JAVASCRIPT TOPIC 7

InfoTech 37
Jomar B. Colao
College Instructor

Intro

String, Number, Boolean, Array, Object, Null, Undefined.

JavaScript Has Dynamic Types

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

Example

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

JavaScript Strings

A string is a variable which stores a series of characters like "John Doe". Strings are written with quotes. You can use single or double quotes:

Example

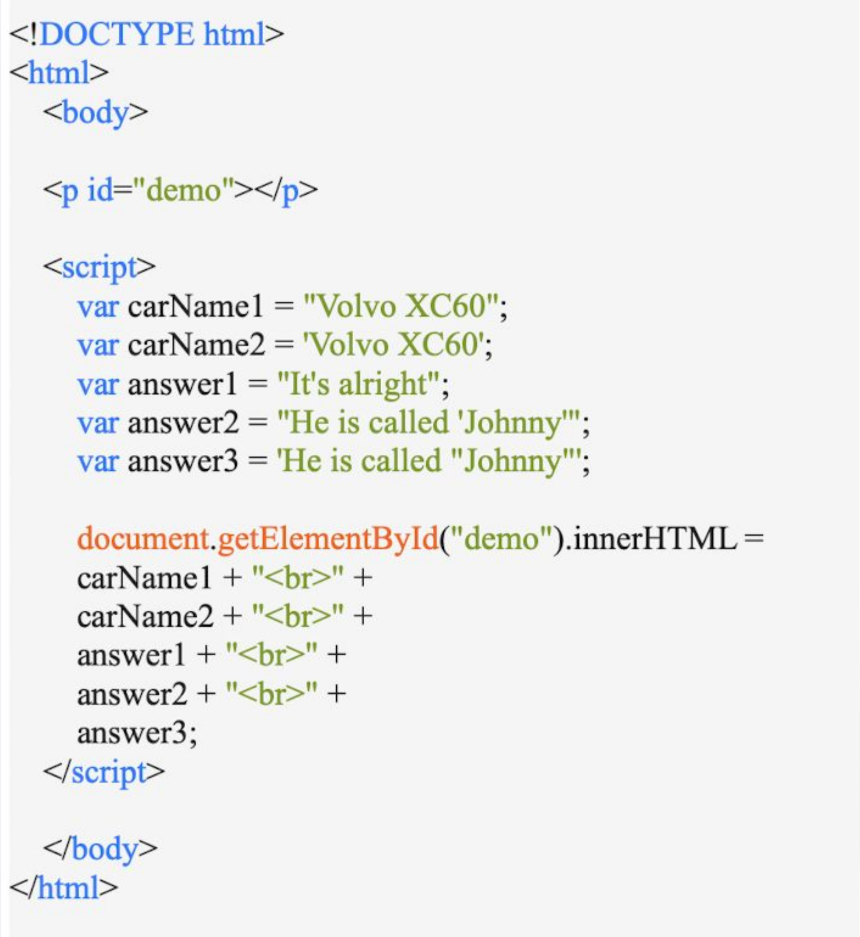
```
var carName = "Volvo XC60";   // Using double quotes
var carName = 'Volvo XC60';   // Using single quotes
```

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

Example

```
var answer = "It's alright";          // Single quote inside double quotes
var answer = "He is called 'Johnny'"; // Single quotes inside double quotes
var answer = 'He is called "Johnny"'; // Double quotes inside single quotes
```

Demo 1



```
<!DOCTYPE html>
<html>
  <body>

  <p id="demo"></p>

  <script>
    var carName1 = "Volvo XC60";
    var carName2 = 'Volvo XC60';
    var answer1 = "It's alright";
    var answer2 = "He is called 'Johnny'";
    var answer3 = 'He is called "Johnny"';

    document.getElementById("demo").innerHTML =
    carName1 + "<br>" +
    carName2 + "<br>" +
    answer1 + "<br>" +
    answer2 + "<br>" +
    answer3;
  </script>

  </body>
</html>
```

JavaScript Numbers

JavaScript has only one type of numbers.

Numbers can be written with, or without decimals:

Example

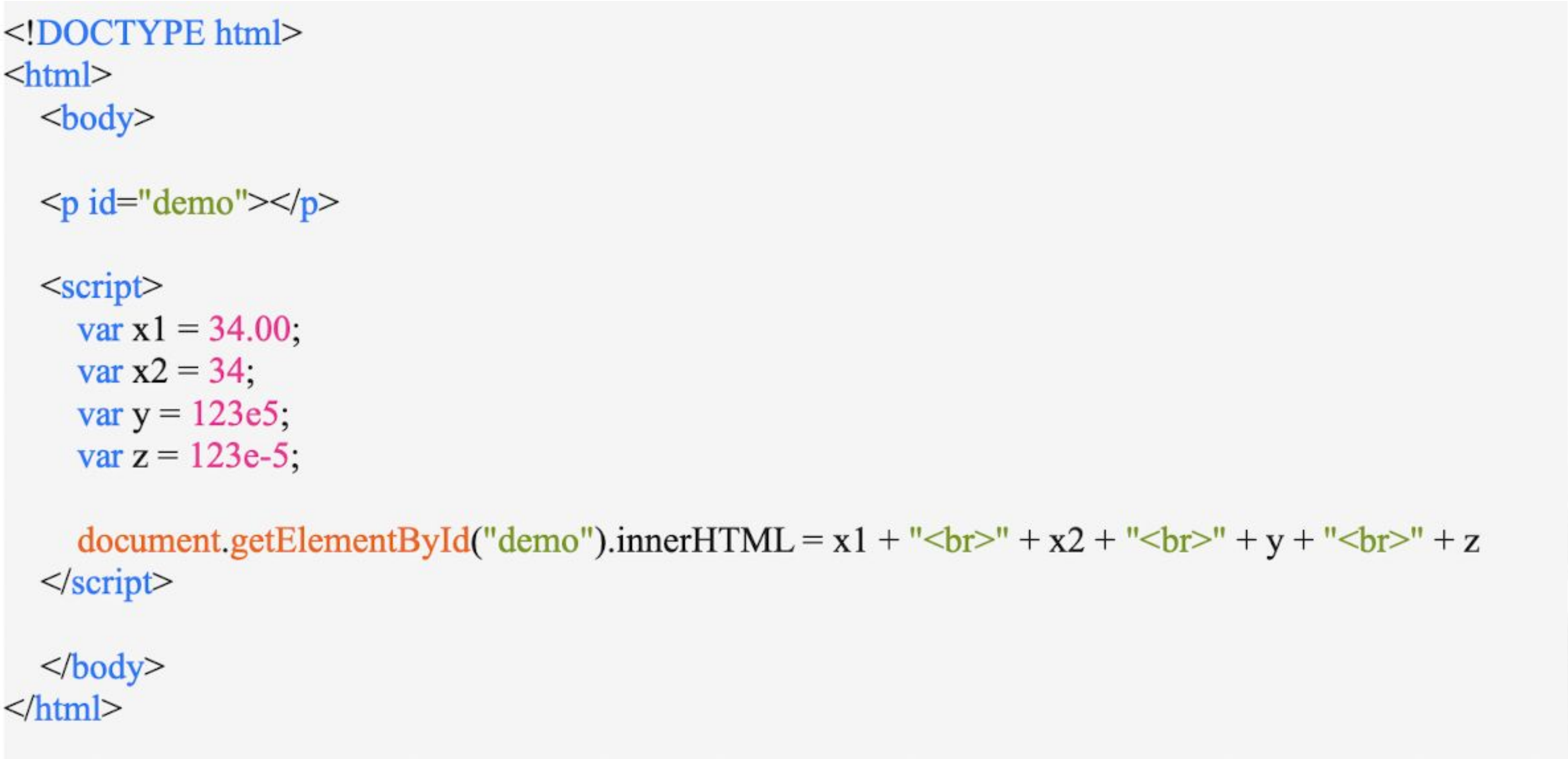
```
var x1 = 34.00;         // Written with decimals
var x2 = 34;      // Written without decimals
```

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

Example

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

Demo 2



```
<!DOCTYPE html>
<html>
  <body>

  <p id="demo"></p>

  <script>
    var x1 = 34.00;
    var x2 = 34;
    var y = 123e5;
    var z = 123e-5;

    document.getElementById("demo").innerHTML = x1 + "<br>" + x2 + "<br>" + y + "<br>" + z
  </script>

  </body>
</html>
```

JavaScript Booleans

Booleans can only have two values: true or false.

```
var x = true;
var y = false;
```

Booleans are often used in conditional testing.

JavaScript Arrays

JavaScript arrays are written with square brackets.

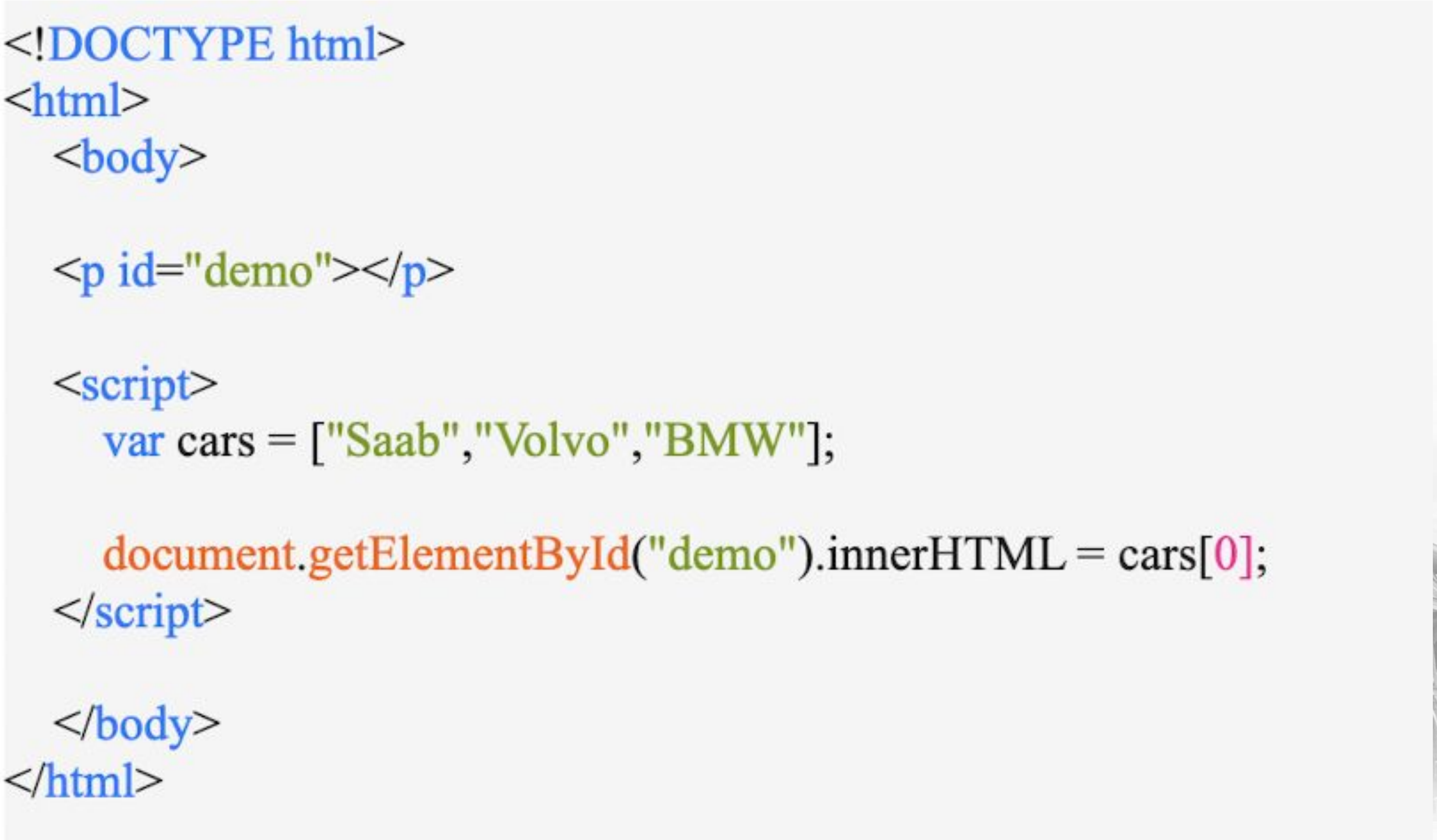
Array items are separated by commas.

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

Example

```
var cars = ["Saab", "Volvo", "BMW"];
```

Demo 3



```
<!DOCTYPE html>
<html>
  <body>

  <p id="demo"></p>

  <script>
    var cars = ["Saab","Volvo","BMW"];

    document.getElementById("demo").innerHTML = cars[0];
  </script>

  </body>
</html>
```

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"};
```

One Statement, Many Variables

Wrong:

```
var lastName = "Doe"; age = 30; job = "carpenter";
```

Right:

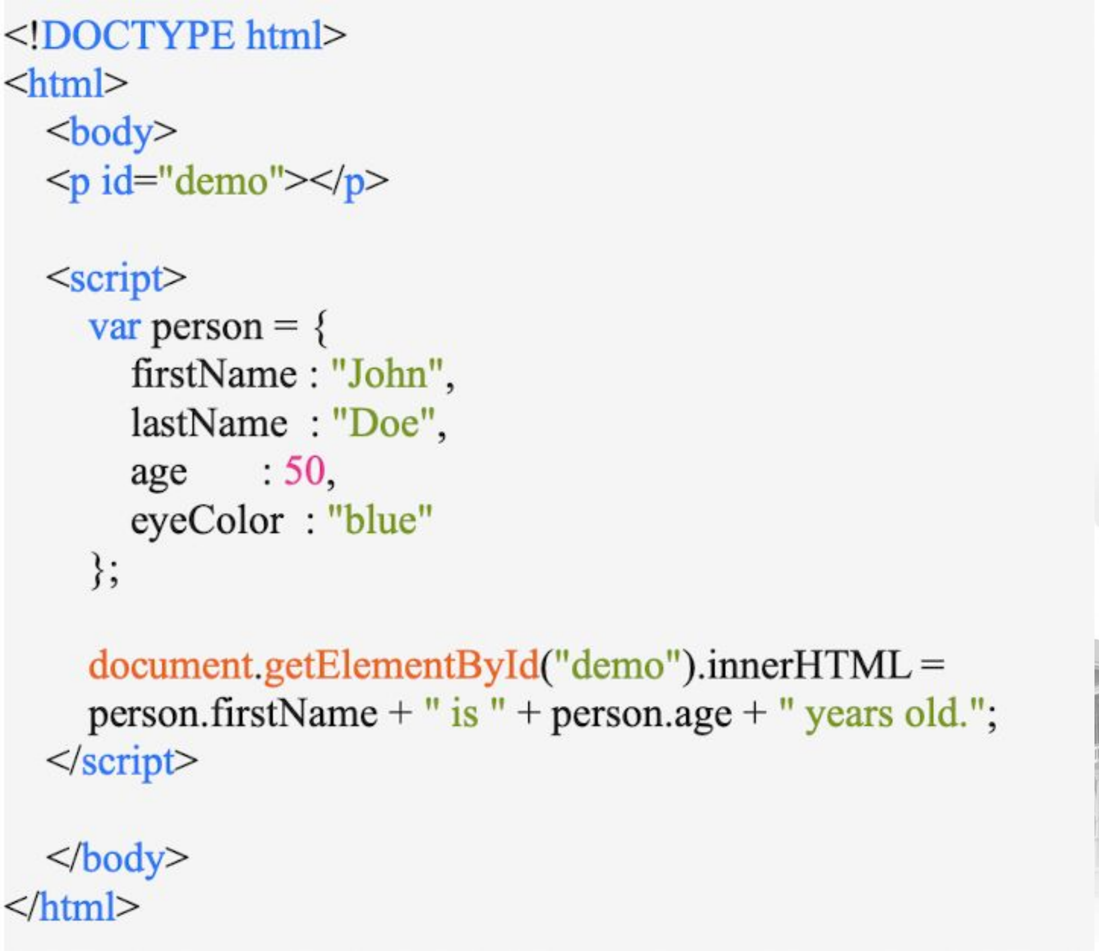
```
var lastName = "Doe"; var age = 30; var job = "carpenter";
```

Value = undefined

- ❑ In computer programs, variables are often declared without a value. The value can be something that has to be calculated, or something that will be provided later, like user input. Variable declared without a value will have the value undefined.
- ❑ The variable carName will have the value undefined after the execution of the following statement:

```
var carName;
```

Demo 4



```
<!DOCTYPE html>
<html>
  <body>
  <p id="demo"></p>

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

    document.getElementById("demo").innerHTML =
    person.firstName + " is " + person.age + " years old.";
  </script>

  </body>
</html>
```

Undefined and Null

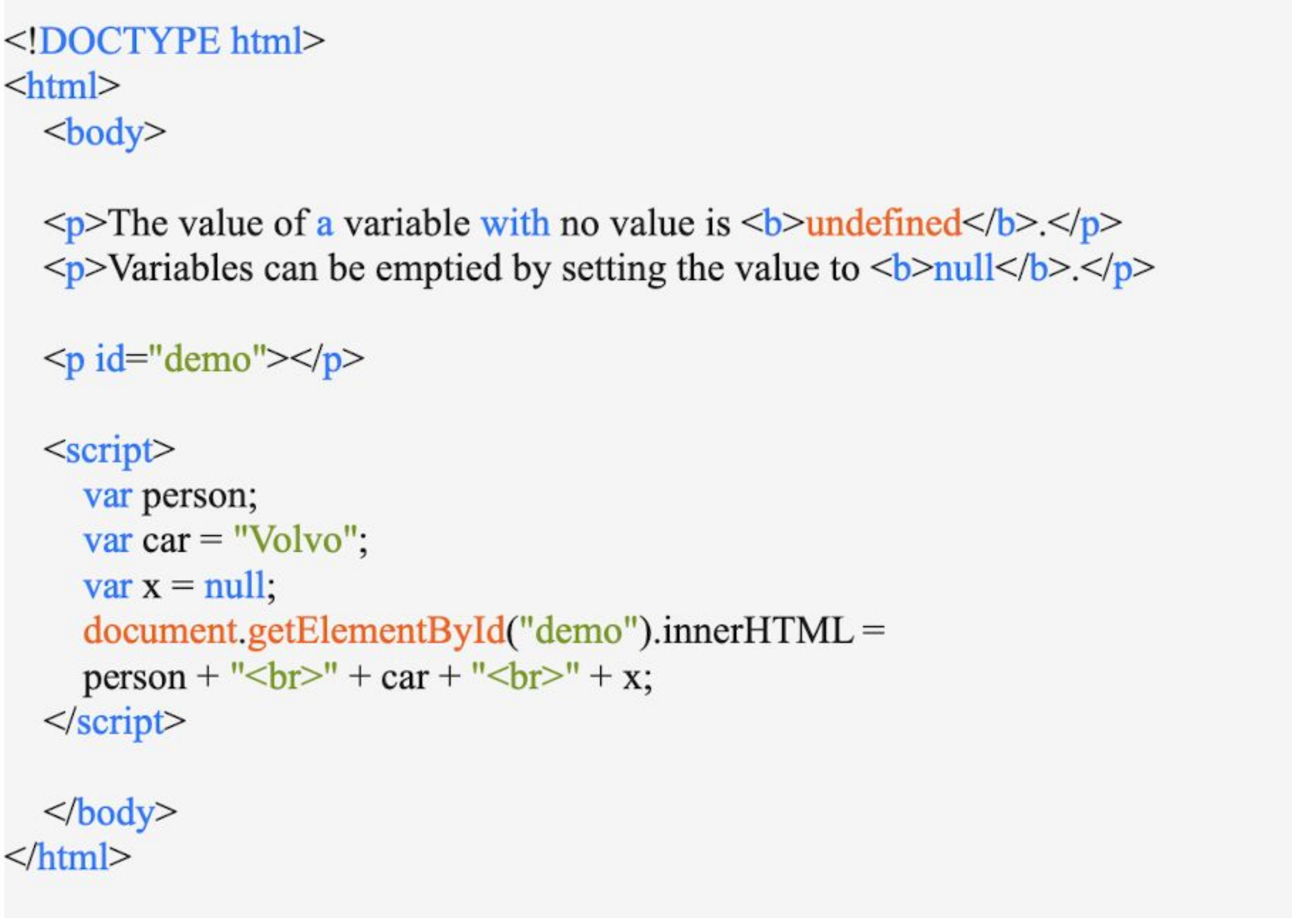
The value of a variable with no value is undefined.

Variables can be emptied by setting the value to null.

Example

```
var cars;            // Value is undefined
var person = null; // Value is null
```

Demo 5



```
<!DOCTYPE html>
<html>
  <body>

  <p>The value of a variable with no value is <b>undefined</b>.</p>
  <p>Variables can be emptied by setting the value to <b>null</b>.</p>

  <p id="demo"></p>

  <script>
    var person;
    var car = "Volvo";
    var x = null;
    document.getElementById("demo").innerHTML =
    person + "<br>" + car + "<br>" + x;
  </script>

  </body>
</html>
```

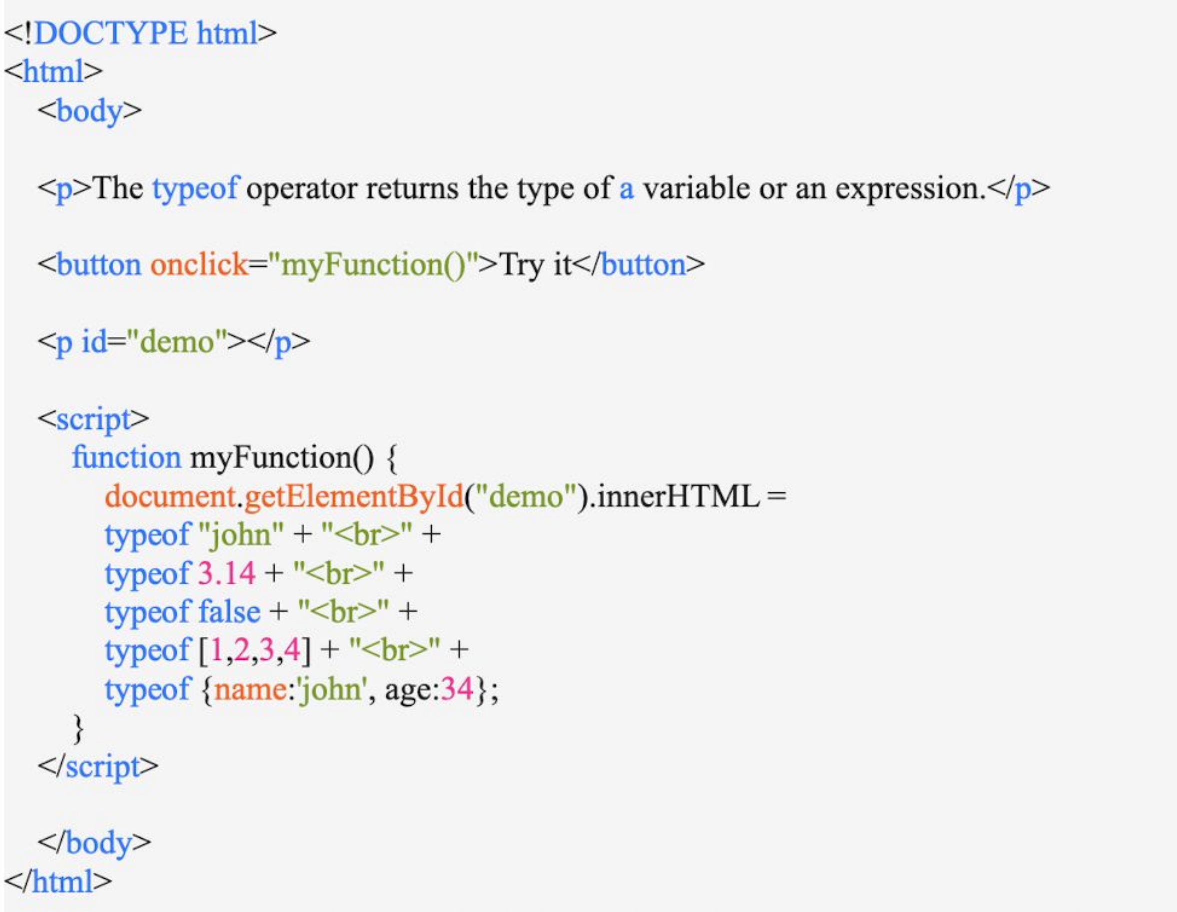
The typeof Operator

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

Example

```
typeof "John"                   // Returns string
typeof 3.14               // Returns number
typeof false              // Returns boolean
typeof [1,2,3,4]                // Returns object
typeof {name:'John', age:34}  // Returns object
```

Demo 6



```
<!DOCTYPE html>
<html>
  <body>

  <p>The typeof operator returns the type of a variable or an expression.</p>

  <button onclick="myFunction()">Try it</button>

  <p id="demo"></p>

  <script>
     function myFunction() {
        document.getElementById("demo").innerHTML =
        typeof "john" + "<br>" +
        typeof 3.14 + "<br>" +
        typeof false + "<br>" +
        typeof [1,2,3,4] + "<br>" +
        typeof {name:'john', age:34};
     }
  </script>

  </body>
</html>
```