

# 200 Interview Questions

## FSD INTERVIEW QUESTIONS

### 1. What is Full Stack development?

Full Stack development involves developing both the front end and back end of the web application/website at the same time. This process includes three layers:

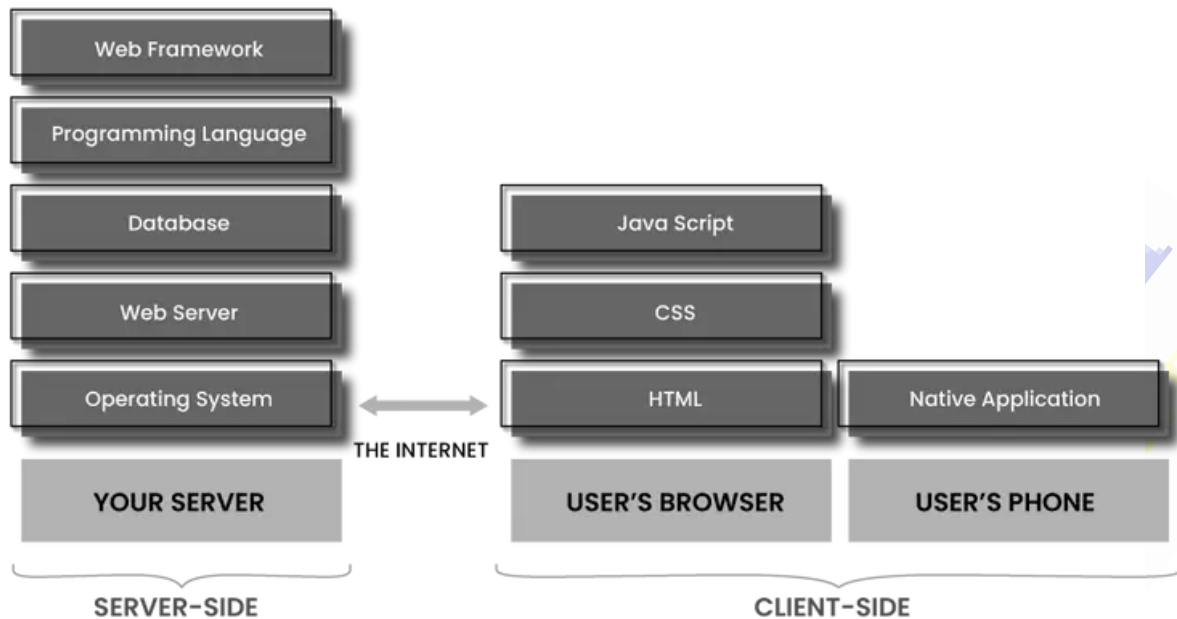
- Presentation layer (frontend part responsible for user experience)
- Business logic layer (backend part refers to the server side of the application)
- Database layer.

### 2. What do Full Stack Web Developers do?

A Full Stack Web Developer is a person who is familiar with developing both client and server software. In addition to mastering CSS and HTML, they are also know how to program browsers, databases, and servers.

To fully comprehend the role of Full Stack developer, you must understand the web development components - front end and back end

The front end comprises a visible part of the application in which the user interacts, while the back end includes business logic.



### 3. Name a few Full Stack developer tools?

Some of the popular tools used by full-stack developers to make development more accessible and efficient are:

- **Backbone**
- **Visual Studio Code**
- **WebStorm**
- **Slack**
- **Electron**
- **TypeScript**
- **CodePen**
- **GitHub**

4. **What skills do you need to be a full-stack developer?**

Basic languages - Must be proficient in basic languages like HTML, CSS, and SQL.

Front-end frameworks - Bootstrap, AngularJS, VueJS, [ReactJS](#), JavaScript, TypeScript, Python, Ruby, PHP

Back-end frameworks - Express, Django, NodeJS, Ruby on Rails

Databases - MySQL, SQLite, Postgres, [MongoDB](#), Cassandra, Apache storm, Sphinx

Additional skills recommended - Git, Machine Learning, SSH, Linux Command, [Data Structures](#), Character encoding.

5. **Explain Pair Programming?**

As the name suggests, Pair Programming is where two programmers share a single workstation. Formally, one programmer at the keyboard called the "driver" writes the code. The other programmer is the "navigator" who views each line of the code written, spell check, and proofread it. Also, programmers will swap their roles every few minutes and vice-versa.

6. **What is CORS?**

Cross-origin resource sharing (CORS) is a process that utilizes additional HTTP headers to tell browsers to provide a web application running at one origin. CORS accesses various web resources on different domains. Web scripts can be integrated using CORS when it requests a resource that has an external origin (protocol, Domain, or port) from its own.

7. **What is Inversion of Control (IoC)?**

Inversion of Control (IoC) is a broad term used by software developers for defining a pattern that is used for decoupling components and layers in the system. It is mostly used in the context of object-oriented programming.

Control of objects or portions of a program is transferred to a framework or container with the help of Inversion of Control. It can be achieved using various mechanisms such as service locator pattern, strategy design pattern, factory pattern, and dependency injection.

8. **What is Continuous Integration?**

Continuous Integration (CI) is a practice where developers integrate code into a shared repository regularly to detect problems early. CI process involves automatic tools that state new code's correctness before integration. Automated builds and tests verify every check-in.

9. **How is GraphQL different from REST?**

This is typically a difficult question to answer, but a good developer will be able to go through this with ease. The core difference is GraphQL doesn't deal with dedicated resources. The description of a particular resource is not coupled to the way you retrieve it. Everything referred to as a graph is connected and can be queried to application needs

10. **What is the Observer pattern?**

The purpose of the Observer pattern is to define a one-to-many dependency between objects, as when an object changes the state, then all its dependents are notified and updated automatically. The object that watches on the state of another object is called the observer, and the object that is being watched is called the subject.

## 11. What is REST API?

REST stands for representational state transfer. A RESTful API (also known as REST API) is an architectural style for an application programming interface (API or web API) that uses HTTP requests to obtain and manage information. That data can be used to POST, GET, DELETE, and OUT data types, which refers to reading, deleting, creating, and operations concerning services.

## 12. What is HTML?

HTML stands for HyperText Markup Language and is the language of the internet. It is the standard text formatting language used for creating and displaying pages on the Internet. HTML documents are made up of the elements and the tags that format it for proper display on pages.

## 13. What are HTML tags?

We use HTML tags for placing the elements in the proper and appropriate format. Tags use the symbols <, and > to set them apart from the HTML content.

The **HTML tags** need not be closed always. For example, in the case of images, the closing tags are not required as <img> tag.

## 14. What are HTML Attributes?

Attributes are the properties that can be added to an HTML tag. These attributes change the way the tag behaves or is displayed. For example, a <img> tag has an src attribute, which you use to add the source from which the image should be displayed.

We add attributes right after the name of the HTML tag, inside the brackets. We can only add the attributes to opening or self-closing tags, but never be in closing tags.

## 15. What is a marquee in HTML?

Marquee is used for scrolling text on a web page. It scrolls the image or text up, down, left, or right automatically. To apply for a marquee, you have to use `</marquee>` tags.

## 16. Define the list types in HTML?

The list types in HTML are as below:

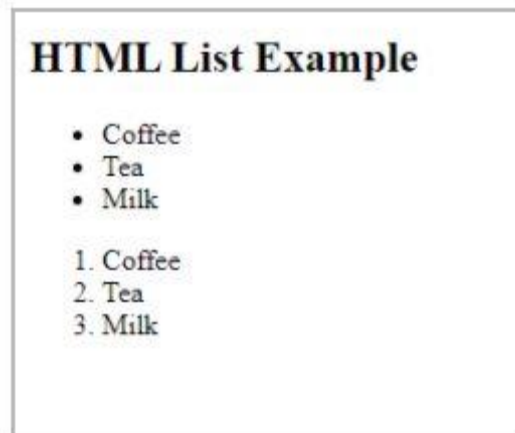
- **Ordered list**–The ordered list uses `<ol>` tag and displays elements in a numbered format.
- **Unordered list**–The unordered list uses `<ul>` tag and displays elements in a bulleted format.
- **Definition list**–The definition list uses `<dl>`, `<dt>`, `<dd>` tags and displays elements in definition form like in a dictionary.

## 17. Differentiate between an Ordered list and an Unordered list?

An unordered list uses `<ul>` `</ul>` tags and each element of the list is written between `<li>` `</li>` tags. The list items are displayed as bullets rather than numbers.

An ordered list uses `<ol>` `</ol>` tags and each element of the list is written between `<li>` `</li>` tags. The list items are displayed as numbers rather than bullet points.

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML List Example</h2>
    <ul>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ul>
    <ol>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ol>
  </body>
</html>
```



**18. How to create a hyperlink in HTML?**

We use the anchor tag `<a>` to create a hyperlink in HTML that links one page to another page.

syntax:

```
<a href="www.novitech.com">NoviTech</a>
```

**19. Which HTML tag is used to display the data in the tabular form?**

**table :** In HTML “table” tag is used to define a table.

**tr :** For defining table rows, the “tr” tag is used.

**th :** “th” stands for a table header, for defining table header “th” tag is used. A table header is bold and centred.

**td :** To insert or add data in each cell, Table Data “td” tag is used.

**20. What is the Box model in CSS? Which CSS properties are a part of it?**

A rectangle box is wrapped around every HTML element. The box model is used to determine the height and width of the rectangular box.

The CSS Box consists of Width and height (or in the absence of that, default values and the content inside), padding, borders, margin.

**Content:** Actual Content of the box where the text or image is placed.

**Padding:** Area surrounding the content (Space between the border and content).

**Border:** Area surrounding the padding.

**Margin:** Area surrounding the border.

**21. What are the advantages of using CSS?**

**Separation of content from presentation** - CSS provides a way to present the same content in multiple presentation formats in mobile or desktop or laptop.

**Easy to maintain** - CSS, built effectively can be used to change the look and feel complete by making small changes. To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

**Bandwidth** - Used effectively, the style sheets will be stored in the browser cache and they can be used on multiple pages, without having to download again.

**22. What are the limitations of CSS?**

**Browser Compatibility:** Some style selectors are supported and some are not. We have to determine which style is supported or not using the @support selector).

**Cross Browser issue:** Some selectors behave differently in a different browser).

**There is no parent selector:** Currently, Using CSS, you can't select a parent tag.

**23. How to include CSS in the webpage?**

**1 - External Style Sheet:** An external file linked to your HTML document:

Using link tag, we can link the style sheet to the HTML page.

```
<link rel="stylesheet" type="text/css" href="mystyles.css" />
```

**2 - Embed CSS with a style tag:** A set of CSS styles included within your HTML page.

```
<style type="text/css">
```



```
/*Add style rules here*/
```

```
</style>
```

Add your CSS rules between the opening and closing style tags and write your CSS exactly the same way as you do in stand-alone stylesheet files.

3 - Add inline styles to HTML elements(CSS rules applied directly within an HTML tag.): Style can be added directly to the HTML element using a style tag.

```
<h2 style="color:red;background:black">Inline Style</h2>
```

4 - Import a stylesheet file (An external file imported into another CSS file):

Another way to add CSS is by using the @import rule. This is to add a new CSS file within CSS itself.

```
@import "path/to/style.css";
```

24. Differentiate between CSS3 and CSS2.

The main difference between **CSS3** and **CSS2** is that CSS divides different sections into modules and supports many browsers. It also contains new General Sibling Combinators responsible for matching similar elements.

25. List the CSS Frameworks.

CSS frameworks are libraries that make web page styling easier. Some of them are Foundation, Bootstrap, Gumby, Ukit, Semantic UI, etc.

26. What is meant by RGB stream?

RGB represents colors in CSS. The three streams are namely Red, Green, and Blue. The intensity of **colors** is represented using numbers 0 to 256. This allows CSS to have a spectrum of visible colors.

27. Define z-index.

This is one of the most frequently asked CSS interview questions. Z-index is used to specify the stack order of elements that overlap each other. Its default value is zero and can take both negative and positive values. A higher z-index value is stacked above the lower index element. It takes the following values- auto, number, initial, and inherit.

28. What is the difference between inline, inline-block, and block?

Block Elements are `<div>` and `<p>`. They usually start on a new line and can take space for an entire row or width.

Inline elements are `<a>`, `<span>`, `<strong>`, and `<img>` tags. They don't start on a new line. However, they appear on the same line as the content and tags beside them.

Inline block elements have padding and margins and set height and width values. Though, they are similar to inline elements.

29. What are Pseudo classes?

Pseudo-classes are the type of pseudo-elements that don't exist in a normal document tree. It allows selecting the regular elements under certain conditions especially when we try to hover over the link; the anchor tags are `:link`, `:visited`, `:hover`, `:active`, `:focus`

In this example, the color will be red on the anchor tag when it's hovered.

```
/* mouse over link */
```

```
a:hover {
```

```
color: #FF00FF;
```

```
}
```

30. What is VH/VW (viewport height/ viewport width) in CSS?

**VH and VW are CSS units used to measure viewport height and viewport width respectively in percentage form in the responsive design techniques. E.g. If the height of the browser is 1000px, then VH is 1/100 of the height of the viewport that is  $1000\text{px} \times (1/100) = 10\text{px}$ , which is the height of the browser. The same applies to VW (viewport width).**

31. How is the border-box different from the content box?

**Border-box consists of properties such as content, padding, and the border with respect to height and width. However, Content-box is a default value property used for the box-sizing as well as it helps to add border and padding to give proper height and width to the box without having a border and padding properties**

32. What is the grid system?

**The CSS grid system is a type of powerful layout of 2 dimensional systems with respect to columns and rows.**

33. How does absolute positioning work?

**Absolute positioning is used to place the element which is then removed from the HTML document from the normal workflow without creating any space for the element in the HTML page layout. The element can be positioned respectively to the closest positioned ancestor; otherwise, if the ancestor is not found, the element is placed with respect to the initial container box. The values provided to the top, right, left and bottom determine the final position of the element.**

**34. What does the CSS float property do?**

The float property of CSS positions an image to the right or left as needed, including text wrapping around it. All properties of elements used before it remain unchanged.

**35. What are the properties of flexbox?**

The properties of flexbox are flex-direction, wrap, flow, content, and align-items, and content.

**36. What is JavaScript?**

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

**37. What are the different data types present in javascript?**

**1. Number**

**2. String**

**3. Boolean**

**4. Undefined**

**5. Null**

**6. Symbol (ES6 and later)**

**7. Object**

**8. BigInt (ES11 and later)**

**9. Function**

**38. Explain Hoisting in javascript.**

Hoisting is a JavaScript behavior where variable and function declarations are moved to the top of their containing scope during the compilation phase. This allows you to use variables and functions before they are declared in the code.

**39. Difference between var and let keyword in javascript.**

var has a function-level scope, while let has a block-level scope.

Variables declared with var are hoisted to the top of their scope, while let variables are hoisted but not initialized.

let allows you to create block-scoped variables, which helps prevent issues related to variable hoisting and redeclaration.

**40. What is the difference between == and === operators**

== is the equality operator that performs type coercion, allowing comparison of values of different types after converting them to a common type.

**===** is the strict equality operator that checks both value and type, ensuring that values are of the same type without performing type coercion.

**41. Name the types of functions**

**Function Declaration**

**Function Expression**

**Arrow Function**

**IIFE (Immediately Invoked Function Expression)**

**42. What is the difference between JavaScript and JScript?**

JavaScript is the scripting language standardized by ECMA International and widely supported in web browsers.

JScript is Microsoft's implementation of ECMAScript, used primarily in Internet Explorer.

**43. How to write a hello world example of JavaScript?**

```
console.log('Hello, World!');
```

**44. What is scope in javascript**

Scope refers to the visibility and accessibility of variables in a particular context during runtime.

JavaScript has two types of scope: global scope and local scope.

Variables declared with var have function-level scope, while those declared with let and const have block-level scope.

**45. What are the key differences between Java and JavaScript? / How is JavaScript different from Java?**

Java is a programming language used for server-side development, while JavaScript is a scripting language primarily used for client-side web development.

Java requires compilation, while JavaScript is an interpreted language.

Java is statically typed, whereas JavaScript is dynamically typed.

Java is primarily used for building standalone applications and enterprise-level systems, while JavaScript is widely used for enhancing the interactivity of web pages.

**46. Is JavaScript case sensitive language?**

Yes, JavaScript is a case-sensitive language.

**47. What is DOM? What is the use of document object?**

The DOM is a programming interface for web documents.

It represents the structure of a document as a tree of objects, where each object corresponds to a part of the document.

The Document object represents the entire HTML or XML document.

**48. What is the use of window object?**

The document object in JavaScript represents the web page or document loaded in the browser.

It provides methods and properties to interact with the document, such as modifying its content, structure, and styles.

**49. How to write HTML code dynamically using JavaScript?**

**// Example: Dynamically create a paragraph element and append it to the body**

```
var paragraph = document.createElement('p');
```

```
var text = document.createTextNode('Hello, dynamically created HTML!');
```

```
paragraph.appendChild(text);
```

```
document.body.appendChild(paragraph);
```

**50. How to create an array in JavaScript?**

**// Example: Creating an array**



```
var fruits = ['apple', 'orange', 'banana'];
```

**51. Difference between Client side JavaScript and Server side JavaScript**

Client-side JavaScript runs in a user's web browser and is used for enhancing the interactivity of web pages.

Server-side JavaScript runs on the server and is used for tasks such as handling requests, accessing databases, and performing server-side logic.

**52. What is the real name of JavaScript?**

The real name of JavaScript is ECMAScript.

**53. What is the difference between undefined value and null value?**

undefined is a variable that has been declared but has not been assigned a value.

null is an assignment value representing the intentional absence of any object value.

**54. What is the requirement of debugging in JavaScript?**

Debugging in JavaScript is necessary to identify and fix errors or bugs in the code.

It helps developers understand the flow of the program, inspect variable values, and catch unexpected behaviors.

**55. What is ReactJS?**

ReactJS is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and efficiently update and render components based on changes in data.

**56. Why ReactJS is Used?**

ReactJS is used for building user interfaces, especially for single-page applications where dynamic content and seamless updates are crucial. It allows developers to create reusable components, making the code modular and maintainable. React facilitates efficient rendering of components by using a virtual DOM, improving performance.

**57. What are the features of ReactJS?**

**Virtual DOM:** Efficiently updates and renders components for improved performance.

**Component-Based Architecture:** Encourages reusability and maintainability.

**Unidirectional Data Flow:** Ensures predictable data flow within the application.

**JSX:** Provides a syntax extension for JavaScript, making it easier to write UI components.

**React Hooks:** Allows the use of state and other React features in functional components.

**58. How to create components in ReactJS?**

Components in ReactJS can be created using either class components or functional components.

Example of a functional component:

```
import React from 'react';
```

```
const MyComponent = () => {  
  return <div>Hello, React!</div>;  
}
```

```
};
```

```
export default MyComponent;
```

**59. What is JSX?**

JSX (JavaScript XML) is a syntax extension for JavaScript.

It allows the mixing of HTML-like code within JavaScript, making it easier to write React components.

Example:

```
const element = <h1>Hello, JSX!</h1>;
```

**60. How is React different from React Native?**

**React:** Used for building web applications. Components are rendered in the browser.

**React Native:** Used for building mobile applications. Components are rendered using native APIs for iOS and Android platforms.

React Native allows the use of React components for building mobile UIs, providing a bridge between JavaScript and native code.

**61. What is the virtual DOM?**

The virtual DOM (Document Object Model) is a lightweight, in-memory representation of the actual DOM.

React uses the virtual DOM to efficiently update and render components by minimizing direct manipulation of the actual DOM, which can be a resource-intensive operation.

**62. Why use React instead of other frameworks, like Angular?**

React's component-based architecture promotes reusability and maintainability.

It provides a more flexible and efficient approach to rendering through the use of a virtual DOM.

React is often chosen for its simplicity, allowing developers to integrate it into existing projects without significant rewrites.

**63. What is an event in React?**

In React, events are interactions triggered by users, such as clicking a button or typing in an input field.

Events in React are handled using synthetic event wrappers, providing consistent behavior across different browsers.

Example:

```
<button onClick={handleClick}>Click me</button>
```

**64. What are forms in React?**

Forms in React are used to handle user input.

React provides a controlled component approach, where form elements are controlled by React state.

```
const [inputValue, setInputValue] = useState("");
```

```
const handleChange = (e) => {  
  setInputValue(e.target.value);  
};
```

```
<input type="text" value={inputValue} onChange={handleChange} />
```

**65. How is React different from Angular?**

React:

- Library for building user interfaces.
- Uses a virtual DOM for efficient rendering.
- Follows a unidirectional data flow.
- Provides JSX for writing components.

Angular:

- Full-fledged framework for building web applications.
- Two-way data binding is a core feature.
- Uses a real DOM.
- Requires a steeper learning curve compared to React.

**66. What is the use of render() in React?**

The render() method in React is responsible for rendering the component's UI.

It returns a description of what the UI should look like, based on the component's current state and props.

**67. What is a state in React?**

State in React is an object that represents the internal data of a component. Changes to the state trigger re-rendering of the component, updating the user interface.

State is mutable and managed using the `setState()` method.

**68. What are props in React?**

Props (short for properties) are inputs that are passed into a React component.

Props allow data to be passed from a parent component to a child component.

Props are immutable and should not be modified within the component.

**69. What are the differences between state and props?**

State:

- Managed within the component.
- Can be changed using `setState()`.
- Local to the component.

Props:

- Received from a parent component.
- Immutable and should not be changed.
- External to the component.

**70. What are the differences between class and functional components?**

Class Components:

- Can have state and lifecycle methods.
- Defined using ES6 class syntax.
- Requires the use of `this` keyword to access props and state.

Functional Components:

- Primarily used for presenting UI.
- Introduced in React as a simpler way to write components.
- Cannot have state or lifecycle methods until the introduction of Hooks in React 16.8.

**71. Explain the difference between frontend and backend development?**

**Frontend Development:**

- Focuses on the user interface and user experience of a website or application.
- Involves building and designing elements that users interact with directly.
- Technologies include HTML, CSS, and JavaScript.

**Backend Development:**

- Concerned with server-side operations, databases, and the overall logic of the application.
- Involves server-side scripting, database management, and server configuration.
- Technologies include server-side languages like Node.js, Python, PHP, and databases like MySQL or MongoDB.

**72. Explain the purpose of the backend?**

The backend serves to handle and process data, perform business logic, and interact with databases.

It manages the server, authentication, authorization, and ensures the overall functionality of an application.

**73. What is the difference between a GET and a POST request?**

**GET Request:**

- Retrieves data from a specified resource.
- Parameters are included in the URL.
- Limited data transfer capacity.
- Cached by browsers.

**POST Request:**

- Submits data to be processed to a specified resource.
- Parameters are sent in the request body.
- Suitable for larger data transfers.
- Not cached by browsers.

**74. What is a web server?**

A web server is software that handles HTTP requests from clients, usually web browsers.

It processes these requests, retrieves the requested resources, and sends them back to the client.

Examples include Apache, Nginx, and Microsoft IIS.

**75. What is API?**

An API is a set of rules and tools that allows different software applications to communicate with each other.

It defines the methods and data formats that applications can use to request and exchange information.

APIs are used to enable the integration of different systems and services.

**76. What is the difference between API and RESTful API?**

**API (Application Programming Interface):**

- A general term referring to a set of rules allowing different software applications to communicate.
- It can follow various architectural styles, including RESTful principles.

**RESTful API:**

- A specific type of API adhering to the principles of Representational State Transfer (REST).
- Utilizes standard HTTP methods (GET, POST, PUT, DELETE) for communication and resource manipulation.

**77. What is Node.js? Where can you use it?**

**Definition:**

- Node.js is a server-side JavaScript runtime environment.

**Use:**

- It is used for building scalable and high-performance network applications.
- Commonly used for developing server-side applications, especially web servers.



## 78. Why use Node.js?

### Non-blocking I/O:

- Handles concurrent connections efficiently without blocking the execution of code.

### JavaScript:

- Enables full-stack development using a single programming language.

### Vibrant Ecosystem:

- Large community and a rich ecosystem of packages (npm) for easy integration of libraries and tools.

## 79. How does Node.js work?

### Event-Driven and Asynchronous:

- Operates on an event-driven, non-blocking model.
- Utilizes callbacks to handle asynchronous operations.

### Single-Threaded:

- Uses a single-threaded event loop to handle multiple concurrent connections efficiently.
- Delegates I/O operations to the operating system, freeing the main thread.

## 80. Explain callback in Node.js.

### Definition:

- A callback in Node.js is a function passed as an argument to another function, to be executed later when a specific operation has been completed.

### Purpose:

- Used to handle asynchronous operations, ensuring that the code doesn't wait for the operation to complete before moving on.
- Enables non-blocking I/O in Node.js.

## 81. What is NPM?

NPM is the default package manager for Node.js.

It allows developers to discover, install, and manage packages (libraries and tools) for their Node.js projects.



**82. What are the modules in Node.js?**

Modules in Node.js are encapsulated units of functionality that can be reused in different parts of a program.

Common types include core modules (built-in), local modules (user-created), and third-party modules (installed via NPM).

**83. Is Node.js cross-platform?**

Yes, Node.js is cross-platform.

It can run on various operating systems, including Windows, macOS, and Linux, providing a consistent development environment.

**84. Does Node.js use JavaScript?**

Yes, Node.js uses JavaScript as its primary programming language. It extends the use of JavaScript from the browser to server-side development.

**85. Name types of API functions supported by Node.js.**

Node.js supports various types of API functions, including:

File System API: For file-related operations.

HTTP API: For creating web servers and handling HTTP requests.

Net API: For creating network-related applications.

Crypto API: For cryptographic operations.

Process API: For interacting with the system's processes.

**86. What is a database?**

A database is a structured collection of data organized for efficient storage, retrieval, and management.

**87. What is a database system?**

A database system is a software application or suite of applications that facilitates the creation, management, and manipulation of databases.

**88. What is RDBMS?**

RDBMS is a type of database management system that organizes data into tables with rows and columns, establishing relationships between them.

**89. How many types of database languages are?**

Two main types of database languages are:

**Data Definition Language (DDL):** Defines and manages the structure of the database, including creating, altering, and deleting tables and relationships.

**Data Manipulation Language (DML):** Deals with the manipulation of data stored in the database, including querying, inserting, updating, and deleting records.

**90. What is MongoDB ?**

MongoDB is a NoSQL database management system.

It stores data in flexible, JSON-like documents, allowing for dynamic and scalable data schemas.

MongoDB is known for its ability to handle large amounts of unstructured or semi-structured data and its horizontal scalability.

**91. What is a Document in MongoDB?**

In MongoDB, a document is a basic unit of data storage.

It is a JSON-like BSON (Binary JSON) object that contains key-value pairs, where values can be various data types, including arrays and other documents.

Documents are organized into collections within MongoDB.

**92. What is the Mongo Shell?**

The Mongo Shell is a command-line interface for interacting with MongoDB.

It allows users to connect to a MongoDB server, execute queries, and perform administrative tasks using JavaScript-like commands.

**93. What does SQL in MySQL stand for?**

In MySQL, SQL stands for "Structured Query Language."

SQL is a standard language used for managing and manipulating relational databases.

**94. What are some of the common MySQL commands?**

**SELECT:** Retrieve data from one or more tables.

**INSERT:** Add new records into a table.

**UPDATE:** Modify existing records in a table.

**DELETE:** Remove records from a table.

**CREATE TABLE:** Create a new table with specified columns and data types.

**ALTER TABLE:** Modify the structure of an existing table.

**DROP TABLE:** Delete a table and its data.

**SHOW DATABASES:** Display a list of available databases.

**95. What is the MySQL server?**

The MySQL server is a relational database management system (RDBMS) that stores and manages structured data.

It handles client requests, executes SQL queries, and manages databases and tables.

**96. What are the data types of MongoDB?**

MongoDB supports various data types, including:

- String
- Integer
- Double
- Boolean
- Object
- Array
- Null
- Date
- ObjectId (a 12-byte identifier typically employed as a primary key)

**97. Differentiate MongoDB and MySQL?**

**MongoDB:**

- NoSQL database.
- Document-oriented.

- Schema-free.
- Uses BSON (Binary JSON) for data representation.
- Horizontal scalability.

**MySQL:**

- SQL database.
- Relational database management system (RDBMS).
- Schema-based.
- Uses tables with predefined schemas.
- Vertical scalability.

**98. Difference between SQL and NOSQL?**

**SQL (Structured Query Language):**

- Relational databases (RDBMS).
- Table-based data representation.
- Predefined schema.
- ACID properties (Atomicity, Consistency, Isolation, Durability).

**NoSQL (Not Only SQL):**

- Non-relational databases.
- Various data models (document, key-value, wide-column, graph).
- Schema-less or dynamic schema.
- CAP theorem (Consistency, Availability, Partition tolerance).

**99. What do you mean by Data Model?**

A data model is a conceptual representation of data and the relationships between different elements within a system.

It defines how data is organized and stored, providing a blueprint for designing databases.

**100. What is server?**

In a general computing context, a server is a computer or software application that provides services or resources to other computers, known as clients, over a network.

In the context of databases, a server often refers to the machine or software responsible for managing and responding to database requests, serving as a central point for data storage and retrieval.

### 101. What is MongoDB ?

- MongoDB is an open-source NoSQL database written in C++ language. It uses JSON-like documents with optional schemas.
- It provides easy scalability and is a cross-platform, document-oriented database.
- MongoDB works on the concept of Collection and Document.
- It combines the ability to scale out with features such as secondary indexes, range queries, sorting, aggregations, and geospatial indexes.
- MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

### 102. What are some of the advantages of MongoDB?

- MongoDB supports field, range-based, string pattern matching type queries. for searching the data in the database
- MongoDB support primary and secondary index on any fields
- MongoDB basically uses JavaScript objects in place of procedures
- MongoDB uses a dynamic database schema

### 103. What is a Document in MongoDB?

A Document in MongoDB is an ordered set of keys with associated values. It is represented by a map, hash, or dictionary. In JavaScript, documents are represented as objects:

```
{"greeting" : "Hello world!"}
```

Complex documents will contain multiple key/value pairs:

```
{"greeting" : "Hello world!", "views" : 3}
```

### 104. What is a Collection in MongoDB?

A collection in MongoDB is a group of documents. If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

Documents within a single collection can have any number of different

“shapes.”, i.e. collections have dynamic schemas.

For example, both of the following documents could be stored in a single collection:

```
{"greeting" : "Hello world!", "views": 3}  
{"signoff": "Good bye"}
```

## 105.What are Databases in MongoDB?

MongoDB groups collections into databases. MongoDB can host several databases, each grouping together collections.

Some reserved database names are as follows:

*admin*

*local*

*config*

## 106. What is the Mongo Shell?

It is a JavaScript shell that allows interaction with a MongoDB instance from the command line. With that one can perform administrative functions, inspecting an instance, or exploring MongoDB.

To start the shell, run the mongo executable:

```
$ mongod  
$ mongo  
MongoDB shell version: 4.2.0  
connecting to: test  
>
```

The shell is a full-featured JavaScript interpreter, capable of running arbitrary JavaScript programs. Let's see how basic math works on this:

```
> x = 100;  
200  
> x / 5;  
20
```

### 107. How to add data in MongoDB?

The basic method for adding data to MongoDB is “inserts”. To insert a single document, use the collection’s **insertOne** method:

```
> db.books.insertOne({"title" : "Start With Why"})
```

For inserting multiple documents into a collection, we use **insertMany**. This method enables passing an array of documents to the database.

### 108. How do you Delete a Document?

The CRUD API in MongoDB provides **deleteOne** and **deleteMany** for this purpose. Both of these methods take a filter document as their first parameter. The filter specifies a set of criteria to match against in removing documents.

For example:

```
> db.books.deleteOne({"_id" : 3})
```

### 109. How to perform queries in MongoDB?

The **find** method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to **find**, which is a document specifying the query criteria.

Example:

```
> db.users.find({"age" : 24})
```

### 110. How is Querying done in MongoDB?

The **find** method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to **find**, which is a document specifying the query criteria.



For example: If we have a string we want to match, such as a "username" key with the value "alice", we use that key/value pair instead:

```
> db.users.find({"username" : "alice"})
```

### 111. What type of NoSQL database MongoDB is?

MongoDB is a document-oriented database. It stores the data in the form of the BSON structure-oriented databases. We store these documents in a collection.

### 112. Explain Namespace?

A namespace is the series of the collection name and database name.

### 113. Differentiate MongoDB and MySQL?

Despite MySQL and MongoDB being freeware and open source databases, there are several differences between them in terms of a data relationship, transaction, performance speed, querying data, schema design, normalization, etc. The comparison between MongoDB and MySQL is similar to the comparison between Non-relational and Relational databases.

### 114. Why MongoDB is the best NoSQL database?

MongoDB is the best NoSQL database due to the following features:

- High Performance
- High Availability
- Easily Scalable
- Rich Query Language
- Document Oriented



### 115. Which languages can we use with MongoDB?

At Present, MongoDB offers driver support to C++, Java, PHP, Perl, Python, Go, Scala, and Ruby.

### 116. Define Collection?

A collection in MongoDB is a grouping of documents. Collections are schema-less, meaning documents in the same collection can have different structures. Collections are similar to tables in traditional relational databases.

### 117. What is the purpose of the DB command?

We use the “DB” command to get the name of the presently selected database.

### 118. What is a primary key in MongoDB?

In MongoDB, the `\_id` field serves as the primary key for a document. It must be unique within a collection and is automatically generated if not provided during document insertion.

### 119. How do you delete data from a MongoDB collection?

You can delete data from a MongoDB collection using methods like ``deleteOne()``, ``deleteMany()``, or ``findOneAndDelete()``. You specify a query to select the documents to delete.

## INTERMEDIATE

### 120. Difference between var and let keyword in javascript.

Some differences are

1. From the very beginning, the 'var' keyword was used in JavaScript programming whereas the keyword 'let' was just added in 2015.
2. The keyword 'Var' has a function scope. Anywhere in the function, the variable specified using var is accessible but in 'let' the scope of a variable declared with the 'let' keyword is limited to the block in which it is declared. Let's start with a Block Scope.
3. In ECMAScript 2015, let and const are hoisted but not initialized. Referencing the variable in the block before the variable declaration results in a ReferenceError because the variable is in a "temporal dead zone" from the start of the block until the declaration is processed.

#### 121. What do you mean by Self Invoking Functions?

Without being requested, a self-invoking expression is automatically invoked (initiated). If a function expression is followed by (), it will execute automatically. A function declaration cannot be invoked by itself.

Normally, we declare a function and call it, however, anonymous functions may be used to run a function automatically when it is described and will not be called again. And there is no name for these kinds of functions.

#### 122. What is the difference between exec () and test () methods in javascript?

- test () and exec () are RegExp expression methods used in javascript.
- We'll use exec () to search a string for a specific pattern, and if it finds it, it'll return the pattern directly; else, it'll return an 'empty' result.
- We will use a test () to find a string for a specific pattern. It will return the Boolean value 'true' on finding the given text otherwise, it will return 'false'.

### 123.What are the types of errors in javascript?

There are two types of errors in javascript.

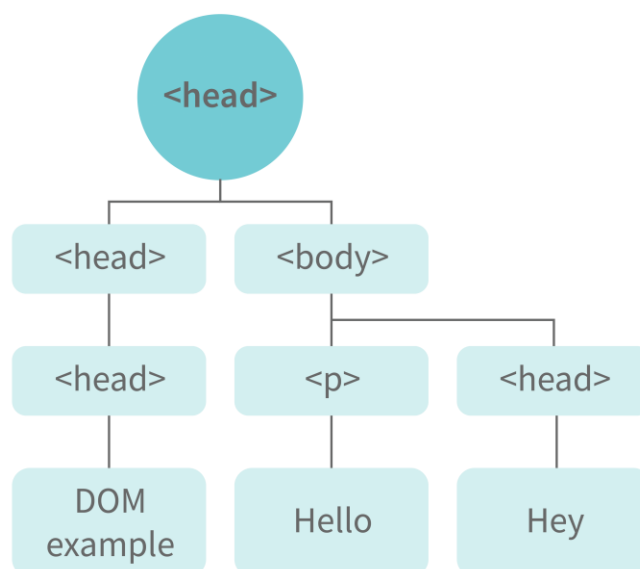
1. **Syntax error:** Syntax errors are mistakes or spelling problems in the code that cause the program to not execute at all or to stop running halfway through. Error messages are usually supplied as well.
2. **Logical error:** Reasoning mistakes occur when the syntax is proper but the logic or program is incorrect. The application executes without problems in this case. However, the output findings are inaccurate. These are sometimes more difficult to correct than syntax issues since these applications do not display error signals for logic faults.

### 124. DOM stands for Document Object Model.

DOM is a programming interface for HTML and XML documents.

- When the browser tries to render an HTML document, it creates an object based on the HTML document called DOM. Using this DOM, we can manipulate or change various elements inside the HTML document.
- Example of how HTML code gets converted to DOM:

```
<html>
<head>
<title>
DOM example
</title>
</head>
<body>
<p id = "para1">Hello</p>
<p id = "para2">Hey</p>
</body>
</html>
```



### 125. What do you mean by BOM?

Browser Object Model is known as BOM. It allows users to interact with the browser. A browser's initial object is a window. As a result, you may call all of the window's functions directly or by referencing the window. The document, history, screen, navigator, location, and other attributes are available in the window object.

### 126. Why do we use callbacks?

A callback function is a method that is sent as an input to another function (now let us name this other function "thisFunction"), and it is performed inside the thisFunction after the function has completed execution.

JavaScript is a scripting language that is based on events. Instead of waiting for a reply before continuing, JavaScript will continue to run while monitoring for additional events. Callbacks are a technique of ensuring that a particular code does not run until another code has completed its execution.

### 127. Difference between Async/Await and Generators usage to achieve the same functionality.

- Generator functions are run by their generator yield by yield which means one output at a time, whereas Async-await functions are executed sequentially one after another.
- Async/await provides a certain use case for Generators easier to execute.
- The output result of the Generator function is always value: X, done: Boolean, but the return value of the Async function is always an assurance or throws an error.

## 128. What are JavaScript Data Types?

There are three major Data types in JavaScript.

- **Primitive**
  - **Numbers**
  - **Strings**
  - **Boolean**
  - **Symbol**
- **Trivial**
  - **Undefined**
  - **Null**
- **Composite**
  - **Objects**
  - **Functions**
  - **Arrays**

## 129. What is the use of the isNaN function?

The number isNaN function determines whether the passed value is NaN (Not a number) and is of the type "Number". In JavaScript, the value NaN is considered a type of number. It returns true if the argument is not a number, else it returns false.

## 130. What do you mean by NULL in JavaScript?

The NULL value represents that no value or no object. It is known as empty value/object.

### 131.What are the benefits of JavaScript over other web-based technologies?

These are the benefits of JavaScript:

- **Interactive Enhancement**

JavaScript interacts with static web pages and makes them respond to users' inputs.

- **Quick Feedback**

There is no reason for a page on the internet to load again when using JavaScript. For example, form input validation.

- **Rich User Interface**

JavaScript assists in making the user interface of web-based applications look and feel better.

- **Frameworks**

JavaScript has vast libraries and frameworks that can be widely used to develop games and web-based applications.

### 132.Is JavaScript a dynamically typed or statically typed language?

JavaScript is a dynamically typed language.

### 133. Explain global variables in JavaScript.

**A variable declared outside of a function definition is referred to as a global variable, and its scope is across your entire program. This implies that its value is accessible and adjustable throughout your program.**

**134. What is the "this" keyword in JavaScript?**

"this" refers to an object running the current line of code. It is a reference to the object which executes the current function. If the function that is being referenced is a regular one, "this" references the global object. If the function is a method of the object, "this" refers to the object itself.

**135. Is JavaScript a case-sensitive language?**

Yes, JavaScript is a **case sensitive** language. The language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

**136. Name media types allowed by CSS.**

The different media types allowed by CSS are:

- speech
- audio
- visual
- tactile media
- continuous or paged media
- grip media or bitmap
- interactive media

**137. Explain responsive web design.**

Responsive Design is a web page creation approach that uses flexible images, flexible layouts, and CSS media queries. This design approach aims to build web pages that detect the orientation and screen size of the visitors so that the layout can be changed accordingly.

### 138. What are the different types of Selectors in CSS?

Universal Selector, Element type Selector, ID selector, Class selector, Descendant combinatory, Child Combinator, General Sibling Combinator, Adjacent sibling combinator, and Attribute selector.

### 139. How is the border-box different from the content box?

Border-box consists of properties such as content, padding, and the border with respect to height and width. However, Content-box is a default value property used for the box-sizing as well as it helps to add border and padding to give proper height and width to the box without having a border and padding properties.

### 140. How does Calc work?

Calc can be used to specify the result of the mathematical operation of two or more elements. For example to specify the width elements by the addition of two or more elements, we can write as

```
.foo {  
  
Width: calc(100px+50px)  
  
}
```

### 141. What does !important mean in CSS?

The style “!important” in the CSS has the highest precedence. Also, the cascaded property will be overridden with it.



#### 142. Does this property work overflow: hidden?

Overflow: the hidden property is used to specify the content's clipping. We need to add scrollbars to the content area at the time of specified container size exceeding the content limit where the content gets enclosed. This makes the content invisible via clipping; also the overflow value will be hidden.

#### 143. Differentiate between absolute and relative in CSS.

The main difference is that relative is used for the same tag in CSS. If we write `right:20 px`, then padding shifts 20 px in the right. Whereas absolute is relative to the non-static parent, i.e., if we write `right:20 px`, the result will be 20 px far from the right edge of the parent element.

#### 144. What is the current version of CSS?

CSS3 is the latest version of CSS.

#### 145. What are CSS HSL Colors?

**HSL:** HSL stands for Hue, Saturation, and Lightness respectively. This format uses the cylindrical coordinate system.

- **Hue:** Hue is the degree of the color wheel. Its value lies between 0 to 360 where 0 represents red, 120 represents green and 240 represents a blue color.
- **Saturation:** It takes a percentage value, where 100% represents completely saturated, while 0% represents completely unsaturated (gray).
- **Lightness:** It takes a percentage value, where 100% represents white, while 0% represents black.

##### Syntax:

```
h1 {  
    color:hsl(H, S, L);  
}
```

**146. Can we add 2D transformations to our project using CSS?**

Yes, we can, a transformation modifies an element by its shape, size, and position. It transforms the elements along the X-axis and Y-axis.

There are six main types of 2D transformations which are listed below:

- `translate()`
- `rotate()`
- `scale()`
- `skewX()`
- `skewY()`
- `matrix()`

**147. Can we add 3D transformations to our project using CSS?**

Yes, it allows changing elements using 3D transformations. In 3D transformation, the elements are rotated along the X-axis, Y-axis, and Z-axis.

There are three main types of transformation which are listed below:

- `rotateX()`
- `rotateY()`
- `rotateZ()`

**148. What are contextual selectors?**

Contextual selectors in CSS allow you to specify different styles for different parts of your document. You can assign styles directly to specific HTML tags, or, you could create independent classes and assign them to tags in the HTML. Either approach lets you mix and match styles.

**149. How would you style an image or element to have rounded corners?**

Use the `border-radius` property to add rounded corners to an image. 50% will make the image circular.

```
border-radius: 50%;
```

### 150. What does CSS selector mean?

A string equivalent of HTML elements by which declarations or a set of it, is declared and is a link that can be referred for linking HTML and Style sheet is CSS selector.

### 151. Enlist the various fonts' attributes?

They are:

- Font-style
- Font-variant
- Font-weight
- Font-size/line-height
- Font-family
- Caption
- Icon

### 152. Why is it easy to insert a file by importing it?

Importing enables combining external sheets to be inserted in many sheets. Different files and sheets can be used to have different functions.

Syntax:

@import notation, used with <Style> tag.

### 153. How can the dimension be defined of an element ?

Dimension properties can be defined by:

- Height
- Max-height
- Max-width
- Min-height
- Min-width
- Width

### 154. Can web browsers read JSX directly?

- Web browsers cannot read JSX directly. This is because they are built to only read regular JS objects and JSX is not a regular JavaScript object
- For a web browser to read a JSX file, the file needs to be transformed into a regular JavaScript object. For this, we use Babel



### 155. Explain how lists work in React

- We create lists in React as we do in regular JavaScript. Lists display data in an ordered format
- The traversal of lists is done using the `map()` function

```
const names = ['Kohli', 'Saif', 'Arun', 'Aamir', 'Arif'];

const listOfNames = () => {
  const listItems = names.map((name) =>
    <li key={name}>
      {name}
    </li>
  );
  return (
    <ul>{listItems}</ul>
  );
}
```

## **156. What is React Router?**

React Router is a routing library built on top of React, which is used to create routes in a React application. This is one of the most frequently asked react interview questions.

## **157 .Why do we need to React Router?**

- It maintains consistent structure and behavior and is used to develop single-page web applications.
- Enables multiple views in a single application by defining multiple routes in the React application.

## **158. Is HTML used in React?**

No, it uses an HTML-in JavaScript syntax called JSX (JavaScript and XML) that converts HTML tags to React elements.

## **159. What is the release date of React?**

March 2013

## **160. What is Redux in React?**

An open-source JavaScript library for front-end development and acts as a container for JavaScript applications that is used for the applications state management.

## **161. In Redux, what is an action?**

In Redux, an action is an object that contains information about an event that has occurred in an application.

### **162. How are ReactJS and React Native different?**

Where ReactJS is a front end open-source JavaScript library for UIs, React Native is an open-source mobile framework for platforms such as Android and iOS.

### **163. Can the values of props be updated?**

No, as props are immutable and top-down. This signifies that parent can transmit prop values to a child but the latter can't modify them.

### **164. How do you implement routing in ReactJS?**

React Router is available as an npm package, and can be added to your application with a single line of code. Once installed, you can define your routes in your application and use the component to link between them.

### **165. What are React Dev Tools?**

It lets you analyze the component hierarchy and includes component props and state. You can find [React Dev Tools](#) both as a standalone app and as a browser extension.

### **166. What is Jest?**

Jest is a JavaScript unit testing framework created by Facebook based on Jasmine. It offers automated mock creation and a jsdom environment. It is also used as a testing component.

### **167. Name two types of React component**

Two types of react Components are:

- Function component
- Class component

### **168. What are the lifecycle steps of React?**

Important lifecycle steps of React js are:

- Initialization
- State/Property updates
- Destruction are the lifecycle of React

### **169. What is Babel in React js?**

Babel, is a JavaScript compiler that converts latest JavaScript like ES6, ES7 into plain old ES5 JavaScript that most browsers understand.

### **170. What are the major issues of using MVC architecture in React?**

Here are the major challenges you will face while handling MVC architecture:

- DOM handling is quite expensive
- Most of the time applications were slow and inefficient
- Because of circular functions, a complex model has been created around models and ideas

### **171. List down some of the methods in a react-dom package**

Important methods for react-dom packages are:

- render()
- hydrate()
- createPortal()
- unmountComponentAtNode()
- findDOMNode()

### **172. How do you manage packages in your node.js project?**

It can be managed by a number of package installers and their configuration file accordingly. Out of them mostly use npm or yarn. Both provide almost

all libraries of javascript with extended features of controlling environment-specific configurations. To maintain versions of libs being installed in a project we use package.json and package-lock.json so that there is no issue in porting that app to a different environment.

### **173. Is it possible to avoid callback hells and how?**

It is possible to avoid callback hells by using promises; they can help to make the code more readable and easier to debug. You can also avoid callback hells using async/wait, libraries, and modularization.

### **174. What is the meaning of HTTP status code 504?**

HTTP status code 504 indicates that the server is unable to process the request. This can be due to several reasons, such as an overloaded server or a network issue.

### **175. How to include HTTP server in the Node module?**

You can include an HTTP server in your Node module by using the http module. The http module provides an API for creating and interacting with HTTP servers and clients. It allows you to handle incoming HTTP requests and send HTTP responses, as well as control the server's behavior, such as listening on a specific port and setting various options.

### **176. What is the purpose of EventEmitter?**

The purpose of EventEmitter is to provide a mechanism for event-driven programming. The EventEmitter class provides a set of methods that allow you to emit events, listen for events, and remove event listeners. When an event is emitted, any listeners that are registered for that event will be invoked. This allows developers to write code that can respond to changes in state, user actions, or other events within the application.



### 177. What is an error-first callback?

*Error-first callbacks* are used to pass errors and data. The first argument is always an error object that the programmer has to check if something went wrong. Additional arguments are used to pass data.

```
fs.readFile(filePath, function(err, data) {  
  if (err) {  
    //handle the error  
  }  
  // use the data object  
});
```

### 178. What is Callback Hell?

The asynchronous function requires callbacks as a return parameter. When multiple asynchronous functions are chained together then callback hell situation comes up.

### 179. List out the differences between AngularJS and NodeJS?

AngularJS is a web application development framework. It's a JavaScript and it is different from other web app frameworks written in JavaScript like jQuery. NodeJS is a runtime environment used for building server-side applications while AngularJS is a JavaScript framework mainly useful in building/developing client-side part of applications which run inside a web browser.

### 180. What is V8?

The V8 library provides Node.js with a JavaScript engine (a program that converts Javascript code into lower level or machine code that microprocessors can understand), which Node.js controls via the V8 C++ API. V8 is maintained by Google, for use in Chrome.

The Chrome V8 engine :

- The V8 engine is written in C++ and used in Chrome and Nodejs.
- It implements ECMAScript as specified in ECMA-262.

- The V8 engine can run standalone we can embed it with our own C++ program.

### 181. What is Event Emmitter?

All objects that emit events are members of EventEmitter class. These objects expose an `eventEmitter.on()` function that allows one or more functions to be attached to named events emitted by the object.

When the EventEmitter object emits an event, all of the functions attached to that specific event are called synchronously.

```
const EventEmitter = require('events');
```

```
class MyEmitter extends EventEmitter {}
```

```
const myEmitter = new MyEmitter();  
myEmitter.on('event', () => {  
  console.log('an event occurred!');  
});  
myEmitter.emit('event');
```

### 182.Explain the event loop in Node.js. How does it work?

The event loop is a crucial part of Node.js, responsible for handling I/O operations asynchronously. It allows Node.js to perform non-blocking operations, making it highly efficient for handling concurrent requests. The event loop continuously checks the event queue for any pending events. When an event is detected, it's processed, and its associated callback function is executed. This cycle repeats, allowing Node.js to handle multiple operations concurrently without blocking the execution thread.

### 183. Explain the purpose of package.json in a Node.js project.

- `package.json` is a manifest file used to define the metadata of a Node.js project, including its name, version, dependencies, scripts, and other configuration settings. It serves several purposes:

- **Dependency management:** It lists all the dependencies required for the project, allowing developers to easily install and manage dependencies using npm or Yarn.
- **Configuration:** It can contain various project configuration settings, such as scripts to run, project metadata, author information, etc.
- **Project metadata:** It provides information about the project, such as its name, version, description, license, etc.
- **Script execution:** It defines custom scripts that can be executed using npm or Yarn, such as start, test, build, etc.

**184. Explain the difference between `require()` and `import` in Node.js.**

- **`require()`** is a commonjs module system function used to import modules in Node.js, whereas **`import`** is an ECMAScript module feature introduced in ES6 and supported in newer versions of Node.js and browsers.

**185. What is the purpose of the `http` module in Node.js? How can you create an HTTP server using it?**

- The **`http`** module is used to create HTTP servers and make HTTP requests in Node.js. An HTTP server can be created by calling **`http.createServer()`** and providing a request listener function.

**186. Explain the difference between npm and yarn. When would you choose one over the other?**

- **npm** and **yarn** are package managers for Node.js. **npm** is the default package manager for Node.js, while **yarn** is a newer alternative. The choice between them often depends on factors like speed, dependency resolution, and features.

**187. Explain the purpose of the `mongodump` and `mongoexport` commands in MongoDB.**

- **`mongodump`** is a command-line utility used to create a binary export of data stored in MongoDB databases. It creates a backup of the data in a

BSON format. **mongorestore** is used to restore data from a **mongodump** backup to a MongoDB instance.

**189. What are TTL indexes in MongoDB? How can you use them to expire documents?**

TTL (Time-To-Live) indexes in MongoDB are special indexes that automatically expire documents after a specified period. They are useful for managing data retention policies or implementing cache invalidation strategies. TTL indexes are created on fields containing date values, and MongoDB automatically removes documents whose indexed field value exceeds the TTL threshold.

**190. Explain the difference between a primary key and a shard key in MongoDB.**

A primary key is a unique identifier for each document within a collection. It is automatically created and indexed by MongoDB. A shard key, on the other hand, is used for sharding purposes to distribute data across shards in a sharded cluster. While a primary key ensures document uniqueness within a collection, a shard key helps distribute data across multiple shards for scalability.

**191. What is the purpose of the `db.collection.createIndex()` method in MongoDB?**

- The **`db.collection.createIndex()`** method is used to create indexes on fields within a collection. Indexes improve query performance by allowing MongoDB to quickly locate documents based on the indexed fields.

**192. What is normalization, and why is it important in database design?**

- Normalization is the process of organizing data in a database to minimize redundancy and dependency. It involves dividing large tables into smaller tables and defining relationships between them. Normalization helps improve data integrity, reduce data duplication, and simplify database maintenance.

**193.Explain the difference between INNER JOIN, LEFT JOIN, and RIGHT JOIN in MySQL.**

**INNER JOIN:** Returns rows when there is a match in both tables.

**LEFT JOIN:** Returns all rows from the left table and the matched rows from the right table. If there is no match, NULL values are returned for the columns from the right table.

**RIGHT JOIN:** Returns all rows from the right table and the matched rows from the left table. If there is no match, NULL values are returned for the columns from the left table.

**194.What are indexes in MySQL, and how do they improve query performance?**

Indexes are data structures that improve the speed of data retrieval operations on database tables. They provide a quick way to look up data based on the values of specific columns. Indexes help reduce the number of rows that need to be scanned during query execution, resulting in faster query performance.

**195.Explain the purpose of the GROUP BY and HAVING clauses in MySQL.**

**GROUP BY:** Groups rows that have the same values into summary rows, typically for use with aggregate functions like COUNT, SUM, AVG, etc.

**HAVING:** Filters groups based on specified conditions after the GROUP BY clause has been applied. It is similar to the WHERE clause but is used with aggregate functions.

**196.Explain the difference between CHAR and VARCHAR data types in MySQL.**

- **CHAR:** Fixed-length character data type that stores strings with a specified length. Trailing spaces are padded to fill the remaining space.
- **VARCHAR:** Variable-length character data type that stores strings with a maximum length specified during column definition. It only uses as much storage space as needed for the actual data, without padding with spaces.

**197. What is the purpose of the UNION operator in MySQL?**

- The **UNION** operator is used to combine the result sets of two or more **SELECT** statements into a single result set. It removes duplicate rows by default. If you want to include duplicates, you can use **UNION ALL**.

**198. What are triggers in MySQL, and how are they used?**

- Triggers in MySQL are special stored procedures that are automatically invoked in response to specific events, such as INSERT, UPDATE, or DELETE operations on a table. They are used to enforce business rules, maintain data integrity, or perform auditing tasks.

**199. Explain the purpose of the ENUM data type in MySQL.**

- The **ENUM** data type in MySQL allows you to define a list of permissible values for a column. It restricts the values that can be inserted into the column to only those specified in the ENUM list.

**200. What is the difference between a subquery and a join in MySQL? When would you use each?**

- A subquery is a query nested within another query and is executed independently. It can be used to retrieve values for comparison or filtering within the main query.
- A join is used to combine rows from two or more tables based on a related column between them. Joins are used to retrieve related data from multiple tables simultaneously.