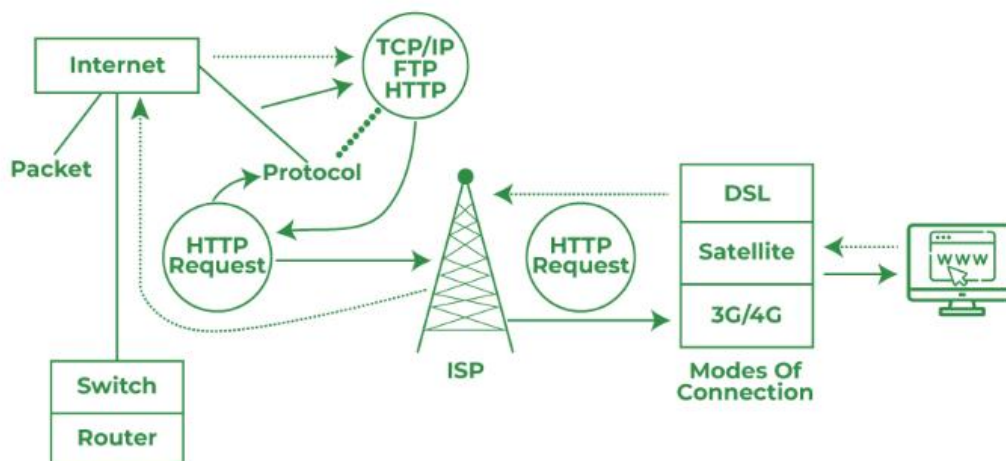


WEB ASSIGNMENT

Q1.How Internet works

The internet is a global network of networks that allows computers and other devices to communicate and share information worldwide. It consists of a vast number of private, public, academic, business, and government networks interconnected through a wide array of electronic, wireless, and optical networking technologies. The internet enables various services such as email, web browsing, file sharing, online gaming, and social networking.

1. Firstly, you'll be required to connect your system or PC with any router or modem to establish a connection. This connection is the base of the internet connection.
2. When you open the browser and start typing something like "www.google.com", your system will push a query command to your ISP (Internet Service Provider) that is connected with other servers that store and process data.
3. Now, the web browser will start indexing the URL that you've entered and will fetch the details in numeric format (in their language to identify the address (*unique*) that you're trying to reach.
4. Next, now your browser will start sending the HTTP request where you're trying to reach and send a copy of the website on the user's system. **Note:** *The server will send data in the form of small packets (from the website to the browser)*
5. Once all the data (of small packets) is received at the user's end (PC/Laptop), the browser will start arranging all those small packets and later will form a collective file (here, the browser will gather all the small packets and rearrange them just like a puzzle) and then you'll be able to see the contents of that website



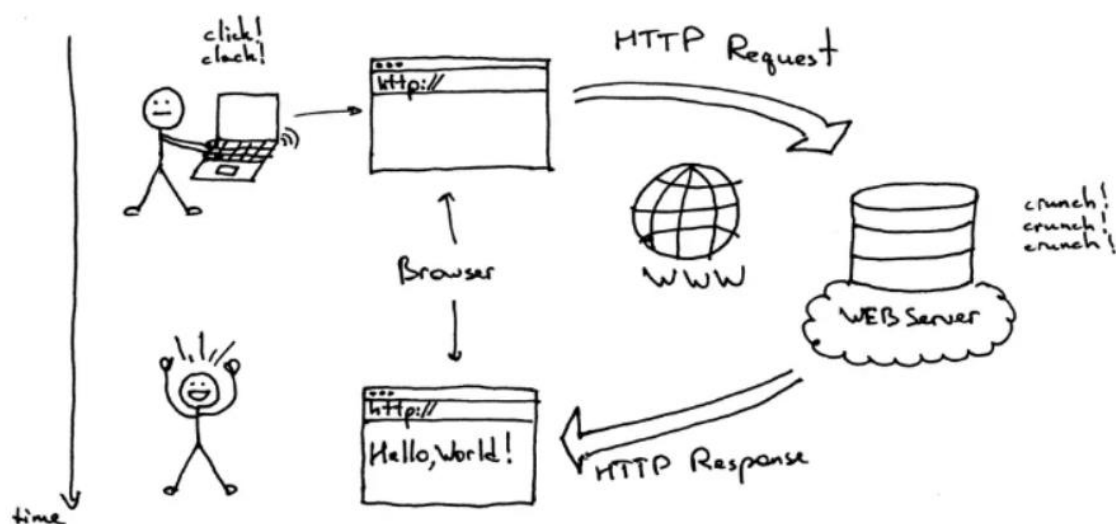
Q2 How web Browser works?

A web browser takes you anywhere on the internet. It retrieves information from other parts of the web and displays it on your desktop or mobile device. The information is transferred using the Hypertext Transfer Protocol, which defines how text, images and video are transmitted on

the web. This information needs to be shared and displayed in a consistent format so that people using any browser, anywhere in the world can see the information.

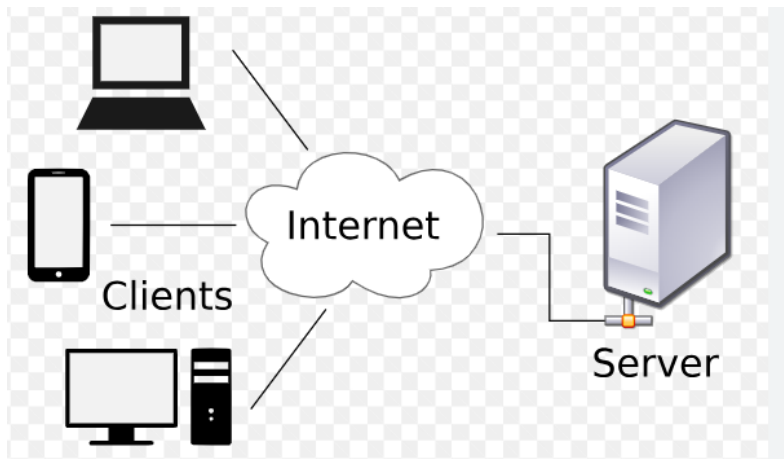
The process begins with Domain Name System (DNS) resolution, where the browser translates the domain name into an IP address to locate the server where the web page is stored.

- The browser then sends an HTTP request to the server, specifying the path and parameters of the requested resource.
- Once the server receives the request, it sends an HTTP response to the browser containing the requested resource in HTML, CSS, and JavaScript code.
- The browser's rendering engine interprets and renders the code to display the web page on the user's device.
- The CSS stylesheets are applied to format the web page's content, including fonts, colors, and layout.
- The browser may also execute JavaScript code on the web page to add interactivity and dynamic behavior.



Q3 What is server?

Server, network computer, computer program, or device that processes requests from a client (see client-server architecture). On the World Wide Web, for example, a Web server is a computer that uses the HTTP protocol to send Web pages to a client's computer when the client requests them. On a local area network, a print server manages one or more printers, and prints files sent to it by client computers. Network servers (which manage network traffic) and file servers (which store and retrieve files for clients) are two more examples of servers.



Q4 What are types of servers available?

Application Server

Users on the network can run and use web apps (software that runs inside a web browser) on application servers without having to install a copy of them on their personal machines. These servers don't have to be connected to the Internet. Web-enabled PCs serve as their clients.

Catalog Server

The catalog servers keep a table of contents or index of the information that is spread out over a wide distributed network. Computers, users, files exchanged via file servers, and web apps can all be part of a distributed network. Directory servers or name servers are a couple of examples of catalog servers. Any computer programme that has to search the network is one of their clients. An email client seeking an email address, a user looking for a file, or a domain member trying to log in are a few examples.

Communications Server

The communication servers keep up the environment required for a communication endpoint to locate and then connect with other endpoints. Depending on the network's openness and security settings, these servers may or may not have a directory containing communication endpoints or a presence detection service. Their customers are points of communication.

Computing Server

Over a network, computing servers share a sizable amount of computational resources, such as CPU and random-access memory. These kinds of servers can be used by any computer programme that requires more CPU and RAM than a single computer is likely able to provide. To implement the client-server concept, which is essential, the client has to be a networked computer.

Database Server

Any type of database can be maintained and shared using database servers via a network. A structured collection of data with predetermined properties that may be shown in a table is referred to as a database. Spreadsheets, asset management software, accounting software, and just about any other computer programme that consumes well-organised information, particularly in huge amounts, are clients of these servers.

Fax Server

By sharing one or more fax machines across a network, fax servers do away with the inconvenience of physical access. These servers' clients might be either fax senders or recipients.

File Server

The file servers share over network files and folders, along with storage space for files and folders, or both. Even though local programmes might be clients, networked machines are the intended recipients.

Mail Server

Similar to how a post office facilitates communication via snail mail, mail servers enable communication via email. Both email senders and recipients are clients of these servers.

Print Server

The inconvenience of physical access is removed by the print servers' network sharing of one or more printers. Computers in need of printing are their customers.

Proxy Server

By receiving incoming traffic from a client and transferring it to the server, the proxy server serves as an intermediate between a client and another server. The use of a proxy server can be justified for a number of reasons, such as content filtering and control, traffic performance enhancement, preventing unwanted network access or simple traffic routing over a big and complicated network. Any networked computer can be one of the clients.

Web Server

Websites are hosted on web servers. The World Wide Web is only feasible because of web servers. There are one or more web servers for each website. Computers equipped with web browsers serve as the clients.



Q5.What is SEO?Importance of SEO?

SEO stands for “search engine optimization.” In simple terms, SEO means the process of improving your website to increase its visibility in Google, Microsoft Bing, and other search engines whenever people search for:

- Products you sell.
- Services you provide.
- Information on topics in which you have deep expertise and/or experience.

The better visibility your pages have in search results, the more likely you are to be found and clicked on. Ultimately, the goal of search engine optimization is to help attract website visitors who will become customers, clients or an audience that keeps coming back.

SEO is a critical marketing channel. First, and foremost. That’s one big reason why the global SEO industry is forecast to reach a staggering \$122.11 billion by 2028. SEO drives real business results for brands, businesses and organizations of all sizes. Whenever people want to go somewhere, do something, find information, research or buy a product/service – their journey typically begins with a search.

But today, search is incredibly fragmented. Users may search on traditional web search engines (e.g., Google, Microsoft Bing), social platforms (e.g., YouTube, TikTok) or retailer websites (e.g., Amazon).

There are three types of SEO:

- **Technical SEO:** Optimizing the technical aspects of a website.
- **On-site SEO:** Optimizing the content on a website for users and search engines.
- **Off-site SEO:** Creating brand assets (e.g., people, marks, values, vision, slogans, catchphrases, colors) and doing things that will ultimately enhance brand awareness and recognition (i.e., demonstrating and growing its expertise, authority and trustworthiness) and demand generation.

Q6.What is Accessibility?

Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them. More specifically, people can:

- perceive, understand, navigate, and interact with the Web
- contribute to the Web

Web accessibility encompasses all disabilities that affect access to the Web, including:

- auditory
- cognitive
- neurological
- physical
- speech
- visual

Web accessibility also benefits people *without* disabilities, for example:

- people using mobile phones, smart watches, smart TVs, and other devices with small screens, different input modes, etc.
- older people with changing abilities due to ageing
- people with “temporary disabilities” such as a broken arm or lost glasses
- people with “situational limitations” such as in bright sunlight or in an environment where they cannot listen to audio
- people using a slow Internet connection, or who have limited or expensive bandwidth

Q7.What is HTML?

HTML is the standard markup language for creating Web page.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Ex:

```

<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>

```

Q9.What is browser engine?

A browser engine (also known as a layout engine or rendering engine) is a core software component of every major web browser that is responsible for transforming HTML documents and other resources of a web page into an interactive visual representation on a user's device. Common browser engines include Blink, Trident, Gecko, and WebKit.

A web browser is a software application used to access information on the World Wide Web, it retrieves information from other parts of the web and displays it on your desktop or mobile device.

The main function of a browser is to present the web resource you choose by requesting it from the server and displaying it in the browser window. The browser's user interface includes elements such as the address bar, back, and forward buttons, and **#bookmarks**. The HTML5 specification does not define the UI elements a browser must have but lists some common elements.

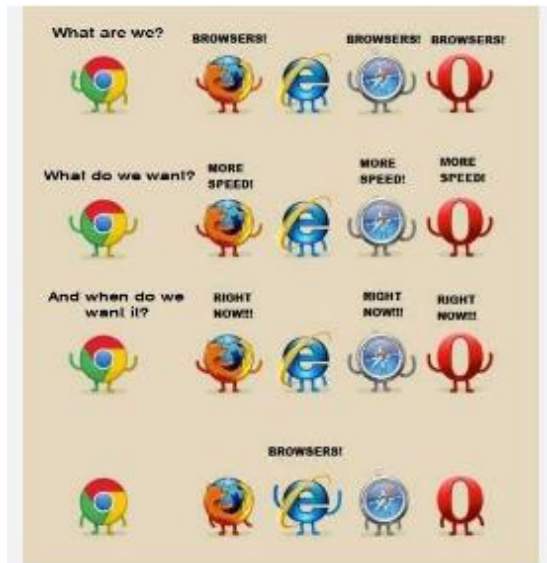
What is the difference between a web browser and a search engine?

A web browser and a search engine are two different types of software that are commonly used to access the internet.

A web browser is a software application that allows users to view and navigate through web pages on the internet. It retrieves information from a web server and displays it on the user's device. The most popular web browsers include **#Google #chrome**, **#Mozilla Firefox**, **#Microsoft Edge**, **#Safari**, and **#Opera**.

A search engine, on the other hand, is a software application or a service that allows users to search for and find information on the internet. It uses algorithms to search through billions of web pages and returns the most relevant results to the user's query. The most popular **#search** engines include **#Google**, **#Bing**, **#Yahoo**, and **#DuckDuckGo**.

In summary, a web browser is used to view and navigate through web pages, while a search engine is used to find specific information on the internet.

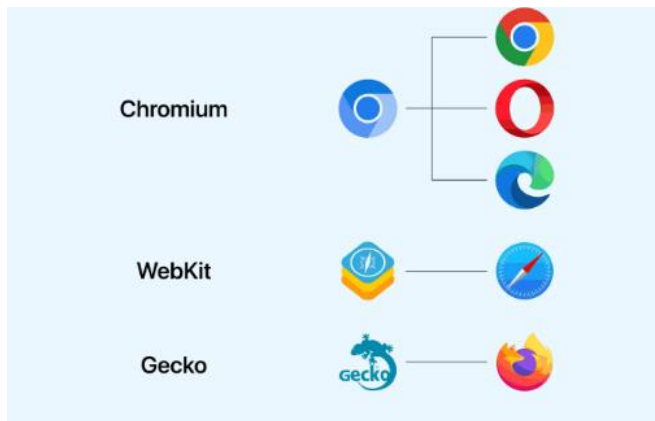


Q10.What is rendering engine?

A rendering engine is a software program that interprets and converts the HTML, CSS, and JavaScript code of a web page into visuals that are displayed on the screen. It is the core component of a web browser and plays a crucial role in the overall performance and compatibility of the browser. Different browsers use different rendering engines with changes that reflect the browser's performance goal and accessibility.

There are three primary rendering engines that power the majority of web browsers today:

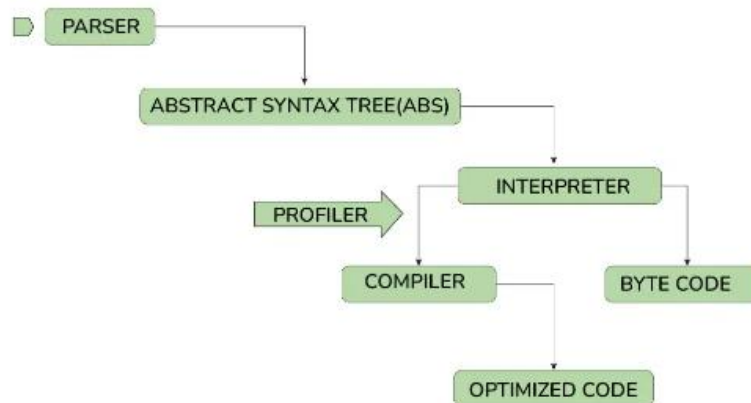
- **Blink:** Blink is an open-source rendering engine developed by Google and is the foundation for the Chrome browser. It is also used by several other browsers, including Opera, Brave, and Vivaldi.
- **Gecko:** Gecko is another open-source rendering engine developed by Mozilla and is the engine that powers the Firefox browser. It is known for its adherence to web standards and strict compatibility with various web technologies.
- **WebKit:** WebKit is an open-source rendering engine initially developed by Apple for its Safari browser. It is also used by the iOS version of the Chrome browser and serves as the basis for the Qt WebEngine framework.
- **EdgeHTML:** EdgeHTML, a fork of Trident (Internet Explorer's engine), is Microsoft's rising star. It's known for its focus on interoperability and compatibility with older web technologies while embracing modern advancements. Think of it as a bridge builder, connecting the past and present of the web, ensuring smooth transitions for users and developers alike.
- **Other Engines:** Beyond these giants, a constellation of niche engines exists, catering to specific needs and platforms. WebKit variations like Presto (Opera Mini) and Goanna (Vivaldi) offer unique mobile experiences. WebRender (Android WebView) prioritizes efficiency on low-powered devices. Each engine, like a specialist chef, brings its own flavor and expertise to the table, enriching the web's diversity and adaptability.



Q11.What is JavaScript engine?Share the available JS engine?Purpose of JS Engine?

JavaScript is a scripting language and is not directly understood by computer but the browsers have inbuilt JavaScript engine which help them to understand and interpret JavaScript codes. These engines help to convert our JavaScript program into computer-understandable language.

A JavaScript engine is a computer program that executes JavaScript code and converts it into computer understandable language.

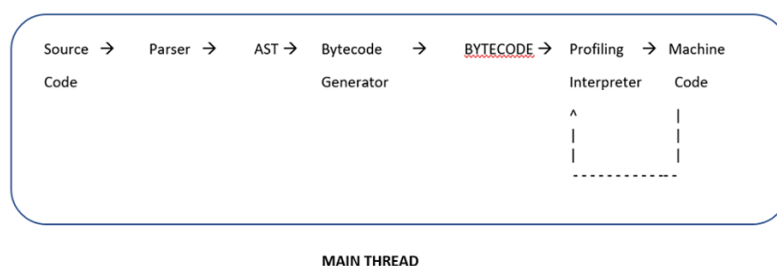


Browser	Name of Javascript Engine
Google Chrome	V8
Edge (Internet Explorer)	Chakra
Mozilla Firefox	Spider Monkey
Safari	Javascript Core Webkit

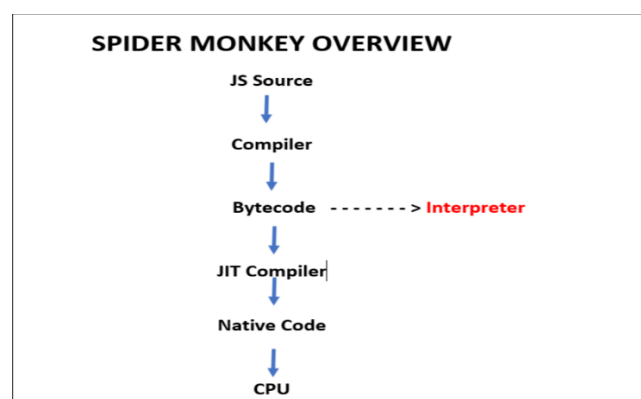
1. V8: V8 is a JavaScript engine developed by the Chromium Project for Google Chrome and Chromium web browsers. It is a JavaScript engine that can run standalone, or be embedded into any C++ application. Using its own parser, it generates an abstract syntax tree. Then, Ignition generates bytecode from this syntax tree using the internal V8 bytecode format. Bytecode is compiled into machine code by TurboFan. It also handles memory allocation for objects, and garbage collects objects it no longer needs. Optimization techniques such as elision of expensive runtime properties, and inline caching. The garbage collector is a generational incremental collector.

V8 provides an edge as it allows JavaScript to run much faster, which improves users' experience of the web, paves the way for the development of web applications, and spurs rapid growth of server-side JavaScript through projects like Node.js.

2. Chakra: Chakra is a JScript engine developed by Microsoft. It is proprietary software. It is used in the Internet Explorer web browser. A distinctive feature of the engine is that it JIT compiles scripts on a separate CPU core, parallel to the web browser.



3. Spider Monkey: SpiderMonkey is the first JavaScript engine, written by Brendan Eich at Netscape Communications, later released as open-source and currently maintained by the Mozilla Foundation. It is still used in the Firefox web browser.



4. Webkit: WebKit is developed by Apple and used in its Safari web browser, as well as all iOS web browsers. It is used by the BlackBerry Browser, PlayStation consoles beginning from the PS3, the Tizen mobile operating systems, and a browser included with the Amazon Kindle e-book reader. WebKit's C++ Application Programming Interface (API) provides a set of classes to display Web content in windows and implements browser features such as following links when clicked by the user, managing a back-forward list, and managing a history of pages recently visited.

Key tasks of a JS engine include:

1. **Parsing:** The engine reads and parses the JavaScript code to understand its syntax and structure.
2. **Compilation:** The engine may compile the parsed code into an intermediate representation or directly into machine code for faster execution.
3. **Optimization:** Modern JS engines often employ various optimization techniques to improve the performance of JavaScript code. This can include inline caching, just-in-time (JIT) compilation, and other optimizations based on runtime profiling.
4. **Execution:** Once optimized, the engine executes the JavaScript code, performing calculations, manipulating data, interacting with the browser's Document Object Model (DOM), handling events, and more.
5. **Memory Management:** JS engines manage memory allocation and deallocation to ensure efficient use of resources.

Popular JS engines include V8 (used in Chrome and Node.js), SpiderMonkey (used in Firefox), JavaScriptCore (used in Safari), and Chakra (used in older versions of Microsoft Edge).

Overall, JS engines play a crucial role in making JavaScript a powerful and versatile language for developing both client-side and server-side applications on the web.

Q12. How does website work?

Step 1: User Requests a Website

It all begins when a user enters a website's URL or clicks on a link leading to a specific site. This seemingly simple action sets in motion a series of events that happen behind the scenes.

Imagine you are sitting at your computer, sipping a cup of coffee, and decide to visit your favorite news website. You type in the URL or click on a bookmark, and in an instant, your request is sent from your browser to a server that hosts the website.

Now, let's take a moment to explore what happens next.

Step 2: Server Processes the Request

Upon receiving the user's request, the server springs into action. It analyzes the URL to determine which website the user wants to visit and which files and data are needed to fulfill the request.

Behind the scenes, the server may need to access databases to retrieve the latest news articles, fetch image files from a storage system, or generate dynamic content based on the user's preferences. This process can involve complex algorithms, database queries, and server-side scripting languages like PHP or Python.

Once the server has gathered all the necessary files and data, it prepares to send them back to the user's browser. But how does it do that?

Step 3: Server Sends Data to the Browser

The server packages up the requested files and data and sends them back to the user's browser. This data typically includes HTML, CSS, JavaScript files, and any other relevant assets needed to render the website correctly.

Think of this data as a set of instructions for the browser. The HTML file provides the structure and content of the website, while the CSS file specifies the styles and layout. The JavaScript file adds interactivity and dynamic behavior to the website.

As the data travels across the internet, it passes through various network routers and switches, each playing a role in delivering the information to its destination. This intricate dance of data transfer happens in a matter of milliseconds, thanks to the robust infrastructure that underpins the internet.

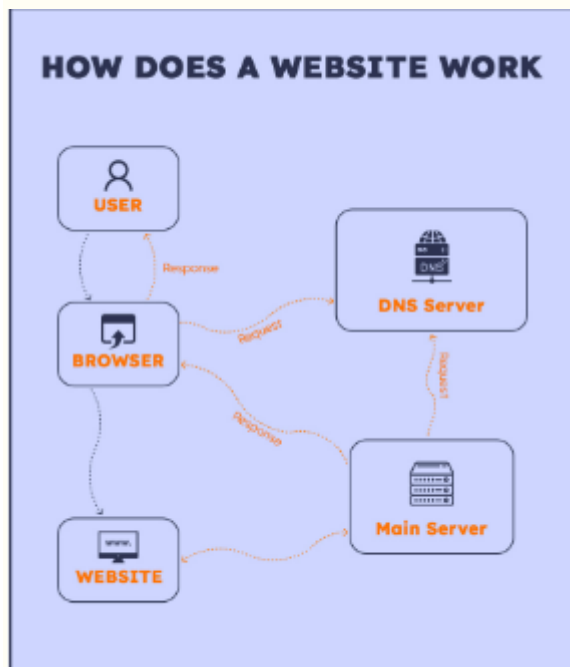
Step 4: Browser Displays the Website

Now comes the moment of truth. The user's browser receives the data and begins to interpret it. It follows the instructions provided in the HTML, CSS, and JavaScript files to render the website.

The browser first parses the HTML file, creating a Document Object Model (DOM) that represents the structure of the website. It then applies the CSS styles to each element, determining their visual appearance and layout on the page. Finally, it executes any JavaScript code, enabling interactivity and dynamic updates.

As the browser works its magic, the website gradually takes shape on the user's screen. Images load, text appears, and buttons become clickable. The user can now navigate the website, click on links, submit forms, and interact with various elements.

And that, in a nutshell, is how a website works. From the user's initial request to the final display of the website, a complex web of technologies and processes collaborates seamlessly to deliver the digital experiences we have come to rely on.

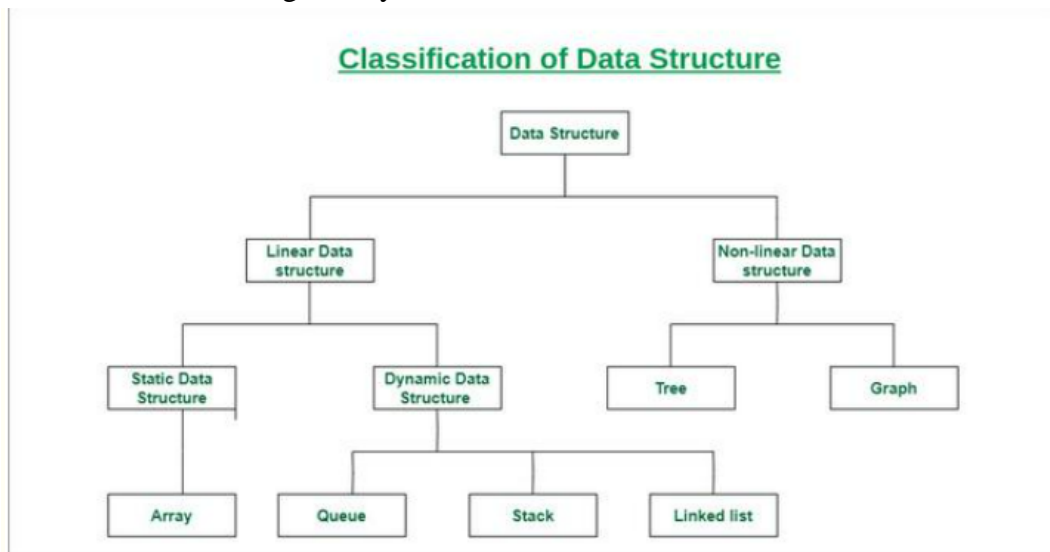


A data structure is a way of organizing and storing data in a computer so that it can be accessed and used efficiently. It refers to the logical or mathematical representation of data, as well as the implementation in a computer program.

- sequentially or linearly, where each element is attached to its previous and next adjacent elements, is called a linear data structure. Examples are array, stack, queue, etc.
- **Non-linear Data Structure:** Data structures where data elements are not placed sequentially or linearly are called non-linear data structures.

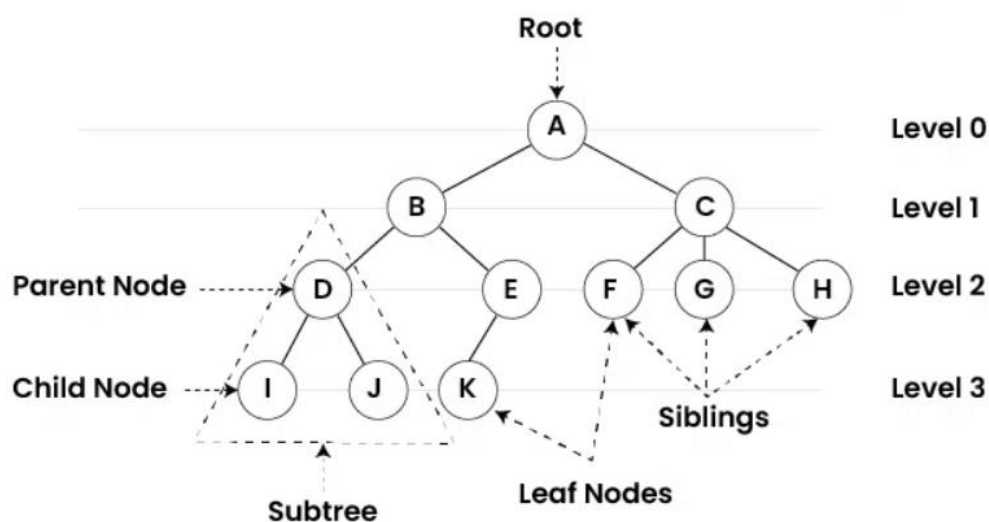
- **Operating systems:** Data structures are used in the design and implementation of operating systems to manage system resources, such as memory and files.

- **Computer graphics:** Data structures are used to represent geometric shapes and other graphical elements in computer graphics applications.
- **Artificial intelligence:** Data structures are used to represent knowledge and information in artificial intelligence systems.



Q14.Explain the Tree Data Structure?

Tree Data Structure is a non-linear data structure in which a collection of elements known as nodes are connected to each other via edges such that there exists exactly one path between any two nodes.



- **Parent Node:** The node which is a predecessor of a node is called the parent node of that node. {B} is the parent node of {D, E}.
- **Child Node:** The node which is the immediate successor of a node is called the child node of that node. Examples: {D, E} are the child nodes of {B}.

- **Root Node:** The topmost node of a tree or the node which does not have any parent node is called the root node. {A} is the root node of the tree. A non-empty tree must contain exactly one root node and exactly one path from the root to all other nodes of the tree.
- **Leaf Node or External Node:** The nodes which do not have any child nodes are called leaf nodes. {K, L, M, N, O, P, G} are the leaf nodes of the tree.
- **Ancestor of a Node:** Any predecessor nodes on the path of the root to that node are called Ancestors of that node. {A,B} are the ancestor nodes of the node {E}
- **Descendant:** A node x is a descendant of another node y if and only if y is an ancestor of x.
- **Sibling:** Children of the same parent node are called siblings. {D,E} are called siblings.
- **Level of a node:** The count of edges on the path from the root node to that node. The root node has level 0.
- **Internal node:** A node with at least one child is called Internal Node.
- **Neighbor of a Node:** Parent or child nodes of that node are called neighbors of that node.
- **Subtree:** Any node of the tree along with its descendant.

Q15.What is user agent?Share the list and its purpose?

User Agent (UA) is a string sent by the user's web browser to a server. It's located in the HTTP header and identifies the browser type and version as well as the operating system. Accessed with JavaScript on the client side using `navigator.userAgent` property, the remote web server uses this information to identify and render the content in a way that's compatible with the user's specifications.

While different structures and information are contained, **most web browsers tend to follow the same format:**

Mozilla/5.0 (<system-information>) <platform> (<platform-details>) <extensions>

1. **Content Negotiation:** Websites use the user agent string to deliver optimized content or different versions of a webpage based on the capabilities of the browser or device.
2. **Browser Detection:** It helps websites detect the type and version of browser accessing the site, which can be useful for debugging or providing specific instructions or content.
3. **Compatibility:** User agent strings assist in ensuring that websites render properly on different browsers and devices by providing information about the browser's rendering capabilities.

Q16.What is hypertext?

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

Q17.What are HTML tags?

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

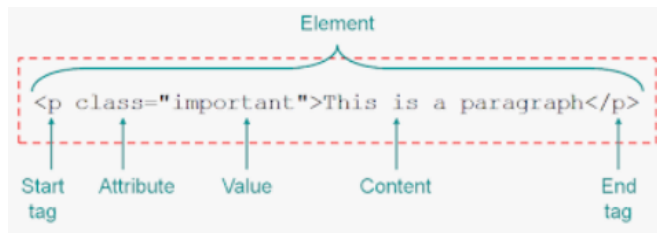
When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h n> ... </h n>	Delimits a level <i>n</i> heading
 ... 	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
 ... 	Brackets an unordered (bulleted) list
 ... 	Brackets a numbered list
 ... 	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
	Displays an image here
 ... 	Defines a hyperlink

Q18.What are HTML attributes?

- HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.
- Each element or tag can have attributes, which defines the behaviour of that element.
- Attributes should always be applied with start tag.
- The Attribute should always be applied with its name and value pair.
- The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
- You can add multiple attributes in one HTML element, but need to give space between two attributes.

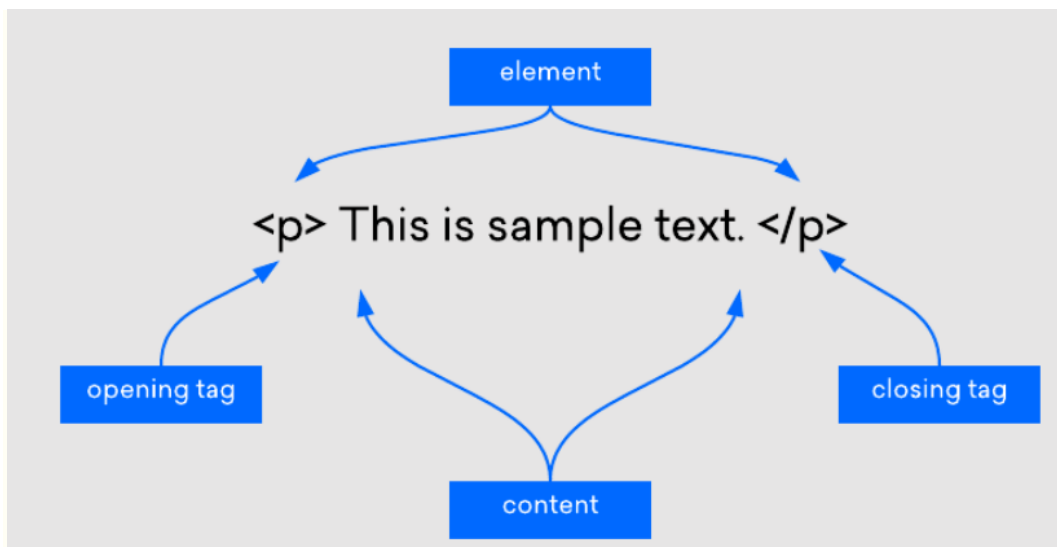


Q19.What are HTML elements?

An HTML element is a component of an HTML document that tells a web browser how to structure and interpret a part of the HTML document. HTML elements can contain formatting instructions, semantic meaning, and content.

For example, HTML elements are used to denote document parts such as headers, paragraphs, and footers and to embed content such as hyperlinks, text, and images. Although HTML can be used to provide formatting instructions, HTML standards strongly encourage using CSS for this purpose instead.

An HTML element is often — but not always — created by opening and closing HTML tags, which wrap around a piece of content. Below is an illustration that labels each of the parts of an HTML element:



Q20.How to convert elements to tree?

In HTML, elements are already structured in a tree-like hierarchy by nature of the Document Object Model (DOM). The DOM represents HTML elements as a hierarchical tree where each element is a node, and elements can have parent, child, and sibling relationships.

Here's how you typically work with HTML elements as a tree:

1. **Understanding the DOM Tree:** The DOM (Document Object Model) represents HTML elements as nodes in a tree structure. The <html> element is the root of the tree, and all other elements are its descendants.
2. **Traversal and Manipulation:** JavaScript is commonly used to manipulate the DOM tree dynamically. You can traverse from one element to its children, siblings, or parents using DOM traversal methods and properties.
3. **Constructing Trees Dynamically:** If you need to dynamically create a tree structure based on data or specific requirements, you can use JavaScript to create, modify, and append elements to the DOM.

Q21.What is DOCTYPE?

The HTML document type declaration, also known as DOCTYPE, is the first line of code required in every HTML or XHTML document. The DOCTYPE declaration is an instruction to the web browser about what version of HTML the page is written in. This ensures that the web page is parsed the same way by different web browsers.

All HTML documents must start with a <!DOCTYPE> declaration.


The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

In HTML 5, the declaration is simple:

```
<!DOCTYPE html>
```

Q22.What are the ways to save html files?

To save an HTML document

1. Make sure the file you want to save is visible in the right pane.
2. Do one of the following:
 - To save the file is saved in its current location with its current name, do one of the following:
 - - On the main menu, click **File > Save**.
 - On the HTML editor toolbar, click the **Save** icon .
 - Press CTRL+S.

- Right-click within the HTML document, click **File** > **Save**.
- To save the file with a new name and/or location, do one of the following:
 - - On the main menu, click **File** > **Save As**.
 - Right-click within the HTML document, click **File** > **Save As**.
 - In the **Save As** dialog box, specify the file name and location, then click **Save**.
- To save all open HTML documents at the same time:
 - On the main menu, click **File** > **Save All**. All open documents are saved. If you have not saved all of them previously, the **Save As** dialog box appears for each new file.
 - a. Specify a name and location for each new file, then click **Save**.

Q23.What are charset?Why do use it?

The charset attribute specifies the character encoding for the HTML document.

The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols in the world. A **character set** is an encoding system to let computers know how to recognize Character, including letters, numbers, punctuation marks, and whitespace

1. **Standardization:** Character sets provide a standardized way to represent and interpret text characters across different computer systems and programming languages. They ensure that text data can be reliably exchanged and understood globally.
2. **Compatibility:** Using a consistent character set ensures compatibility between different software applications, operating systems, and devices. It allows text to be displayed correctly regardless of where or how it is accessed.
3. **Multilingual Support:** Character sets support the representation of characters from multiple languages and writing systems. This is crucial for internationalization and localization efforts, enabling software to handle diverse linguistic requirements.
4. **Encoding:** Character sets define how characters are encoded into binary data for storage or transmission. This encoding ensures that each character can be represented by a unique sequence of bits, facilitating efficient storage and communication of text data.
5. **Unicode:** The Unicode standard is a widely adopted character set that encompasses a vast range of characters from various languages and scripts worldwide. It supports over

a million distinct characters, making it comprehensive for global communication and data interchange.

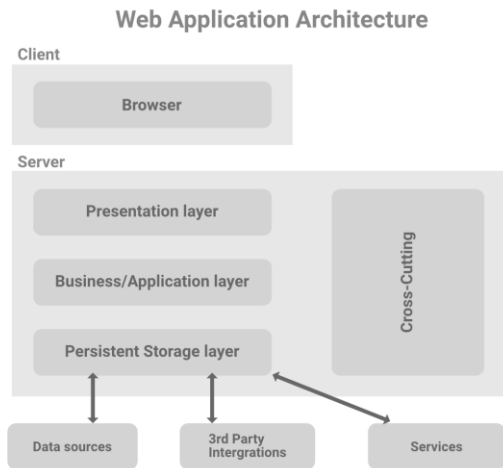
Q24.What is metadata?What is its purpose?

Metadata is defined as **the information that describes and explains data**. It provides context with details such as the **source, type, owner, and relationships to other data sets**. So, it can help you understand the relevance of a particular data set and guide you on how to use it. In a nutshell: Metadata is a cornerstone of a modern enterprise data stack.

1. **Metadata should provide a dataset with context.** This means explaining what it covers, the themes, keywords to describe it, how the data has been collected and information such as any numerical units used, such as dollars, inches or centimeters.
2. **Metadata should make a dataset unique.** It has to differentiate it from other, similar datasets so that users can choose between them with confidence.
3. **Metadata should provide the framework for subsequent uses of a dataset.** This includes licensing conditions, whether it can be used externally as well as internally, and any organizational rules around who can use the data, and for what purposes.
4. **Metadata should make you want to reuse a dataset.** It has to be comprehensive and compelling, providing clear descriptions that accelerate usage, while outlining the formats it is available in, and suggesting potential ways it can be reused.
5. **Metadata should make a dataset interoperable.** It should follow set internal or external standards so that data can be confidently used or compared with information in other datasets. At a basic level this means standardizing how fields are described, and formats such as dates.
6. **Metadata should provide reassurance regarding the dataset's reliability.** By including information on the source, how often it is updated, and what it covers, users should be able to ascertain how reliable the data is.

Q25.Explain web Application Architecture.

Web application architectural patterns are separated into many different layers or tiers which is called Multi- or Three-Tier Architecture. You can easily replace and upgrade each layer independently.



Presentation Layer: This layer is accessible to the client via a browser and it includes user interface components and UI process components. As we have already discussed that these UI components are built with HTML, CSS, and JavaScript (and its frameworks or library) where each of them plays a different role in building the user interface.

Business Layer: It is also referred to as a Business Logic or Domain Logic or Application Layer. It accepts the user's request from the browser, processes it, and regulates the routes through which the data will be accessed. The whole workflow is encoded in this layer. You can take the example of booking a hotel on a website. A traveler will go through a sequence of events to book the hotel room and the whole workflow will be taken care of by the business logic.

Persistence Layer: It is also referred to as a storage or data access layer. This layer collects all the data calls and provides access to the persistent storage of an application. The business layer is closely attached to the persistence layer, so the logic knows which database to talk to and the process of retrieving data becomes more optimized. A server and a database management system software exist in data storage infrastructure which is used to communicate with the database itself, applications, and user interfaces to retrieve data and parse it. You can store the data in hardware servers or in the cloud.

Some other parts of the web application which is separated from the main layers that exist in the architecture are...

- **Cross-cutting code:** This part handles communications, operational management, and security. It affects all parts of the system but should never mix with them.
- **Third-party integrations:** Using third-party APIs we can integrate payment gateways, social logins, GDSs in travel websites, etc.

