Lab Assignment 3

1.Explain the differences between var, let, and const with respect to scope and hoisting.

Ans :-

Scope

- var (Function Scope / Global Scope): Variables declared with var are scoped to the nearest function or, if declared outside any function, to the global scope. They are not limited to the block (e.g., inside an if statement or for loop) in which they are defined.

EX:-

```
if(true) {
var x = 10;
}
console.log(x); //Output:10
```

- let and const (Block Scope): Variables declared with let and const are block-scoped. A block is any code enclosed within {} curly braces (e.g., loops, if statements, or simple blocks). The variable is not accessible outside that specific block.

EX:-

```
if (true) {
  lety= 10;
  const z = 20;
} console.log(y); // ReferenceError: y is not defined
console.log(z); // ReferenceError: z is not defined
```

Hoisting

Hoistingis a JavaScript mechanism where variable and function declarations are moved to the top of their scope during the compilation phase. The key difference lies in how they are initialized.

var (Hoistedand Initialized): var declarations are hoisted to the top of their scope(function or global) and automatically initialized with the value undefined.This means you can access a var before it is declared in the codewithouta ReferenceError.

EX:-

```
console.log(a); // Output: undefined
var a = 5;
```

let and**const** (Hoisted but Uninitialized - TDZ): let and const declarations arealsohoisted, but they are not initialized with undefined. They remain uninitializedand are placed in a Temporal Dead Zone (TDZ) from the start oftheblockuntil the declaration line is executed. Attempting to access themwithinthe TDZ results in a ReferenceError.

EX:-

```
console.log(b);
letb =5;
console.log(c);
constc=5;
```

2. Describe the various Control Flow statements in JavaScript, specifically highlighting the difference between for, while, and do-while loops.

Ans :-

JavaScriptuses several control flowstatements to execute code blocks conditionally or repeatedly. These primarily include conditional statements (if...else, switch) and loops (for, while, do-while), as well as other statements like break and continue [1].

The primary difference between the for, while, and do-while loops lies in their syntax and the timing of their condition check.

Loops

Loops are used to repeatedly run a block of code until a specific condition is met.

for loop

The for loop is most commonly used when you know exactly how many times you want the loop to run. It combines the initialization of a counter, the condition to check, and the counter update all in a single line.

- **Syntax:** for(initialization; condition; increment/decrement) { code block }
- **Condition Check:** The condition is checked before each iteration.

EX :-

```
for(let i =0; i < 5; i++){
console.log(i); // Prints 0, 1, 2, 3, 4
}
```

while loop

The while loop is used when the number of iterations is not known in advance, and the loop should continue as long as a specified condition remains true.

- **Syntax:** while (condition) { code block }
- **Condition Check:** The condition is checked before each iteration. If the condition is false initially, the code block is never executed.

EX :-

```
let i = 0;
while (i < 5) {
console.log(i); // Prints 0, 1, 2, 3, 4
i++;
}
```

do-while loop

The do-while loop is similar to the while loop, with one crucial difference.

- **Syntax:** do { code block } while (condition);
- **Condition Check:** The condition is checked after the code block is executed. This guarantees that the loop body will run at least once, regardless of the initial condition.

EX :-

```
let i = 5; // The condition (i < 5) is initially false
do {
console.log(i); // Prints 5
i++;
```

} while (i < 5); // The loop body runs once, then the condition is checked and found to be false.

3. What is the Document Object Model (DOM)? Explain how to select elements and modify their content using innerText and innerHTML.

Ans :-

The DOM is a standard programming interface for web documents. It is not the HTML code itself, but rather a representation of that code created by the browser.

- **Tree Structure:** When a web page loads, the browser converts the HTML tags into a tree-like structure of objects.
- **Nodes:** Every part of the document (elements like , text inside elements, and attributes) is a "node" in this tree.
- **Bridge:** It acts as a bridge connecting your static HTML structure to dynamic JavaScript, allowing you to change styles, content, and attributes programmatically.

➢ Selecting Elements

To manipulate the DOM, you first need to "select" or "grab" the specific element you want to change. Here are the two primary methods used in modern JavaScript:

❖ getElementById()

This is the most common method for beginners. It selects a single element based on its unique ID attribute.

- **Syntax:** document.getElementById("id-name")
- **Return Value:** The element object (if found) or null.

❖ **querySelector()**

This is a more versatile method.It allows you to select an element using CSS selectors (like classes, IDs, or tag names).

- **Syntax:** document.querySelector(".class-name") or document.querySelector("h1")
- **Return Value:** The first matching element found.

➢ **Modifying Content:** innerText vs innerHTML

Once you have selected an element, you can modify its content. In an assignment, it is critical to distinguish between these two properties.

❖ **innerText**

This property represents the visible text contained in a node.

- **Function:** It treats content as raw text.
- **Behavior:** If you assign a string containing HTML tags (like), they will not be rendered as bold; they will literally appear as text characters on the screen.
- **Best for:** Updating names, numbers, or simple text messages where no styling is required inside the text.

❖ **innerHTML**

This property represents the HTML markup contained within the element.

- **Function:** It parses the content as HTML.
- **Behavior:** If you assign a string containing HTML tags, the browser will read them and render the visual result (e.g., making text bold, creating links, or adding lists).
- **Best for:** Inserting complex structures, formatted text, or new elements dynamically.

Name:- Ashutosh Kr. Rana ERP:- 10321