HTML/CSS

Frequently Asked CSS Interview Questions

1. Tell us something about CSS3.

CSS3 is divided into modules and is supported by almost every browser. Many graphics-related characteristics are introduced in CSS3 like box-shadow, Border-radius, and flexbox. A user can create precise multiple background images using properties like background-position, background-repeat, and background-image styles.

2. How is a CSS selector used?

With a CSS selector, we can choose the content we want to style to bridge between HTML files and style sheets. CSS selector syntax is "select" HTML elements created on their class, id, type, etc.

3. What are CSS image scripts?

A group of images placed into one image is a CSS image script. It can reduce load time and project multiple images into a single web page.

4. Explain CSS specificity.

CSS specificity is a rank or score that decides style declaration to be used to an element. ID selectors have high specificity, while universal selector * has low specificity. The four CSS categories that authorize the selector's specificity level are IDs, inline style, elements/pseudo-elements, and classes and attributes.

5. Define gradients in CSS.

A property of CSS that allows displaying smooth transformation between two or more specified colors. The types of gradients are linear and radial.

6. What are the properties of flexbox?

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

7. Tell us about the use of the CSS Box Model.

The CSS Box model is a box binding HTML element that includes padding, border, margin, and the actual content. With the box model, we get the authority to add a border all around elements and define space between elements.

8. What are the position states in CSS?

The four-position states in CSS are relative, static, absolute, and fixed. The default position state is static.

9. 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.

10. What is common between class and ID?

Both class and ID are used in HTML to assign a value from CSS. The ID is used as an element, whereas the class is used as a block.

Important HTML/CSS Interview Questions

Q.1 What are the new features introduced in HTML5 and CSS3?

Answer: New features in HTML5 include semantic elements like <header>, <footer>, and <section>, as well as new form input types and attributes. CSS3 introduced features like Flexbox, Grid Layout, transitions, and animations.

Q.2 Explain the box model in CSS.

Answer: The box model in CSS refers to how elements are structured with content, padding, border, and margin.

Q.3 How do you make a website responsive?

Answer: A responsive website adapts its layout and design to different screen sizes and devices using techniques like media queries and flexible grids.

Q.4 What is the difference between display: none; and visibility: hidden;?

Answer:

display: none; hides an element completely, including its space in the layout, while **visibility: hidden;** hides the element but still occupies its space.

Q.5 What are the key responsibilities of Web Developers?

Answer: It is generally expected that web developers will be able to perform the following tasks:

- Build products using HTML, CSS, JavaScript, PHP (Hypertext Preprocessor), and other relevant coding languages.
- Design, develop, test, debug, and deploy applications in a cross-platform, cross-browser environment.
- Coordination with designers and programmers for the development of projects.
- Develop design specifications/patterns for optimizing web programs.
- Identifying and fixing bugs, troubleshooting, and resolving website issues.
- Taking care of the technical aspects of the site, such as its cache and performance (which indicate how fast a site will run and how much traffic it can handle).
- Providing support and assistance with web management best practices.
- Keep up with the latest technology.
- Maintain and update websites to meet modern web standards.
- Monitor web traffic.

Q.6 Explain CORS (Cross-Origin Resource Sharing) and Write its Importance.

Answer: CORS stands for Cross-origin resource sharing. It is basically defined as a browser mechanism that enables web pages from one domain to have controlled access to resources that are located at different domains (cross-domain request). In other words, it allows scripts running on a browser client to interact with and access resources from other origins. It provides and extends flexibility to the SOP (Same-Origin Policy). A same-origin policy restricts a website's ability to access resources outside its source domain. For example, if a JavaScript app wanted to call an API (Application Programming Interface) running on another domain, it would be blocked and prevented from doing so because of the SOP. Due to restrictions caused by the same-origin policy, CORS was introduced. When a website's CORS policy is set up poorly, it also poses the risk of cross-domain attacks. As such, it cannot prevent cross-origin attacks such as CSRF (Cross-Site Request Forgery).

Q.6 What do you mean by ETag (Entity Tag) and how does it work?

Answer: The ETag (entity tag) is a part of the HTTP protocol. This is one of several mechanisms that HTTP provides to validate Web caches, which allows conditional requests to be made from a browser to resources. Moreover, Etags make sure that simultaneous updates of the same resource don't overwrite each other (mid-air collisions).

ETags are opaque identifiers assigned by a server to a specific version of a resource found at a specific URL. Every time the resource representation at that URL changes, an entirely new

ETag is assigned. As such, ETags can be compared in the same way as fingerprints and determine if two representations of a resource are identical.

Syntax:

ETag: W/"<etag_value>"

ETag: "<etag_value>"

Q.8 Explain Webpack.

Answer: Webpack is a tool that bundles JavaScript modules, also known as static module bundlers. Modules are reusable chunks of code that are built from the JavaScript, node_modules, images, and CSS styles of your application, and packaged so that they can be easily added to your website. If you have a large number of files, Webpack generates a single (or a few) file that runs your application. When Webpack processes your application or package, it generates a dependency graph, which consists of various modules that your webapp needs in order to function as expected. Based on this graph, it then creates a new package that contains only the bare minimum files required, often only one or a few bundle.js files which can be easily plugged into the HTML file and used in the application.

Q.9 What is the difference between inline and block-level elements in HTML?

Answer:

Inline elements:

- Inline elements do not start on a new line and only take up as much width as necessary.
- Examples of inline elements include , <a>, , , <img>, <input>, and
.
- Inline elements cannot have width, height, margin, or padding applied to them.

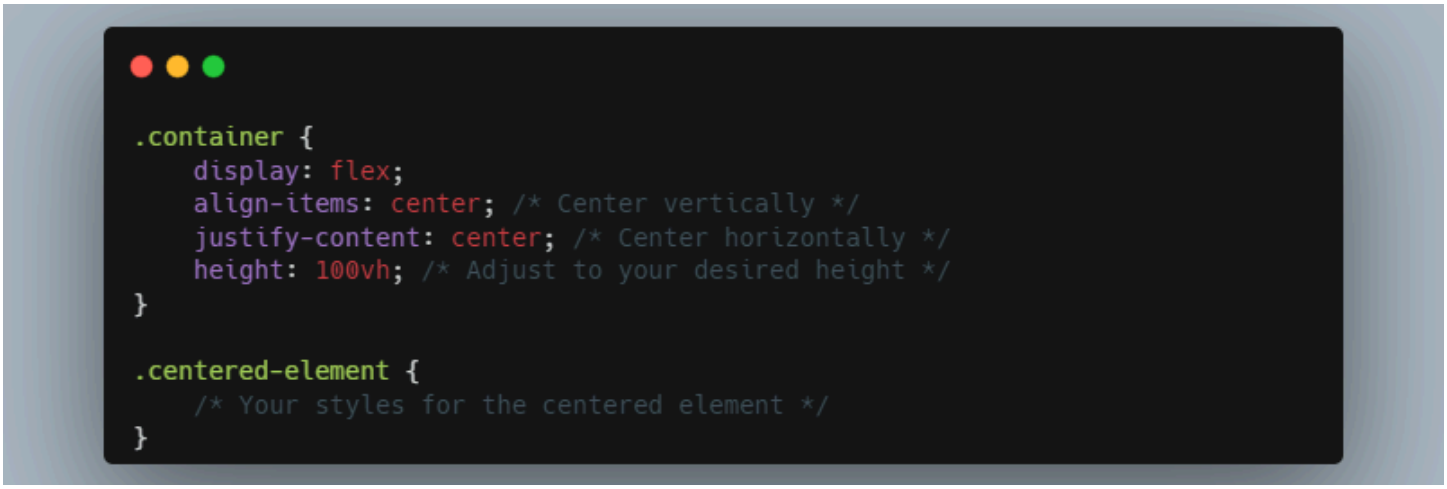
Block-level elements:

- Block-level elements start on a new line and take up the full width available.
- Examples of block-level elements include , <p>, <h1> to <h6>, <ul>, <ol>, <li>, <table>, and <form>.
- Block-level elements can have width, height, margin, and padding applied to them.

Q.10 How do you vertically center an element in CSS?

Answer:

Vertically centering an element in CSS can be achieved using various methods. One common method is using the CSS flexbox layout:



```css
.container {
    display: flex;
    align-items: center; /* Center vertically */
    justify-content: center; /* Center horizontally */
    height: 100vh; /* Adjust to your desired height */
}

.centered-element {
    /* Your styles for the centered element */
}
```

This CSS code centers the .centered-element vertically and horizontally within its parent container with the class .container. The align-items: center; property aligns the items along the cross-axis (vertical axis), and justify-content: center; aligns them along the main axis (horizontal axis) of the flex container. Adjust the height of the .container as needed.

Q.11 What are CSS preprocessors, and why are they used?

Answer:

CSS preprocessors are scripting languages that extend the functionality of CSS by introducing features like variables, nesting, mixins, inheritance, and mathematical operations. Examples of CSS preprocessors include Sass, Less, and Stylus.

CSS preprocessors are used for several reasons:

1. **Modularity:** Preprocessors allow you to organize CSS code into reusable modules using features like mixins and partials, making it easier to maintain and update stylesheets.
2. **Efficiency:** Features like variables and nesting help reduce repetition and make CSS code more concise and readable.
3. **Code organization:** Preprocessors provide tools for structuring and organizing CSS code, making it easier to manage large projects with complex styling requirements.
4. **Compatibility:** Preprocessors generate standard CSS code, ensuring compatibility with all modern web browsers.

Q.12 What is the purpose of CSS media queries?

Answer:

CSS media queries are used to apply different styles to a webpage based on various device characteristics, such as screen width, height, orientation, and resolution. Media queries allow you to create responsive designs that adapt to different devices and screen sizes, providing optimal viewing experiences across desktops, tablets, and smartphones.

Here's an example of a media query that applies styles only when the screen width is less than 768 pixels:

Q.13 What are some common performance optimization techniques for web applications?

Answer: Some common performance optimization techniques include:

- Reducing HTTP requests by combining files, using sprites, inlining CSS and JavaScript
- Minimizing file sizes through minification, gzip compression and image optimization
- Caching assets on the client-side and server-side
- Using a CDN to reduce latency
- Improving server response time

4. What is a content delivery network (CDN)?

A content delivery network (CDN) is a distributed system of servers strategically placed across the globe to deliver web content (e.g., HTML, images, videos) to users more quickly and efficiently. CDNs minimize delays in loading web page content by caching and delivering it from the server geographically closest to the user.

Q.15 Name some CSS frameworks.

CSS frameworks are libraries that make web page styling easier. Some of them are

Foundation, Bootstrap, Gumby, Ukit, Semantic UI, etc.

Intermediate CSS Interview Questions

Q.1 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.

Q.2 What are the benefits of CSS Sprites?

With CSS sprites, loading multiple images is not an issue.

- Blinking is not seen.
- Advanced downloading of assets does not take place until needed.

Q.3 How can you target h3 and h2 with the same styling?

Multiple elements can be targeted by separating with a comma:

h2, h3 {color: red;}

Q.4 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

Q.5 How can you use CSS to control image repetition?

The background-repeat property is used to control the image. Example:

h3 {

background-repeat: none;

}

Q.6Tell us about the property used for image scroll controlling?

The background-attachment property is used to set whether the background image is fixed or it scrolls with the rest of the page. Example for a fixed background-image:

body {

background-image: url(‘url_of_image’);

background-repeat: no-repeat;

background-attachment: fixed;

}

Q.7 Name some font-related CSS attributes.

The font-related attributes are Font- style, variant, weight, family, size, etc.

Q.8 Define contextual selectors.

In CSS, contextual selectors allow developers to specify styles of different parts of the document. Styles can be assigned directly to specific HTML tags or create independent classes and assign tags to them.

Q.9 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.

Q.10 Tell us about the general CSS nomenclature.

In CSS, the styling commands are written in value and property fashion. CSS includes a system terminator- semicolon. The entire style is wrapped in curly braces and attached to the selector. This creates a style sheet that can be applied to an HTML page.

Q.11 What are the limitations of CSS?

1. CSS cannot always assure compatibility with every browser; you need to be cautious while choosing the style selector.
2. The parent selector tag is not available, thus you can't select the parent selector tag.
3. Some selectors can lead to cross-browser issues due to their less browser-friendly behavior.
4. We cannot request a webpage through CSS.

Q.12 How to include CSS in the webpage?

1. With the help of a link tag, you can include an external style sheet file as a CSS file into your HTML file.
2. You can add CSS styles included within your HTML page and write it in the stand-alone stylesheet form of CSS.
3. CSS can be included directly in the HTML tag by adding an inline style to HTML elements.
4. One can import an external stylesheet file as a new CSS file by using the @import rule.

JavaScript

JAVASCRIPT FUNDAMENTALS

1. What is JavaScript?

A high-level, interpreted programming language called JavaScript makes it possible to create interactive web pages and online apps with dynamic functionality. Commonly referred to as

the universal language, Javascript is primarily used by developers for front-end and back-end work.

2. What are the different data types in JavaScript?

JavaScript has six primitive data types:

- Number
- String
- Boolean
- Null
- Undefined
- Symbol
- It also has two compound data types:
- Object
- Array

3. What is hoisting in JavaScript?

Hoisting is a JavaScript concept that refers to the process of moving declarations to the top of their scope. This means that variables and functions can be used before they are declared, as long as they are declared before they are used in a function.

For example, the following code will print "Hello, world!" even though the greeting variable is not declared until after the console.log() statement.

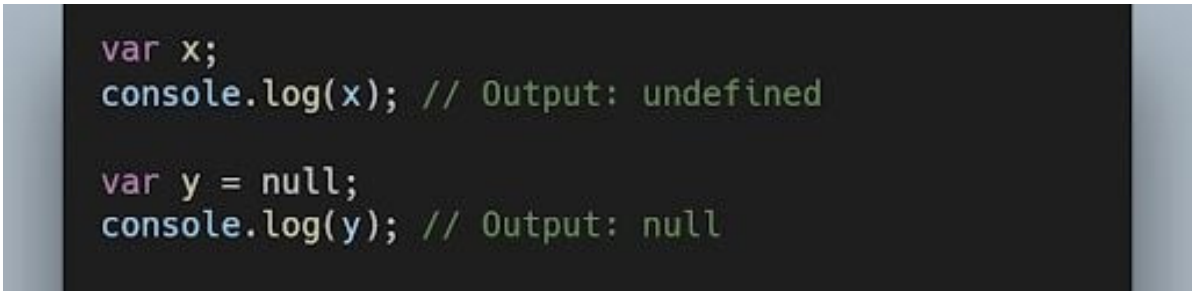
```
var greeting = "Hello, world!";

function sayHello() {
  console.log(greeting);
}

sayHello();
```

4. What is the difference between null and undefined?

null is an assignment value that represents no value or an empty value, while undefined is a variable that has been declared but not assigned a value.



```
var x;
console.log(x); // Output: undefined

var y = null;
console.log(y); // Output: null
```

5. Why do we use the word “debugger” in JavaScript?

The word “debugger” is used in JavaScript to refer to a tool that can be used to step through JavaScript code line by line. This can be helpful for debugging JavaScript code, which is the process of finding and fixing errors in JavaScript code. To use the debugger, you need to open the JavaScript console in your browser. Then, you can use debugger commands to comb through your code line by line.

It's essential to know debugging techniques as well as the more general ideas behind code optimization and speed improvement. In addition to operating smoothly, efficient code significantly enhances the user experience.

For example, the following code will print the value of the x variable at each step of the debugger.

6. What is the purpose of the “this” keyword in JavaScript?

The this keyword refers to the object that is executing the current function or method. It allows access to object properties and methods within the context of that object.

7. What is the difference between == and === operators in JavaScript?

The equality == operator is a comparison operator that compares two values and returns true if they are equal. The strict equality === operator is also a comparison operator, but it compares two values and returns true only if they are equal and of the same type.

For example, the following code will return true, because the values of the x and y variables are equal.

```
var x = 10;
var y = 10;

console.log(x == y);
```

8. What is the difference between “var” and “let” keywords in JavaScript?

The var and let keywords are both used to declare variables in JavaScript. However, there are some key differences between the two keywords.

The var keyword declares a global variable, which means that the variable can be accessed from anywhere in the code. The let keyword declares a local variable, which means that the variable can only be accessed within the block of code where it is declared.

```
{
  let x = 10;

  console.log(x); // 10
}
```


9. What are closures in JavaScript?

Closures (closureFn) are functions that have access to variables from an outer function even after the outer function has finished executing. They “remember” the environment in which they were created.

```
function outer() {
  var outerVar = "Hello";

  function inner() {
    console.log(outerVar);
  }

  return inner;
}

var closureFn = outer();
closureFn(); // Output: Hello
```

10. What is event delegation in JavaScript?

Event delegation is a technique where you attach a single event listener to a parent element, and that event listener handles events occurring on its child elements. It helps optimize

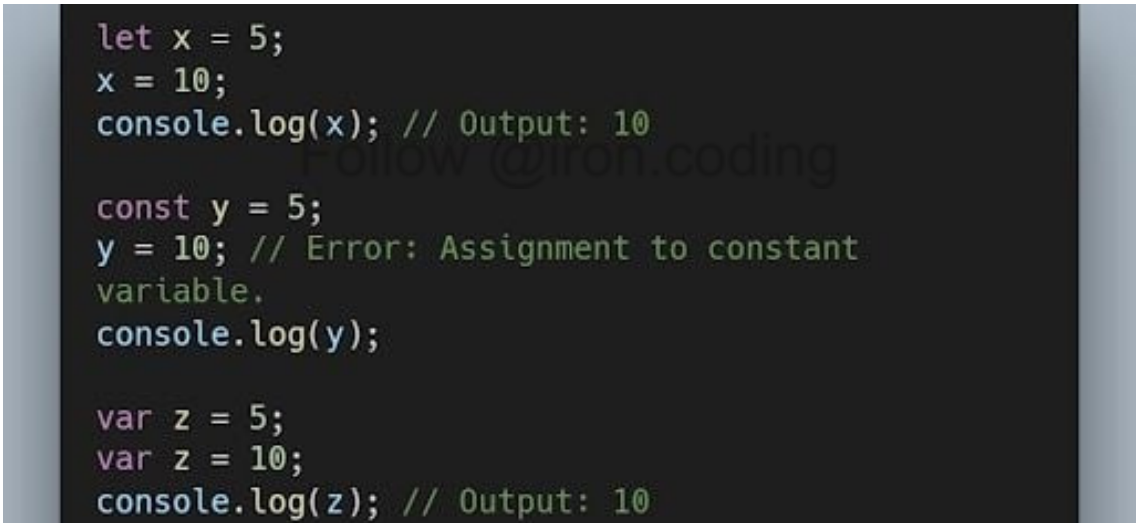
performance and reduce memory consumption.

```
// HTML:
<ul id="list">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>

// JavaScript:
document.getElementById("list").addEventListener
("click", function(event) {
  if (event.target.nodeName === "LI") {
    console.log(event.target.textContent);
  }
});
```

11. What is the difference between “let”, “const”, and “var”?

let and const were introduced in ES6 and have block scope. let is reassignable, and const is non-reassignable. var is function-scoped and can be redeclared and reassigned throughout the function.



```
let x = 5;
x = 10;
console.log(x); // Output: 10

const y = 5;
y = 10; // Error: Assignment to constant
variable.
console.log(y);

var z = 5;
var z = 10;
console.log(z); // Output: 10
```


12. What is implicit type coercion in JavaScript?

Implicit type coercion is a JavaScript concept that refers to the process of converting a value from one type to another. If you try to add a string to a number, JavaScript will implicitly coerce the string to a number before performing the addition operation.

For example, the following code will add the string "10" to the number 5. This is because JavaScript will implicitly coerce the string "10" to a number before performing the addition operation.

```
var x = 5;
var y = "10";

console.log(x + y); // 15
```

13. Explain the concept of prototypes in JavaScript.

Prototypes are a mechanism used by JavaScript objects for inheritance. Every JavaScript object has a prototype, which provides properties and methods that can be accessed by that object.

```
function Person(name) {
  this.name = name;
}

Person.prototype.greet = function() {
  console.log("Hello, " + this.name);
};

var person = new Person("John");
person.greet(); // Output: Hello, John
```

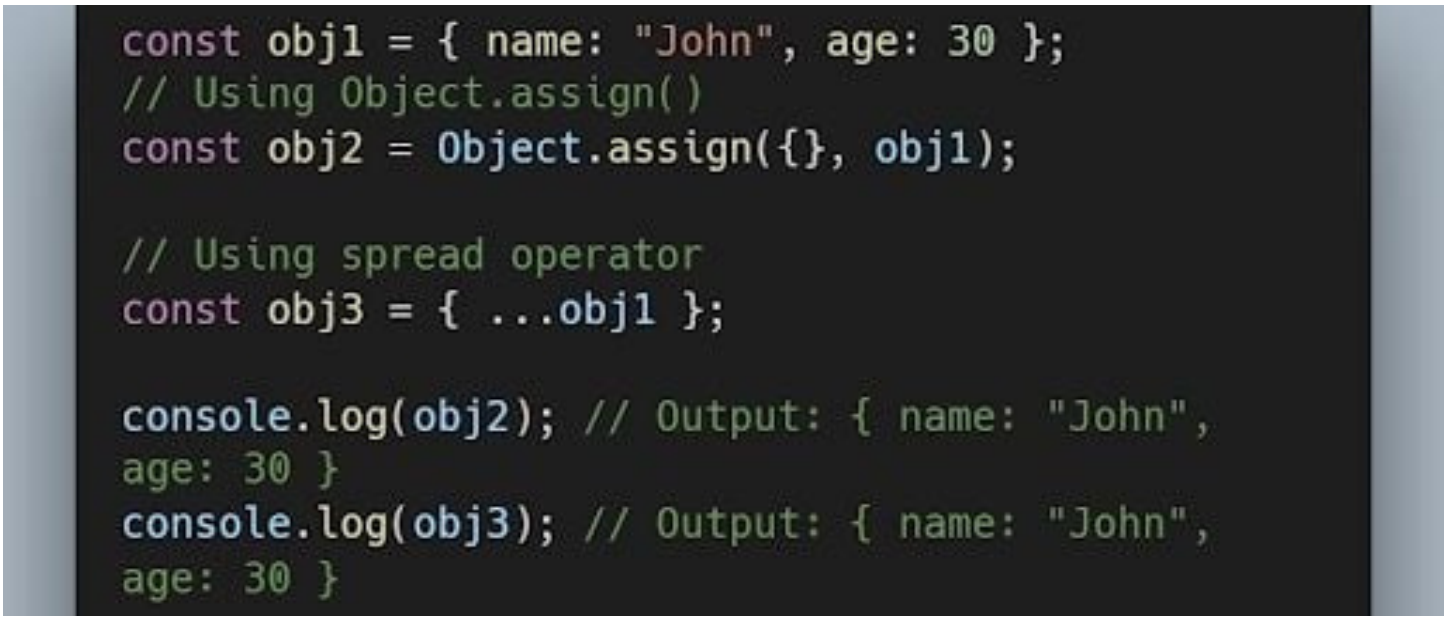
14. What is the output of the following code?

```
console.log(3 + 2 + "7");
```

The output will be "57". The addition operation is performed from left to right, and when a string is encountered, it performs concatenation.

15. How can you clone an object in JavaScript?

There are multiple ways to clone an object in JavaScript. One common method is using the Object.assign() method or the spread operator (...).



```
const obj1 = { name: "John", age: 30 };
// Using Object.assign()
const obj2 = Object.assign({}, obj1);

// Using spread operator
const obj3 = { ...obj1 };

console.log(obj2); // Output: { name: "John",
age: 30 }
console.log(obj3); // Output: { name: "John",
age: 30 }
```

Frequently Asked JavaScript Questions

Q.1 What is event delegation and how does it work?

Answer: Event delegation is a technique where you attach an event listener to a parent element instead of individual child elements, allowing you to handle events for dynamically added elements.

Q.2 Explain the concept of closures in JavaScript.

Answer: Closures occur when a function accesses variables from its outer scope even after the outer function has finished executing.

Q.3 What is the difference between == and ===?

Answer: == is a loose equality operator that performs type coercion, while === is a strict equality operator that checks both value and type.

Q.4 How do you handle asynchronous operations in JavaScript?

Answer: Asynchronous operations in JavaScript are handled using callbacks, promises, or async/await.

Q.5 What is the difference between null and undefined?

Answer: null represents the absence of a value, while undefined means a variable has been declared but not assigned a value.

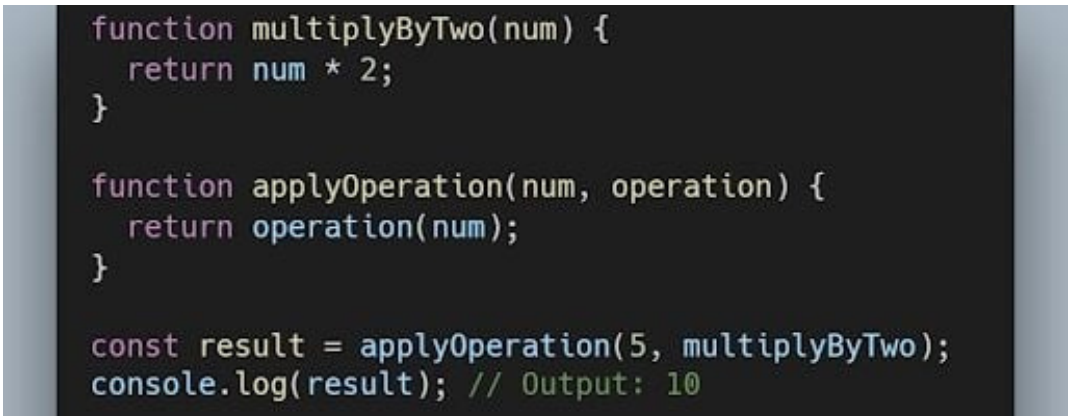
Q.6 What are the different types of errors in JavaScript?

Answer: JavaScript can throw a variety of errors, including:

- **Syntax errors:** These errors occur when the JavaScript code is not syntactically correct.
- **Runtime errors:** These errors occur when the JavaScript code is executed and there is a problem.
- **Logical errors:** These errors occur when the JavaScript code does not do what it is supposed to do.

Q.7 What are higher-order functions in JavaScript?

Answer: Higher order functions are functions that can accept other functions as arguments or return functions as their results. They enable powerful functional programming patterns in JavaScript.



Q.8 What is the purpose of the bind() method in JavaScript?

Answer: The bind() method is used to create a new function with a specified this value and an initial set of arguments. It allows you to set the context of a function permanently.

Q.9 What is the difference between function declarations and function expressions?

Answer: Function declarations are defined using the function keyword, while function expressions are defined by assigning a function to a variable. Function declarations are hoisted, while function expressions are not.

Q.10 What is memoization in JavaScript?

Answer: Memoization is a technique that can be used to improve the performance of JavaScript code. Memoization works by storing the results of expensive calculations in a cache. This allows the JavaScript code to avoid re-performing the expensive calculations if the same input is provided again.

For example, the following code calculates the factorial of a number. The factorial of a number is the product of all the positive integers from one to the number.

```
function factorial(n) {
  if (n === 0) {
    return 1;
  } else {
    return n * factorial(n - 1);
  }
}
```

This code can be memoized as follows:

```
function factorial(n) {
  if (factorialCache[n] !== undefined) {
    return factorialCache[n];
  } else {
    factorialCache[n] = n * factorial(n - 1);
    return factorialCache[n];
  }
}
```

Q.11 What is recursion in JavaScript?

Answer: Recursion is a programming technique that allows a function to call itself. Recursion can be used to solve a variety of problems, such as finding the factorial of a number or calculating the Fibonacci sequence.

Q.12 What is the use of a constructor function in JavaScript?

Answer: A constructor function is a special type of function that is used to create objects. Constructor functions are used to define the properties and methods of an object.

Q.13 What is the difference between a function declaration and a function expression in JavaScript?

Answer: A function declaration is a statement that defines a function. A function expression is an expression that evaluates to a function.

Q.14 What is a callback function in JavaScript?

Answer: A callback function is a function passed as an argument to another function, which is then invoked inside the outer function. It allows asynchronous or event-driven programming.

Q.15 What are promises in JavaScript?

Answer: Promises are objects used for asynchronous operations. They represent the eventual completion or failure of an asynchronous operation and allow chaining and handling of success or error cases.

Q.16 What is the difference between synchronous and asynchronous programming?

Answer: In synchronous programming, the program execution occurs sequentially, and each statement blocks the execution until it is completed. In asynchronous programming, multiple tasks can be executed concurrently, and the program doesn't wait for a task to finish before moving to the next one.

Q.17 How do you handle errors in JavaScript?

Errors in JavaScript can be handled using try-catch blocks. The try block contains the code that may throw an error, and the catch block handles the error and provides an alternative execution path.

Q.18 Explain the concept of event bubbling in JavaScript.

Answers: Event bubbling is the process where an event triggers on a nested element, and then the same event is propagated to its parent elements in the document object model (DOM) tree. It starts from the innermost element and goes up to the document root.

Q.19 What are arrow functions in JavaScript?

Answers: Arrow functions are a concise syntax for writing JavaScript functions. They have a more compact syntax compared to traditional function expressions and inherit the this value from their surrounding scope.

Q.20 What is the difference between querySelector and getElementById?

Answers: querySelector is a more versatile method that allows you to select elements using CSS-like selectors, while getElementById specifically selects an element with the specified ID.

Q.21 What is the purpose of the setTimeout() function in JavaScript?

Answers: The setTimeout() function is used to delay the execution of a function or the evaluation of an expression after a specified amount of time in milliseconds.

Top 10 JavaScript coding interview questions

Link: click here

Frameworks/Libraries

Q.1 What is the difference between Angular, React, and Vue.js?

Q.2 Explain the concept of virtual DOM in React.

Q.3 How does data binding work in Angular?

Q.4What is the purpose of state management libraries like Redux in React?

Q.5 What are the advantages of using a front-end framework like Bootstrap or Foundation?

Backend Development

Q.1 Compare and contrast the features of Node.js, Django, and Ruby on Rails.

Q.2 What is RESTful API and how do you design one?

Q.3 Explain the difference between SQL and NoSQL databases.

Q.4 How do you handle authentication and authorization in web applications?

Q.5 What is middleware in Express.js?

Testing and Debugging

Q.1 What are unit tests and how do you write them for web applications?

Q.2 Explain the difference between black-box testing and white-box testing.

Q.3 How do you debug JavaScript code?

Performance Optimization

Q.1 What techniques would you use to improve the performance of a website?

Q.2 How does browser caching work and how can you leverage it?

Q.3 What is lazy loading and why is it useful?

Q.4 What are common security vulnerabilities in web applications and how do you prevent them?

Q.5 Explain the concept of XSS (Cross-Site Scripting) and how to mitigate it.

Q.6 What is CSRF (Cross-Site Request Forgery) and how can it be prevented?

General

Q.1 Describe your workflow for developing a web application from start to finish.

Q.2 How do you stay updated with the latest trends and technologies in web development?

Q.3 Can you explain a challenging problem you faced in a project and how you solved it?

Q.4 What are your favorite development tools and why?

Q.5 Do you have experience working in an Agile/Scrum environment?

These questions cover a wide range of topics commonly encountered in web development interviews. Depending on the specific role and company, the interviewer may focus more on certain areas than others.