Interview Questions

**HTML**

1. **What is Doctype, and what does it do?**

The <!DOCTYPE> declaration, or "doctype," is an instruction that tells the browser which version of HTML is being used in the document.

1. **What is HTML and what do you mean by hyper?**

In HTML, "Hyper" refers to "Hypertext," which is text containing links (or "hyperlinks") that allow users to jump from one page or resource to another.

1. **What are semantic HTML elements? Give examples and explain why they are important?**

Semantic HTML elements clearly describe their meaning in a human- and machine-readable way. Examples include <header>, <footer>, <article>, <section>, and <aside>. They improve accessibility, SEO, and readability by helping browsers and search engines understand the content structure.

1. **What is the purpose of the alt attribute in an <img> tag? Why is it important for accessibility and SEO?**

The alt attribute provides alternative text for an image if it fails to load or for screen readers used by visually impaired users. It also helps with SEO by providing context to search engines about the image content.

1. **How can you create a table with merged rows and columns?**

You can use the rowspan and colspan attributes within <td> or <th> elements to merge rows and columns. For example, <td colspan="2"> merges two columns, and <td rowspan="3"> merges three rows.

1. **What are defer and async attributes on a <script> tag, and how do they affect script loading?**

async loads the script asynchronously, allowing the HTML to continue parsing while the script is fetched and executed. defer loads the script after the HTML document has finished parsing, maintaining execution order. defer is ideal for scripts that rely on the DOM.

1. **Explain the difference between <input type="button"> and <button> elements?**

<input type="button"> creates a button without any nested HTML, while <button> can contain text, icons, or other HTML elements. <button> is more flexible, and using <button type="submit"> allows for form submissions.

1. **Give me 10 Html5 tage?**

Header,footer,section,article,aside,nav,main,audio,video,source,details,summary and progress.

1. **What do you mean by row spanning and column spanning?**

Row spanning is used to merge (combine) two or more rows.Column spanning is used to merge (combine) two or more columns.

**CSS**

**10.What is the differencs between grid and flex box?**

**Grid:**

* Two-Dimensional Layout – handles both rows and columns.
* Complex Layouts - to control both the horizontal and vertical positioning of elements.
* Explicit Positioning
* Container-based
* Responsive Design

**Flex box:**

* One-Dimensional Layout - it deals with either a row or a column at a time.
* Simpler Layouts - It is ideal for simpler layouts where you only need to control the alignment and spacing along a single axis.
* Content-Based.
* Alignments and Distribution.
* Alignment Properties.

**11. what is the difference between the mediaquerry and keyframes in css?**

**Media Queries:**

Media queries are used to apply CSS styles conditionally based on the characteristics of the user’s device or viewport, such as screen size, orientation, or resolution.

They help in creating responsive designs by allowing different layouts or styles on various devices (e.g., mobile, tablet, desktop).

**Keyframes:**

Keyframes define the stages of an animation by specifying the styles at certain points in the animation sequence.

They create CSS animations by describing changes in properties (e.g., position, opacity) over time, allowing for smooth transitions or effects without JavaScript.

**12.**  **What is the difference between session storage , local storage and cookies?**

* **Local Storage**: Data persists even after the browser is closed and reopened. No expiration date, remains until explicitly deleted.
* **Session Storage**: Data only persists within the same tab or window; it’s cleared once the tab/window is closed. Ends when the session ends
* **Cookies**: Can persist until the expiration date set; otherwise, expires when the session ends.

**13. What is the difference between HTML Css and JavaScript And how it will works?**

**HTML (HyperText Markup Language)**

* **Role**: Structure
* **Purpose**: HTML is the backbone of any web page. It provides the structure and content of the page by using a system of elements (tags).
* **How it works**: HTML elements are the building blocks of web pages. They define headers, paragraphs, links, images, and other types of content.

**CSS (Cascading Style Sheets)**

* **Role**: Presentation
* **Purpose**: CSS is used to style and layout web pages. It allows you to change colors, fonts, spacing, and positioning of HTML elements.
* **How it works**: CSS rules are applied to HTML elements to control their appearance. This is done either inline within HTML tags, within <style> tags in the HTML document, or through external CSS files.

**JavaScript**

* **Role**: Behavior
* **Purpose**: JavaScript adds interactivity and dynamic behavior to web pages. It allows you to manipulate HTML and CSS, handle events, validate forms, create animations, and more.
* **How it works**: JavaScript code is executed in the browser. It can be included directly in HTML documents using <script> tags or in external .js files.

**JAVASCRIPT**

**14.Why do we need JavaScript?**

1. **Client-Side Interactivity:** JavaScript allows developers to create interactive elements on web pages. This includes things like form validation, dynamic content updates without refreshing the page (AJAX), interactive maps, sliders, and much more.
2. **Enhanced User Experience:** By using JavaScript, developers can create smoother and more responsive user interfaces. Actions such as animations, transitions, and smooth scrolling can all be implemented to enhance the overall user experience.
3. **Cross-Browser Compatibility:** JavaScript helps in ensuring that web applications work across different browsers and devices. Modern JavaScript frameworks and libraries often provide solutions for handling browser inconsistencies and optimizing performance.
4. **Server-Side Communication:** Through AJAX (Asynchronous JavaScript and XML) and modern frameworks like React, Angular, or Vue.js, JavaScript enables efficient communication with the server, allowing for dynamic updates and real-time data retrieval.
5. **Rich Web Applications:** JavaScript is crucial for building Single Page Applications (SPAs) where the entire application runs in the browser, providing a more desktop-like experience. Frameworks like React, Angular, and Vue.js facilitate the development of complex, interactive web applications.
6. **Community and Ecosystem:** JavaScript has a vast ecosystem of libraries, frameworks, and tools that simplify and accelerate development. This ecosystem fosters innovation and allows developers to leverage existing solutions to solve common problems.

**15.What is prototype in javascript?**

The JavaScript prototype property allows you to add new properties to object constructors,prototype property also allows you to add new methods to objects constructors.

**16**. **What are the data types in JavaScript?**

| **Data Type** |
| --- |
| 1. String |
| 1. Number |
| 1. BigInt |
| 1. Boolean |
| 1. Undefined |
| 1. Null |
| 1. Symbol |
| 1. Object |
| 1. Array |
| 1. Function |
| 1. Date |  |
|  |  |

**17.What is the difference between primitive and non-primitive data types?**

**Primitive:**

**Stored by Value:** Primitive data types (like numbers, strings, booleans, etc.) are stored directly in memory as simple.

**Immutable:** Primitive values are immutable, meaning their values cannot be changed. If you modify a variable holding a primitive value, you're actually creating a new value in memory rather than modifying the original.

**Non-primitive:**

**Stored by Reference:** Objects in JavaScript (arrays, functions, and objects themselves) are stored by reference in memory. This means that when you assign an object to a variable, you're actually storing a reference (or a pointer) to the memory location where the object is stored.

**Mutable:** Objects are mutable, meaning you can change their properties and values directly. Modifying an object's property does not create a new object; it modifies the existing object in memory.

**18. What are the differences between null and undefined?**

null is an assignment value that represents the intentional absence of a value, while undefined means a variable has been declared but not yet assigned a value.

**19. what is difference between =,==,=== ?**

**= (Assignment Operator)**: This is used to assign a value to a variable.

**== (Equality Operator)**: This checks if two values are equal in value but **not necessarily in type**. JavaScript performs type coercion, converting the values to a common type before comparison.

**===(Strict Equality Operator)**: This checks if two values are equal in both **value and type**. No type conversion is performed.

**20. What is event delegation, and why would you use it in a JavaScript application?**

Event delegation allows you to add a single event listener to a parent element that handles events on its child elements. This is useful when dealing with dynamic elements and improves performance by reducing the number of event listeners.

**21. what is the difference between parameter and argument?**

* Parameters are used when defining a function.
* Parameters act as placeholders for the values that the function will receive.
* Arguments are used when calling (or invoking) a function.
* Arguments are the actual values supplied to the function.

**22**. **what is the difference between slice and splice?**

**Slice:** Creates a shallow copy of a portion of an array.

**Splice:** Changes the contents of an array by removing, replacing, or adding elements.

**23. What is the difference between the spread operator and rest operator?**

**Spread operator:** Expands or "spreads" elements of an array or object into individual elements.

**Rest operator:** Collects multiple elements into a single array or collects remaining properties in an object.

**24.How does javascript works in sync or async?**

**25.How does promises works and what are stages?**

Promises in JavaScript are used to handle asynchronous operations. They represent a value that may be available now, in the future, or never, depending on the success or failure of an async operation.

1. Pending
2. Fulfilled
3. Rejected

**26.What is callback function?**

A **callback function** is a function passed into another function as an argument to be executed later, often after some asynchronous operation or specific event occurs.

**27. What is the difference between the foreach and map?**

**forEach**

* Purpose: Executes a provided function once for each array element.
* does not return anything; it simply performs an operation on each array element.
* **Mutability**: Typically used when you want to perform side effects, like modifying an array in place or logging each element.

**map**

* **Purpose**: Creates a new array by applying a function to each array element
* **Return Value**: map returns a new array containing the results of calling the provided function on each element.
* **Immutability**: map does not modify the original array; it generates a transformed array instead.

**28.What is async and await and how does it works?**

An **async** function always returns a promise. If the function returns a value, the promise will be resolved with that value. If the function throws an exception, the promise will be rejected.

An async function always returns a promise. If the function returns a value, the promise will be resolved with that value. If the function throws an exception, the promise will be rejected.

**await** makes a function wait for a Promise. The await keyword in JavaScript is used within async functions to pause the execution of the function until a Promise is settled (either fulfilled or rejected).

**29. What are the es6 features?**

1. **Arrow Functions**
2. **Classes**
3. **Template Literals**
4. **Destructuring Assignment**
5. **Default Parameters**
6. **Rest and Spread Operators**
7. **Modules**
8. **Promises**
9. **Map and Set**
10. **Iterators and Generators**
11. **Symbol Type**

**30. What is API ?**

In JavaScript, the fetch function is a powerful and modern way to make asynchronous HTTP requests to retrieve data from a server or to send data to it. It is part of the **Fetch API**, which is a replacement for the older XMLHttpRequest.

**31. What is Hoisting?**

We can use variables and functions before they are declared in the code.

**32. What is Closure?**

When a function is defined inside another function, the inner function has access to the variables of the outer function due to **lexical scoping**.

**33. What is the difference between fetch Api and Rest Api ?**

**Fetch API**: A JavaScript method for making HTTP requests.

**REST API**: A set of rules and conventions for building web APIs that interact over HTTP.

**34.What is the usage of exception handling? In a try catch blog who much catch and finally we can use?**

**React**

**35. Why do we need react and why can’t we able to create application in JavaScript?**

React is a JavaScript library for building user interfaces, particularly single-page applications where you need a fast and interactive user experience. It's used for its component-based architecture, reusability, and efficiency.

**36.What are the key features of React?**

Key features include components, props, state, lifecycle methods, hooks, and virtual DOM.

**37.What is JSX and how does it work?**

JSX is a syntax extension for JavaScript that allows you to write HTML-like code in your JavaScript files. It's compiled to JavaScript at runtime.

**38.What are functional components and class components?**

Functional components are pure functions that take props as input and return JSX. Class components are classes that extend React.Component. and must contain a render() method, which returns JSX.

**39.How do you pass data from a parent component to a child component?**

You pass data using props.

**40.** **What is the difference between state and props?**

**State** is internal and mutable within a component, used for dynamic data that changes over time.

**Props** are external and immutable, used to pass data and event handlers from parent to child components.

State is local to a component and can change over time. Props are immutable and passed from a parent component.

**41. What is the difference between state and props?**

**State** is internal and mutable within a component, used for dynamic data that changes over time.

**Props** are external and immutable, used to pass data and event handlers from parent to child components.

State is local to a component and can change over time. Props are immutable and passed from a parent component.

**42. What are hooks in React and why are they used?**

**43.What is the difference between useState and useReducer?**