**Declaring Variables:**

* **Var**

function scoped and can’t be accessible outside the function.

* **let Keyword**

let creates a local variable in JavaScript & can be re-assigned.

Initialization during the declaration of a let variable is optional.

A let variable will contain undefined if nothing is assigned to it.

* **const Keyword**

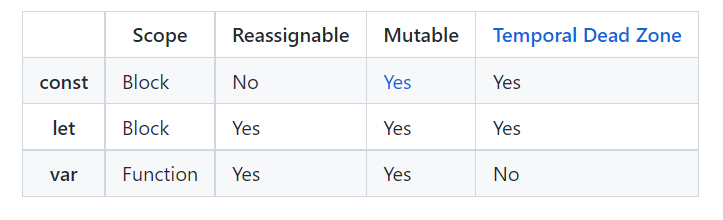
A constant variable can be declared using the keyword const.

It must have an assignment.

Any attempt of re-assigning a const variable will result in JavaScript runtime error.

Constant variables *must* be assigned a value when declared.

If you try to declare a const variable without a value, we will get a Syntax Error.

****

* Javascript is case sensitive.
* **Console.log()** – console is the object ; whatever is put inside the parenthesis it will be printed.
* document.write() inserts text into the document (the webpage in the browser window); console.log() does not and it is special one included in the browser for developers to debug and test their code.
* The difference is scoping. var is scoped to the **nearest function block** and let is scoped to the nearest ***enclosing* block**, which can be smaller than a function block. Both are **global** if outside any block.
* **String Interpolation -** a template literal is wrapped by backticks ` (this key is usually located on the top of your keyboard, left of the 1 key).
* Inside the template literal, you’ll see a placeholder, ${myPet}. The value of myPet is inserted into the template literal.
* Used for readability of code.

**Operators:**

**>, <, <=, >=, ==, ===, !=, !==, &&, ||, !**

**Conditionals :**

1. IF
2. If else
3. IF elseIf else
4. Switch
5. Ternary operator
6. Comparison operator
7. Logical Operator
8. short-circuit evaluation – checking the two cases using or operator.EX : let defaultName = username || 'Stranger';

**SCOPE :**

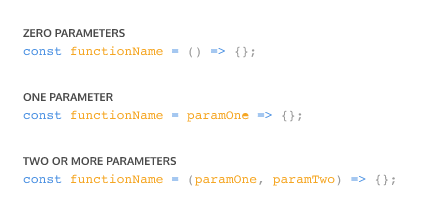
1. In global scope, variables are declared outside of blocks. These variables are called global variables.
2. Variables that are declared with block scope are known as local variables because they are only available to the code that is part of the same block.
3. Scope pollution is when we have too many global variables that exist in the global namespace, or when we reuse variables across different scopes. Scope pollution makes it difficult to keep track of our different variables and sets us up for potential accidents.

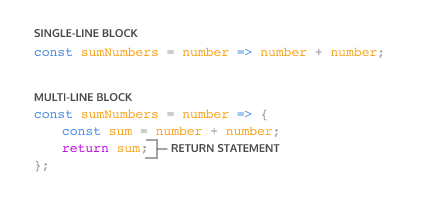
**NOTES:**

* Empty strings like "" or ''
* null which represent when there is no value at all
* undefined which represent when a declared variable lacks a value
* NaN, or Not a Number

**FUNCTIONS:**

1. A *function* is a reusable block of code that groups together a sequence of statements to perform a specific task.
2. Function functionname() {}
3. If a variable is passed it is a parameter and if a value for that variable is passed it is called an argument.
4. Function called within another function is called Helper Function.
5. Function Expression -> there is no function name and instead it is assigned to a const variable.
6. [**Arrow functions**](https://www.codecademy.com/resources/docs/javascript/arrow-functions?page_ref=catalog) remove the need to type out the keyword function every time you need to create a function. Instead, you first include the parameters inside the ( ) and then add an arrow => that points to the function body.
7. Concise arrow function:





Functions :

A higher-order function is a function that either accepts functions as parameters, returns a function, or both! We call functions that get passed in as parameters call back functions. Call back functions get invoked during the execution of the higher-order function.