

## WWW Overview

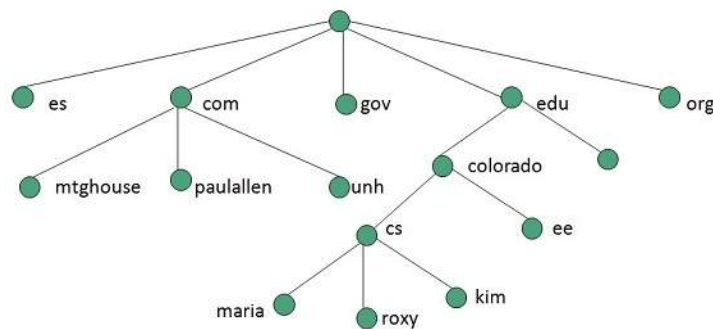
**WWW** stands for **World Wide Web**. A technical definition of the World Wide Web is : all the resources and users on the Internet that are using the Hypertext Transfer Protocol (HTTP).

A broader definition comes from the organization that Web inventor **Tim Berners-Lee** helped found, the **World Wide Web Consortium (W3C)**.

The World Wide Web is the universe of network-accessible information, an embodiment of human knowledge.

In simple terms, The World Wide Web is a way of exchanging information between computers on the Internet, tying them together into a vast collection of interactive multimedia resources.

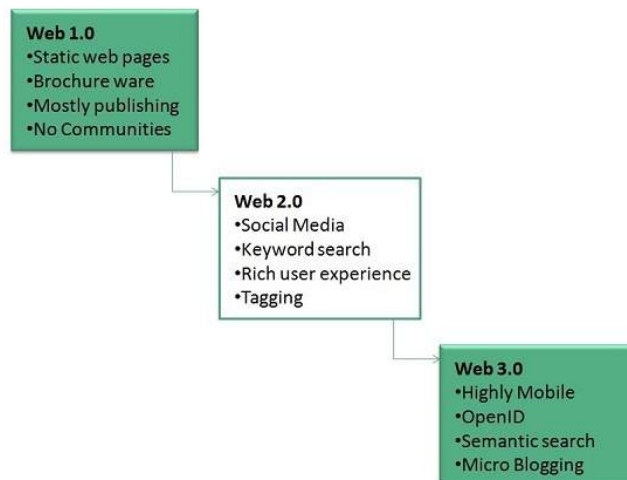
**Internet** and **Web** is not the same thing: Web uses internet to pass over the information.



## Evolution

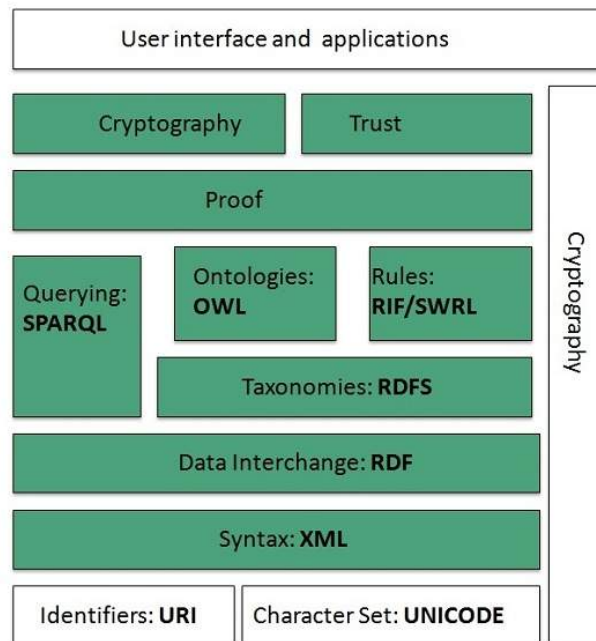
**World Wide Web** was created by **Timothy Berners Lee** in 1989 at **CERN** in **Geneva**. World Wide Web came into existence as a proposal by him, to allow researchers to work together effectively and efficiently at **CERN**. Eventually it became **World Wide Web**.

The following diagram briefly defines evolution of World Wide Web:



## WWW Architecture

WWW architecture is divided into several layers as shown in the following diagram:



### Identifiers and Character Set

**Uniform Resource Identifier (URI)** is used to uniquely identify resources on the web and **UNICODE** makes it possible to build web pages that can be read and write in human languages.

### Syntax

**XML (Extensible Markup Language)** helps to define common syntax in semantic web.

### Data Interchange

**Resource Description Framework (RDF)** framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

### Taxonomies

**RDF Schema (RDFS)** allows more standardized description of **taxonomies** and other **ontological** constructs.

### Ontologies

**Web Ontology Language (OWL)** offers more constructs over RDFS. It comes in following three versions:

- OWL Lite for taxonomies and simple constraints.

- OWL DL for full description logic support.
- OWL for more syntactic freedom of RDF

## Rules

**RIF** and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL**. Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

## Proof

All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

## Cryptography

**Cryptography** means such as digital signature for verification of the origin of sources is used.

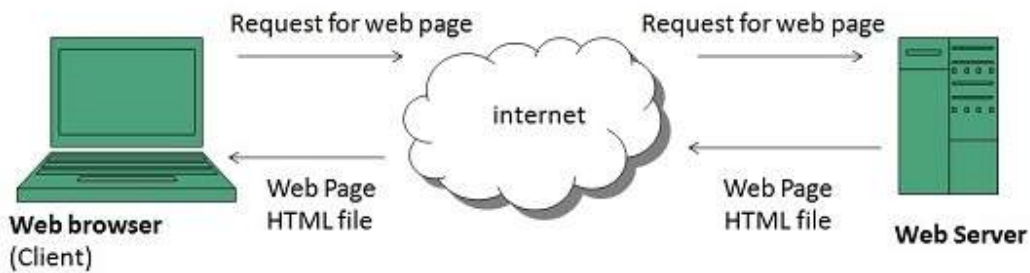
## User Interface and Applications

On the top of layer **User interface and Applications** layer is built for user interaction.

## WWW Operation

**WWW** works on client- server approach. Following steps explains how the web works:

1. User enters the URL of the web page in the address bar of web browser.
2. Then browser requests the Domain Name Server for the IP address corresponding to `www.tutorialspoint.com`.
3. After receiving IP address, browser sends the request for web page to the web server using HTTP protocol which specifies the way the browser and web server communicates.
4. Then web server receives request using HTTP protocol and checks its search for the requested web page. If found it returns it back to the web browser and close the HTTP connection.
5. Now the web browser receives the web page, it interprets it and display the contents of web page in web browser's window.



## Future

There had been a rapid development in field of web. It has its impact in almost every area such as education, research, technology, commerce, marketing etc. So the future of web is almost unpredictable.

Apart from huge development in field of WWW, there are also some technical issues that W3 consortium has to cope up with.

## User Interface

Work on higher quality presentation of 3-D information is under deveopment. The W3 Consortium is also looking forward to enhance the web to full fill requirements of global communities which would include all regional languages and writing systems.

## Technology

Work on privacy and security is under way. This would include hiding information, accounting, access control, integrity and risk management.

## Architecture

There has been huge growth in field of web which may lead to overload the internet and degrade its performance. Hence more better protocol are required to be developed.

# Web Pages

## Web Page

**web page** is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.

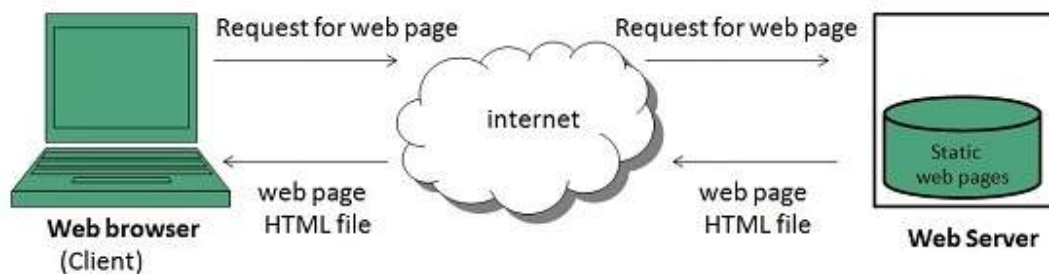
A web page can contain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.

Collection of linked web pages on a web server is known as **website**. There is unique **Uniform Resource Locator (URL)** is associated with each web page.

### Static Web page

**Static web pages** are also known as flat or stationary web page. They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information. User can only read the information but can't do any modification or interact with the information.

Static web pages are created using only HTML. Static web pages are only used when the information is no more required to be modified.



### Dynamic Web page

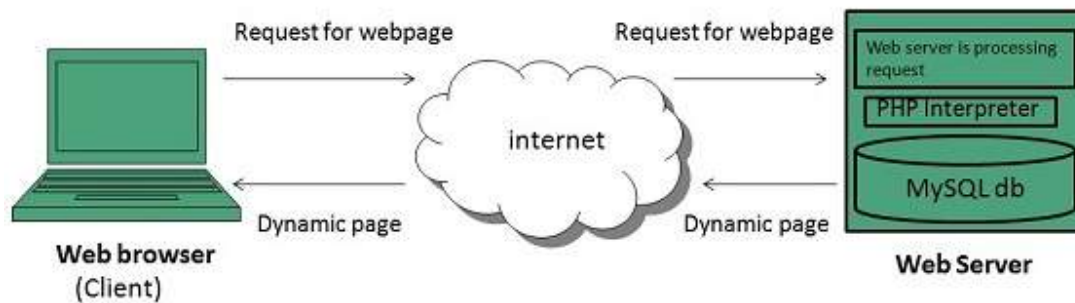
**Dynamic web page** shows different information at different point of time. It is possible to change a portion of a web page without loading the entire web page. It has been made possible using **Ajax** technology.

#### Server-side dynamic web page

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

#### Client-side dynamic web page

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



## Scripting Languages

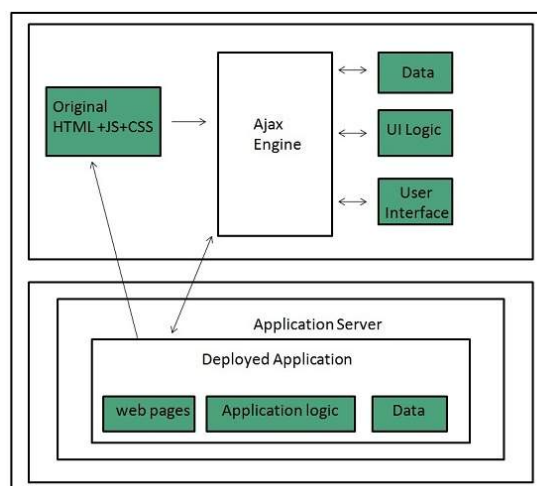
Scripting languages are like programming languages that allow us to write programs in form of script. These scripts are interpreted not compiled and executed line by line.

Scripting language is used to create dynamic web pages.

## Client-side Scripting

**Client-side scripting** refers to the programs that are executed on client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.

Client-side scripting programs can be embedded into HTML files or also can be kept as separate files.

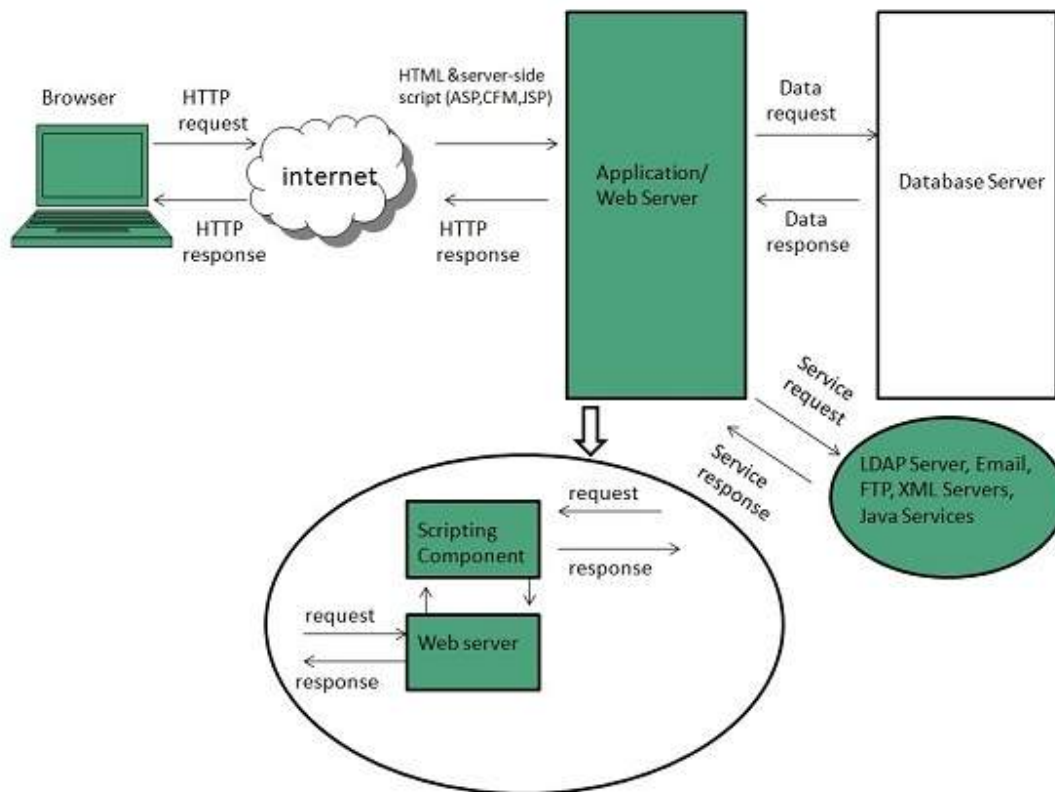


Following table describes commonly used Client-Side scripting languages:

S.N.	Scripting Language Description
1.	<b>JavaScript</b> It is a prototype based scripting language. It inherits its naming conventions from java. All java script files are stored in file having <b>.js</b> extension.
2.	<b>ActionScript</b> It is an object oriented programming language used for the development of websites and software targeting Adobe flash player.
3.	<b>Dart</b> It is an open source web programming language developed by Google. It relies on source-to-source compiler to JavaScript.
4.	<b>VBScript</b> It is an open source web programming language developed by Microsoft. It is superset of JavaScript and adds optional static typing class-based object oriented programming.

### Server-side Scripting

**Sever-side scripting** acts as an interface for the client and also limit the user access the resources on web server. It can also collect the user's characteristics in order to customize response.



Following table describes commonly used Server-Side scripting languages:

S.N.	Scripting Language Description
1.	<b>ASP</b> Active Server Pages (ASP) is server-side script engine to create dynamic web pages. It supports <b>Component Object Model (COM)</b> which enables ASP web sites to access functionality of libraries such as DLL.
2.	<b>ActiveVFP</b> It is similar to PHP and also used for creating dynamic web pages. It uses native <b>Visual Foxpro</b> language and database.
3.	<b>ASP.net</b> It is used to develop dynamic websites, web applications, and web services.
4.	<b>Java</b> Java Server Pages are used for creating dynamic web applications. The Java code is compiled into byte code and run by <b>Java Virtual Machine (JVM)</b> .
5.	<b>Python</b>



	It supports multiple programming paradigms such as object-oriented, and functional programming. It can also be used as non-scripting language using third party tools such as <b>Py2exe</b> or <b>Pyinstaller</b> .
6.	<b>WebDNA</b> It is also a server-side scripting language with an embedded database system.

## Web Browser

### Web Browser

**web Browser** is an application software that allows us to view and explore information on the web. User can request for any web page by just entering a URL into address bar.

Web browser can show text, audio, video, animation and more. It is the responsibility of a web browser to interpret text and commands contained in the web page.

Earlier the web browsers were text-based while now a days graphical-based or voice-based web browsers are also available. Following are the most common web browser available today:

Browser	Vendor
Internet Explorer	Microsoft
Google Chrome	Google
Mozilla Firefox	Mozilla
Netscape Navigator	Netscape Communications Corp.
Opera	Opera Software
Safari	Apple
Sea Monkey	Mozilla Foundation
K-meleon	K-meleon

### Architecture

There are a lot of web browsers available in the market. All of them interpret and display information on the screen however their capabilities and structure varies depending upon implementation. But the most basic component that all web browsers must exhibit are listed below:

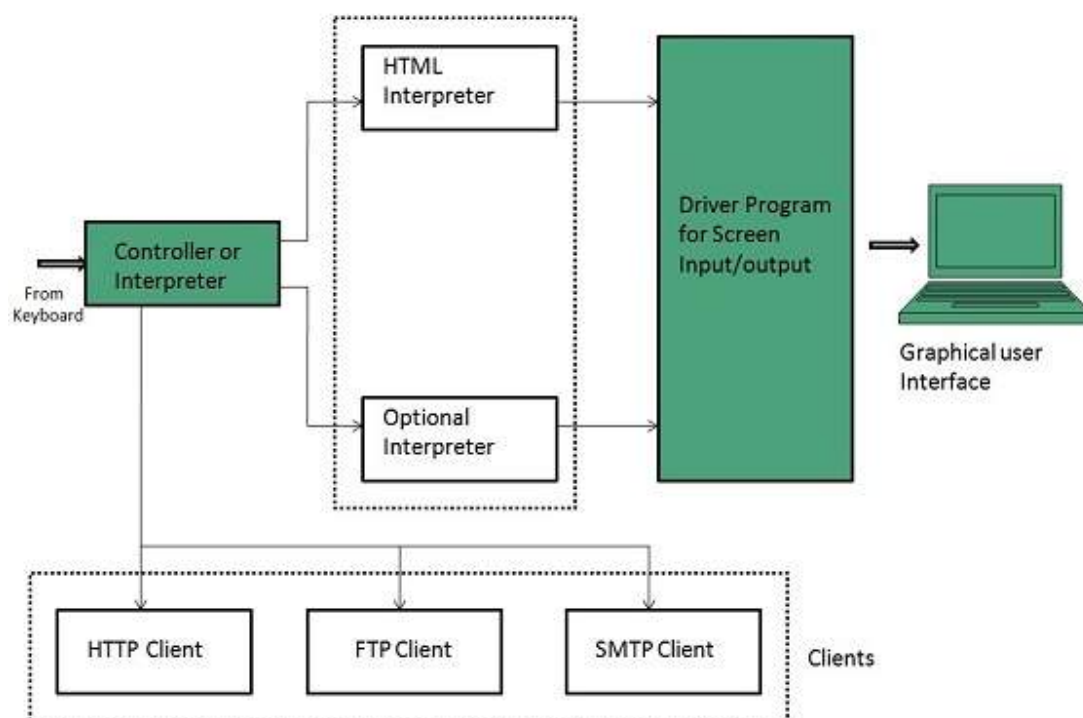
- Controller/Dispatcher
- Interpreter
- Client Programs

**Controller** works as a control unit in CPU. It takes input from the keyboard or mouse, interpret it and make other services to work on the basis of input it receives.

**Interpreter** receives the information from the controller and execute the instruction line by line. Some interpreter are mandatory while some are optional For example, HTML interpreter program is mandatory and java interpreter is optional.

**Client Program** describes the specific protocol that will be used to access a particular service. Following are the client programs tat are commonly used:

- HTTP
- SMTP
- FTP
- NNTP
- POP



# Web Server

## Overview

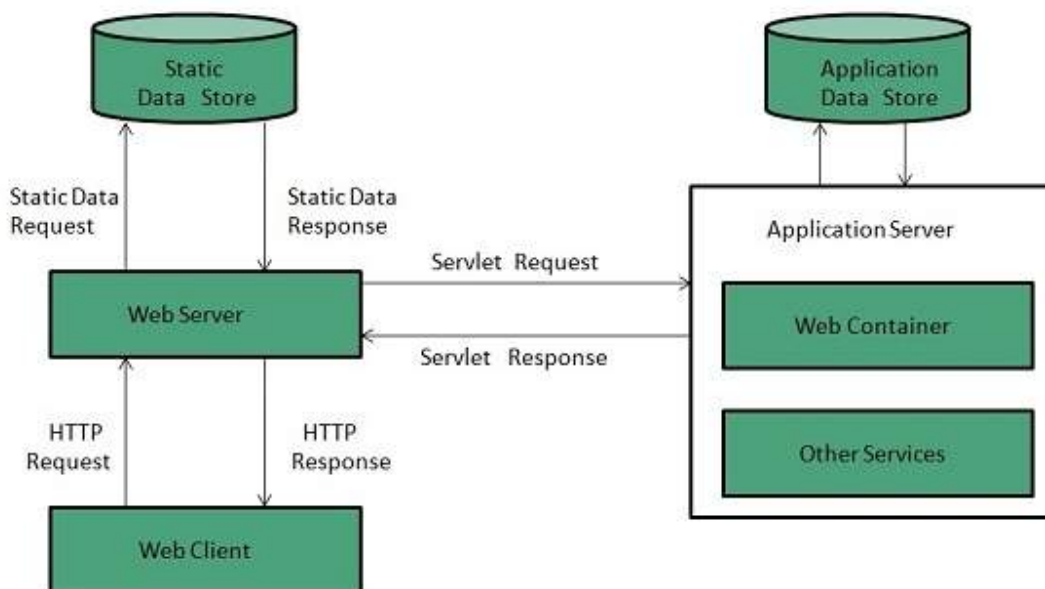
**Web server** is a computer where the web content is stored. Basically web server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

Web site is collection of web pages while web server is a software that respond to the request for web resources.

## Web Server Working

Web server respond to the client request in either of the following two ways:

- Sending the file to the client associated with the requested URL.
- Generating response by invoking a script and communicating with database



## Key Points

- When client sends request for a web page, the web server search for the requested page if requested page is found then it will send it to client with an HTTP response.
- If the requested web page is not found, web server will the send an **HTTP response:Error 404 Not found.**
- If client has requested for some other resources then the web server will contact to the application server and data store to construct the HTTP response.

## Architecture

Web Server Architecture follows the following two approaches:

1. Concurrent Approach
2. Single-Process-Event-Driven Approach.

### Concurrent Approach

Concurrent approach allows the web server to handle multiple client requests at the same time. It can be achieved by following methods:

- Multi-process
- Multi-threaded
- Hybrid method.

### Multi-processing

In this a single process (parent process) initiates several single-threaded child processes and distribute incoming requests to these child processes. Each of the child processes are responsible for handling single request.

It is the responsibility of parent process to monitor the load and decide if processes should be killed or forked.

### Multi-threaded

Unlike Multi-process, it creates multiple single-threaded process.

### Hybrid

It is combination of above two approaches. In this approach multiple process are created and each process initiates multiple threads. Each of the threads handles one connection. Using multiple threads in single process results in less load on system resources.

Following table describes the most leading web servers available today:

S.N.	Web Server Descriptino
1	<b>Apache HTTP Server</b> This is the most popular web server in the world developed by the Apache Software Foundation. Apache web server is an open source software and can be installed on almost all operating systems including Linux, UNIX, Windows, FreeBSD, Mac OS X and more. About 60% of the web server machines run the Apache Web Server.
2.	<b>Internet Information Services (IIS)</b> The Internet Information Server (IIS) is a high performance Web Server from Microsoft. This web server runs on Windows NT/2000 and 2003 platforms (and may be on upcoming new Windows version also). IIS comes bundled with Windows NT/2000 and 2003; Because IIS is tightly integrated with the operating system so it is relatively easy to administer it.
3.	<b>Lighttpd</b> The lighttpd, pronounced lighty 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. Lighttpd can also run on Windows, Mac OS X, Linux and Solaris operating systems.
4.	<b>Sun Java System Web Server</b> This web server from Sun Microsystems is suited for medium and large web sites. Though the server is free it is not open source. It however, runs on 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, and Ruby on Rails, ASP and Coldfusion etc.
5.	<b>Jigsaw Server</b> 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, and Mac OS X Free BSD etc. Jigsaw has been written in Java and can run CGI scripts and PHP programs.

## Client Server environment

**Client-server architecture**, architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service. Clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform. In hospital data processing, for example, a client computer can be running an application program for entering patient information while the server computer is running another program that manages the database in which the information is permanently stored. Many clients can access the server's information simultaneously, and, at the same time, a client computer can perform other tasks, such as sending e-mail. Because both client and server computers are considered independent devices, the client-server model is completely different from the old mainframe model, in which a centralized mainframe computer performed all the tasks for its associated "dumb" terminals, which merely communicated with the central mainframe.

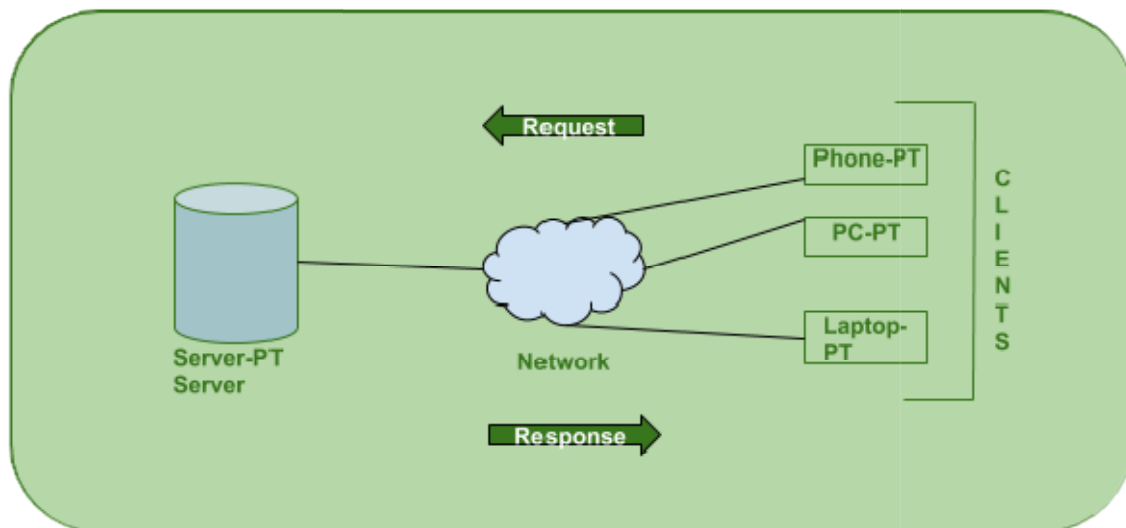
<https://www.geeksforgeeks.org/client-server-model/>

The Client-server model is a distributed application structure that partitions task or workload between the providers of a resource or service, called servers, and service requesters called clients. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and deliver the data packets requested back to the client. Clients do not share any of their resources. Examples of Client-Server Model are Email, World Wide Web, etc.

### How the Client-Server Model works ?

- **Client:** When we talk the word **Client**, it mean to talk of a person or an organization using a particular service. Similarly in the digital world a **Client** is a computer (**Host**) i.e. capable of receiving information or using a particular service from the service providers (**Servers**).
- **Servers:** Similarly, when we talk the word **Servers**, It mean a person or medium that serves something. Similarly in this digital world a **Server** is a remote computer which provides information (data) or access to particular services.

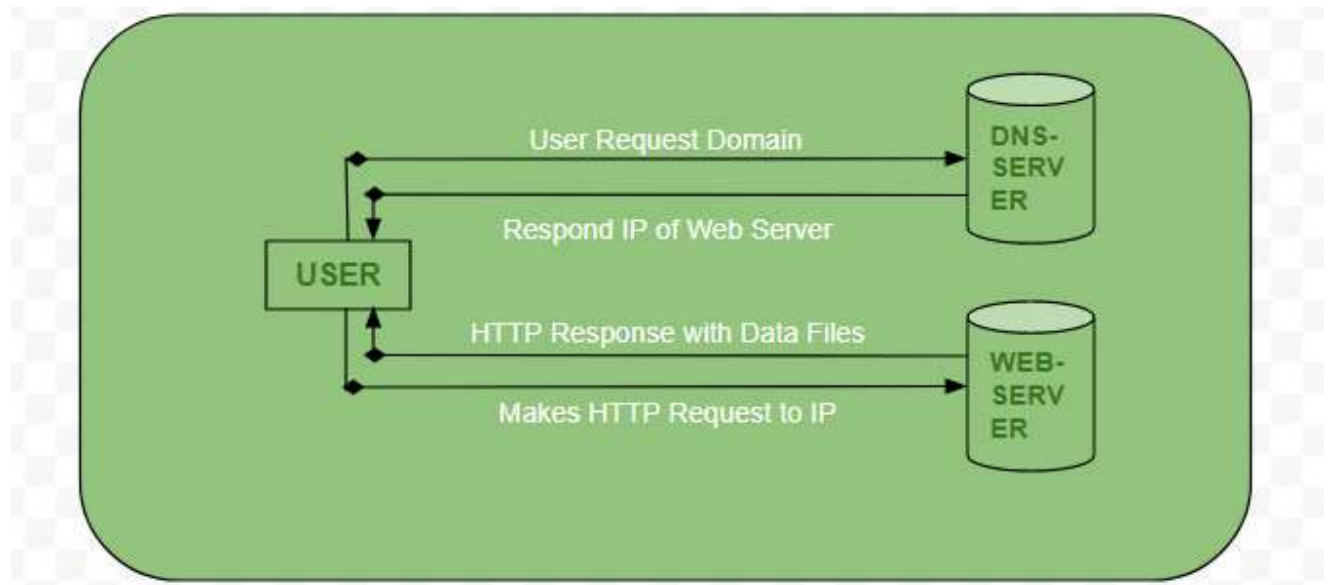
So, its basically the **Client** requesting something and the **Server** serving it as long as its present in the database.



### How the browser interacts with the servers ?

There are few steps to follow to interact with the servers a client.

- User enters the **URL**(Uniform Resource Locator) of the website or file. The Browser then requests the **DNS**(DOMAIN NAME SYSTEM) Server.
- **DNS Server** lookup for the address of the **WEB Server**.
- **DNS Server** responds with the **IP address** of the **WEB Server**.
- Browser sends over an **HTTP/HTTPS** request to **WEB Server's IP** (provided by **DNS server**).
- Server sends over the necessary files of the website.
- Browser then renders the files and the website is displayed. This rendering is done with the help of **DOM** (Document Object Model) interpreter, **CSS** interpreter and **JS Engine** collectively known as the **JIT** or (Just in Time) Compilers.



**Advantages of Client-Server model:**

- Centralized system with all data in a single place.
- Cost efficient requires less maintenance cost and Data recovery is possible.
- The capacity of the Client and Servers can be changed separately.

**Disadvantages of Client-Server model:**

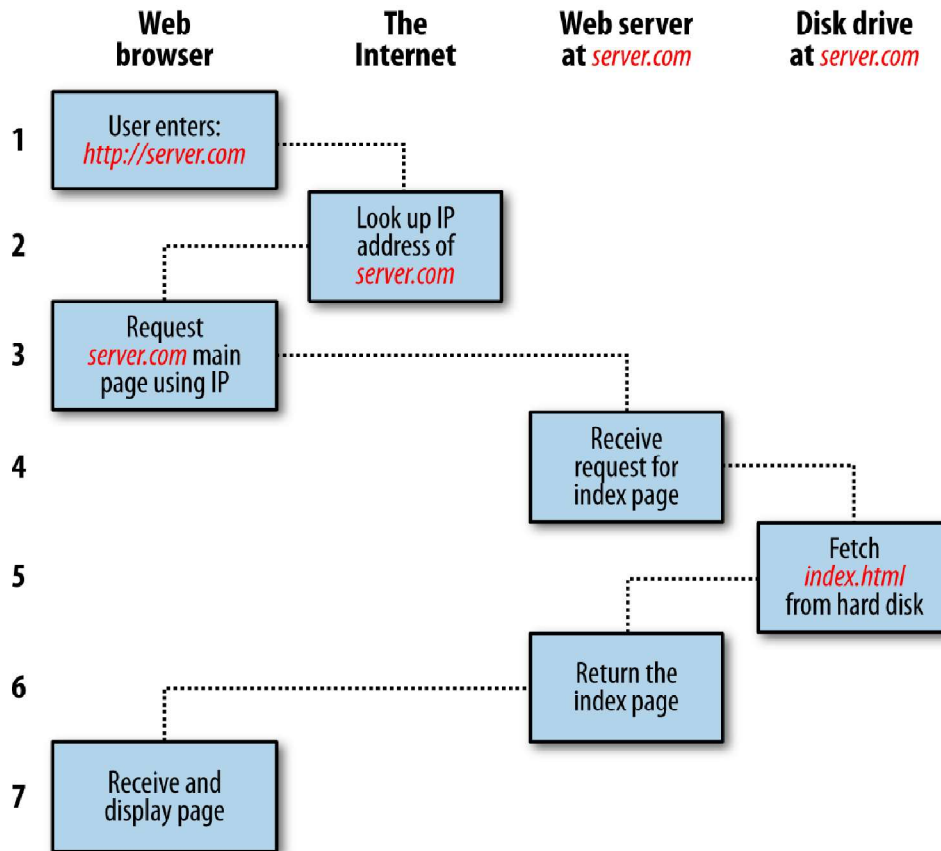
- Clients are prone to viruses, Trojans and worms if present in the Server or uploaded into the Server.
- Server are prone to Denial of Service (DOS) attacks.
- Data packets may be spoofed or modified during transmission.
- Phishing or capturing login credentials or other useful information of the user are common and MITM(Man in the Middle) attacks are common.



## **Text Book Content**

### **The Request/Response Procedure**

At its most basic level, the request/response process consists of a web browser asking the web server to send it a web page and the server sending back the page. The browser then takes care of displaying the page (see **Figure 1-1**).



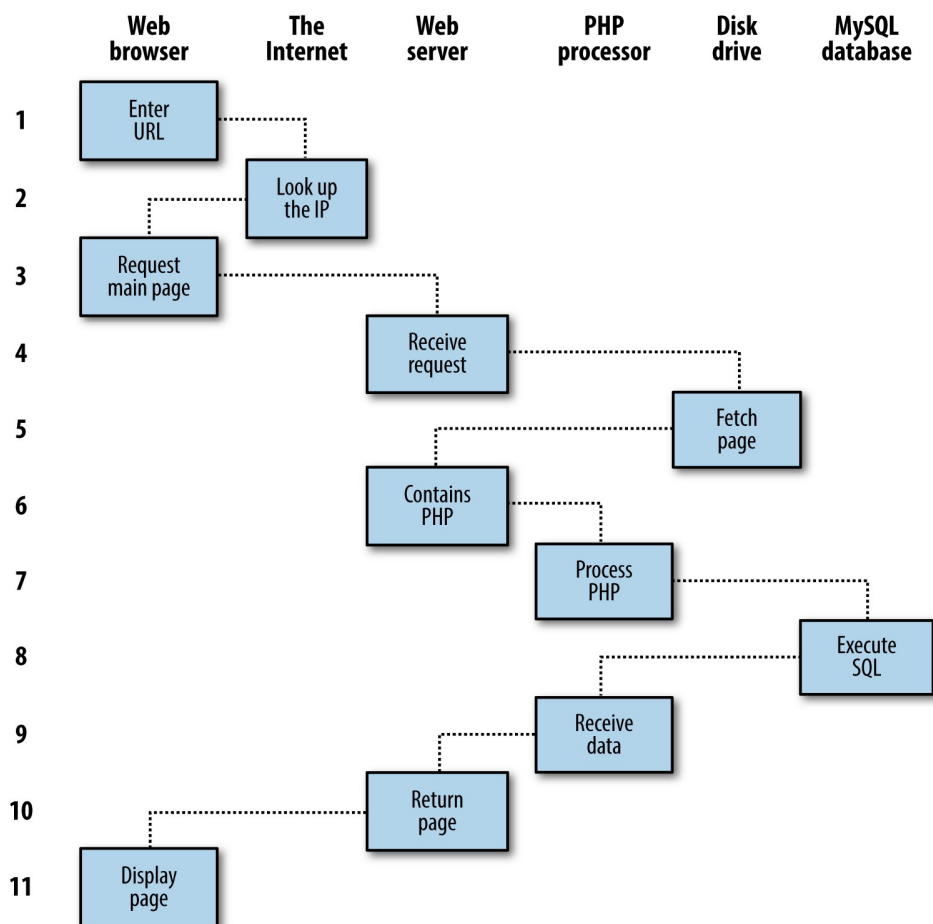
**Figure 1-1. The basic client/server request/response sequence**

Each step in the request and response sequence is as follows:

1. You enter *http://server.com* into your browser's address bar.
2. Your browser looks up the IP address for *server.com*.
3. Your browser issues a request for the home page at *server.com*.
4. The request crosses the Internet and arrives at the *server.com* web server.
5. The web server, having received the request, looks for the web page on its hard disk.
6. The web page is retrieved by the server and returned to the browser.
7. Your browser displays the web page.

For an average web page, this process takes place once for each object within the page:

a graphic, an embedded video or Flash file, and even a CSS template. In step 2, notice that the browser looked up the IP address of *server.com*. Every machine attached to the Internet has an IP address—your computer included. But we generally access web servers by name, such as *google.com*. As you probably know, the browser consults an additional Internet service called the Domain Name Service (DNS) to find its associated IP address and then uses it to communicate with the computer. For dynamic web pages, the procedure is a little more involved, because it may bring both PHP and MySQL into the mix (see **Figure 1-2**).



**Figure 1-2. A dynamic client/server request/response sequence**

Here are the steps for a dynamic client/server request/response sequence:

1. You enter `http://server.com` into your browser's address bar.
2. Your browser looks up the IP address for `server.com`.
3. Your browser issues a request to that address for the web server's home page.
4. The request crosses the Internet and arrives at the `server.com` web server.
5. The web server, having received the request, fetches the home page from its harddisk.
6. With the home page now in memory, the web server notices that it is a file incorporating
7. The PHP interpreter executes the PHP code.
8. Some of the PHP contains MySQL statements, which the PHP interpreter now passes to the MySQL database engine.
9. The MySQL database returns the results of the statements back to the PHP interpreter.
10. The PHP interpreter returns the results of the executed PHP code, along with the results from the MySQL database, to the web server.
11. The web server returns the page to the requesting client, which displays it. Although it's helpful to be aware of this process so that you know how the three elements work together, in practice you don't really need to concern yourself with these details, because they all happen automatically.

HTML pages returned to the browser in each example may well contain JavaScript, which will be interpreted locally by the client, and which could initiate another request— the same way embedded objects such as images would.