JavaScript & Web Development Concepts

Inline Element and Block Element

Inline elements do not start on a new line and take up only as much width as necessary (e.g., <span>, <a>). Block elements start on a new line and take up the full width of their container (e.g., <div>, <p>).

Why Need JavaScript

JavaScript enables interactivity, dynamic content, form validation, AJAX requests, and much more, making websites interactive and user-friendly.

Non-Replaced Element

Non-replaced elements are elements that are rendered without a fixed size or image, such as <div>, <p>, and <span>. These elements don't have an external representation like an image or input element.

Canvas and SVG

Canvas is an HTML element for rendering graphics dynamically with JavaScript. SVG (Scalable Vector Graphics) is a markup language for vector graphics that scales without losing quality.

Cascading Meaning

Cascading refers to the order of priority in which CSS rules are applied to elements. If multiple rules target the same element, the one with higher specificity or later in the code wins.

Specificity

Specificity determines which CSS rule applies when multiple rules target the same element. It's based on the types of selectors used (ID > Class > Element).

Flexbox

Flexbox is a layout model that allows easy arrangement of items in rows or columns. It provides flexibility to align, justify, and distribute space between items within a container.

Box Model in CSS

The CSS box model describes how elements are structured: content, padding, border, and margin. It determines how much space an element occupies on the page.

Box-Sizing (Border-box, Content-box)

In the box-sizing model, content-box means the padding and border are outside the element's specified width/height, while border-box includes padding and border inside the element's width/height.

Data Types (Primitive and Non-Primitive)

Primitive data types include strings, numbers, booleans, null, undefined, and symbols. Non-primitive data types are objects, arrays, and functions.

Var and Let Difference

Var is function-scoped and can be redeclared. Let is block-scoped and cannot be redeclared in the same block.

Slice and Splice

slice() creates a shallow copy of a portion of an array. splice() changes the content of an array by removing, replacing, or adding elements.

Map and Promises

Map is a method that allows mapping over arrays. Promises represent the eventual completion (or failure) of an asynchronous operation.

DOM

The DOM (Document Object Model) is a representation of the webpage as a tree of objects, allowing JavaScript to manipulate HTML and CSS dynamically.

Event Bubbling and Capturing

Event bubbling propagates events from the target element upwards to the root, while event capturing propagates events from the root down to the target.

Advanced JavaScript & React Concepts

Hooks Rest vs Spread

Hooks are used for managing state and side effects in React. Spread is a shorthand for copying and merging objects or arrays.

Local Storage vs Session Storage

Local Storage persists data until it is explicitly deleted, while Session Storage is cleared when the browser session ends.

Module CommonJS vs ES Modules

CommonJS uses 'require' and 'module.exports' for module loading, while ES Modules use 'import' and 'export'.

Virtual DOM vs Shadow DOM

Virtual DOM is a lightweight representation of the real DOM used by React to optimize rendering. Shadow DOM provides encapsulation for style and structure in web components.

Reconciliation

Reconciliation is the process by which React updates the DOM by comparing the new and old virtual DOM to apply only necessary changes.

Props vs States

Props are read-only inputs passed to components, while state represents mutable data that can change within a component.

Refs in React

Refs are used to access and modify DOM elements or React components directly, bypassing the usual data flow.

Query Optimization in MongoDB (Find vs Aggregate)

find() is a simpler query method for retrieving documents, while aggregate() is used for more complex queries and transformations.

Sharding in Indexing

Sharding is the process of distributing data across multiple servers, allowing for more efficient querying and storage.

Migrate Data from Relational DB to NoSQL

This involves transforming data from a relational schema to a document-based format suitable for NoSQL databases like MongoDB.

MapReduce in MongoDB

MapReduce is a method for processing large data sets by splitting tasks into smaller sub-tasks, allowing parallel processing.

Managed vs Unmanaged Cursor in MongoDB

A managed cursor is automatically handled by MongoDB, while an unmanaged cursor requires manual management of resources.

Object ID Role in MongoDB

ObjectID is a unique identifier used for documents in MongoDB collections to ensure data integrity.

State in React

State represents data that can change over time within a component, and it triggers re-rendering when updated.

Command Pattern in Software Design

The Command Pattern is a behavioral design pattern where an object encapsulates a request as a command.

Pre-Linked Set Clusters in MongoDB

Pre-linked sets in MongoDB represent collections that are linked or joined using specific relationships.

JS Data Structures (Map, Set, Template Literals)

Map and Set are data structures that allow storing unique key-value pairs and unique values, respectively. Template literals provide an easy way to work with strings in JavaScript.

Promises in JavaScript

Promises represent the completion or failure of an asynchronous operation and its resulting value.

Event Loop in JavaScript

The event loop is a mechanism that handles asynchronous events, allowing non-blocking execution of code in JavaScript.

Call, Apply, Bind in JavaScript

Call and Apply are methods for invoking functions with a specified 'this' value. Bind creates a new function with a specified 'this' value.

Object Types and Hashing

Objects in JavaScript can be of various types, and hashing is used for efficient storage and retrieval of objects in structures like maps.

Creating Objects in JavaScript

Objects can be created using object literals, constructor functions, or the 'new' keyword in JavaScript.

React's Declarative Approach

React is declarative because you describe how your UI should look based on state and props, and React takes care of rendering the DOM.

JS's Imperative Approach

JavaScript is imperative in nature, meaning you specify the steps to achieve a desired result.

Filter Method in JavaScript

The filter() method creates a new array with elements that pass the test implemented by the provided function.