1. What is an array in JavaScript? How do you declare and initialize an array?

(ans) arrays are a special type of objects. The typeof operator in javascript returns “objects” for arrays. Using an array literal is the easiest way to create a javascript array. Syntax: const array\_name=[item1,item2,…] it is a common practice to declare arrays with the const keyword.

1. Explain the methods push(), pop(), shift(), and unshift() used in arrays.

(ans) push(): add elements to the end of an array. Array push. Using push() we can add elements to the end of an array.

Pop(): remove an element from end of an array.

Shift(): remove an element from the front of an array.

Unshift(): add elements from the front of an array.

Objects

1. What is an object in JavaScript? How are objects different from arrays?

(ans) arrays are used when we need to collect a list of elements of the same data type or structure. On the other hand, objects are used when grouping multiple sets of data that belong together using labels, where each property or key has its own value of any type.

1. Explain how to access and update object properties using dot notation and bracket notation.

(ans) to access javascript object properties using bracket notation, the key should be

1. Property identifiers have to be string or a variable that references a string.
2. We can use variables, spaces, and strings that starts with numbers.
3. It can be an expression.
4. Obj[“prop”], obj[“prop name”] is correct.

JavaScript Events

1. What are JavaScript events? Explain the role of event listeners

(ans) javascript’s event listener function allows you to create custom responses to events like mouse clicks, keyboard clicks, and window resizing. The programming paradigm of waiting and responding to real-time events is called event handling.

1. How does the addEventListener() method work in JavaScript? Provide an example.

(ans) the addeventlistener function takes two arguments: the type of the event you want to listen for (e.g. “click” or “keydown”) and the function that should be called when the event is detected. Let button = document.queryselector(‘#my-button’); button.

DOM Manipulation

1. What is the DOM (Document Object Model) in JavaScript? How does JavaScript interact with the DOM?

(ans) the document object model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

1. Explain the methods getElementById(), getElementsByClassName(),and querySelector() used to select elements from the DOM.

(ans) getElementById(): selects an element with a specific id attribute. It returns a single dom element that matches the given id.

getElementsByClassName(): selects all elements with a specific class name. it returns a html collection of all elements with the specified class name.

querySelector(): selects the first element that matches a css selector. This method allows you to use any valid css selector to target an element.

JavaScript Timing Events (setTimeout, setInterval)

1. Explain the setTimeout() and setInterval() functions in JavaScript. Howare they used for timing events?

(ans) there are two methods for it: settimeout allows us to run a function once after the interval of time. Setinterval allows us to run a function repeatedly, starting after the interval of time, then repeating continuously at that interval.

1. Provide an example of how to use setTimeout() to delay an action by 2 seconds.

(ans) the settimeout() method calls a function after a number of milliseconds.

1 second = 1000 milliseconds. The settimeout() is executed only once. If we need repeated executions, use setinterval() instead. Use the cleartimeout() method to prevent the function from starting.

JavaScript Error Handling

1. What is error handling in JavaScript? Explain the try, catch, and finally blocks with an example.

(ans) the try statement defines the code block to run (to try). The catch statement defines a code block to handle any error. The finale statement defines a code block to run regardless of the result. The throw statement defines a custom error.

1. Why is error handling important in JavaScript applications?

(ans) effective error handling isn’t just about preventing crashes, but building a better user experience and ultimately shipping more reliable applications. As you continue your javascript journey, make exception handling an integral part of your coding practice.