**Javascript – Data Types**

There are 3 types of Data Types in JS:

* String (for example “**Bob**”)
* Number (for example **1**)
* Boolean (**true** or **false**)

**Adding Different Types of Data**

When adding a number and a string, JavaScript will treat the number as a string:

let x = 16 + "Volvo"; **is equivalent to** let x = "Volvo" + 16;

**🡺** 16Volvo

JavaScript evaluates expressions from left to right. Different sequences can produce different results:

let x = 16 + 4 + "Volvo"; 🡺 20Volvo

let x = "Volvo" + 16 + 4; 🡺 Volvo164

In the first example, JavaScript treats 16 and 4 as numbers, until it reaches "Volvo".

In the second example, since the first operand is a string, all operands are treated as strings.

**JavaScript Types are Dynamic**

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

let x;           // Now x is undefined  
x = 5;           // Now x is a Number  
x = "John";      // Now x is a String

**Numbers**

Numbers can be written with, or without decimals:

let x1 = 34.00;     // Written with decimals  
let x2 = 34;        // Written without decimals

Extra large or extra small numbers can be written with scientific (exponential) notation:

let y = 123e5;      // 12300000  
let z = 123e-5;     // 0.00123

**Boolean**

Booleans can only have two values: **true** or**false**.

let x = 5;  
let y = 5;  
let z = 6;  
(x == y)       // Returns true  
(x == z)       // Returns false

**Strings**

The **String** object is used to represent and manipulate a **sequence of characters**.

Strings are **written with quotes**. You can use single or double quotes, they have exactly the same effect:

let carName1 = "Volvo XC60";   // Using double quotes  
let carName2 = 'Volvo XC60';   // Using single quotes

You can use quotes inside a string, as long as they don't match the quotes surrounding the string:

let answer1 = "It's alright";             // Single quote inside double quotes

let answer2 = "He is called 'Johnny'";    // Single quotes inside double quotes

let answer3 = 'He is called "Johnny"';    // Double quotes inside single quotes

**Useful String Methods**

**Length Method**

The **length** property returns the length of a string.

The **length** property of an empty string is 0.

**Sintax:** *string*.length

|  |  |
| --- | --- |
| **Code** | **Result** |
| let text = "Hello World!"; let length = text.length; | 12 |

**Slice Method**

The **slice()** method returns selected elements in an array, as a new array.

The **slice()** method selects from a given start, up to a (not inclusive) given end.

The **slice()** method does not change the original array.

**Sintax:** *array*.slice(start, end)

|  |  |
| --- | --- |
| **Code** | **Result** |
| const fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"]; const citrus = fruits.slice(1, 3); |  |
| const fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"]; const myBest = fruits.slice(-3, -1); |  |