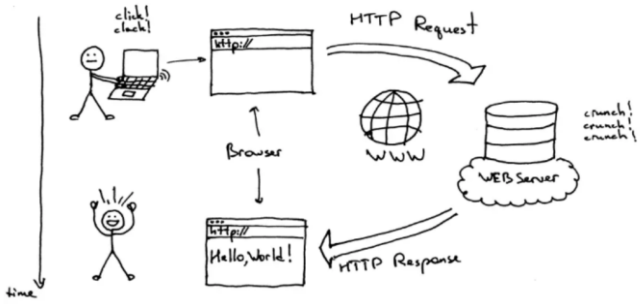
1.How Internet Works?

The Internet works by connecting networks together through a series of routers and switches. A router forwards packets of data between different networks while a switch links devices within a single network. This enables computers to communicate with each other and access content stored on remote servers.

2. How browser works?



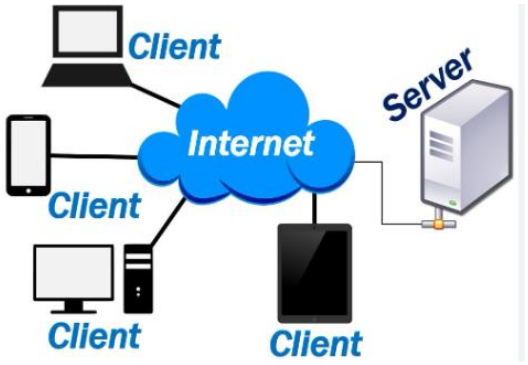
The process begins with Domain Name System (DNS) resolution, where the browser translates the domain name into an IP address to locate the server where the web page is stored.

- 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.

As new content is loaded or changes are made to the web page, the browser updates the display accordingly.

3. What is Server?

A server is a computer system or a software application that provides services, resources, data, or programs to other computers, known as clients, over a network. Servers are essential components of networked computing environments, including the internet, local area networks (LANs), and other distributed systems.



4. what are the types of server available?

- **Web Server**: Serves web pages to users when they visit websites.
- **File Server**: Stores and manages files for users to access over a network.
- **Database Server**: Stores and manages databases, handling data queries and updates.
- **Application Server**: Provides infrastructure for running applications and managing their execution.
- **Mail Server**: Manages email communication, sending and receiving emails.
- **FTP Server**: Facilitates file transfers between computers.
- **Proxy Server**: Acts as an intermediary for clients seeking resources from other servers.
- **DNS Server**: Translates domain names into IP addresses to locate resources on the internet.
- **Virtual Server**: Runs multiple virtual machines on a single physical server.

5. What is SEO? Importance of SEO?

SEO (Search Engine Optimization) is all about making your website more visible and attractive to search engines like Google. When people search for something online, they usually click on the websites that appear at the top of the search results. SEO helps your website get closer to the top of those results.

Importance of SEO:

1. **More Visitors**: SEO helps bring more people to your website because it makes it easier for them to find you when they're searching for something related to your business or content.
2. **Cost-Effective**: It's a cost-effective way to attract visitors because you're not paying for each click like with online ads. Once your website ranks well, you can get continuous traffic without paying more.
3. **Builds Trust**: Being high in search results makes your website look more trustworthy and credible to users. They're more likely to click on your link and explore your site.

4. **Better User Experience**: SEO involves making your website faster, easier to navigate, and more user-friendly. This improves the experience for visitors, which can lead to more engagement and conversions.
5. **Long-Term Results**: Unlike ads that stop showing when you stop paying, SEO can provide long-lasting results. Once you rank well, you can continue to get traffic over time without as much ongoing effort.
6. **Competitive Advantage**: If your competitors are using SEO and you're not, they're likely getting more visitors and customers than you. SEO helps level the playing field and gives you a chance to compete more effectively.
7. **Measurable Results**: You can track your SEO efforts and see exactly how they're helping your website. This allows you to adjust your strategy to improve your results over time.

6. What is Accessibility?

Accessibility means making sure that websites, apps, and digital content are easy to use for everyone, including people with disabilities. It's about designing things so that everyone can access and understand them, no matter how they interact with technology.

Why Accessibility Matters:

1. **Inclusivity**: It ensures that everyone, including people with disabilities, can use and benefit from digital products and services.
2. **Legal and Ethical**: Many laws require websites to be accessible, and it's considered the right thing to do to ensure equal access for all.
3. **Better User Experience**: When websites are accessible, they're easier to use for everyone, not just people with disabilities. Clear navigation and readable content benefit everyone.
4. **Business Benefits**: It can broaden your audience, improve customer satisfaction, and enhance your brand reputation.
5. **Technical Compatibility**: Accessibility also means making sure that websites work well with different devices and assistive technologies like screen readers.

7. What is Markup Language?

A **markup language** is a system for annotating or adding additional information to text in order to define its structure, presentation, or semantics. It uses tags or codes embedded within the text to indicate how the text should be processed or displayed.

Key Points about Markup Languages:

1. **Structure and Presentation**: Markup languages define the structure of content and how it should appear when rendered (displayed) on different devices or platforms.
2. **Tags and Elements**: Tags or elements are the fundamental components of markup languages. They are enclosed in angle brackets (< >) and typically come in pairs—a start tag and an end tag—that surround the content they affect.
3. **Examples of Markup Languages**:
    - **HTML (HyperText Markup Language)**: Used for creating web pages. HTML tags define elements such as headings, paragraphs, links, images, and more.
    - **XML (eXtensible Markup Language)**: Designed to carry data. It allows users to define their own tags and structure to represent complex data in a readable format.
    - **Markdown**: A lightweight markup language with plain-text formatting syntax, often used to format readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor.

4. **Functionality**: Markup languages provide instructions for how text should be processed. They are essential for web development, document formatting, data exchange, and content management.
5. **Human-Readable**: While markup languages include tags that computers can understand, they are designed to be readable by humans as well, making them versatile for both technical and non-technical users.
6. **Evolution and Standards**: Markup languages evolve over time, and there are various standards and specifications that govern their usage, ensuring consistency and interoperability across different systems and applications.

8. What is HTML?

HTML, or HyperText Markup Language, is the basic language used to create websites. It uses tags to define different parts of a web page, like headings, paragraphs, images, and links. These tags tell web browsers how to display content, making it readable and interactive for users.

9. What is browser engine?

A **browser engine**, also known as a layout engine or rendering engine, is a core component of web browsers that interprets and renders HTML documents and other resources like CSS (Cascading Style Sheets) and JavaScript.

Here's a simple explanation:

1. **Interpreting Code**: When you visit a website, your browser engine interprets the HTML, CSS, and JavaScript code that makes up the web page.
2. **Handling Content**: It processes the HTML to understand the structure of the page (headings, paragraphs, images, etc.) and the CSS to style these elements (colors, fonts, layout).
3. **Executing Scripts**: The browser engine also executes JavaScript code embedded in the web page, handling interactions and dynamic content generation.
4. **Rendering**: After processing and interpreting the code, the browser engine renders or paints the content on your screen, displaying it in a way that matches the instructions provided by the HTML and CSS.
5. **Different Engines**: Different web browsers use different browser engines. For example, Chrome and Edge use Blink (a fork of WebKit), Firefox uses Gecko, and Safari uses WebKit.
6. **Optimizations**: Browser engines are optimized to handle web content efficiently, ensuring smooth and responsive user experiences while adhering to web standards.

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

A rendering engine, also known as a layout engine, is a software component responsible for rendering web pages in web browsers. It interprets HTML, CSS, and other resources to display content visually on the screen.

Here's an overview of some of the major rendering engines used in popular web browsers:

1. **Blink**:
    - **Used by**: Google Chrome, Microsoft Edge (Chromium version), Opera, and other Chromium-based browsers.

    - **Description**: Blink is a fork of the WebKit rendering engine, developed by Google. It focuses on speed, stability, and supporting modern web standards.
2. **WebKit**:
    - **Used by**: Safari, formerly used by Google Chrome and other browsers.
    - **Description**: WebKit is an open-source rendering engine originally developed by Apple. It includes the WebCore layout engine and is known for its rendering accuracy and performance.
3. **Gecko**:
    - **Used by**: Mozilla Firefox.
    - **Description**: Gecko is developed by Mozilla and is known for its flexibility, robustness, and adherence to web standards. It powers Firefox's rendering capabilities.
4. **Trident** (Legacy):
    - **Used by**: Older versions of Microsoft Internet Explorer.
    - **Description**: Trident was developed by Microsoft and was the rendering engine used in Internet Explorer. It's no longer actively developed or supported.
5. **EdgeHTML** (Legacy):
    - **Used by**: Legacy versions of Microsoft Edge (replaced by Chromium-based Edge).
    - **Description**: EdgeHTML was developed by Microsoft for the original Microsoft Edge browser. It aimed to improve performance and compatibility with modern web standards.

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

A **JavaScript engine** is a software component that interprets and executes JavaScript code. JavaScript is a high-level, interpreted programming language primarily used for creating interactive and dynamic content on web pages.

JavaScript Engine:

1. **Purpose**:
    - **Execution**: JavaScript engines are designed to parse, interpret, and execute JavaScript code. They handle tasks such as variable assignments, function calls, object manipulations, and more.
    - **Performance**: They aim to optimize code execution for speed and efficiency, allowing web applications to run smoothly and responsively.
    - **Compatibility**: JavaScript engines ensure that JavaScript code behaves consistently across different browsers and platforms.
2. **Available JavaScript Engines**:
    - **V8**:
        - **Used by**: Google Chrome, Chromium-based browsers (like Microsoft Edge, Opera).
        - **Description**: V8 is an open-source JavaScript engine developed by Google. It compiles JavaScript directly into machine code for faster execution.
    - **SpiderMonkey**:
        - **Used by**: Mozilla Firefox.
        - **Description**: SpiderMonkey is Mozilla's JavaScript engine. It was one of the first JavaScript engines and is known for its flexibility and adherence to web standards.

- **JavaScriptCore (Nitro)**:
  - **Used by**: Safari (Apple's web browser).
  - **Description**: JavaScriptCore, also known as Nitro, is the JavaScript engine developed by Apple. It is designed for optimal performance on macOS and iOS devices.
- **Chakra (Legacy)**:
  - **Used by**: Legacy versions of Microsoft Edge.
  - **Description**: Chakra was developed by Microsoft for EdgeHTML, the original rendering engine of Microsoft Edge. It focused on optimizing JavaScript performance and compatibility.
- **Rhino**:
  - **Description**: Rhino is an open-source JavaScript engine developed by Mozilla. It is written in Java and is used primarily for embedding JavaScript in Java applications.

Purpose of JavaScript Engines:

JavaScript engines play a crucial role in the functionality and performance of web browsers and other environments where JavaScript is used.

Their main purposes include:

- **Interpreting and Executing Code**: JavaScript engines parse JavaScript code, interpret it, and execute it to perform tasks specified by the code.
- **Optimizing Performance**: They optimize code execution to improve speed and efficiency, enabling web applications to run faster and more responsively.
- **Ensuring Compatibility**: JavaScript engines ensure that JavaScript code behaves consistently across different browsers and platforms, maintaining compatibility and reliability for developers and users alike.

12. How website works?

1. **Domain Name**: You type a website's name (like www.example.com) into your browser.
2. **DNS Lookup**: Your computer finds the IP address (like 192.0.2.1) of the website's server using DNS (Domain Name System).
3. **HTTP Request**: Your browser sends a request to that server asking for the web page you want (like index.html).
4. **Web Server**: The server gets the request, finds the web page, and sends it back to your browser.
5. **Browser Processing**: Your browser reads the HTML, CSS, and JavaScript in the web page to show you text, images, videos, and more.
6. **Rendering**: The browser combines all these elements and displays the web page on your screen.
7. **Interaction**: You can click links, fill out forms, and interact with the page. Your browser may send more requests to the server for new content or updates.
8. **Closing**: When you leave the website or close your browser, connections to the server are closed.

13. What is Data Structure?

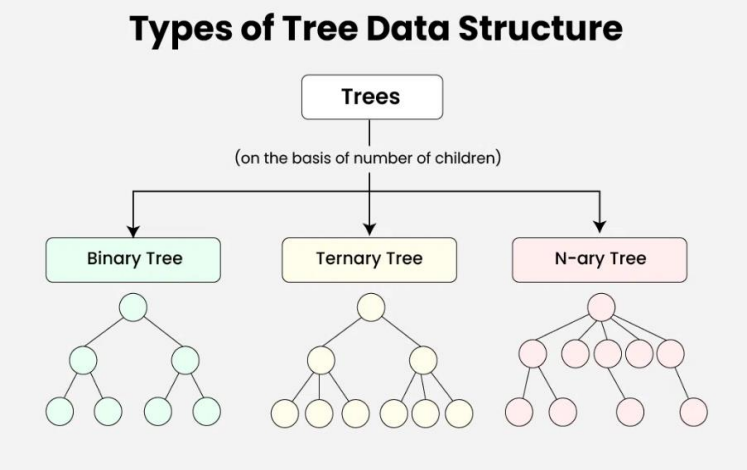
A **data structure** is a way of organizing and storing data in a computer so that it can be accessed and used efficiently. It defines a format for storing and organizing data values, relationships, and operations that can be performed on the data.

Types of Data Structures:

- **Arrays**: A collection of elements stored at contiguous memory locations.
- **Linked Lists**: Elements (nodes) linked together in a sequence where each node contains data and a reference (link) to the next node.
- **Stacks**: Follows the Last In First Out (LIFO) principle, where elements are added and removed from the top.
- **Queues**: Follows the First In First Out (FIFO) principle, where elements are added at the rear and removed from the front.
- **Trees**: Hierarchical structure with nodes starting from a root node, branching out into parent-child relationships.
- **Graphs**: A collection of nodes (vertices) and edges that connect these nodes, representing relationships.
- **Hash Tables**: Data structure that stores data in an associative manner, using keys to access values directly.

14. Explain Tree Data Structure?

Tree data structure is a hierarchical structure that is used to represent and organize data in a way that is easy to navigate and search. It is a collection of nodes that are connected by edges and has a hierarchical relationship between the nodes.



1. Binary Tree:

- A binary tree is a hierarchical data structure where each node has at most two children, referred to as the left child and the right child.

2. Ternary Tree:

- A ternary tree is a tree data structure where each node can have up to three children.

3. N-ary Tree:

- An N-ary tree is a tree data structure where each node can have any number of children, not limited to two or three.

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

A **user agent** in the context of web browsing refers to the software application or device that sends requests to servers on behalf of the user. It identifies itself to servers, typically through an HTTP header, providing information about the browser, operating system, and device characteristics.

Purpose of User Agent:

1. **Communication**: It helps your browser talk to websites by sharing information about its type (like Chrome or Safari), version (like 91.0), and operating system (like Windows or iOS).
2. **Optimization**: Websites use this information to show you web pages that look good and work well on your specific device. For example, a mobile site might look different from a desktop site based on your user agent.
3. **Analytics**: Websites use user agents to gather data about what browsers and devices people use most. This helps them improve their websites and services.

Examples of User Agents:

- Google Chrome (Desktop)**:**
- Apple Safari (Mac)**:**
- iOS Safari (iPhone)**:**

16. What is Hypertest?

Hypertext is a system of organizing information in a non-linear format that allows users to navigate and access related pieces of text or multimedia content through hyperlinks.

Examples: Web pages on the internet are examples of hypertext, where links (hyperlinks) connect different web pages, images, videos, and other types of content.

17. What is HTML Tags?

HTML (Hypertext Markup Language) tags are elements used to define the structure and content of web pages. They are the building blocks of HTML documents and instruct web browsers on how to display and render content.

HTML tags are keywords or code snippets surrounded by angle brackets < >. They specify elements that define the structure and appearance of content on a web page.

Syntax:

- Tags typically come in pairs: an opening tag `<tag>` and a closing tag `</tag>`.
- The opening tag denotes the beginning of an element, and the closing tag signifies the end. Content is placed between the opening and closing tags.

Common HTML Tags:

- **`<html>`**: Defines the root of an HTML document.

- **`<head>`**: Contains metadata about the document, such as its title and links to stylesheets.
- **`<title>`**: Sets the title of the document that appears in the browser's title bar or tab.
- **`<body>`**: Contains the main content of the document that is displayed in the browser window.
- **`<p>`**: Defines a paragraph.
- **`<a>`**: Defines a hyperlink.
- **`<img>`**: Embeds an image.
- **`<div>`**: Defines a division or section in a document.
- **`<table>`**: Defines a table.
- **`<form>`**: Defines an HTML form for user input.

18. What is HTML Attributes?

Attributes are additional properties or parameters that can be applied to HTML elements to customize their behavior or appearance.

Syntax:

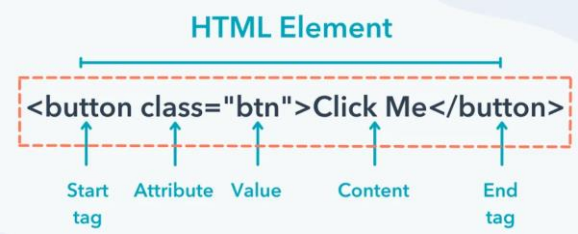
Attributes are specified within the opening tag of an HTML element and typically come in name-value pairs, separated by an equal sign `=`. The syntax is: `attribute_name="value"`.

Example:

Visit Example Website

19. What is HTML Elements?

The HTML **element** is everything from the start tag to the end tag.



20. How do convert elements to tree?

React HTML parser works by parsing the HTML string into a tree of nodes, with each node representing an HTML element. The parser traverses the generated node tree, converting each node into a corresponding React element. The result is a tree of React elements that mirrors the structure of the original HTML.

21. What is DOCTYPE?

HTML Doctypes declared at the beginning of HTML documents, inform browsers about the document type and version, ensuring correct rendering. They aren't HTML tags but provide essential information. The most common doctype is **<!DOCTYPE html>**, used for HTML5, ensuring modern web standards.

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

Text Editor: Use software like Notepad (Windows), TextEdit (Mac), Sublime Text, or Visual Studio Code. Write or paste IITML code, then save with a .html extension.

Web Development IDE: Tools such as Visual Studio, WebStorm, or Dreamweaver allow crcation and saving of HTML files within their environments, typically through a "Save" option.

Browser: Open the HTML. file in a web browser (e.g., Chrome, Firefox, Safari), then use the browser's "Save As" function to save the webpage locally with a .html extension.

Command Line: Navigate to the directory via terminal, use text editors like nano or vim to create/edit HTML files, and save with commands (Ctrl + O in nano, wq in vim).

23. What is charset? Why do we need to use this?

HTML. Charset is also called HTML Character Sets or HTMI. Encoding. The term "charset" refers to the character encoding scheme used to map character codes (such as ASCII or Unicode) to actual characters that can be displayed or processed by a computer system. It determines how textual data is represented in binary form.

The HTML5 specification encourages web developers to use the UTF-8 character set, which covers almost all of the characters and symbols

Example: meta charset="UTF-8">

Need of charset

Consistent Text: Ensures characters appear the same across different devices and software.

Multilingual Support: Allows computers to represent a vast range of languages.

Accurate Display: Guarantees characters on a web page or document match what you intended.

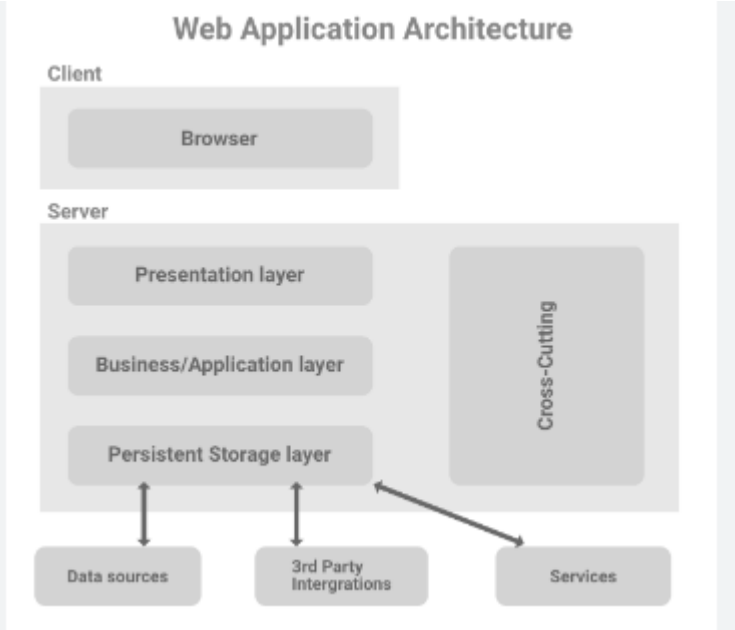
Universal Understanding: Acts like a common language for computers to interpret text.

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

Meta data means "data about data", is essentially information that describes and explains other data. Metadata in web development refers to hidden data in HTML that describes a webpage but isn't displayed to users.

25. Explain Web Application Architecture?

Web application architecture is a mechanism that gives us a clarification on how the connection is established between the client and the server.



Web applications are typically built using a layered approach.It typically consists of 3 layers as follows.

1. **Presentation Layer**: This layer is responsible for presenting data to the user. It handles the rendering of the user interface, which might involve HTML, CSS, and JavaScript. It processes client requests and sends appropriate responses back to the client.
2. **Business/Application Layer**: The core logic of the application resides here. This layer processes user inputs, makes logical decisions, and performs computations. It implements the business rules and workflows.

3. **Persistent Storage Layer**: This layer manages data storage and retrieval. It interacts with databases or other storage systems to ensure data is saved and can be queried as needed.