

- var is function-scoped or globally-scoped, meaning that if it is declared within a function, it is only accessible within that function. If it is declared outside any function, it is accessible from anywhere in the code. Variables declared with var can be reassigned and redeclared.

- let is block-scoped, meaning that it is only accessible within the block of code in which it is defined. Variables declared with let can be reassigned, but cannot be redeclared within the same block.

- const is also block-scoped, but variables declared with const cannot be reassigned or redeclared.

- scope: Determines where in your code a variable can be accessed; which part of your script can access the variable.

- global scope: Global scope is the default scope for variables declared outside any function. Global variables can be accessed from anywhere in your code.

- Block scope is the scope for variables declared inside a block of code. A block of code is a group of statements enclosed in curly braces ({ }).

-

Keyword	Scope	Can be reassigned?	Can be redeclared?
var	Function or global	Yes	Yes
let	Block	Yes	No
const	Block	No	No

Here are some additional examples of how and when to use let and const:

`let` can be used to declare variables that are used in loops or conditional statements. For example, you could use `let` to declare a counter variable in a loop.

`const` can be used to declare variables that represent constants, such as mathematical constants or environmental variables. For example, you could use `const` to declare the speed of light or the pi constant.

`let` can be used to declare variables that are passed as arguments to functions. For example, you could use `let` to declare a variable that represents the number of items in an array.

`const` can be used to declare variables that are returned by functions. For example, you could use `const` to declare a variable that represents the result of a mathematical calculation.

In JavaScript, there are three keywords that you can use to declare variables: `var`, `let`, and `const`.

`var` is the oldest of these keywords, and it's been around since the beginning of JavaScript. `let` and `const` were introduced in ES6 (in 2015), which is the sixth major version of the JavaScript language.

`let` and `const` are very similar, but there are a few key differences between them. The first difference is that `let` and `const` have different scopes. The scope of a variable is the part of your code where the variable is visible. `Var` variables have global scope, which means that they're visible throughout your entire code. `Let` and `const` variables have block scope, which means that they're only visible within the block of code where they're declared.

The second difference between `let` and `const` is that `let` variables can be reassigned, but `const` variables cannot. This means that you can change the value of a `let` variable, but you can't change the value of a `const` variable.