`Var VS let & const`

`var`, `let`, and `const` are all keywords used to declare variables in JavaScript, but they differ in their scope and behavior. `var` is function-scoped, while `let` and `const` are block-scoped. `let` allows reassignment of the variable's value, while `const` declares a constant that cannot be reassigned after initialization. In modern JavaScript, `let` and `const` are generally preferred over `var` due to their more predictable scoping.

Detailed Explanation:

- `var`:
- **Scope:** Function-scoped. A variable declared with `var` is accessible throughout the entire function in which it is declared, or globally if declared outside of a function.
- **Hoisting:** `var` declarations are hoisted to the top of their scope, meaning the variable is available before the line of code where it's declared. However, the value is `undefined` until the line where it's assigned.
- **Reassignment:** `var` allows reassignment of the variable's value.
- `let`:
- **Scope:** Block-scoped. A variable declared with `let` is only accessible within the block (enclosed by curly braces `{}`) where it's defined.
- **Hoisting:** `let` declarations are also hoisted, but unlike `var`, they are not initialized to `undefined` before the line of declaration. Accessing a `let` variable before its declaration results in a `ReferenceError` (Temporal Dead Zone).
- **Reassignment:** `let` allows reassignment of the variable's value.
- `const`:
- **Scope:** Block-scoped, like `let`.
- **Hoisting:** Similar to `let`, `const` declarations are hoisted but not initialized.
- **Reassignment:** `const` declares a constant, meaning its value cannot be reassigned after initialization. However, if the `const` variable holds an object or array, the contents of that object or array can be modified.

Example:

JavaScript

```javascript
function example() {
  var x = 10;
  let y = 20;
  const z = 30;

  if (true) {
    var x = 100; // Reassigns the outer var
    let y = 200; // Creates a new let within the block
    const z = 300; // Creates a new const within the block
    console.log("Inside block:", x, y, z); // Output: Inside block: 100 200 300
  }

  console.log("Outside block:", x, y, z); // Output: Outside block: 100 20 30
}

example();
```

Best Practices:

- In modern JavaScript, it is generally recommended to use `let` and `const` instead of `var` to avoid potential scoping issues and create more predictable code.
- Use `const` for variables that should not be reassigned and `let` for variables that might need to be reassigned.
- Be mindful of the Temporal Dead Zone when using `let` and `const`.