Unit-I

Introduction - Internet & Website

❖ Internet:

Definition: Internet is defined as an international network of computers which exchange the information.

Or

Definition: The internet is a global system of interconnected computer network that uses internet protocol TCP/IP (Transmission Control Protocol / Internet Protocol) communicate between network and devices.

- The internet is consists of private public academic business and government networks linked by electronic wireless and optical networking technologies.
- Internet uses the standard Internet Protocol (TCP/IP).
- Every computer in internet is identified by a unique IP address.
- IP Address is a unique set of numbers (such as 110.22.33.114) which identifies a computer location.
- A special computer DNS (Domain Name Server) is used to give name to the IP Address so that user can locate a computer by a name.
- For example, a DNS server a name **http://www.bdcp.edu.in** to a particular IP address to uniquely identify the computer on which this website is hosted.
- Internet is accessible to every user all over the world.

❖ History and Evolution of Internet:

The concept of Internet was originated in 1969 and has undergone several technological & Infrastructural changes as.

- The origin of Internet devised from the concept of **Advanced Research Project Agency Network (ARPANET).**
- **ARPANET** was developed by United States Department of Defense.
- ARPANET used packet switching to allow multiple computers to communicate on a single network
- In 1969 are ARPANET delivered first message in node to node communication from one computer to another.

- In 1970 TCP IP transmission control protocol internet protocol where used for data transmission.
- In 1990 the computer scientist Tim Berners Lee invented the World Wide Web to access data online in form of websites and hyperlinks.
- Now the internet became most powerful network of computers to access information.

❖ Internet Protocols:

Definition: Internet Protocols are a set of rules for connecting computers.

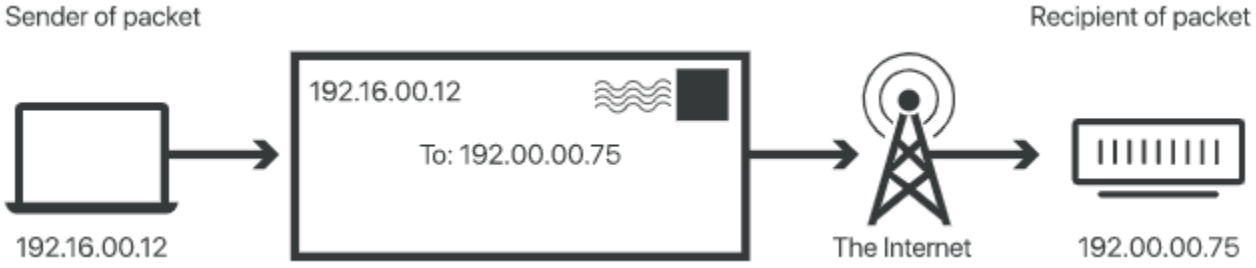
The internet protocol is used for the communication and exchange of data over the internet. Both the sender and receiver should follow the same protocols in order to communicate the data. The internet protocol defined in the TCP/IP (Transmission Control Protocol / Internet Protocol) model used for sending the packets from source to destination. Each large data sent between two network devices is divided into smaller packets.

Types of internet protocol:

The Internet Protocols are of different types having different uses:-

1) TCP/IP (Transmission Control Protocol/ Internet Protocol):

These are a set of standard rules that allows different types of computers to communicate with each other. The IP protocol ensures that each computer that is connected to the Internet is having a specific serial number called the IP address. TCP specifies how data is exchanged over the internet and how it should be broken into IP packets. It also makes sure that the packets have the information about the source of the message data, the destination of the message data and checks if the message has been sent correctly to the specific destination.



The functionality of TCP/IP is divided into 4 layers with each one having specific protocols:

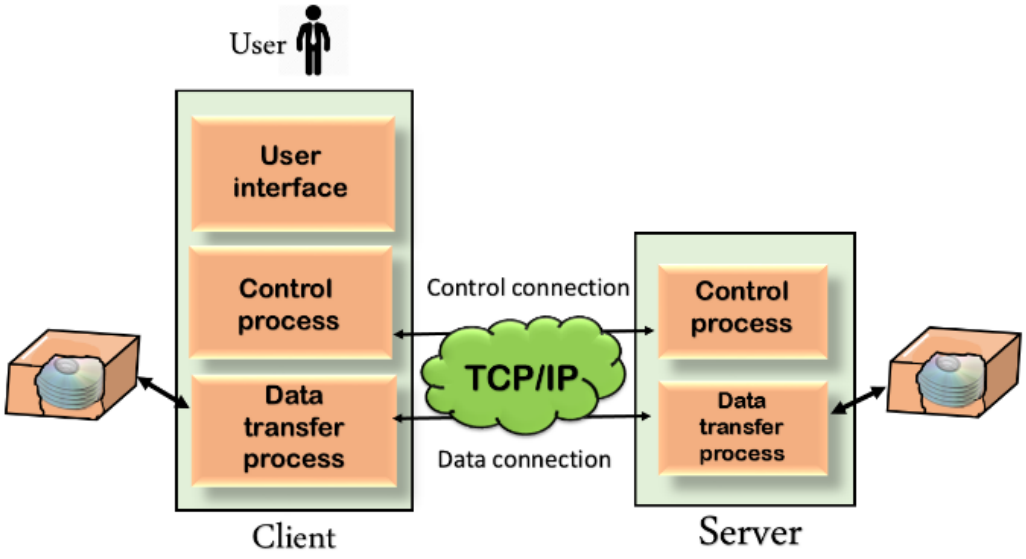
1. **Application Layer:** The application layer makes sure that the data from the sending end is received in a format that is acceptable and supported at the receiving end.

2. **Transport Layer:** The transport layer is responsible for the smooth transmission of data from one end to the other. It is also responsible for reliable connectivity, error recovery, and flow control of the data.
3. **Internet Layer:** This Internet Layer moves packets from source to destination by connecting independent networks.
4. **Network Access Layer:** The Network Access Layer sees how a computer connects to a network.

2) FTP (File Transfer Protocol):

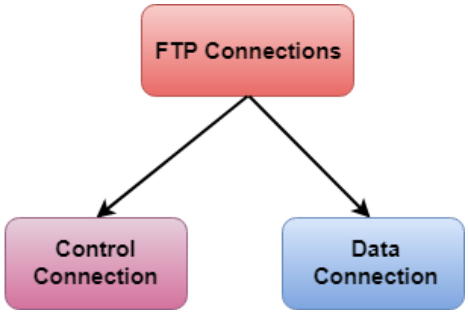
- FTP stands for File transfer protocol.
- This protocol is used for transferring files from one system to the other.
- This works on a client-server model. When a machine requests for file transfer from another machine, the FTO sets up a connection between the two and authenticates each other using their ID and Password.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.

Mechanism of FTP:



The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

There are two types of connections in FTP:



i) **Control Connection:** The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time.

ii) **Data Connection:** The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes.

Advantages of FTP:

- **Speed:** One of the biggest advantages of FTP is speed. The FTP is one of the fastest way to transfer the files from one computer to another computer.
- **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
- **Security:** To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.

Disadvantages of FTP:

- The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.
- FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.
- Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
- It is not compatible with every system.

3) HTTP (Hypertext Transfer Protocol):

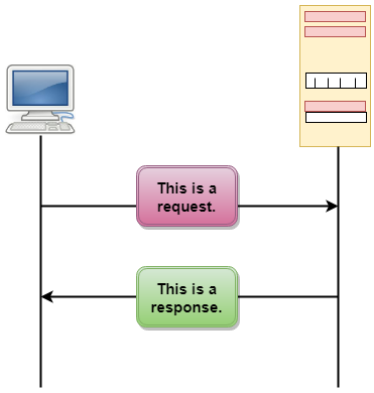
This protocol is used to transfer hypertexts over the internet and it is defined by the www (World Wide Web) for information transfer. This protocol defines how the information needs to be formatted and

transmitted. And, it also defines the various actions the web browsers should take in response to the calls made to access a particular web page. Whenever a user opens their web browser, the user will indirectly use HTTP as this is the protocol that is being used to share text, images, and other multimedia files on the World Wide Web.

Note: ***Hypertext refers to the special format of the text that can contain links to other texts.***

- HTTP stands for Hypertext Transfer Protocol.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as Hypertext Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.
- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

HTTP Transactions



The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

Messages: HTTP messages are of two types: request and response. Both the message types follow the same message format.

Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.

Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body

Difference between HTTP and HTTPS:

HTTP	HTTPS
1. It is an abbreviation of Hypertext Transfer Protocol	1. It is an abbreviation of Hypertext Transfer Protocol Secure.
2. This protocol operates at the application layer.	2. This protocol operates at the transport layer.
3. The data which is transferred in HTTP is plain text.	3. The data which is transferred in HTTPS is encrypted, i.e., ciphertext.
4. The URL (Uniform Resource Locator) of HTTP start with http://	4. The URL (Uniform Resource Locator) of HTTPS start with https://
5. This protocol does not need any certificate.	5. But, this protocol requires an SSL (Secure Socket Layer) certificate.
6. The speed of HTTP is fast as compared to HTTPS.	6. The speed of HTTPS is slow as compared to HTTP.
7. It is un-secure.	7. It is highly secure.
8. Examples of HTTP websites are Educational Sites, Internet Forums, etc.	8. Examples of HTTPS websites are shopping websites, banking websites, etc.

❖ World Wide Web:

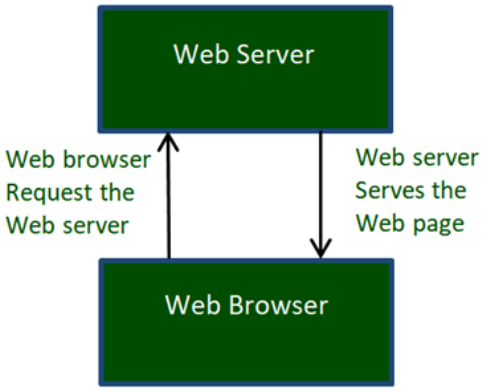
World Wide Web, which is also known as a Web, is a collection of websites or web pages stored in web servers and connected to local computers through the internet. These websites contain text pages, digital images, audios, videos, etc. Users can access the content of these sites from any part of the world over the

internet using their devices such as computers, laptops, cell phones, etc. The WWW, along with internet, enables the retrieval and display of text and media to your device.

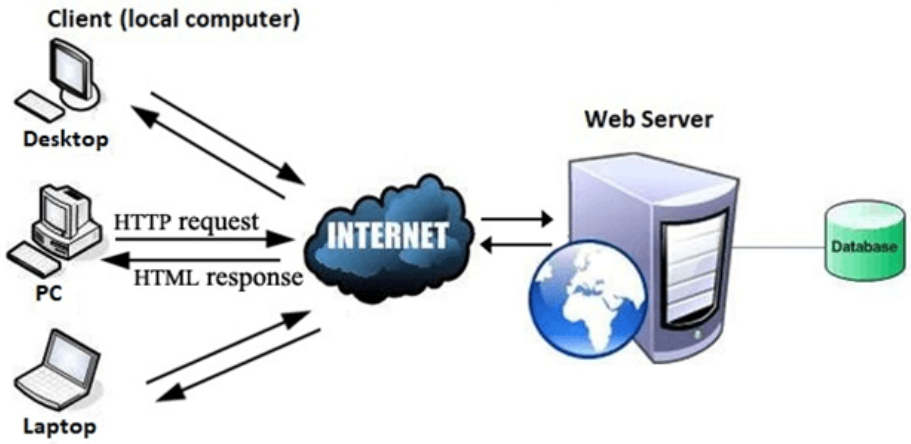
The development of the World Wide Web was begun in 1989 by Tim Berners-Lee and his colleagues in Geneva, Switzerland. They created a protocol, HyperText Transfer Protocol (HTTP), which standardized communication between servers and clients. Their text-based Web browser was made available for general release in January 1992.

The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP. These links are electronic connections that link related pieces of information so that users can access the desired information quickly. Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

A web page is given an online address called a Uniform Resource Locator (URL). A particular collection of web pages that belong to a specific URL is called a website, e.g., www.facebook.com, www.google.com, etc. So, the World Wide Web is like a huge electronic book whose pages are stored on multiple servers across the world.



The Web works as per the internet's basic client-server format. The servers store and transfer web pages or information to user's computers on the network when requested by the users. A web server is a software program which serves the web pages requested by web users using a browser. The computer of a user who requests documents from a server is known as a client. Browser, which is installed on the user' computer, allows users to view the retrieved documents.



Note: All the websites are stored in web servers.

❖ IP address:

An IP address is a unique address that identifies a device on the internet or a local network. IP stands for "Internet Protocol," which is the set of rules for communication over the internet or local network. An IP address is a number which identify a computer system or another device on the Internet to communicate other device, send files, send emails, streaming video, share information's, etc. with other systems via internet. IP address helps identify the different systems uniquely.

An IP address is a string of numbers separated by periods. IP addresses are expressed as a set of four numbers — an example address might be 192.158.1.38. Each number in the set can range from 0 to 255. So, the full IP addressing range goes from 0.0.0.0 to 255.255.255.255. Example: 123.45.67.89

Types of IP Address:

There are different categories of IP addresses, and within each category, different types.

1) **Consumer IP addresses:** Every individual or business with an internet service plan will have two types of IP addresses: their private IP addresses and their public IP address. The terms public and private relate to the network location — that is, a private IP address is used inside a network, while a public one is used outside a network.
2) **Private IP addresses:** Every device that connects to your internet network has a private IP address. This includes computers, smartphones, and tablets but also any Bluetooth-enabled devices like speakers, printers, or smart TVs.
3) **Public IP addresses:** A public IP address is an IP address that can be accessed directly over the internet and is assigned to your network router by your internet service provider (ISP).

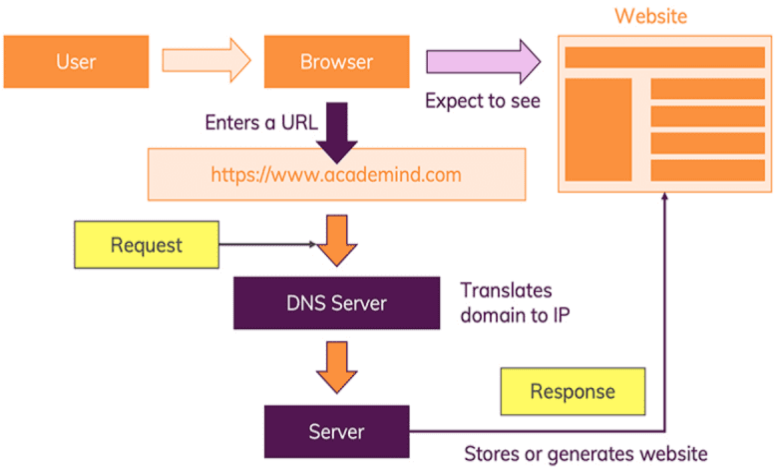
Public IP addresses come in two forms – dynamic and static.

i) **Dynamic IP addresses:** A dynamic IP address is a temporary address for devices connected to a network that continually changes over time. for example. There are security benefits, too, because a changing IP address makes it harder for criminals to hack into your network interface.

ii) **Static IP addresses:** In contrast to dynamic IP addresses, static addresses remain consistent. Once the network assigns an IP address, it remains the same.

❖ Working of website:

- A user enters a URL into a browser (for example, Google.com. This request is passed to a domain name server.
- The domain name server returns an IP address for the server that hosts the Website (for example, 68.178.157.132).
- The browser requests the page from the Web server using the IP address specified by the domain name server.
- The Web server returns the page to the IP address specified by the browser requesting the page. The page may also contain links to other files on the same server, such as images, which the browser will also request.
- The browser collects all the information and displays to your computer in the form of Web page.



❖ Web Browsers and its types:

A web browser (commonly referred to as a browser) is application software for accessing the World Wide Web. When a user requests a web page from a particular website, the web browser displays the necessary content from a web server and then displays the page on the user's device.

Web Browsers are software installed on your PC. To access the Web, you need a web browser, such as Netscape Navigator, Microsoft Internet Explorer or Mozilla Firefox. On the Web, when you search the web pages, this is commonly known as web browsing or web surfing. There are web browsers are available today − Explorer, Firefox, Chrome, Netscape, Opera, and Safari etc.

History of Web Browser:

Today web browsers can be used on devices like computer, laptops, mobile phones, etc.

1. "World Wide Web" was the first web browser created by Tim Berners Lee in 1990.
2. In 1993, the "Mosaic" web browser was released. It had the feature of adding images and an innovative graphical interface. It was the "the world's first popular browser"
3. After this, in 1994, Marc Andreessen (leader of Mosaic Team) started working on a new web browser, which was released and was named "Netscape Navigator"
4. In 1995, "Internet Explorer" was launched by Microsoft.
5. In 2002, "Mozilla Firefox" was introduced which was equally as competent as Internet Explorer
6. Apple too launched a web browser in the year 2003 and named it "Safari". This browser is commonly used in Apple devices only and not popular with other devices
7. Finally, in the year 2008, Google released "Chrome" and is one of the most commonly used web browsers across the world

Browsers types:

Internet Explorer:

Internet Explorer (IE) is a web browser developed by Microsoft and included in all Microsoft operating systems. This is the most commonly used browser in the universe. This was introduced in 1995 along with Windows 95. Internet Explorer will discontinued on 15th June 2022, after which the alternative will be Microsoft Edge.

Google Chrome:

This web browser is developed by Google and its beta version was first released on September 2, 2008 for Microsoft Windows. Google Chrome is the most popular browser that people use today. The main reason for this is its speed. It's a fast browser. It opens quickly and loads multiple tabs and pages in just one click.

Mozilla Firefox:

Firefox is a new browser derived from Mozilla. Firefox was slower than Chrome. It was released in 2004 and has grown to be the second most popular browser on the Internet.

Safari:

Safari is a web browser developed by Apple Inc. and included in Mac OS X. It was first released as a public beta in January 2003. Safari is a simple and clean browser with several features that make it a popular choice. It offers all the basic functionalities – the ability to open multiple tabs, easy bookmarking, fast speeds, and a plugin library.

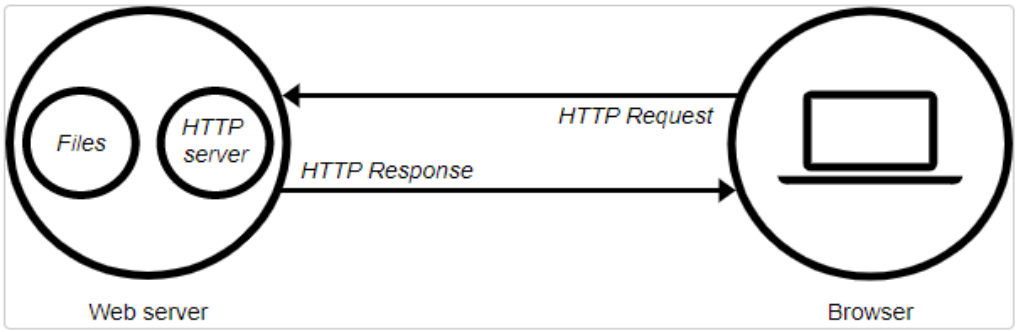
Opera:

Opera is smaller and faster than most other browsers, yet it is full- featured. Fast, user-friendly, with keyboard interface, multiple windows, zoom functions, and more. Java and non-Java-enabled versions available.

❖ Web Server:

Web server is a computer where the web content is stored. A **web server** stores and delivers the content for a website – such as text, images, video, and application data – to clients that request it. A web server communicates with a web browser using the **Hypertext Transfer Protocol** (HTTP). The content of most web pages is created in Hypertext Markup Language (HTML). The content can be static or dynamic. To deliver dynamic content, most web servers support server-side scripting languages. Commonly supported languages include Active Server Pages (ASP), Javascript, PHP, Python, and Ruby.

When a web browser, like Google Chrome or Firefox, needs a file that's hosted on a web server, the browser will request the file by HTTP. When the request is received by the web server, the HTTP server will accept the request, find the content and send it back to the browser through HTTP.



Types of Web Server / Examples of Web Server:

1) **Apache HTTP Server:**

   It is one of the most widely used web servers worldwide. The biggest advantage of using this server is thatit supports almost all operating systems such as Windows, Linux, Apple Mac OS, Unix, and others. Around 60%of the web server machines worldwide, run the Apache Web Server. The Apache HTTP web server was developed by the **Apache Software Foundation. It is an open- source software** which means that it is available for free.

2) **Microsoft Internet Information Services (IIS):**

   IIS (Internet Information Services) is a high performing web server developed by Microsoft. It has all the features of the Apache HTTP Server except that **it is not an open-source software**. It can be easily installed in any Windows device.

3) **Lighttpd:**

   The **lighttpd**, pronounced *lighty* server. It is also a free web server that is distributed with the FreeBSD operating system. This open source web server is fast, secure and consumes much less CPU power. It can also runon Windows, Mac OS X, Linux and Solaris operating systems.

4) **Sun Java System:**

   Sun Java System Web Server is a multi-process, multi-threaded, secure web server built on industry standards. Thisweb server developed by Sun Microsystems is suited for medium and large websites. It is not open source. It runson Windows, Linux and Unix platforms. The Sun Java System web server supports various languages, scripts and technologies required for Web 2.0 such as JSP, Java Servlets, PHP, Perl, Python, Ruby on Rails, ASP and Coldfusion etc.

5) **Jigsaw Server:**

   Jigsaw (W3C's Server) comes from the World Wide Web Consortium. It is open source and free and can run on various platforms like Linux, Unix, Windows, Mac OS X Free BSD etc. Jigsaw is written in Java programminglanguage, which is designed to work in an object-oriented framework.

   Note: The W3C is an international organization committed to improving the web.

❖ What is Website?

A website is a collection of many web pages, and web pages are digital files that are written using HTML (Hypertext Markup Language). The web pages that may contain text, images, audio and video. The first page of a website is called home page. Websites also have become a medium of entertainment like playing online games,watching movies, listening to music, and so on.

Types of Website:

There are basically two main types of website - static and dynamic.

1) **Static website-** A static website is the most basic type of website and contains web pages with fixed content. Static websites are built using simple languages such as HTML, CSS, or JavaScript.

   Examples: Personal websites, Informational websites etc.

   Advantages of static website:

   - Static websites can be developed quickly and much easily in comparison to dynamic websites.
   - Static websites are not costly & it just display information about their business.
   - Hosting of static website is cheaper in comparison to dynamic website.
   - Static websites are processed much faster in comparison to dynamic websites.

   Disadvantages of static website

   - Static website cannot be updated easily. To make any changes a web developer is needed to change thehtml files.
   - Static websites offer limited functionalities like displaying links, images, videos, etc.
   - Static websites are not interactive in nature.

2) **Dynamic website -** In Dynamic Websites, Web pages are returned by the server which are processed during runtime means dynamic websites are designed by using server-side scripting languages such as ASP,PHP, Node.js, ASP.NET, Java etc. So, they are slower than static websites.

   Examples: Facebook website, E-Commerce Websites, Online shopping, booking websites etc.

   Advantages of dynamic website

   1. Dynamic websites are interactive in nature.
   2. Provides better user experience
   3. Dynamic website are easy to update, delete, retrieve information

   Disadvantages of dynamic website

   1. Can be slower to load the dynamic webpage as server will process the input from the user and render theresult page.
   2. Requires technically skilled programmers thus costly.
   3. Hosting dynamic website is more costly as compared to static websites as you also need to purchasedatabase and server.

❖ Difference between Static and Dynamic Websites:

Static Website	Dynamic Website
Content of Web pages cannot be change at runtime.	Content of Web pages can be changed.
No interaction with database possible.	Interaction with database is possible
It is faster to load as compared to dynamic website.	It is slower than static website.
Cheaper Development costs.	More Development costs.
No feature of Content Management.	Feature of Content Management System.
HTML, CSS, JavaScript is used for developing the website.	Server side languages such as PHP, Node.js are used.
Same content is delivered every time the page is loaded.	Content may change every time the page is loaded.

❖ Basics of Web Hosting:

Web hosting is a service of providing online space for storage of web pages. These web pages are made availablevia World Wide Web. The companies which offer website hosting are known as **Web hosts.**

The servers on which web site is hosted remain switched on 24 x7. These servers are run by web hosting companies. Each server has its own IP address. Since IP addresses are difficult to remember therefore, webmasterpoints their domain name to the IP address of the server their website is stored on.

It is not possible to host your website on your local computer, to do so you would have to leave your computer on24 hours a day. This is not practical and cheaper as well. This is where web hosting companies comes in.

You need following two things to host a website.

1. Web hosting service provider
2. Domain name

1) **Web hosting service providers:** This offer you ready-to-use web servers to host your website. They provideeasy to use tools to manage their hosting.
2) **Domain name:** A domain name is the address of your website that people type in the browser to visit yourwebsite. For example, bdc.com will bring you to this website.

Types of Hosting:

1) **Shared Hosting:** In shared hosting, the hosting company stores thousands of website on the same physical server. Each customer has their own allocation of physical web space. As all websites share same physical memory,MYSQL server and Apache server, one website on the server experiencing high traffic load will affect performanceof all websites on the server.
2) **Virtual Private Server (VPS):** It is also known as Virtual Dedicated Server. It is a server which is partitionedinto smaller servers. In this customer is given their own partition, which is installed with its own operating system.
3) **Dedicated Server:** In this kind of hosting, single dedicated server is setup for just one customer. It is commonlyused by the businesses that need the power, control and security that a dedicated server offers.
4) **Reseller Hosting:** A reseller acts as a middle man and sells hosting space of someone else's server.

Web Hosting Companies

Blue Host	Go Daddy	Host Gator	just Host	Laughing Squid	Hivelocity
liquid Web	Media TempleServInt		Wired Tree	Wild West Domains	Wix
WIPL	Big Rock				

❖ Web Development Life Cycle

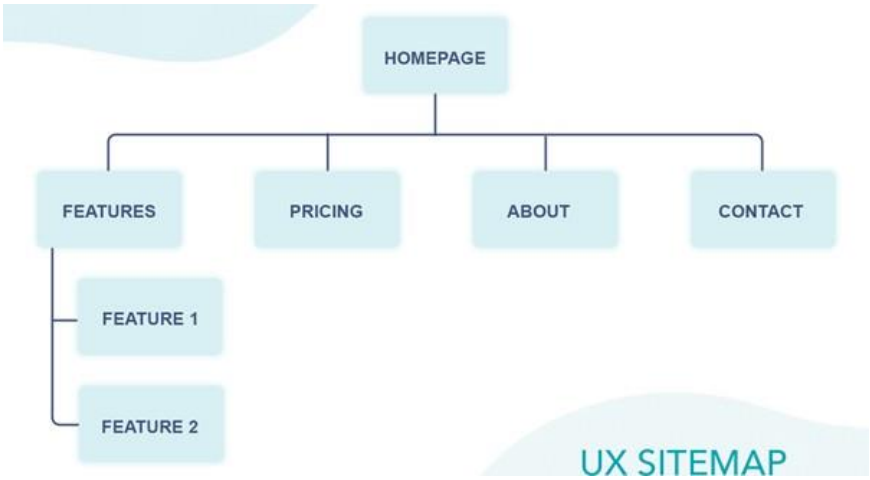
Web development life cycle consist of different stages go into building the website. There are various stagesin Web development life cycle.

1) **Information Gathering:**

The first stage is the most important as it involves understanding the client's requirements. This step involves identifying the purpose, goals, and target audience of your website. You'll determine the requirementsof the audience using this website and the scope of your project.

2) **Planning & Analysis:**

Using the information gathered in the first step, you'll form a sitemap of the website. Based on the information that was collected in the previous phase, the **sitemap** is created. Here, we decide the menus, contents & navigational system for the website.



3) Designing (Design & Layout):

During the design phase, your website takes shape. All the visual content, such as images, photos, and videos is created at this step. The website layout is the result of a designer's work. It can be a graphic sketch or anactual graphic design.

4) Development:

The coding phase is the fourth phase of the web design and development cycle. In this stage, the actual development of the website takes place. You can finally start creating the website itself. Usually, the home page is created first, and then all sub-pages are designed, according to the website sitemap that was previously created in planning and analysis stage.

There are two segments in software development:

1) Frontend development
2) Backend development

1) **Frontend Development:** The website are built by the front end developers using Java script, HTML, CSS, Bootstrap frameworks. Developers also use frameworks or libraries like React, Angular etc. for better frontend development.

2) **Backend Development:** Backend development improves the working of the website. It will be used inupdating the website from time to time. List of best languages for web development in the back end such as PHP, Python, SQL, Ruby, Golang etc.

3) **Testing, Review, and Launch:**

Testing is probably the most routine part of a process. In testing phase every single link should be tested tomake sure that there are no broken ones among them. You should check every form, every script, spell-checking in your website. After you check and re-check your website, it's time to upload it to a server. An FTP(File Transfer Protocol) software is used for that purpose.

4) **Maintenance: Opinion Monitoring and Regular Updating:**

It is important to remember is that a website is more of a service than a product. The feedback system added to the site will allow you to detect possible problems the end-users face. It will also check for cross browser testing. The other important thing is keeping your website up to date.

Unit II - Introduction to HTML

❖ Introduction to HTML:

HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0.

History of HTML:

HTML which is the most widely used language on web to develop web pages.

- **1991- Tim Berners-Lee** invents HTML 1.0
- **1993- HTML 1.0** is released. Not many developers are creating websites at this time.
- **1995- HTML 2.0** is published. This contains the features of HTML 1.0 plus new features. This remained the standard markup language for designing and creating websites until 1997.
- **1997- HTML 3.0** was invented. Here, **Dave Raggett** introduced a fresh draft on HTML, which improved new features of HTML and gave more powerful characteristics for webmasters in designing websites. Unfortunately, the powerful features slowed down the browser in applying further improvements.
- **1999-** The widely-used **HTML 4.0** comes out. It is very successful.
- **2014**- HTML 5.0 is released and used worldwide. It is said to be the extended version of HTML 4.01 which was published in 2012.

❖ Features / advantages of HTML:

1. HTML is easy to use, learn and implement.
2. No special software is required and there is no need to buy HTML software.
3. HTML contains powerful text formatting tags.
4. In HTML finding an error is easy.
5. Because of hyperlinking facility visitors or users can travels to any HTML document.
6. HTML is case insensitive language.
7. It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
8. It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.

❖ Disadvantages/ limitations of HTML:

1. HTML is not programming language in true sense.
2. Any type of calculation cannot be done in HTML and it cannot be used to display any given data.
3. No separate special debugging is provided.
4. Syntax errors are not identified or displayed by HTML.
5. In HTML, scripting language like VB script or JavaScript are required to handle calculation, form validation and events in HTML document.
6. It cannot produce dynamic output alone since it is a static language.
7. HTML does not pass information two pages.
8. All the browsers may not support all the tags of HTML.
9. The security feature is not good.

❖ Structure of HTML:

HTML document can be created using any text editor like notepad and HTML editor. Let us see the structure of HTML document.

```
<!DOCTYPE html>
<html>
<head>
      <title>Title of document </title>
</head>
<body>
       Body of the document
</body>
</html>
```

HTML document is divided into two sections.

1) **Head section**-the head section contains the information about the document that is not displayed on screen.
2) **Body section**-All information in body section is displayed in the browser.

<!DOCTYPE>: It tells the browser about the version of HTML.

<html> tag: This is the most important tag in web page. Browser can read the html page due to this tag. HTML document start with <html> and ends with </ html> tag. But in the new version of internet explorer it is not necessary to write this tag. Its default mode is HTML.

<head> tag: It is used to store the information about HTML document. This information is not going to be displayed in browser. Head section start with <head> tag and ends with </head> tag.

<title> tag: It is used to display a title for a document in title bar of windows. Title section start with <title> tag and ends with </title> tag.

<body> tag: It is the main part of HTML page. Information which is displayed on browser appear in body section of document. Start with <body> tag and ends with </body>tag.

Attributes of <body> tag:

S. N.	Attribute	Description
1	bgcolor	Changes the default background colour of web page. The user can specify a color name or its hexadecimal number. E.g.<body bgcolor=”red”>
2	Background	Specifies the name of image file. It displays background image. E.g. <body background=”rose.jpg”>
3	Text	It changes the text color from its default value to the specified color. E.g.<body text=”red”>

Example:

```
<html>
    <head>
        <title>My First Program</title>
    </head>
    <body>
        Balasaheb Desai College, Patan
    </body>
</html>
```

❖ HTML Tags & Attributes:

HTML tags are like keywords which defines that how web browser will format and display the content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

- All HTML tags must enclosed within < > these brackets
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

HTML elements that have opening tag and a closing tag are called as **non-empty** tags such as . HTML elements that don't have closing tags such as
, <hr>, and <img> are considered **empty tags.**

Syntax: <tag> content </tag>

HTML tags are divided into two levels Block Level Tags & Text Level Tags

1) Block Level Tags:

Block level tags are applied to block of text or paragraphs. Following are block level tags.

i) **Heading tags:** HTML defines six levels of heading tags that is <h1>, <h2>, <h3>, <h4>, <h5> and <h6>. Where <h1> tag is higher level and <h6> is smallest level of heading. Each heading displays bold text in browser.

Attribute Heading Tag:

Align-by using align attribute of heading tags we can place heading to left, centre or right from the document.

Example 1:

```
<html>
<head>
      <title>Heading Tags</title>
</head>
<body>
        <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>
```

Example 2:

```
<html>
<head>
      <title>Heading Tags</title>
</head>
<body>
        <h1 align=”Center”>Heading 1</h1>
        <h2 align=”Left”>Heading 2</h2>
        <h3 align=”Right”>Heading 3</h3>
        <h4 align=”Center”>Heading 4</h4>
        <h5 align=”Left”>Heading 5</h5>
        <h6 align=”Right”>Heading 6</h6>
</body>
</html>
```

ii) <p> tag or Paragraph tag: The p tag is used to indicate paragraph in document. Browser automatically add some space before and after each p tag. P tag places a blank line above and below the text of paragraph.

Attributes Paragraph tag:

Align-by using align attribute of <p> tags we can place paragraph to left, centre or right from the document.

Example:

```
<html>
<head>
      <title>Paragraph Tags</title>
</head>
<body>
     <p>This is first paragraph</p>
     <p>This is second paragraph</p>
     <p>This is third paragraph</p>
     </body>
        </html>
```

iii) <Comment> tag or < !-- and --!>: The everything between <comment> tag and </comment> tag is ignored by browser (internet explorer). For other browsers <!-- and --!> sequences are used for comment.

Example:

```
<html>
<head>
        <title>Paragraph Tags</title>
</head>
<body>
                BCA DEPARTMENT
```

```
        <!-- We are BCA students --!>
</body>
</html>
```

iv) <pre> tag: <pre> tag is used for preformatted text. It displays text as it is typed in the editors (notepad).

Example:

```
<html>
<head>
    <title>Pre Tag</title>
</head>
<body>
<pre> This is sample
    T
        E
            X
                T
</pre>
</body>
</html>
```

**v)
 tag:**
 tag is used to insert the line break in html document.
 tag has no ending tag.

Example:

```
<html>
<head>
    <title>Break Tag</title>
</head>
<body>
    Koyana Education Society's
    <br>
    Balasaheb Desai College, Patan
</body>
</html>
```

vi) <hr> tag: <hr> tag is used to draw horizontal line in HTML document.

Attributes of <hr> tag:

S. N.	Attribute	Description
1	Align	It aligns the line to left center or right of the browser screen by default the alignment of line is center.
2	width	It indicates the width or length of the line.
3	Size	It indicates the thickness of the lines in terms of pixels.
4	Color	It sets or specifies the color of the line.

Example:

```
<html>
<head>
    <title>Horizontal Line</title>
```

```
</head>
<body>
        Department of Computer <hr>
        <hr align=”left” color=”green” size=”30”>
        Balasaheb Desai College, Patan
</body>
</html>
```

vii) <font> tag: <font>tag is used to change the font style, font size and font color of the text within HTML document.

Attributes of <font> tag:

S. N.	Attribute	Description
1	face	It is used to change the font style of text.
2	size	It is used to specifies the size of text,
3	color	It is used to change the color of text.

Example:

```
<html>
<head>
      <title>Horizontal Line</title>
</head>
<body>
        <font size=20 color=”green” face=”Arial”>Department of Computer </font> <br>
        <font color=”red”>BDC Patan <font>
</body>
</html>
```

viii) tag: It is used to align text to center. It start with and ends with .

Example:

```
<html>
<head>
                    <title>Center Tag</title>
 </head>
<body>
    <center>
                    Koyana Education Society’s
                    <br>
                    Balasaheb Desai College, Patan
    </center>
</body>
</html>
```

2) Text Formatting Tag:

Text formatting tags consists of several tags which are used to make text as bold, italic, subscript, superscript and more. Text formatting tags are two types.

i) Physical style tag:

Physical Style tag specifies how the text should be displayed in the browser.

S. N.	Tag	Description
1		Apply bold style to the text between and tag.
2		Apply italic style to the text between and tag.
3	<u>	Apply underline style to the text between <u> and </u> tag.
4	<strike>	Apply strikethrough effect to the text between <strike> and </strike> tag.
5	<sup>	Apply superscript style to the text between ^{and} tag. E.g. 5th o/p- 5^{th}
6	<sub>	Apply superscript style to the text between _{and} tag. E.g. H₂O o/p- H_2O
7	<big>	This tag is used to increase the font size by 1.
8	<small>	This tag is used to decrease the font size by 1.
9	<del>	This tag is used to display deleted text.
10	<ins>	This tag is used to display added text or content.
11	<tt>	Apply monospaced (Type writer) font to the text in between <tt>&</tt> tag.

Example-

```
<html>
<head>
        <title>Physical Style tags</title>
</head>
<body>
        <b> This is bold text </b><br>
        <i> This is italic text </i> <br>
        <u> This is underlined text <u> <br>
        <strike> This is strikethrough text </strike> <br>
        5 <sup> th </sup> standard. <br>
        H <sub>2</sub> O <br>
        <big> bigger font size than current font size </big> <br>
        <small> smaller font size than current font size </small> <br>
        <tt>Balasaheb Desai College, Patan</tt> <br>
        I want <del>10</del> 20 Mango<br>
        I want <del>10</del> <ins>20</ins> Mango <br>
 </body>
</html>
```

ii) Logical Style Tag:

Logical Tags are used in HTML to display the text according to the logical styles.

S. N.	Tag	Description
1		It displays the text in strong style i.e. similar to bold text.
2	<code>	It displays the coded text usually displayed in a courier font.
3	<var>	It indicates variable. often displayed in italics or underlined
4		It is used to display content in italic. (emphasized)
5	<mark>	If you want to mark or highlight a text, you should write content between <mark> & </mark>

Example

```
<html>
<head>
        <title>Physical Style tags</title>
</head>
<body>
        <strong>This is strong text</strong> <br>
        <code>This is code text</code> <br>
        <var>BCA-II</var> <br>
        <em> This is emphasized text</em> <br>
        I want <mark>Mango</mark> <br>
</body>
</html>
```

iii) <marquee>Tag:

<marquee> tag is used to scroll image or text horizontally or vertically. In simple word we can say that it scrolls the image or text up, down, left, or right automatically. The text between <marquee> and </marquee> tag scrolls from right to the left or left to right. By default scrolling direction is left.

Attribute of <marquee> tag:

S. N.	Attribute	Description
1	align	It aligns the text to top, middle, or bottom.
2	bgcolor	It specifies the background colour of marquee.
3	direction	It specifies the direction of scrolling text. It may be left, right, up and down.
4	scrollamount	It specifies or sets the speed of scrolling text.
5	loop	It specifies the number of times marquee text or scrolling text display. By default value is infinite.
6	behavior	It indicates how the text will move. The possible values are scroll, alternate and slide. The default value is scroll.

7	height	It specifies the height of the marquee in pixels or %.
8	width	It specifies the width of marquee in pixels or %.
9	vspace	It sets vertical space around marquee in pixels.
10	hspace	It sets vertical space around marquee in pixels.

Example:

```
<html>
<head>
    <title>Marquee Tag Demo</title>
</head>
<body>
        <marquee width="100%" behavior="alternate" bgcolor="pink" >
        This is an example of a marquee text
        </marquee>
</body>
</html>
```

❖ **List Tag in HTML:** HTML lists allows us to group a set of related items in lists.

There are 3 types of list in HTML.

1. Ordered list.
2. Unordered list.
3. Definition list.

1) Ordered list: Ordered list is a list of items which have specific sequence. tag is used to create ordered list. It starts with <ol> tag and ends with tag. Each list items in order list is start with <li> tag and ends with tag.

Attributes

S. N.	Attribute	Description
1	Type	It specifies the numbering style like 1, A, a, I, i. The default numbering style is 1.
2	start	It specifies the starting value of list item.
3	Value	It changes numbering sequence in middle of ordered list.

Example-1	Example-2	Example-3
<html> <body> - HTML - Java - SQL </body> </html>	<html> <body> <ol Type=A> - HTML - Java - SQL </body> </html>	<html> <body> <ol Type=I start=7> - HTML - Java - SQL </body> </html>
Output- 1. HTML 2. Java 3. SQL	**Output-** A. HTML B. Java C. SQL	**Output-** VII. HTML VIII. Java IX. SQL

2) Unordered list: Unordered list is a list of items which have no specific sequence or order. Unordered list starts with <ul> tag and ends with tag. Each list items in unordered list is start with <li> tag and ends with tag.

Attribute:

i) Type: It indicates the type of list item. The possible values are disk, circle and square etc. the default type is disc.

Example-1

```
<html>
<body>
<ul>
    <li>HTML</li>
    <li>Java</li>
    <li>SQL</li>
</ul>
</body>
</html>
```

Output-

- HTML
- Java
- SQL

Example-2

```
<html>
<body>
<ul Type=”Square”>
    <li>HTML</li>
    <li>Java</li>
    <li>SQL</li>
</ul>
</body>
</html>
```

Output-

- HTML
- Java
- SQL

3) Definition List/ Description List: It is a list of items with description of each item. Definition list start with <dl> tag and ends with </dl> tag. Each list items start with <dt> tag and ends with </dt> tag. Description of each list item or data term start with <dd> tag and ends with </dd> tag. Here <dt> tag defines data term. <dd> tag defines data definition (description).

Example:

```
<html>
<body>
    <dl>
        <dt>Coffee</dt>
```

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

Output-

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

❖ Inserting Image to HTML:

<img> tag: <img> tag is used to display image on the web page or insert image in to the HTML document. <img> tag does not have a closing tag.

Attribute of <img> tag

S. N.	Attribute	Description
1	src	It specifies the path or location or url of image.
2	alt	It specifies the alternate text for the image.
3	height	It specifies the height of image in pixels.
4	width	It specifies the width of image in pixels.
5	align	It align the image to left, center or right of the browser screen.
6	border	It specifies the border to image.

Example-

```
<html>
<head>
    <title>Image Tag</title>
</head>
<body>
  <img src="animal.jpg" height="180" width="300" alt="animal image">
</body>
</html>
```

❖ Hyperlink in HTML: <a> tag:

The <a> tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

Attributes of <a> tag:

S. N.	Attribute	Description
1	href	It specifies the URL or location of file which is hyperlink.
2	target	It specifies where to open the linked file.

		Possible values are- _self- Default. Opens the document in the same window. _blank - Opens the document in a new window or tab _parent - Opens the document in the parent frame _top - Opens the document in the full body of the window
3	name	It is used to link the information on the same page.

Example

i) Hyperlink.html

```
<html>
<head>
    <title>Image Tag</title>
</head>
<body>
        If you want college list in Satara district then click on below link <br>
        <a href=”college.html”> College List</a>
</body>
</html>
```

ii) Image as hyperlink- If you want image as hyperlink then you can use <img> tag inside <a> tag.

```
<html>
<head>
        <title>Image as Hyperlink</title>
</head>
<body>
        <a href=”college.html”> <img src=”1.jpg” height=200 width=200> </a>
</body>
</html>
```

❖ **Table in HTML:** A table is two dimensional matrix, consisting of rows and columns. Table is used to display data in tabular format. <table> tag is used to insert table in html and it consists of one <table> element and one or more <tr>, <th>, and <td> elements.

i) <table>: It defines a table.

ii) <tr>:It defines a row in a table.

iii) <th>:It defines a header cell in a table.

iv) <td>:It defines a cell in a table.

v) <caption>:It defines the table caption.

i) <tr> tag : It creates or defines row in a table.

Attribute of <tr> tag:

Attribute	Description
align	It specifies the alignment of cell data to left, right or center.
bgcolor	It sets the background color of the row.
valign	It controls the vertical alignment of data to top, middle or bottom.

ii) <td> tag: <td> tag is used to define a cell in table. Td stands for table data.

Attribute of <td> tag:

Attribute	Description
align	It aligns the data in cell to left, right or center.
colspan	This attribute combines or merge number of columns in to single cell.
rowspan	This attribute combines or merge number of rows in to single cell.
height	It sets the height of the cell in pixels.
width	It sets the width of the cell in pixels.

iii) <th> tag: <th> tag is used to define a cell in table. The text between <th> and </th> are bold and centered by default.

iv) <caption> tag: The <caption> tag is used to provide caption for a table. This caption can be either appears above or below the table. <caption> tag is always written after <table> tag and before the first <tr>tag.

Align attribute of <caption> tag is used to set caption top or bottom of the table.

Student Grades	
Student	**Grade**
Ajay	B+
Vijay	A+

Example:

```
<html>
<head>
        <title>Table Tag</title>
</head>
<body>
<table border="1">
        <caption align="top"> Student Grades </caption>
        <tr>
                <th> Student</th>
                <th> Grade </th>
        </tr>
        <tr>
                <td> Ajay </td>
                <td> B+ </td>
```

```
        </tr>
        <tr>
                <td> Vijay </td>
                <td> A+ </td>
        </tr>
</table>
</body>
</html>
```

Attributes of <table> tag:

S. N.	Attribute	Description
1	align	It specifies the alignment of table to left, right or center.
2	border	It specifies the table have border or not. Its value from 0 to 10
3	cellspacing	It controls the space between adjacent cells.
4	cellpadding	It controls the distance between data in cell and boundaries of the cell.
5	bgcolor	It sets the background color of the table.
6	bordercolor	It sets the border color of the table.
7	width	It sets the width of table in number of pixels.
8	valign	It controls the vertical alignment of data to top, middle or bottom.

❖ HTML Frameset Tag:

<frameset> Tag: The <frameset> tag in HTML is used to define the frameset. The <frameset> element contains one or more frame elements. It is used to specify the number of rows and columns in frameset with their pixel of spaces. Each element can hold a separate document.

Note: The <frameset> tag is not supported in HTML5.

Attributes frameset tag:

S.N.	Attribute	Description
1	rows	It divides the browser window in to multiple rows. The possible values of this attribute are 1) No. of pixels 2) Percentage (%) of browser window. 3) * Which indicate the remaining space.
2	cols	It divides the browser window in to multiple columns. The possible values of this attribute are 1) No. of pixels 2) Percentage (%) of browser window. 3) * Which indicate the remaining space.
3	frameborder	It specifies the frameset has border or not. The possible values are yes (1) or no (0).
4	framespacing	It specifies the amount of space between frames in a frameset.
5	bordercolor	It specifies the border color of frame.

<frame> tag: A <frame> tag is used with <frameset>, and it divides a webpage into multiple sections or frames, and each frame can contain different web pages. **OR** It defines single frame in a frameset.

Attribute of <frame> tag:

S.N.	Attribute	Description
1	src	It is the address of the HTML file or document to be loaded in to frame.
2	name	It specifies the name of frame.
3	scrolling	It specifies whether to display scrollbar or not on frame. 1. yes- Scroll bars. 2. no- no scrollbars. 3. auto- let the browser choose default.
4	noresize	It disable the frame resizing capability.
5	marginheight	It specifies the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example marginheight = "10".
6	marginwidth	It specifies the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example marginwidth = "10".

Example: Rows

```
<html>
<head>
   <title>Frame tag</title>
</head>
<frameset rows="50%,50%">
         <frame src="Frame1.html">
         <frame src="Frame2.html">>
</frameset>
</html>
```

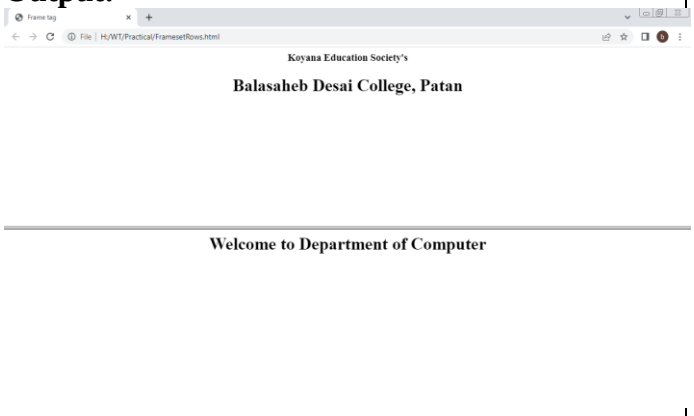
Frame1.html

```
<html>
<head>
   <title>Frame tag</title>
</head>
<body>
<h3 align="center">Koyana Education Society's</h3>
<h1 align="center">Balasaheb Desai College, Patan</h1>
</body>
</html>
```

Frame2.html

```
<html>
<head>
   <title>Frame tag</title>
</head>
<body>
<h1 align="center">Welcome to Department of Computer</h1>
</body>
</html>
```

Output:



Example: Cols

```
<html>
<head>
   <title>Frame tag</title>
</head>
<frameset cols="50%,50%">
          <frame src="Frame1.html">
          <frame src="Frame2.html">>
</frameset>
</html>
```

Frame1.html

```
<html>
<head>
   <title>Frame tag</title>
</head>
<body>
<h3 align="center">Koyana Education Society's</h3>
<h1 align="center">Balasaheb Desai College, Patan</h1>
</body>
</html>
```

Frame2.html

```
<html>
<head>
   <title>Frame tag</title>
</head>
<body>
<h1 align="center">Welcome to Department of Computer</h1>
</body>
</html>
```

Output:

❖ HTML Form elements:

<Form> tag is used to collect the input information from the user. Forms are used to creating front ends (i.e. graphical user inteface) that can contain elements like button, text fields, radio button , checkboxes, dropdown list or select list. HTML forms are created using <form> tag.

<FORM> tag: <FORM> tag is used to create form in HTML. <FORM> tag is used to collect input information from the user. <FORM> tag start with <FORM> tag and ends with </FORM> tag.

Attribute of <FORM> tag

S. N.	Attribute	Description	Example
1	Name	specify the name of form.	<form Name="frmStudent"></form>
2	Action	specifies the URL of the program that will handle the form data.	<form Action="Student.asp"></form>
3	Method	sets the method get or post by which the browser sends the form data to the server for processing.	<form Mehod="Get"></form>

<INPUT> Tag: <INPUT> tag is used to create form elements including text, password, radio, checkbox, button, submit, reset etc.

Attributes of <INPUT> tag:

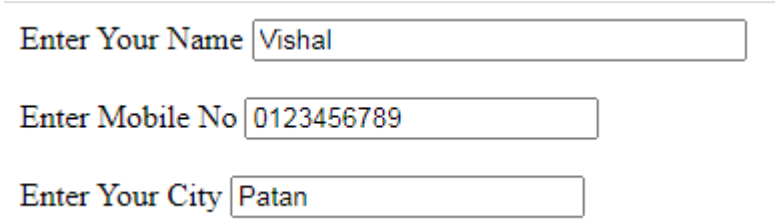
S. N.	Attribute	Description
1	Type	Specifies the the type of input field for form element. Values of type attributes are text, password, radio, checkbox, submit, reset etc.

2	Name	Assign name to the form element.
3	Size	Sets the width of input field.
4	Value	Sets the default input value.
5	Maxlength	Sets the maximum number of character you can enter into a text or password field.
6	src	Specifies the location of image for submit button.

Creating Form Elements:

1) **Text Field:** Represents a text field on a form.

```
<html>
<head>
    <title>Form Text Field</title>
</head>
<body>
<form name="form1">
        Enter Your Name <input type="text" name="txtName" size="30px"></input>
        <br>
        Enter Mobile No <input type="text" name="txtMobile" maxlength="10"></input>
        <br>
        Enter Your City <input type="text" name="txtName" value="Patan"></input>
</form>
</body>
</html>
```



2) **Password:** Represent password field in HTML form.It allow a user to enter the password securely in a webpage. The entered text in password field converted into "*" or ".", so that it cannot be read by another user.

```
<html>
<head>
    <title>Form Password Field</title>
</head>
<body>
        <form name="form1">
            Enter Password <input type="password" name="txtPassword"></input>
        </form>
</body>
</html>
```

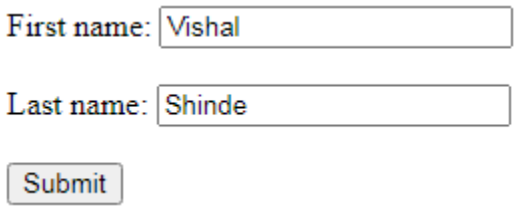
3) **Button:** The <button> tag in HTML is used to define the clickable button. <button> tag is used to submit the content.

```
<html>
```

```
<head>
		<title>Form Button Field</title>
</head>
<body>
	<form name="form1">
		<input type="button" name="btn" value="Click Me"></input>
	</form>
</body>
</html>
```

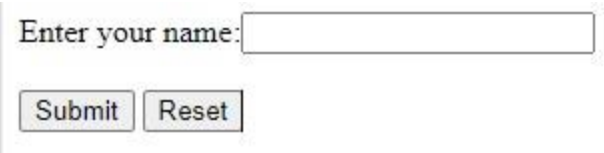
4) **Submit:** The submit button sends the information on form to server for processing. It defines a button for submitting the form data to a form-handler. The form-handler is typically a file on the server with a script for processing input data. The form-handler is specified in the form's action attribute.

```
<html>
<head>
	<title>Form Submit Button</title>
</head>
<body>
<form action="/action_page.php">
	First name: <input type="text" id="fname" name="fname" value="Vishal"><br><br>
	Last name: <input type="text" id="lname" name="lname" value="Shinde"><br><br>
	<input type="submit" value="Submit">
</form>
</body>
</html>
```



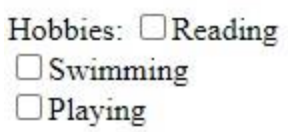
5) **Reset button:** Reset button clears or resets form values to default values.

```
<html>
<head>
	<title>Form Reset Button</title>
</head>
<body>
	<form name=”form1”>
	Enter your name:<input type=”text” name=”text1”> </input>
	<input type="Submit"  value="Submit" > </input>
	<input type="Reset"  value="Reset" > </input>
</form>
</body>
</html>
```



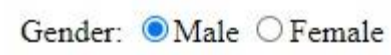
6) **Checkbox:** Checkbox allows user to select one or more option from available options.

```
<html>
<head>
        <title>Form Checkbox</title>
</head>
<body>
<form name="form1">
        Hobbies: <input type="checkbox" name="chk1">Reading </input>  <br>
        <input type="checkbox" name="chk1">Swimming </input> <br>
        <input type="checkbox" name="chk1">Playing </input>     <br>
</form>
</body>
</html>
```



7) Radio button: Radio button are similar to check boxes but difference between the radio button and checkbox is that we can select only one radio button at a time.

```
<html>
<head>
        <title>Form Radiobutton</title>
</head>
<body>
<form name="form1">
        Gender: <input type="radio" name="rd1">Male </input>
        <input type="radio" name="rd1">Female </input>
</form>
</body>
</html>
```



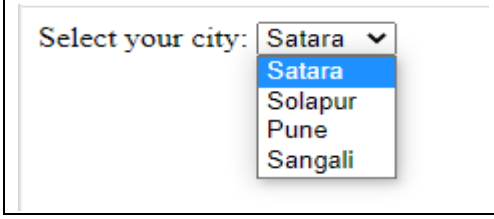
8) Select or dropdown List: <Select> tag is used to create a Dropdown list. User can select one or more options from down list. The option tag inside the <select> tag defines available option in list.

Attributes of <select> tag

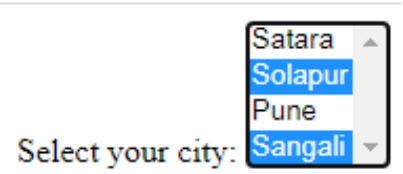
S.N.	Attribute	Description
1	Name	It specifies the unique name for drop down list.
2	Size	It specifies the number of list items to be displayed at one time.
3	Multiple	It is used to select more than one option from the list.
4	Selected	It is used to display selected item in the list.

Example:

```
<html>
<head>
        <title>Form Select Tag</title>
</head>
<body>
<form name="frmCity">
Select your city:
        <select name=”city”>
                <option> Satara </option>
                <option> Solapur </option>
                <option> Pune </option>
                <option> Sangali </option>
        </select>
</form>
</body>
</html>
```




```
<html>
<head>
<title>Form Select Tag</title>
</head>
<body>
<form name="frmCity">
Select your city:
        <select name=”city” size="4" multiple>
                <option> Satara </option>
                <option> Solapur </option>
                <option> Pune </option>
                <option> Sangali </option>
        </select>
</form>
</body>
</html>
```



9) <textarea>tag: <Textarea> tag is used to create multi line text box or text field in HTML form. To display the default text in text area field you can enter the text in between <textarea> and </textarea> tag.

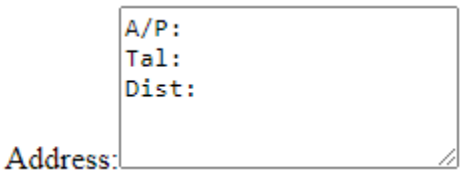
Attributes of <select> tag:

S. N.	Attribute	Description
1	Name	It specifies the unique name for textarea.
2	rows	It sets the number of rows displayed in the text area.
3	cols	It sets the number of columns displayed in the text area.

Example:

```
<html>
<head>
        <title>Form Textarea</title>
</head>
<body>
<form name="frmCity">
        Address:
        <textarea rows=5 cols=20>
        A/P:
        Tal:
        Dist:
```

```
        </textarea>
    </form>
    </body>
    </html>
```



❖ HTML <form> method GET & POST:

The **HTML <form> method Attribute** is used to specify the HTTP method used to send data while submitting the form. There are two kinds of HTTP methods, which are **GET** and **POST**. The method attribute can be used with the **<form>** element.

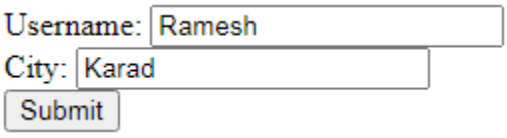
Attribute Values:

- **GET:** In the GET method, after the submission of the form, the form values will be visible in the address bar of the new browser tab. It has a limited size of about 3000 characters
- **POST:** In the post method, after the submission of the form, the form values will not be visible in the address bar of the new browser tab as it was visible in the GET method.

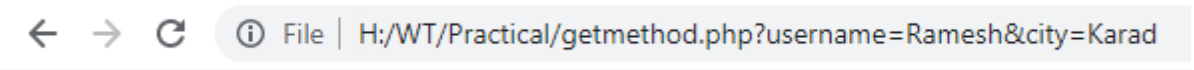
Example: Get Method

```
<html>
<head>
    <title>Get Method</title>
</head>
<body>
<form action="getmethod.php" method="GET">
        Username: <input type="text" name="username" /> <br>
        City: <input type="text" name="city" /> <br>
        <input type="submit" />
</form>
</body>
</html>
```

Output:



After submit the form values are visible in address bar

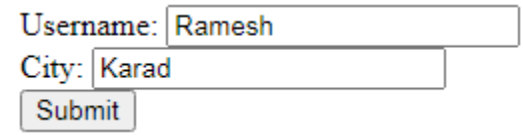


Example: Method POST

```
<html>
<head>
    <title>Post Method</title>
```

```
</head>
<body>
<form action="postmethod.php" method="POST">
        Username: <input type="text" name="username" /> <br>
        City: <input type="text" name="city" /> <br>
        <input type="submit" />
</form>
</body>
</html>
```

Output:



After submit the form values are not visible in address bar

❖ Difference between Get & Post Method in Form

GET	POST
1) In case of Get request, only **limited amount of data** can be sent because data is sent in header.	In case of post request, **large amount of data** can be sent because data is sent in body.
2) Get request is **not secured** because data is exposed in URL bar.	Post request is **secured** because data is not exposed in URL bar.
3) Get request **can be bookmarked.**	Post request **cannot be bookmarked.**
4) Easier to hack.	More difficult to hack
5) Get request is **more efficient** and used more than Post.	Post request is **less efficient** and used less than get.
6) GET method should not be used when sending passwords or other sensitive information.	POST method used when sending passwords or other sensitive information.

❖ Basics of CSS:

CSS stands for cascading style sheet. CSS are needed for creating dynamic web pages. They give more control in page layout styling of document. They can be declared in head section of the page or directly within HTML page. All major browsers supports cascading style sheet. Cascading style sheet are normally saved in external **.CSS file**. External style sheet helps to change the appearance (look) of all web pages in website.

CSS syntax: Selector {property: value}

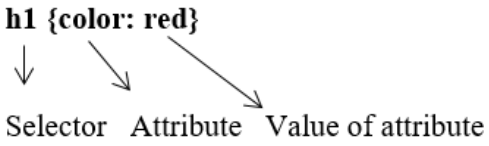
Where

Selector- any HTML tag.

Property- It is attribute of HTML tag.

Value- Value of attribute.

Example:



If the value multiple words then puts quotes (“ ”) around the value.

Example

P{font-family:”Times New Roman”}

If the properties are more, then they are separated by semicolon (;)

Example

P {text-align: centre; color: red}

Types of CSS:

There are three types of CSS

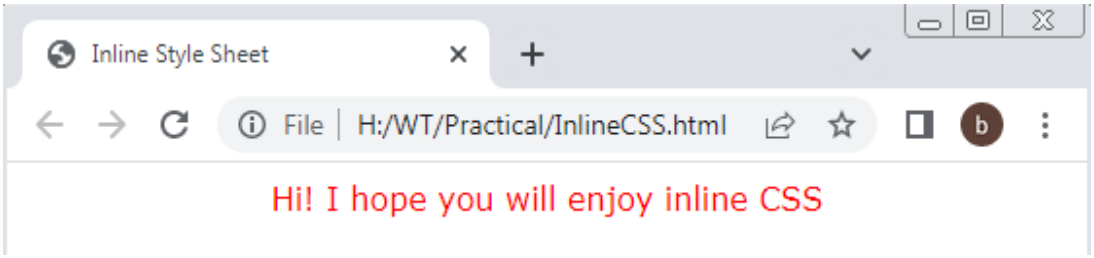
1) Inline style sheet
2) Internal style sheet
3) External style sheet

1) **Inline style sheet:** Inline style sheet is used to apply CSS properties to individual elements or HTML tags on web page.

Example:

```
<html>
<head>
 <title> Inline Style Sheet </title>
</head>
<body>
      <p style="color:red;text-align:center;font-family:Verdana">Hi! I hope you will enjoy inline CSS</p>
</body>
</html>
```

Output:



2) **Internal Style sheet:** An internal style sheet is used when style is to be applying to particular one webpage.

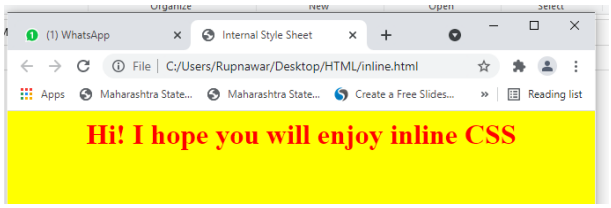
The internal style sheet can be defined in **head section by using <style> tag.**

Example:

```
<html>
<head>
```

```
    <title> Internal Style Sheet </title>
    <style>
            body{background-color:yellow}
            h1{color:red;text-align:center}
    </style>
</head>
<body>
        <h1 > Hi! I hope you will enjoy inline CSS </h1>
</body>
</html>
```

Output:



3) **External Style Sheet:** An external style sheet is ideal when the style is applied to many pages. External style sheet is simply a plain text file containing the style specification for HTML tag and having extension **.CSS.** External style sheet is used in HTML document by using **<link> tag** within head section. External file should not contain any HTML tags.

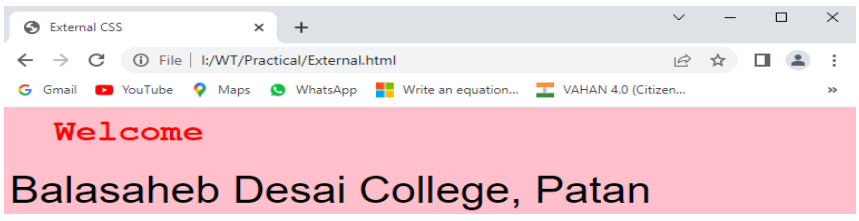
Example: Demo.css

```
body{background-color:pink;}
h1{color:red;margin-left: 40px;font-family:"Courier";}
font{font-size:40;font-family:"Arial";}
```

Example: External.css

```
<html>
<head>
    <title>External CSS</title>
    <link type="text/css" href="Demo.css" rel="stylesheet"> </link>
</head>
<body>
    <h1>Welcome</h1>
    <font>Balasaheb Desai College, Patan</font>
</body>
</html>
```

Output:



❖ Importance of CSS:

CSS Selectors-Group, id, class

1) **Class selector:** With the class selector you can define different style for the same type of HTML tag. A class selector is always start with a period (.) which is followed by class name. The style is defined in curly braces and contains style property and its value.

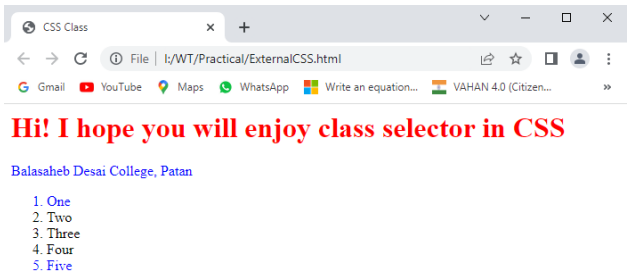
Syntax: .classname{property:value}

The class selectors are defined in <style> tag within the head section of the HTML document.

Example:

```
<html>
<head>
        <title>CSS Class</title>
            <style>
                    .sample {color: red; text-align: left;}
                    .bca {color: blue; text-align: left;}
            </style>
</head>
<body>
        <h1 class="sample" > Hi! I hope you will enjoy class selector in CSS </h1>
            <font class="bca">Balasaheb Desai College, Patan</font>
            <ol>
                    <li class="bca">One</li>
                    <li>Two</li>
                    <li>Three</li>
                    <li>Four</li>
                    <li class="bca">Five</li>
            </ol>
  </body>
  </html>
```

Output:



2) **Id Selector:** The functionality of both class and ID selector is same. The ID attribute is used to define a unique style for an element or HTML tag. ID selector is always start with hash character (#).

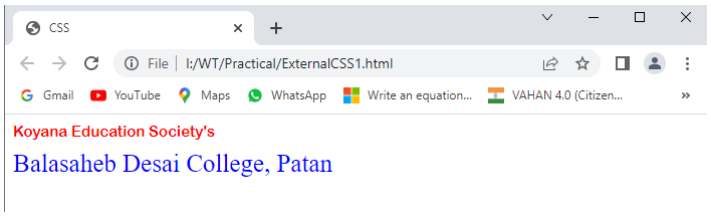
Syntax: #idname{property:value}

Example:

```
<html>
<head>
        <title>CSS</title>
        <style>
```

```
        #sample {color: red; text-align: left;font-family:"Arial Rounded MT";font-size: 15px;}
        #bca {color: blue; text-align: left;font-size: 25px;}
    </style>
</head>
<body>
    <h1 id="sample">Koyana Education Society's</h1>
    <font id="bca">Balasaheb Desai College, Patan</font>
</body>
</html>
```

Output:



If another tag <p> tag uses this

<p id=”sample”> BCA </P>

Then the rule will not match.id selector is unique an applies to only one element or tag.

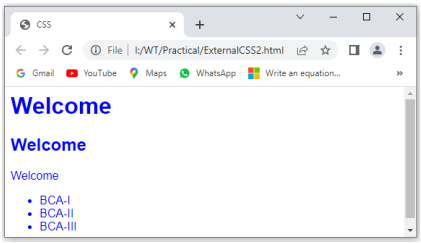
3) **Group Selector:** If we want to apply same style to two or more elements for HTML tags then we can group such element or tag. Group selectors are separated with comma.

Syntax: Selector 1, selector 2, …. {Property: value}

Example:

```
<html>
<head>
    <title>CSS</title>
    <style>
        h1, p, h2, h3,ul {color: blue; text-align: left;font-family:"Arial";}
    </style>
</head>
<body>
    <h1> Welcome </h1>
    <h2> Welcome </h2>
    <p> Welcome </p>
    <ul>
        <li>BCA-I</li>
        <li>BCA-II</li>
        <li>BCA-III</li>
    </ul>
</body>
</html>
```

Output:



❖ CSS Properties:

1) Border Properties: The **border** property in CSS is used to style the border of an element. This property is a combination of three other properties border-width, border-style, and border-color.

There are three properties of a border you can change −

i) The **border-color** specifies the color of a border.

ii) The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.

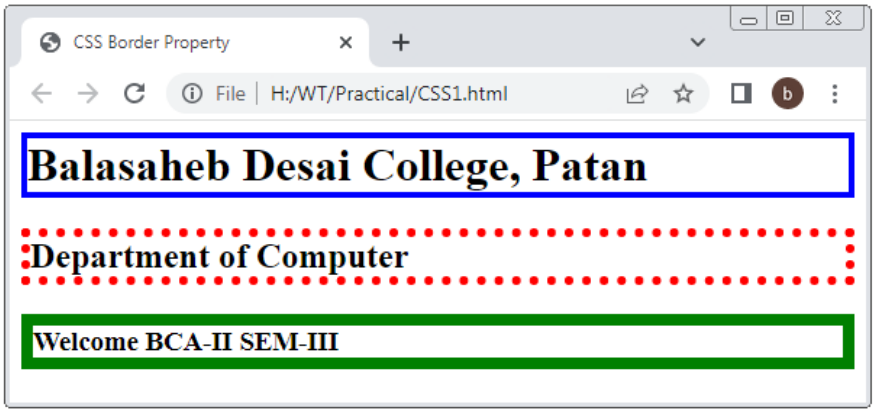
iii) The **border-width** specifies the width of a border.

i) The border-color Property: The border-color property allows you to change the color of the border.

- **border-bottom-color** changes the color of bottom border.
- **border-top-color** changes the color of top border.
- **border-left-color** changes the color of left border.
- **border-right-color** changes the color of right border.

Example:

```
<html>
<head>
<title>CSS Border Property</title>
<style>
  h1
  {
        border-width:4px;
        border-style:solid;
        border-color:blue;
  }
  h2
  {
        border-width:6px;
        border-style:dotted;
        border-color:red;
  }
  h3
  {
        border-width:8px;
        border-style:solid;
        border-color:green;
  }
</style>
</head>
<body>
  <h1>Balasaheb Desai College, Patan</h1>
  <h2> Department of Computer </h2>
  <h3> Welcome BCA-II SEM-III</h3>
</body>
</html>
```



2) CSS Margin: CSS Margin property is used to define the space around elements.

CSS Margin Properties

Property	Description
margin	This property is used to set all the properties in one declaration.
margin-left	It is used to set left margin of an element.
margin-right	It is used to set right margin of an element.
margin-top	It is used to set top margin of an element.
margin-bottom	It is used to set bottom margin of an element.

CSS Margin Values

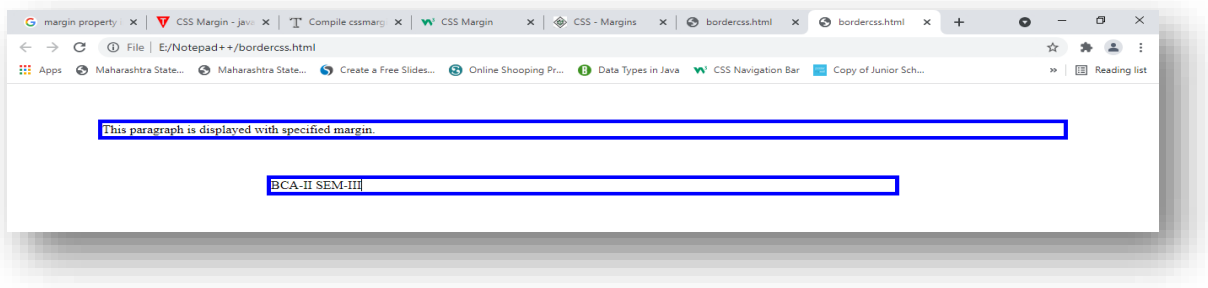
These are some possible values for margin property.

Value	Description
auto	This is used to let the browser calculate a margin.
length	It is used to specify a margin pt, px, cm, etc. its default value is 0px.
%	It is used to define a margin in percent of the width of containing element.
inherit	It is used to inherit margin from parent element.

Example:

```
<html>
<head>
   <title>CSS Border Property</title>
<style>
p{
   margin-top: 50px;
   margin-bottom: 50px;
   margin-right: 100px;
   margin-left: 100px;
   border:5px solid blue;
}
</style>
</head>
```

```
<body>
        <p>This paragraph is displayed with specified margin.</p>
        <p style=”margin-left: 300px;margin-right:300px;”>BCA-II SEM-III</p>
</body>
</html>
```



Margin - Shorthand Property:

To shorten the code, it is possible to specify all the margin properties in one property.

i) If the margin property has four values:

e.g. margin: 25px 50px 75px 100px;

- top margin is 25px
- right margin is 50px
- bottom margin is 75px
- left margin is 100px

ii) If the margin property has three values:

e.g. margin: 25px 50px 75px;

- top margin is 25px
- right and left margins are 50px
- bottom margin is 75px

iii) If the margin property has two values:

e.g. margin: 25px 50px;

- top and bottom margins are 25px
- right and left margins are 50px

iv) If the margin property has one value:

e.g. margin: 25px;

- all four margins are 25px

3) CSS Background: CSS background property is used to define the background effects on element. There are 5 CSS background properties.

i) background-color

ii) background-image

iii) background-repeat

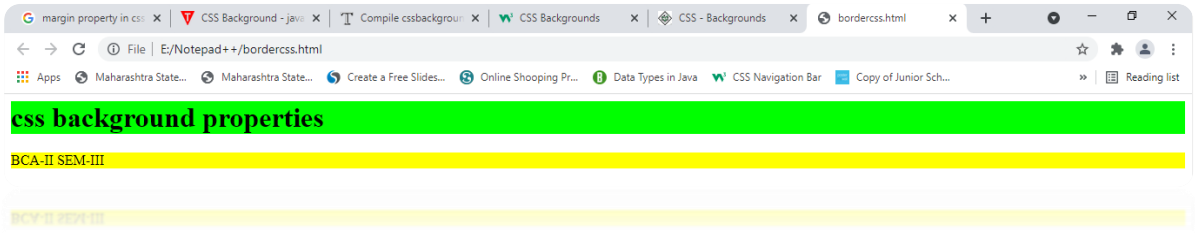
iv) background-attachment

v) background-position

i) CSS background-color: The background-color property is used to specify the background color of the element.

Example:

```
<html>
<head>
<style>
        p{
                background-color:yellow;
        }
        h1{
          background-color:lime;
  }
</style>
</head>
<body>
        <h1>css background properties</h1>
        <p>BCA-II SEM-III</p>
</body>
</html>
```



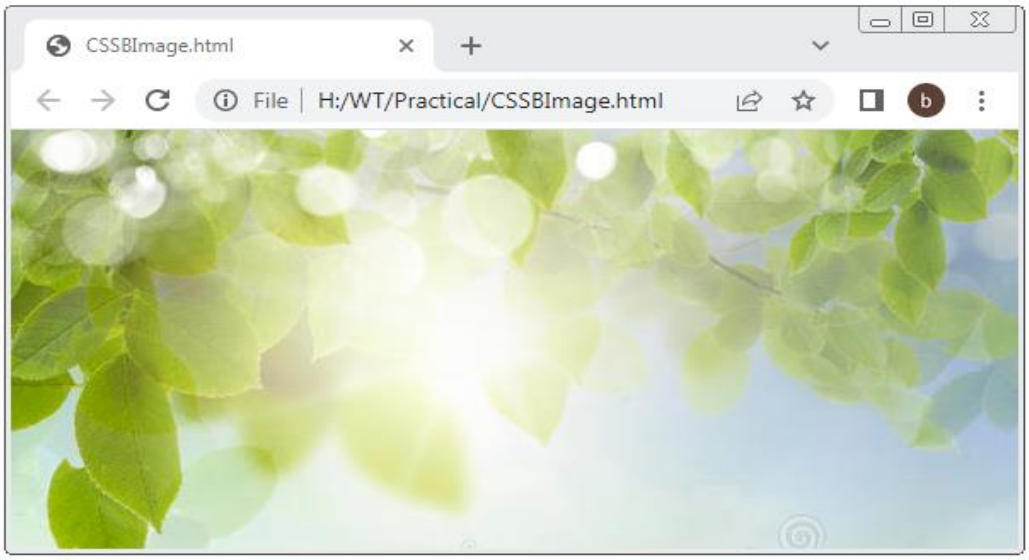
ii) CSS background-image: The background-image property is used to set an image as a background of an element.

Example:

```
<html>
<head>
<title>CSS Background Image</title>
  <style>
  body {
    background-image:url("../img1.jpg");
  }
```

```
</style>
</head>
<body>
</body>
</html>
```

Example:



iii) **CSS background-repeat:** By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

- If we set background-repeat: repeat-x; then image will repeat horizontally.
- If we set background-repeat: repeat-y; then image will repeat vertically.
- If we set background-repeat: no-repeat; then image will not repeat.

iv) **CSS background-attachment:** The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):

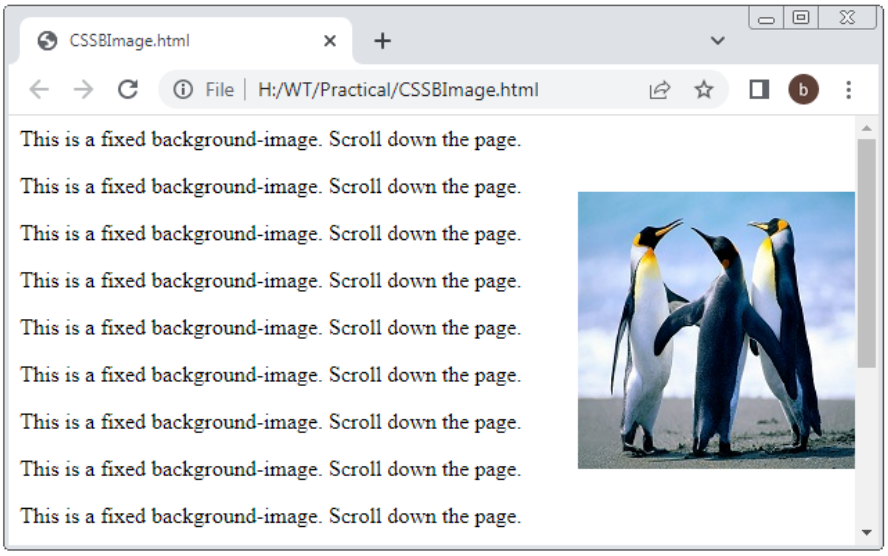
v) **CSS background-position:** The background-position property is used to specify the position of the background image.

You can set the following positions: center, top, bottom, left, right

Example:

```
<html>
<head>
<style>
body  {
background: url('../Penguins.jpg');
background-repeat: no-repeat;
background-attachment: fixed;
background-position:right;
```

```
}
</style>
</head>
<body>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>This is a fixed background-image. Scroll down the page.</p>
<p>If you do not see any scrollbars, Resize the browser window.</p>
</body>
</html>
```



4) CSS Image Properties: Images play an important role in any webpage. With the help of css we can control the display of image. We can set following css properties for image.

i) The **border** property is used to set the width of an image border.

ii) The **height** property is used to set the height of an image.

iii) The **width** property is used to set the width of an image.

iv) The **opacity** property is used to set the opacity of an image.

i) **The Image Border Property:** The border property of an image is used to set the width of an image border. This property can have a value in length or in %. A width of zero pixels means no border.

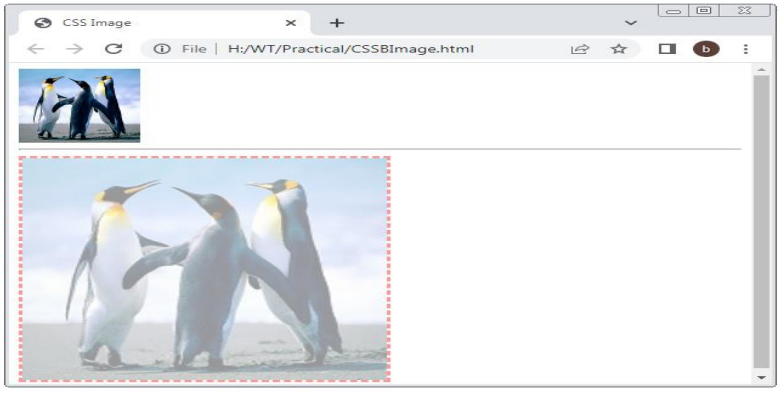
ii) **The Image Height Property:** The height property of an image is used to set the height of an image. This property can have a value in length or in %.

iii) **The Image Width Property:** The width property of an image is used to set the width of an image. This property can have a value in length or in %.

iv) **The opacity Property:** The opacity property of an image is used to set the opacity of an image. The opacity property can take a value from 0.0 - 1.0. The lower value, the more transparent:

Example:

```
<html>
<head>
  <title>CSS Image</title>
</head>
<body>
  <img style = "border:0px;height:100px;width:100px;" src="../Penguins.jpg" />
  <hr>
  <img style = "border:3px dashed red;height:300px;width:300px;opacity:0.4;"
       src="../Penguins.jpg" />
</body>
</html>
```



5) CSS Lists: There are various CSS properties that can be used to control lists. Lists can be classified as ordered lists and unordered lists.

The CSS properties to style the lists are given as follows:

i) **list-style-type:** This property is responsible for controlling the appearance and shape of the marker.

ii) **list-style-image:** It sets an image for the marker instead of the number or a bullet point.

iii) **list-style-position:** It specifies the position of the marker.

iv) **list-style:** It is the shorthand property of the above properties.

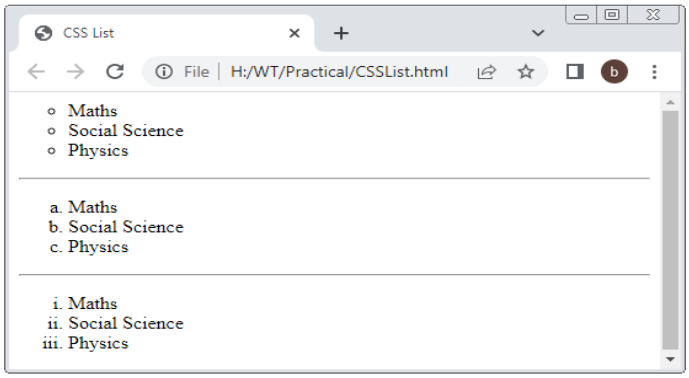
v) **marker-offset:** It is used to specify the distance between the text and the marker. It is unsupported in IE6 or Netscape 7.

i) **The list-style-type property:** It allows us to change the default list type of marker to any other type such as square, circle, roman numerals, Latin letters, and many more. By default, the ordered list items are numbered with Arabic numerals (1, 2, 3, etc.), and the items in an unordered list are marked with round bullets (•).

Example:

```
<html>
<head>
  <title>CSS List</title>
</head>
<body>
  <ul style = "list-style-type:circle;">
        <li>Maths</li>
        <li>Social Science</li>
        <li>Physics</li>
  </ul>
  <hr />
  <ol style = "list-style-type:lower-alpha;">
        <li>Maths</li>
        <li>Social Science</li>
        <li>Physics</li>
  </ol>
  <hr />
  <ol style = "list-style-type:lower-roman;">
        <li>Maths</li>
        <li>Social Science</li>
        <li>Physics</li>
  </ol>
</body>
</html>
```

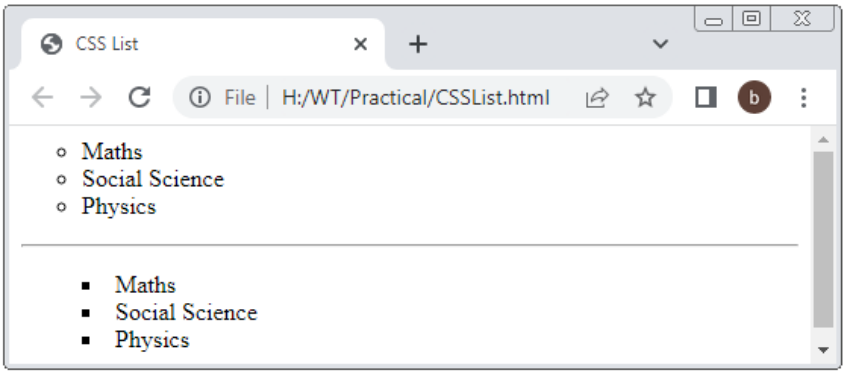
Output:



ii) **The list-style-position property:** It represents whether the appearing of the marker is inside or outside of the box containing the bullet points.

Example:

```
<html>
<head>
  <title>CSS List</title>
</head>
<body>
  <ul style = "list-style-type:circle; list-stlye-position:outside;">
            <li>Maths</li>
            <li>Social Science</li>
            <li>Physics</li>
  </ul>
  <hr>
  <ul style = "list-style-type:square;list-style-position:inside;">
            <li>Maths</li>
            <li>Social Science</li>
            <li>Physics</li>
  </ul>
</body>
</html>
```

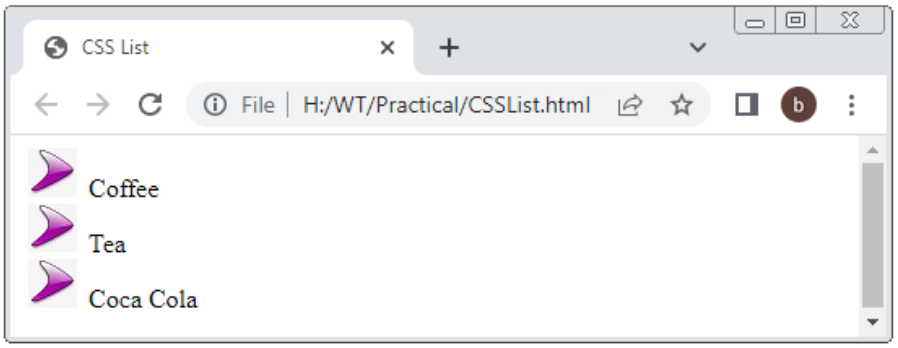


iii) **The list-style-image property:** It specifies an image as the marker. Using this property, we can set the image bullets.

Example:

```
<html>
<head>
  <title>CSS List</title>
</head>
<body>
<ul style="list-style-image: url('../bullet.jpg');">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
</body>
```

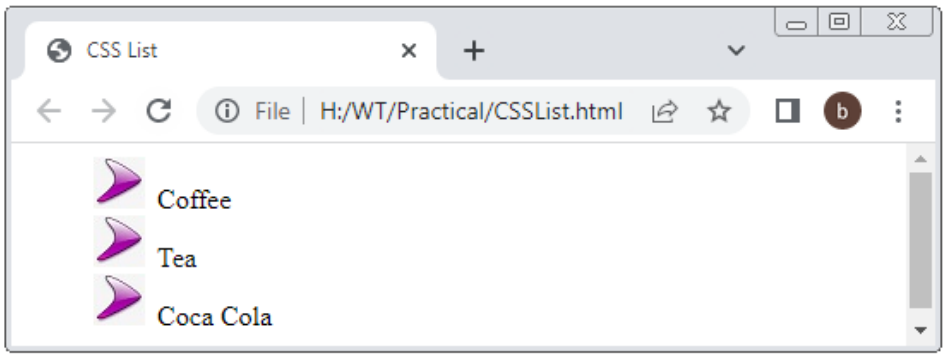
</html>



iv) The list-style Property: The list-style allows you to specify all the list properties into a single expression.

Example:

```
<html>
<head>
  <title>CSS List</title>
</head>
<body>
<ul style="list-style:square inside url('../bullet.jpg');">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
</body>
</html>
```



Advantages of CSS:

1. CSS saves time: You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

2. Easier to maintenance:

CSS takes less maintenance time. This is because if any changes are required for any web page or document then changes can be simply made in a CSS file.

3. Positioning of web page elements:

You can change the position of an HTML tag with the help of CSS. You can place the elements like an image on any part of the webpage as and when required.

4. Multiple Device Compatibility:

Like HTML, CSS supports all types of electronic gadgets such as smartphone, laptop, tablet, desktop etc.

6. CSS improves the loading speed of the page.

7. Platform Independent: The scripts are platform independent and support all the latest browsers.

8. Data Integrity: CSS allows to maintain the integrity of data.

Limitations/Disadvantages of CSS:

1. CSS cannot perform any logical operations like if/else or for/while or +/-.
2. We cannot read any files using CSS.
3. It cannot interact with databases.
4. CSS cannot request a web page.
5. CSS has cross-browser issues.
6. It has multiple levels like **CSS1, CSS2, CSS3,** which are sometimes confusing for beginners.
7. Sometimes CSS can be messy and can create complications in code.
8. One of the major disadvantages of CSS is that it has limited security.

****

Unit-III

JAVA Script

❖ Introduction of JavaScript:

JavaScript is a lightweight, interpreted **programming** language. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform. JavaScript is used in millions of Web pages to improve the design, validate forms, detect browsers, create cookies, and much more. JavaScript is the most popular scripting language on the internet, and works in all major browsers, such as Internet Explorer, Mozilla, Firefox, Netscape, Opera etc.

JavaScript is executed by Java compatible web browsers. Almost all popular current web browsers supports JavaScript. The JavaScript was formerly developed by **Brandon Eich** from Netscape communication in 1995. The language was initially called **LiveScript** and was later renamed JavaScript. Java and JavaScript are two completely different languages. JavaScript was first supported by Netscape navigator browsers.

❖ Features of JavaScript:

1) **Validating User's Input:**

   JavaScript is very useful while using forms. It has the capability to validate user input for errors and also saves time. E.g. If the user leaves a required field empty or the information is incorrect, JavaScript checks for them before sending the data over to the server.

2) **Light Weight Scripting Language:**

   JavaScript is a lightweight scripting language because it is made for data handling at the browser only. Since it is not a general-purpose language so it has a limited set of libraries. Also as it is only used for client-side execution, hence the lightweight nature of JavaScript is a great feature.

3) **Simple Client-side Calculations:**

   Since JavaScript is a client-side technology, it can perform basic calculations on the browser. The browser does not need to ask server time for every task. This is especially helpful when a user needs to perform these calculations repeatedly.

4) **Platform Independent:**

   JavaScript is platform-independent or we can say it is portable; which simply means that you can simply write the Java script programs once and run it anywhere and anytime. In general, you can write your JavaScript applications and run them on any platform or any browser.

5) **Interpreted Language:**

JavaScript is an interpreted language which means the script written inside javascript is processed line by line. These Scripts are interpreted by JavaScript interpreter which is a built-in component of the Web browser

6) **JavaScript can react to events:**

A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.

7) **Handling Dates and Time:**

Unlike other programming languages, JavaScript has built-in functions to determine the date and time. Thus it is very easy to code only by using methods like **.getDate().**

8) **JavaScript can put dynamic text into an HTML page:**

A JavaScript statement like this: document.write("<h1>" + name + "</h1>") can write a variable text into an HTML page

❖ Limitation of JavaScript:

1) Client-side JavaScript does not allow the reading or writing of files.
2) It cannot be used for networking applications because there is no such support available.
3) It doesn't have any multithreading or multiprocessor capabilities.
4) This may be difficult to develop large applications.
5) The main problem or disadvantage in JavaScript is that the code is always visible to everyone anyone can view JavaScript code.
6) If the error occurs in the JavaScript, it can stop the output to the whole website. Browsers doesn't show the JavaScript errors.

❖ Difference between Client Side Scripting and Server Side Scripting:

Sr. No.	Client Side Scripting	Server Side Scripting
1	The client side scripts are programs runs on client computer.	The server side scripts are programs runs on server.
2	Client side scripting depends on memory and CPU speed of a client.	Server side scripting depends on memory and CPU speed of server.
3	Client side code executed at the client side and displays the result in its browser.	The server side code is executed on server side and display the result on requested client.
4	It is faster than server side code.	It is slower than client side code.

5	It can handle limited amount of data.	It can handle large amount of data.
6	Client side scripting cannot be used to connect to the databases on the web server.	Server side scripting is used to connect the database that resides on the web server.
7	Client side scripting provides less security than Server side scripting.	Server side scripting provides more security than Client side scripting.
8	Source code is visible to user.	Source code is not visible to user.
9	It cannot access the file system that resides at the web server.	Server side scripting can access the file system residing at the web server.
10	For example Client side script languages-JavaScript, VB script.	For example server side scripting languages- ASP, PHP, Pearl etc.

❖ First JavaScript Program:

```
<html>
<head>
    <title>JavaScript Programs</title>
    <script language="JavaScript">
            document.write("Welcome to JavaScript!");
    </script>
</head>
<body>
    My First JavaScript Program
</body>
</html>
```

❖ Variables in JavaScript:

A JavaScript variable is simply a name of storage location. Variable are used to store the value. A variable's value can change during the script. In JavaScript variables declared with var, let, or const keyword.

- The var keyword is used in all JavaScript code from 1995 to 2015.
- The let and const keywords were added to JavaScript in 2015.
- If you want your code to run in older browsers, you must use var.

Declare a Variable:

You can create a variable with the var statement:

```
var var_name = some value;
```

Example: var age=18; OR let age=18; OR const age=18;

You can also create a variable without the var statement:

var_name = some value;

Example: age=18;

There are two types of variables in JavaScript: local variable and global variable. A JavaScript local variable is accessible within the function or block only. A JavaScript global variable is accessible from any function.

Rules for declaring variable:

- Variable name always begin with the letter.
- It does not contain any blank space.
- It cannot be reserved words.
- Variable name are case sensitive.

IMPORTANT! JavaScript is case-sensitive! A variable named strname is not the same as a variable named STRNAME.

❖ Functions in JavaScript:

A JavaScript function is a block of code designed to perform a particular task. JavaScript functions are used to perform operations. We can call JavaScript function many times to reuse the code.

There are two types of function in JavaScript.

1) Built in function
2) User defined function

1) Built-in Functions: JavaScript provides several built-in functions that can be used to perform explicit type conversions. Some of them are **eval(), parseInt()** and **parseFloat().**

i) eval() : The **eval()** function can be used to convert a string expression to a numeric value. **Example:**

var a = eval("10 * 10 + 5");

Output: a=105

ii) parseInt(): The parseInt() function is used to covert a string value to an integer. The parseInt() function returns the integer value from string expression.

Example: var a = parseInt("123xyz");

It returns the 123 and assign to variable a;

var a = parseInt("xyz");

Output: a=NaN (Not A Number)

iii) parseFloat(): The parseFloat() function is used to covert a string value to Floating type value. The parseFloat() function returns the Floating type value from string expression.

Example: var a = parseFloat(“123.5xyz”);

It returns the 123.5 and assign to variable a;

2) **User defined functions in JavaScript:** JavaScript allows us to create user-defined functions also. We can create functions in JavaScript using the keyword **function.** The basic syntax to create a function in JavaScript is

Syntax:

```
function functionname(parameters)
{
        //function body
}
```

Where,

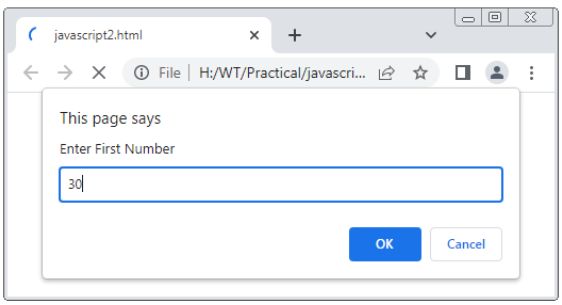
functionname: It is the name of function.

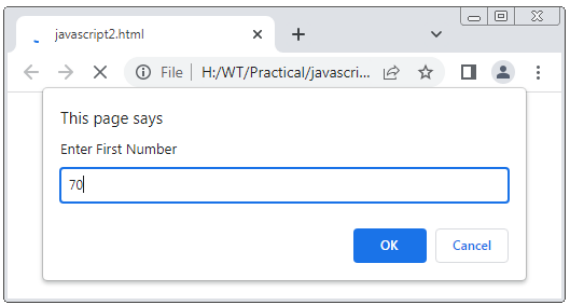
parameters: It is the list of arguments passed to the function.

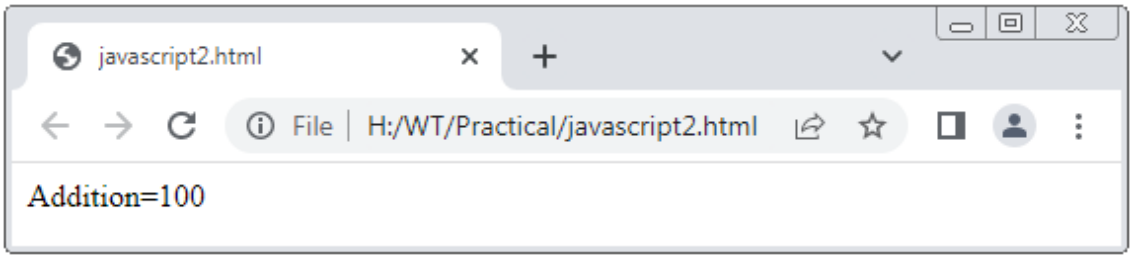
Example:

```
<html>
<head>
<script type="text/javascript">
        function add()
        {
                var a,b,c;
                a=prompt("Enter First Number");
                b=prompt("Enter Second Number");
                c= parseInt(a)+parseInt(b);
                document.write("Addition="+c);
        }
</script>
</head>
<body>
<script type="text/javascript">
        //add();
</script>
</body>
</html>
```

Output:







❖ Dialog Boxes:

Dialog boxes are used to give important notification to the user and also can take inputs from user. There are mostly three types of dialog boxes in JavaScript. They are used to either show confirmation messages, raise an error, or show a warning message. You can get input also from these dialog boxes. The following are the dialog boxes in JavaScript:

1) Alert Dialog Box
2) Prompt Dialog Box
3) Confirmation Dialog Box

1) Alert() Dialog box:

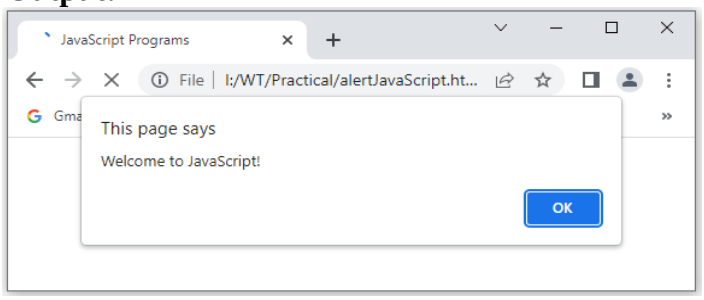
It is used to provide a warning message to users. It is one of the most widely used dialog box in JavaScript. It has only one **'OK'** button to continue and select the next task.

Syntax: alert(“message”);

Example:

```
<html>
<head>
<title>JavaScript Programs</title>
<script language="JavaScript">
        alert("Welcome to JavaScript!");
</script>
</head>
<body>
</body>
</html>
```

Output:



2) Confirm() dialog box:

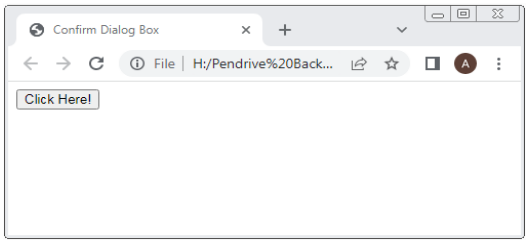
It displays confirmation dialogue box. Confirm dialogue box displays message with **ok** and **cancel** button. This method returns true if ok button is pressed.

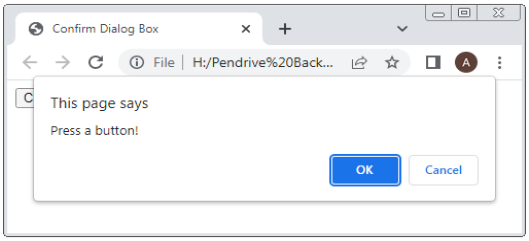
Syntax: confirm(“message”);

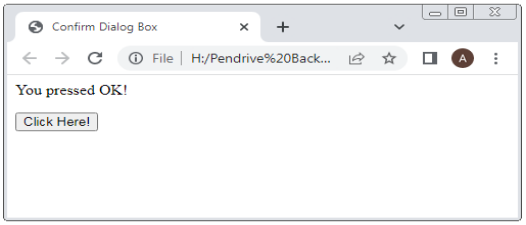
Example: confirm("Are You Sure To Exit!");

```
<html>
<head>
<title>Confirm Dialog Box</title>
```

```
<script>
	function myFunction()
	{
		var text;
		if (confirm("Press a button!") == true)
		{
			text = "You pressed OK!";
		}
		else
		{
			text = "You canceled!";
		}
			document.getElementById("demo").innerHTML = text;
	}
</script>
</head>
<body>
<p id="demo"></p>
<form name="myForm">
<input type="button" value="Click Here!" onclick="myFunction()">
</form>
</body>
</html>
```







3) Prompt() dialog box:

Prompt dialogue box is used to take input from the user.

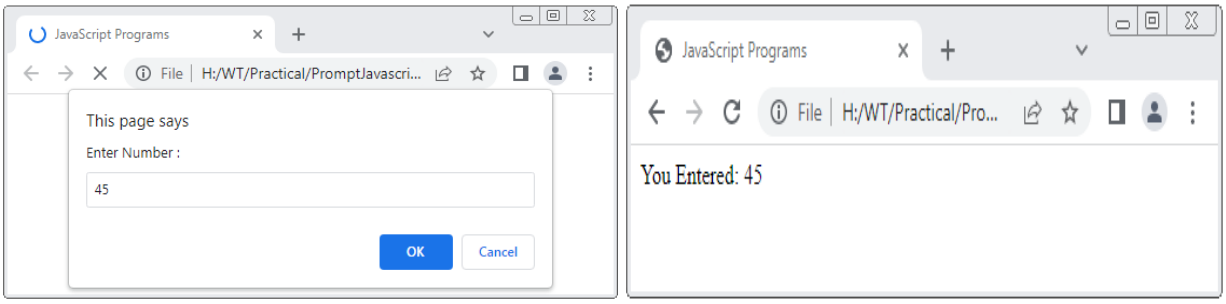
Syntax: prompt("message","default value");

Example: prompt("Enter number","5");

```
<html>
<head>
	<title>JavaScript Programs</title>
	<script language="JavaScript">
		var number;
		number=prompt("Enter Number :");
	</script>
```

```
</head>
<body>
        <script language="JavaScript">
                document.write("You Entered: "+number);
        </script>
</body>
</html>
```

Output:



❖ JavaScript Identifiers:

An identifier is a name that is given to entities like variables, functions, class, etc.

Java identifiers follow the following rules:

i) Identifier names must start with either a letter, an underscore _, or the dollar sign $.

For example: let _age; or var _age; var $name; //Valid identifiers

ii) Identifier names cannot start with numbers.

For example: var 01age; // Invalid identifier

iii) JavaScript is case-sensitive. So age and Age are different identifiers.

For example: var age=18; var Age=20;

iv) Keywords cannot be used as identifier names.

For example: var while; // Invalid identifier because while is the keyword.

❖ JavaScript Operators:

In JavaScript, an operator is a special symbol used to perform operations on operands (values and variables). There are following types of operators in JavaScript.

1) Arithmetic Operators
2) Comparison (Relational) Operators
3) Bitwise Operators
4) Logical Operators
5) Assignment Operators
6) Increment & Decrement Operators
7) Special Operators

1) Arithmetic Operators:

Arithmetic operators are used to perform arithmetic operations on the operands.

Operator	Meaning	Example	Result
+	Addition	a+b	10+20 = 30
_	Subtraction	a-b	20-10 = 10
*	Multiplication	a*c	2 * 5 = 10
/	Division	a/b	10 / 2 = 5
%	Modulo Division (Remainder)	a%b	20 / 10 = 0

2) Comparison (Relational) Operators:

We often compare two quantities, and depending on their relation, take certain decisions.

These comparisons can be done with the help of relational operators.

Operator	Meaning	Example	Result
<	is less than	a<b	20<10 = false
<=	is less than or equal to	a<=b	20<=10 = false
>	is greater then	a>b	20>10 = true
>=	is greater than or equal to	a>=b	20>=10 = true
==	is equal to	a==b	10==20 = false
!=	is not equal to	a!=b	10!=20 = true

3) Bitwise Operators:

Bitwise operators are used for manipulation of data at values of bit level.

Operator	Meaning	Example
&	bitwise AND	5 & 1=1
|	bitwise OR	5 | 1=5
^	bitwise exclusive OR	5 ^ 1=4
~	one's complement	~ 5=10
<<	shift left	5 << 1=10
>>	shift right	5 >> 1=2

4) Logical Operators:

Java has three logical operators.

Operator	Meaning	Example
&&	Logical AND	a>10 && a<20
||	Logical OR	a>10 || a<20
!	Logical NOT	a!=10

5) Assignment Operators:

Assignment operators are used to assign the value of an expression to a variable.

Statement with simple assignment operator	Statement with shorthand operator
a=a+1	a+=1
a=a-1	a-=1
a=a*(n+1)	a*=n+1
a=a/(n+1)	a/=n+1
a=a%b	a%=1

6) Increment & Decrement Operators:

JavaScript has two very useful operators. These are the increment and decrement operators: ++ and --.

The operator ++ adds 1 to the operand while -- subtracts 1.

Syntax: ++m or m++ is equivalent to m=m+1 or m+=1

7) JavaScript Special Operators:

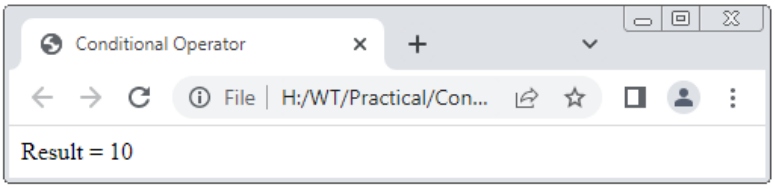
The following operators are known as JavaScript special operators.

i) Conditional Operator (? :) : Conditional Operator returns value based on the condition. It is like if-else.

Syntax: expl ? exp2 : exp3
Example: Y = (6>5) ? 6 : 5 therefore Y=6

```
<html>
<head>
<title>Conditional Operator</title>
</head>
<body>
<script type="text/javascript">
        var a = 10;
        var b =20;
        var c=(a<b) ? a : b;
        document.write("Result = " + a);
</script>
</body>
</html>
```

Output:



ii) typeof Operator: It returns the type of a variable.

Syntax: typeof variable;

```
<html>
<head>
```

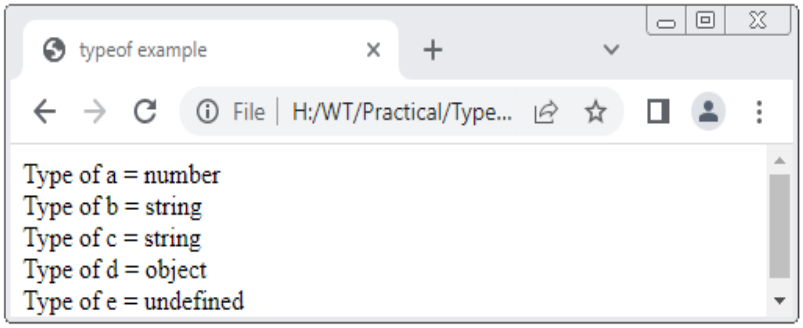
```
        <title>typeof example </title>
</head>
<body>
<script type="text/javascript">
        var a = 17;
        var b = "BCA-II";
        var c = "";
        var d = null;

        document.write("Type of a = " + (typeof a));
        document.write("<br>");
         document.write("Type of b = " + (typeof b));
        document.write("<br>");

        document.write("Type of c = " + (typeof c));
        document.write("<br>");
         document.write("Type of d = " + (typeof d));
        document.write("<br>");

        document.write("Type of e = " + (typeof e));
        document.write("<br>");
</script>
</body>
</html>
```

Output:



❖ Control Structures in JavaScript:

The structure which determines the flow or jump of program depending on condition is called as control structure. Every programming language, basically, has three types of control statements.

1) Conditional Statements (Decision)
2) Iterative Statements (Loop)
3) Jumps Statements.

1) Conditional Statements (Decision Making):

Conditional statement results in either True or False. Whatever the condition is passed, if that is true, then the program moves to the next step and if the condition is False, then the program moves to another step. Following are the different types of Conditional Statements:

i) if Statement ii) if …else iii) switch statement

i) if statement:

IF statement is a conditional branching statement. In 'IF' statement, if the condition is true a group of statement is executed. And if the condition is false, the following statement is skipped.

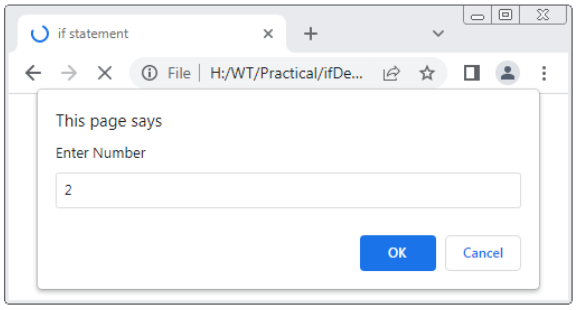
Syntax:

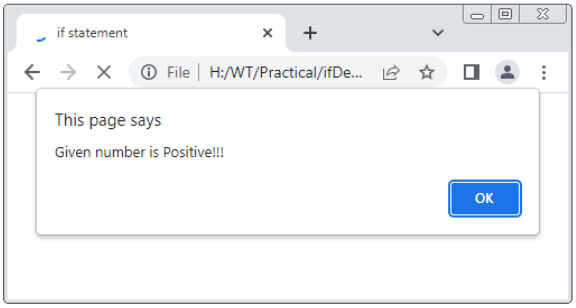
```
if(condition)
{
        //Statement 1;
        //Statement 2;
}
```

Example:

```
<html>
<head>
    <title>if statement</title>
</head>
<body>
<script type="text/javascript">
    var num = prompt("Enter Number");
    if (num > 0)
    {
        alert("Given number is Positive!!!");
    }
</script>
</body>
</html>
```

Output:





ii) If – Else Statement:

If – Else is a two-way decision statement. It is used to make decisions and execute statements conditionally.

Syntax:

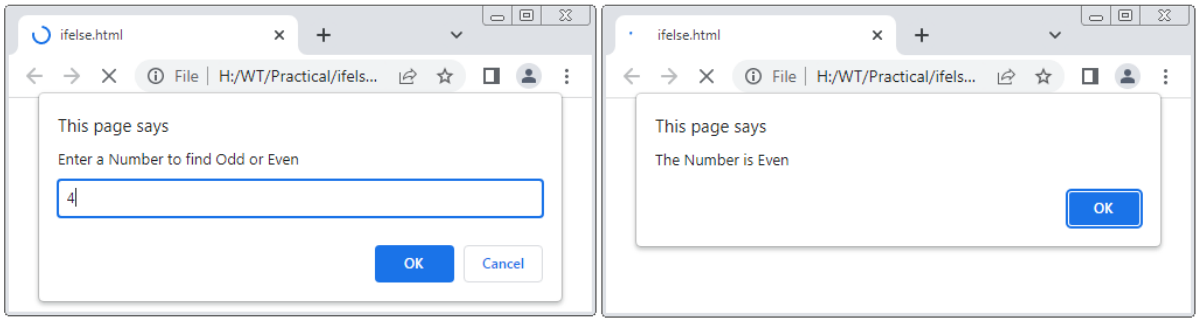
```
if (condition)
{
    //Statement 1;
}
else if(condition)
{
    //Statement 2;
```

```
    }
    else
    {
        //Statement 3;
    }
```

Example:

```
<html>
<head>
    <script type="text/javascript">
        var no = prompt("Enter a Number to find Odd or Even");
        no = parseInt(no);
        if (isNaN(no))
        {
            alert("Please Enter a Number");
        }
        else if (no == 0)
        {
            alert("The Number is Zero");
        }
        else if (no % 2)
        {
            alert("The Number is Odd");
        }
        else
        {
            alert("The Number is Even");
        }
    </script>
</head>
</html>
```

Output:



iii) Switch Statement:

The switch statement evaluates an expression. The value of the expression is then compared with the values of each case in the structure. If there is a match, the associated block of code is executed. Switch is multi way decision statement.

Syntax:

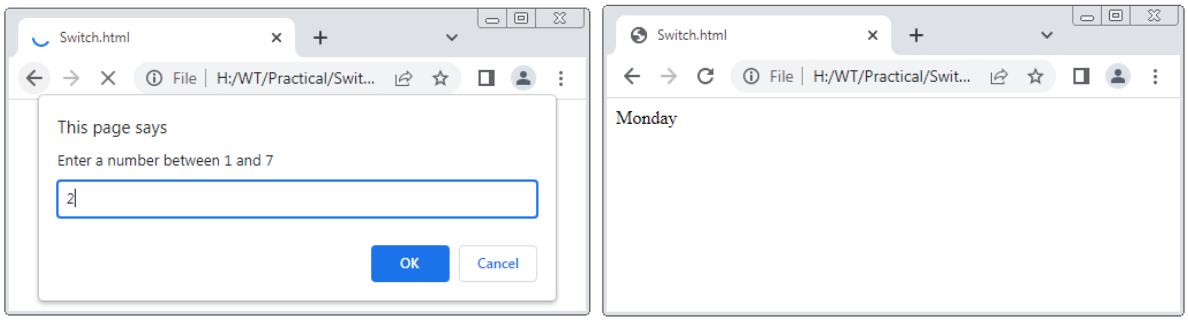
```
switch(expression)
{
    case condition 1:
        //Statements;
        break;
    case condition 2:
        //Statements;
        break;
    case condition 3:
        //Statements;
        break;
    …….
    …….
    case condition n:
        //Statements;
        break;
    default:
        //Statement;
}
```

Example:

```
<html>
<head>
<script type="text/javascript">
    var day = prompt("Enter a number between 1 and 7");
    switch (day)
    {
        case (day="1"):
            document.write("Sunday");
            break;
        case (day="2"):
            document.write("Monday");
            break;
        case (day="3"):
            document.write("Tuesday");
            break;
        case (day="4"):
            document.write("Wednesday");
            break;
        case (day="5"):
            document.write("Thursday");
            break;
        case (day="6"):
```

```
            document.write("Friday");
            break;
        case (day="7"):
            document.write("Saturday");
            break;
        default:
            document.write("Invalid Weekday");
            break;
    }
</script>
</head>
</html>
```

Output:



2) Iterative Statements (Loop): Sometimes it is the requirement of programming to repeat the code until certain condition is true this is called as a loop and traversing through a loop is called as iteration. There are three Iterative statements:

i) while

ii) do-while

iii) for

i) While loop:

While loop is entry controlled loop. In while loop first condition is checked if it is it true then body of loop is executed. If condition is false then body of the loop will be not executed.

Syntax:

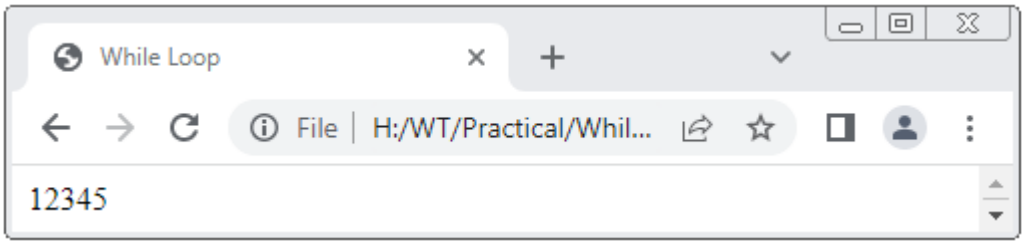
```
initialization;
while (condition)
{
        statements;
        increment/decrement;
}
```

Example:

```
<html>
<head>
    <title>While Loop</title>
```

```
</head>
<body>
<script type="text/javascript">
    var i = 1;
    while( i <= 5)
    {
            document.write(i);
             i++;
    }
</script>
</body>
</html>
```

Output:



ii) Do-while Loop:

Do while is exit controlled loop. Do while loop is similar to while loop but the do while loop first execute and then check the condition. If condition is true then it again executes until condition is true and if the condition is false then do while loop is not execute.

Syntax:

```
initialization;
do
{
        statements;
        increment/decrement;
}while (condition);
```

Example:

```
<html>
<head>
    <title>Do While Demo</title>
</head>
<body>
<script type="text/javascript">
    var i = 1;
```

```
    do
    {
            document.write(i);
             i++;
    }while (i <= 5);
</script>
</body>
</html>
```

Output:

iii) For loop:

For loop is entry controlled loop.

It includes three important parts:

1. Loop Initialization
2. Test Condition
3. Iteration

All these three parts come in a single line separated by semicolons(;).

Syntax:

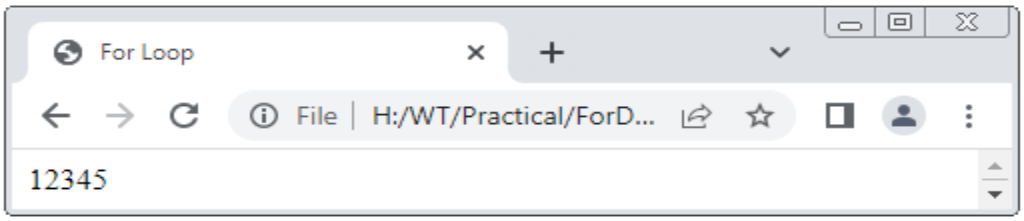
```
for(initialization;condition;increment/decrement)
{
    statements;
}
```

Example:

```
<html>
<head>
    <title>For Loop</title>
</head>
<body>
<script type="text/javascript">
    var i;
    for(i=1;i<=5;i++)
    {
            document.write(i);
```

```
    }
</script>
</body>
</html>
```

Output:



3) Jumping statements:

Jumping statements are control statements that transfer execution control from one point to another point in the program. There are two Jump statements.

i) Break statement

ii) Continue statement.

i) Break statement:

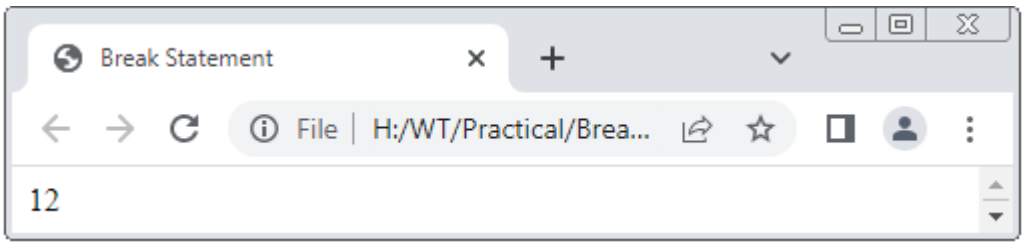
Break statement is used to terminate a loop. When a break statement is encounters in the loop then loop is terminated and control is transferred to the statement which appears immediately after the loop.

Syntax: break;

Example:

```
<html>
<head>
    <title>Break Statement</title>
</head>
<body>
<script type="text/javascript">
    var i;
    for(i=1;i<=5;i++)
    {
        if(i==3)
        {
            break;
        }
        document.write(i);
    }
</script>
</body>
</html>
```

Output:



ii) Continue Statement:

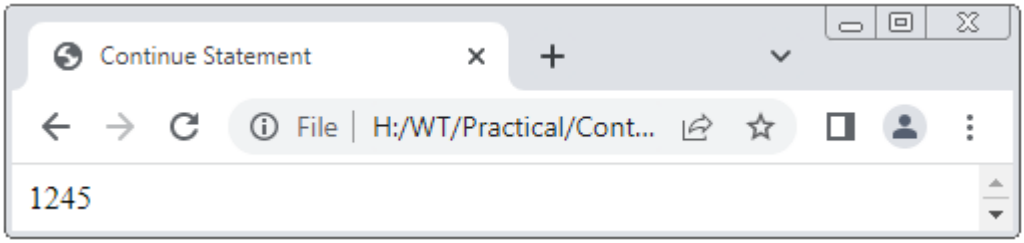
When continue statement encounters in the loop then remaining statement in the loop are ignored and control is transfer back to condition of loop.

Syntax: continue;

Example:

```
<html>
<head>
    <title>Continue Statement</title>
</head>
<body>
<script type="text/javascript">
    var i;
    for(i=1;i<=5;i++)
    {
            if(i==3)
            {
                    continue;
            }
            document.write(i);
    }
</script>
</body>
</html>
```

Output:



❖ JavaScript events and event handling:

Event:

An event is an action that takes place when a user interacts with a program. For example, an event can be a mouse click or pressing a key on the keyboard. When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Event handler: Event handlers can be used to handle and verify user input, user actions, and browser actions. JavaScript has number of Event Handlers.

JavaScript event handlers, can be divided into 2 types:

1) **Interactive:** Depends on user interaction with HTML page. Ex: onClick, onMouseOver…etc.
2) **Non-Interactive:** Doesn't need user interaction to be invoked Ex: onLoad, onUnload… etc

Following are the list of events in JavaScript.

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	onkeydown	When the user press the key
Keyup	onkeyup	When the user 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

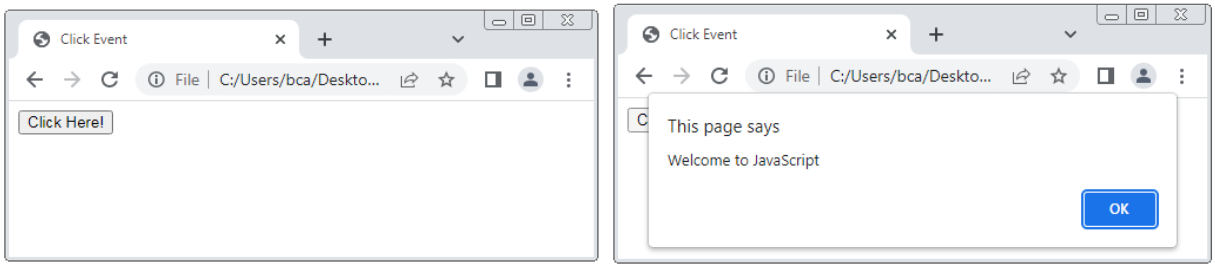
Window/Document events:

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

❖ JavaScript Events Examples:

Click Event Example:

```
<html>
<head>
    <title>Click Event</title>
</head>
<body>
<script language="Javascript" type="text/Javascript">
    function clickevent()
    {
            alert("Welcome to JavaScript");
    }
</script>
<form>
        <input type="button" onclick="clickevent()" value="Click Here! "/>
</form>
</body>
</html>
```



❖ JavaScript objects:

A javaScript object is an entity having properties and method. JavaScript is an object-based language. Everything is an object in JavaScript. JavaScript also has four built-in objects: Array, Date, Math, and String. Each object has special-purpose properties and methods associated with it.

1) Date object:

The Date object works with dates and times. We can create date object using the new Date() keyword.

Syntax: var variable_name=new Date();

Example: var d=new Date();

Date Object Methods

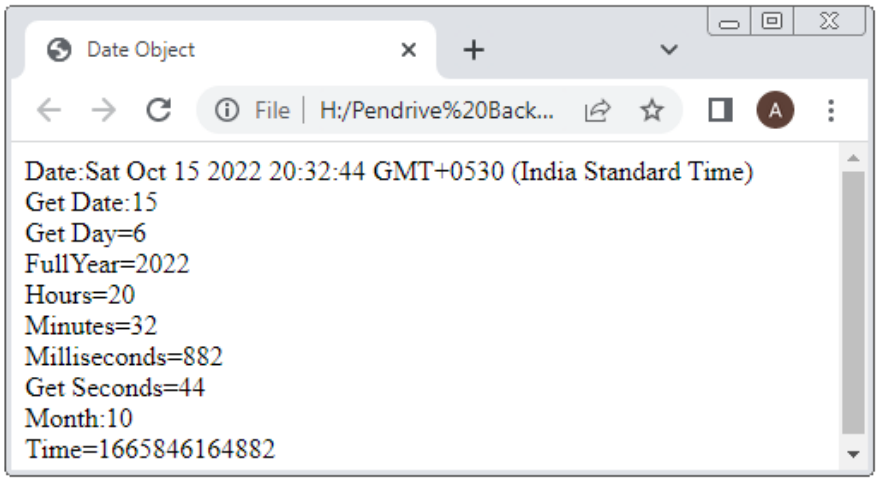
Sr. No.	Method	Description
1	getDate()	Returns the day of the month (from 1-31).
2	getDay()	Returns the day of the week (from 0-6).
3	getFullYear()	Returns the year (four digits).
4	getHours()	Returns the hour (from 0-23).
5	getMilliseconds()	Returns the milliseconds (from 0-999).
6	getMinutes()	Returns the minutes (from 0-59).
7	getMonth()	Returns the month (from 0-11).
8	getSeconds()	Returns the seconds (from 0-59).
9	getTime()	Returns the number of milliseconds since midnight Jan 1, 1970.
10	setDate()	Sets the day of the month from 1 to 31
11	setFullYear()	Sets the full year of a date object.
12	setHours()	Sets the hours from 0 to 23
13	setMonth()	Sets the month from 0 to 11
14	setMilliseconds()	Set the milliseconds (0-999)
15	setMinutes()	Set the minutes (0-59)
16	setSeconds()	Set the seconds (0-59)
17	setTime()	Set the time (milliseconds since January 1, 1970)

Example:

```
<html>
<head>
        <title>Date Object</title>
</head>
<body>
<script type="text/javascript">
        var d =new Date();
        document.write("Date:"+d);
        document.write("<br>");
        document.write("Get Date:"+d.getDate()+"<br>");
        document.write("Get Day="+d.getDay()+"<br>");
        document.write("Month:"+(d.getMonth()+1)+"<br>");
        document.write("FullYear="+d.getFullYear()+"<br>");
        document.write("Hours="+d.getHours()+"<br>");
```

```
        document.write("Minutes="+d.getMinutes()+"<br>");
        document.write("Get Seconds="+d.getSeconds()+"<br>");
        document.write("Milliseconds="+d.getMilliseconds()+"<br>");
        document.write("Time="+d.getTime()+"<br>");
</script>
</body>
</html>
```

Output:

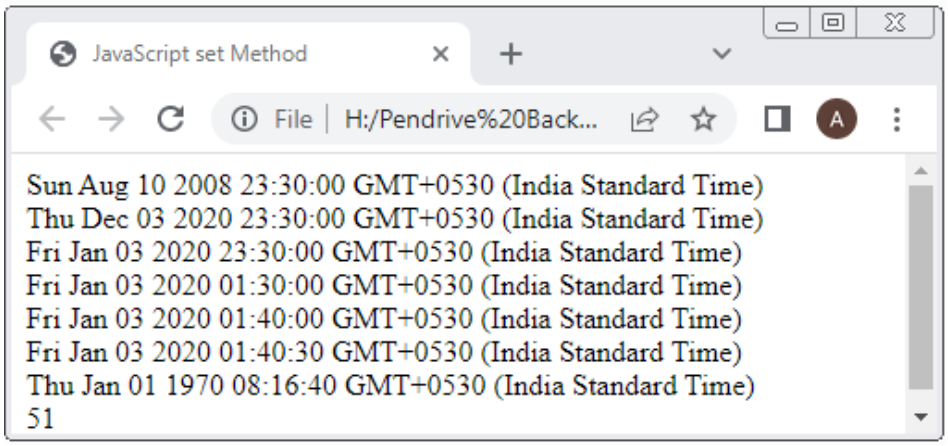


Example:

```
<html>
<head>
        <title>JavaScript set Method</title>
</head>
<body>
<script type = "text/javascript">
        var dt = new Date( 'Aug 28, 2008 23:30:00');
        dt.setDate(10)
        document.write(dt+"<br>");
        dt.setFullYear(2020, 11, 3);
        document.write(dt+"<br>");
        dt.setMonth(0);
        document.write(dt+"<br>");
        dt.setHours(1);
        document.write(dt+"<br>");
        dt.setMinutes(40);
        document.write(dt+"<br>");
        dt.setSeconds(30);
        document.write(dt+"<br>");
        dt.setTime(10000000);
        document.write(dt+"<br>");
        var dateobj = new Date('October 13, 1996');
        // New millisecond of 51 is being set in above Date
```

```
        dateobj.setMilliseconds(51);
        var B = dateobj.getMilliseconds();
        // Printing new millisecond
        document.write(B);
</script>
</body>
</html>
```

Output:



2) Math Object: The JavaScript math object is used to perform mathematical task or calculation. Math object is built in JavaScript object.

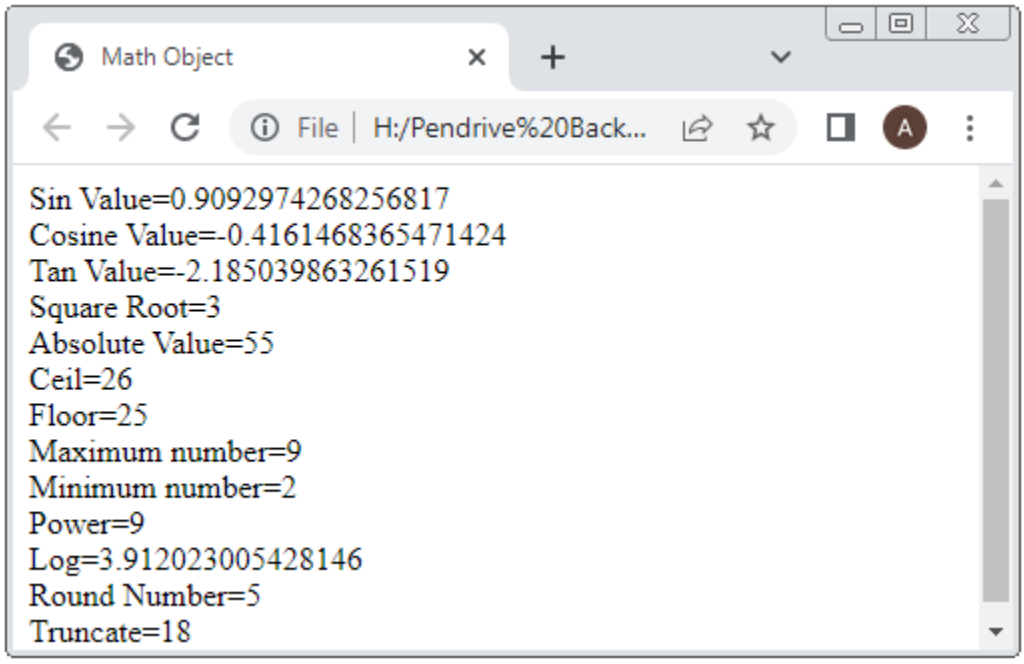
Sr. No.	Method	Description
1	sin()	It returns the sin of number.
2	cos()	It returns the cos of number.
3	tan()	It returns the tangent of number.
4	sqrt()	It returns the square root of number.
5	abs()	It returns the absolute value of a number.
6	ceil()	Returns the next integer greater than or equal to a given number (rounding up).
7	floor()	Returns the next integer less than or equal to a given number (rounding down).
8	max(x, y, ..)	Returns the maximum number in a list of numbers.
9	min(x, y, ..)	Returns the minimum number in a list of numbers.
10	pow(x,y)	It returns x^y
11	log(x)	It returns natural logarithm of x.
12	round()	It returns the value of a number rounded to the nearest integer.

13	trunc(x)	It returns the integer part of a number by removing any fractional digits.

Example:

```
<html>
<head>
        <title>Math Object</title>
</head>
<body>
<script type="text/javascript">
        document.write("Sin Value="+Math.sin(2)+"<br>");
        document.write("Cosine Value="+Math.cos(2)+"<br>");
        document.write("Tan Value="+Math.tan(2)+"<br>");
        document.write("Square Root="+Math.sqrt(9)+"<br>");
        document.write("Absolute Value="+Math.abs(-55)+"<br>");
        document.write("Ceil="+ Math.ceil(25.6)+"<br>");
        document.write("Floor="+ Math.floor(25.6)+"<br>");
        document.write("Maximum number="+Math.max(3,4,2,7,5,9)+"<br>");
        document.write("Minimum number="+Math.min(3,4,2,7,5,9)+"<br>");
        document.write("Power="+Math.pow(3,2)+"<br>");
        document.write("Log="+Math.log(50)+"<br>");
        document.write("Round Number="+Math.round(4.6)+"<br>");
        document.write("Truncate="+Math.trunc(18.23)+"<br>");
        </script>
</body>
</html>
```

Output:



3) String Object: String is a collection or sequence of letters, numbers, special characters or combination of all. We can create a string object by using new keyword.

Syntax: var variablename=new String("text");

var s=new String("welcome");

String object properties:

Length- It calculate the number of characters in the string.

Example: document.write(s.length);

Methods of String object:

1) toUpperCase(): It converts string to Upper case.

Example: **v**ar s="programming";

document.write(s.toUpperCase());

2) toLowerCase(): It converts string to lower case.

Example: var s="PROGRAMMING";

document.write(s.toLowerCase());

3) charAt(): It returns the character at the specified location.

Syntax: charAt(position);

Example: var s="PROGRAMMING";

document.write(s.charAt(2));

Output: O

4) substring(): It returns the part of string from specified position.

Syntax: Substring(start,end);

Example: var s="PROGRAMMING";

document.write(s.substring(1,5));

Output: ROGRA

5) concat(): It combines two string into one string.

Syntax: string1.concat(string2)

Example: var s="java";

var p="script";

document.write(s.concat(p));

Output: javascript

6) **indexOf(): It finds the string in another string and returns the position of the first occurrence of that string.**

Syntax: indexOf(String,Start)

Where,

String: It is substring that to be search within the string.

Start: It is optional argument.it specifies the position from which search is start.

Example:
```
var s= “learn javascript”;
document.write(s.indexOf(“a”));
```

Output: 2

Example:
```
var s= “learn javascript”;
document.write(s.indexOf(“a”,3);
```

Output: 7

7) **lastIndexOf(): It finds the string in another string and returns the position of the last occurrence of that string.**

Syntax: lastIndexOf(string,start)

Example:
```
var s= “learn javascript”;
document.write(s.lastIndexOf(“a”);
```

Output: 9

8) **replace(): It is used to replace a part of the given string with some another string or a regular expression.**

Syntax: replace(searchValue, newValue)

Where,

searchValue : The value, or regular expression, to search for.

newValue: The new value (to replace with).

Example:
```
var str = "JS will, JS will rock you";
document.write(str.replace("JS","JavaScript"));
```

As you can see from the output, only the first occurrence of the substring JS was replaced with the new substring JavaScript.

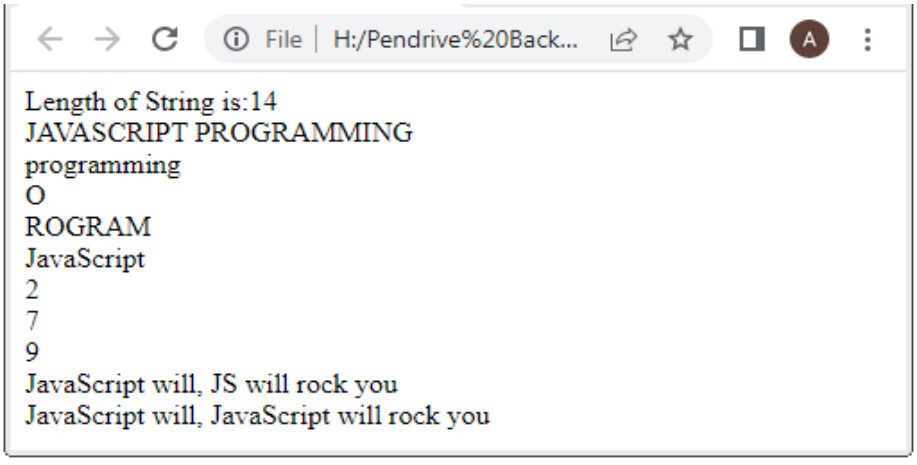
Example:
```
varstr = "JS will, JS will rock you!";
document.write(str.replace(/JS/g,"JavaScript"));
```

The following example uses the global flag (g) to replace all occurrences of the JS in the str by the JavaScript:

❖ String Object Example:

```
<html>
<head>
        <title>JavaScript String Methods</title>
</head>
<body>
<script language="javascript">
        var s=new String("welcome BCA-II");
        document.write("Length of String is:"+s.length+"<br>");
        var s="JavaScript Programming";
        document.write(s.toUpperCase()+"<br>");
        var s="PROGRAMMING";
        document.write(s.toLowerCase()+"<br>");
        var s="PROGRAMMING";
        document.write(s.charAt(2)+"<br>");
        var s="PROGRAMMING";
        document.write(s.substring(1,7)+"<br>");
        var s="Java";
        var p="Script";
        document.write(s.concat(p)+"<br>");
        var s= "learn javascript";
        document.write(s.indexOf("a")+"<br>");
        var s= "learn javascript";
        document.write(s.indexOf("a",3)+"<br>");
        var s= "learn javascript";
        document.write(s.lastIndexOf("a")+"<br>");
        var str = "JS will, JS will rock you";
        document.write(str.replace("JS","JavaScript")+"<br>");
        varstr = "JS will, JS will rock you!";
        document.write(str.replace(/JS/g,"JavaScript"));
</script>
</body>
</html>
```

Output:



❖ JavaScript validations:

JavaScript provides facility to validate the form on the client-side so data processing will be faster than server-side validation. Most of the web developers prefer JavaScript form validation. Through JavaScript, we can validate name, password, email, date, mobile numbers and more fields.

Form validation generally performs two functions.

- **Basic Validation** − First of all, the form must be checked to make sure all the mandatory fields are filled in.
- **Data Format Validation** − Secondly, the data that is entered must be checked for correct form and value. Your code must include appropriate logic to test correctness of data.

Basic Form Validation Example:

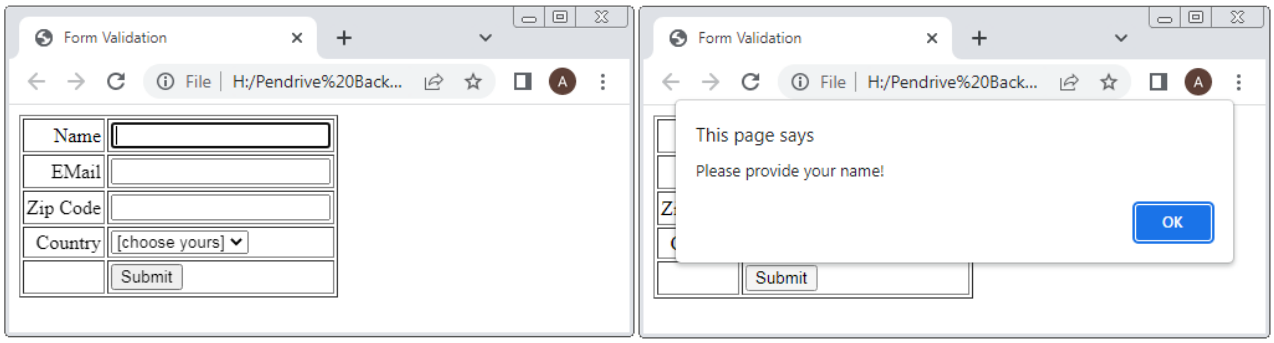
First let us see how to do a basic form validation. In the below form, we are calling validate() to validate data when onsubmit event is occurring.

```
<html>
<head>
<title>Form Validation</title>
<script type = "text/javascript">
    <!--
    // Form validation code will come here.
    function validate()
    {
        if( document.myForm.Name.value == "" )
        {
            alert( "Please provide your name!" );
            document.myForm.Name.focus() ;
            return false;
        }
        if( document.myForm.EMail.value == "" )
        {
            alert( "Please provide your Email!" );
            document.myForm.EMail.focus() ;
            return false;
        }
        if( document.myForm.Zip.value == "" || isNaN( document.myForm.Zip.value ) ||
document.myForm.Zip.value.length != 5 )
        {
            alert( "Please provide a zip in the format #####." );
            document.myForm.Zip.focus() ;
            return false;
```

```
            }
            if( document.myForm.Country.value == "-1" )
            {
                  alert( "Please provide your country!" );
                  return false;
            }
            return( true );
      }
//-->
</script>
</head>
<body>
<form name = "myForm" onsubmit = "return(validate());">
      <table cellspacing = "2" cellpadding = "2" border = "1">
        <tr>
          <td align = "right">Name</td>
          <td><input type = "text" name = "Name" /></td>
        </tr>
        <tr>
          <td align = "right">EMail</td>
          <td><input type = "text" name = "EMail" /></td>
        </tr>
        <tr>
          <td align = "right">Zip Code</td>
          <td><input type = "text" name = "Zip" /></td>
        </tr>
        <tr>
          <td align = "right">Country</td>
          <td>
            <select name = "Country">
              <option value = "-1" selected>[choose yours]</option>
              <option value = "1">USA</option>
              <option value = "2">UK</option>
              <option value = "3">INDIA</option>
            </select>
          </td>
        </tr>
        <tr>
          <td align = "right"></td>
          <td><input type = "submit" value = "Submit" /></td>
        </tr>
      </table>
</form>
</body>
```

</html>

Output:



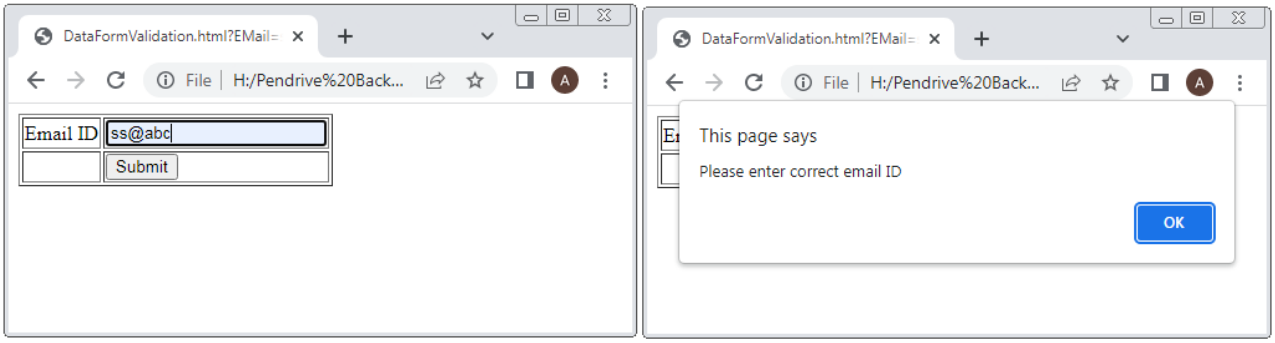
Data Format Validation Example:

Now we will see how we can validate our entered form data before submitting it to the web server. The following example shows how to validate an entered email address. An email address must contain at least a '@' sign and a dot (.). Also, the '@' must not be the first character of the email address, and the last dot must at least be one character after the '@' sign.

```
<html>
<head>
<script type = "text/javascript">
  <!--
    function validateEmail()
   {
      var emailID = document.myForm.EMail.value;
      atpos = emailID.indexOf("@");
      dotpos = emailID.lastIndexOf(".");
      if (atpos < 1 || ( dotpos - atpos < 2 ))
      {
        alert("Please enter correct email ID")
        document.myForm.EMail.focus() ;
        return false;
       }
      return( true );
     }
  //-->
</script>
</head>
<body>
<form name="myForm" onsubmit="validateEmail()">
<table border="1">
<tr>
      <td>Email ID</td>
      <td> <input type="text" name="EMail"> </td>
</tr>
```

```
<tr>
    <td></td>
    <td><input type="submit" value="Submit"> </td>
</tr>
</table>
</form>
</body>
</html>
```

Output:



UNIT-IV HTML 5

❖ Introduction to HTML5:

- HTML stands for Hyper Text Markup Language. It is used to design web pages.
- HTML 5 is the fifth and current version of HTML.
- Html-5 is the next version of Html-4, this is completed in October 2014.
- HTML5 introduces a number of new elements and attributes that can help you in building modern websites.
- Two major organizations W3C (World Wide Web Consortium) and WHATWG (Web Hypertext Application Technology Working Group) have developed HTML5.
- HTML5 supports all the latest browsers.
- Html5 is more power full and easier than Html4.
- It allows to play audio and video on browser
- Using Html5 you can draw shapes like circle, rectangle, triangle etc. Which is not possible in Html previous version.
- In Html5 direct you can use header and footer tag for define Header and footer section

❖ Difference Between HTML AND HTML5:

S.N.	HTML	HTML5
1	Hypertext Markup Language is a primary language for developing web pages.	HTML5 is the new version of HTML, which has new functionalities with Markup language
2	HTML doesn't support for video and audio in the language.	Support Audio and Video with use of <audio> and <video> tag.
3	HTML is less mobile friendly	HTML5 is more mobile friendly
4	Shapes like circle, rectangle, triangle, etc. are not possible to draw in HTML	Shapes like circle, rectangle, triangle, etc. are easy to draw in HTML5.
5	As it's older version, it is not fast, efficient, and flexible with respect to HTML5.	HTML5 is efficient, faster and flexible in comparison to HTML.
6	It works with all old browsers.	It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc.
7	For storage browser cache can be used as temporary storage	It uses SQL databases and application cache to store offline data.

❖ HTML5 Attribute:

Elements (tags) may contain attributes that are used to set various properties of an element. All attributes have a name and a value. Attributes may only be specified within **start tags** and must never be used in **end**

tags. HTML5 attributes are case insensitive and may be written in all uppercase or mixed case, and also lowercase.

Standard Attribute:

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

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
contenteditable	true, false	Specifies whether the content of an element is editable by the user or not.
draggable	true, false	Specifies whether an element is draggable or not.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.
id	name	Specifies a unique identifier (ID) for an element which must be unique in the whole document.
spellcheck	true, false	Specifies whether the element may be checked for spelling errors or not.
style	CSS Style sheet	Specifies an inline style for an element.
tabindex	number	Specifies the tabbing order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

❖ Examples:

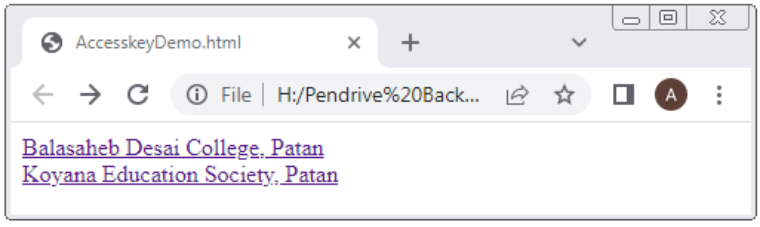
i) Accesskey: The accesskey attribute specifies a shortcut key to activate/focus an element

```
<html>
<body>
<a href="https://www.bdcp.edu.in" accesskey="b">Balasaheb Desai College, Patan</a><br>
<a href="https://www.koyanaeducation.org" accesskey="k">Koyana Education Society, Patan</a>
</body>
</html>
```

Note: The shortcut is varying in different browsers:

- Edge, IE, Chrome, Safari, Opera 15+: [ALT] + accesskey
- Opera prior version 15: [SHIFT] [ESC] + accesskey
- Firefox: [ALT] [SHIFT] + accesskey

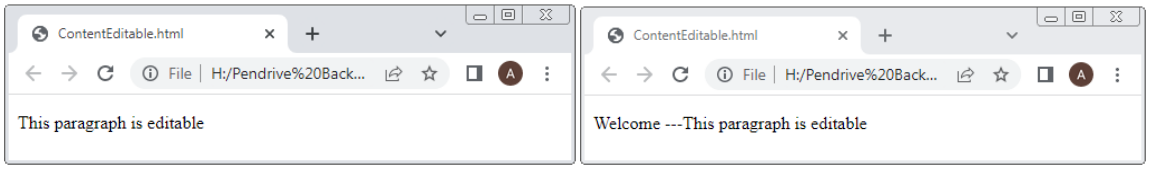
Output:



ii) **Contenteditable:** Contenteditable is a new attribute for text elements. Default value of contenteditable is false. To edit a paragraph or text, add contenteditable="true".

```
<html>
<body>
<p contenteditable="true">This paragraph is editable</p>
</body>
</html>
```

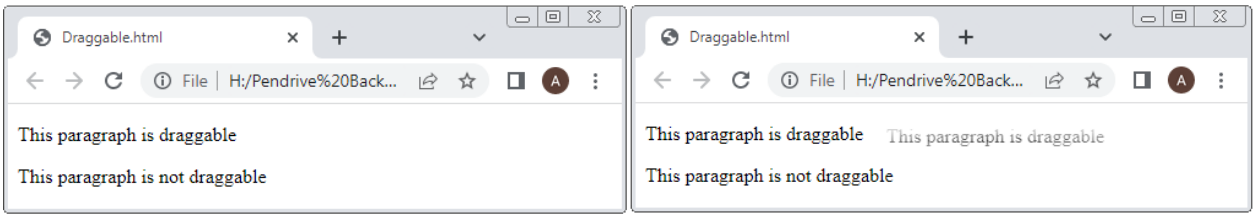
Output:



iii) **Draggable:** draggable attribute is used if an element is draggable using HTML5 drag and Drop Events. By default, only selected text, images and hyperlinks are draggable, but using draggable="true" any html5 element can be draggable.

```
<html>
<body>
<p draggable="true">This paragraph is draggable</p>
<p draggable="false">This paragraph is not draggable</p>
</body>
</html>
```

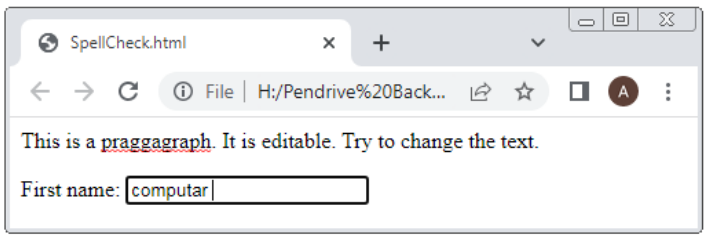
Output:



iv) **Spellcheck:** Specifies whether the element may be checked for spelling errors or not.

```
<html>
<body>
<p contenteditable="true" spellcheck="true">This is a praggagraph. It is editable. Try to change the text.</p>
First name: <input type="text" name="fname" spellcheck="true">
</body>
</html>
```

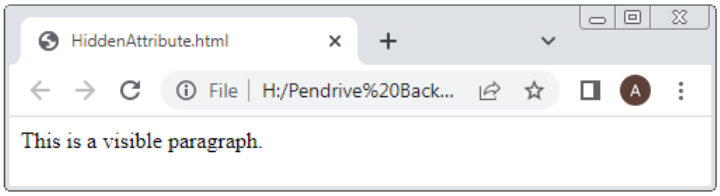
Output:



v) **Hidden Attribute:** Specifies whether element should be visible or not.

```
<html>
<body>
<p hidden>This paragraph should be hidden.</p>
<p>This is a visible paragraph.</p>
</body>
</html>
```

Output:



❖ HTML5 Event:

Event:- Events are action that are trigger when user does something like clicking of button or moving mouse over hyperlink or pressing of key on the keyboard.

Event handler: It is a script written in JavaScript that handles event.

Window Event Attributes:

Events triggered for the window object (applies to the <body> tag):

Attribute	Value	Description
onafterprint	*script*	Script to be run after the document is printed
onbeforeprint	*script*	Script to be run before the document is printed
onerror	*script*	Script to be run when an error occurs
onload	*script*	Fires after the page is finished loading
onresize	*script*	Fires when the browser window is resized
onunload	*script*	Fires once a page has unloaded (or the browser window has been closed)

Form Events: Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

Attribute	Value	Description
onchange	*script*	Fires the moment when the value of the element is changed
onfocus	*script*	Fires the moment when the element gets focus
oninput	*script*	Script to be run when an element gets user input
onreset	*script*	Fires when the Reset button in a form is clicked
onselect	*script*	Fires after some text has been selected in an element
onsubmit	*script*	Fires when a form is submitted

Mouse Events: Events that occur due to the user interacting with a pointing device such as a mouse:

Attribute	Value	Description
onclick	*script*	Fires on a mouse click on the element
ondblclick	*script*	Fires on a mouse double-click on the element
onmousedown	*script*	Fires when a mouse button is pressed down on an element
onmousemove	*script*	Fires when the mouse pointer is moving while it is over an element
onmouseover	*script*	Fires when the mouse pointer moves over an element
onmouseup	*script*	Fires when a mouse button is released over an element
onmousewheel	*script*	Event fire when mouse wheel being rotated

Keyboard Events: Events that occur by the user interaction with the keyboard:

Attribute	Value	Description
onkeydown	*script*	Fires when the user presses a key.
onkeypress	*script*	Fires when the user presses an alphanumeric key.
onkeyup	*script*	Fires when the user releases a key.

Drag Events:

Attribute	Value	Description
ondrag	*script*	Script to be run when an element is dragged
ondragend	*script*	Script to be run at the end of a drag operation
ondragleave	*script*	Script to be run when an element leaves a valid drop target
ondragstart	*script*	Script to be run at the start of a drag operation

❖ HTML5 Audio & Video:

The HTML5 <audio> and <video> tags make it simple to add media to a website

i) **<Audio>tag: <audio>** tag is used to insert **audio** in web page. **src** is compulsory attribute in audio tag. controls attribute can show control bar to user. User can play/pause, change time line, mute, and increase volume of audio playing using controls.

Attribute	Use
src	Name of audio file which you want to play.
controls	This attribute adds audio controls, like play, pause, and volume.
autoplay	To play audio automatically. Possible values are on or off
loop	To play audio continuously even after it ends.

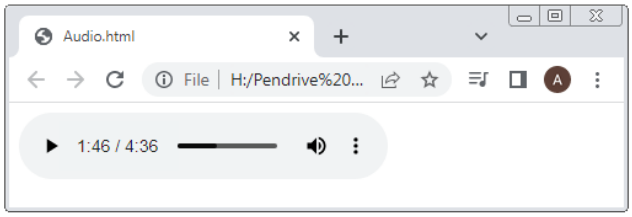
Example:

```
<audio src="sound.mp3" controls></audio>
```

Example:

```
<html>
<body>
<audio src="All is Well.mp3" controls></audio>
</body>
</html>
```

Output:



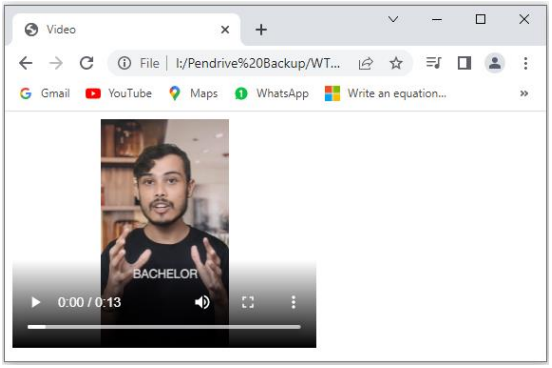
ii) <Video> Tag: The HTML <video> element is used to show a video on a web page.

Attribute	Use
src	Name of video file which you want to play.
controls	This attribute adds video controls, like play, pause, and volume.
autoplay	To play video automatically. Possible values are on or off.
loop	To play video continuously even after it ends.
height	It specifies the height of video player.
width	It specifies the width of video player.

Example:

```
<html>
<head>
    <title>Video</title>
</head>
<body>
<video width="320" height="240" controls>
    <source src="Video.MP4" type="video/mp4">
</video>
</body>
</html>
```

Output:



❖ HTML5 Web Forms 2.0:

- An HTML form is a special part of a web page that contains certain controls, including labels, text fields, password fields, hidden fields (used by the software), radio buttons, checkboxes, fieldsets and submit buttons.
- HTML5 form elements are supported by many browsers.

<input> elements in HTML5:

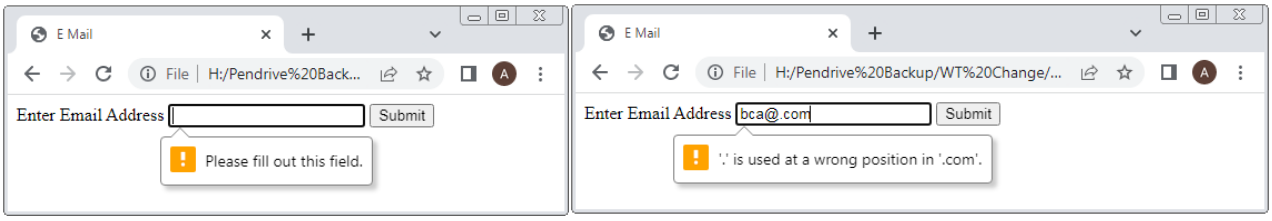
Some new type attributes that are added to the <input> element are as follows:

Type	Description
date	Date i.e. year, month, and day.
month	A date consist of a year and a month.
week	A date consist of a year and a week number.
time	It is time i.e. hour, minute, seconds.
number	It accepts only a numerical value.
range	It is used for input fields that contain a value from a range of numbers.
email	It accepts only the email value. It is used for input fields that contain an e-mail address.
url	It accepts only the URL value. It is used for input fields that contain a URL address.

Examples

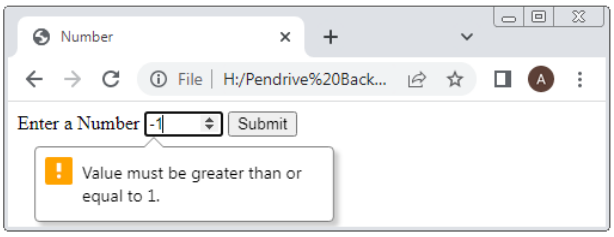
i) Email- It accepts only the email value. It is used for input fields that contain an e-mail address.

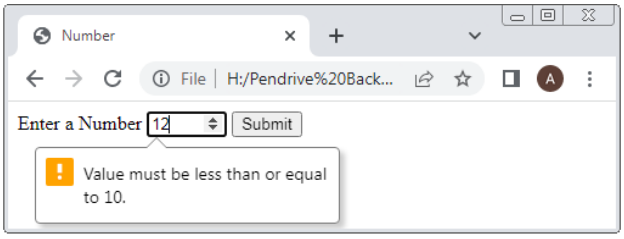
```
<html>
<head>
    <title>E Mail</title>
</head>
<body>
<form>
    Enter Email Address <input type="email" id="myemail" required>
    <input type="submit">
</form>
</body>
</html>
```



ii) Number- It accepts only a numerical value.

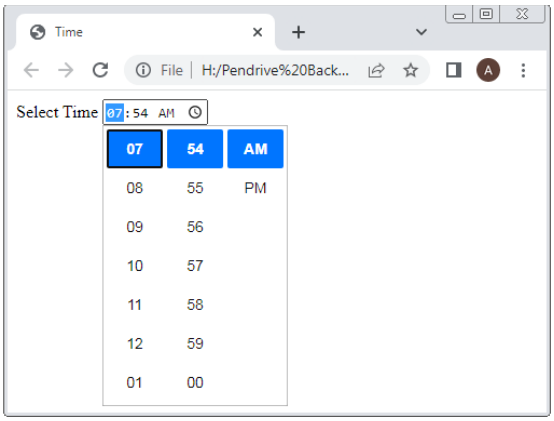
```
<html>
<head>
   <title>Number</title>
</head>
<body>
<form>
   Enter a Number <input type="number" min="1" max="10" id="mynumber">
   <input type="submit">
</form>
</body>
</html>
```





iii) Time- It is time i.e. hour, minute, seconds, fractional seconds encoded according to ISO 8601.

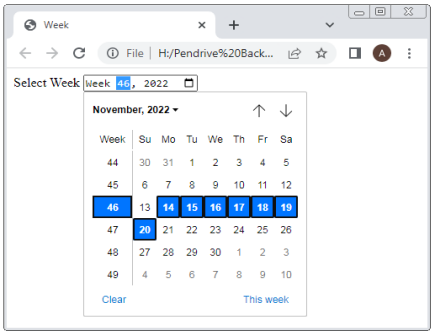
```
<html>
<head>
   <title>Time</title>
</head>
<body>
<form>
   Select Time <input type="time" id="mytime">
</form>
</body>
</html>
```



iv)Week: A date consist of a year and a week number is encoded according to ISO 8601.

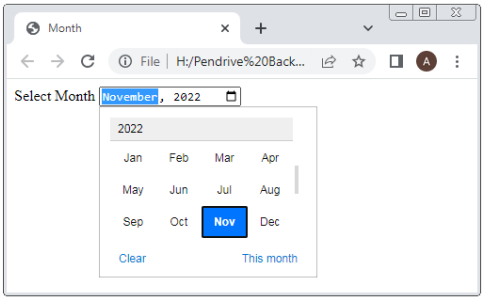
```
<html>
<head>
   <title>Week</title>
</head>
<body>
```

```
<form>
   Select Week <input type="week" id="myweek">
</form>
</body>
</html>
```



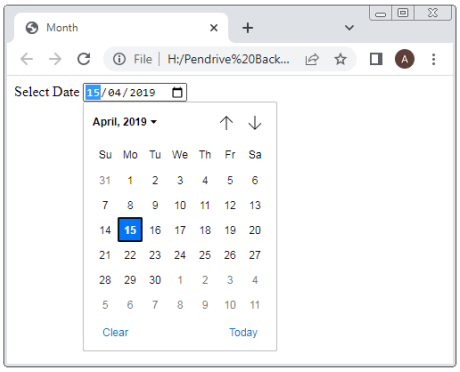
v) Month: A date consist of a year and a month encoded according to ISO 8601.

```
<html>
<head>
   <title>Month</title>
</head>
<body>
<form>
   Select Month <input type="month" id="mymonth">
</form>
</body>
</html>
```



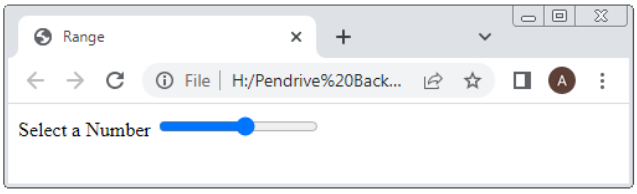
vi) Date- Date i.e. year, month, and day is encoded according to ISO 8601.

```
<html>
<head>
   <title>Month</title>
</head>
<body>
<form>
   Select Date <input type="date" value="2019-04-15" id="mydate">
</form>
</body>
</html>
```



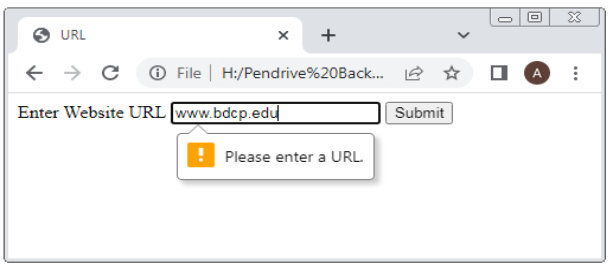
vii) Range: It is used for input fields that contain a value from a range of numbers.

```
<html>
<head>
   <title>Range</title>
</head>
<body>
<form>
   Select a Number <input type="range" min="1" max="10" step="0.5" id="mynumber">
</form>
</body>
</html>
```



viii) URL: It accepts only the URL value. It is used for input fields that contain a URL address.

```
<html>
<head>
   <title>URL</title>
</head>
<body>
<form>
   Enter Website URL <input type="url" id="myurl" required>
   <input type="submit">
</form>
</body>
</html>
```



❖ HTML5 canvas:

The HTML 5 <canvas> tag is used to draw graphics using scripting language like JavaScript. Canvas has several methods for drawing boxes, circles, text, and adding images.

A canvas is a rectangular area on an HTML page. By default, a canvas has no border and no content.

Syntax: <canvas id="myCanvas" width="200" height="100"></canvas>

Example:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid  black;">
</canvas>
```

Drawing Line on Canvas:

If you want to draw a straight line on the canvas, you can use the following two methods.

moveTo(x,y): It is used to define the starting point of the line.

lineTo(x,y): It is used to define the ending point of the line.

Stroke(): It is used to draw the line.

Example:

```
<!DOCTYPE html>
<html>
<head>
    <title>Canvas Line Draw</title>
</head>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;">
Your browser does not support the HTML canvas tag.</canvas>
<script>
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.moveTo(0,0);
    ctx.lineTo(200,100);
    ctx.stroke();
</script>
</body>
</html>
```

Output:

Drawing Circle on Canvas:

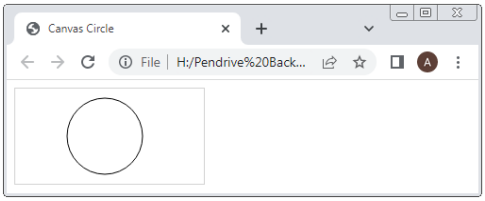
If you want to draw a circle on the canvas, you can use the arc() method:

Syntax: arc(x, y, r, start, stop);

Example:

```
<html>
<head>
    <title>Canvas Circle</title>
</head>
<body>
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #d3d3d3;"></canvas>
<script>
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.beginPath();
    ctx.arc(95,50,40,0,10);
    ctx.stroke();
</script>
</body>
</html>
```

Output:



Drawing a Rectangle: You can create rectangle and square shapes using the rect() method. This method requires four parameters x, y position of the rectangle and its width and height.

Syntax: context.rect(x, y, width, height);

Example:

```
<html>
<head>
    <title>Canvas Rectangle</title>
</head>
<body>
<canvas id="myCanvas" width="300" height="200" style="border:1px solid #d3d3d3;"> </canvas>
<script>
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.rect(50,50,200,100);
    ctx.stroke();
</script>
</body>
</html>
```

Output: