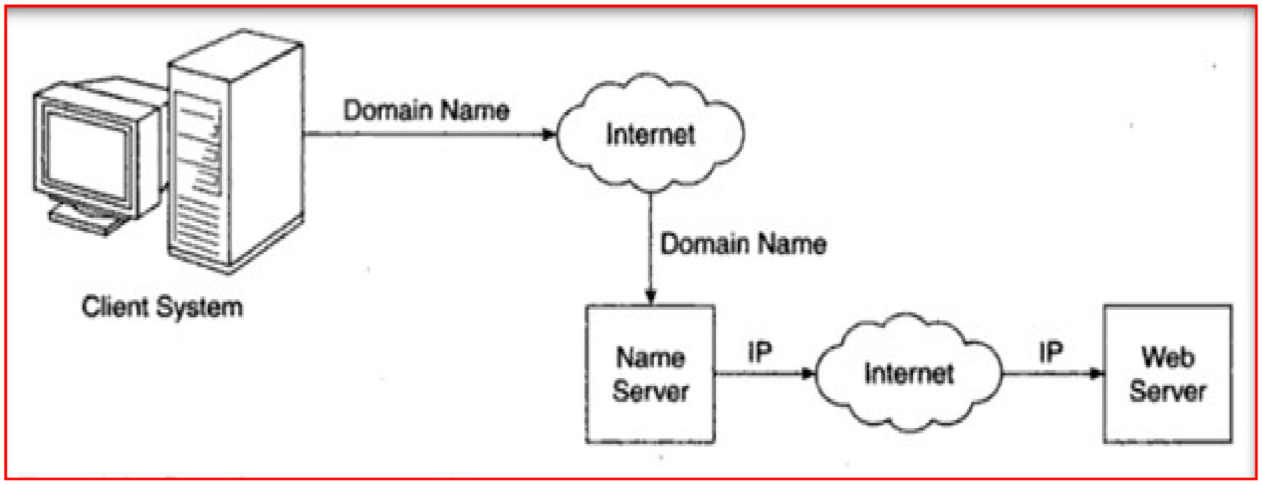
- ➢ Each of the 4 bytes is represented by a number from 0 to 255, separated with dots
- For example **192.168.1.110**

Domain Names:

- The IP addresses are numbers. Hence, it would be difficult for the users to remember IP address. To solve this problem, text based names were introduced. These are technically known as domain name system (DNS).
- These names begin with the names of the host machine, followed by progressively larger enclosing collection of machines, called domains. There may be two, three or more domain names.
- DNS is of the form **hostname.domainName.domainName**.
- Ex: https://dtek.karnataka.gov.in/, gptkoppal.in

Domain Name Conversion:

- Domain name has to be converted to IP address before destination is reached.
- Conversion is needed because computer understands only numbers.
- Conversion is done with the help of name servers.
- As soon as the domain name is provided, it will send across the internet to contact server.
- The name server is responsible for converting domain name to IP-address.
- If one of the name server is not able to convert domain name to IP-address it contact the another name server.
- This process continues until the IP-address is generated, Once the IP address is generated the host can be accessed.



Creating Web Pages using HTML5

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

Static Web pages

- Web page is a document available on World Wide Web. Web Pages are stored on web server and can be viewed using a web browser.
- A web page can contain huge information including text, graphics, audio, video and hyper links.
- Static web pages are also known as flat or stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can't do any modification or interact with the information.
- Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.

HTML Elements

An HTML element is defined by a start tag, some content, and an end tag.

The HTML **element** is everything from the start tag to the end tag:

- <tagname>Content</tagname>

Examples of some HTML elements:

- <h1>My First Heading</h1>
- <p>My first paragraph.</p>

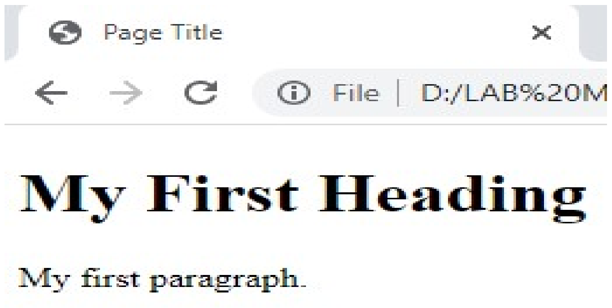
HTML elements can be nested (this means that elements can contain other elements).

All HTML documents consist of nested HTML elements.

The following example contains four HTML elements (<html>, <body>, <h1> and <p>):

```
<html>
<head>
<title>Page Title</title>
</head>
<body>
```

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



My First Heading

My first paragraph.

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

- HTML elements with no content are called **empty elements**.
- The <br> tag defines a line break, and is an empty element without a closing tag:
- <p>This is a <br> paragraph with a line break.</p>
- HTML tags are **not case sensitive**: <P> means the same as <p>.

HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

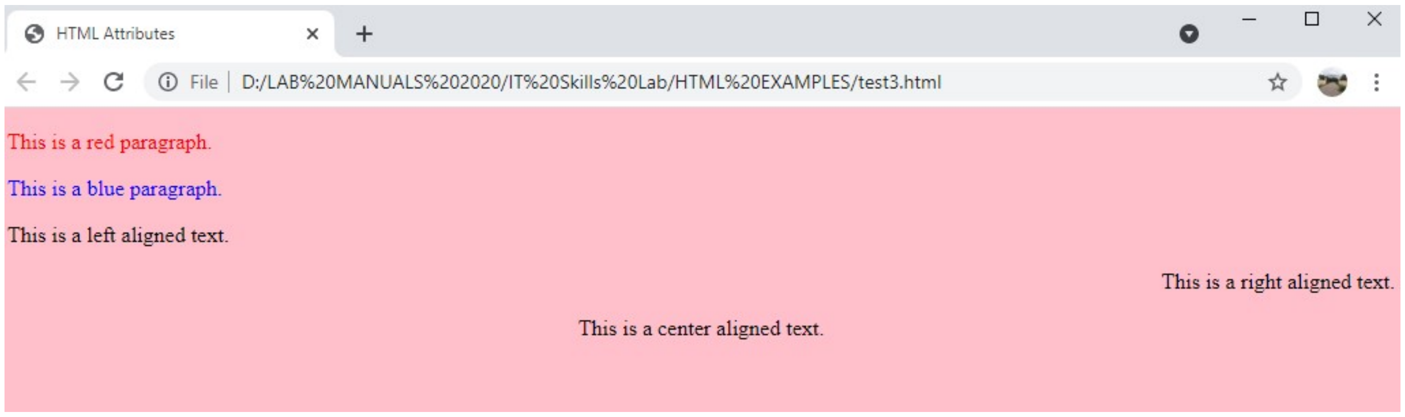
<element attribute_name="value">content</element>

The style attribute is used to add styles to an element, such as color, font, size, and more.

```
<p style="color:red;">This is a red paragraph.</p>
```

Example:

```
<html>
<head>
<title>HTML Attributes</title>
</head>
<body bgcolor="pink">
<p style="color:red;">This is a red paragraph.</p>
<p style="color:blue;">This is a blue paragraph.</p>
<p align="left">This is a left aligned text.</p>
<p align="right">This is a right aligned text.</p>
<p align="center">This is a center aligned text.</p>
</body>
</html>
```


HTML Attributes
File | D:/LAB%20MANUALS%202020/IT%20Skills%20Lab/HTML%20EXAMPLES/test3.html

```
This is a red paragraph.
This is a blue paragraph.
This is a left aligned text.
                                                    This is a right aligned text.
                          This is a center aligned text.
```

HTML headings are titles or subtitles that we want to display on a webpage.

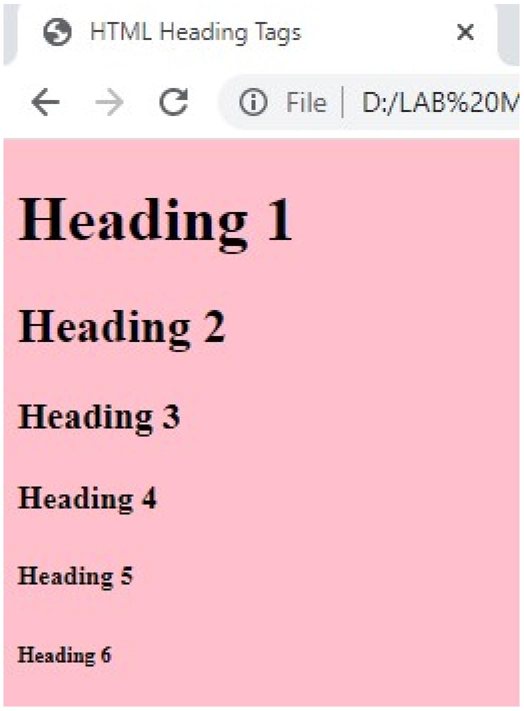
HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Example:

```
<html>
<head>
<title>HTML Heading Tags </title>
</head>
<body bgcolor="pink">
```

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>
```



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

HTML Paragraph

The HTML **<p>** element defines a paragraph.

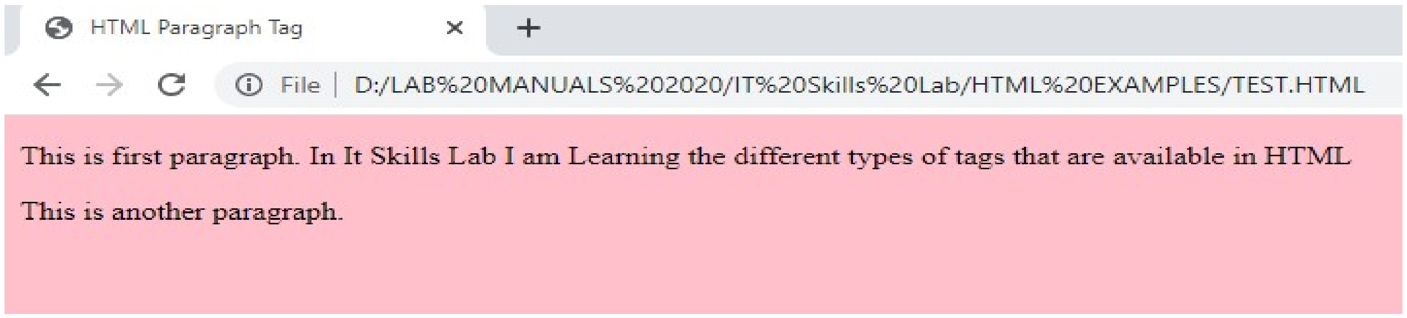
A paragraph always starts on a new line, and browsers automatically add some white space before and after a paragraph.

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

Example:

```
<html>
<head>
<title>HTML Paragraph Tag </title>
</head>
```

```
<body  bgcolor="pink">
<p>This is first paragraph. In It Skills Lab I am Learning the different types of tags that are available in
HTML</p>
<p>This is another paragraph. </p>
</body>
</html>
```


HTML Paragraph Tag
File
D:/LAB%20MANUALS%202020/IT%20Skills%20Lab/HTML%20EXAMPLES/TEST.HTML

This is first paragraph. In It Skills Lab I am Learning the different types of tags that are available in HTML

This is another paragraph.

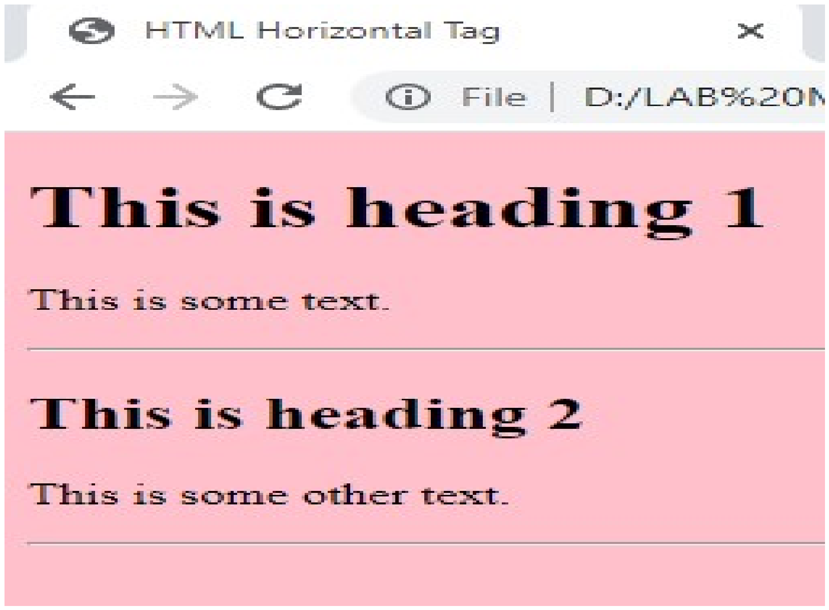
HTML Horizontal Rules

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> element is used to separate content (or define a change) in an HTML page:

Example:

```
<html>
<head>
<title>HTML Horizontal Tag </title>
</head>
<body bgcolor="pink">
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
</body>
</html>
```



This is heading 1

This is some text.

This is heading 2

This is some other text.

HTML Line Breaks

The HTML <br> element defines a line break.

Use <br> if you want a line break (a new line) without starting a new paragraph:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

HTML <pre> Element

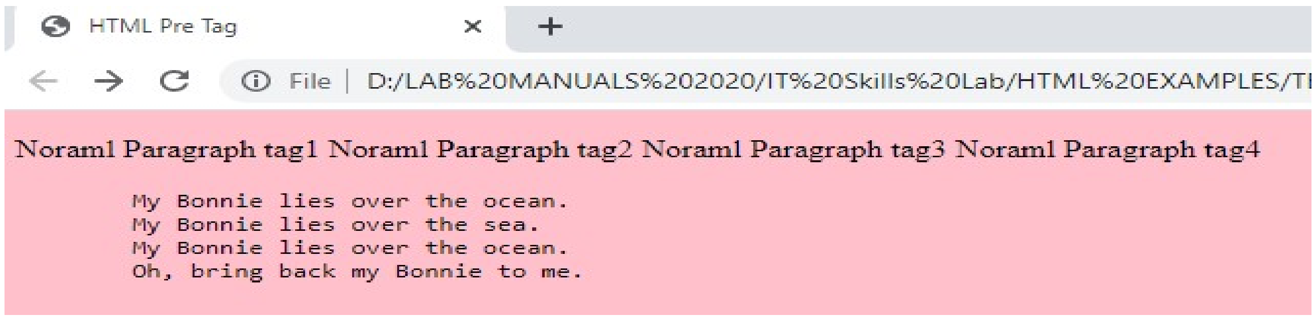
The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

Example:

```
<html>
<head>
<title>HTML Pre Tag </title>
</head>
<body bgcolor="pink">
<p> Noraml Paragraph tag1
Noraml Paragraph tag2
Noraml Paragraph tag3
Noraml Paragraph tag4 </p>
<pre>
        My Bonnie lies over the ocean.
        My Bonnie lies over the sea.
```

```
    My Bonnie lies over the ocean.
    Oh, bring back my Bonnie to me.
</pre>
</body>
</html>
```


HTML Pre Tag
File | D:/LAB%20MANUALS%202020/IT%20Skills%20Lab/HTML%20EXAMPLES/TI

Noraml Paragraph tag1 Noraml Paragraph tag2 Noraml Paragraph tag3 Noraml Paragraph tag4

```
        My Bonnie lies over the ocean.
        My Bonnie lies over the sea.
        My Bonnie lies over the ocean.
        Oh, bring back my Bonnie to me.
```

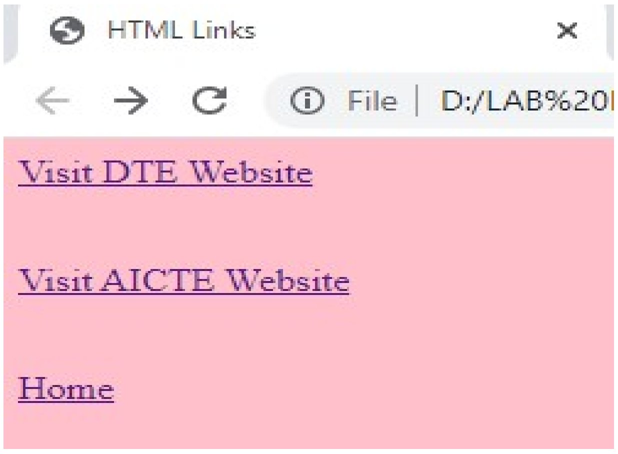
HTML links/hyperlinks

- HTML links are hyperlinks. We can click on a link and jump to another document.
- When you move the mouse over a link, the mouse arrow will turn into a little hand.
- A link does not have to be text. A link can be an image or any other HTML element!
- A link is specified using HTML tag **<a>.** This tag is called **anchor** tag and anything between the opening tag and the closing tag becomes part of the link and a user can click that part to reach to the linked document.
- The most important attribute of the <a> element is the **href** attribute, which indicates the link's destination

  <a href="https://dtek.karnataka.gov.in/">Visit DTE Website</a>

Example:

```
<html>
<head>
<title>HTML Links</title>
</head>
<body bgcolor="pink">
<a href="https://dtek.karnataka.gov.in/">Visit DTE Website</a> <br><br><br>
<a href="https://www.aicte-india.org/">Visit AICTE Website</a> <br><br><br>
<a href="test.html"> Home </a>
</body>
</html>
```



Visit DTE Website

Visit AICTE Website

Home

HTML img tag (Image) is used to display image on the web page.

HTML **img** tag is an empty tag that contains attributes only; closing tags are not used in HTML image element.

The <img> tag is used to embed an image in an HTML page.

The **src** attribute specifies the path to the image to be displayed:

```
<img src="flag.jpg">
```

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):

```
<img src="flagl.jpg" width="200" height="200">
```

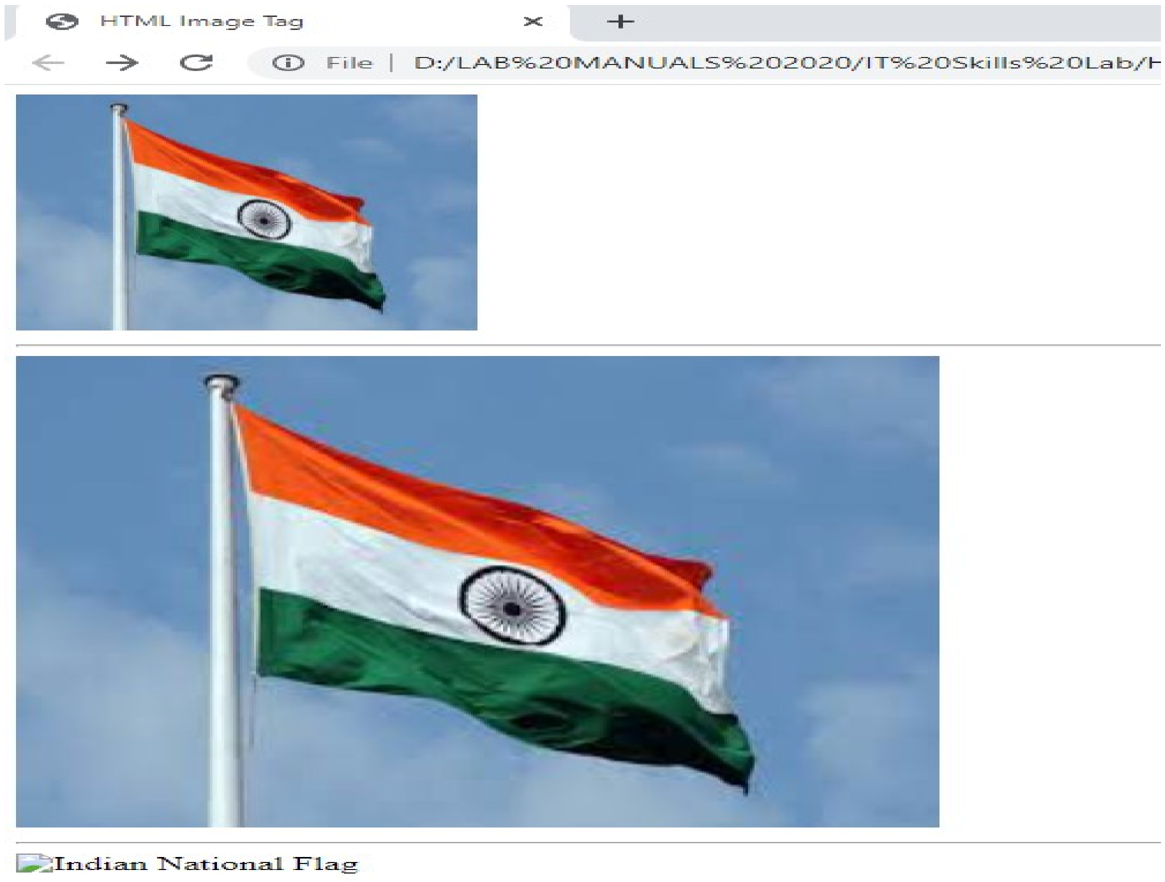
The alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

```
<img src="flag.jpg" alt="Indian National Flag">
```

Example:

```
<html>
<head>
<title>HTML Image Tag </title>
</head>
<body>
<img src="flag.jpg" width="200" height="200">
<hr>
<img src="flag.jpg" width="400" height="400">
```

```
<hr>
<img src="flag2.jpg" alt = "Indian National Flag">
</body>
</html>
```



HTML Lists

HTML lists allow web developers to group a set of related items in lists.

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:

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:

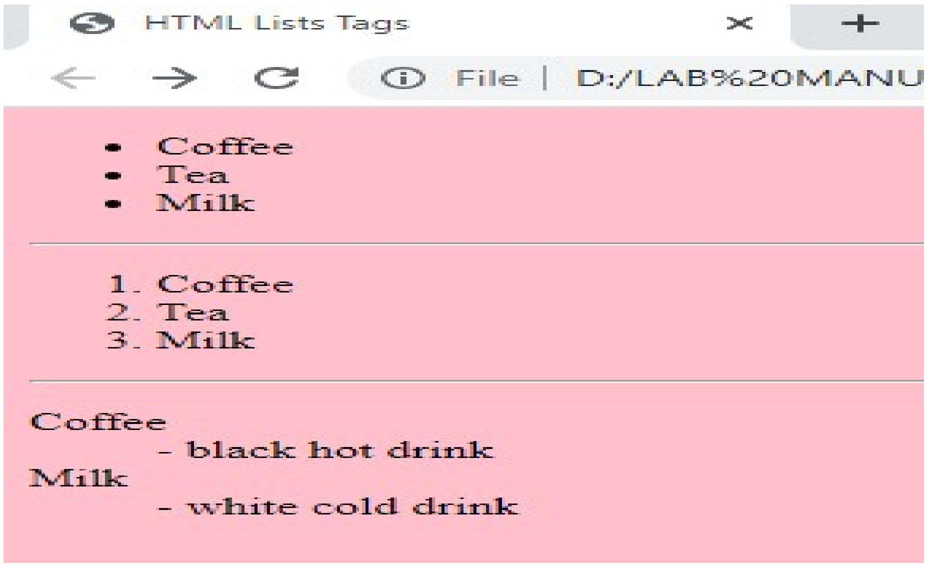
A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

Example:

```
<html>
```

```
<head>
<title>HTML Lists Tags </title>
</head>
<body bgcolor="pink">
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
<hr>
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
<hr>
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
</body>
</html>
```



```
    • Coffee
    • Tea
    • Milk

    1. Coffee
    2. Tea
    3. Milk

Coffee
        - black hot drink
Milk
        - white cold drink
```

HTML Tables

- HTML tables allow web developers to arrange data into rows and columns.
- Each table row starts with a <tr> and end with a </tr> tag.

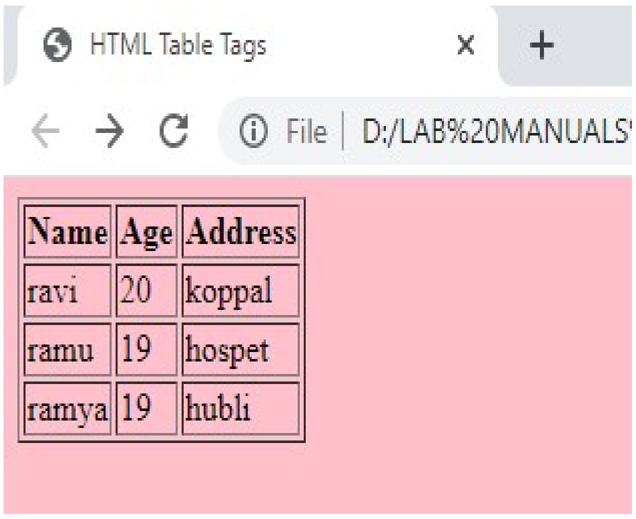
- A table in HTML consists of table cells inside rows and columns.
- Each table cell is defined by a <td> and a </td> tag.
- Everything between <td> and </td> are the content of the table cell.
- Sometimes we want our cells to be headers, in those cases use the <th> tag instead of the <td> tag:

```
<table>
  <tr>
    <td>ravi</td>
    <td>23</td>
    <td> koppal </td>
  </tr>
</table>
```

Example:

```
<html>
<head>
<title>HTML Table Tags </title>
</head>
<body bgcolor="pink">
<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>Address</th>
  </tr>
  <tr>
    <td>ravi</td>
    <td>20</td>
    <td>koppal</td>
  </tr>
  <tr>
    <td>ramu</td>
    <td>19</td>
    <td>hospet</td>
  </tr>
  <tr>
    <td>ramya</td>
    <td>19</td>
    <td>hubli</td>
  </tr>
```

```
</table>
```



Name	Age	Address
ravi	20	koppal
ramu	19	hospet
ramya	19	hubli

HTML Forms:

- HTML Forms are required, when we want to collect some data from the end user.
- For example, during user registration we would like to collect information such as name, email address, credit card, etc.
- There are various form elements like text fields, textarea fields, drop-down menus, buttons, radio buttons, checkboxes, etc.
- The HTML <form> tag is used to create an HTML form and it has following syntax –

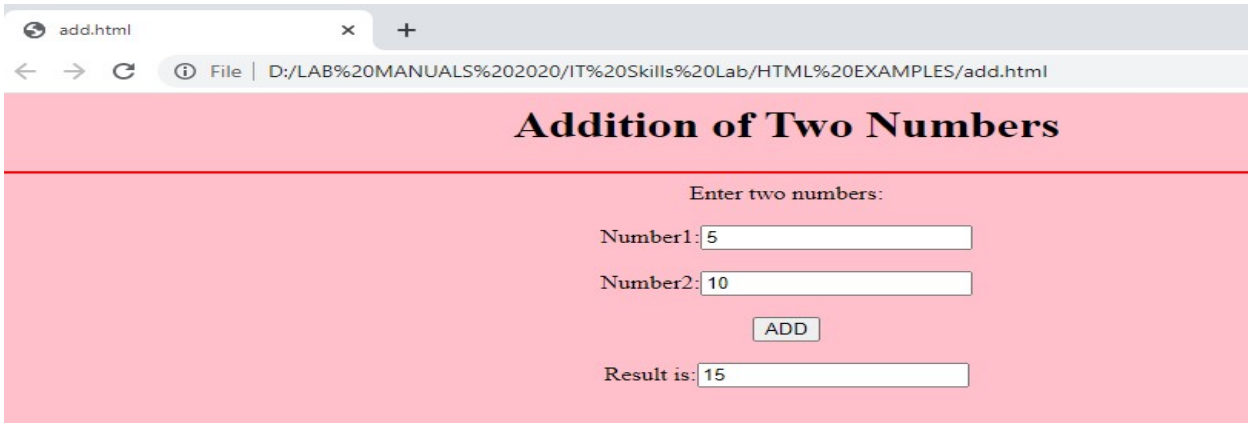
```
<form action = "Script URL" method = "GET|POST">
 form elements like input, textarea etc.
</form>
```

- There are different types of form controls that we 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

Example:

```
<html>
<head>
<script type="text/javascript">
function add()
{
var x,y,z;
x=parseInt(document.myform.n1.value);
y=parseInt(document.myform.n2.value);
z=x+y;
document.myform.result.value=z;
}
</script>
</head>
<body bgcolor="pink">
<h1 align="center"> Addition of Two Numbers</h1>
<hr color="red">
<center>
Enter two numbers:
<br><br>
<form name="myform">
Number1:<input type="text" name="n1" value=""><br><br>
Number2:<input type="text" name="n2" value=""><br><br>
<input type="button" value="ADD" onClick="add()"><br><br>
Result is:<input type="text" name="result" value="">
</form>
</body>
</html>
```


add.html
File | D:/LAB%20MANUALS%202020/IT%20Skills%20Lab/HTML%20EXAMPLES/add.html
ADD

Addition of Two Numbers

Enter two numbers:

Number1: 5

Number2: 10

Result is: 15

HTML Formatting Elements

Formatting elements were designed to display special types of text.

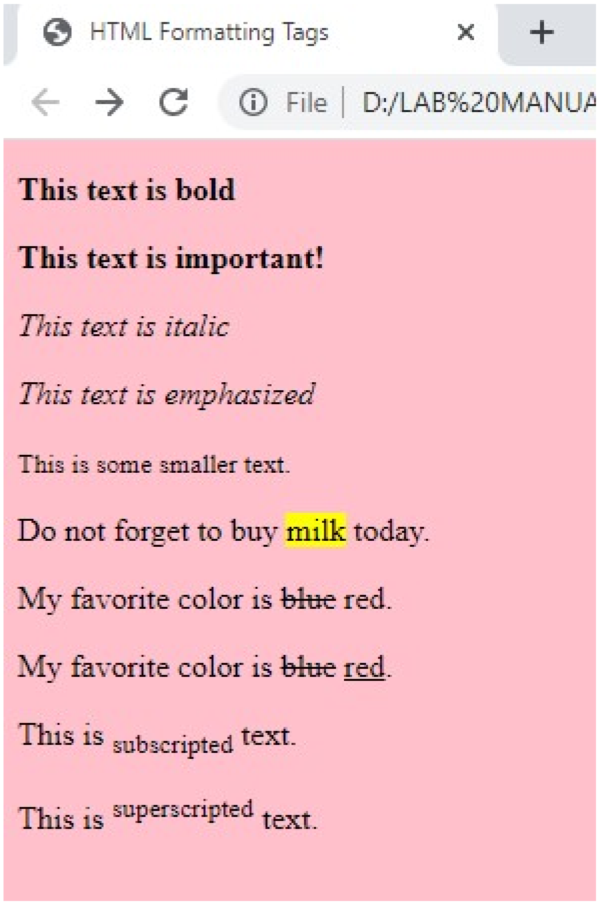
- <b> - The HTML <b> element defines bold text, without any extra importance.

- <strong> - The HTML <strong> element defines text with strong importance. The content inside is typically displayed in bold.
- <i> - The HTML <I> element defines italic text.
- <em> - The HTML <em> element defines emphasized text. The content inside is typically displayed in italic.
- <small> - The HTML <small> element defines smaller text
- <mark> - The HTML <mark> element defines text that should be marked or highlighted:
- <del> - The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text.
- <ins> - The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text
- <sub> - The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O:
- <sup> - The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font.

Example:

```
<html>
<head>
<title>HTML Formatting Tags </title>
</head>
<body bgcolor="pink">
<p> <b>This text is bold</b> </p>
<p> <strong>This text is important!</strong> </p>
<p> <i>This text is italic</i> </p>
<p> <em>This text is emphasized</em> </p>
<p> <small>This is some smaller text.</small> </p>
<p> Do not forget to buy <mark>milk</mark> today.</p>
<p> My favorite color is <del>blue</del> red.</p>
<p> My favorite color is <del>blue</del> <ins>red</ins>.</p>
<p> This is <sub>subscripted</sub> text.</p>
<p> This is <sup>superscripted</sup> text.</p>
```

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



This text is bold

This text is important!

This text is italic

This text is emphasized

This is some smaller text.

Do not forget to buy milk today.

My favorite color is ~~blue~~ red.

My favorite color is ~~blue~~ red.

This is subscripted text.

This is superscripted text.

HTML Iframes

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

```
<iframe src="url" title="description"></iframe>
```

Use the height and width attributes to specify the size of the iframe. The height and width are specified in pixels by default:

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

To remove the border, add the style attribute and use the CSS border property:

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

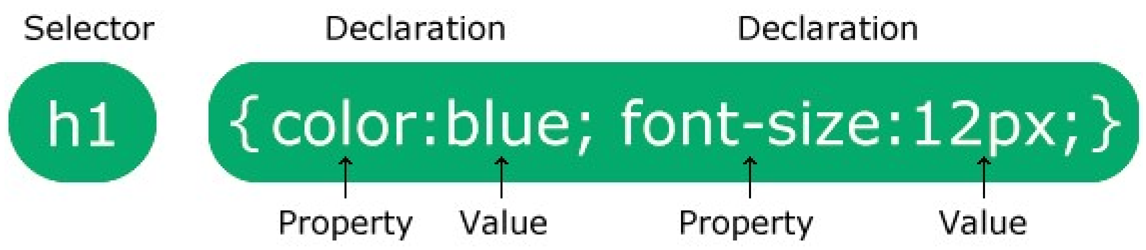
An iframe can be used as the target frame for a link.

The target attribute of the link must refer to the name attribute of the iframe:

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

Formatting web pages with style sheets (CSS).

- CSS stands for **Cascading Style Sheets**
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- A CSS rule consists of a selector and a declaration block.
- CSS Syntax



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.
- Example

```
p {
  color: red;
  text-align: center;
}
```

Example Explained

- p is a selector in CSS (it points to the HTML element you want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

CSS can be added to HTML documents in 3 ways:

i) Inline - by using the style attribute inside HTML elements

ii) Internal - by using a <style> element in the <head> section

iii) External - by using a <link> element to link to an external CSS file

i) Inline CSS

- An inline CSS is used to apply a unique style to a single HTML element.
- An inline CSS uses the style attribute of an HTML element.
- The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:
- **Ex:**

```
<h1 style="color:blue;">This is a blue heading</h1>
<p style="color:red;">This is a red paragraph.</p>
```

ii) Internal CSS

- An internal CSS is used to define a style for a single HTML page.
- An internal CSS is defined in the <head> section of an HTML page, within a <style> element.
- The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

Example:

```
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

CSS Selectors:

- CSS selectors are used to "find" (or select) the HTML elements you want to style.
- Simple selectors (select elements based on name, id, class)

CSS element name Selector:

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {
 text-align: center;
 color: red;
 }
```

Example:

```
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
</body>
</html>
```

CSS id Selector:

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1 {
 text-align: center;
 color: blue;
 }
```

Example:

```
<html>
<head>
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

CSS class Selector:

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

Example

In this example all HTML elements with class="center" will be red and center-aligned:

```
.center {
 text-align: center;
 color: red;
 }
```

Example:

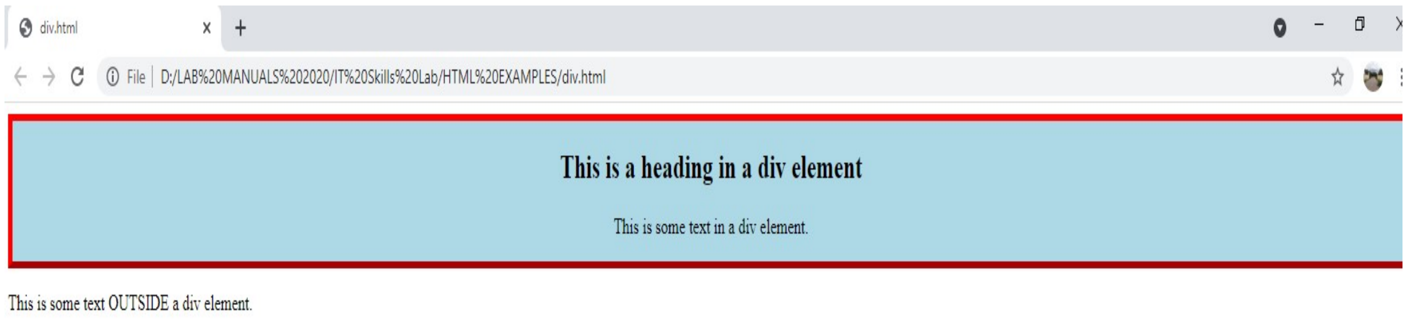
```
<html>
<head>
<style>
.center {
  text-align:center;
  color: red;
}
</style>
</head>
<body>
<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
</body>
</html>
```

The <div> tag:

- The <div> tag defines a division or a section in an HTML document.
- The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- The <div> tag is easily styled by using the class or id attribute.
- Any sort of content can be put inside the <div> tag!
- Example: A <div> section in a document that is styled with CSS:<html>
- Example:

```
<head>
<style>
.myDiv {
 border: 5px outset red;
 background-color: lightblue;
 text-align: center;
 }
```

```
</style>
</head>
<body>
<div class="myDiv">
 <h2>This is a heading in a div element</h2>
 <p>This is some text in a div element.</p>
</div>
<p>This is some text in OUTSIDE div element.</p>
</body>
</html>
```



div.html
File | D:/LAB%20MANUALS%202020/IT%20Skills%20Lab/HTML%20EXAMPLES/div.html

This is a heading in a div element

This is some text in a div element.

This is some text OUTSIDE a div element.

CSS Colors

- In CSS, a color can be specified by using a predefined color name:
  - Orange, blue, green, black, red ,tomato, gray, violet.
- CSS Background Color :We can set the background color for HTML elements:

Example

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Welcome to IT Skills Lab...</p>
```

- CSS Text Color : We can set the color of text:

Example

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:Blue;">Welcome to web desiging</p>
<p style="color:Green;">I am learning CSS</p>
```

- CSS Border Color :We can set the color of borders:

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:3px solid Blue;">Hello World</h1>
<h1 style="border:5px solid Violet;">Hello World</h1>
```

The float Property:

- The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.
- The float property can have one of the following values:
    - ✓ left - The element floats to the left of its container
    - ✓ right - The element floats to the right of its container
    - ✓ none - The element does not float (will be displayed just where it occurs in the text). This is default
    - ✓ inherit - The element inherits the float value of its parent
- In its simplest use, the float property can be used to wrap text around images.

Ex:
```
img {
    float: right;
    }
```

The position Property

- The position property specifies the type of positioning method used for an element.
- There are four different position values:
    - ➢ static
    - ➢ fixed
    - ➢ relative
    - ➢ absolute
- Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first.

i) position: static;

- HTML elements are positioned static by default.
- Static positioned elements are not affected by the top, bottom, left, and right properties.
- An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

Ex:
```
div.static {
    position: static;
    border: 3px solid #73AD21;
    }
```

ii) position: fixed;

- An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.
- The top, right, bottom, and left properties are used to position the element.
- A fixed element does not leave a gap in the page where it would normally have been located.

Ex:
```
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
    }
```

iii) position: relative;

- An element with position: relative; is positioned relative to its normal position.
- Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.
- Other content will not be adjusted to fit into any gap left by the element.

Ex:
```
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
    }
```

iv) position: absolute;

- An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
- However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

Ex:
```
div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
    }
```

CSS Margins

- The CSS margin properties are used to create space around elements, outside of any defined borders.
- With CSS, we have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).
- CSS has properties for specifying the margin for each side of an element:
  - ✓ margin-top
  - ✓ margin-right
  - ✓ margin-bottom
  - ✓ margin-left
- All the margin properties can have the following values:
  - ✓ auto - the browser calculates the margin
  - ✓ length - specifies a margin in px, pt, cm, etc.
  - ✓ % - specifies a margin in % of the width of the containing element
  - ✓ inherit - specifies that the margin should be inherited from the parent element

Ex:
```
p {
margin-top: 100px;
margin-bottom: 100px;
margin-right: 150px;
margin-left: 80px;
}
```

CSS Padding

- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).
- CSS has properties for specifying the padding for each side of an element:
  - ✓ padding-top
  - ✓ padding-right
  - ✓ padding-bottom
  - ✓ padding-left
- All the padding properties can have the following values:
  - ✓ length - specifies a padding in px, pt, cm, etc.
  - ✓ % - specifies a padding in % of the width of the containing element
  - ✓ inherit - specifies that the padding should be inherited from the parent element

Ex:
```
div {
padding-top: 50px;
padding-right: 30px;
padding-bottom: 50px;
padding-left: 80px;
}
```

CSS Borders:

- The CSS border properties allow you to specify the style, width, and color of an element's border.
- The border-style property specifies what kind of border to display.
- The following values are allowed:
  - ✓ dotted - Defines a dotted border
  - ✓ dashed - Defines a dashed border
  - ✓ solid - Defines a solid border
  - ✓ double - Defines a double border
  - ✓ groove - Defines a 3D grooved border. The effect depends on the border-color value
  - ✓ ridge - Defines a 3D ridged border. The effect depends on the border-color value
  - ✓ inset - Defines a 3D inset border. The effect depends on the border-color value
  - ✓ outset - Defines a 3D outset border. The effect depends on the border-color value
  - ✓ none - Defines no border
  - ✓ hidden - Defines a hidden border
- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

Example:

Demonstration of the different border styles:

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
```

CSS Fonts

- CSS Font property is used to control the look of texts. By the use of CSS font property we can change the text size, color, style and more.
- Generic Font Families: In CSS there are five generic font families:
  i. Serif fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
  ii. Sans-serif fonts have clean lines. They create a modern and minimalistic look.
  iii. Monospace fonts - here all the letters have the same fixed width. They create a mechanical look.
  iv. Cursive fonts imitate human handwriting.
  v. Fantasy fonts are decorative/playful fonts.
- These are some important font attributes:
  - ✓ CSS Font color: This property is used to change the color of the text. (standalone attribute)
  - ✓ CSS Font family: This property is used to change the face of the font.
  - ✓ CSS Font size: This property is used to increase or decrease the size of the font.
  - ✓ CSS Font style: This property is used to make the font bold, italic or oblique.
  - ✓ CSS Font variant: This property creates a small-caps effect.
  - ✓ CSS Font weight: This property is used to increase or decrease the boldness and lightness of font.

The CSS font-family Property:

- In CSS, we use the font-family property to specify the font of a text. The font names should be separated with comma.

Ex:

```
.p1 {
 font-family: "Times New Roman", Times, serif;
 }
.p2 {
 font-family: Arial, Helvetica, sans-serif;
 }
.p3 {
 font-family: "Lucida Console", "Courier New", monospace;
 }
```

The CSS font-size Property:

- The font-size property sets the size of the text.
- Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

Ex:

```
h1 {
 font-size: 40px;
 }
h2 {
 font-size: 30px;
 }
p {
 font-size: 14px;
 }
```

The CSS font-style Property:

- The font-style property is mostly used to specify italic text.This property has three values:
  - ✓ normal - The text is shown normally
  - ✓ italic - The text is shown in italics
  - ✓ oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

Ex:
```
p.italic { font-style: italic; }
p.normal { font-style: normal; }
p.oblique { font-style: oblique;}
```

The CSS font-variant Property:

- The font-variant property specifies whether or not a text should be displayed in a small-caps font.

Ex:
```
p.normal {
font-variant: normal;
}
p.small {
font-variant: small-caps;
}
```

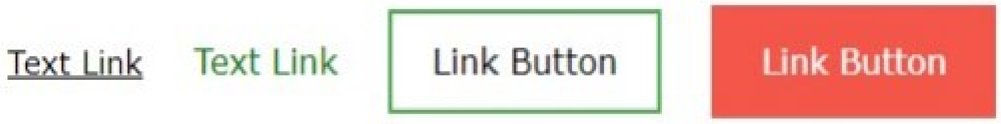
The CSS font-weight Property:

- The font-weight property specifies the weight of a font:

Ex:
```
p.normal {
font-weight: normal;
}
p.thick {
font-weight: bold;
}
```

Styling Links:

- With CSS, links can be styled in different ways.



- The links can be styled differently depending on what state they are in. The four links states are:
  - ✓ a:link - a normal, unvisited link
  - ✓ a:visited - a link the user has visited
  - ✓ a:hover - a link when the user mouses over it
  - ✓ a:active - a link the moment it is clicked.

```
/* unvisited link */
a:link { color: red;}
/* visited link */
a:visited { color: green;}
/* mouse over link */
a:hover { color: hotpink;}
/* selected link */
a:active { color: blue;}
```

Creating a dynamic web page using JavaScript.

Dynamic Web page:

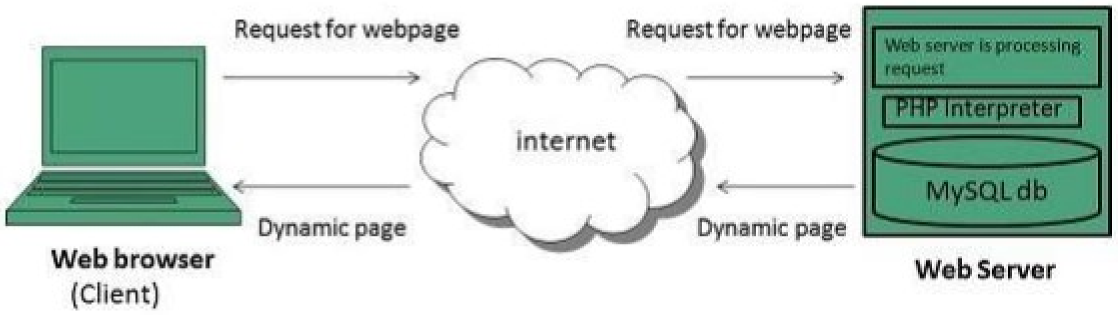
Dynamic web page shows different information at different point of time. It is possible to change a portion of a web page without loading the entire web page. It has been made possible using Ajax technology.

Server-side dynamic web page:

It is created by using server-side scripting. There are server-side scripting parameters that determine how to assemble a new web page which also includes setting up of more client-side processing.

Client-side dynamic web page:

It is processed using client side scripting such as JavaScript and then passed in to Document Object Model (DOM).



Scripting Languages:

- Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line.
- Scripting language is used to create dynamic web pages.

Client-side Scripting:

- It refers to the programs that are executed on client-side. Client-side scripts contain the instruction for the browser to be executed in response to certain user’s action.
- Client-side scripting programs can be embedded into HTML files or also can be kept as separate files.
- Some of examples of client side scripting languages are: JavaScript (JS), ActionScript,

- ✓ Dart and VBScript.
- ✓ Introduction to JavaScript:
- ✓ JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications.
- ✓ It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

- Advantages of JavaScript
  - ✓ Less server interaction
  - ✓ Immediate feedback to the visitors
  - ✓ Increased interactivity
  - ✓ Richer interfaces.

JavaScript Synatx:

- JavaScript can be implemented using JavaScript statements that are placed within the <script>... </script> HTML tags in a web page.
- We can place the <script> tags, containing our JavaScript, anywhere within our web page, but it is normally recommended that we should keep it within the <head> tags.
- The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows.

```
<script ...>
        JavaScript code
</script>
```

  - The script tag takes two important attributes – Language and Type
  - So our JavaScript segment will look like –

```
<script language = "javascript" type = "text/javascript">
            JavaScript code
</script>
```

JavaScript Functions:

- A function is a group of reusable code which can be called anywhere in a program. This eliminates the need of writing the same code again and again.
- It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

Function Definition

- Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.
- The basic syntax is shown here.

```
<script type = "text/javascript">
 function functionname(parameter-list)
 {
 statements
 }
```

```
</script>
```

Ex:

- The following example defines a function called sayHello that takes no parameters.

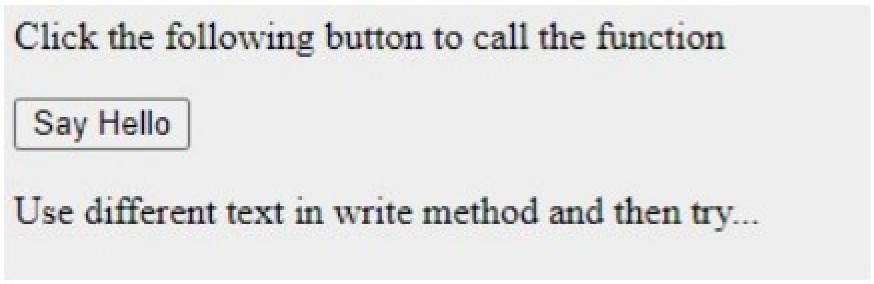
```
<script type = "text/javascript">
 function sayHello()
 {
 alert("Hello there");
 }
 </script>
```

Calling a Function

- To invoke a function somewhere later in the script, we simply need to write the name of that function as shown in the following code.

```
<html>
 <head>
 <script type = "text/javascript">
 function sayHello()
 {
     document.write ("Hello there!");
 }
 </script>
 </head>
 <body>
 <p>Click the following button to call the function</p>
 <form>
 <input type = "button" onclick = "sayHello()"  value = "Say Hello">
 </form>
 <p>Use different text in write method and then try...</p>
 </body>
</html>
```

Output:



Click the following button to call the function

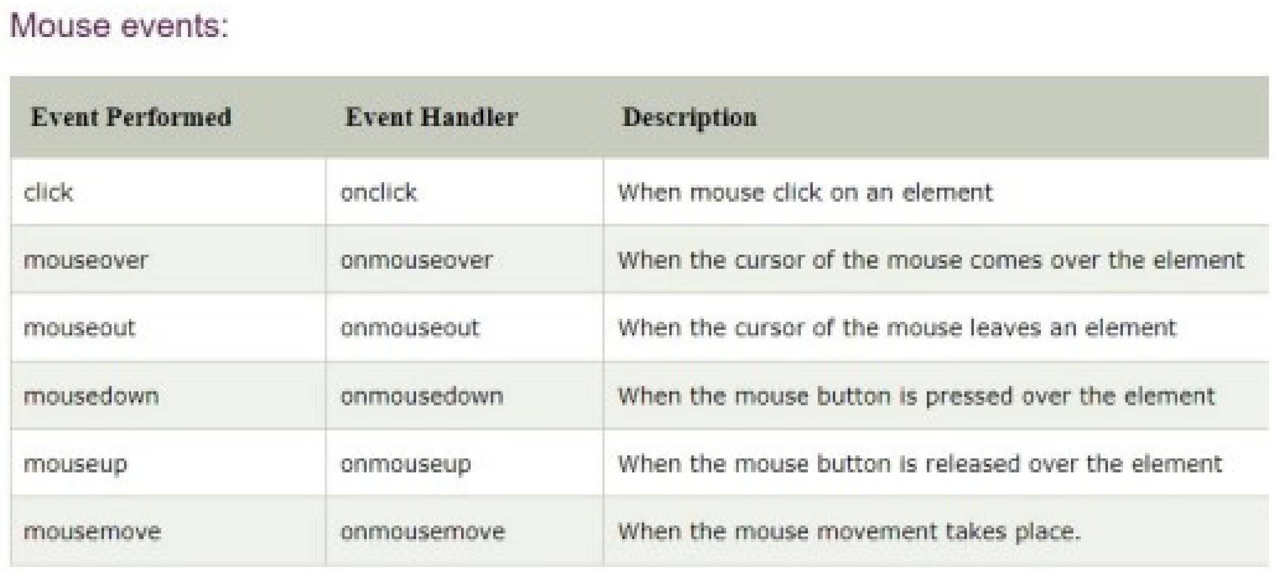
Say Hello

Use different text in write method and then try...

JavaScript – Events:

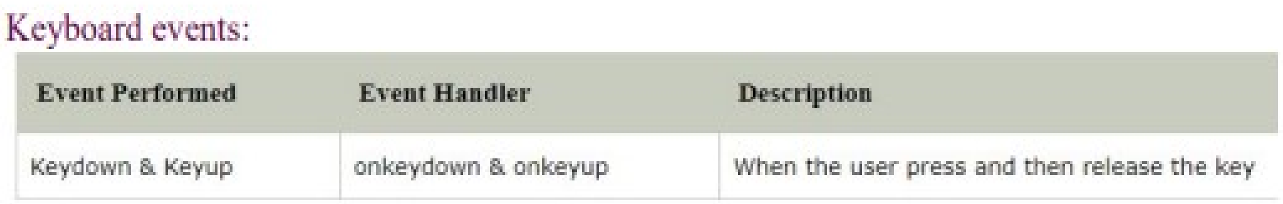
- The change in the state of an object is known as an Event.
- In html, there are various events which represents that some activity is performed by the user or by the browser.

- This process of reacting over the events is called Event Handling. Thus, js handles the HTML events via Event Handlers.
- Some of the HTML events and their event handlers are:



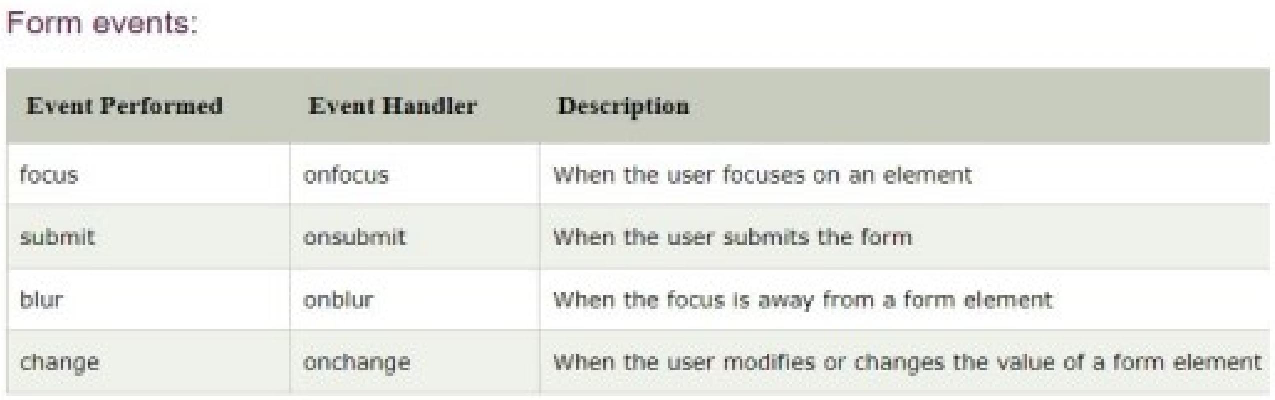
Mouse events:

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.



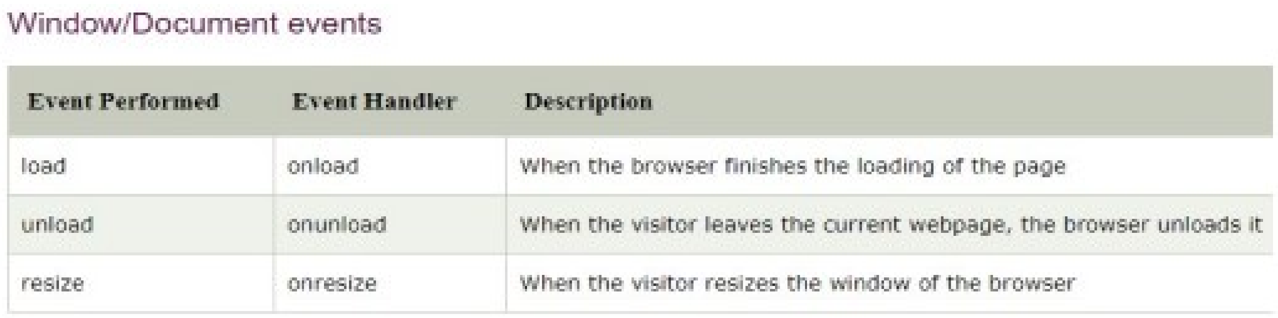
Keyboard events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key



Form events:

Event Performed	Event Handler	Description
focus	onfocus	When the user focuses on an element
submit	onsubmit	When the user submits the form
blur	onblur	When the focus is away from a form element
change	onchange	When the user modifies or changes the value of a form element



Window/Document events

Event Performed	Event Handler	Description
load	onload	When the browser finishes the loading of the page
unload	onunload	When the visitor leaves the current webpage, the browser unloads it
resize	onresize	When the visitor resizes the window of the browser