**1. With sited examples give the different between Let, Const and Var keywords in JavaScript.**

**Scope**:

* **Var**: Has function scope, meaning variables declared with var are accessible throughout the entire function, even outside the block where they are defined.
* **Let** and **Const**: Both have block scope, meaning variables declared within a block ({}) are accessible only within that block.

**Re-declaration**:

* **Var**: Allows re-declaration of the same variable within the same scope.
* **Let** and **Const**: Do not allow re-declaration within the same scope. Attempting to do so will throw a syntax error.

**Re-assignment**:

* **Var**: Allows re-assignment of values.
* **Let**: Also allows re-assignment of values.
* **Const**: Does not allow re-assignment. Variables declared with const must be initialized at the time of declaration and cannot be changed later.

**Usage**:

* **Var**: Should generally be avoided in modern JavaScript because of its function scoping and potential for unexpected behavior.
* **Let**: Ideal for variables that will change over time (e.g., counters, loop variables).
* **Const**: Best suited for variables that should remain constant and not be reassigned (e.g., configuration settings, constants).

**Examples**

**a. var Scope**

if (true) {

var x = 10;

}

console.log(x);

**b. Let Scope**:

if (true) {

let y = 20;

}

console.log(y);

**c. const Assignment**:

const z = 30;

z = 40;