* **WEB TECHNOLOGY NOTES :-**

**UNIT :-1**

**INTRODUCTION:- Web Technology refers to the various tools and techniques that are utilized in the process of communication between different types of devices over the internet. A web browser is used to access web pages. Web browsers can be defined as programs that display text, data, pictures, animation, and video on the Internet.**

* **History and evolution of internet protocols:-**

**IP was the connectionless datagram service in the original Transmission Control Program introduced by Vint Cerf and Bob Kahn in 1974**, which was complemented by a connection-oriented service that became the basis for the Transmission Control Protocol (TCP).

* **IP ADDRESS :-** All the computers of the world on the Internet network communicate with each other with underground or underwater cables or wirelessly. If I want to download a file from the internet or load a web page or literally do anything related to the internet, my computer must have an address so that other computers can find and locate mine in order to deliver that particular file or webpage that I am requesting. In technical terms, that address is called **IP Address or Internet Protocol Address.**

**Working of IP addresses:-**

The working of IP addresses is similar to other languages. It can also use some set of rules to send information. Using these protocols we can easily send, and receive data or files to the connected devices. There are several steps behind the scenes. Let us look at them

* Your device directly requests your Internet Service Provider which then grants your device access to the web.
* And an IP Address is assigned to your device from the given range available.
* Your internet activity goes through your service provider, and they route it back to you, using your IP address.
* Your IP address can change. For example, turning your router on or off can change your IP Address.
* When you are out from your home location your home IP address doesn’t accompany you. It changes as you change the network of your device.

**Types of IP Address**

IP Address is of two types:

* **IPv4:**Internet Protocol version 4. It consists of 4 numbers separated by the dots. Each number can be from 0-255 in decimal numbers. But computers do not understand decimal numbers, they instead change them to binary numbers which are only 0 and 1.

IPv4 can be written as:

*189.***123189.123.123.90**

**2. IPv6:-**

IPv6 or Internet Protocol Version 6 is a network layer protocol that allows communication to take place over the network. IPv6 was designed by Internet Engineering Task Force (IETF) in December 1998 with the purpose of superseding the IPv4 due to the global exponentially growing internet users.

#### **Types of IPv6 Address**

Now that we know about what is IPv6 address let’s take a look at its different types.

* **Unicast addresses** It identifies a unique node on a network and usually refers to a single sender or a single receiver.
* **Multicast addresses** It represents a group of IP devices and can only be used as the destination of a datagram.
* **Anycast addresses** It is assigned to a set of interfaces that typically belong to different nodes.

#### **Advantages of IPv6:-**

* Reliability
* **Faster Speeds:** IPv6 supports multicast rather than broadcast in IPv4.This feature allows bandwidth-intensive packet flows (like multimedia streams) to be sent to multiple destinations all at once.
* **Stronger Security:** IPSecurity, which provides confidentiality, and data integrity, is embedded into IPv6.
* Routing efficiency:

Most importantly it’s the final solution for growing nodes in Global-network.

#### **Disadvantages of IPv6**

* **Conversion:** Due to widespread present usage of IPv4 it will take a long period to completely shift to IPv6.
* **Communication:** IPv4 and IPv6 machines cannot communicate directly with each other. They need an intermediate technology to make that possible.

| **IPv4** | **IPv6** |
| --- | --- |
| IPv4 has a 32-bit address length | IPv6 has a 128-bit address length |
| It Supports Manual and DHCP address configuration | It supports Auto and renumbering address configuration |
| In IPv4 end to end, connection integrity is Unachievable | In IPv6 end to end, connection integrity is Achievable |
| It can generate 4.29×109 address space | Address space of IPv6 is quite large it can produce 3.4×1038 address space |
| The Security feature is dependent on application | IPSEC is an inbuilt security feature in the IPv6 protocol |
| Address representation of IPv4 is in decimal | Address Representation of IPv6 is in hexadecimal |

### ISP(internet service provider)

An ISP (internet service provider) is a company that provides individuals and organizations access to the internet and other related services. An ISP has the equipment and the telecommunication line access required to have a [point of presence](https://www.techtarget.com/searchnetworking/definition/point-of-presence-POP) on the internet for the geographic area served.

ISPs make it possible for customers to access the internet while also providing additional services such as email, domain registration and [web hosting](https://www.techtarget.com/whatis/definition/hosting-Web-site-hosting-Web-hosting-and-Webhosting). ISPs may also provide different internet connection types, such as cable and fiber. Connections can also come in the form of high-speed broadband or non-broadband. The Federal Communications Commission (FCC) states that to be considered high-speed, a connection must have download speeds of at least 25 megabits per second ([Mbps](https://www.techtarget.com/searchnetworking/definition/Mbps)) and upload speeds a minimum of 3 Mbps.

### How do ISPs work?

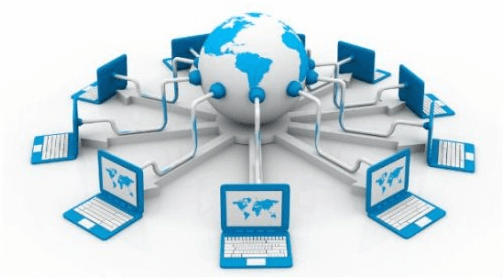
ISPs are connected to one or more high-speed internet lines. Larger ISPs have their own high-speed leased lines, so they are less dependent on telecommunications services and can provide better service to their customers.

ISPs also keep thousands of servers in data centers -- the number of servers depends on their internet service area. These large data centers manage all customer traffic. Multiple ISPs are also connected to large [backbone](https://www.techtarget.com/searchnetworking/definition/backbone) routing centers.

ISPs are grouped into the following three tiers:

* **Tier 1 ISPs.** These ISPs have the most global reach and own enough physical network lines to carry most traffic on their own. They also negotiate with other tier 1 networks to allow free traffic to pass through to other tier 1 providers. Tier 1 ISPs typically sell network access to tier 2 ISPs.
* **Tier 2 ISPs.** These ISPs have regional or national reach and are service providers that connect tier 1 and tier 3 ISPs. They have to purchase access to larger tier 1 networks, but are peers with other tier 2 ISPs. Tier 2 networks focus on consumer and commercial customers.
* **Tier 3 ISPs.** These ISPs connect customers to the internet using another ISP's network. Tier 3 ISPs use and pay higher-tier ISPs for access to internet services. They focus on providing internet access to local businesses and consumer market.
* **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 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.

* **Domain Name System(DNS):-**

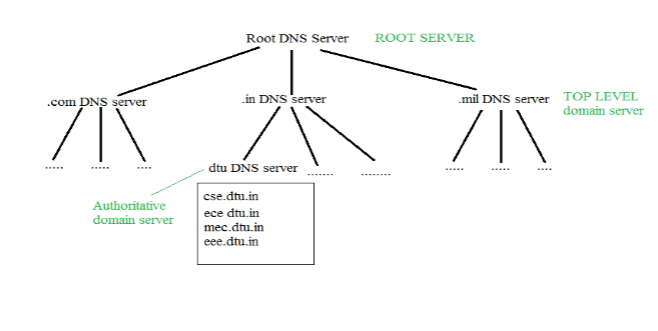
DNS is a hostname for IP address translation service. DNS is a distributed database implemented in a hierarchy of name servers. It is an application layer protocol for message exchange between clients and servers.

**Requirement:** Every host is identified by the IP address but remembering numbers is very difficult for the people also the IP addresses are not static therefore a mapping is required to change the domain name to the IP address. So DNS is used to convert the domain name of the websites to their numerical IP address.

**Domain:** There are various kinds of DOMAIN:

1. Generic domain: .com(commercial) .edu(educational) .mil(military) .org(non profit organization) .net(similar to commercial) all these are generic domain.
2. Country domain .in (india) .us .uk
3. Inverse domain if we want to know what is the domain name of the website. Ip to domain name mapping. So DNS can provide both the mapping for example to find the ip addresses of geeksforgeeks.org then we have to type nslookup www.geeksforgeeks.org.

**Organization of Domain:**

[](https://media.geeksforgeeks.org/wp-content/cdn-uploads/gq/2017/02/DNS.png)

**Dynamic Domain Name System (DDNS) :** It is a method of automatically updating a name server in the Domain Name Server (DNS), often in real-time, with the active DDNS configuration of its configured hostnames, addresses, or other information. In DDNS, when a binding between a name and an address is determined, the information is sent, usually by [DHCP (Dynamic Host Configuration Protocol)](https://www.geeksforgeeks.org/dynamic-host-configuration-protocol-dhcp/) to a primary DNS server. The primary server updates the zone. The secondary servers are notified either actively or passively. Inactive notification, the primary server sends a message to secondary servers, whereas, in the passive notification, the secondary servers periodically check for any changes. In either case, after being notified about the change, the secondary requests information about the entire zone (zone transfer). DDNS can use an authentication mechanism to provide security and prevent unauthorized changes in DNS records.

**Advantages :**

1. It saves time required by static addresses updates manually when network configuration changes.
2. It saves space as the number of addresses are used as required at one time rather than using one for all the possible users of the IP address.
3. It is very comfortable for users point of view as any IP address changes will not affect any of their activities.
4. It does not affect accessibility as changed IP addresses are configured automatically against URL’s.

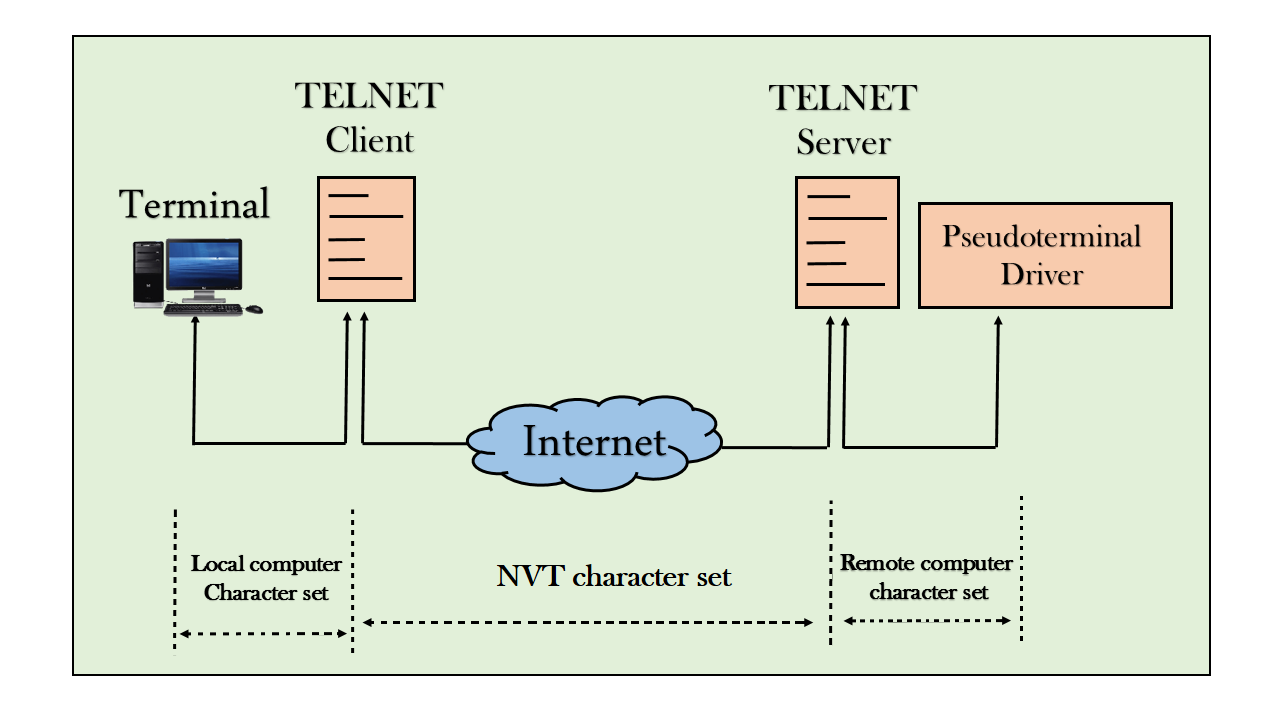
**Disadvantages :**

1. It is less reliable due to lack of static IP addresses and domain name mappings.
2. Dynamic DNS services alone can not make any guarantee about the device you are attempting to connect is actually your own.

[**TELNET**](https://practice.geeksforgeeks.org/problems/explain-telnet)stands for**Tel**etype **Net**work. It is a type of protocol that enables one computer to connect to local computer. It is a used as a standard [**TCP/IP protocol**](https://www.geeksforgeeks.org/tcp-ip-in-computer-networking/) for virtual terminal service which is given by [**ISO**](https://www.geeksforgeeks.org/iso-full-form/). Computer which starts connection known as the**local computer**. Computer which is being connected to i.e. which accepts the connection known as **remote computer**. When the connection is established between local and remote computer. During telnet operation whatever that is being performed on the remote computer will be displayed by local computer. Telnet operates on client/server principle. Local computer uses telnet client program and the remote computers uses telnet server program.

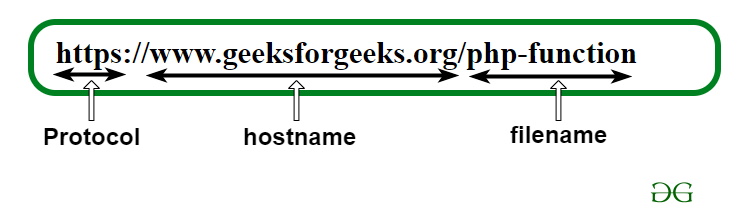
**TELNET Commands :**   
Commands of the telnet are identified by a prefix character, Interpret As Command (IAC) which is having code 255. IAC is followed by command and option codes. Basic format of the command is as shown in the following figure :





# **URL**

**URL is the abbreviation of Uniform Resource Locator.** It is the resource address on the internet. The URL (Uniform Resource Locator) is created by **Tim Berners-Lee** and the Internet Engineering working group in 1994. URL is the character string (address) which is used to access data from the internet. The URL is the type of URI (Uniform Resource Identifier).



A URL contains the following information which is listed below:

* Protocol name
* A colon followed by double forward-slash (://)
* Hostname (domain name) or IP address
* A colon followed by port number (optional – unless specified otherwise, “:80” is the default when using HTTP, and “:443” is the default when using HTTPS)
* Path of the file

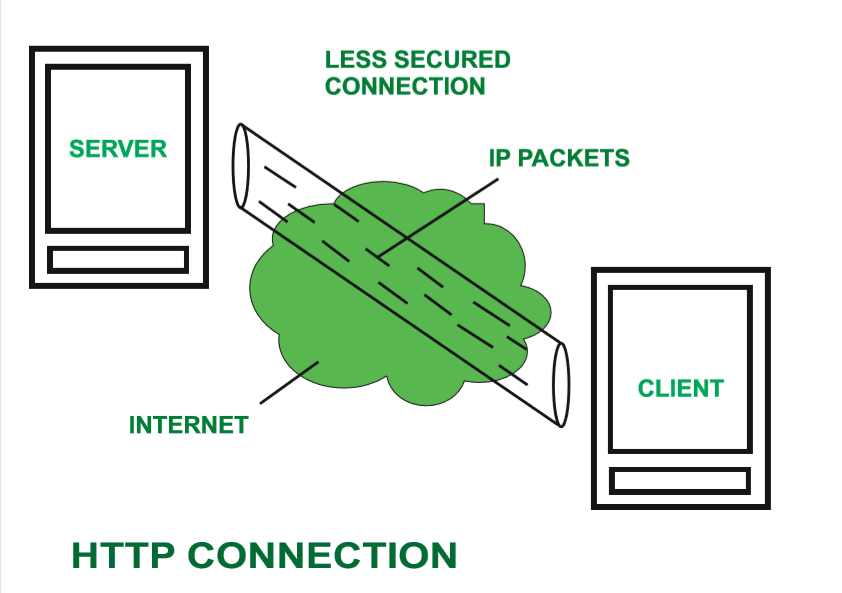
# **HTTP**

**HTTP** stands for HyperText Transfer Protocol. It is invented by **Tim Berner**. HyperText is the type of text which is specially coded with the help of some standard coding language called [HyperText Markup Language (HTML)](https://www.geeksforgeeks.org/html-introduction/). **HTTP/2** is the successor version of HTTP, which was published on May 2015. HTTP/3 is the latest version of HTTP, which is published in 2022.

The protocols that are used to transfer hypertext between two computers is known as HyperText Transfer Protocol.   
HTTP provides standard between a web browser and web server to establish communication. It is set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files are shared on the World Wide Web. Whenever a web user opens their web browser, user indirectly uses HTTP. It is an application protocol which is used for distributed, collaborative, hypermedia information systems.

**How it works ?**   
First of all, whenever we want to open any website then first we open web browser after that we will type URL of that website (e.g., www.facebook.com ). This URL is now sent to [Domain Name Server (DNS)](https://www.geeksforgeeks.org/domain-name-server-dns-in-application-layer/). Then DNS first check records for this URL in their database, then DNS will return IP address to web browser corresponding to this URL. Now browser is able to sent request to actual server.

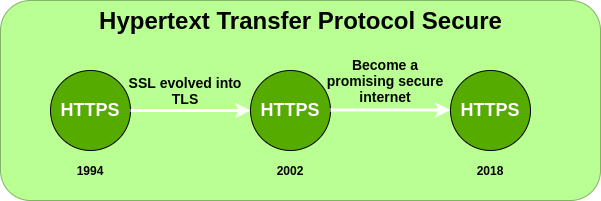
After server sends data to client, connection will be closed. If we want something else from server we should have to re-establish connection between client and server.



# **HTTPS**

**HTTPS** stands for **Hyper Text Transfer Protocol Secure**. HTTP Secure (HTTPS), could be a combination of the Hypertext Transfer Protocol with the SSL/TLS convention to supply encrypted communication and secure distinguishing proof of a arrange web server.   
If the URL of that site is just HTTP, at that point anything you’re perusing or whatever points of interest you’re putting on that site, on the off chance that a programmer needs to take your data.   
Therefore, HTTPS is more secure than HTTP because HTTPS is certified by the SSL(Secure Socket Layer). Whatever website you are visiting on the internet, if its URL is HTTP, then that website is not secure.   
If a website has an SSL certificate installed then the URL of that website will be HTTPS that website will completely secure. You can give any information about your credit card, debit cards, OTP and anything else.

#### HTTP Improved years:



#### Characteristics of HTTPS:

* **Security:**Nowadays there’s a lot of cyber-attacks on the web. And online installments have also expanded. That’s why we need to be secure.If there is no security in any website,then no will use that website.
* **Need of SSL:**Some SEO specialists accept that by introducing SSL on the site, there are a few SEO benefits from Google. And by applying SSL, the positioning of the site in Google is additionally boosted.
* **Authentication:** HTTPS encrypts all message substance, including the HTTP headers and the request/response data.The verification perspective of HTTPS requires a trusted third party to sign server-side digital certificates.
* **Browsing Privately:**HTTPS is presently utilised more frequently by web clients than the first non-secure HTTP, fundamentally to ensure page genuineness on all sorts of websites,secure accounts and to keep client communications.

#### **Advantages of HTTPS:**

* Secures your information in-transit.
* Help you boost income per client.
* Protects your site from Phishing, MITM and other information breaches.
* Builds believe on your site visitors. Removes “NOT Secure” warnings.
* Help you move forward website ranking.

#### **Disadvantages of HTTPS:**

* A web ask with HTTPS is slower which regularly comes about in moderate page stacking.
* Pages with HTTPS can never be cached could be a shared cache.
* A few intermediary serves or firewall frameworks don’t permit get to to locales with HTTPS.
* If you’re making web site which has static contents or if there’s no private information exchange, you’ll select the HTTP.
* Overhead incorporates time to encrypt and decode the information, additional header input for encrypt information, handshaking some time recently exchanging genuine information.

# Secure Socket Layer (SSL)

* Difficulty Level : [Easy](https://www.geeksforgeeks.org/easy/)
* Last Updated : 08 Jun, 2022

 Read

 Discuss

 Practice

 Video

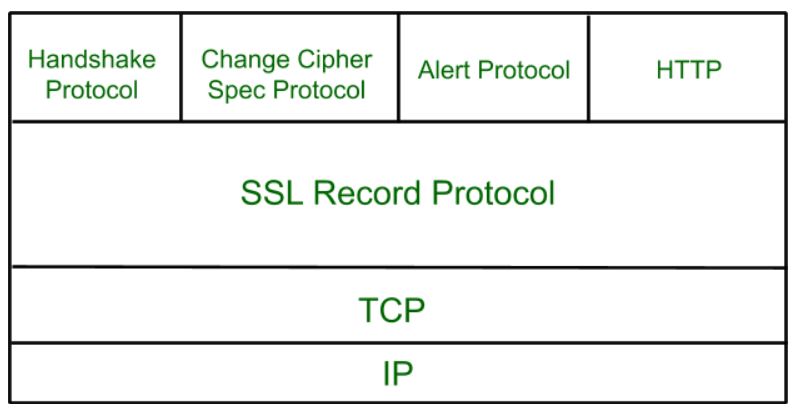
 Courses

[**Secure Socket Layer (SSL)**](https://practice.geeksforgeeks.org/problems/what-is-ssl) provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

**Secure Socket Layer Protocols:**

* SSL record protocol
* Handshake protocol
* Change-cipher spec protocol
* Alert protocol

### SSL Protocol Stack:

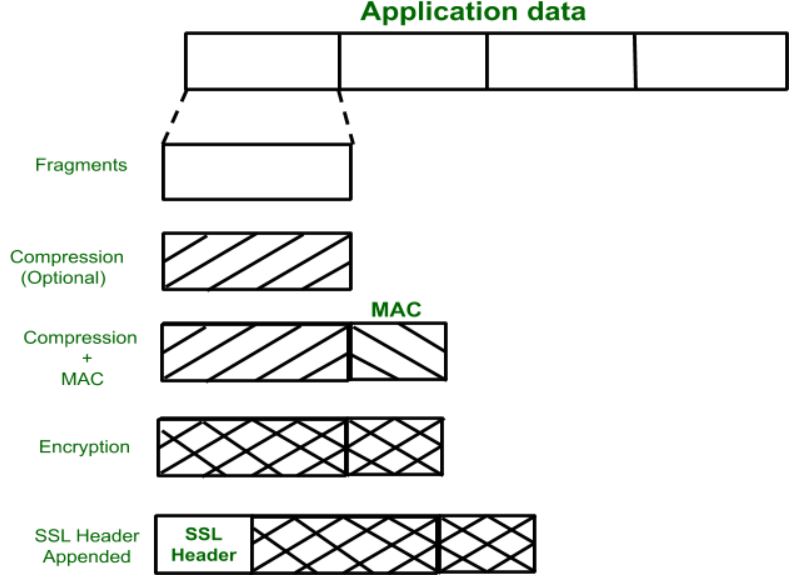


### ****SSL Record Protocol:****

SSL Record provides two services to SSL connection.

* Confidentiality
* Message Integrity

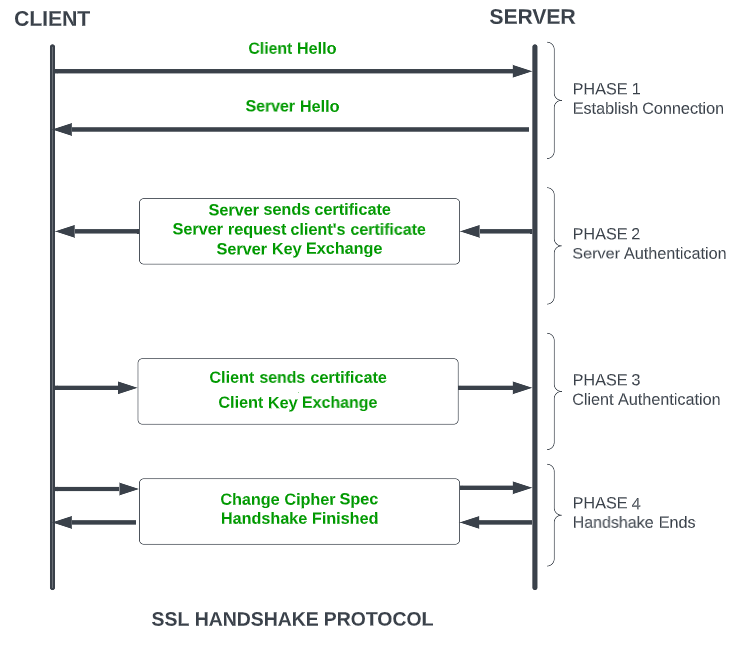
In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



**Handshake Protocol:**

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

* **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
* **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
* **Phase-3:** In this phase, Client replies to the server by sending his certificate and Client-exchange-key.
* **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.



*SSL Handshake Protocol Phases diagrammatic representation*

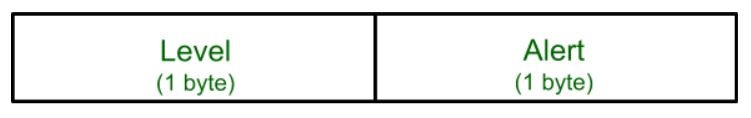
### Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state.   
Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol’s purpose is to cause the pending state to be copied into the current state.



### Alert Protocol:

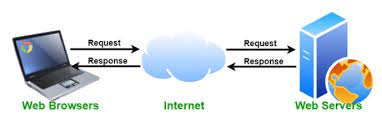
This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.



# **Web Browser**

The first web browser WorldWideWeb was invented in the year of 1990 by Tim Berners-Lee.

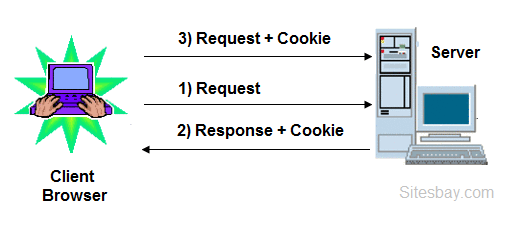
The web browser is an application software to explore www (World Wide Web). It provides an interface between the server and the client and requests to the server for web documents and services. It works as a compiler to render HTML which is used to design a webpage. Whenever we search anything on the internet, the browser loads a web page written in HTML, including text, links, images, and other items such as style sheets and JavaScript functions. Google Chrome, Microsoft Edge, Mozilla Firefox, Safari are examples of web browsers.

****

## **What are cookies on websites:-**

Cookies are small files of information that a web server generates and sends to a web browser. cookies help of website to show data which data user want to see of website. Website can have cookies or can not have cookies can Web browsers store the cookies they receive for a predetermined period of time, or for the length of a user's session on a website. They attach the relevant cookies to any future requests the user makes of the web server.

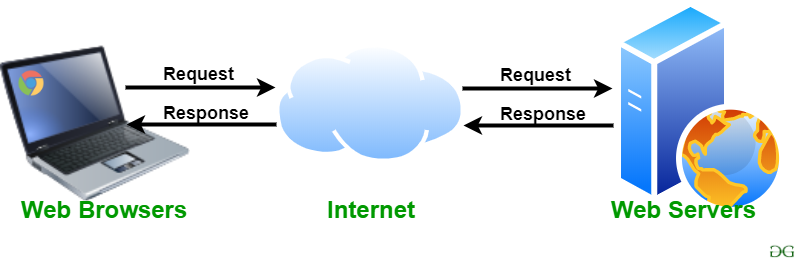
Cookies help inform websites about the user, enabling the websites to personalize the user experience.



# **Web Server:-**

Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

web servers are computers used to store HTTP files which makes a website and when a client requests a certain website, it delivers the requested website to the client. For example, you want to open Facebook on your laptop and enter the URL in the search bar of google. Now, the laptop will send an HTTP request to view the facebook webpage to another computer known as the webserver. This computer (webserver) contains all the files (usually in HTTP format) which make up the website like text, images, gif files, etc. After processing the request, the webserver will send the requested website-related files to your computer and then you can reach the website.

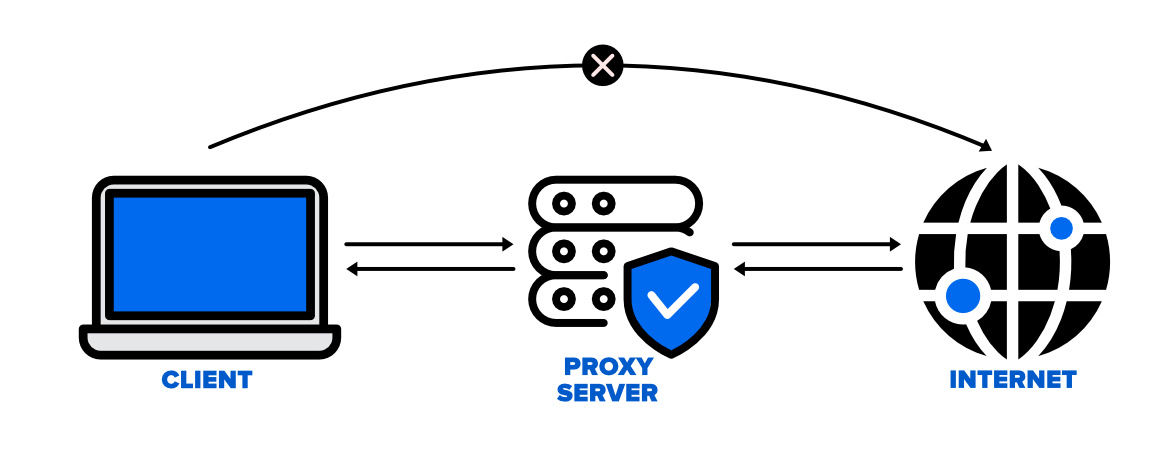


* **PROXY SERVER:-**

Proxy server is a type of server It is help of client to communicate with web server.

By proxy server we can access data from blocked website or ban website without generate our IP address. Proxy server use another IP address to fetch data from block or ban website.

Proxy server refers to a server that acts as an intermediary between the request made by clients, and a particular server for some services or requests for some resources. There are different types of proxy servers available that are put into use according to the purpose of a request made by the clients to the servers. The basic purpose of Proxy servers is to protect the direct connection of Internet clients and internet resources. The proxy server also prevents the identification of the client’s IP address when the client makes any request is made to any other servers.



* **WEB APPLICATION:-**

A web application multiple users can use that application from other other place. Web application (Web app) is **an application program that is stored on a remote server and delivered over the Internet through a browser interface**. Web services are Web apps by definition and many, although not all, websites contain Web apps. According to Web.

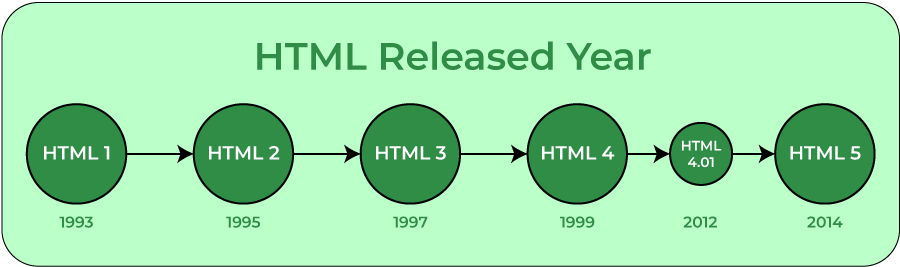
Example of web application:-

Facebook ,Netflix ,Gmail etc.

* **web design principle:-**

**planning the site and navigation:-**

**UNIT:-2:-**

* **HTML:-**
* **HTML** stands for HyperText Markup Language. It is used to design web pages using a markup language. 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. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.
* 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, but the first standard version was HTML 2.0, published in 1995.
* 
* **DHTML:-**

**DHTML** stands for **Dynamic Hypertext Markup language** i.e., **Dynamic HTML**.

Dynamic HTML is not a markup or programming language but it is a term that combines the features of various web development technologies for creating the web pages dynamic and interactive.

The DHTML application was introduced by Microsoft with the release of the 4th version of IE (Internet Explorer) in 1997.

## **Components of Dynamic HTML**

DHTML consists of the following four components or languages:

* HTML 4.0
* CSS
* JavaScript
* DOM.

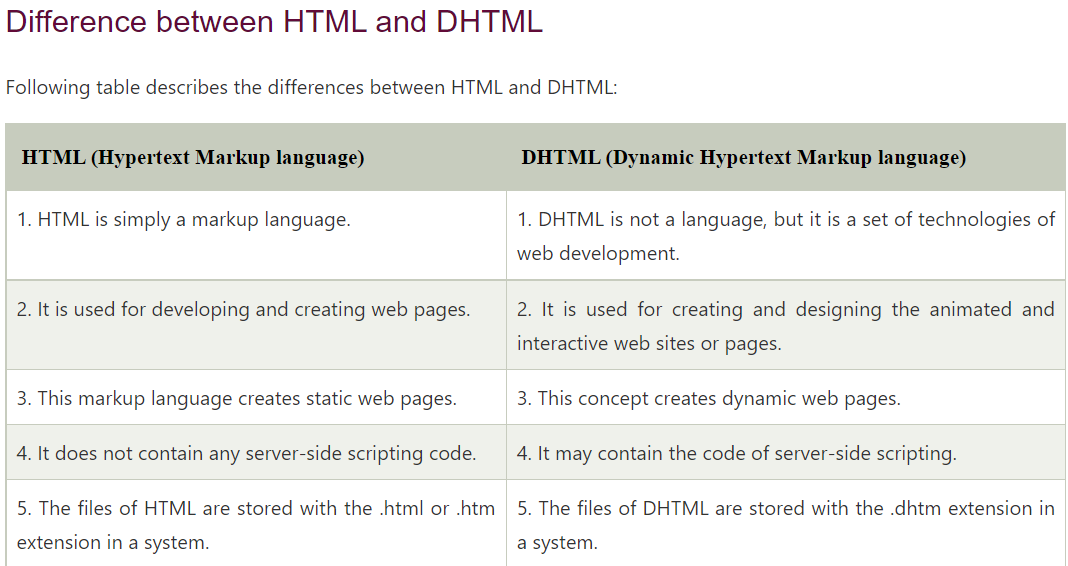
## **Uses of DHTML**

Following are the uses of DHTML (Dynamic HTML):

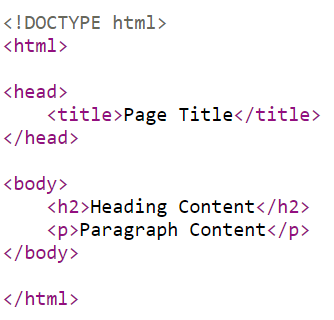
* It is used for designing the animated and interactive web pages that are developed in real-time.
* DHTML helps users by animating the text and images in their documents.
* It allows the authors for adding the effects on their pages.
* It also allows the page authors for including the drop-down menus or rollover buttons.
* This term is also used to create various browser-based action games.
* It is also used to add the ticker on various websites, which needs to refresh their content automatically.

## **Features of DHTML**

Following are the various characteristics or features of DHTML (Dynamic HTML):

* Its simplest and main feature is that we can create the web page dynamically.
* **Dynamic Style** is a feature, that allows the users to alter the font, size, color, and content of a web page.
* It provides the facility for using the events, methods, and properties. And, also provides the feature of code reusability.
* It also provides the feature in browsers for data binding.
* Using DHTML, users can easily create dynamic fonts for their web sites or web pages.
* 

**structure of html document:-**



# **HTML Tag:-**

**Img:-**  **<img src="img\_girl.jpg" alt="Girl in a jacket">**

**LINKS:-** <head>  
   <link rel="stylesheet" href="styles.css">  
 </head>.

**AUDIO** :- <audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

</audio>

**TABLE:-** <table>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>$100</td>  
  </tr>  
</table>

**FORM:-**

<form action="/action\_page.php" method="get">  
  <label for="fname">First name:</label>  
  <input type="text" id="fname" name="fname"><br><br>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"><br><br>  
  <input type="submit" value="Submit">  
</form>

## **Multimedia:-**

Multimedia on the web is sound, music, videos, movies, and animations.

Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Web pages often contain multimedia elements of different types and formats.

Multimedia elements (like audio or video) are stored in media files.

## **HTML Layout Elements:-**

HTML has several semantic elements that define the different parts of a web page:

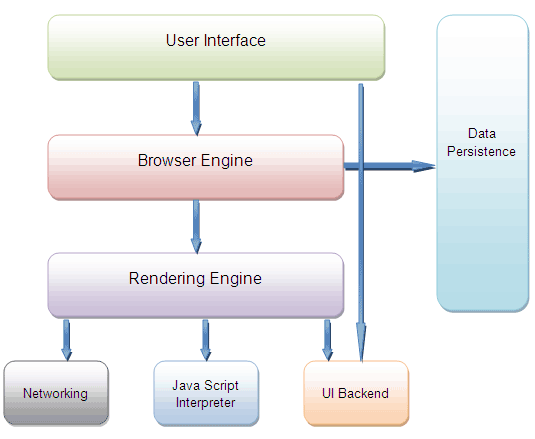
|  |  |
| --- | --- |
| HTML5 Semantic Elements | * <header> - Defines a header for a document or a section * <nav> - Defines a set of navigation links * <section> - Defines a section in a document * <article> - Defines an independent, self-contained content * <aside> - Defines content aside from the content (like a sidebar) * <footer> - Defines a footer for a document or a section * <details> - Defines additional details that the user can open and close on demand * <summary> - Defines a heading for the <details> element |

**META TAG:-**

* The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.
* <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.
* Metadata will not be displayed on the page, but is machine parsable.
* Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

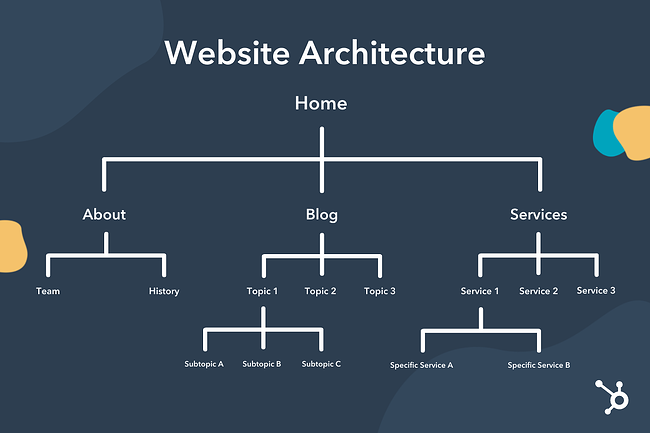
**Architecture of a Browser:-**

The browser’s main functionality is to fetch the files from the server and to display them on the screen. It basically displays html files containing images, PDF, videos, flashes, etc in an ordered layout. A browser is a group of structured codes that performs plenty of tasks to display a webpage on the screen. These codes are separated in to different components according to their tasks performed. The structure of a browser is shown in the below image.



## **website architecture:-**

Website architecture is the hierarchical structure of your website pages. This structure is reflected through internal linking. Your website’s structure should help users easily find information and help search engine crawlers understand the relationship between different pages.



**overview and features of html5:-**

HTML stands for Hypertext Markup Language, and it is the standard markup language for creating web pages and web applications. HTML5 is the 5th version of HTML. With invent of features in HTML5, it’s not only possible to create better websites, but we can also create dynamic websites.

**Now let’s have a look at all the new features that were added in HTML5 that make it better than HTML :**

1. **Intro of**[**audio**](https://www.geeksforgeeks.org/html5-audio/)**and**[**video**](https://www.geeksforgeeks.org/html5-video/)**:**

     Audio and Video tags are the two major addition to HTML5. It allows developers to embed a video or audio on their website.

**2**.[**Vector Graphics**](https://www.geeksforgeeks.org/html-svg-basics/)**:**

It can be used to draw graphics with various shapes and colors via scripting usually JS.

**3**.[**Header**](https://www.geeksforgeeks.org/html-5-header-tag/)**and**[**Footer:**](https://www.geeksforgeeks.org/html5-footer-tag/)

     With these new tags, there is no longer a need to identify the two elements with a <div> tag.

**4**.[**Nav tag:**](https://www.geeksforgeeks.org/html-nav-tag/)

     The <nav> tag defines a set of navigation links. It is used for the part of an internet site that links to different pages at the website.

**5.**[**Progress tag:**](https://www.geeksforgeeks.org/html-5-progress-tag/)

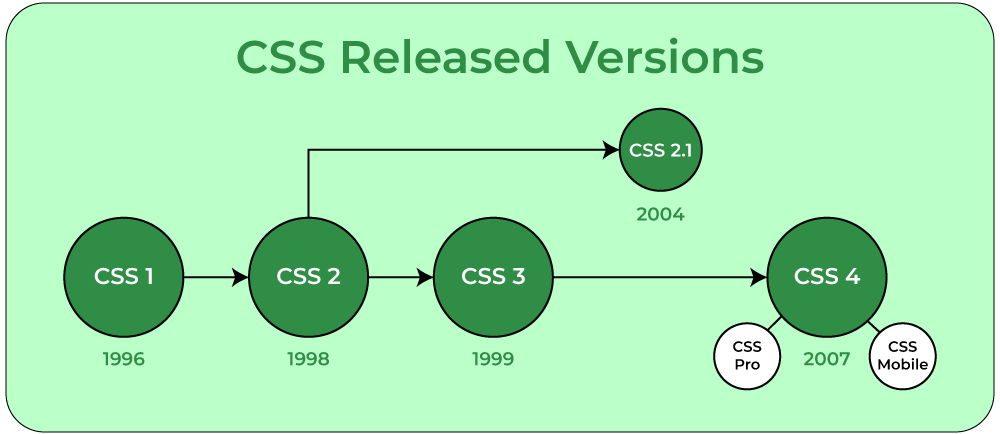
     The progress tag is used to check the progress of a task during the execution.

**6**.[**Placeholder Attribute:**](https://www.geeksforgeeks.org/html-placeholder-attribute/)  
The placeholder attribute specifies a short hint that describes the expected value of an input field/text area.

**UNIT:-3**

## **What is CSS**

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



# **Types of CSS (Cascading Style Sheet)**

Cascading Style Sheet(CSS) is used to set the style in web pages that contain HTML elements. It sets the background color, font-size, font-family, color, … etc property of elements on a web page.   
There are three types of CSS which are given below: 

* Inline CSS
* Internal or Embedded CSS
* External CSS

**Inline CSS:** Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.   
**Example:**

* html

|  |
| --- |
| <!DOCTYPE html>  <**html**>      <**head**>          <**title**>Inline CSS</**title**>      </**head**>        <**body**>          <**p** style = "color:#009900; font-size:50px;                  font-style:italic; text-align:center;">              GeeksForGeeks          </**p**>        </**body**>  </**html**> |

**Internal or Embedded CSS:-**

 This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.   
**Example:**

* html

|  |
| --- |
| <!DOCTYPE html>  <**html**>      <**head**>          <**title**>Internal CSS</**title**>          <**style**>              .main {                  text-align:center;              }              .GFG {                  color:#009900;                  font-size:50px;                  font-weight:bold;              }              .geeks {                  font-style:bold;                  font-size:20px;              }          </**style**>      </**head**>      <**body**>          <**div** class = "main">              <**div** class ="GFG">GeeksForGeeks</**div**>                <**div** class ="geeks">                  A computer science portal for geeks              </**div**>          </**div**>      </**body**>  </**html**> |

**External CSS:** External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using **link** tag. This means that for each element, style can be set only once and that will be applied across web pages.  
**Example:** The file given below contains CSS property. This file save with .css extension. For Ex: **geeks.css**

body {

|  |
| --- |
| <!DOCTYPE html>  <**html**>      <**head**>          <**link** rel="stylesheet" href="geeks.css"/>      </**head**>        <**body**>          <**div** class = "main">              <**div** class ="GFG">GeeksForGeeks</**div**>              <**div** id ="geeks">                  A computer science portal for geeks              </**div**>          </**div**>      </**body**>  </**html**> |

## **position Property:-**

The position property specifies the type of positioning method used for an element.

There are five different position values:

* static
* relative
* fixed
* absolute
* sticky

## **position: static:-**

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

This <div> element has position: static;

Example

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

## **position: relative:-**

An element with position: relative; is positioned relative to its normal position.

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

This <div> element has position: relative;

Here is the CSS that is used:

### Example

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

## **position: fixed:-**

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

A fixed element does not leave a gap in the page where it would normally have been located.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

### Example

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

## **position: absolute:-**

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

**Note:** Absolute positioned elements are removed from the normal flow, and can overlap elements.

Here is a simple example:

This <div> element has position: relative;

This <div> element has position: absolute;

Here is the CSS that is used:

### Example

div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}  
  
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}

## **position: sticky:-**

An element with position: sticky; is positioned based on the user's scroll position.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

**Note:**Internet Explorer does not support sticky positioning. Safari requires a -webkit- prefix (see example below). You must also specify at least one of top, right, bottom or left for sticky positioning to work.

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

### Example

div.sticky {  
  position: -webkit-sticky; /\* Safari \*/  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}

# **CSS3 Introduction:-**

* CSS3 is the latest standard for CSS.
* CSS3 is completely backwards-compatible with earlier versions of CSS.
* This section teaches you about the new features in CSS3!

## **CSS3 Modules:-**

CSS3 has been split into "modules". It contains the "old CSS specification" (which has been split into smaller pieces). In addition, new modules are added.

Some of the most important CSS3 modules are:

* Selectors
* Box Model
* Backgrounds and Borders
* Image Values and Replaced Content
* Text Effects
* 2D/3D Transformations
* Animations
* Multiple Column Layout
* User Interface
* **CSS Text Formeting:-**

1. Text color

## 2.Text Alignment and Text Direction.

* text-align
* text-align-last
* direction
* unicode-bidi
* vertical-align

## 3.Text Decoration

* text-decoration-line
* text-decoration-color
* text-decoration-style
* text-decoration-thickness
* text-decoration

## 4.Text Transformation

The text-transform property is used to specify uppercase and lowercase letters in a text.

p.uppercase {  
  text-transform: uppercase;  
}  
  
p.lowercase {  
  text-transform: lowercase;  
}  
  
p.capitalize {  
  text-transform: capitalize;  
}

## 5.text Spacing:-

In this chapter you will learn about the following properties:

* text-indent
* letter-spacing
* line-height
* word-spacing
* white-space

## Text Shadow:-

The text-shadow property adds shadow to text.

h1 {  
  text-shadow: 2px 2px;  
}

**UNIT-4**

# **Introduction to JavaScript:-**

which is also known as the scripting language for webpages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for [**Client-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [**Server-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments. Javascript is both imperative and declarative type of language. JavaScript contains a standard library of objects, like [**Array**](https://www.geeksforgeeks.org/arrays-in-javascript/), [**Date**](https://www.geeksforgeeks.org/javascript-date-objects/), and [**Math**](https://www.geeksforgeeks.org/javascript-math-object/), and a core set of language elements like [**operators**](https://www.geeksforgeeks.org/javascript-operators/), **control structures**, and [**statements**](https://www.geeksforgeeks.org/javascript-statements/).

# **JavaScript History:-**

**JavaScript** was invented by **Brendan Eich** in 1995.

It was developed for **Netscape 2**, and became the **ECMA-262** standard in 1997

# **JavaScript Versions:-**

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997.

ECMAScript is the official name of the language.

ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6.

Since 2016, versions are named by year (ECMAScript 2016, 2017, 2018, 2019, 2020).

# **JavaScript Basic Syntax:-**

**JavaScript** is a lightweight and dynamic computer programming language. It is used to create client-side dynamic pages. It is an open-source and cross-platform language.

**Basic Syntax:**

* Javascript

|  |
| --- |
| <script>  document.write("Basic Print method in JavaScript");  </script> |

**JavaScript Variables:** A JavaScript variable is the simple name of storage location where data to be stored. There are two types of variables in JavaScript which are listed below:

* **Local variables:** Declare a variable inside of block or function.
* **Global variables:** Declare a variable outside function or with a window object.

**Example:**

|  |
| --- |
| <script>    // Declare a variable and initialize it  // Global variable declaration  **var** Name="Apple";    // Function definition  **function** MyFunction() {        // Local variable declaration  **var** num = 45;        // Display the value of Global variable      document.writeln(Name);        // Display the value of local variable      document.writeln("<br>" + num );  }    // Function call  MyFunction();    </script> |

# **JavaScript Data Types:-**

JavaScript variables can hold different data types: numbers, strings, objects,booleans and more:

## **JavaScript Objects:-**

JavaScript objects are written with curly braces {}.

Object properties are written as name:value pairs, separated by commas.

### Example

const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

## **JavaScript Statements:-**

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

This statement tells the browser to write "Hello Dolly." inside an HTML element with id="demo":

### Example

document.getElementById("demo").innerHTML = "Hello Dolly.";

Most JavaScript programs contain many JavaScript statements.

The statements are executed, one by one, in the same order as they are written.

## **Types of JavaScript Operators:-**

There are different types of JavaScript operators:

* Arithmetic Operators
* Assignment Operators
* Comparison Operators
* Logical Operators
* Conditional Operators

# **JavaScript Functions:-**

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

## **JavaScript Function Syntax:-**

A JavaScript function is defined with the function keyword, followed by a **name**, followed by parentheses **()**.

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas:  
**(parameter1, parameter2, ...)**

The code to be executed, by the function, is placed inside curly brackets: **{}**

function name(parameter1, parameter2, parameter3) {  
  // code to be executed  
}

# **JavaScript Arrays:-**

An array is a special variable, which can hold more than one value:

const cars = ["Saab", "Volvo", "BMW"];

## **Why Use Arrays:-**

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

let car1 = "Saab";  
let car2 = "Volvo";  
let car3 = "BMW";

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

## **Creating an Array:-**

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

var array\_name = [item1, item2, ...];

# **JavaScript Objects:-**

In JavaScript, almost "everything" is an object.

* Booleans can be objects (if defined with the new keyword)
* Numbers can be objects (if defined with the new keyword)
* Strings can be objects (if defined with the new keyword)
* Dates are always objects
* Maths are always objects
* Regular expressions are always objects
* Arrays are always objects
* Functions are always objects
* Objects are always objects

All JavaScript values, except primitives, are objects.

## **JavaScript Primitives:-**

A **primitive value** is a value that has no properties or methods.

**3.14** is a primitive value

A **primitive data type** is data that has a primitive value.

JavaScript defines 7 types of primitive data types:

## **Examples**

* string
* number
* boolean
* null
* undefined
* symbol
* bigint

# **Dialog Boxes:-**

JavaScript supports three important types of dialog boxes. These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the users. Here we will discuss each dialog box one by one.

### Example

<html>

<head>

<script type = "text/javascript">

<!--

function Warn() {

alert ("This is a warning message!");

document.write ("This is a warning message!");

}

//-->

</script>

</head>

</html>

## **Confirmation Dialog Box:-**

A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: **OK** and **Cancel**.

If the user clicks on the OK button, the window method **confirm()** will return true. If the user clicks on the Cancel button, then **confirm()** returns false. You can use a confirmation dialog box as follows.

### Example

<html>

<head>

<script type = "text/javascript">

<!--

function getConfirmation() {

var retVal = confirm("Do you want to continue ?");

if( retVal == true ) {

document.write ("User wants to continue!");

return true;

} else {

document.write ("User does not want to continue!");

return false;

}

}

//-->

</script>

</head>

<body>

<p>Click the following button to see the result: </p>

<form>

<input type = "button" value = "Click Me" onclick = "getConfirmation();" />

</form>

</body>

</html>

## **Prompt Dialog Box:-**

The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus, it enables you to interact with the user. The user needs to fill in the field and then click OK.

<script type = "text/javascript">

<!--

function getValue() {

var retVal = prompt("Enter your name : ", "your name here");

document.write("You have entered : " + retVal);

}

//-->

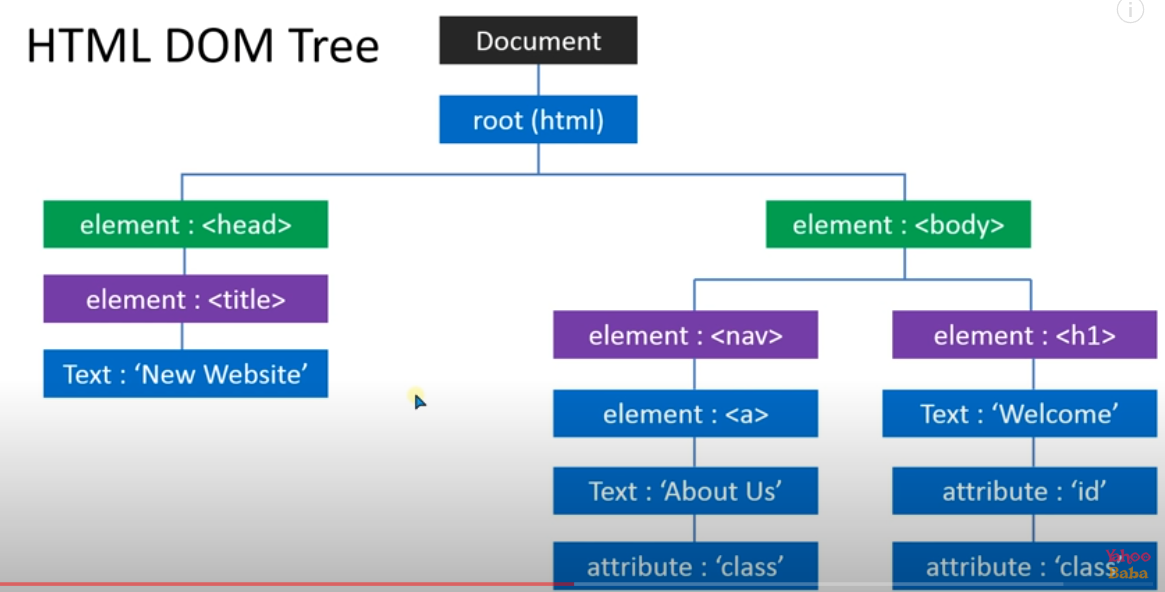
</script>

**javascript DOM:-**

**DOM** :- Document Object Module .

In JavaScript DOM We can Get ,Set , Add New element in Html , Delete Old element in Html.

DOM is work as a tree.



**UNIT-5**

# **PHP - Introduction**

 first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP is an acronym for "PHP: Hypertext Preprocessor"
* PHP is a widely-used, open source scripting language
* PHP scripts are executed on the server
* PHP is free to download and use

## **What is a PHP File:-**

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code is executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".php"

## **What Can PHP Do:-**

* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* **MYSQL:-**

MYSQL is an open source relational database management system that was originally released in 1995.

**MySQL is an open source relational database management system that was originally released in 1995**. MySQL is popular among all databases, and is ranked as the 2nd most popular database, only slightly trailing Oracle Database. Among open source databases, MySQL is the most popular database in use today.

## **PHP Data Types:-**

Variables can store data of different types, and different data types can do different things.

PHP supports the following data types:

* String(A string can be any text inside quotes. You can use single or double quotes:)
* Integer
* Float (floating point numbers - also called double)
* Boolean
* Array
* Object
* NULL
* Resource

## **PHP Variables:-**

A vaiable is use to contain the value like :-

Var x = “Pankaj”

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume).

Rules for PHP variables:

* A variable starts with the $ sign, followed by the name of the variable
* A variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
* Variable names are case-sensitive ($age and $AGE are two different variables)

## **PHP Variables Scope:-**

In PHP, variables can be declared anywhere in the script.

The scope of a variable is the part of the script where the variable can be referenced/used.

PHP has three different variable scopes:

* local
* global
* static

## **static Keyword:-**

Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job.

To do this, use the static keyword when you first declare the variable:

### Example

<?php  
function myTest() {  
  static $x = 0;  
  echo $x;  
  $x++;  
}  
  
myTest();  
myTest();  
myTest();  
?>

# **Decision Making**

The if, elseif ...else and switch statements are used to take decision based on the different condition.

You can use conditional statements in your code to make your decisions. PHP supports following three decision making statements −



* **if...else statement** − use this statement if you want to execute a set of code when a condition is true and another if the condition is not true
* **elseif statement** − is used with the if...else statement to execute a set of code if **one** of the several condition is true
* **switch statement** − is used if you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..elseif..else code.

## **The If...Else Statement**

If you want to execute some code if a condition is true and another code if a condition is false, use the if....else statement.

### Syntax

if (*condition*)

*code to be executed if condition is true;*

else

*code to be executed if condition is false;*

<html>

<body>

<?php

$d = date("D");

if ($d == "Fri")

echo "Have a nice weekend!";

else

echo "Have a nice day!";

?>

</body>

</html>

## **The ElseIf Statement**

If you want to execute some code if one of the several conditions are true use the elseif statement

### Syntax

if (*condition*)

*code to be executed if condition is true;*

elseif (*condition*)

*code to be executed if condition is true;*

else

*code to be executed if condition is false;*

<html>

<body>

<?php

$d = date("D");

if ($d == "Fri")

echo "Have a nice weekend!";

elseif ($d == "Sun")

echo "Have a nice Sunday!";

else

echo "Have a nice day!";

?>

</body>

</html>

## **The Switch Statement:-**

If you want to select one of many blocks of code to be executed, use the Switch statement.

The switch statement is used to avoid long blocks of if..elseif..else code.

### Syntax

switch (*expression*){

case *label1:*

*code to be executed if expression = label1;*

break;

case *label2:*

*code to be executed if expression = label2;*

break;

default:

*code to be executed*

*if expression is different*

*from both label1 and label2;*

}

## **PHP Loops:-**

Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops.

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

* while - loops through a block of code as long as the specified condition is true
* do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
* for - loops through a block of code a specified number of times
* foreach - loops through a block of code for each element in an array

## **while Loop:-**

The while loop executes a block of code as long as the specified condition is true.

### Syntax

while (*condition is true*) {  
*code to be executed*;  
}

## **do...while Loop:-**

The do...while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

### Syntax

do {  
*code to be executed;*} while (*condition is true*);

## **for Loop:-**

The for loop is used when you know in advance how many times the script should run.

### Syntax

for (*init counter; test counter; increment counter*) {  
  *code to be executed for each iteration;*  
}

## **Defined Functions:-**

Besides the built-in PHP functions, it is possible to create your own functions.

* A function is a block of statements that can be used repeatedly in a program.
* A function will not execute automatically when a page loads.
* A function will be executed by a call to the function.

## **Create a User Defined Function in PHP:-**

A user-defined function declaration starts with the word function:

### Syntax

function *functionName*() {  
*code to be executed*;  
}

Example

<?php  
function writeMsg() {  
  echo "Hello world!";  
}  
  
writeMsg(); // call the function  
?>

## **Array:-**

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

$cars1 = "Volvo";  
$cars2 = "BMW";  
$cars3 = "Toyota";

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

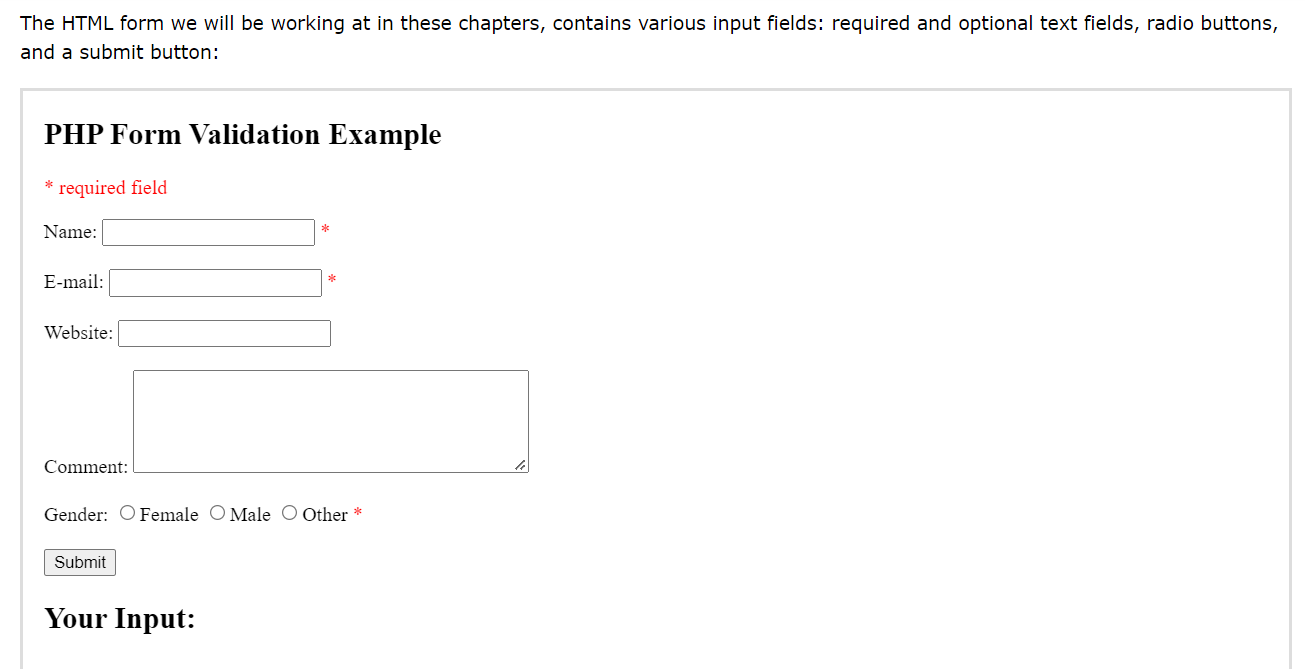
The solution is to create an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

### Example

<?php  
$cars = array("Volvo", "BMW", "Toyota");  
echo count($cars);  
?>

# **PHP Form Validation:-**



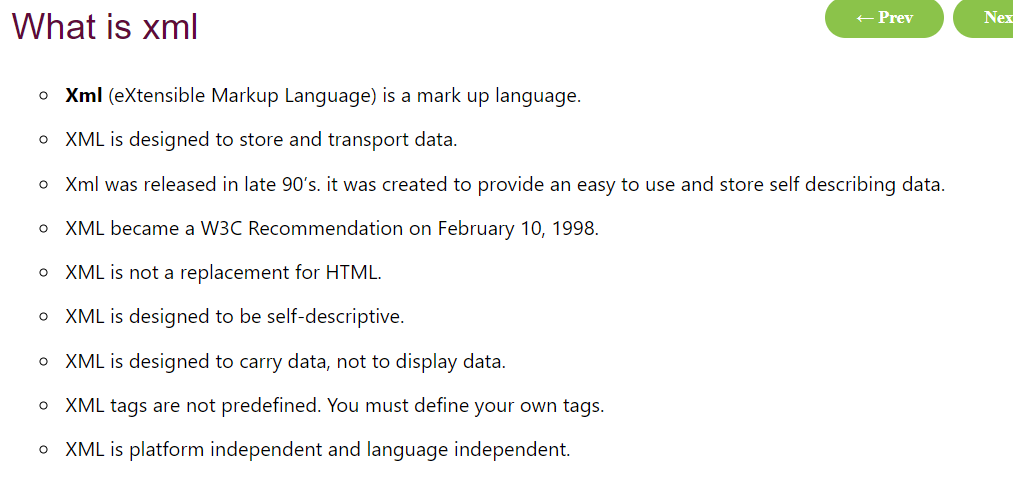
## **PHP - Validate E-mail**

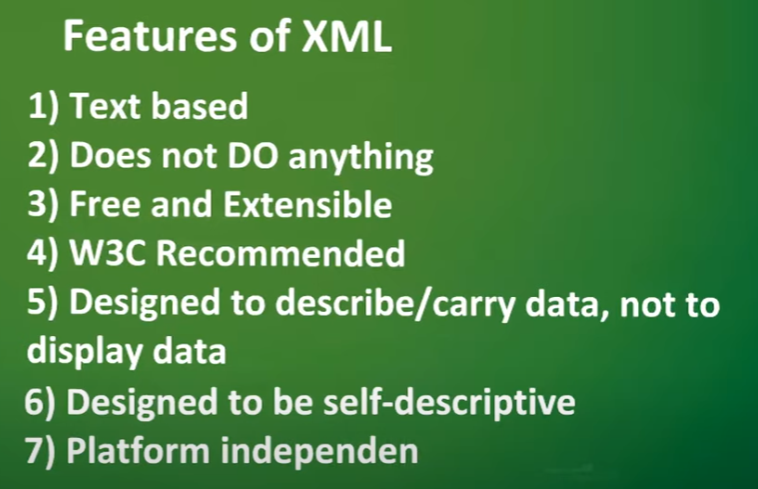
The easiest and safest way to check whether an email address is well-formed is to use PHP's filter\_var() function.

In the code below, if the e-mail address is not well-formed, then store an error message:

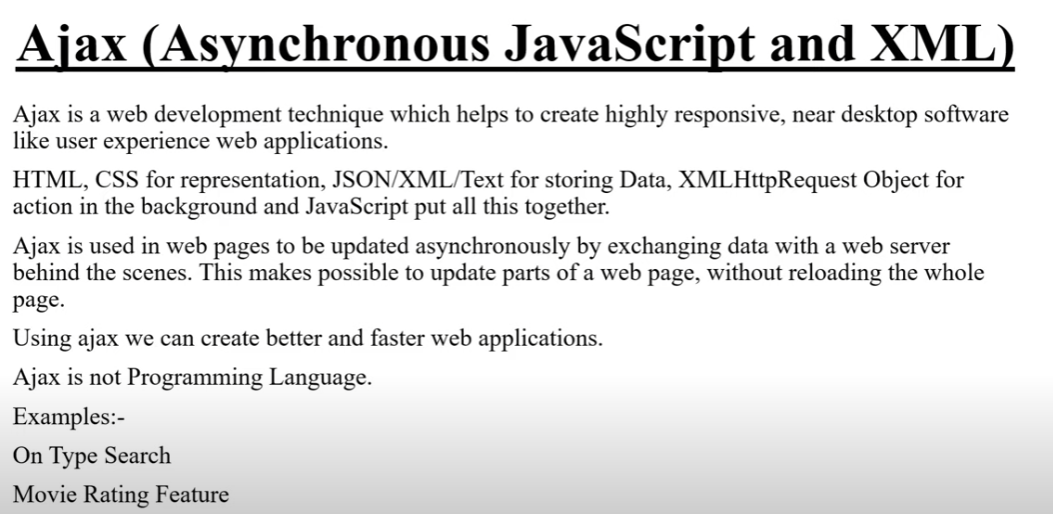
**UNIT-6**

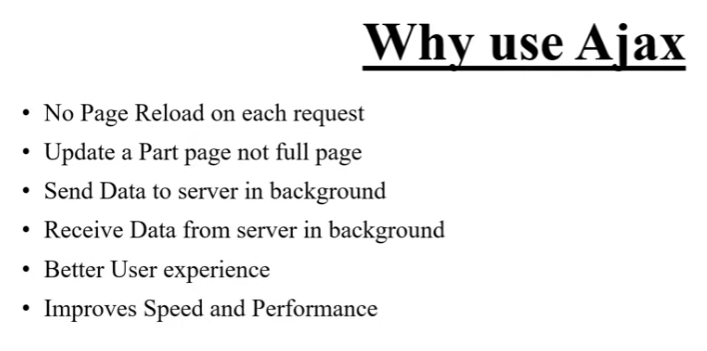
* **XML:-**



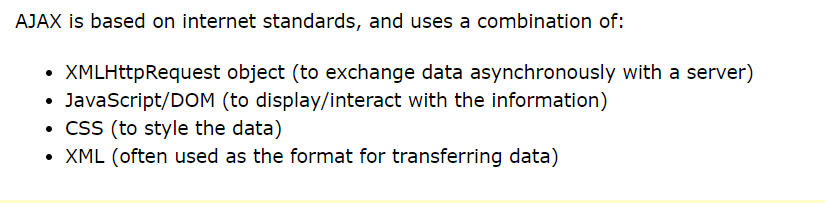


* **AJAX:-**





* **AJAX COMPONENTS:-**



# **XMLHttpRequest:-**

XMLHttpRequest (XHR) objects are used to interact with servers. You can retrieve data from a URL without having to do a full page refresh. This enables a Web page to update just part of a page without disrupting what the user is doing.

XMLHttpRequest is used heavily in [AJAX](https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX) programming.

EventTargetXMLHttpRequestEventTargetXMLHttpRequest

Despite its name, XMLHttpRequest can be used to retrieve any type of data, not just XML.

If your communication needs to involve receiving event data or message data from a server, consider using [server-sent events](https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events) through the [EventSource](https://developer.mozilla.org/en-US/docs/Web/API/EventSource) interface. For full-duplex communication, [WebSockets](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API) may be a better choice.

**JSON** :-JavaScript Object Notation, referred to as JSON in short, is one of the most popular formats for data storage and data interchange over the internet. The simplicity of the JSON syntax makes it very easy for humans and machines to read and write.

Despite its name, the use of the JSON data format is not limited to JavaScript. Most programming languages implement data structures that you can easily convert to JSON and vice versa.

JSON provide API to communicate with server.

# **JSON Syntax:-**

The JSON syntax is a subset of the JavaScript syntax. **JSON** stands for **J**ava**S**cript **O**bject **N**otation. It is a format for structuring data. This format is used by different web applications to communicate with each other. JSON is the replacement of the XML data exchange format in JSON. It is easy to struct the data compare to XML. It supports data structures like arrays and objects and the JSON documents that are rapidly executed on the server. It is also a Language-Independent format that is derived from JavaScript.

**Features of JSON:**

* **Easy to understand:** JSON is easy to read and write.
* **Format:** It is a text-based interchange format. It can store any kind of data in an array of video, audio, and image anything that you required.
* **Dependency:** It is an Independent language that is text-based. It is much faster compared to other text-based structured data.

**Advantages of JSON:-**

* JSON stores all the data in an array so data transfer makes easier. That’s why JSON is the best for sharing data of any size even audio, video, etc.
* Its syntax is very easy to use.

**Disadvantages of JSON:-**

* The main disadvantage for JSON is that there is no error handling in JSON,
* JSON has limited supported tools that we can use during JSON development.

## **JSON Syntax Rules:-**

JSON syntax is derived from JavaScript object notation syntax:

* Data is in name/value pairs
* Data is separated by commas
* Curly braces hold objects
* Square brackets hold arrays
* Example
* "name":"John"

# **JSON Data Types:-**

## **Valid Data Types**

In JSON, values must be one of the following data types:

* a string
* a number
* an object (JSON object)
* an array
* a boolean
* null

# **JSON Object Literals:-**

This is a JSON string:

'{"name":"John", "age":30, "car":null}'

Inside the JSON string there is a JSON object literal:

{"name":"John", "age":30, "car":null}

JSON object literals are surrounded by curly braces {}.

JSON object literals contains key/value pairs.

Keys and values are separated by a colon.

Keys must be strings, and values must be a valid JSON data type:

* string
* number
* object
* array
* boolean
* null

Each key/value pair is separated by a comma.

It is a common mistake to call a JSON object literal "a JSON object".

JSON cannot be an object. JSON is a string format.

The data is only JSON when it is in a string format. When it is converted to a JavaScript variable, it becomes a JavaScript object.

You can create a JavaScript object from a JSON object literal:

Example

myObj = {"name":"John", "age":30, "car":null};

# **JSON Server:-**

A common use of JSON is to exchange data to/from a web server.

When receiving data from a web server, the data is always a string.

Parse the data with JSON.parse(), and the data becomes a JavaScript object.