JAVA SCRIPT

- What is javascript ?

JavaScript (js) is a light-weight object-oriented interpreted programming language. Its is commonly used to create dynamic and interactive element in web applications. Javasecript is a very easy to emplement because it is integrated with HTML. It is open and cross-platform.

- History of a Javascript ?

JavaScript was initially created by Brendan Eich of NetScape and was first announced in a press release by Netscape in 1995. It has a bizarre history of naming; initially, it was named Mocha by the creator, which was later renamed LiveScript. In 1996, about a year later after the release, NetScape decided to rename it to JavaScript with hopes of capitalizing on the Java community (although JavaScript did not have any relationship with Java) and released Netscape 2.0 with the official support of JavaScript.

- Java Script Versions

JavaScript, invented by Brendan Eich, achieved the status of an ECMA standard in 1997 and adopted the official name ECMAScript. This language has evolved through several versions, namely ES1, ES2, ES3, ES5, and the transformative ES6. These updates have played a crucial role in improving and

standardizing JavaScript, making it widely used and valuable in the ever-changing field of web development.

• How to run Java secript

JavaScript can be run in the browser by including the external script file using the `script` tag, writing it within the HTML page using the `script` tag again, running it in the browser console or you can also use REPL.

Here are some of the most important and commonly asked questions in JavaScript:

1. What are JavaScript Data Types?

- JavaScript has 7 primitive data types: `Undefined`, `Null`, `Boolean`, `Number`, `BigInt`, `String`, and `Symbol`.
- Non-primitive type: `Object`.

2. What is the difference between `null` and `undefined`?

- `null` is an assignment value representing no value or object.
- `undefined` is a type itself and indicates that a variable has been declared but not assigned a value.

3. What is the difference between `let`, `const`, and `var`?

- `var`: Function-scoped, can be re-assigned, and can be re-declared in the same scope.
- `let`: Block-scoped, can be re-assigned but not re-declared in the same scope.
- `const`: Block-scoped, cannot be re-assigned or re-declared.

4. What is a Closure?

- A closure is a function that remembers its lexical scope even when the function is executed outside that scope. This allows for the function to access variables from its outer function.

5. What is a callback function in JavaScript?

- A callback function is a function passed as an argument to another function, to be executed later once the task is completed.

6. What is the event loop in JavaScript?

- The event loop is a mechanism that allows JavaScript to execute non-blocking code by handling asynchronous tasks (callbacks, promises, etc.) in a single-threaded environment.

7. What is the `this` keyword?

- `this` refers to the context in which the function is called. It changes depending on how the function is invoked.

8. What are Promises in JavaScript?

- A Promise is an object representing the eventual completion (or failure) of an asynchronous operation. It has three states: pending, fulfilled, and rejected.

9. What is async/await?

- `async/await` is a modern approach to handle asynchronous code. `async` makes a function return a promise, and `await` pauses the execution of the function until the promise is resolved.

10. What is the difference between == and ===?

- == is the equality operator that performs type coercion.
- === is the strict equality operator that checks both value and type without performing coercion.

11. What is an arrow function?

- An arrow function is a concise way to write functions in JavaScript. It does not have its own `this` binding and inherits `this` from the outer lexical scope.

12. What is hoisting?

- Hoisting is JavaScript's default behavior of moving variable and function declarations to the top of their containing scope during compile time.

13. What is the spread/rest operator?

- Spread (. . .) is used to expand elements of an array or object.
- Rest (. . .) is used to collect remaining arguments into an array.

14. What is destructuring in JavaScript?

- Destructuring allows you to unpack values from arrays or objects into distinct variables in a more concise way.

15. What is the difference between `apply()`, `call()`, and `bind()`?

- `apply()` and `call()` both invoke a function with a specified `this` context, but `apply()` takes an array as arguments, while `call()` takes arguments individually.
- `bind()` creates a new function that, when invoked, has its `this` context set to the specified value.

16. What is a JavaScript Module?

- A JavaScript module is a file that contains code which can be exported and imported into other files to be used. It allows for better organization and code reuse.

17. What are higher-order functions?

- Higher-order functions are functions that take other functions as arguments or return functions as results.

18. What is a Promise chain?

- A Promise chain allows you to sequence asynchronous operations by chaining `.then()` and `.catch()` to handle multiple promises.

19. What is the difference between `setTimeout()` and `setInterval()`?

- `setTimeout()` runs a function once after a specified delay.
- `setInterval()` runs a function repeatedly at specified intervals.

20. What are JavaScript "falsy" values?

- The falsy values in JavaScript are: `false`, `0`, `""` (empty string), `null`, `undefined`, and `NaN`.