**WEB DEV QUESTIONS :**

**HTML Questions**

1. **What is the purpose of the <!DOCTYPE> declaration in HTML?**
   * It defines the document type and version of HTML being used, helping browsers render the page correctly.
2. **What are semantic HTML elements? Give examples.**
   * Semantic HTML elements clearly describe their meaning in a human- and machine-readable way. Examples include <header>, <footer>, <article>, and <section>.
3. **Explain the difference between <div> and <span>.**
   * <div> is a block-level element used to group other block-level elements and create structure. <span> is an inline element used to style a portion of text or other inline elements.
4. **How do you include external CSS and JavaScript files in an HTML document?**
   * CSS: <link rel="stylesheet" href="styles.css"> JavaScript: <script src="script.js"></script>
5. **What are data attributes in HTML? How are they used?**
   * Data attributes store custom data in HTML elements and can be accessed via JavaScript using dataset. Example: <div data-info="value"></div>
6. **What is the purpose of the <meta> tag?**
   * The <meta> tag provides metadata about the HTML document, such as character encoding, author, and viewport settings.
7. **How do you create a form in HTML? What are some common input types?**
   * Use the <form> tag. Common input types include <input type="text">, <input type="password">, <input type="checkbox">, and <input type="radio">.
8. **What are HTML5 APIs and how are they used?**
   * HTML5 APIs provide additional functionalities such as geolocation, local storage, and offline capabilities. They are used via JavaScript to enhance web applications.
9. **Explain the purpose of the <head> section in an HTML document.**
   * The <head> section contains metadata, links to stylesheets, and scripts that define the document's settings and resources.
10. **How do you make an HTML page accessible?**
    * Use semantic elements, provide alt text for images, ensure keyboard navigation, and use ARIA roles for assistive technologies.
11. **What is the difference between <ol> and <ul>?**
    * <ol> creates an ordered list with numbered items, while <ul> creates an unordered list with bullet points.
12. **How do you add multimedia elements such as images, audio, and video in HTML?**
    * Images: <img src="image.jpg" alt="description"> Audio: <audio src="audio.mp3" controls></audio> Video: <video src="video.mp4" controls></video>
13. **What is the purpose of the <title> tag?**
    * The <title> tag specifies the title of the HTML document, which is displayed on the browser's title bar or tab.
14. **How do you create hyperlinks in HTML?**
    * Use the <a> tag. Example: <a href="https://www.example.com">Link</a>
15. **What are the differences between id and class attributes in HTML?**
    * id is unique to a single element and used for specific styling or JavaScript access. class can be applied to multiple elements for shared styling.
16. **How do you use the alt attribute for images?**
    * The alt attribute provides alternative text for an image if it cannot be displayed. Example: <img src="image.jpg" alt="Description of image">
17. **What is the purpose of the <form> element and its attributes?**
    * The <form> element collects user input. Attributes include action (URL where data is sent) and method (HTTP method used).
18. **How do you create tables in HTML?**
    * Use <table>, <tr> for rows, <td> for cells, and <th> for headers. Example: <table><tr><th>Header</th></tr><tr><td>Data</td></tr></table>
19. **What are <article> and <section> tags used for?**
    * <article> represents a self-contained piece of content, while <section> groups related content within a page.
20. **How do you create a dropdown menu in HTML?**
    * Use the <select> element with <option> elements for each dropdown item. Example: <select><option>Item 1</option></select>
21. **What are fieldset and legend used for in forms?**
    * <fieldset> groups related form elements, and <legend> provides a caption for the group.
22. **How can you make your HTML content responsive?**
    * Use responsive design techniques such as media queries, flexible grid layouts, and responsive images.
23. **What is the meta viewport tag used for?**
    * The meta viewport tag controls the layout on mobile browsers, specifying how to scale and size the content.
24. **How do you handle character encoding in HTML?**
    * Use the <meta charset="UTF-8"> tag in the <head> to specify the character encoding.
25. **Explain the role of the <base> tag in HTML.**
    * The <base> tag specifies a base URL for all relative URLs in the document.

**CSS Questions**

1. **What is the CSS box model?**
   * The box model describes the rectangular boxes generated for elements, including margins, borders, padding, and content.
2. **What are the different ways to apply CSS to a web page?**
   * Inline styles, internal style sheets using <style>, and external style sheets using <link>.
3. **How do CSS selectors work? Explain different types.**
   * CSS selectors target HTML elements to apply styles. Types include element selectors, class selectors, ID selectors, and attribute selectors.
4. **What is the difference between padding and margin?**
   * Padding is the space between the element's content and its border. Margin is the space outside the border, separating the element from other elements.
5. **Explain the use of z-index in CSS.**
   * The z-index property controls the stacking order of positioned elements. Higher values are positioned above lower values.
6. **What are pseudo-classes and pseudo-elements? Give examples.**
   * Pseudo-classes style elements based on their state (e.g., :hover, :focus). Pseudo-elements style specific parts of elements (e.g., ::before, ::after).
7. **How does Flexbox work and what are its key properties?**
   * Flexbox arranges items in a flexible container. Key properties include display: flex, justify-content, align-items, and flex-direction.
8. **What is the purpose of CSS Grid?**
   * CSS Grid provides a two-dimensional layout system, allowing for complex grid-based designs with rows and columns.
9. **How do media queries work in CSS?**
   * Media queries apply styles based on device characteristics such as screen size. Example: @media (max-width: 600px) { ... }
10. **Explain the difference between position: relative, absolute, fixed, and sticky.**
    * relative: Position relative to the element’s normal position. absolute: Positioned relative to the nearest positioned ancestor. fixed: Positioned relative to the viewport. sticky: Hybrid of relative and fixed positioning based on scroll position.
11. **What is the box-sizing property and how does it affect layout?**
    * The box-sizing property controls how the width and height of an element are calculated, including padding and borders (content-box vs. border-box).
12. **How can you create a responsive layout with CSS?**
    * Use flexible grids, media queries, and relative units like percentages and em for scalable layouts.
13. **What is the transform property in CSS?**
    * The transform property applies 2D or 3D transformations to elements, such as rotate, scale, or translate.
14. **What are CSS transitions and animations? How are they used?**
    * Transitions create smooth changes between property values, while animations define keyframes for more complex sequences.
15. **How do you center a block element in CSS?**
    * Use margin: auto with a specified width or use Flexbox or Grid for centering.
16. **What are custom properties (CSS variables) and how do you use them?**
    * Custom properties (variables) store values for reuse throughout a stylesheet. Example: --main-color: #333;
17. **How do you apply styles to a specific HTML element based on its state?**
    * Use pseudo-classes like :hover, :focus, or :active.
18. **What is the display property and what are its values?**
    * The display property defines how an element is displayed. Values include block, inline, inline-block, flex, and grid.
19. **How do you style form elements using CSS?**
    * Use CSS selectors specific to form elements (input, select, textarea) and apply styles accordingly.
20. **What are @import and @media rules in CSS?**
    * @import imports other stylesheets. @media applies styles based on media queries.
21. **How do you optimize CSS performance?**
    * Minify CSS files, use shorthand properties, and avoid excessive use of selectors.
22. **What is a CSS preprocessor and how does it differ from regular CSS?**
    * CSS preprocessors like Sass or Less extend CSS with variables, nesting, and functions. They compile into standard CSS.
23. **Explain the concept of specificity in CSS.**
    * Specificity determines which CSS rule is applied when multiple rules match the same element, based on selector types and counts.
24. **How do you handle browser compatibility issues with CSS?**
    * Use vendor prefixes, polyfills, and test across different browsers to ensure compatibility.
25. **What are the differences between inline, block, and inline-block display values?**
    * inline: Does not break the line, only takes up as much width as necessary. block: Takes up the full width of its container and starts on a new line. inline-block: Like inline, but allows width and height to be set.

**JavaScript Questions**

1. **What are closures and how do they work in JavaScript?**
   * Closures are functions that have access to variables from their outer scope even after the outer function has returned.
2. **What is the difference between var, let, and const?**
   * var: Function-scoped, can be redeclared and updated. let: Block-scoped, can be updated but not redeclared in the same scope. const: Block-scoped, cannot be updated or redeclared.
3. **Explain the event loop and asynchronous programming in JavaScript.**
   * The event loop handles asynchronous code execution, allowing JavaScript to perform non-blocking operations by placing callbacks in the event queue.
4. **What is the purpose of this keyword and how is it determined?**
   * this refers to the context in which a function is executed. It is determined by how the function is called (e.g., object method, constructor).
5. **What is the difference between == and === operators?**
   * == compares values with type conversion, while === compares values and types without type conversion.
6. **What are promises and how do they work?**
   * Promises represent the result of an asynchronous operation. They can be in a pending, fulfilled, or rejected state and are used with .then() and .catch() methods.
7. **Explain how the fetch API works for making HTTP requests.**
   * The fetch API provides a modern way to make network requests, returning a promise that resolves to the response. Example: fetch(url).then(response => response.json()).
8. **What are arrow functions and how do they differ from regular functions?**
   * Arrow functions provide a concise syntax and do not have their own this binding, inheriting this from the enclosing context.
9. **What is the difference between synchronous and asynchronous code?**
   * Synchronous code executes sequentially, blocking further execution until complete. Asynchronous code allows other operations to continue while waiting for a task to complete.
10. **How do you handle errors in JavaScript?**
    * Use try...catch blocks to handle exceptions and .catch() method for promises. Example: try { // code } catch (error) { // handle error }.
11. **What are JavaScript data types and how are they classified?**
    * JavaScript data types include primitive types (string, number, boolean, null, undefined, symbol) and non-primitive types (object, array, function).
12. **Explain how prototypal inheritance works in JavaScript.**
    * Objects inherit properties and methods from other objects via their prototype chain, allowing for shared behavior among instances.
13. **What is the purpose of bind, call, and apply methods?**
    * bind: Creates a new function with a specified this context. call: Invokes a function with a specified this context and arguments. apply: Similar to call, but arguments are passed as an array.
14. **What are JavaScript modules and how do you use them?**
    * JavaScript modules allow for code encapsulation and reuse by exporting and importing functionality between files using export and import statements.
15. **How does the Object.create() method work?**
    * Object.create() creates a new object with the specified prototype object and properties, allowing for object inheritance.
16. **What is event delegation and why is it useful?**
    * Event delegation involves attaching a single event listener to a parent element to manage events for multiple child elements, improving performance and reducing code.
17. **What are setTimeout and setInterval methods?**
    * setTimeout: Executes a function after a specified delay. setInterval: Repeatedly executes a function at specified intervals.
18. **How do you manipulate the DOM using JavaScript?**
    * Use methods like getElementById(), querySelector(), and addEventListener() to interact with and modify HTML elements and attributes.
19. **What are higher-order functions? Give examples.**
    * Higher-order functions take other functions as arguments or return functions. Examples include map(), filter(), and reduce().
20. **What is the role of the constructor function in JavaScript?**
    * The constructor function initializes new objects and sets up their initial properties and methods.
21. **How do you implement a debounce function?**
    * A debounce function limits how often a function can be called by delaying its execution until a certain time has passed. Example implementation:

javascript

Copy code

function debounce(func, wait) {

let timeout;

return function(...args) {

clearTimeout(timeout);

timeout = setTimeout(() => func.apply(this, args), wait);

};

}

1. **What is the purpose of the apply method in JavaScript?**
   * The apply method calls a function with a given this value and arguments provided as an array.
2. **What is the difference between null and undefined?**
   * null is an intentional absence of value, while undefined indicates a variable has been declared but not assigned a value.
3. **How do you perform deep cloning of objects in JavaScript?**
   * Use methods like JSON.parse(JSON.stringify(obj)) or libraries such as Lodash (\_.cloneDeep(obj)).
4. **What are IIFEs (Immediately Invoked Function Expressions)?**
   * IIFEs are functions that are defined and executed immediately. Example: (function() { // code })();

**React.js Questions**

1. **What are React hooks and how do they work?**
   * React hooks are functions that let you use state and other React features in functional components. Common hooks include useState, useEffect, and useContext.
2. **Explain the difference between class components and functional components.**
   * Class components use ES6 classes and have lifecycle methods. Functional components are simpler and use hooks for state and effects.
3. **What is the virtual DOM and how does it work in React?**
   * The virtual DOM is an in-memory representation of the real DOM. React uses it to optimize updates by only re-rendering changed components.
4. **How do you manage state in a React application?**
   * State is managed using useState hook in functional components or this.state in class components. Global state can be managed using context or libraries like Redux.
5. **What is JSX and how does it differ from HTML?**
   * JSX is a syntax extension for JavaScript that looks similar to HTML but allows embedding expressions and JavaScript code.
6. **Explain the purpose of useEffect hook.**
   * The useEffect hook performs side effects in functional components, such as data fetching, subscriptions, and manually changing the DOM.
7. **What are controlled and uncontrolled components in React?**
   * Controlled components have their form data controlled by React state. Uncontrolled components manage their own state internally.
8. **How do you handle forms and form validation in React?**
   * Handle forms with controlled components and validate using libraries like Formik or custom validation functions.
9. **What is context API and how is it used in React?**
   * The context API provides a way to pass data through the component tree without props drilling, using React.createContext() and Context.Provider.
10. **Explain the concept of props and how they are used in React.**
    * Props are read-only attributes passed to components to configure them and pass data from parent to child components.
11. **What are React lifecycle methods and how are they used?**
    * Lifecycle methods are hooks in class components that allow code to run at specific points during a component's lifecycle (e.g., componentDidMount, componentDidUpdate).
12. **How do you optimize performance in a React application?**
    * Use techniques like memoization with React.memo, lazy loading components, and optimizing rendering with shouldComponentUpdate or useMemo.
13. **What is Redux and how does it work with React?**
    * Redux is a state management library that stores application state in a central store and uses reducers to manage state updates.
14. **What are higher-order components (HOCs)?**
    * HOCs are functions that take a component and return a new component with additional props or functionality.
15. **How do you handle routing in a React application?**
    * Use react-router for routing, with components like BrowserRouter, Route, and Link to manage navigation.
16. **What is React Suspense and how is it used?**
    * React Suspense allows for lazy loading components and handling asynchronous operations, providing a fallback UI while components are loading.
17. **What is the useReducer hook and when would you use it?**
    * useReducer is a hook for managing more complex state logic, similar to useState but with a reducer function and dispatch actions.
18. **Explain the purpose of React.memo and how it works.**
    * React.memo is a higher-order component that memoizes a component, preventing re-renders if the props haven't changed.
19. **How do you perform error handling in React components?**
    * Use error boundaries by defining a class component with componentDidCatch and static getDerivedStateFromError methods.
20. **What are the differences between useCallback and useMemo hooks?**
    * useCallback returns a memoized version of a callback function, while useMemo returns a memoized value, optimizing performance.
21. **What is the role of React.StrictMode?**
    * React.StrictMode helps identify potential problems in an application by activating additional checks and warnings during development.
22. **How do you handle asynchronous operations in React?**
    * Use the useEffect hook for side effects and manage asynchronous operations with async/await or promise-based functions.
23. **What is the key prop and why is it important in lists?**
    * The key prop helps React identify which items have changed, are added, or are removed, improving performance in lists.
24. **How do you use context to manage global state in React?**
    * Create a context with React.createContext(), provide the context value using Context.Provider, and consume it with useContext or Context.Consumer.
25. **What are render props and how are they used in React?**
    * Render props are a pattern where a component uses a function as a prop to share code between components, allowing for flexible rendering.

**SQL Questions**

1. **What is the difference between INNER JOIN and LEFT JOIN?**
   * INNER JOIN returns rows that have matching values in both tables. LEFT JOIN returns all rows from the left table and matched rows from the right table, with NULL for unmatched rows.
2. **How do you create a new table in SQL?**
   * Use the CREATE TABLE statement. Example:

sql

Copy code

CREATE TABLE table\_name (

column1 datatype,

column2 datatype,

...

);

1. **What are primary keys and foreign keys?**
   * A primary key uniquely identifies each record in a table. A foreign key is a column that creates a relationship between two tables by referencing the primary key of another table.
2. **What is normalization in SQL and why is it important?**
   * Normalization is the process of organizing data to reduce redundancy and improve data integrity. It involves dividing tables into related tables and using foreign keys.
3. **Explain the GROUP BY clause and its use.**
   * The GROUP BY clause groups rows that have the same values into summary rows, often used with aggregate functions like COUNT, SUM, AVG.
4. **What is an aggregate function in SQL? Give examples.**
   * Aggregate functions perform calculations on a set of values and return a single value. Examples include COUNT(), SUM(), AVG(), MAX(), MIN().
5. **How do you use the HAVING clause?**
   * The HAVING clause filters groups created by the GROUP BY clause based on specified conditions.
6. **What is the WHERE clause used for in SQL?**
   * The WHERE clause filters records based on specified conditions, used in SELECT, UPDATE, DELETE statements.
7. **What are subqueries and how are they used?**
   * Subqueries are queries nested inside another query. They can be used in SELECT, INSERT, UPDATE, and DELETE statements to perform operations based on the results of the inner query.
8. **Explain the use of DISTINCT keyword.**
   * The DISTINCT keyword removes duplicate rows from the result set, returning only unique values.
9. **What is a view in SQL and how is it created?**
   * A view is a virtual table based on the result of a query. It is created with the CREATE VIEW statement. Example:

sql

Copy code

CREATE VIEW view\_name AS

SELECT column1, column2

FROM table\_name

WHERE condition;

1. **How do you perform a JOIN operation in SQL?**
   * Use JOIN clauses to combine rows from two or more tables based on a related column. Example:

sql

Copy code

SELECT columns

FROM table1

INNER JOIN table2 ON table1.id = table2.id;

1. **What is the ALTER TABLE statement used for?**
   * The ALTER TABLE statement modifies the structure of an existing table, such as adding, deleting, or modifying columns.
2. **How do you handle NULL values in SQL?**
   * Use IS NULL or IS NOT NULL to check for NULL values. Use functions like COALESCE to replace NULL with a specified value.
3. **Explain the INSERT INTO statement.**
   * The INSERT INTO statement adds new rows to a table. Example:

sql

Copy code

INSERT INTO table\_name (column1, column2)

VALUES (value1, value2);

1. **What is a transaction in SQL and how is it managed?**
   * A transaction is a sequence of SQL statements that are executed as a single unit. Transactions are managed with BEGIN TRANSACTION, COMMIT, and ROLLBACK.
2. **What are indexes and how do they improve query performance?**
   * Indexes are database objects that improve query performance by allowing faster retrieval of rows based on indexed columns.
3. **How do you perform an UPDATE operation in SQL?**
   * Use the UPDATE statement to modify existing records. Example:

sql

Copy code

UPDATE table\_name

SET column1 = value1

WHERE condition;

1. **What is the purpose of LIMIT and OFFSET clauses?**
   * LIMIT restricts the number of rows returned by a query. OFFSET specifies the number of rows to skip before starting to return rows.
2. **How do you delete records from a table in SQL?**
   * Use the DELETE statement to remove records. Example:

sql

Copy code

DELETE FROM table\_name

WHERE condition;

1. **What is the CASE statement and how is it used?**
   * The CASE statement provides conditional logic in SQL queries. It returns different values based on conditions.

sql

Copy code

SELECT column1,

CASE

WHEN condition1 THEN result1

WHEN condition2 THEN result2

ELSE result3

END

FROM table\_name;

1. **What are stored procedures and how are they used?**
   * Stored procedures are precompiled SQL statements stored in the database. They are executed by calling the procedure name.
2. **How do you use the JOIN clause to combine data from multiple tables?**
   * Use different types of JOIN (e.g., INNER JOIN, LEFT JOIN) to combine rows from multiple tables based on related columns.
3. **What is a UNION operation in SQL?**
   * The UNION operation combines the result sets of two or more SELECT queries into a single result set, removing duplicates.
4. **Explain the LIKE operator and its use.**
   * The LIKE operator searches for a specified pattern in a column. It uses wildcards such as % (zero or more characters) and \_ (single character).