**JavaScript Data Types**

JavaScript variables can hold many **data types**: numbers, strings, objects and more:

**Example:**

var length = 16; // Number

var lastName = "Johnson"; // String

var x = {firstName:"John", lastName:"Doe"}; // Object

## JavaScript Types are Dynamic

JavaScript has dynamic types. This means that the same variable can be used to hold different data types:

**Example:**

var x; // Now x is undefined

x = 5; // Now x is a Number

x = "John"; // Now x is a String

## JavaScript Strings

A string (or a text string) is a series of characters like "John Doe".

Strings are written with quotes. You can use single or double quotes:

**Example:**

var carName1 = "Volvo XC60"; // Using double quotes

var carName2 = 'Volvo XC60'; // Using single quotes

## JavaScript Numbers

JavaScript has only one type of numbers.

Numbers can be written with, or without decimals.

**Example:**

var x1 = 34.00; // Written with decimals

var x2 = 34; // Written without decimals

## JavaScript Booleans

Booleans can only have two values: true or false.

**Example:**

var x = 5;

var y = 5;

var z = 6;

(x == y) // Returns true

(x == z) // Returns false

## JavaScript Arrays

JavaScript arrays are written with square brackets.

Array items are separated by commas.

The following code declares (creates) an array called cars, containing three items (car names):

**Example:**

var cars = ["Audi", "Volvo", "BMW"];

## JavaScript Objects

JavaScript objects are written with curly braces {}.

Object properties are written as name:value pairs, separated by commas.

**Example:**

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

## The typeof Operator

You can use the JavaScript typeof operator to find the type of a Write the difference between var, let and const with code examples.JavaScript variable.

The typeof operator returns the type of a variable or an expression:

**Example:**

typeof "" // Returns "string"

typeof "John" // Returns "string"

typeof "John Doe" // Returns "string"

## Undefined

In JavaScript, a variable without a value, has the value undefined. The type is also undefined.

**Example:**

var car;    // Value is undefined, type is undefined

## Empty Values

An empty value has nothing to do with undefined.

An empty string has both a legal value and a type.

**Example:**

var car = "";    // The value is "", the typeof is "string"

## Null

In JavaScript null is "nothing". It is supposed to be something that doesn't exist.

Unfortunately, in JavaScript, the data type of null is an object.

**Example:**

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};  
person = null;    // Now value is null, but type is still an object