Date- 18/11/2025

Assignment

1. Difference between var, let, and const.

	let	var	const
Redeclaration	Not Allowed (in same scope)	Allowed	Not Allowed
Reassignment	Allowed	Allowed	Not Allowed
Scope	Block scope {}	Function () or Global scope	Block scope {}
Hoisting	Hoisted but not initialized (Temporal Dead Zone)	Hoisted and initialized with undefined	Hoisted but not initialized (Temporal Dead Zone)
Declaration & Initialization	Can be declared without initialization	Can be declared without initialization	Cannot be declared without initialization

2. Function

Function is a block of code that is used to perform a specific task. It can accept input as parameters, execute a set of instructions and returns a value. It allows code reusability. Java Script supports both user defined and built-in functions.

Function Declaration:

It can be declared by using function keyword followed by name of function and parenthesis and block of code contained in curly brackets '{}'.

For e.g.

```
function Vedika(){
        console.log("My name is Vedika!");
}
```

Function Calling:

Function is called or invoked by using function name followed by parenthesis.

For e.g.

```
Vedika();
```

3. **What is Hoisting in JavaScript?**

   Hoisting refers to the behaviour where JavaScript moves the declarations where JavaScript moves declarations of variables, functions, and classes to the top of their scope during the compilation phase. This can sometimes lead to surprising results, especially when using var, let, const or function expression.

   For e.g.

   ```
   a = 10;
   console.log(a);
   var a;
   ```

4. **Difference between Statically typed language and Dynamically typed language.**

Statically typed language	Dynamically typed language
Type checking is performed at compile-time	Type checking is performed at runtime
Type of a variable is known at compile time	Type of variable can be changed at runtime

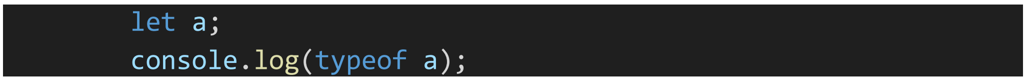
Date- 18/11/2025

Programmer must specify the type of data	Programmer need not to specify type of data
For e.g. C++, C, C#, Java	For e.g. JavaScript, PHP, Python

5. typeof

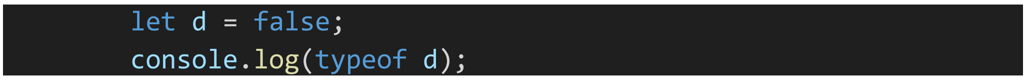
The typeof operator in JavaScript is used to determine the data type of a variable or value. It returns a string indicating the type.

1) Undefined- For undefined variable



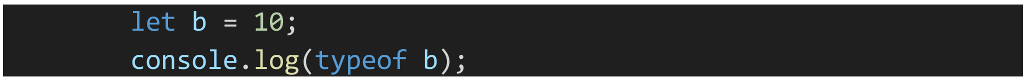
```
let a;
console.log(typeof a);
```

2) Boolean- For true or false



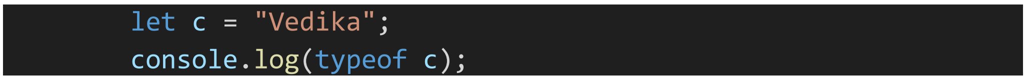
```
let d = false;
console.log(typeof d);
```

3) Number- For numeric values



```
let b = 10;
console.log(typeof b);
```

4) String- For text in quotes



```
let c = "Vedika";
console.log(typeof c);
```

When the typeof operator is applied to a function, it returns the string “function”, indicating that the operand is a function.

6. Difference between array in Java and JavaScript

Array in Java	Array in JavaScript
Size is fixed, determined at creation	Size is dynamic, arrays in javascript are resizable

Java arrays can only store elements of the same data type	JavaScript arrays can hold elements of different data types within the same array.
Contiguous memory for primitives	Not necessarily contiguous