Scripts:

- A set of instructions or code written in a programming language.
- Executed by an interpreter or a scripting engine.
- Used for automation and specific tasks such as system administration, data manipulation, or web page scripting.
- Smaller and simpler than programs.
- Focused on specific tasks.
- Can be written in the same programming languages as programs.
- Generally require less resources to run than programs.

Programs:

- A more complex set of instructions that typically includes multiple scripts and other resources.
- Usually compiled and run as standalone applications on a computer or other electronic device.
- Can be larger and more complex than scripts.
- Used for a variety of tasks depending on the specific application.
- Can be written in the same programming languages as scripts.
- May require significant resources to run.

Applications of Scripting Languages :

1. To automate certain tasks in a program
2. Extracting information from a data set
3. Less code intensive as compared to traditional programming languages

Applications of Programming Languages :

1. They typically run inside a parent program like scripts
2. More compatible while integrating code with mathematical models

3. Languages like JAVA can be compiled and then used on any platform

Let us see the differences in a tabular form -:

	Scripting Language	Programming Language
1.	A scripting language is a language that uses a naive method to bring codes to a runtime environment	A Programming language is a language which is used by humans to navigate their communication with computers.
2.	These are made for a particular runtime environment.	Programming languages are of three types -: • low-level Programming language • Middle-level Programming language • High-level Programming language
3.	They are used to create dynamic web applications	Programming languages are used to write computer programs.
4.	Scripting languages contain different libraries	They are high-speed languages.
5.	Example -: Bash, Ruby, Python, JavaScript etc.	Example -: C++, Java, PHP High-level etc.
6.	Scripting languages can be easily ported among various operating systems.	Programming languages are translation free languages
7.	These languages requires a host.	These languages are self executable.
8.	Do not create a *.exe* file.	These generate *.exe* files.

↳ executables (windows OS)

9.	Most of the scripting languages are interpreted language.	Most of the programming languages are compiled languages.
10.	All the scripting languages are programming languages.	All the programming languages are not scripting languages.
11.	It is easier to learn than programming language.	It can take significant amount of time to learn.
12.	It is less code intensive when compared with programming language.	It is code intensive.
13.	It does not create any binary files.	It does creates binary files.
14.	It is easy for the beginner to write and understand the code.	It is difficult for the beginner to write and understand the code.
15.	It is run inside another program.	It is independently run.
16.	It needs lesser line of codes.	It needs numerous lines of code.
17.	It has low maintenance cost.	It has high maintenance cost.

the origin of scripting:

- The origin of scripting can be traced back to the early days of computing.
- Initially, large programming languages such as Fortran and COBOL were used to create programs for mainframe computers.
- As computers became more powerful and accessible, there was a need for a simpler and more flexible way to automate tasks and perform routine operations.
- The development of scripting languages allowed users to

write small programs or scripts that could be executed quickly and easily.
- The Unix shell script was one of the earliest scripting languages, developed in the 1970s.
- Other popular scripting languages include Perl, Python, Ruby, and JavaScript.
- Scripting is an essential part of many computing environments today, and is used to automate tasks in a wide variety of industries.

Characteristics of Scripting Languages

Scripting languages have several characteristics that differentiate them from other programming languages. Here are some key characteristics of scripting languages:

- **Interpreted:** Scripting languages are usually interpreted rather than compiled. This means that the code is executed by an interpreter at runtime, rather than being compiled into machine code beforehand.

- **Dynamic Typing:** Scripting languages typically use dynamic typing, which means that the type of a variable is determined at runtime, rather than being declared explicitly.

- **High-level:** Scripting languages are typically high-level languages, which means that they are designed to be easy for humans to read and write, and abstract away many of the details of low-level programming.

- **Easy to Learn:** Scripting languages are usually easier to learn and use than compiled languages, making them popular among beginners and experienced developers alike.

- **Rapid Development:** Scripting languages are often used for rapid prototyping and development, as they allow developers to quickly write and test code.

- **Built-in Functions:** Scripting languages often come with a large number of built-in functions and libraries, which makes it easy to perform common tasks such as file I/O, string manipulation, and network communication.

- **Cross-platform:** Scripting languages are often cross-platform, meaning that they can run on multiple operating systems without requiring modification.

  Overall, scripting languages are designed to be flexible, easy to use, and quick to develop with, making them ideal for tasks such as automation, web development, and data analysis.

Uses for Scripting Languages

Scripting languages are used in a wide variety of applications and industries. Here are some common uses for scripting languages:

- **Automation:** Scripting languages are often used to automate repetitive or time-consuming tasks, such as system administration, file management, and data processing.

- **Web Development**: Scripting languages such as JavaScript, PHP, and Python are widely used in web development, for tasks such as dynamic content generation, form validation, and data visualization.

- **Game Development:** Scripting languages such as Lua are popular in game development, where they are used to create game logic, user interfaces, and artificial intelligence.

- **Data Analysis:** Scripting languages such as Python and R are widely used in data analysis and scientific computing, for tasks such as data manipulation, statistical analysis, and visualization.

- **Testing and Debugging:** Scripting languages are often used for testing and debugging software, as they allow developers to automate repetitive testing tasks and easily identify and fix errors.

- **Graphics and Multimedia:** Scripting languages such as Processing and JavaScript are commonly used in graphics and multimedia applications, for tasks such as animation, video editing, and image processing.

- **Cross-platform Development:** Scripting languages are often used for cross-platform development, as they can run on multiple operating systems without requiring modification.

    Overall, scripting languages are versatile and flexible tools that can be used in a wide variety of applications, making them popular among developers in many different industries.
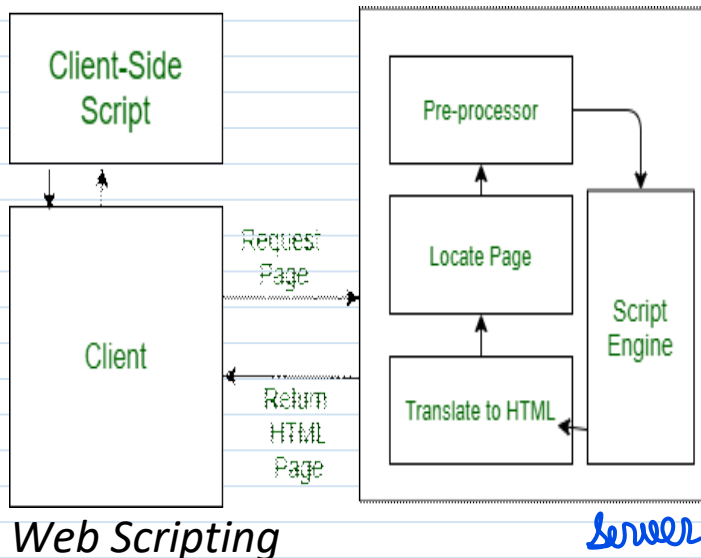
web-scripting

The process of creating and embedding scripts in a web page is known as **web-scripting**. A script or a computer-script is a list of commands that are embedded in a web-page normally and are interpreted and executed by a certain program or scripting engine.

- Scripts may be written for a variety of purposes such as for automating processes on a local-computer or to generate web pages.
- The programming languages in which scripts are written are called scripting language, there are many scripting languages available today.
- Common scripting languages

are [VBScript](), [JavaScript](), [ASP](), [PHP](), [PERL](), [JSP]() etc.

Types of Script :

Scripts are broadly of following two type :

Web Scripting   *Server*

Client-Side Scripts :

1. Client-side scripting is responsible for interaction within a web page. The client-side scripts are firstly downloaded at the client-end and then interpreted and executed by the browser (default browser of the system).

2. The client-side scripting is browser-dependent. i.e., the client-side browser must be scripting enables in order to run scripts

3. Client-side scripting is used when the client-side interaction is used. Some example uses of client-side scripting may be :
   - To get the data from user's screen or browser.
   - For playing online games.
   - Customizing the display of page in browser without reloading or reopening the page.

4. Here are some popular client-side scripting languages VBScript, JavaScript, Hypertext Processor(PHP)

Server-Side Scripts :

1. Server-side scripting is responsible for the completion or carrying out a task at the server-end and then sending the result to the client-end.

2. In server-side script, it doesn't matter which browser is being used at client-end, because the server does all the work.
3. Server-side scripting is mainly used when the information is sent to a server and to be processed at the server-end. Some sample uses of server-scripting can be :
   - Password Protection.
   - Browser Customization (sending information as per the requirements of client-end browser)
   - Form Processing
   - Building/Creating and displaying pages created from a database.
   - Dynamically editing changing or adding content to a web-page.
4. Here are some popular server-side scripting languages PHP, Perl, ASP (Active Server Pages), JSP (Java Server Pages).
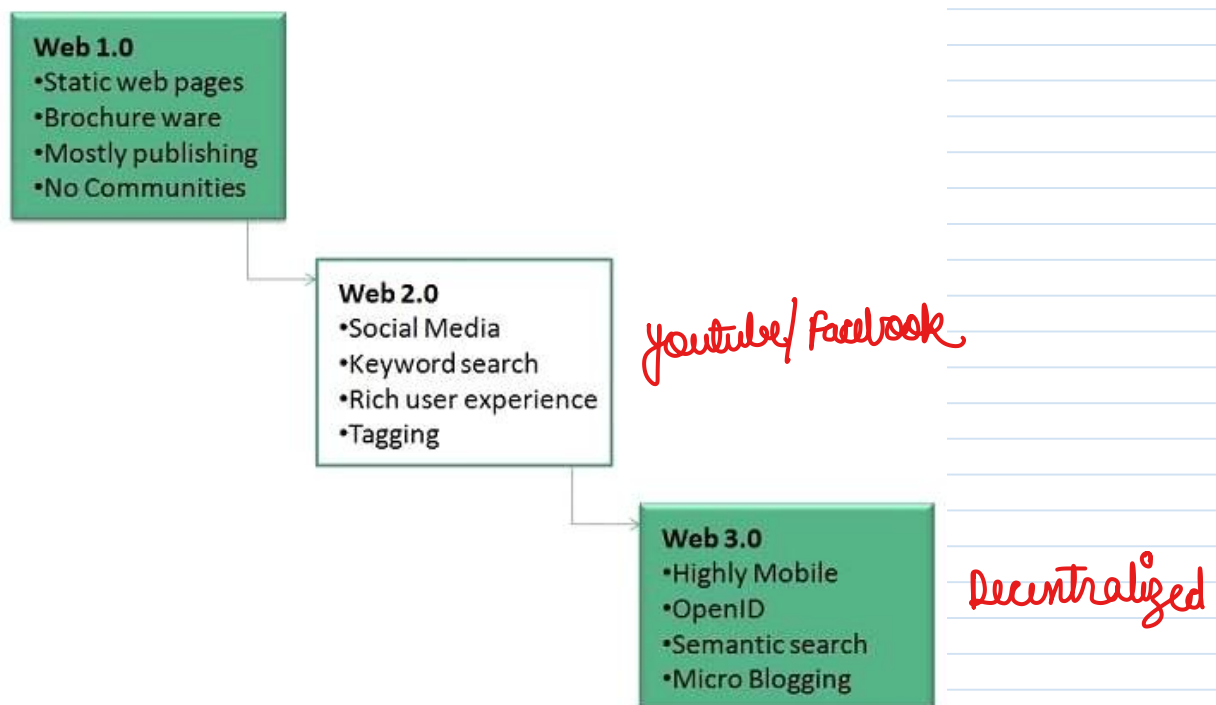
World Wide Web

1. The World Wide Web (WWW or Web) is an information system on the internet that allows users to access and share data and resources using web browsers.
2. The Web is based on a client-server architecture, where web browsers act as clients that request and receive data from web servers over the internet.
3. The Web consists of a vast collection of interconnected hypertext documents and other resources, linked by hyperlinks and URLs (Uniform Resource Locators).
4. The Web was created in 1989 by Tim Berners-Lee while working at CERN, the European Organization for Nuclear Research.
5. Web pages are typically written in HTML (Hypertext Markup Language), a markup language that defines the structure and content of web

pages.

6. The Web is organized into a hierarchical system of domains, each of which is identified by a unique domain name.
7. Web pages are linked together using hyperlinks, which allow users to navigate from one page to another by clicking on links.
8. The Web also uses other technologies such as CSS (Cascading Style Sheets) for styling web pages, JavaScript for client-side scripting, and various web development frameworks and content management systems (CMS) to create and manage websites.
9. The Web has become an essential tool for communication, commerce, entertainment, and education.
10. The Web has revolutionized many aspects of modern life and is continuously evolving to meet the changing needs of users and businesses.

The following diagram briefly defines evolution of World Wide Web:

Web 1.0
•Static web pages
•Brochure ware
•Mostly publishing
•No Communities

Web 2.0
•Social Media
•Keyword search
•Rich user experience
•Tagging

youtube/ facebook

Web 3.0
•Highly Mobile
•OpenID
•Semantic search
•Micro Blogging

Decentralized

WWW Operation

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

1. User enters the URL (say, **http://www.tutorialspoint.com**) 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.
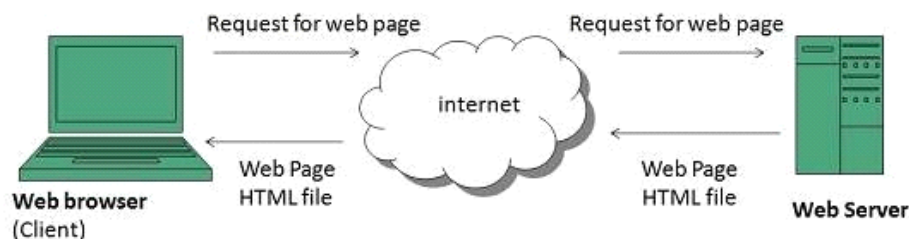
What is the Internet?

The Internet is a global network that connects computer networks all over the globe. Copper lines, wireless networks, and fiber optic cables are used to connect the devices. For data communication, it employs the TCP/IP protocol. Computers, routers, hubs, switches, repeaters, and other devices are all part of this network. To identify a device on the

internet, it has a unique IP address.
It facilitates the process of global
communication. A computer with an internet
connection is sufficient to connect you with
the person sitting miles away from your
location. A plethora of information is
disseminated on the web through the
internet.
Think of a games store, where the store is
your internet, and the games available in the
store are the websites from which you will get
your information
Services offered by the internet –

- File transfer
- E-mail
- Mobile apps
- Social media apps
- World wide web (www)

Difference between Internet and WWW :

S.No.	INTERNET	WWW
1	Internet is a global network of networks.	WWW stands for World wide Web.
2	Internet is a means of connecting a computer to any other computer anywhere in the world.	World Wide Web which is a collection of information which is accessed via the Internet.
3	Internet is infrastructure.	WWW is service on top of that infrastructure.

4	Internet can be viewed as a big book-store.	Web can be viewed as collection of books on that store.
5	At some advanced level, to understand we can think of the Internet as hardware.	At some advanced level, to understand we can think of the WWW as software.
6	Internet is primarily hardware-based.	WWW is more software-oriented as compared to the Internet.
7	It is originated sometimes in late 1960s.	English scientist Tim Berners-Lee invented the World Wide Web in 1989.
8	Internet is superset of WWW.	WWW is a subset of the Internet.
9	The first version of the Internet was known as ARPANET.	In the beginning WWW was known as NSFNET.
10	Internet uses IP address.	WWW uses HTTP.

HTTP

- 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.

Features of HTTP:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the

connection. The connection between client and server exist only during the current request and response time only.

- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME-type header.

- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.
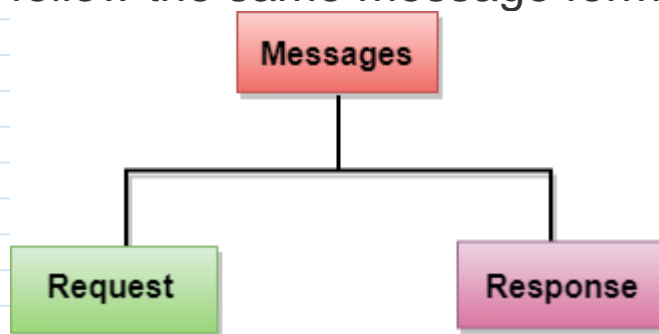
HTTP Transactions
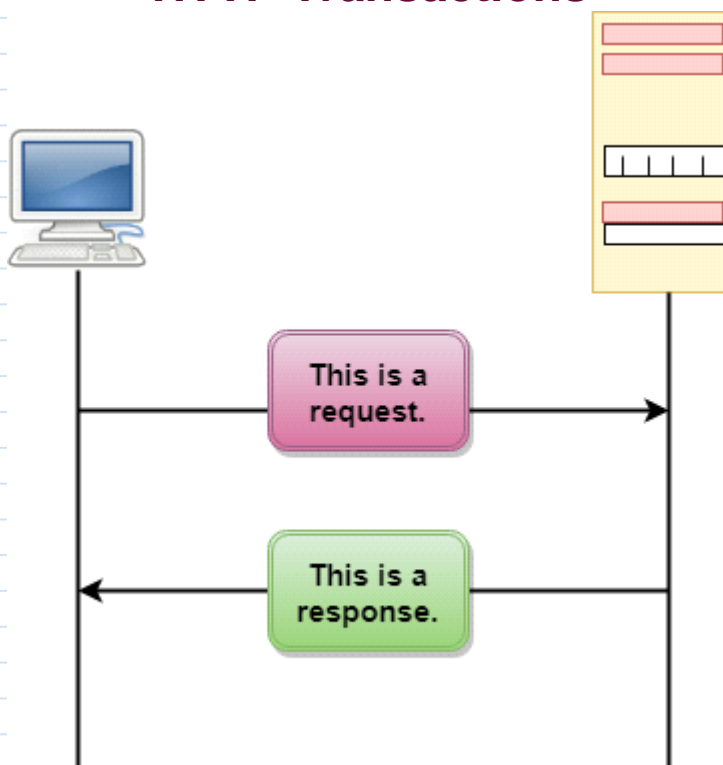
The above figure shows the HTTP transaction between client and server.

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.
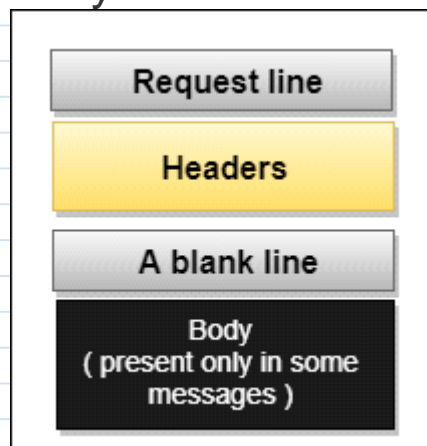
Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.
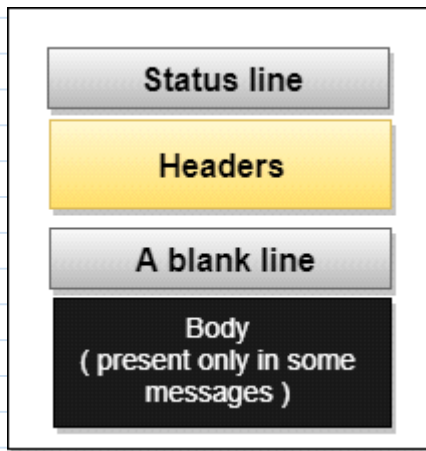
Uniform Resource Locator (URL)

- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.

- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the

computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.

- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contain slashes that separate the directories from the subdirectories and files.

What is the Web Browser?

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 for 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, and Safari are examples of web browsers.

How does a Web Browser Work?

A web browser helps us find information anywhere on the internet. It is installed on the client computer and requests information from the web server such a type of working model is called a client-server model.
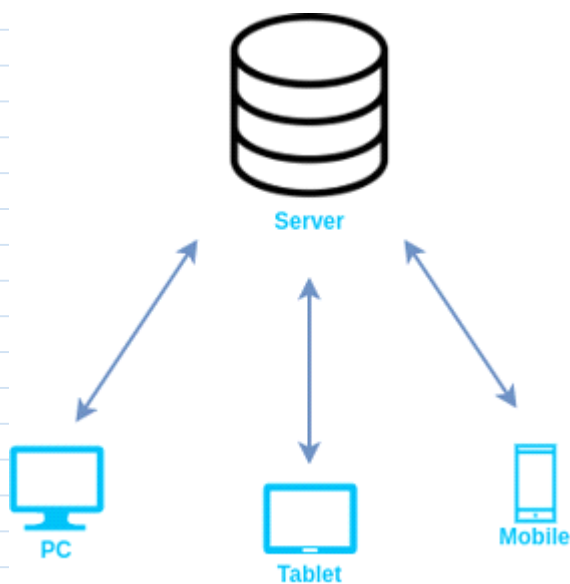
Client-server model

The browser receives information through HTTP protocol. In which transmission of data is defined. When the browser received data from the server, it is rendered in HTML to user-readable form and, information is displayed on the device screen.
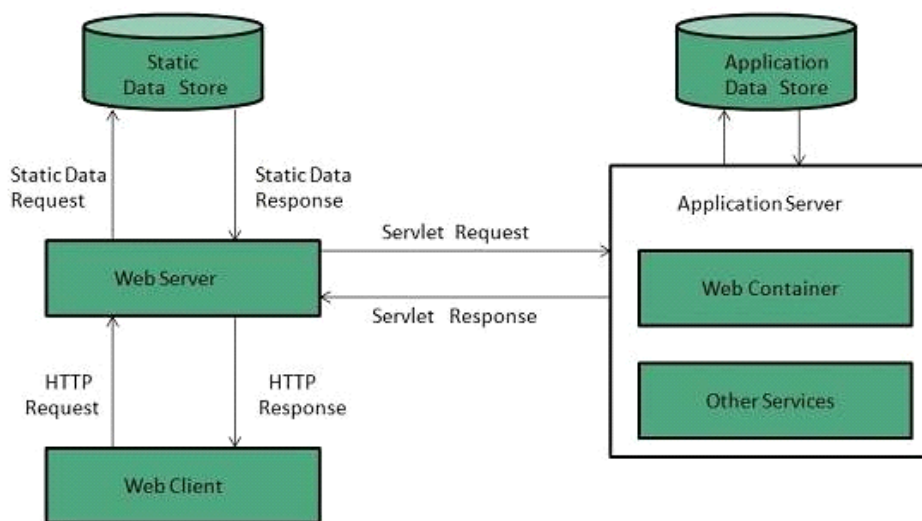
Web server

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 whileweb 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.

Factor	Web Browser	Web Server
Purpose	Web Browser is a software which is used to browse and display pages available over internet.	Web server is a software which provides these documents when requested by web browsers.
Function	A web browser sends request to server for web based documents and services.	Web server sees and approves those requests made by web browsers and sends the document in response

Process	Web browsers send HTTP Request and receive HTTP Response	Web servers receive HTTP Request and send HTTP Response.
Processing Model	Web browser has no processing model.	Web servers follow three major processing models: process based, thread based or hybrid.
Data Storage	Web browsers stores user data in cookies in local machine.	Web servers provide an area to store the website.
Installation	Web Browser is installed on user's machine.	Web servers can be installed anywhere but it need to be on a network or on local computer.
Acts as	Web browser acts as an interface between the web server and client.	Web servers act as the sender of web resources like web pages.
Responsibility	Web browser is responsible to request for a website or webpage located on the internet.	Web server is responsible for hosting websites, processing web requests, and sending the demanded document to the client.
Example	Examples of web browsers include Google Chrome, Internet explorer, Mozilla Firefox, etc.	Apache Server is an example of a web server.