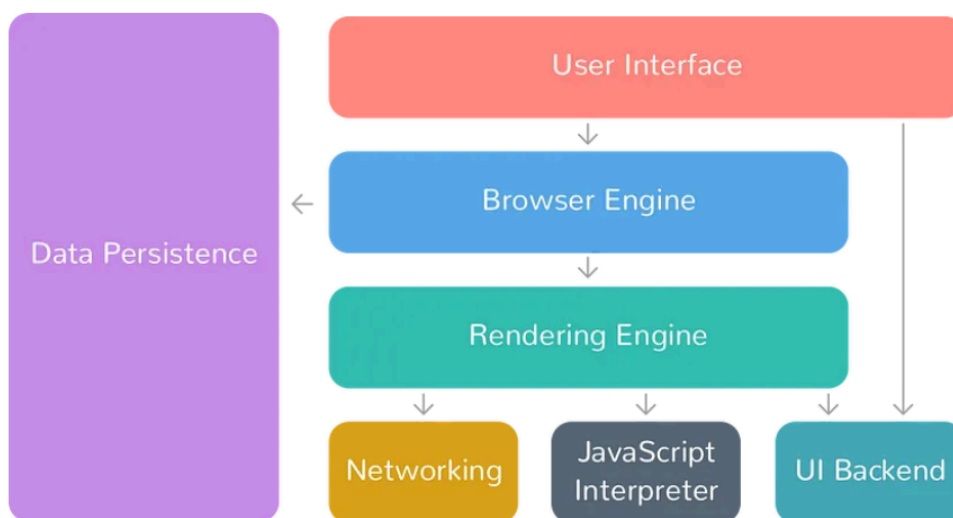


### 1. How internet works?

Ans: The Internet is a network of networks. It works by using a technique called packet switching, and by relying on standardized networking protocols that all computers can interpret.

### 2. How browser works?

Ans: Browsers are responsible for retrieving and displaying web content to users. When a user enters a URL or clicks on a link, the browser initiates a complex series of actions to retrieve the web content from a server and display it on the user's device.



### 3. What is Server?

Ans: A server is a computer program or device that provides a service to another computer program and its user, also known as the client. In a data-center, the physical computer that a server program runs on is also frequently referred to as a server. That machine might be a dedicated server or it might be used for other purposes.

### 4. what are the types of server available?

- Web Server
- Database Server
- Email Server
- Web Proxy Server
- DNS Server
- FTP Server
- File Server
- DHCP Server
- Cloud Server
- Application Server
- Print Server
- NTP Server
- Radius Server
- Syslog Server

- Physical Server

## 5. What is SEO? Importance of SEO?

Ans: 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.

Importance:

- 1. Search Engines Generate Lots of Traffic
- 2. SEO Can Be Highly Cost Effective
- 3. Search Visibility Drives Brand Awareness and Authority
- 4. SEO Success Supports Other Marketing Efforts

## 6. What is Accessibility?

Ans: Accessibility is the practice of making your websites usable by as many people as possible. We traditionally think of this as being about people with disabilities, but the practice of making sites accessible also benefits other groups such as those using mobile devices, or those with slow network connections.

### ACCESSIBILITY

is about making IT solutions easier to use for people with impairments - either temporary, permanent, or situational - that affect:



## 7. What is Markup Language?

Ans: A markup language is a set of rules that defines how the layout and presentation of text and images should appear in a digital document. It allows structuring documents, adding formatting, and specifying how different elements should be displayed (or “rendered”) on webpages.

## 8. What is HTML?

- 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

## 9. What is browser engine?

Ans: A web browser is a software application that lets you explore the internet. It retrieves and displays web pages, images, videos, and other content from web servers. Each piece of content has a unique address called a URL (Uniform Resource Locator), which tells the browser where to find it. Think of a browser engine as the heart of your web browser. **It is the**

essential software that acts as a bridge between the web page's code (HTML, CSS, JavaScript) and the visual experience you see on your screen.

## BROWSER ENGINES



10. What is rendering engine? share the available rendering engine?

Ans: Rendering Engine. As the name suggests, this component is responsible for rendering a specific web page requested by the user on their screen. It interprets HTML and XML documents along with images that are styled or formatted using CSS, and a final layout is generated, which is displayed on the user interface.

- **Blink:** Developed by Google, Blink is a fork of the WebKit rendering engine (originally developed by Apple). It is used in Google Chrome, Microsoft Edge (since 2020), and various other browsers.
- **WebKit:** Developed by Apple, WebKit is the engine used in Safari. It was originally derived from the KHTML engine (used in the Konqueror browser).
- **Gecko:** Developed by Mozilla, Gecko is the engine used in the Firefox browser. It is known for its standards compliance and open-source nature.
- **Trident (MSHTML):** Developed by Microsoft, Trident was the engine used in older versions of Internet Explorer (up to version 11). It has largely been replaced by the EdgeHTML engine and subsequently Blink in Microsoft Edge.
- **EdgeHTML:** Developed by Microsoft, EdgeHTML was the original engine used in Microsoft Edge (up to version 18). It has been replaced by the Blink engine in newer versions of Microsoft Edge.

11. What is JavaScript Engine? share the available JS engine? Purpose of JS Engine?

Ans: A JavaScript engine is a crucial component of web browsers and other applications that run JavaScript code. It interprets and executes JavaScript code so that web pages and web applications can function dynamically.

### Available JavaScript Engines:

1. **V8:** Developed by Google, V8 is a high-performance JavaScript engine used in Google Chrome and also in Node.js. It compiles JavaScript code directly into native machine code before executing it.

2. **SpiderMonkey:** Developed by Mozilla, SpiderMonkey is the JavaScript engine used in the Firefox browser. It was the first JavaScript engine to be implemented and is known for its robustness and compliance with JavaScript standards.
3. **JavaScriptCore:** Developed by Apple, JavaScriptCore (also known as Nitro) is the JavaScript engine used in the Safari browser. It powers JavaScript execution on macOS and iOS devices.
4. **Chakra (Legacy):** Developed by Microsoft, Chakra was the JavaScript engine used in the Edge browser (prior to switching to Chromium-based Edge). It was known for its performance optimizations and support for Microsoft's proprietary features.

### Purpose of JavaScript Engine:

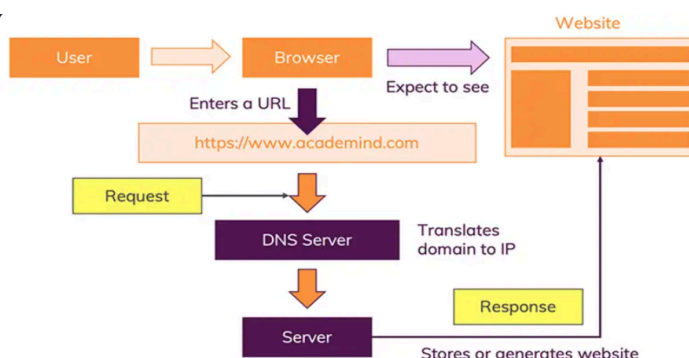
The main purpose of a JavaScript engine is to:

- **Interpret JavaScript Code:** It reads and understands JavaScript code written by developers.
- **Compile and Execute:** It translates JavaScript code into machine code that the computer's processor can execute directly.
- **Optimize Performance:** It optimizes the execution of JavaScript code to improve speed and efficiency.

### 12. How website works?

AnsA website works by serving web pages to users through a combination of client-side and server-side technologies:

1. **Client-side:** When a user enters a URL or clicks a link, their web browser sends a request to a server using HTTP. The server responds with HTML, CSS, JavaScript, and other resources. The browser then interprets and displays these resources, rendering the web page for the user to interact with.
2. **Server-side:** On the server, web applications process requests, generate dynamic content, and interact with databases or other services as needed. Server-side technologies like PHP, Python, Node.js, etc., handle these tasks before sending responses back to the client's browser.



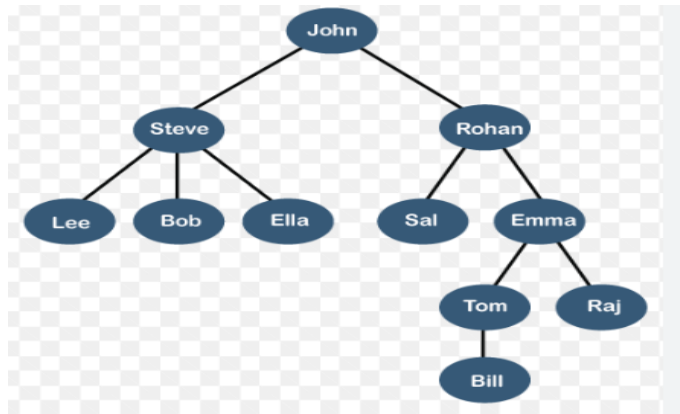
### 13. What is Data Structure?

Ans: DATA STRUCTURES UNIT-1: Data Structure: It is a particular way of organizing and storing data in a computer. So that it can be accessed and modified efficiently. A data

structure will have a collection of data and functions or operations that can be applied on the data.

#### 14. Explain Tree Data Structure?

Ans: A tree data structure is defined as a collection of objects or entities known as nodes that are linked together to represent or simulate hierarchy. A tree data structure is a non-linear data structure because it does not store in a sequential manner.



#### 15. What is user agent? share the list and its purpose?

Ans: A user agent is a string of text sent by a client (usually a web browser) to a server, identifying the client software and its capabilities. This string helps servers determine how to format and deliver content back to the client.

##### **Purpose of User Agents:**

1. **Content Negotiation:** Servers use the user agent string to determine the capabilities of the client (such as browser type, version, operating system) to provide content that is compatible and optimized for that client.
2. **Browser Compatibility:** Websites may adjust their layout, features, or functionality based on the user agent to ensure a consistent and user-friendly experience across different browsers and devices.
3. **Analytics and Logging:** User agent strings can be used by website administrators for analytics purposes to track browser usage statistics, device types, and operating systems of visitors.
4. **Security:** User agents can also play a role in security, helping servers identify potentially malicious requests or patterns based on unusual or unexpected user agents.

#### 16. What is Hypertext?

Ans: Hypertext is a concept in information technology that refers to text displayed on a computer or other electronic device that contains links to other texts. These links, known as hyperlinks or simply links, allow users to navigate between different pieces of information by clicking on them.

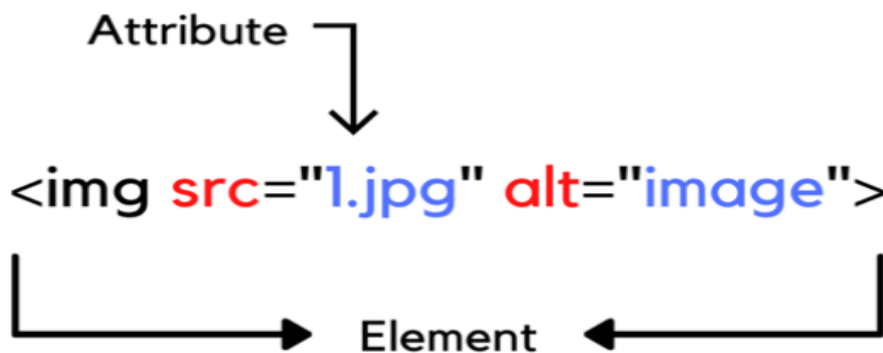
#### 17. What is HTML Tags?

Ans: HTML tags are the keywords on a web page that define how your web browser must format and display your web page. Almost all tags contain two parts, an opening, and a closing tag.



18. What is HTML Attributes?

Ans: An HTML attribute is a piece of markup language used to adjust the behavior or display of an HTML element. For example, attributes can be used to change the color, size, or functionality of HTML elements.



19. What is HTML Elements?

Ans: 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.

20. How do convert elements to tree?

Ans: Converting elements into a tree structure, such as the Document Object Model (DOM) in web development, happens through parsing and organizing HTML elements into a hierarchical representation:

1. **Parsing:** The web browser parses the HTML document, breaking it down into individual elements, attributes, and text nodes based on the HTML syntax.
2. **Building the DOM Tree:** After parsing, the browser constructs a DOM tree by organizing elements into a hierarchical structure. Elements become nodes, with parent-child relationships established based on their nesting in the HTML markup.

21. What is DOCTYPE?

Ans: DOCTYPE (Document Type Declaration) is an important element in HTML documents that specifies the type and version of HTML (or XHTML) used in a web page. It is a

declaration at the beginning of an HTML document that helps web browsers to correctly interpret and render the content.

22. What are the ways we can save html file?

- Navigate to the web page, right-click on the page and select Save as...
- Select or create a new folder to save the file, images, and associated items from the web page.
- Enter a file name and select Webpage, Complete (\*.htm;\*.html) for the Save as type.
- Click the Save button.

23. What is charset? why we need to use this?

Ans:Charset (character set) in the context of web development refers to the encoding system that defines how characters are represented and stored as binary data (bits and bytes). Here's why it's essential:

1. **Text Representation:** Charset specifies how characters (letters, numbers, symbols) are mapped to binary code for computers to process and display correctly.
2. **Compatibility:** It ensures that web browsers interpret and display text content accurately. Incorrect or missing charset declarations can lead to garbled text, especially for non-ASCII characters.

24. What is meta data? what is the purpose of it?

Ans:Metadata refers to data that provides information about other data. In the context of web development and HTML, metadata is often represented using `<meta>` tags within the `<head>` section of an HTML document. Here's its purpose and importance:

1. **Purpose of Metadata:**
  - **Informational Context:** Metadata provides additional context and details about the web page or document, such as authorship, keywords, description, character encoding (`charset`), viewport settings for responsive design, and more.
  - **SEO (Search Engine Optimization):** Metadata like meta keywords and meta descriptions can impact how search engines index and rank the web page, influencing its visibility in search results.
  - **Browser Behavior:** Some metadata, such as specifying the `charset`, helps browsers interpret and display the content correctly.
  - **Accessibility:** Metadata can include accessibility-related information, helping assistive technologies understand and present content to users with disabilities.

25. Explain Web Application Architecture?

Ans: Web application architecture refers to the fundamental structure and organization of a web application. It defines how various components of the application interact with each other to deliver functionality and data to users. A well-designed architecture ensures scalability, maintainability, and performance of the web application. Here's an overview of common components and their roles in web application architecture:



## Components of Web Application Architecture:

1. **Client-side:** The client-side components reside on the user's device (typically a web browser) and handle user interaction and presentation of the application.
  - **User Interface (UI):** The UI layer includes HTML, CSS, and JavaScript that render the user interface and handle user interactions such as clicks and inputs.
  - **Client-side Frameworks:** Libraries and frameworks like React.js, Angular, or Vue.js provide structure and tools for building interactive user interfaces.
2. **Server-side:** The server-side components handle the logic, processing, and storage of data on the server.
  - **Web Server:** Receives requests from clients and sends back responses. Common web servers include Apache, Nginx, and Microsoft IIS.
  - **Application Server:** Executes the application code and interacts with backend systems such as databases or external APIs. Examples include Node.js, Java EE (Enterprise Edition), ASP.NET, and Ruby on Rails.
  - **Business Logic:** The core of the application that implements specific rules and workflows, ensuring proper functioning and data integrity.
  - **Database:** Stores persistent data required by the application. Common databases include MySQL, PostgreSQL, MongoDB, and Redis.
3. **Middleware:** Middleware components provide additional services and functionalities that facilitate communication between client-side and server-side components.
  - **APIs (Application Programming Interfaces):** RESTful APIs or GraphQL endpoints that allow the client-side to request and receive data from the server.
  - **Authentication and Authorization:** Middleware for user authentication (login) and authorization (access control) to secure the application's resources.
  - **Caching and Load Balancing:** Techniques to optimize performance by caching frequently accessed data and distributing client requests across multiple servers (load balancing).
4. **Data Storage and Management:** Components responsible for storing and managing data used by the application.
  - **Relational Databases:** Structured data stored in tables with relationships (SQL databases like MySQL, PostgreSQL).
  - **NoSQL Databases:** Non-relational databases for handling unstructured or semi-structured data (MongoDB, Redis).
  - **File Storage:** Services for storing and retrieving files (Amazon S3, Google Cloud Storage).
5. **Networking and Infrastructure:** Infrastructure components provide the foundation for running the web application, ensuring availability, scalability, and security.
  - **Hosting and Deployment:** Platforms for deploying and hosting web applications (cloud services like AWS, Azure, Heroku, etc.).
  - **Content Delivery Networks (CDNs):** Distributed servers that deliver web content efficiently to users based on their geographical location.
  - **Security:** Measures such as SSL/TLS encryption, firewalls, and monitoring to protect against cyber threats and ensure data privacy.



