VARIABLE DECLARATION

In JavaScript, you can declare variables using three keywords: var, let, and const. The choice of which keyword to use depends on the scope and mutability requirements of the variable.

1. var (Function-scoped or Globally-scoped):

Variables declared with var are function-scoped or globally-scoped, meaning they are accessible throughout the entire function or globally if declared outside any function.

eg: var x = 10;

Variables declared with var are hoisted, which means the declaration is moved to the top of its scope during the compilation phase.

eg: function example() {

```
  console.log(y); // Outputs: undefined
  var y = 20;
  console.log(y); // Outputs: 20
}
```

2. let (Block-scoped, Reassignable):

eg: let a = 5;

Variables declared with let are block-scoped, meaning they are only accessible within the block or statement where they are defined.

Unlike var, variables declared with let are not hoisted until the line of code where they are defined.

eg: if (true) {

```
  let b = 10;
  console.log(b); // Outputs: 10
}
// console.log(b); // Error: b is not defined
```

3. const (Block-scoped, Immutable):

eg: const PI = 3.14;

Variables declared with const are block-scoped like let, but they cannot be reassigned once they are assigned a value.

It's important to note that while const makes the variable itself immutable, it doesn't make the object it points to immutable. If the variable is an object or an array, the properties or elements of that object or array can still be modified.

In modern JavaScript, it's generally recommended to use let and const over var for better scoping and to avoid potential issues associated with hoisting. Use let when you need to reassign a variable, and use const when you want to create an immutable variable.