

Q1) - What is the difference between Java & JavaScript?

- **Java vs JavaScript: Key Difference**

The primary difference between Java and JavaScript is that Java is an OOP programming language, whereas JavaScript is an OOP programming script. JavaScript code is entirely written in text and only needs to be translated. Contrarily, Java needs to be compiled.

Q2. What is JavaScript?

JavaScript is a programming language used for web development. It runs in web browsers and helps make web pages interactive and dynamic. It's mainly used for client-side scripting, responding to user actions like clicks. JavaScript supports asynchronous programming, has object-oriented features, and is also used on the server side with technologies like Node.js. It's a key tool for creating modern, responsive web applications.

Q3. What are the data types supported by JavaScript?

String: Represents a sequence of characters, e.g., "Hello, World!".

Number: Represents numeric values, including integers and floating-point numbers, e.g., 42 or 3.14.

Boolean: Represents a logical value indicating either true or false.

Undefined: Represents a variable that has been declared but not assigned any value.

Null: Represents the intentional absence of any object value.

Symbol: Represents a unique identifier. Symbols are often used as property keys in objects to avoid naming conflicts.

Object: Represents a collection of key-value pairs. Objects can hold various data types and functions.

BigInt: Represents integers of arbitrary precision. It allows you to work with numbers beyond the limit of the Number type.

Q4. What are the scopes of a variable in JavaScript?

Global Scope:

- Variables declared outside of any function or block have global scope.
- They can be accessed and modified from any part of the code, including inside functions.

Local Scope:

- Variables declared inside a function or block have local scope.
- They are only accessible within the function or block where they are declared.
- Local variables take priority over global variables with the same name inside the same function.

Block scope:

- Block scope allows variables to be scoped to a specific block of code.
- Block scope helps in preventing unintended variable hoisting and makes the code more predictable and maintainable.

Q5. What is Callback?

A callback is a function that is passed as an argument to another function and is executed after the completion of a specific task. Callbacks are commonly used in asynchronous operations, such as handling events, making API requests, or reading files. They play a crucial role in managing the flow of execution in asynchronous and event-driven programming.

Q6. What is Closure? Give an example.

A closure is a function having access to the parent scope, even after the parent function has closed.

Example

```
const add = (function () {  
    let counter = 0;  
    return function () {counter += 1; return counter}  
})();  
// self-invoking function is assign to variable add  
// The self-invoking function only runs once. It sets the counter to zero (0),  
// and returns a function expression. This way add becomes a function.  
// This is called a JavaScript closure  
  
add();  
add();  
add();  
// the counter is 3 now
```

Q7. What is the difference between the operators '==' & '==='?

In JavaScript, == and === are comparison operators used to check equality between two values or variables.

The main difference between the two operators is how they compare values. The == operator compares the values of two variables after performing type conversion if necessary. On the other hand, the === operator compares the values of two variables without performing type conversion.

Q8. What is the difference between null & undefined?

Null means an empty or non-existent value. Null is assigned, and explicitly means nothing.

Undefined means a variable has been declared, but the value of that variable has not yet been defined.

Q9. What would be the result of 2+5+"3"?

In JavaScript, the + operator is used for both addition and string concatenation. When the + operator encounters a string, it performs string concatenation instead of addition.

so, by breaking down of the expression 2 + 5 + "3"

2+5 =7

7+"3" =73 (3 is string)

Result = 73

Q10. What is the difference between Call & Apply?

- The call() method takes arguments separately.
- The apply() method takes arguments as an array.