VARIABLE TYPES

<u>**Variables Types**</u>

1	Let	• The let keyword was introduced in ES6 (2015). • Variables declared with let have Block Scope. • Variables declared with let must be Declared before use. • Variables declared with let cannot be Redeclared in the same scope. • The let keyword in JavaScript is used to declare block-scoped variables, potentially limiting the scope of a variable to the block, statement, or expression in which it is used.	
2	Scope	• JavaScript had Global Scope and Function Scope. • ES6 introduced the two new JavaScript keywords: let and const. These two keywords provided Block Scope in JavaScript.	

3	Let Example 1	• Variables declared inside a { } block cannot be accessed from outside the block
		• {
		let x = 2;
		}
		// x can NOT be used here	

4	Let Example 2 - No Redeclare	• Variables defined with let can not be redeclared. • You can not accidentally redeclare a variable declared with let. • Eg: let x = "John Bravo"; let x = 2;	
5	Let Example 3 - Let Hoisting	• Variables defined with let or const are also hoisted to the top of the block, but not initialized • Using a let or const variable before it is declared will result in a ReferenceError. • Eg: carModel = "Vivo La Rasa"; let carModel = "Viva";	
6	Var	• Variables declared with **var** have Global Scope. • Variables defined with var can be redeclared. • Variables declared with var are either function-scoped or globally scoped if declared outside any function. • If var is used inside a function, the variable is function-scoped. If var is used outside of any function, it declares a global variable • Variables declared with var are hoisted, which means they are moved to the top of their enclosing scope (function or global) before code execution. However, only the declaration is hoisted, not the initialization. Accessing the variable before the declaration results in undefined. • Within its scope, a variable declared with var can be re-declared without causing an error, which can lead to potential bugs if not carefully managed. • Redeclaring a JavaScript variable with var is allowed anywhere in a program.	
7	Var Example - Var Hoisting	• Variables defined with var are hoisted to the top and can be initialized at any time. • However, only the declaration is hoisted, not the initialization. If you try to use a variable before it is declared and initialized, its value will be undefined. • Eg: carModel = "Vivo La Rasa"; var carModel;	Even though the assignment appears before the declaration in the code, the variable carName is already declared (due to hoisting) by the time the assignment happens. Therefore, the code works without throwing a ReferenceErro and carName ends up being "Vivo La Rasa".

8	Const	• The const keyword was introduced in ES6 (2015).
		• Variables defined with const cannot be Redeclared
		• Variables defined with const cannot be Reassigned.
		• Variables defined with const have Block Scope.
		• JavaScript const variables must be assigned a value when they are declared.
			Eg:
			const PI = 3.14899; // This is Correct
			
			// This is Wrong
			const PI;
			PI = 31.4899;

9	When to use Const?	Always declare a variable with const when you know that the value should not be changed.	Use const when you declare: • A new Array • A new Object • A new Function • A new RegExp