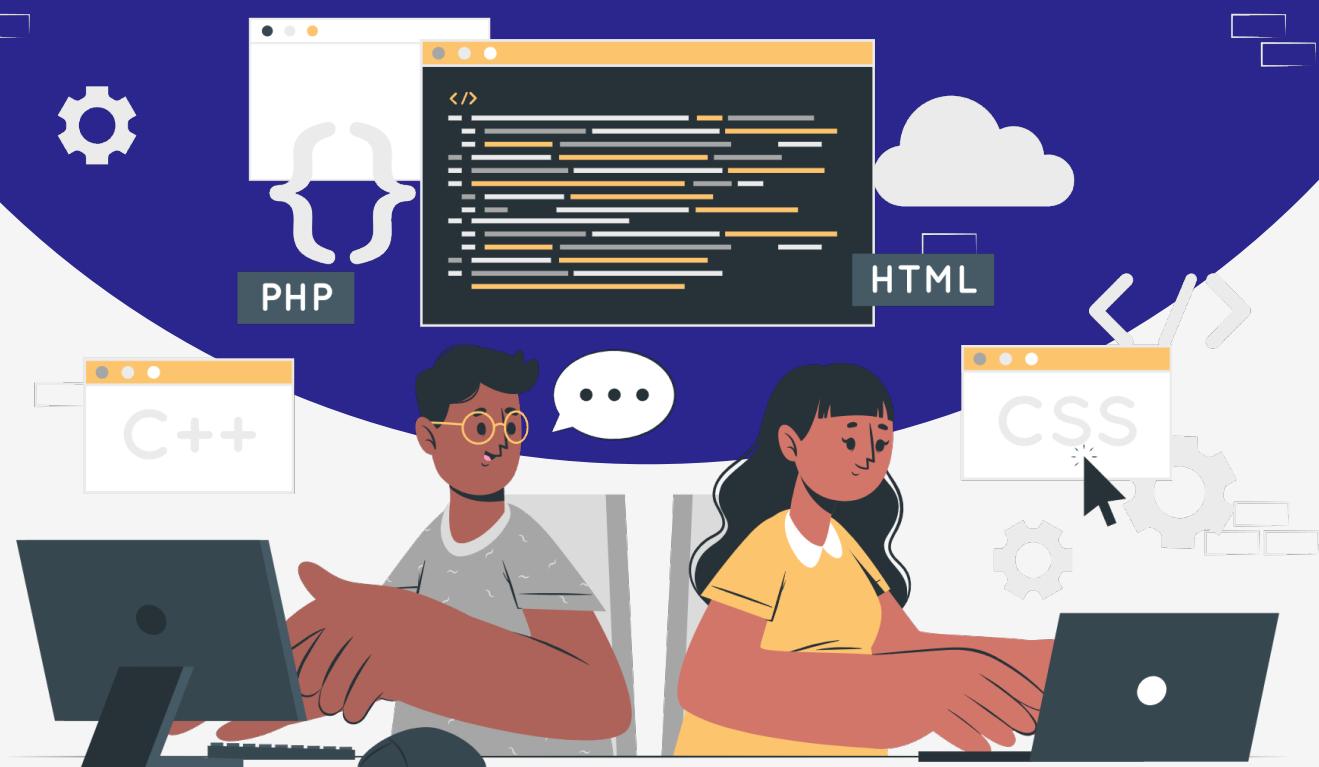


Lesson:

Fundamentals of Web Development



Topics to be covered in

1. Web Browser
2. Internet

Web Browser

Introduction to web browsers

A web browser is a software application that enables users to access and browse the World Wide Web. It provides a graphical user interface for displaying and interacting with web pages and allows users to navigate between pages, view multimedia content, and interact with web-based applications. There are a variety of web browsers available, each with its own unique features and capabilities.



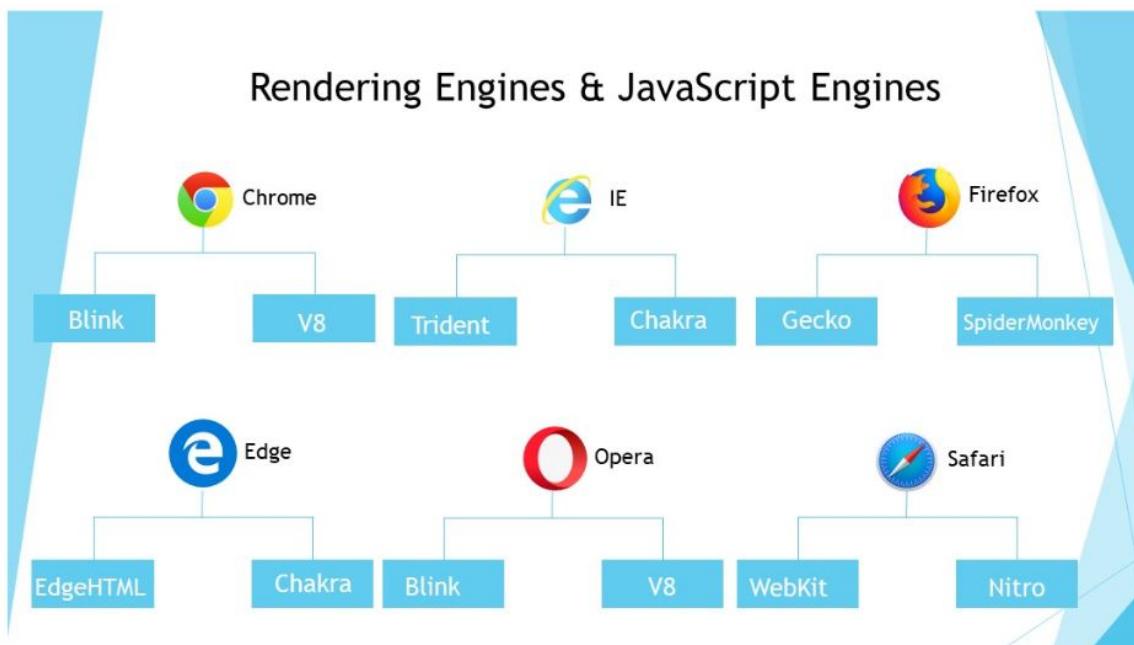
Features of web browsers

Web browsers offer a wide range of features and capabilities to enhance the user's browsing experience. Here are some common features found in most web browsers:

- **Tabbed browsing:** Most modern web browsers allow users to open multiple web pages in separate tabs within a single window, making it easier to switch between them.
- **Bookmarks:** Browsers allow users to save links to their favourite web pages for easy access later.
- **History:** Web browsers keep track of the user's browsing history, allowing them to quickly revisit sites they have previously visited.
- **Privacy and Security:** Browsers include features such as private browsing mode, which prevents the browser from saving any data about the user's browsing session. Additionally, browsers include security features such as phishing and malware protection.
- **Extensions and Plugins:** Browsers offer various extensions and plugins, allowing users to customize their browsers and add new features or functionality.
- **Developer Tools:** Most browsers include a suite of tools for web developers, including the ability to inspect and debug web pages, view page source code, and manipulate the Document Object Model (DOM).

Introduction to Browser Engine

A browser engine, also known as a rendering engine, is a software component that processes HTML, CSS, and JavaScript code to display web pages on a user's screen. When a user requests a web page, the browser engine retrieves the page's code from the server and processes it to create a visual representation of the page. Overall, the purpose of a browser engine is to enable users to browse the web and interact with web pages in a fast, efficient, and reliable manner, while also providing developers with the tools they need to build and maintain high-quality web applications.



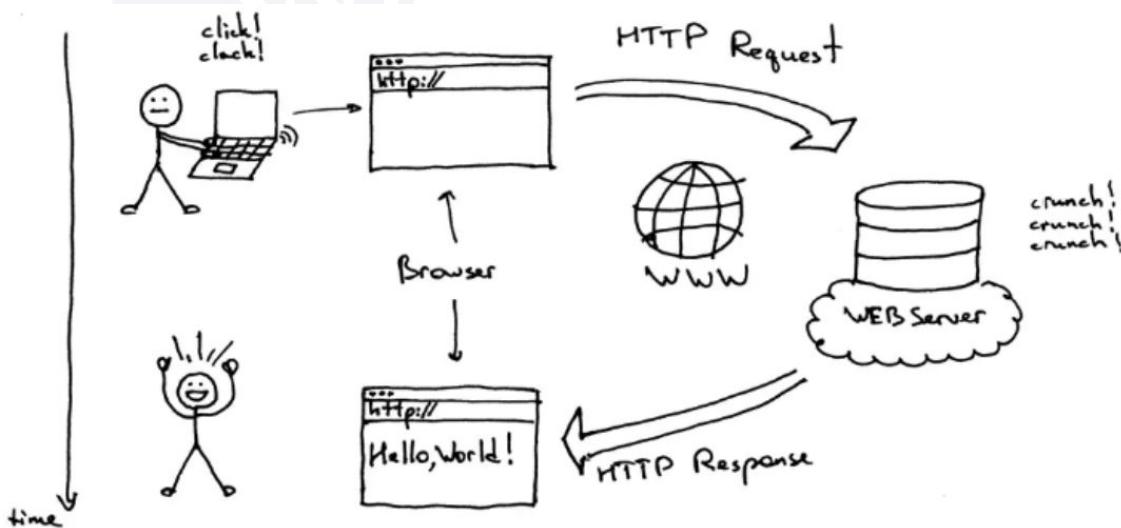
Some major responsibilities of the browser engine are as follows:-

- **Rendering web pages:** The primary purpose of a browser engine is to render web pages on a user's screen. When a user visits a web page, the browser engine processes the HTML, CSS, and JavaScript code to create a visual representation of the page. This process includes parsing the code, determining the layout of the page, and rendering the content on the screen.
- **Handling user interactions:** In addition to rendering web pages, browser engines are also responsible for handling user interactions with those pages. This includes responding to user clicks, handling form submissions, and updating the page in real time in response to user actions.
- **Optimizing page load times:** Browser engines play a critical role in optimizing page load times. They use techniques such as preloading and caching to ensure that pages load as quickly as possible. For example, a browser engine may preload resources that are likely to be needed next to reduce the amount of time it takes to load a page.

- **Supporting web standards:** Browser engines are responsible for supporting web standards and ensuring that web pages are displayed correctly across different browsers and devices. This includes implementing the latest HTML, CSS, and JavaScript standards, as well as supporting older legacy technologies that are still in use.
- **Providing developer tools:** Finally, browser engines provide a set of developer tools that allow web developers to debug and inspect web pages as they're being rendered. These tools can be used to identify and fix issues with web pages, optimize performance, and test for compatibility across different browsers and devices.



How does a browser display a website from a url ?



- The process of displaying a web page from url in a browser is followed by these steps –

1. Domain Name System (DNS)

- a. The browser begins by performing a DNS lookup to translate the website's domain name (e.g., www.pwskills.com) into an IP address.
- b. It sends a request to DNS servers to retrieve the IP address associated with the domain.

2. Establishing a TCP Connection

- a. The browser establishes a TCP (Transmission Control Protocol) connection with the web server identified by the IP address obtained from the DNS lookup.
- b. TCP ensures reliable communication by dividing data into packets and managing their transmission.

3. Sending an HTTP Request

- a. The browser sends an HTTP (Hypertext Transfer Protocol) request to the web server.
- b. The request includes the HTTP method (such as GET or POST) and the requested resource's URL.

4. Server Processing

- a. The web server receives the HTTP request and processes it.
- b. It may execute server-side code, access databases, or perform other necessary operations.

5. Generating the HTTP Response

- a. The web server generates an HTTP response, which includes the requested resource (such as an HTML file) and associated metadata.
- b. The response is typically encoded in HTML, but it can also include CSS, JavaScript, images, or other resources.

6. Transmitting the HTTP Response

- a. The web server sends the HTTP response back to the browser over the established TCP connection.
- b. The response is divided into packets, and the TCP protocol ensures their reliable delivery.

7. Rendering the HTML

- a. The browser receives the HTTP response and begins rendering the HTML content.
- b. It parses the HTML document from top to bottom, creating a Document Object Model (DOM) representation of the webpage's structure.

8. Fetching External Resources

- a. While parsing the HTML, the browser encounters external resources such as CSS stylesheets, JavaScript files, or images.
- b. It sends additional requests to retrieve these resources, allowing them to be rendered and applied to the webpage.

9. Executing JavaScript

- a. If the HTML includes JavaScript code, the browser executes it.
- b. JavaScript can dynamically modify the webpage's structure, content, and behavior.

10. Rendering the Webpage

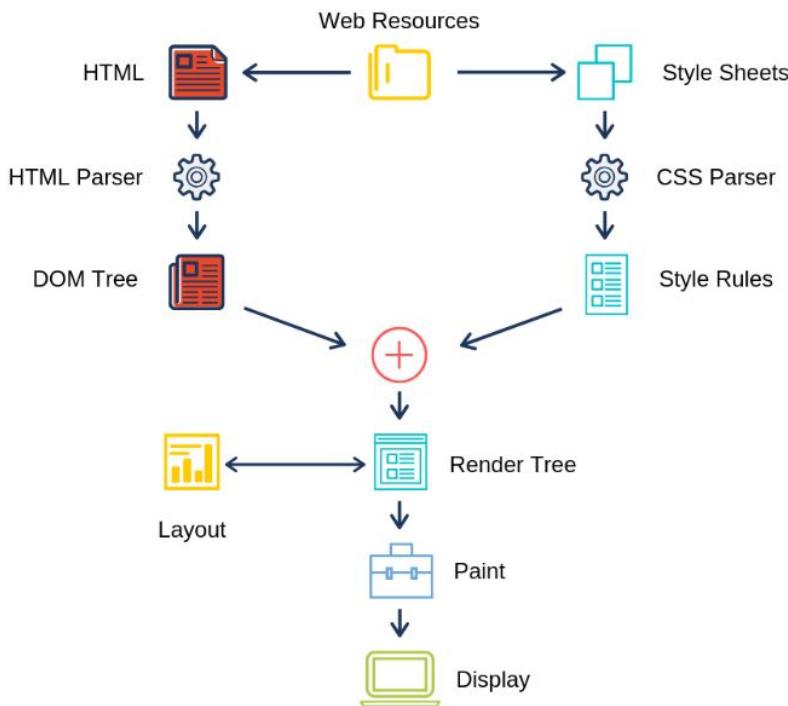
- a. The browser combines the parsed HTML, applied CSS styles, and executed JavaScript to render the webpage visually.
- b. The rendering process involves laying out elements, applying styles, and displaying the final result on the screen.

11. User Interaction and Event Handling

- a. The rendered web page becomes interactive, allowing users to interact with buttons, forms, links, and other elements.
- b. The browser handles user events (such as clicks or form submissions) and triggers appropriate actions.

12. Ongoing Interaction

- a. The browser maintains the established TCP connection with the web server for potential future requests or updates.
- b. It continues to process user interactions, handle additional HTTP requests, and update the webpage dynamically as needed.



Common browser compatibility issues

Here is a list of some common browser compatibility issues, which is generally faced by the users -

1. CSS rendering differences

Variation in how browsers interpret and display CSS styles, leading to differences in layout, positioning, and visual appearance of elements.

2. JavaScript Compatibility

Differences in JavaScript implementations and support for various ECMAScript versions across different browsers, resulting in inconsistent behavior or errors.

3. HTML Markup Interpretation

Variations in how browsers parse and interpret HTML markup, potentially causing differences in rendering and functionality.

4. Cross-Browser CSS3 Features

Limited or inconsistent support for CSS3 features, requiring fallback options or alternative approaches to ensure consistent rendering across different browsers.

Note: We can see the compatibility table of any properties or functions that can be used in different browsers on platforms like MDN.

Internet



Basic of internet

Some important terminologies



Network: A network is a collection of devices, such as computers, servers, printers, mobiles, and other hardware devices, that are connected to each other and can communicate and share resources with each other.

World Wide Web (WWW): The World Wide Web, also known as the Web, is a collection of websites and web pages that are accessible via the Internet. The Web is an important part of the internet and is the primary way that people access and share information online.

IP Address: An IP address is a unique identifier that is assigned to every device connected to the internet. IP addresses allow devices to communicate with each other over the internet and are essential for transmitting data between different networks.

Domain Name: A domain name is a human-readable address that is used to identify a website or web page on the internet. Domain names are easier to remember than IP addresses and are typically made up of a name and a top-level domain (TLD), such as .com, .org, or .net. For example pwskills.com, pw.live etc

URL: A Uniform Resource Locator (URL) is a string of characters that identifies the location of a web page or other online resource. URLs are used to navigate the Web and typically include a protocol (such as HTTP or HTTPS), a domain name, and a path to the resource.

Web Browser: A web browser is a software application that is used to access and display websites and web pages on the Internet. Popular web browsers include Google Chrome, Mozilla Firefox, and Microsoft Edge.

Server: A server is a computer or other device that stores and manages data and applications that are accessible over the internet. When a user requests a web page or other resource, the server sends the data to the user's device such as a laptop, mobile, etc.

HTTP and HTTPS: Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS) are protocols used for transmitting data over the Internet. HTTP is used for standard web traffic, while HTTPS is a secure version of HTTP that encrypts data to protect it from unauthorised access.

Search Engine: A search engine is a web-based tool that allows users to search for information on the internet. Popular search engines include Google, Bing, and Yahoo!

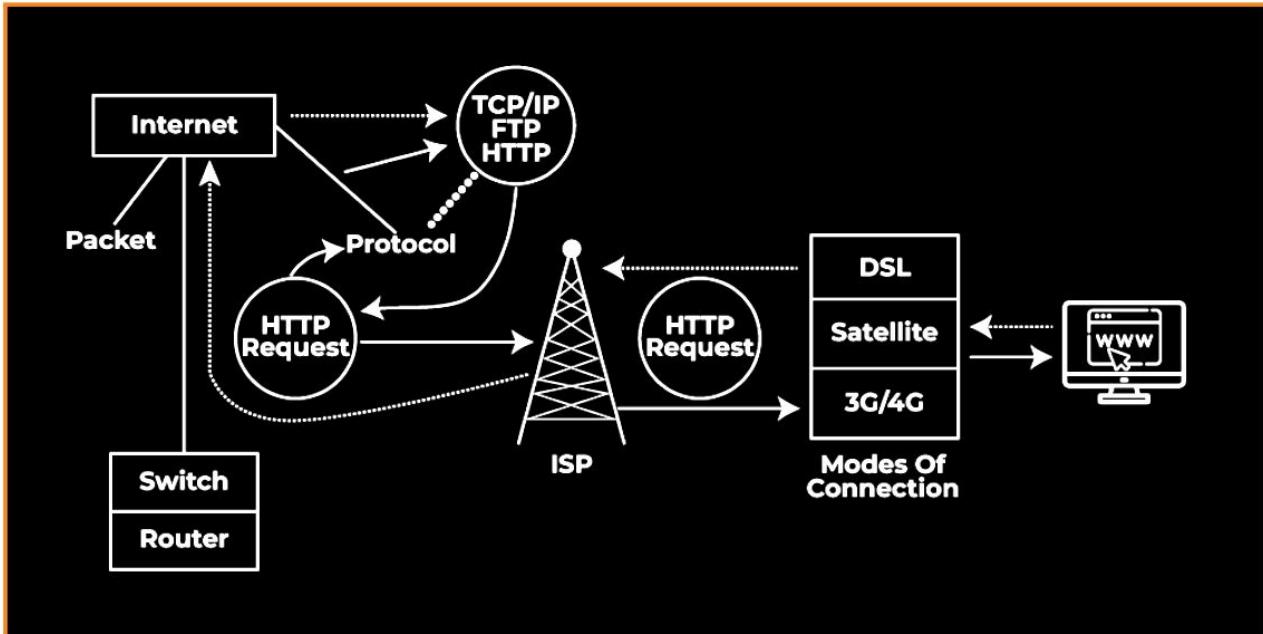
Introduction to internet

The Internet is a global network of interconnected computer networks that allows for communication and information exchange across the world. It is a vast and complex system of devices, computers, and servers that are linked together to enable the transfer of data, files, and other digital content. In simple terms, we can say that it is a network of networks.

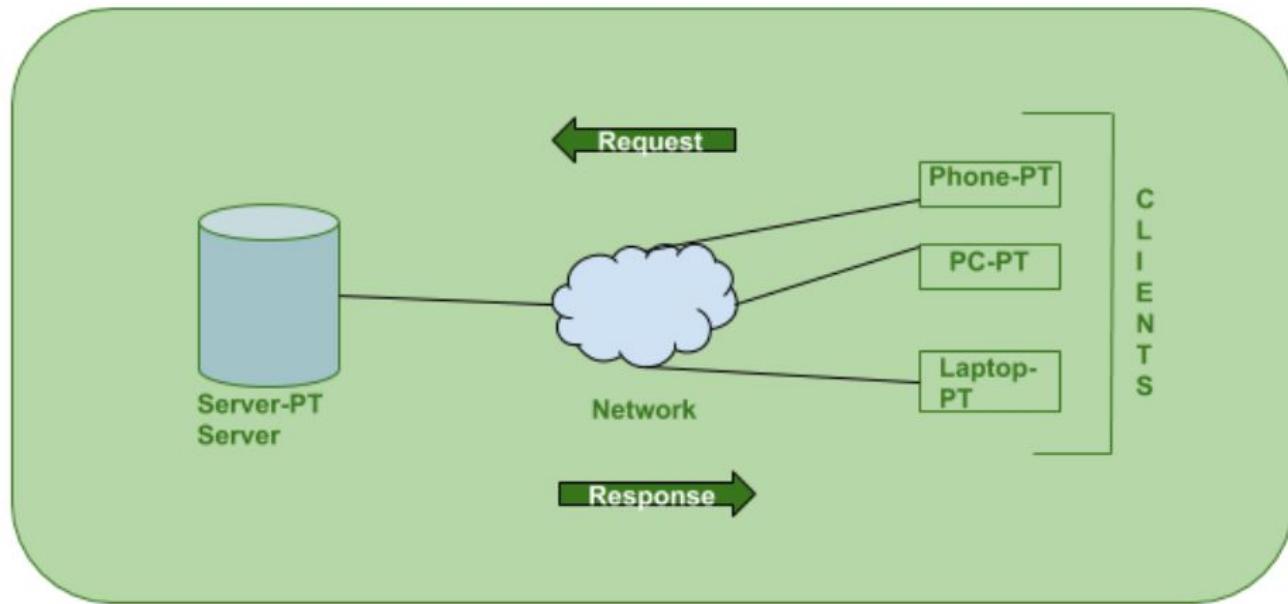
How does internet works

The workflow of the internet is described in the following steps, this is a simple explanation of the step-by-step procedure of working on the internet.

1. When you type a website address into your web browser, such as "www.pwskills.com", your computer sends a request to a server that hosts the website you want to visit.
2. The server responds to your request by sending the website's files back to your computer, which your web browser then uses to display the website's content.
3. This process of requesting and sending information happens through a system of connections and protocols that allow data to be transmitted between the client's devices and servers.
4. Data is broken up into small "packets" of information that are sent across the internet to their destination. Along the way, these packets travel through a series of routers that help them get to their final destination.
5. The Domain Name System (DNS) translates website addresses into IP addresses, which computers use to communicate with each other. For example, if we will type <https://pwskills.com/> then it will be translated as 76.76.21.21
6. Internet Service Providers (ISPs) provide users with access to the Internet through various connection methods such as cable, fiber, or wireless signals.

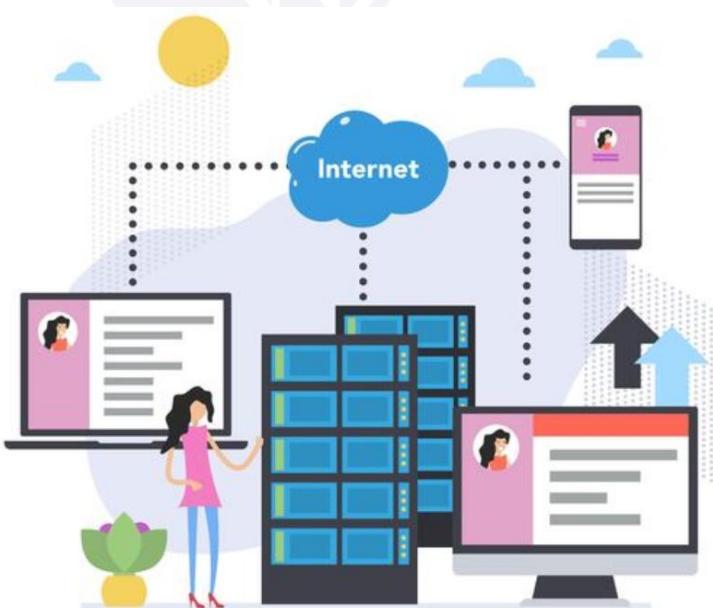


What are Client and Server ?



Client and server are two fundamental components of modern computer networks, used to facilitate communication and data exchange between devices. In this article, we will explore what client and server mean, how they work together, and why they are important in the modern era of computing.

Introduction to Client



In the context of computer networks, a client refers to any device or software application that requests services or resources from a server. Examples of clients include web browsers, email clients, and instant messaging applications.

When a client requests services from a server, it sends a message or request containing the specific details of what it is looking for. This request may include information such as the type of resource or service being requested, the format in which the data should be returned, and any authentication or security credentials that may be required.

Once the server receives the request from the client, it processes the request and returns a response containing the requested information or resources. This response may include data such as HTML pages, images, or other media content.

Client devices:- These refer to electronic devices such as computers, laptops, smartphones, tablets, and other computing devices that are used by individuals or organisations to access services or resources provided by a server or network.

Introduction to Server



Do you remember sometimes your favourite apps like Whatsapp and Instagram do not work? You are not able to send messages or see any new content in your feed. Sometimes the phrase "**server is down**" is also very common. Let us understand what a server is.

In contrast to clients, servers are devices or software applications that provide services or resources to clients. Servers are responsible for processing requests from clients, performing the necessary actions or calculations, and returning the results back to the client in the requested format.

Examples of servers include web servers, email servers, and file servers. Each type of server is designed to provide specific services or resources to clients, such as hosting websites, storing and retrieving files, or managing email communication.

Servers are typically connected to the internet or other computer networks, allowing them to communicate with clients located anywhere in the world. They are designed to be highly reliable and available, with advanced features such as redundancy and load balancing to ensure that they can handle high volumes of traffic and requests.

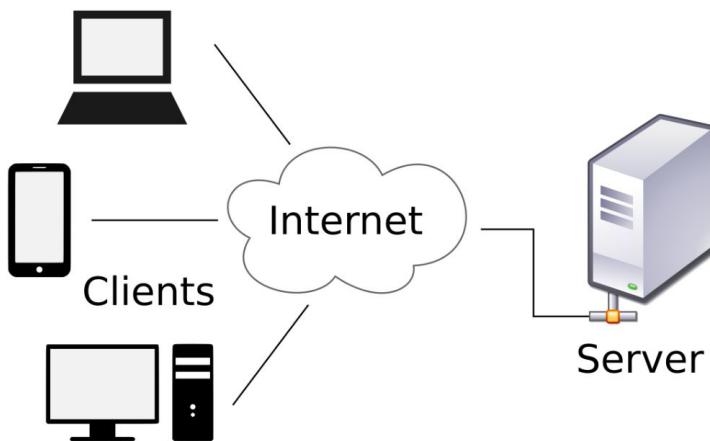
How do Clients and Servers Work Together ?

Clients and servers work together in a request-response model, in which clients request resources or services from servers, and servers provide the requested resources or services back to the clients. This model is based on the principles of client-server architecture, which is a common design pattern used in modern computer networks.

In this model, clients initiate requests by sending messages or packets containing specific details of what they are looking for. Servers receive these requests, process them, and return a response containing the requested information or resources. This response is sent back to the client, which then processes the information and presents it to the user.

Why are Clients and Servers Important ?

Clients and servers are important components of modern computer networks because they enable devices to communicate and exchange data in a structured and efficient manner. This enables users to access a wide range of services and resources, such as websites, email, and file sharing, from anywhere in the world through client devices like mobile phones and laptops.



In addition to their technical benefits, clients and servers also have significant social and economic implications. They have enabled the creation of new industries and businesses, such as online shopping, social media, and cloud computing, and have transformed the way we communicate, work, and live.

How do clients and servers communicate ?

These are the steps involved in the communication between a client and server-

- 1.** The client (such as a web browser) sends a request to the server (such as a web server) over the internet.
- 2.** The request contains information such as the URL of the page being requested, any parameters or data being sent, and any headers such as authentication credentials or expected response type.
- 3.** The server receives the request, processes it, and generates a response.
- 4.** The response contains the requested data, such as the HTML code, images, and other media files needed to display the webpage.
- 5.** The server sends the response back to the client over the Internet.
- 6.** The client receives the response and processes the data, parsing the HTML code and rendering the images and other media files to display the webpage.
- 7.** The client can then interact with the webpage by clicking on links, filling out forms, or navigating to other pages within the website.
- 8.** Each interaction triggers a new request from the client to the server, and the process repeats.



**THANK
YOU !**