JavaScript Data Types

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

## **The Concept of Data Types**

In programming, data types is an important concept.

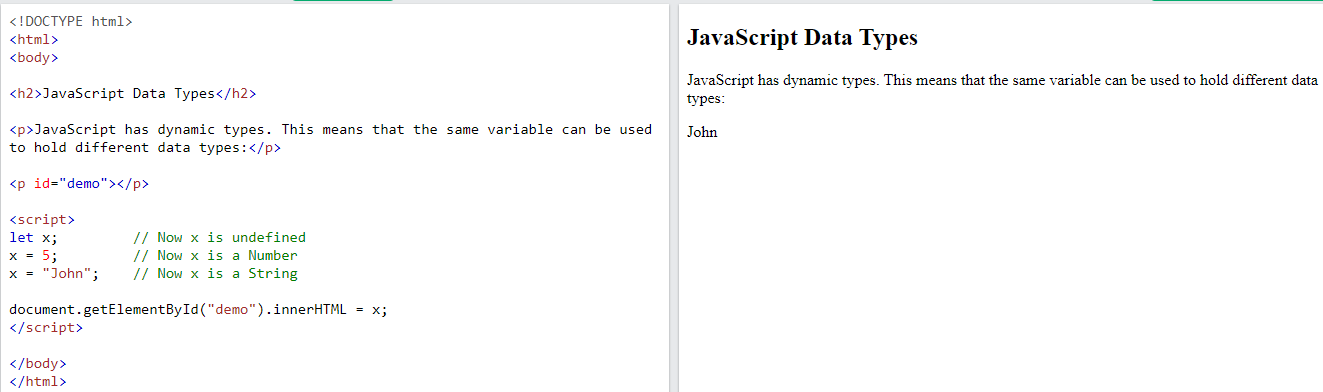
To be able to operate on variables, it is important to know something about the type.

Without data types, a computer cannot safely solve this:

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

## **JavaScript Types are Dynamic**

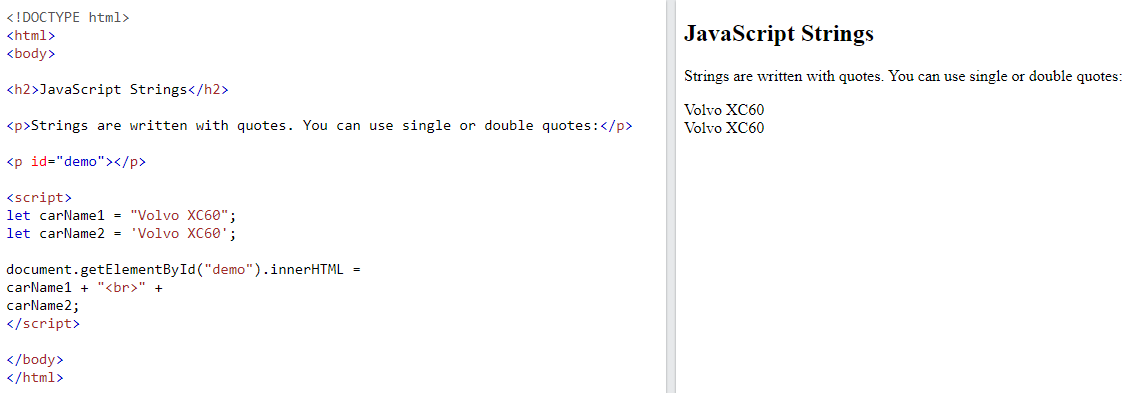
JavaScript has dynamic types. This means that the same variable can be used to hold different data types:



## **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:



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

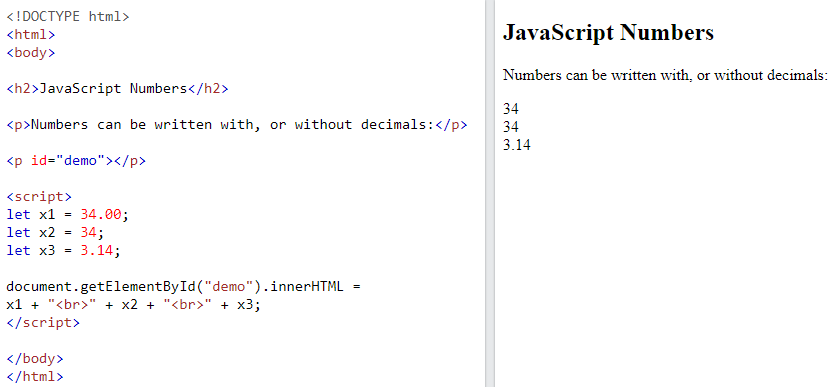
### **Example**

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

## **JavaScript Numbers**

JavaScript has only one type of numbers.

Numbers can be written with, or without decimals:



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

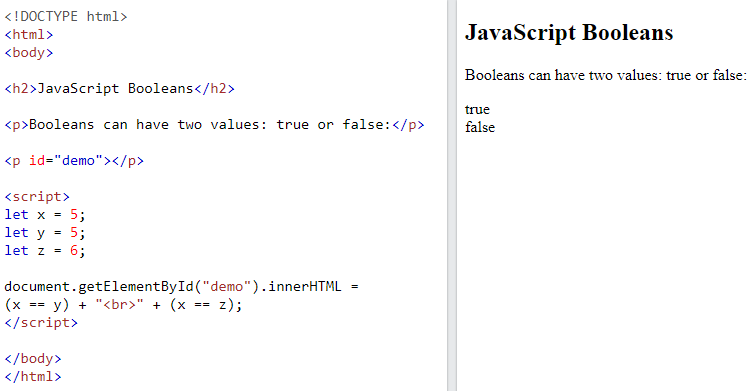
### **Example**

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

## **JavaScript Booleans**

Booleans can only have two values: true or false.

Booleans are often used in conditional testing.



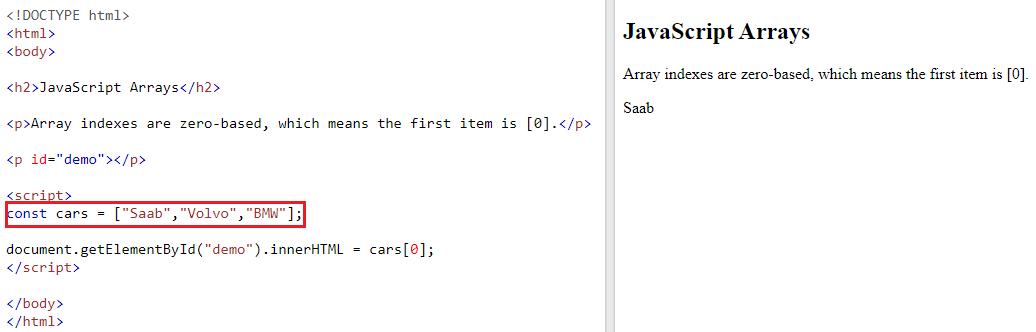
## **JavaScript Arrays**

JavaScript arrays are written with square brackets.

Array items are separated by commas.

Array indexes are zero-based, which means the first item is [0], second is [1], and so on.

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



## **JavaScript Objects**

JavaScript objects are written with curly braces {}.

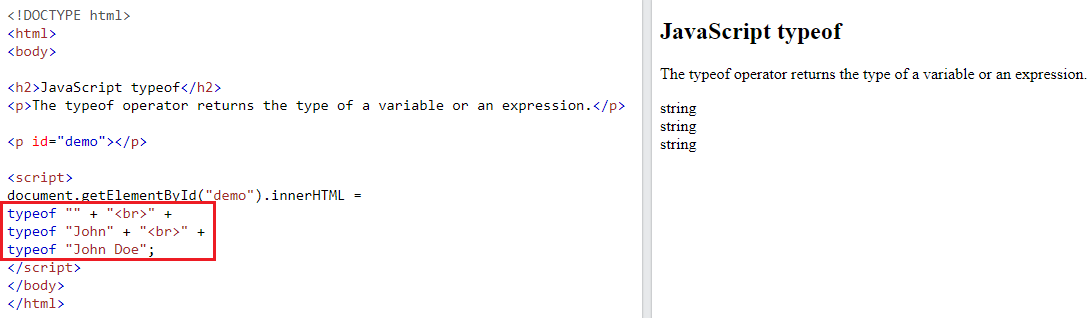
Object properties are written as name:value pairs, separated by commas.



## **The typeof Operator**

You can use the JavaScript typeof operator to find the type of a JavaScript variable.

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

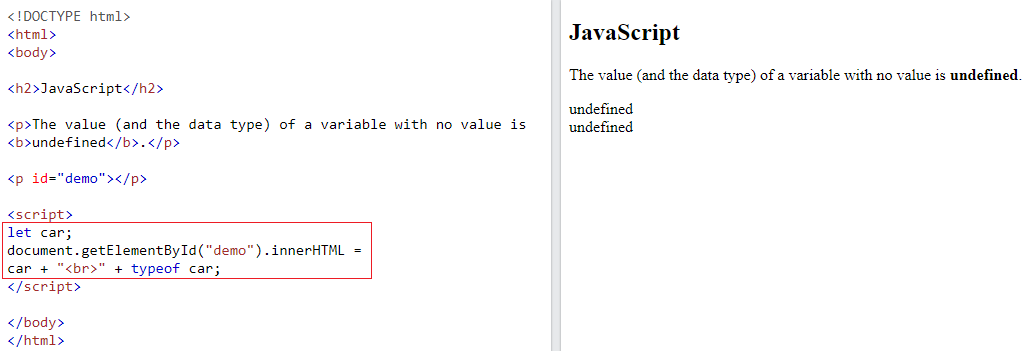


### **Example**

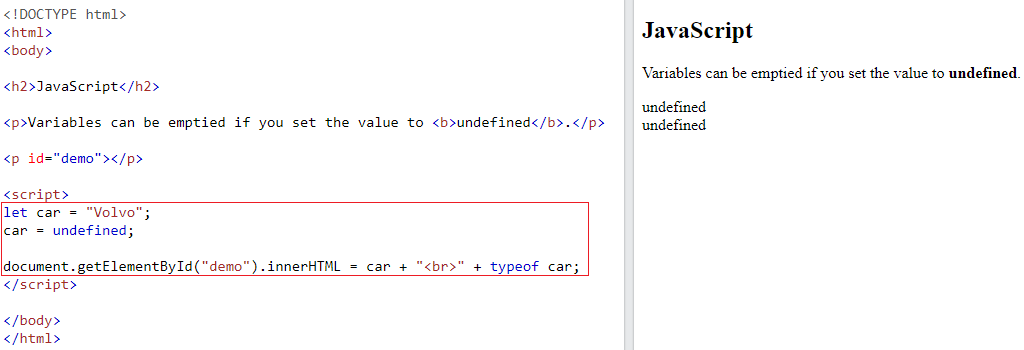
typeof 0              // Returns "number"  
typeof 314            // Returns "number"  
typeof 3.14           // Returns "number"  
typeof (3)            // Returns "number"  
typeof (3 + 4)        // Returns "number"

## **Undefined**

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



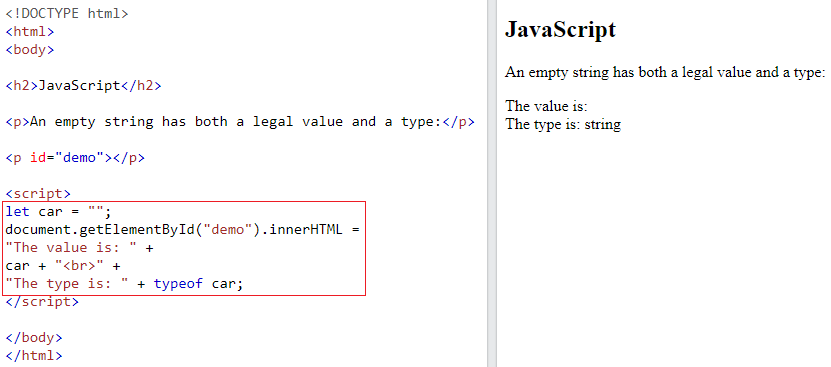
Any variable can be emptied, by setting the value to undefined. The type will also be undefined.



## **Empty Values**

An empty value has nothing to do with undefined.

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



References

1. https://www.w3schools.com/js/js\_datatypes.asp