Introdução - JavaScript

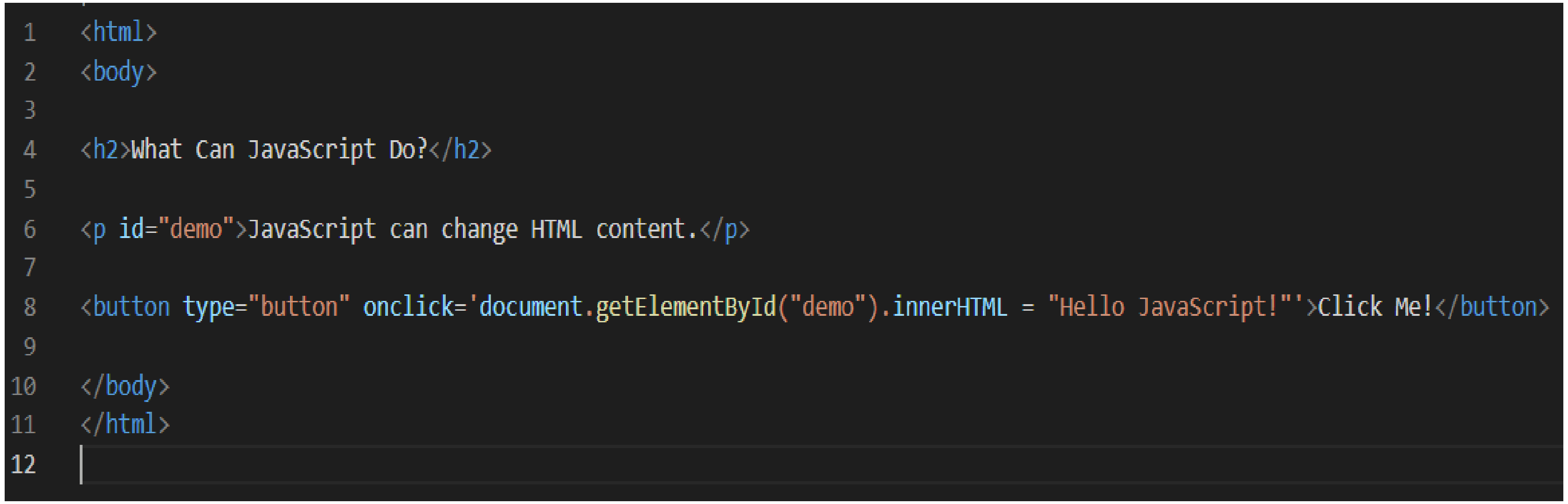
Origem JavaScript

- A linguagem de programação JavaScript foi inventada em 1995 por Brendan Eich.
- Atualmente o JavaScript está na versão: ECMAScript 2018
- Para mais informações: https://www.w3schools.com/js/js_versions.asp

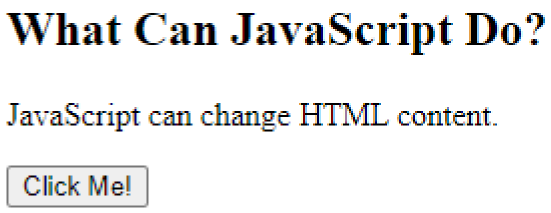
Introdução

- Um dos métodos mais utilizados do JavaScript é o getElementById()

- Exemplo:

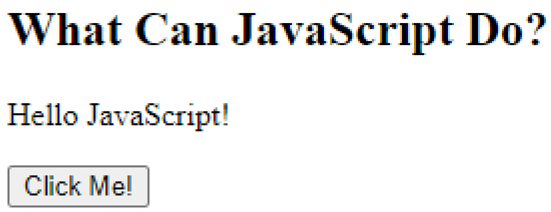


```
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button>

</body>
</html>

```



What Can JavaScript Do?

JavaScript can change HTML content.

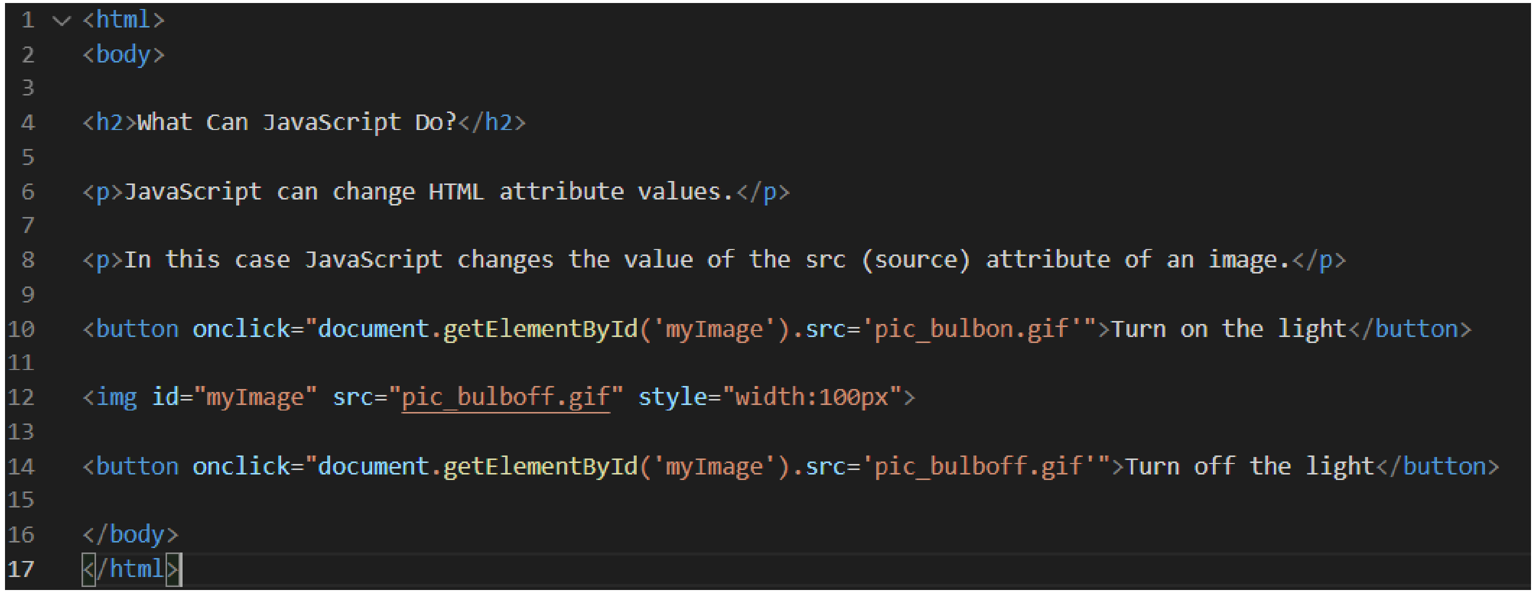


What Can JavaScript Do?

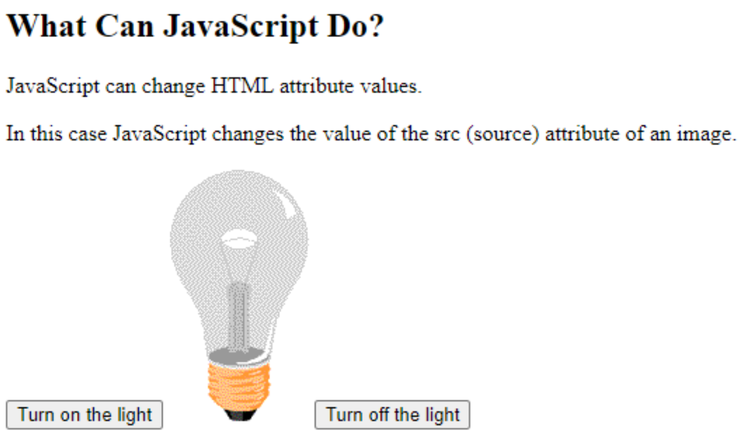
Hello JavaScript!

*É possível utilizar aspas duplas ou aspas simples.

Mudando valores “src” do atributo <img>



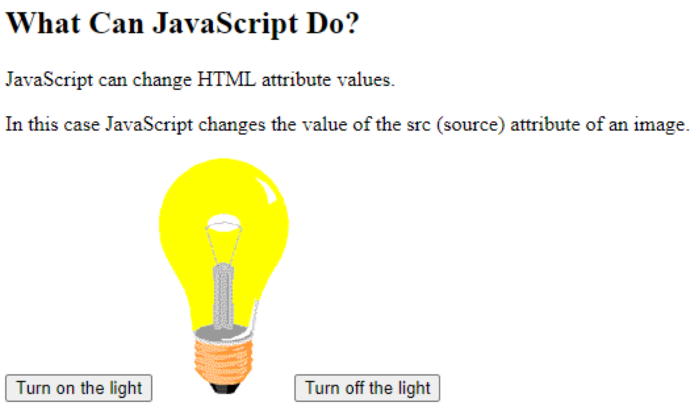
```
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can change HTML attribute values.</p>

<p>In this case JavaScript changes the value of the src (source) attribute of an image.</p>

<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the light</button>

<img id="myImage" src="pic_bulboff.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the light</button>

</body>
</html>
```



What Can JavaScript Do?

JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.

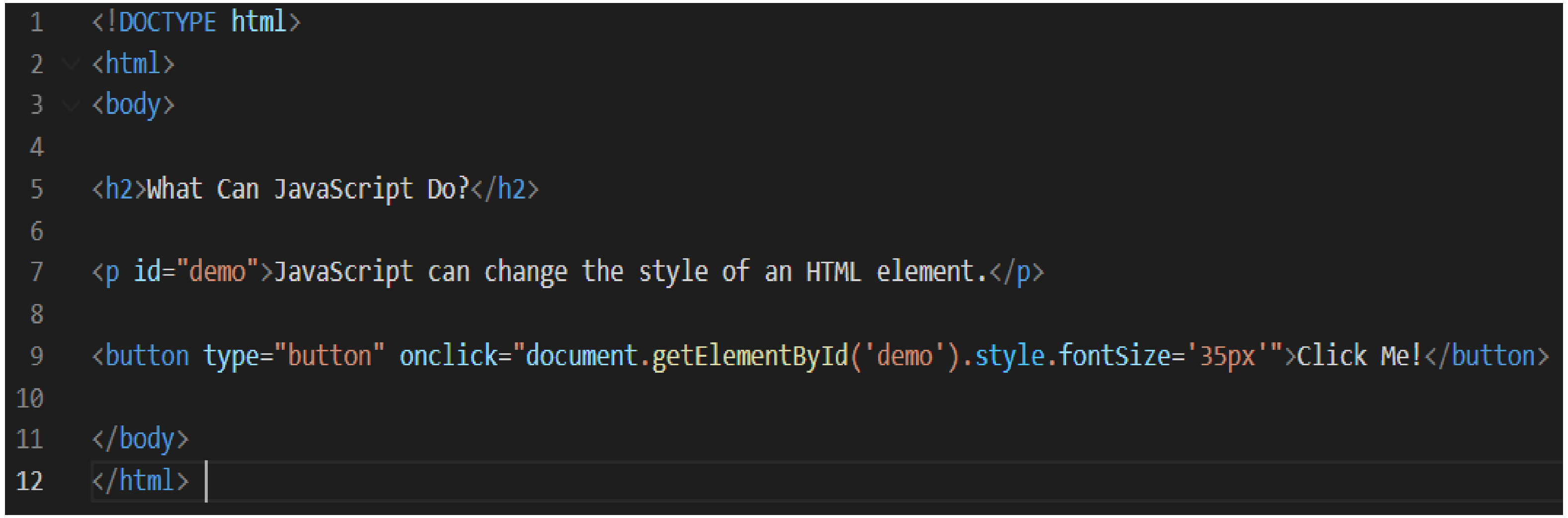


What Can JavaScript Do?

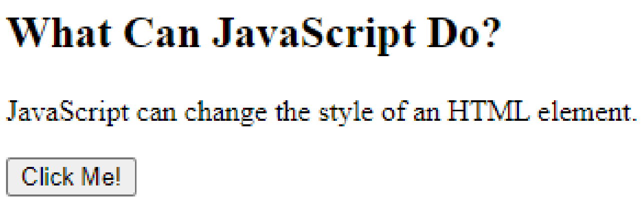
JavaScript can change HTML attribute values.

In this case JavaScript changes the value of the src (source) attribute of an image.

Com o JavaScript é possível mudar estilos (css)

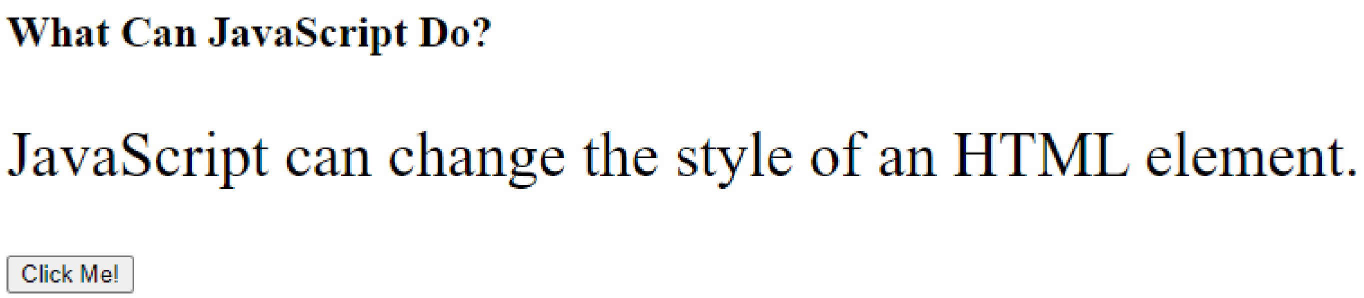


```html
<!DOCTYPE html>
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button" onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>

</body>
</html>
```



What Can JavaScript Do?

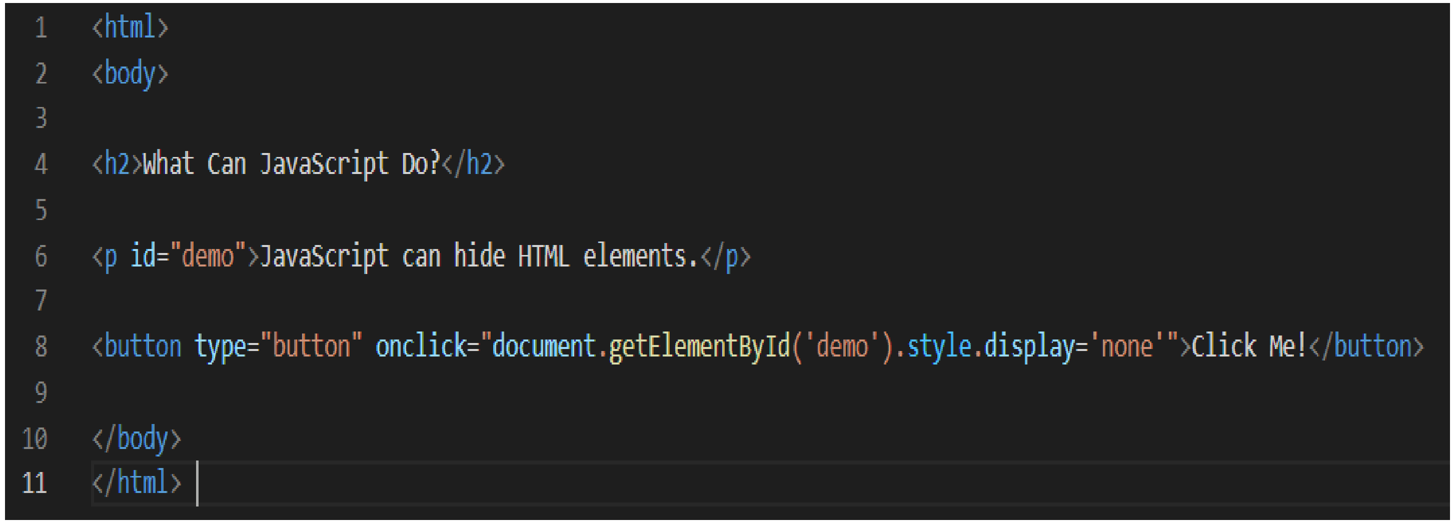
JavaScript can change the style of an HTML element.



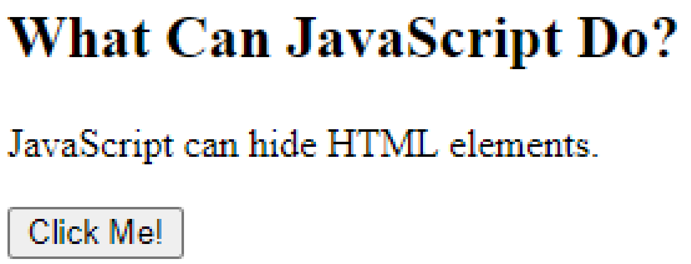
What Can JavaScript Do?

JavaScript can change the style of an HTML element.

Com o JavaScript é possível mudar estilos (css)

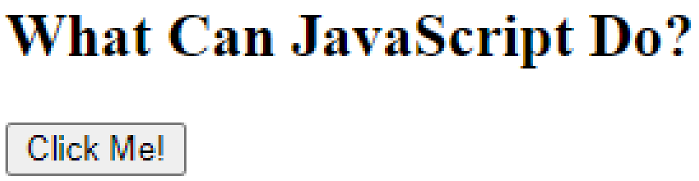


```html
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p id="demo">JavaScript can hide HTML elements.</p>

<button type="button" onclick="document.getElementById('demo').style.display='none'">Click Me!</button>

</body>
</html>
```



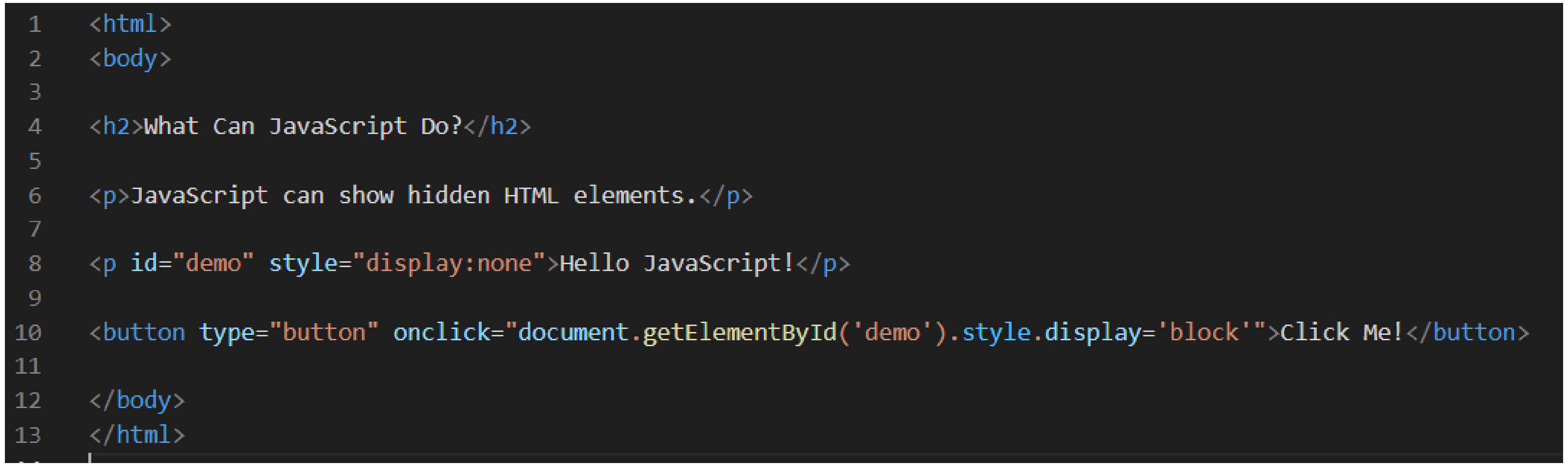
What Can JavaScript Do?

JavaScript can hide HTML elements.

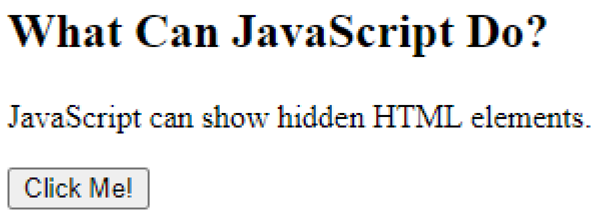


What Can JavaScript Do?

Com o JavaScript é possível exibir os componentes



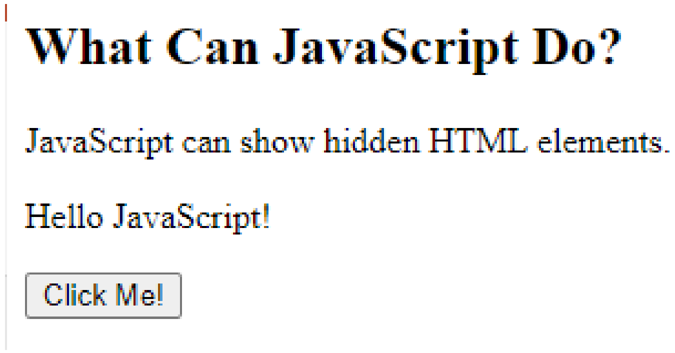
```
<html>
<body>

<h2>What Can JavaScript Do?</h2>

<p>JavaScript can show hidden HTML elements.</p>

<p id="demo" style="display:none">Hello JavaScript!</p>

<button type="button" onclick="document.getElementById('demo').style.display='block'">Click Me!</button>

</body>
</html>
```



What Can JavaScript Do?

JavaScript can show hidden HTML elements.

Click Me!



What Can JavaScript Do?

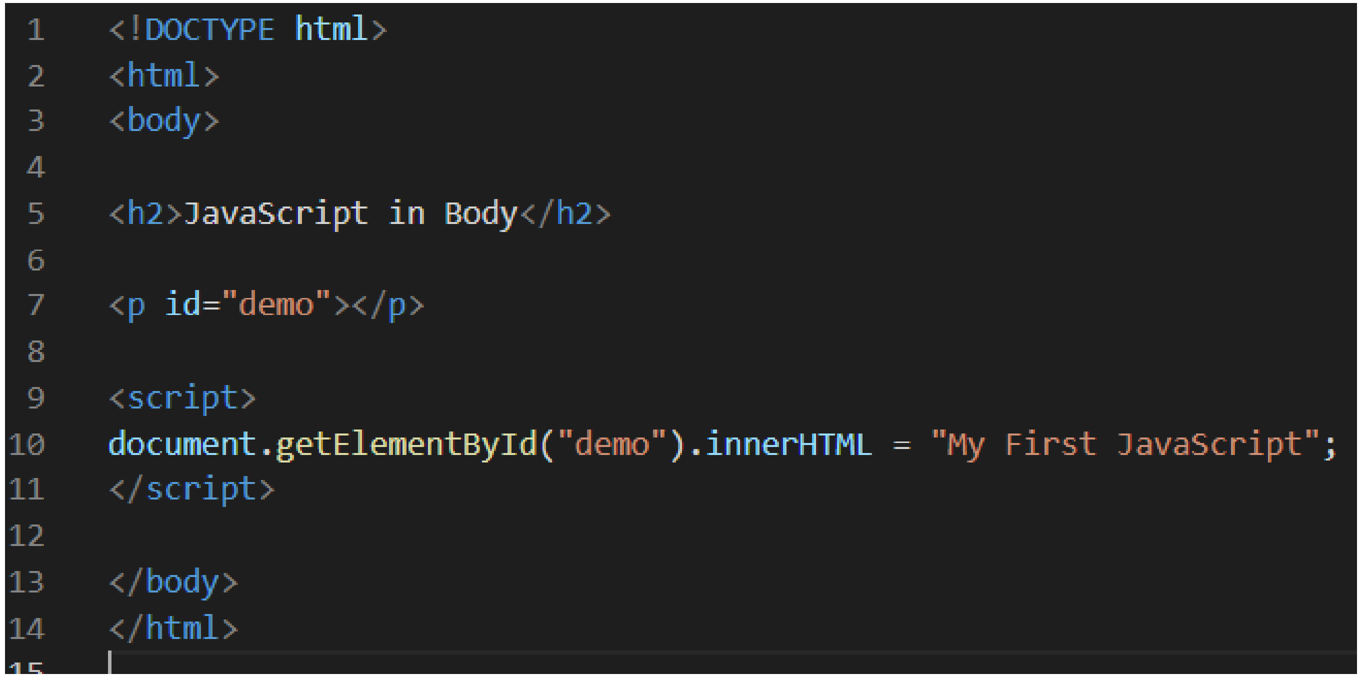
JavaScript can show hidden HTML elements.

Hello JavaScript!

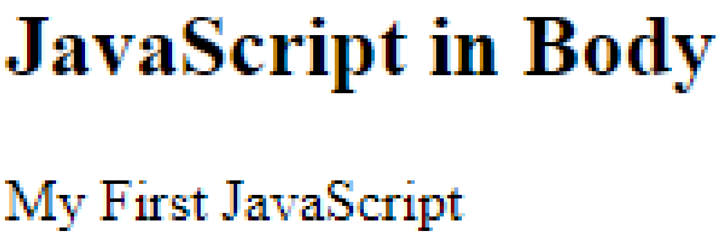
Click Me!

Onde/Como o JavaScript é inserido nas páginas?

- No HTML, o código JavaScript é inserido entre as tags <script> e </script>.



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

<h2>JavaScript in Body</h2>

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

<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>

</body>
</html>
```

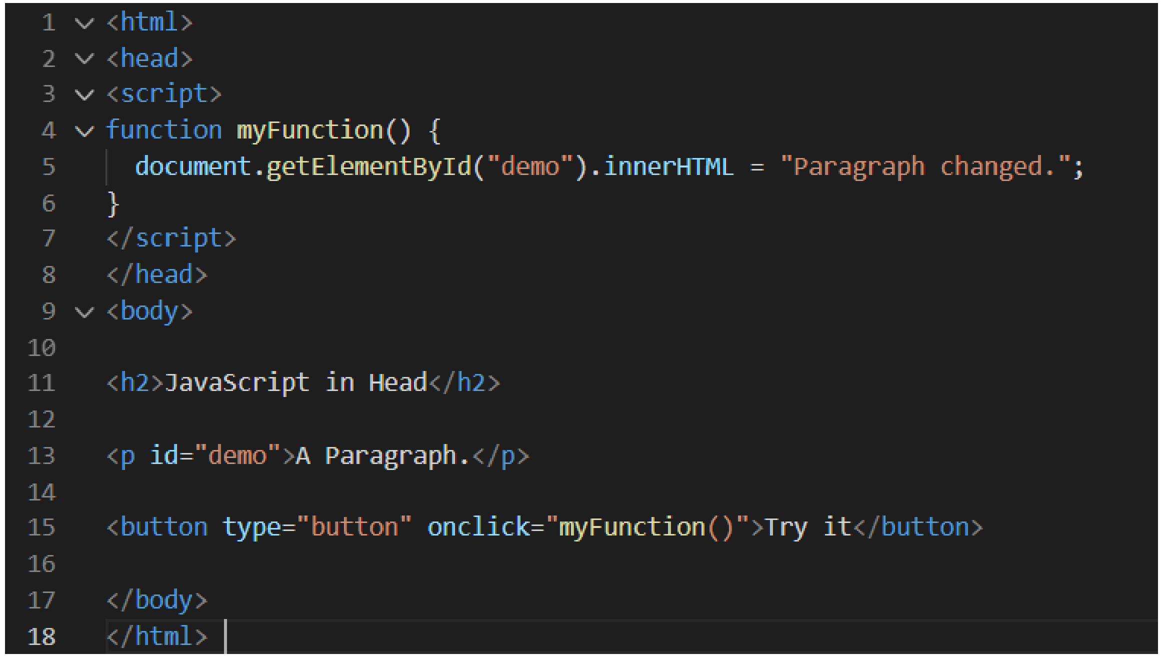


JavaScript in Body

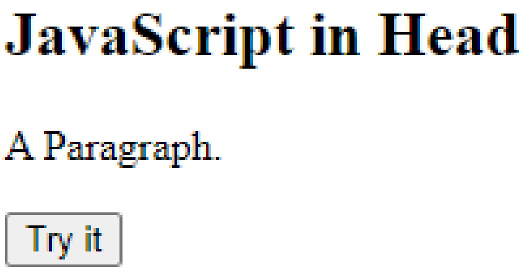
My First JavaScript

Funções e Eventos JavaScript

- Uma função JavaScript é um bloco de código que pode ser executado/chamado, por exemplo ao clicar em um botão posso acionar uma função.
- Geralmente o bloco JavaScript é declarado na sessão <head> da estrutura HTML (a interpretação do script é feita ao carregar a pagina).

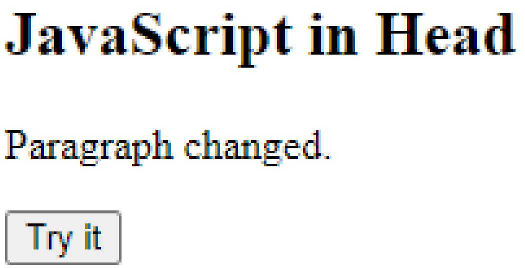


```html
<html>
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h2>JavaScript in Head</h2>

<p id="demo">A Paragraph.</p>

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

</body>
</html>
```



JavaScript in Head

A Paragraph.

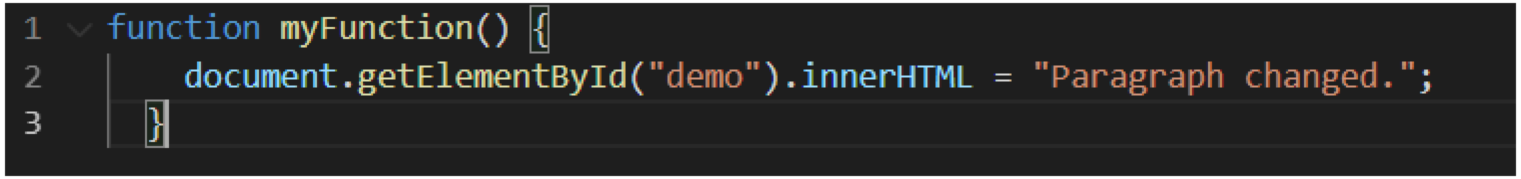


JavaScript in Head

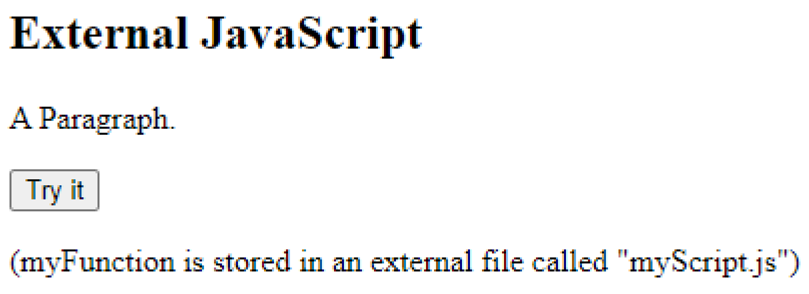
Paragraph changed.

External JavaScript

- myScript.js
- Exemplo8.html



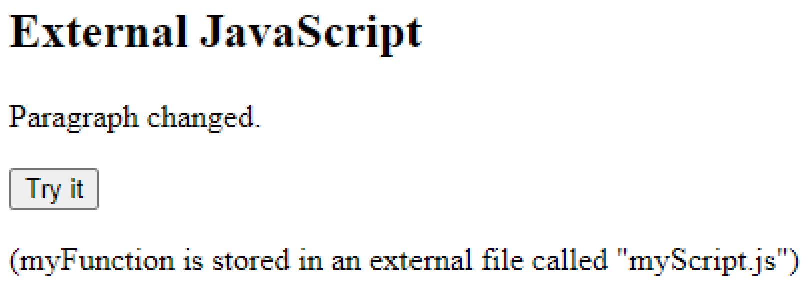
```
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```



External JavaScript

A Paragraph.

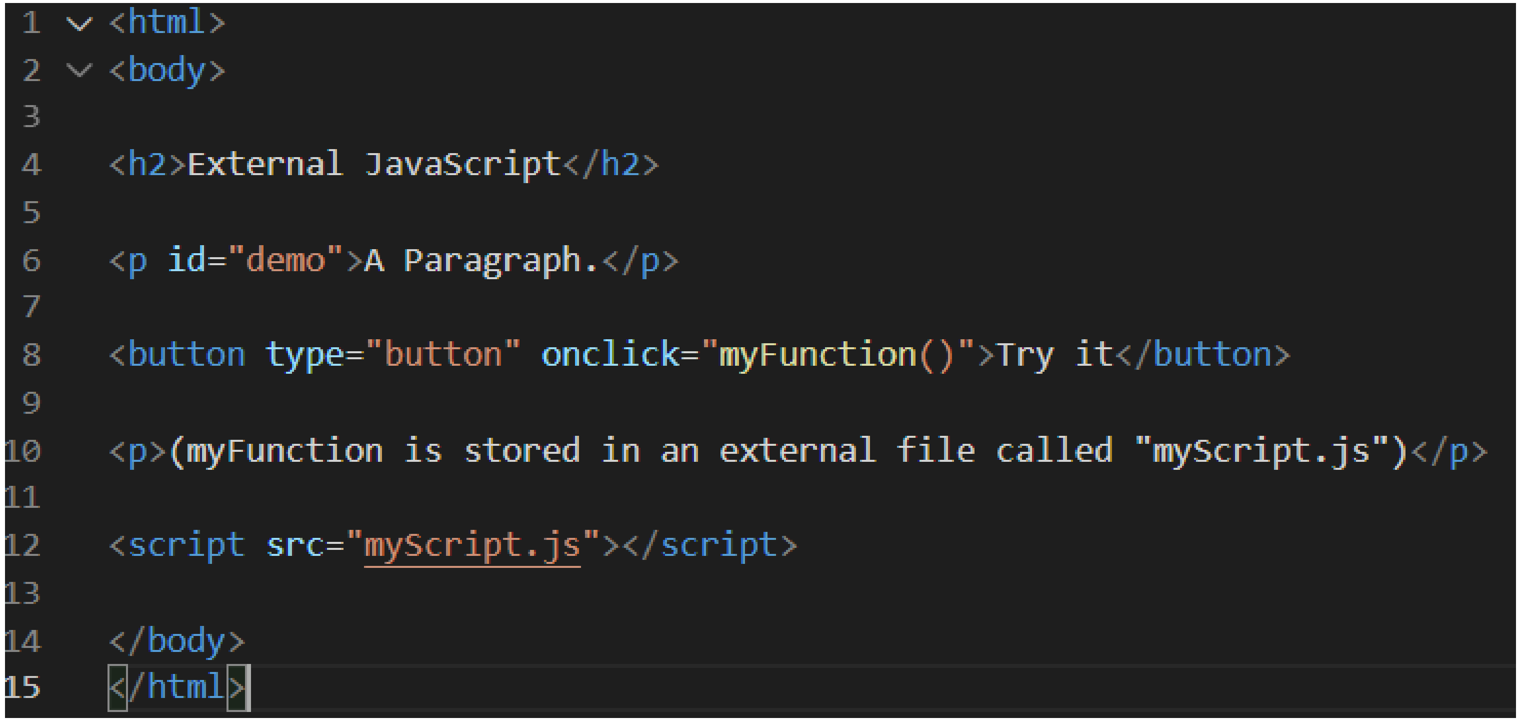
(myFunction is stored in an external file called "myScript.js")



External JavaScript

Paragraph changed.

(myFunction is stored in an external file called "myScript.js")

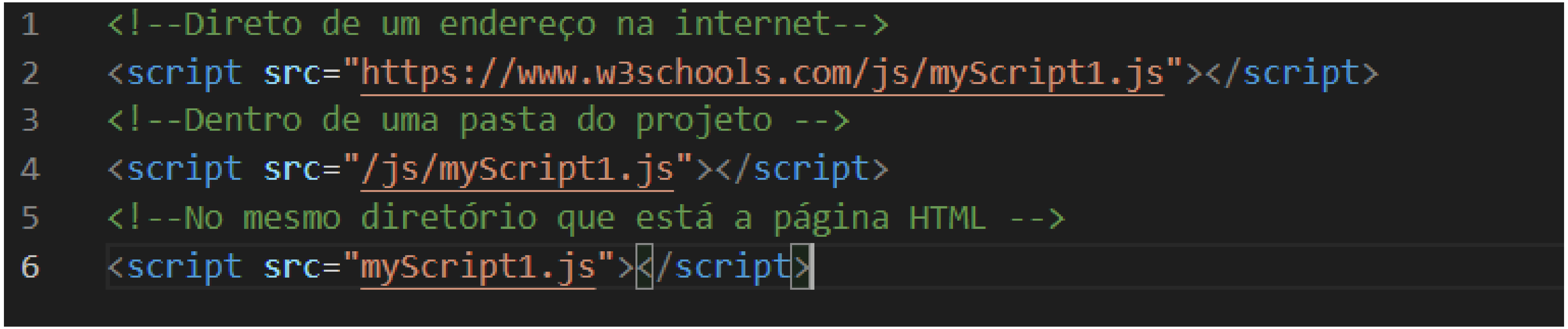


```
<html>
<body>

<h2>External JavaScript</h2>

<p id="demo">A Paragraph.</p>

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

<p>(myFunction is stored in an external file called "myScript.js")</p>

<script src="myScript.js"></script>

</body>
</html>
```

Vantagens sobre a externalização dos scripts

- O código HTML fica separado do código JavaScript
- Mais fácil para leitura e manutenção
- Os arquivos *.js são colocados em Cache pelo navegador, o que pode gerar melhor performance para carregar as páginas.

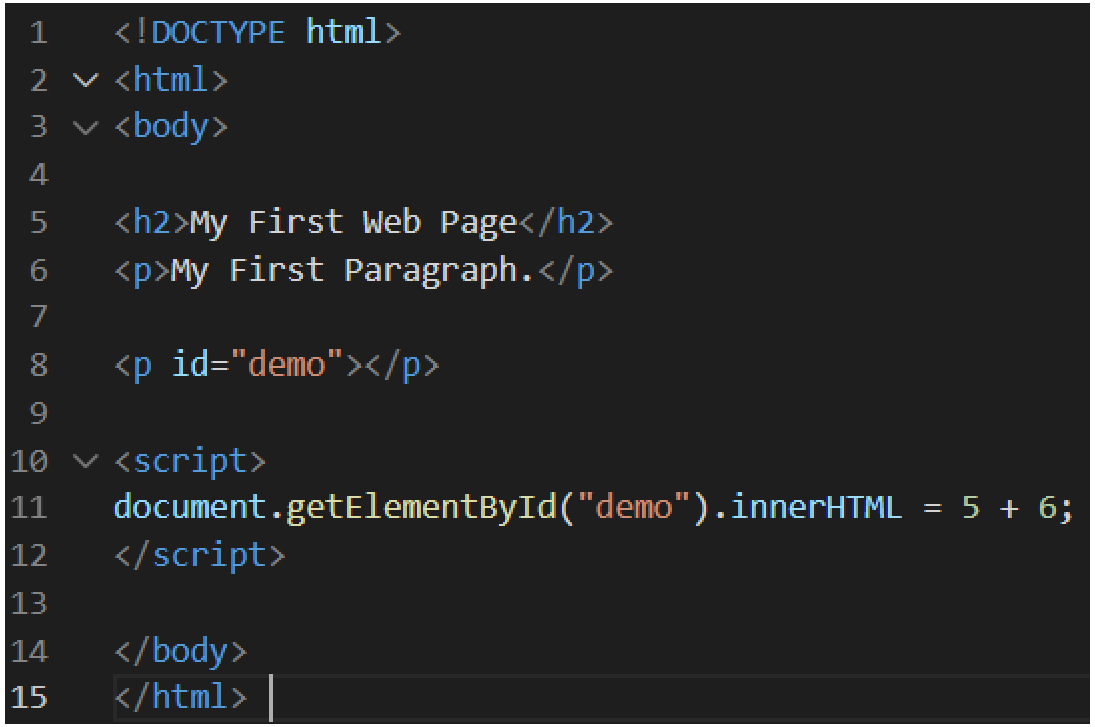
Referências Externas



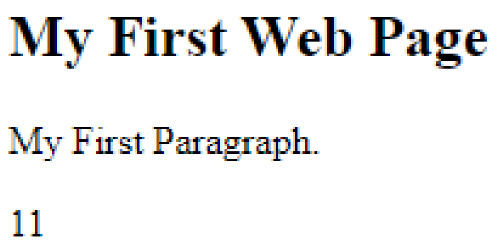
```html
<!--Direto de um endereço na internet-->
<script src="https://www.w3schools.com/js/myScript1.js"></script>
<!--Dentro de uma pasta do projeto -->
<script src="/js/myScript1.js"></script>
<!--No mesmo diretório que está a página HTML -->
<script src="myScript1.js"></script>
```

Saídas do JavaScript

- innerHTML



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

<h2>My First Web Page</h2>
<p>My First Paragraph.</p>

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

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```

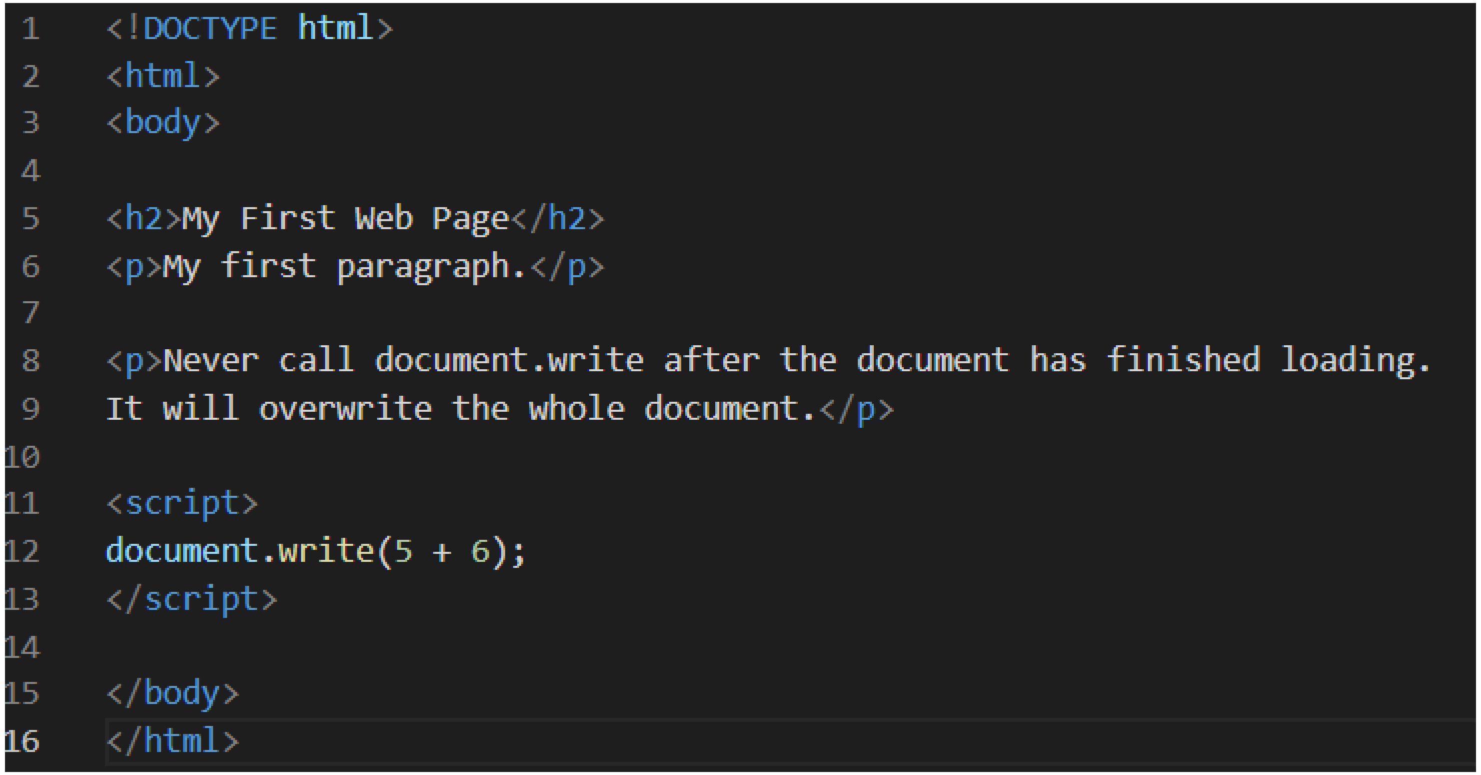


My First Web Page

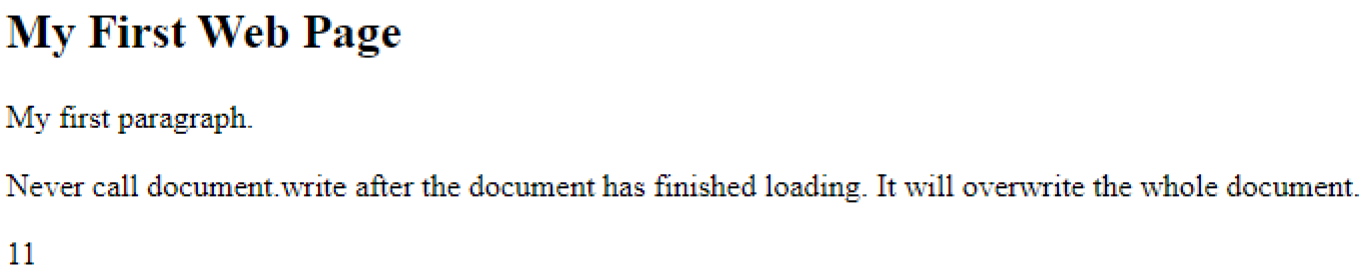
My First Paragraph.

11

- document.write()

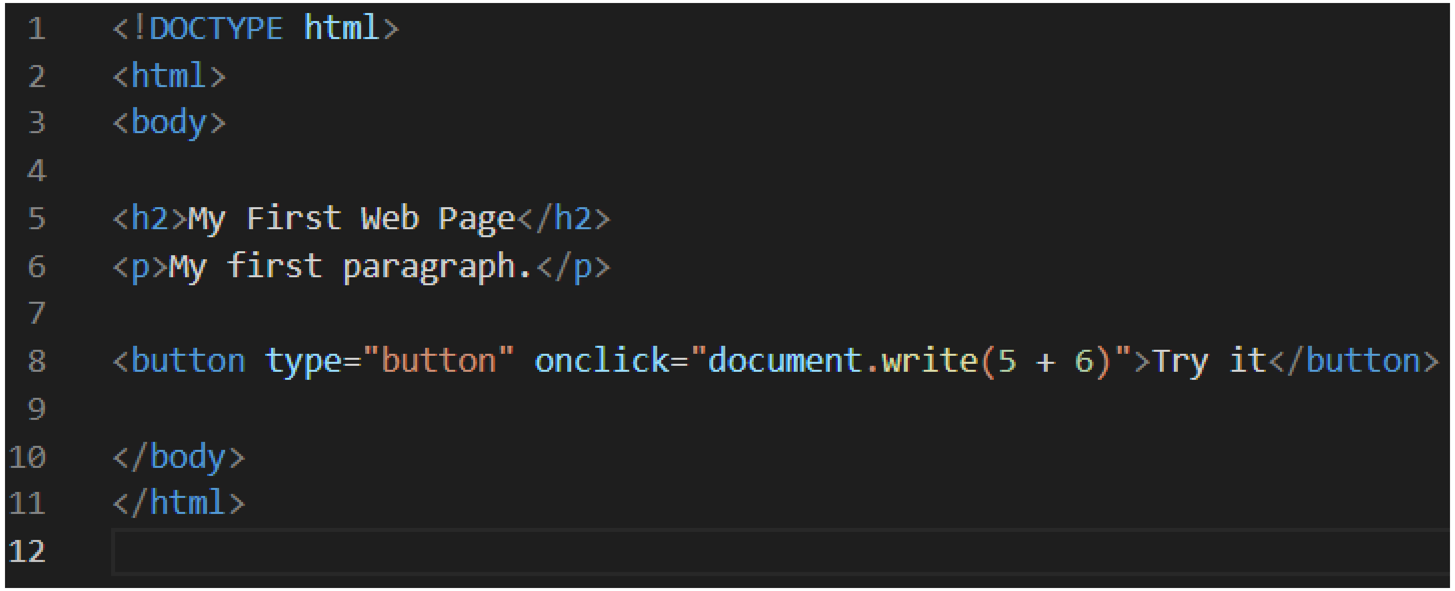


```
<!DOCTYPE html>
<html>
<body>

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<p>Never call document.write after the document has finished loading.
It will overwrite the whole document.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

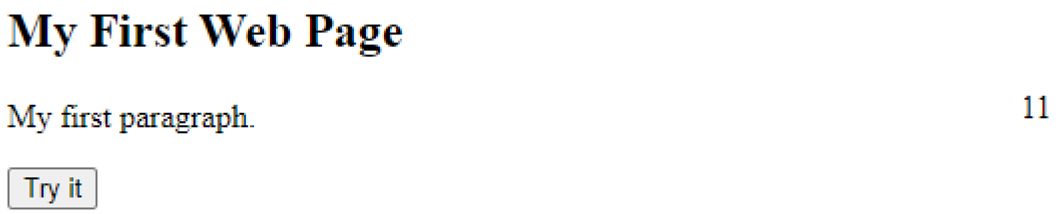


```
My First Web Page

My first paragraph.

Never call document.write after the document has finished loading. It will overwrite the whole document.

11
```

- Deleta todo o conteúdo HTML:



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

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">Try it</button>

</body>
</html>

```

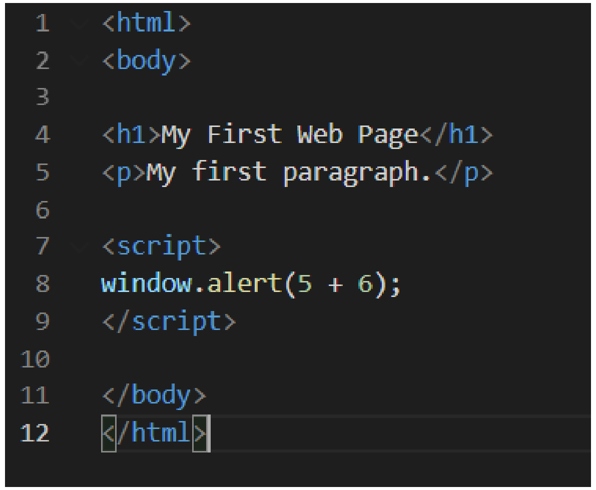


My First Web Page

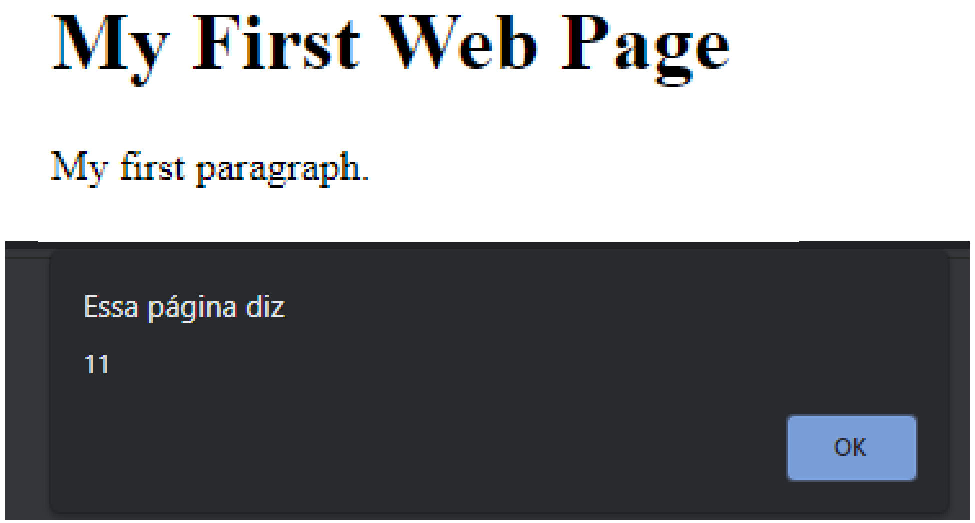
My first paragraph.

11

- window.alert()



```
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```

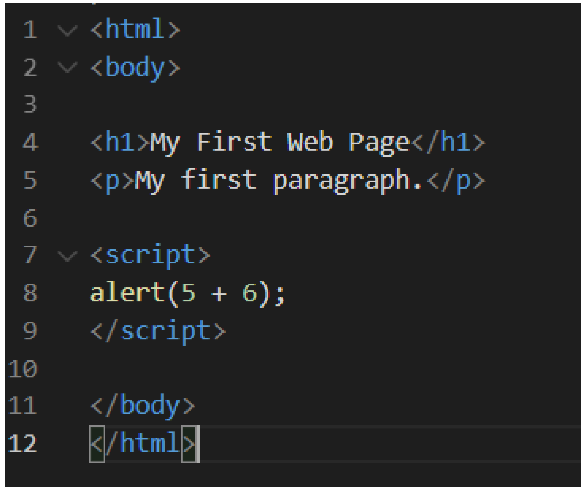


My First Web Page

My first paragraph.

Essa página diz

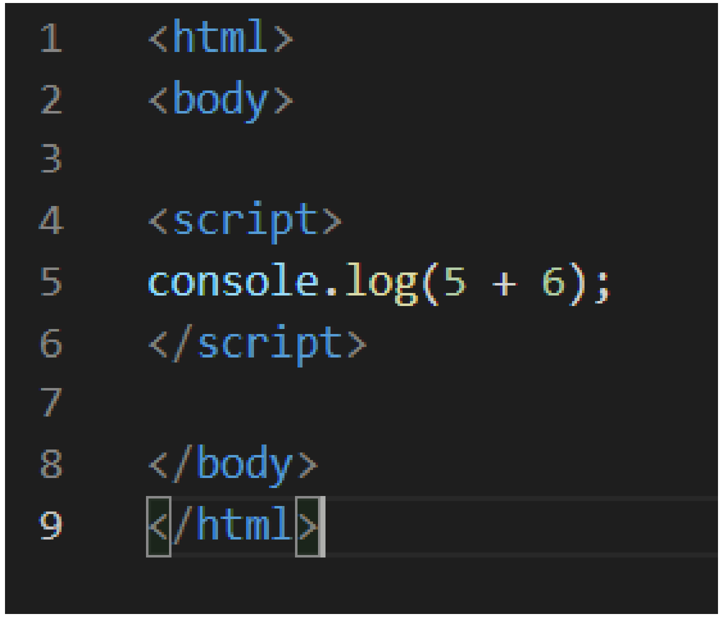
11



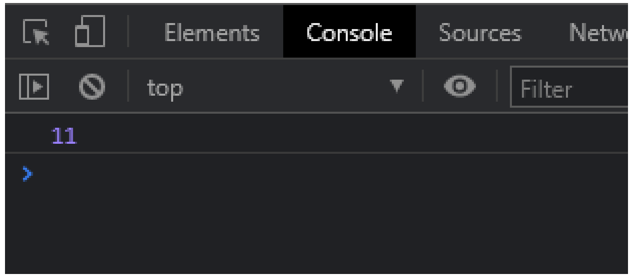
```
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
alert(5 + 6);
</script>

</body>
</html>
```

* Como o objeto “window” é o escopo global, não é necessário especificar.

- console.log()



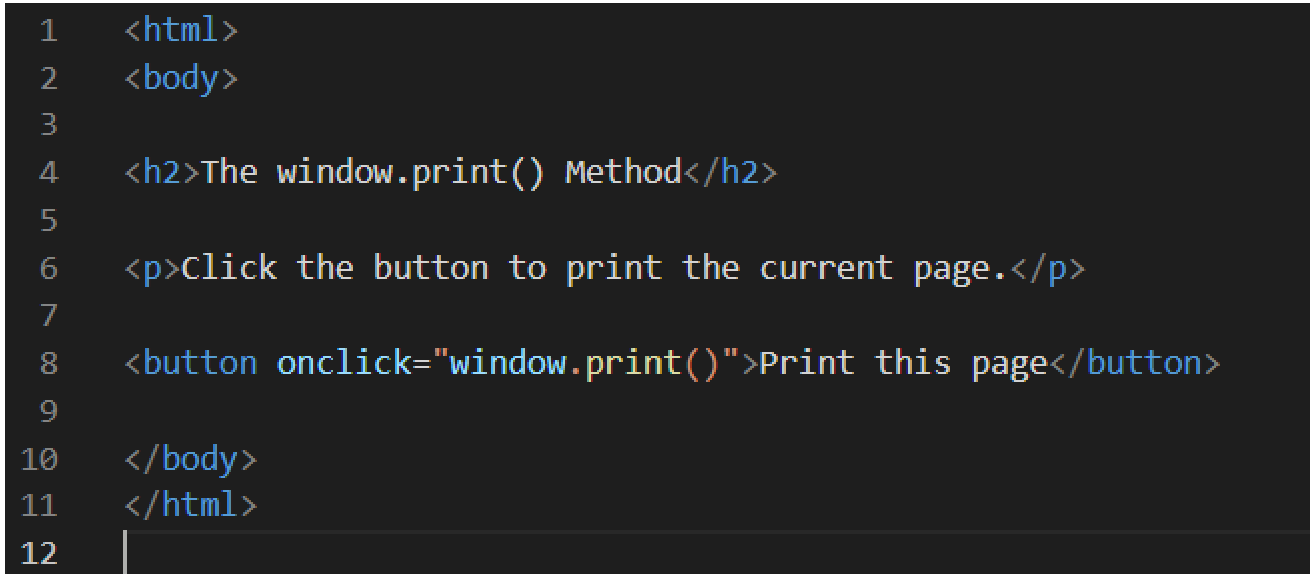
```
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```



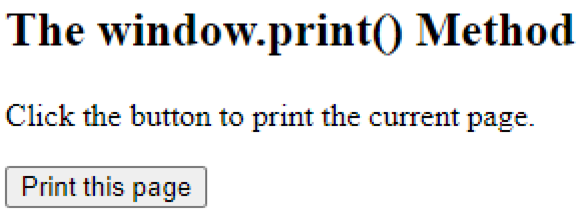
```
11
>
```

* Para que seja possível visualizar o console, é necessário
Utilizar a função F12 ou clicar com o botão direito Inspecionar.

JavaScript Print

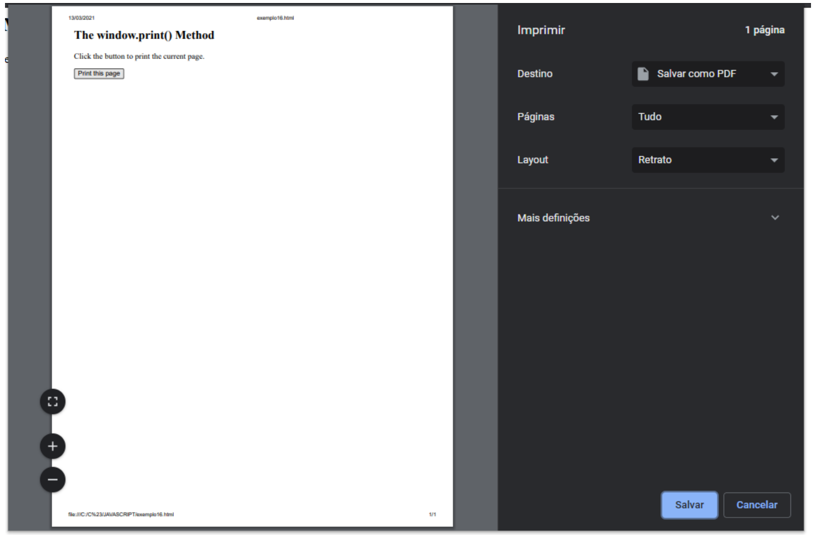


```html
<html>
<body>

<h2>The window.print() Method</h2>

<p>Click the button to print the current page.</p>

<button onclick="window.print()">Print this page</button>

</body>
</html>

```



The window.print() Method

Click the button to print the current page.



13/03/2021 exemplo16.html

The window.print() Method

Click the button to print the current page.

file:///C:/C%23/JAVASCRIPT/exemplo16.html 1/1

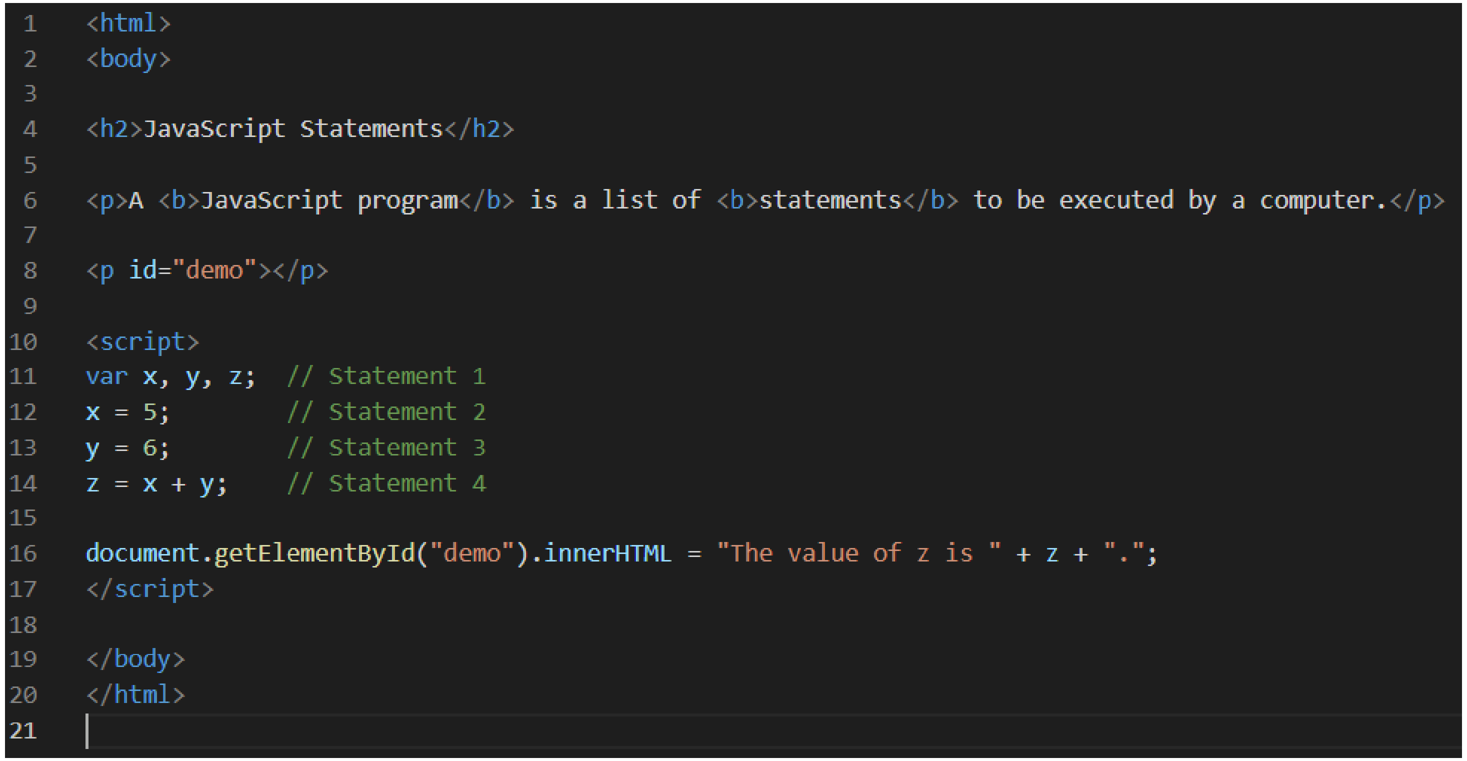
Imprimir 1 página

Destino: Salvar como PDF

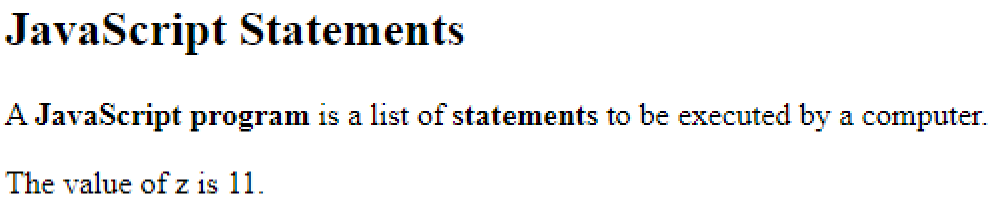
Páginas: Tudo

Layout: Retrato

Declaração em JavaScript



```
<html>
<body>

<h2>JavaScript Statements</h2>

<p>A <b>JavaScript program</b> is a list of <b>statements</b> to be executed by a computer.</p>

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

<script>
var x, y, z;  // Statement 1
x = 5;        // Statement 2
y = 6;        // Statement 3
z = x + y;    // Statement 4

document.getElementById("demo").innerHTML = "The value of z is " + z + ".";
</script>

</body>
</html>

```

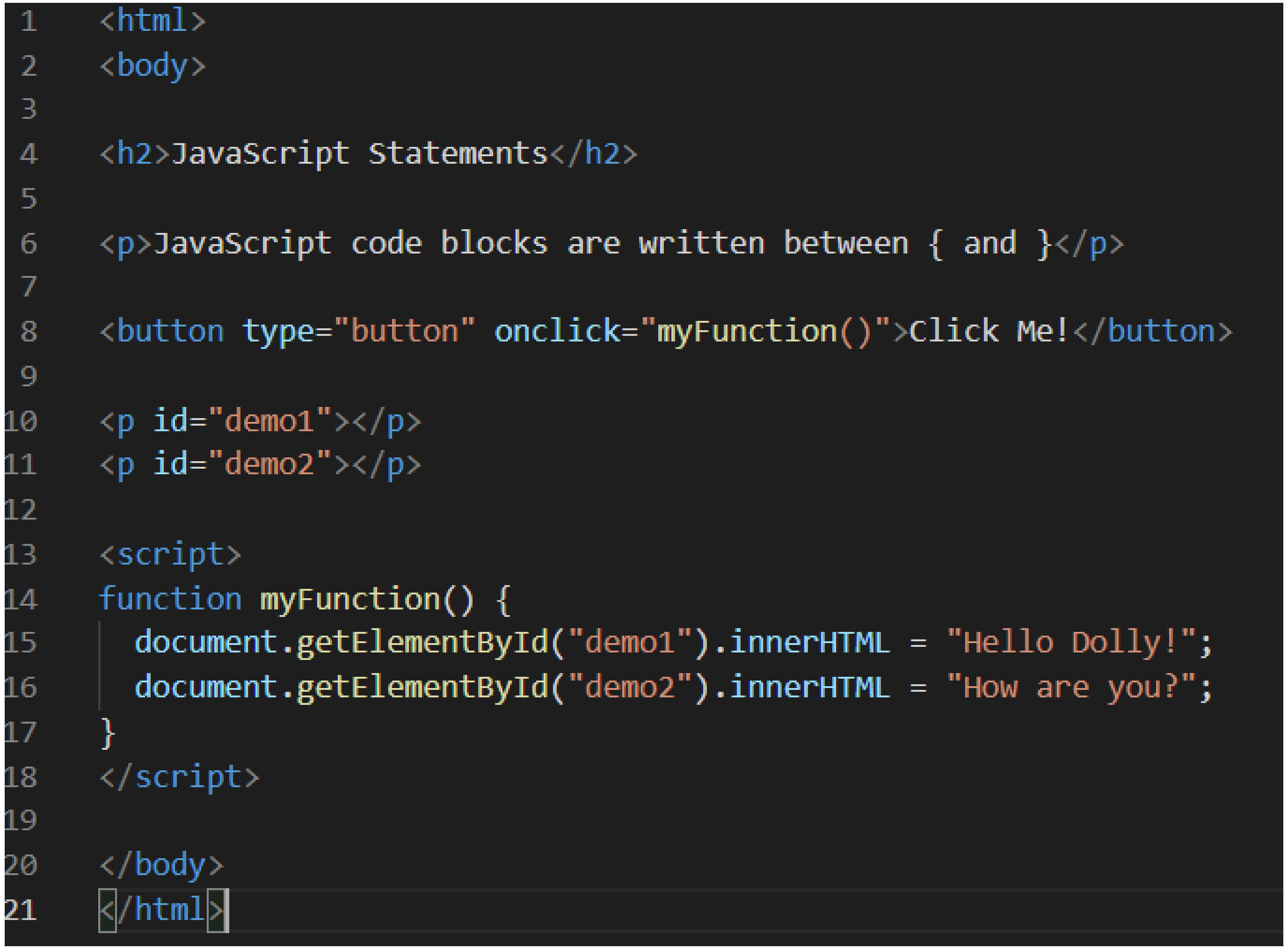


JavaScript Statements

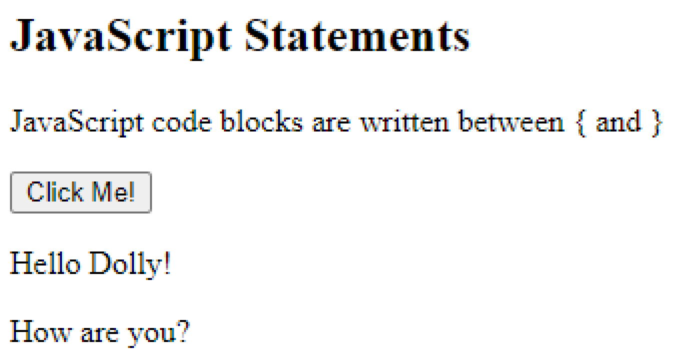
A **JavaScript program** is a list of **statements** to be executed by a computer.

The value of z is 11.

Código em blocos



```html
<html>
<body>

<h2>JavaScript Statements</h2>

<p>JavaScript code blocks are written between { and }</p>

<button type="button" onclick="myFunction()">Click Me!</button>

<p id="demo1"></p>
<p id="demo2"></p>

<script>
function myFunction() {
  document.getElementById("demo1").innerHTML = "Hello Dolly!";
  document.getElementById("demo2").innerHTML = "How are you?";
}
</script>

</body>
</html>
```



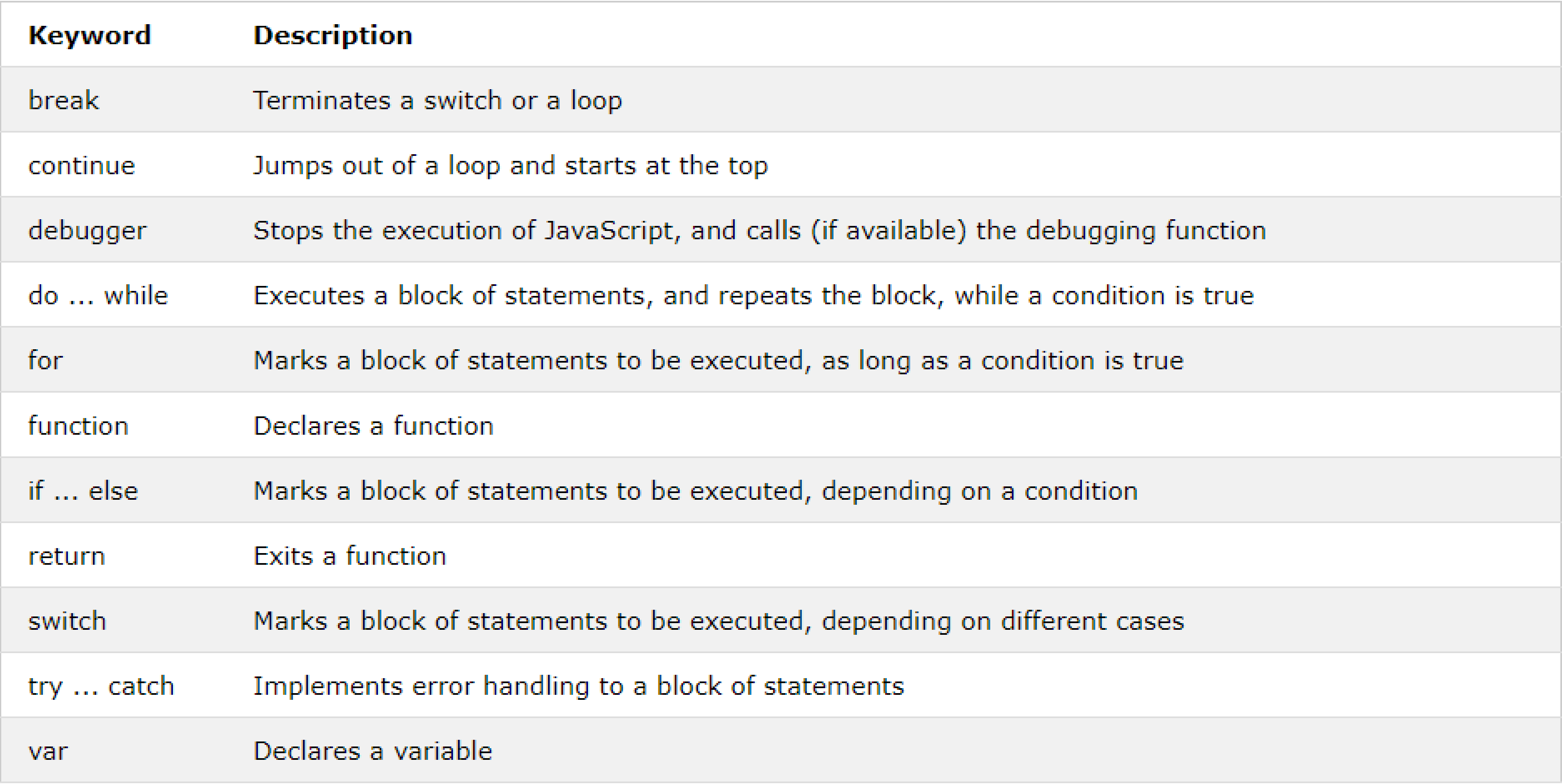
JavaScript Statements

JavaScript code blocks are written between { and }

Hello Dolly!

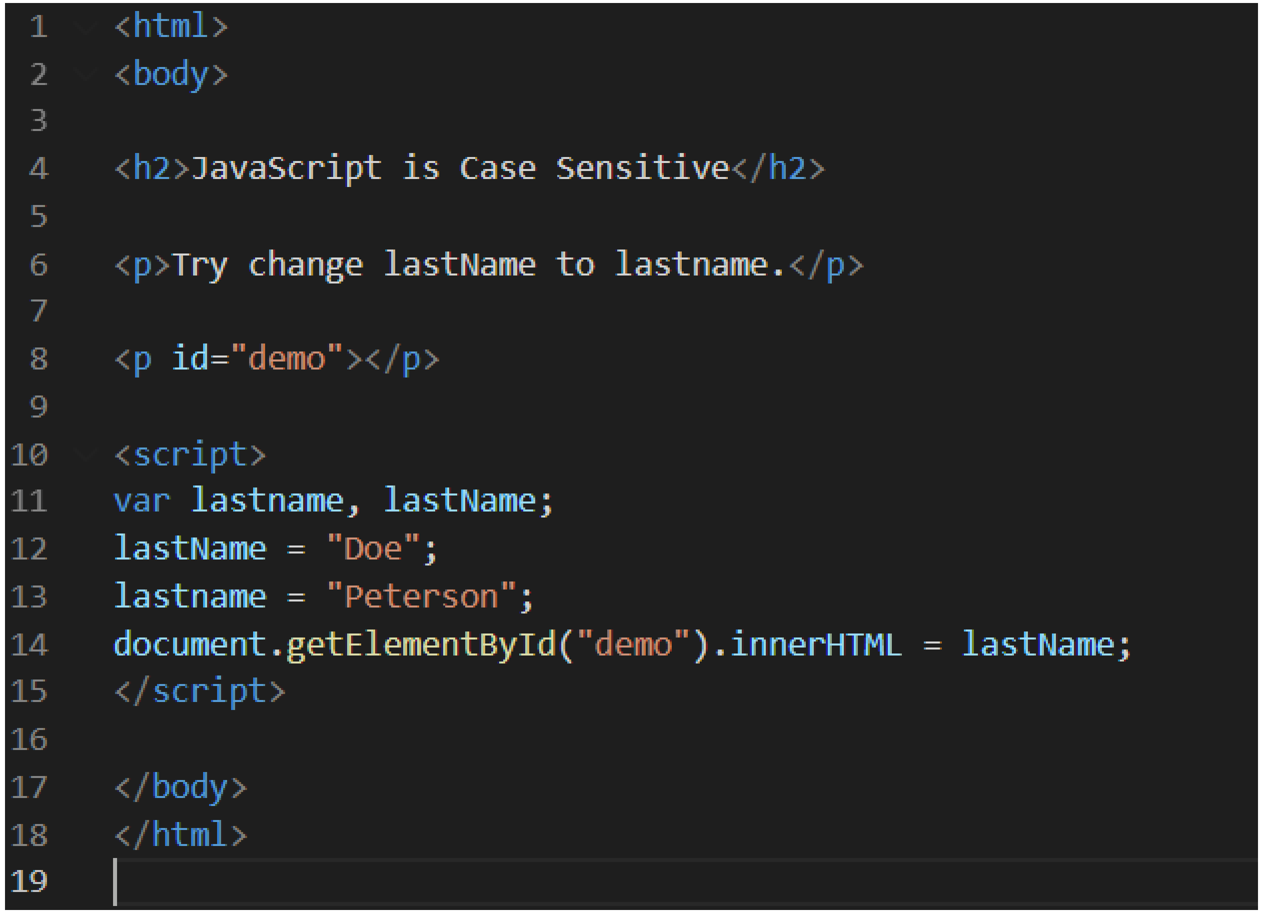
How are you?

Palavras Chave - JavaScript

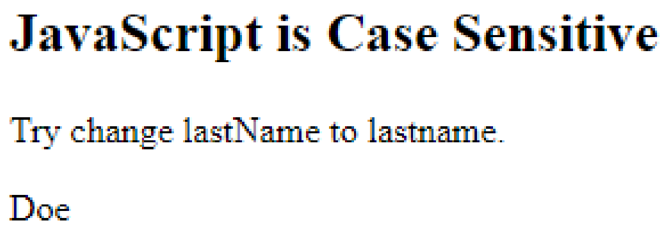


Keyword	Description
break	Terminates a switch or a loop
continue	Jumps out of a loop and starts at the top
debugger	Stops the execution of JavaScript, and calls (if available) the debugging function
do ... while	Executes a block of statements, and repeats the block, while a condition is true
for	Marks a block of statements to be executed, as long as a condition is true
function	Declares a function
if ... else	Marks a block of statements to be executed, depending on a condition
return	Exits a function
switch	Marks a block of statements to be executed, depending on different cases
try ... catch	Implements error handling to a block of statements
var	Declares a variable

JavaScript é “Case Sensitive”



```
<html>
<body>

<h2>JavaScript is Case Sensitive</h2>

<p>Try change lastName to lastname.</p>

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

<script>
var lastname, lastName;
lastName = "Doe";
lastname = "Peterson";
document.getElementById("demo").innerHTML = lastName;
</script>

</body>
</html>

```

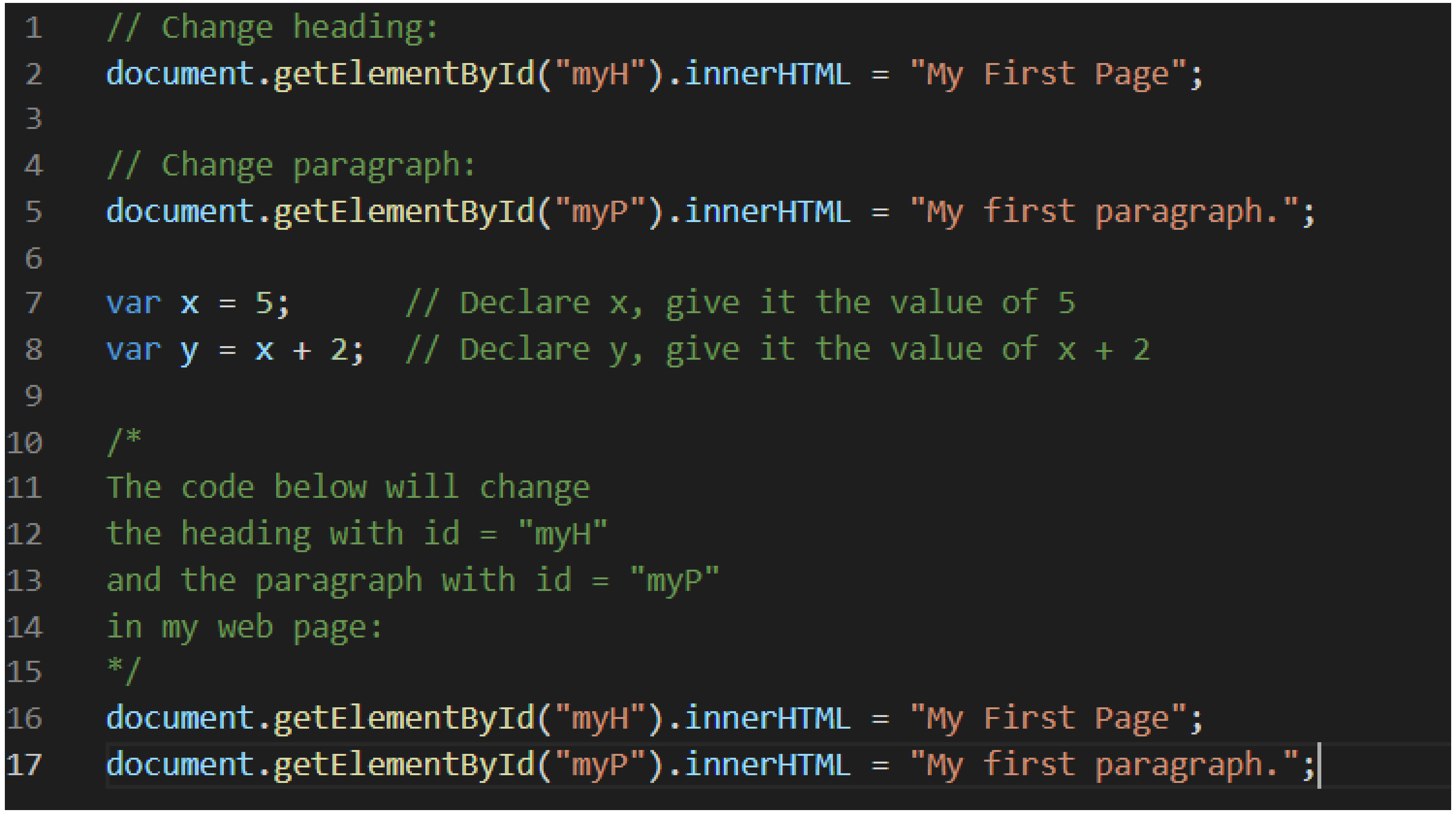


JavaScript is Case Sensitive

Try change lastName to lastname.

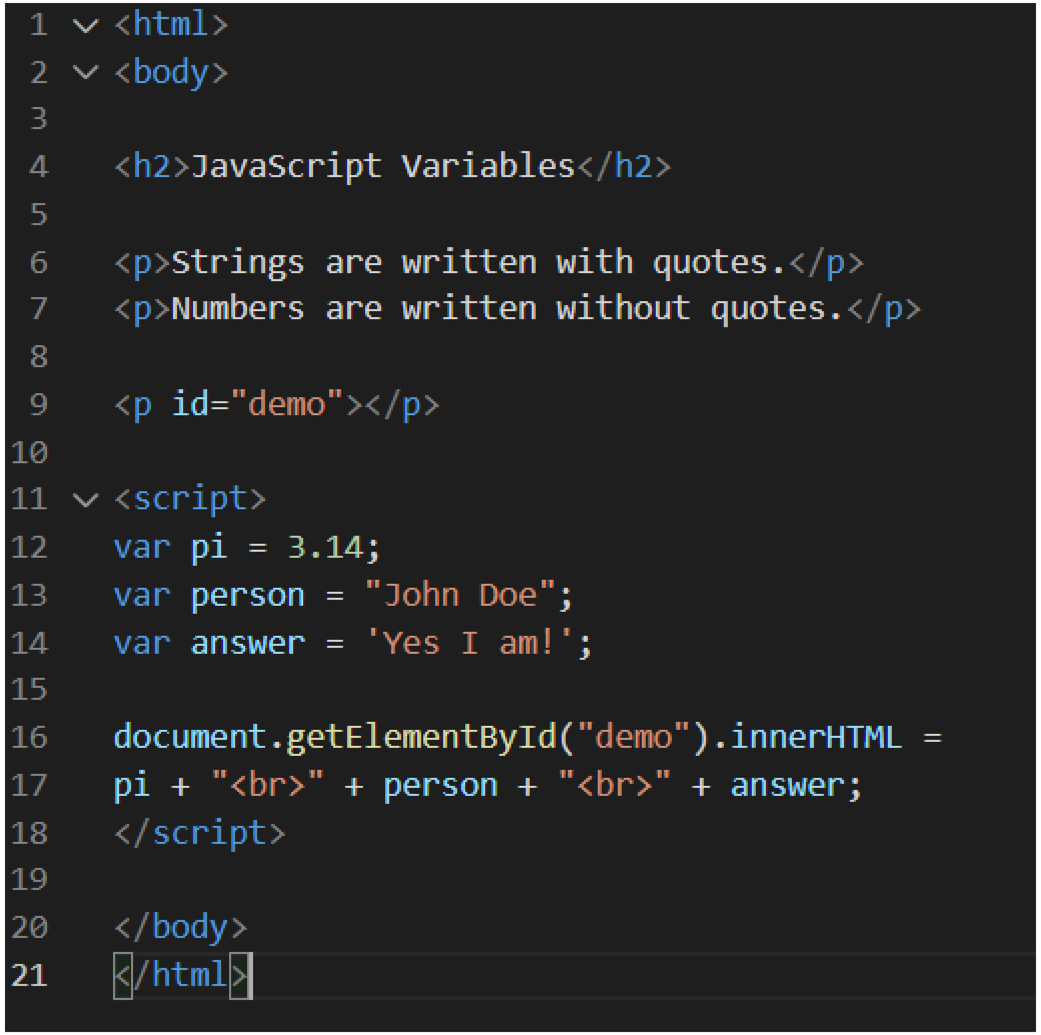
Doe

Comentário

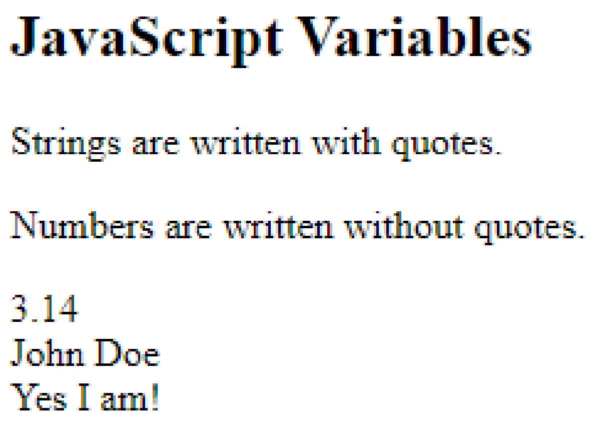


```
// Change heading:
document.getElementById("myH").innerHTML = "My First Page";

// Change paragraph:
document.getElementById("myP").innerHTML = "My first paragraph.";

var x = 5;      // Declare x, give it the value of 5
var y = x + 2;  // Declare y, give it the value of x + 2

/*
The code below will change
the heading with id = "myH"
and the paragraph with id = "myP"
in my web page:
*/
document.getElementById("myH").innerHTML = "My First Page";
document.getElementById("myP").innerHTML = "My first paragraph.";
```

Variáveis



```html
<html>
<body>

<h2>JavaScript Variables</h2>

<p>Strings are written with quotes.</p>
<p>Numbers are written without quotes.</p>

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

<script>
var pi = 3.14;
var person = "John Doe";
var answer = 'Yes I am!';

document.getElementById("demo").innerHTML =
pi + "<br>" + person + "<br>" + answer;
</script>

</body>
</html>
```



JavaScript Variables

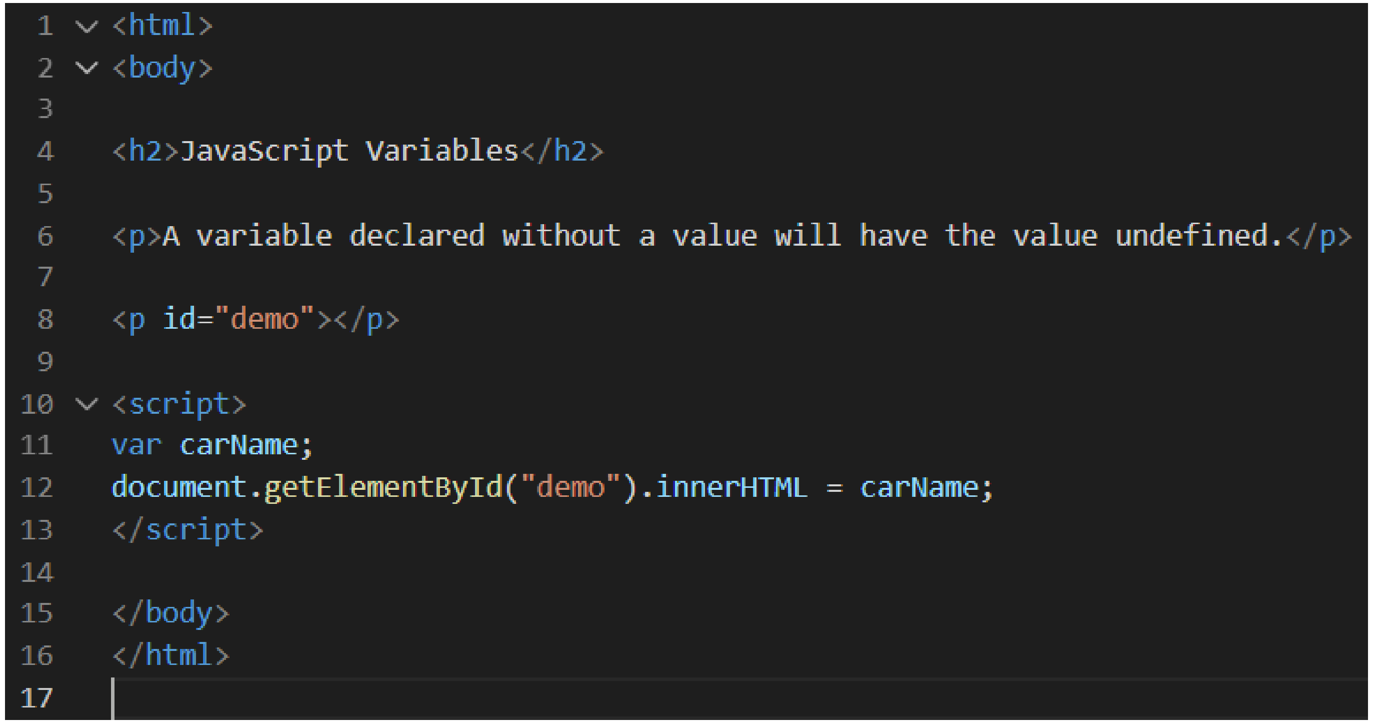
Strings are written with quotes.

Numbers are written without quotes.

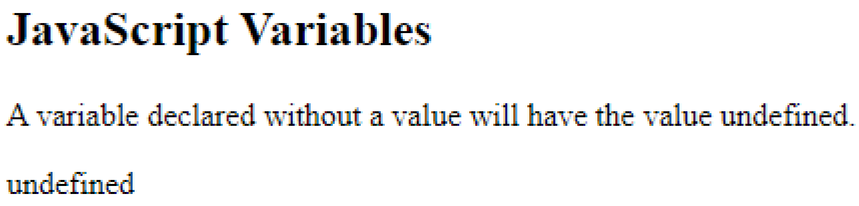
3.14
John Doe
Yes I am!

```javascript
var person = "John Doe", carName = "Volvo", price = 200;

var person = "John Doe",
carName = "Volvo",
price = 200;
```

Value = undefined



```
<html>
<body>

<h2>JavaScript Variables</h2>

<p>A variable declared without a value will have the value undefined.</p>

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

<script>
var carName;
document.getElementById("demo").innerHTML = carName;
</script>

</body>
</html>

```

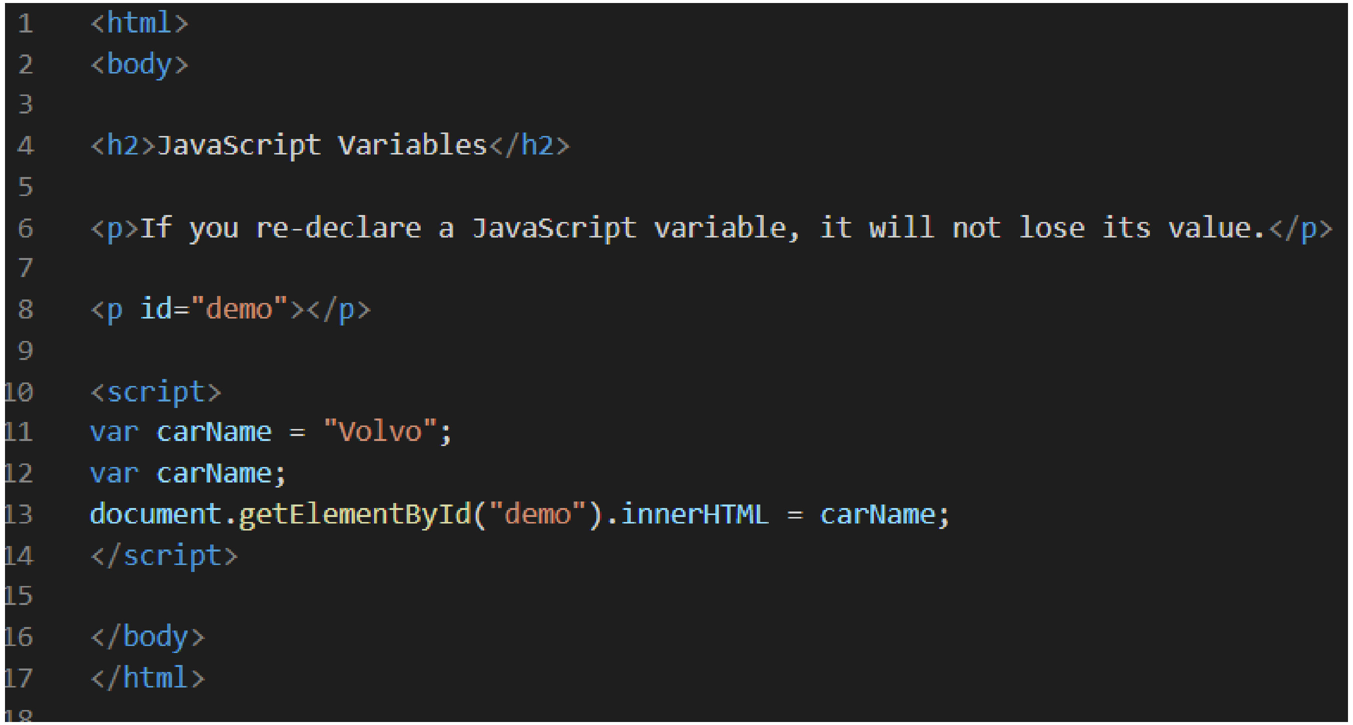


JavaScript Variables

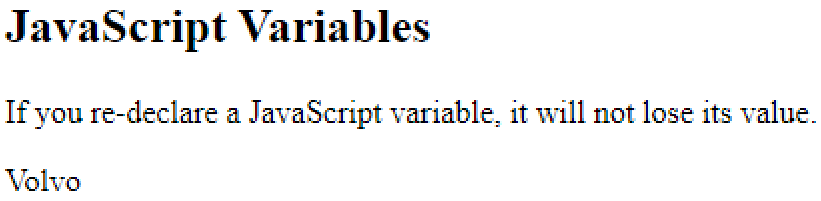
A variable declared without a value will have the value undefined.

undefined

Declarar novamente uma variável



```
<html>
<body>

<h2>JavaScript Variables</h2>

<p>If you re-declare a JavaScript variable, it will not lose its value.</p>

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

<script>
var carName = "Volvo";
var carName;
document.getElementById("demo").innerHTML = carName;
</script>

</body>
</html>
```

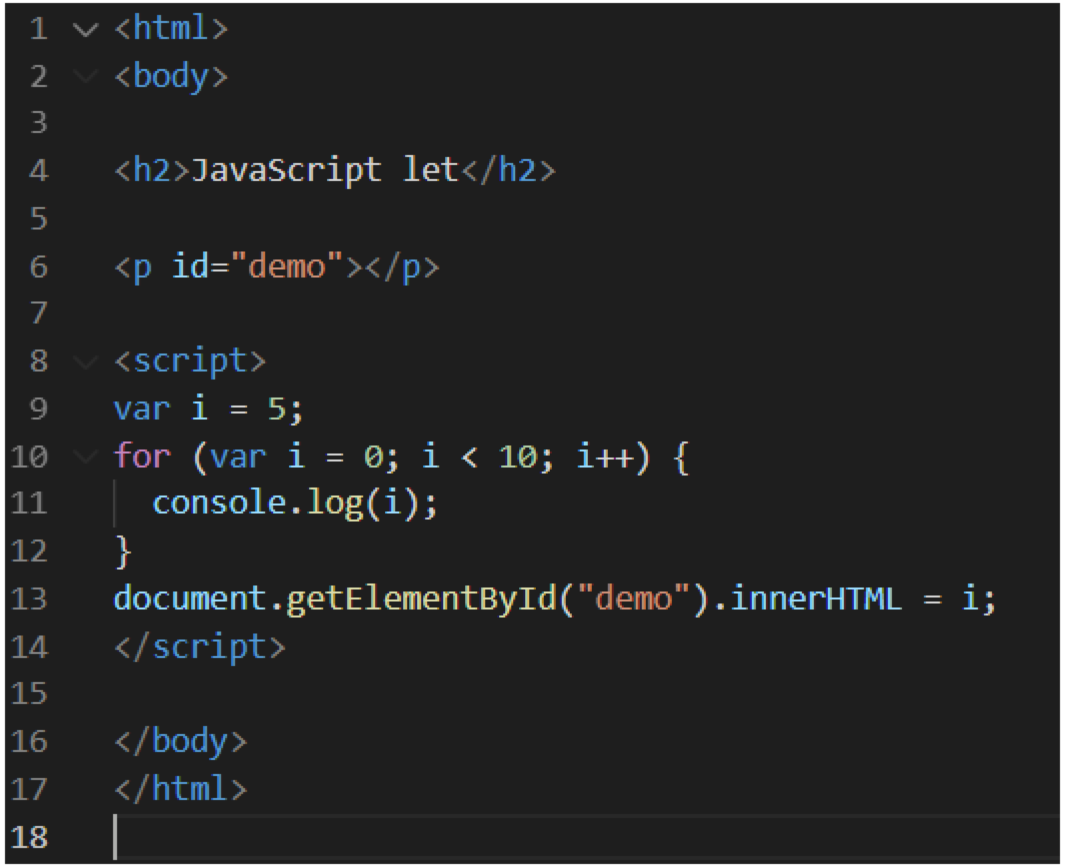


JavaScript Variables

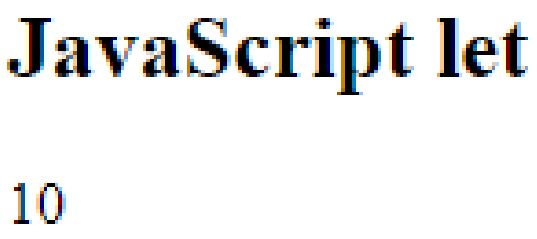
If you re-declare a JavaScript variable, it will not lose its value.

Volvo

For - JavaScript

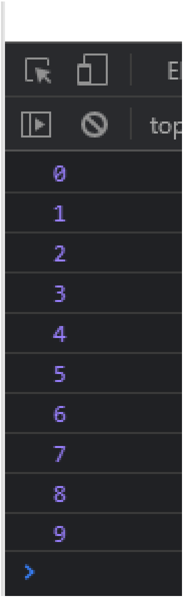


```
<html>
<body>

<h2>JavaScript let</h2>

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

<script>
var i = 5;
for (var i = 0; i < 10; i++) {
  console.log(i);
}
document.getElementById("demo").innerHTML = i;
</script>

</body>
</html>

```



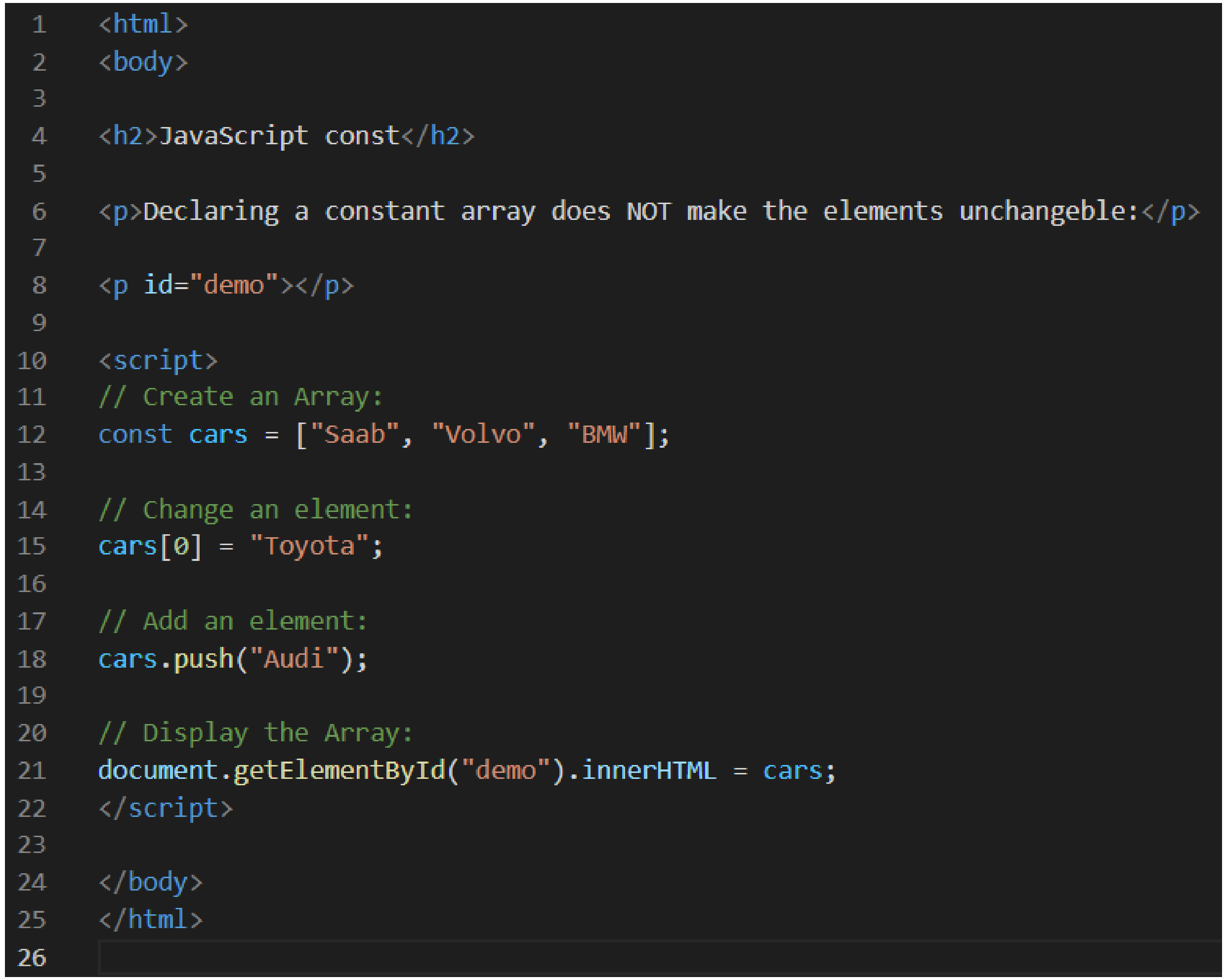
JavaScript let

10

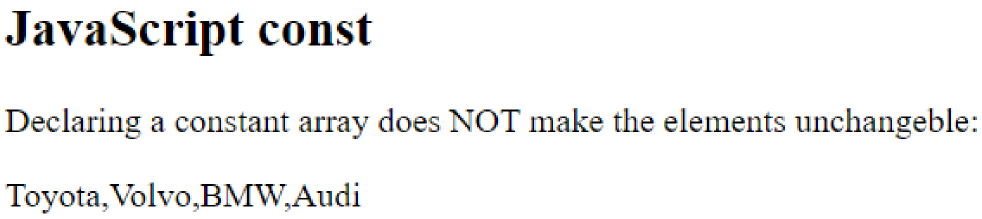


```
0
1
2
3
4
5
6
7
8
9
>
```

Constante



```html
<html>
<body>

<h2>JavaScript const</h2>

<p>Declaring a constant array does NOT make the elements unchangeble:</p>

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

<script>
// Create an Array:
const cars = ["Saab", "Volvo", "BMW"];

// Change an element:
cars[0] = "Toyota";

// Add an element:
cars.push("Audi");

// Display the Array:
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>

```

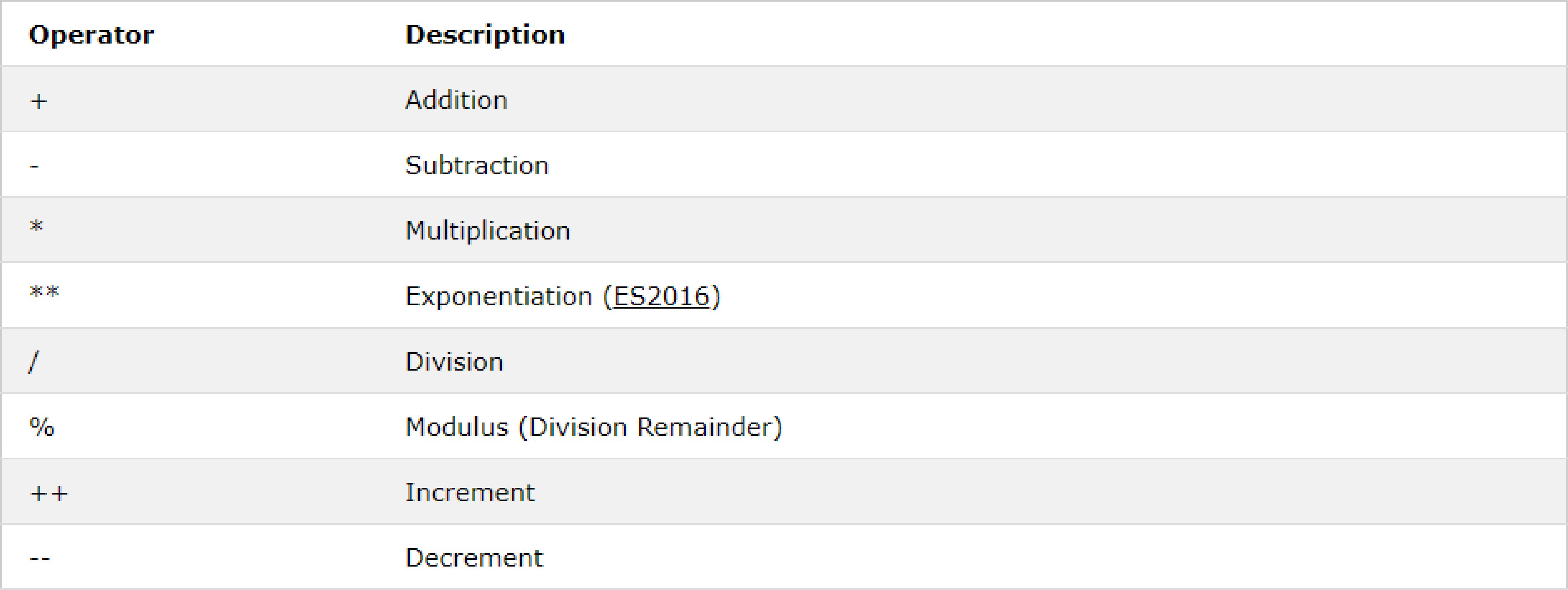


JavaScript const

Declaring a constant array does NOT make the elements unchangeble:

Toyota,Volvo,BMW,Audi

Operadores matemáticos



Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (ES2016)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

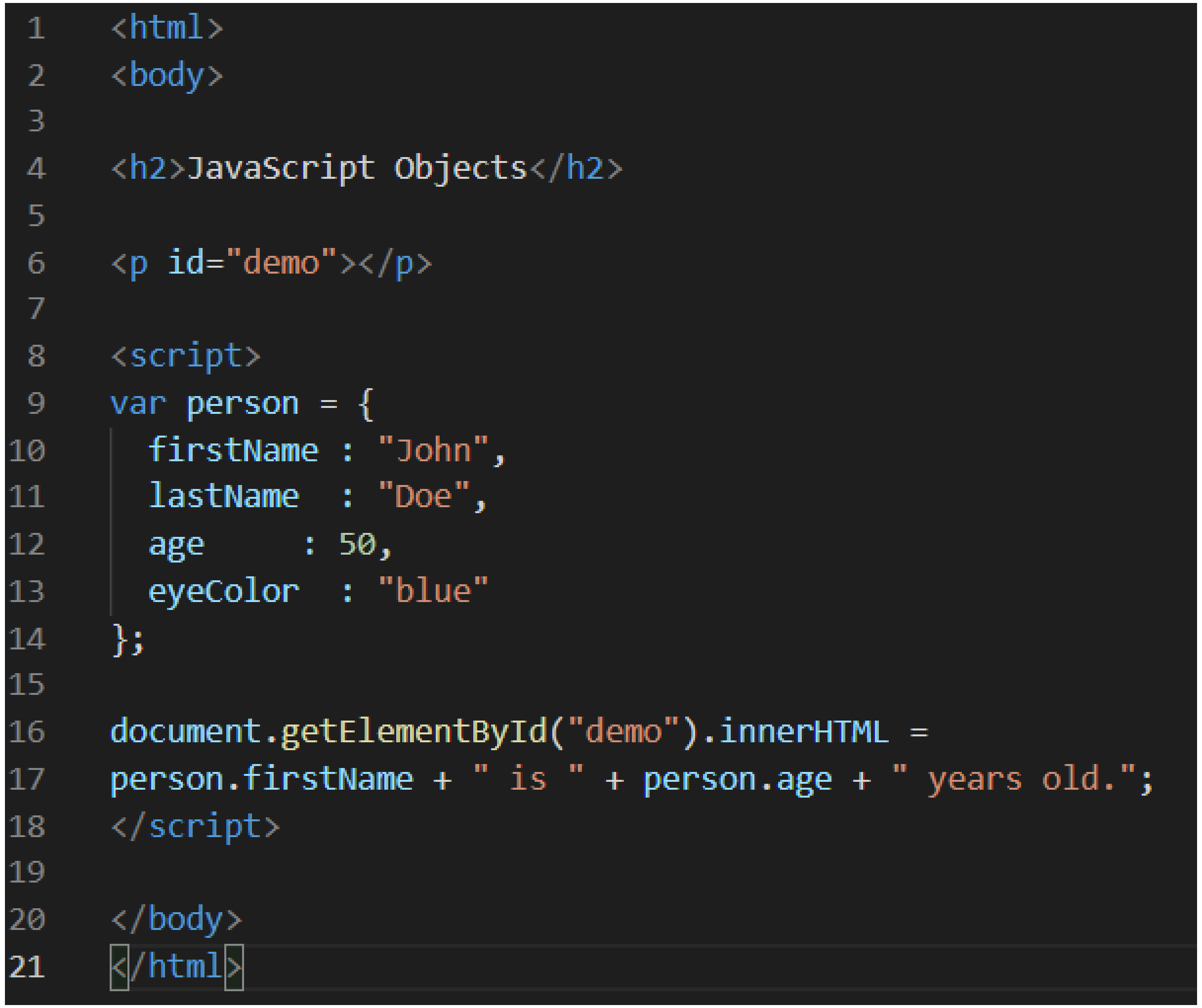
Tipos de Dados

```
var length = 16;                                        // Number
var lastName = "Johnson";                               // String
var x = {firstName:"John", lastName:"Doe"};    // Object
```

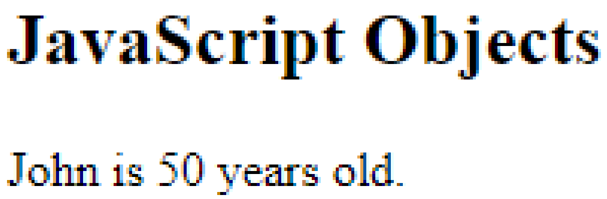
```
var x = "Volvo" + 16; Volvo16
```

```
var x = 16 + 4 + "Volvo"; 20Volvo
```

Objeto JavaScript

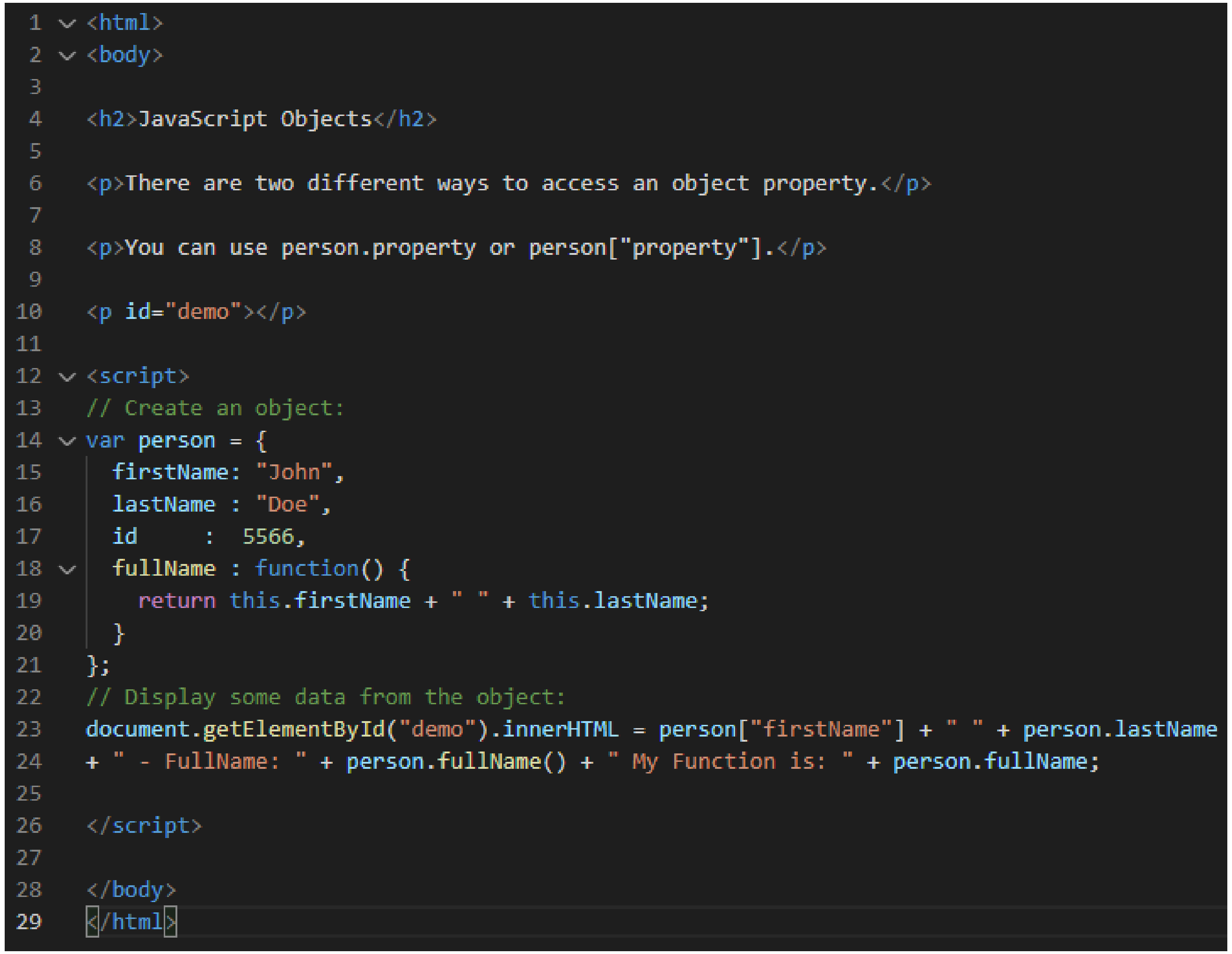


```
<html>
<body>

<h2>JavaScript Objects</h2>

<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>
```

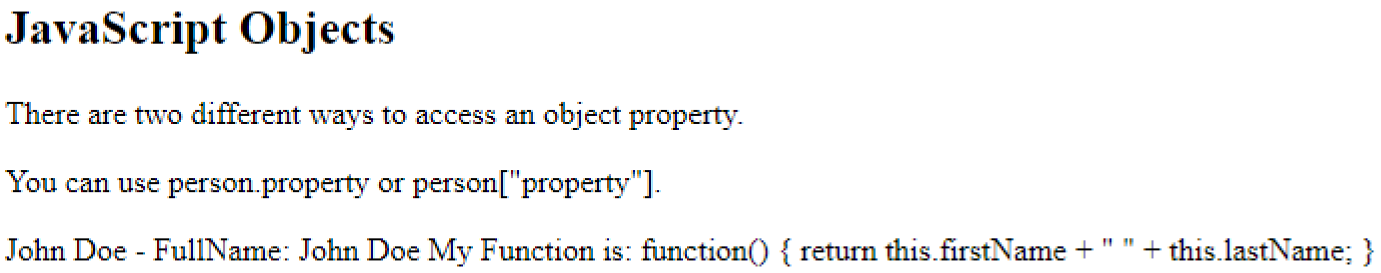


JavaScript Objects

John is 50 years old.



```
<html>
<body>

<h2>JavaScript Objects</h2>

<p>There are two different ways to access an object property.</p>

<p>You can use person.property or person["property"].</p>

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

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  id     :  5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
// Display some data from the object:
document.getElementById("demo").innerHTML = person["firstName"] + " " + person.lastName
+ " - FullName: " + person.fullName() + " My Function is: " + person.fullName;

</script>

</body>
</html>
```



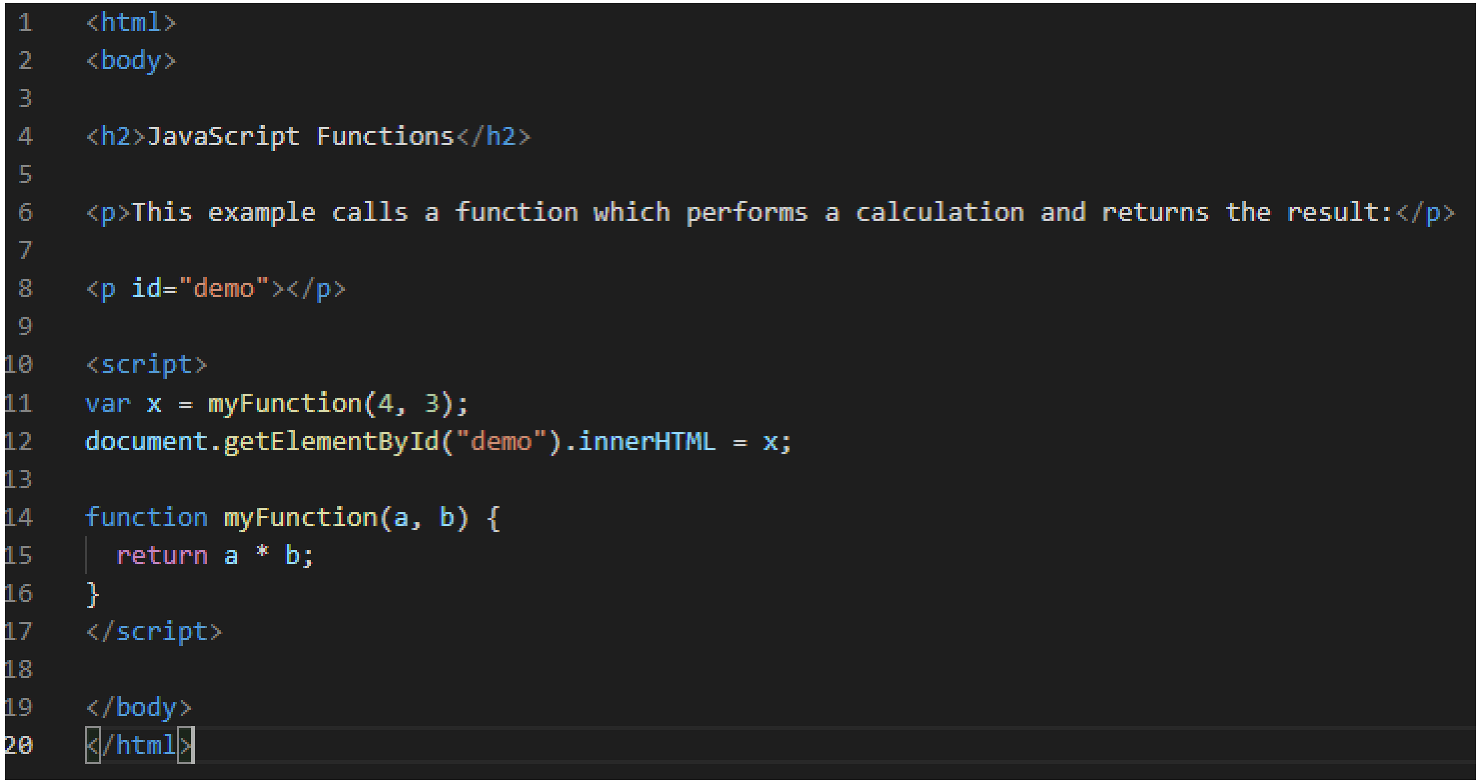
JavaScript Objects

There are two different ways to access an object property.

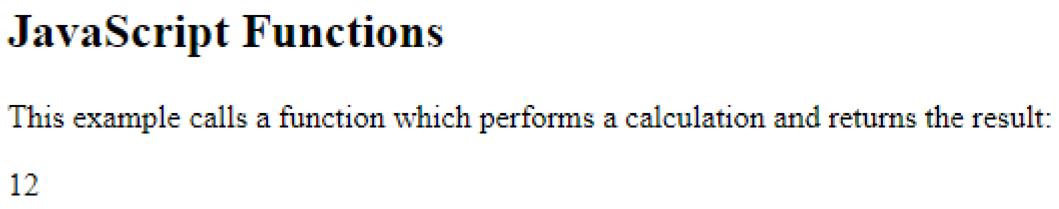
You can use person.property or person["property"].

John Doe - FullName: John Doe My Function is: function() { return this.firstName + " " + this.lastName; }

Funções JavaScript



```
<html>
<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function which performs a calculation and returns the result:</p>

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

<script>
var x = myFunction(4, 3);
document.getElementById("demo").innerHTML = x;

function myFunction(a, b) {
  return a * b;
}
</script>

</body>
</html>
```



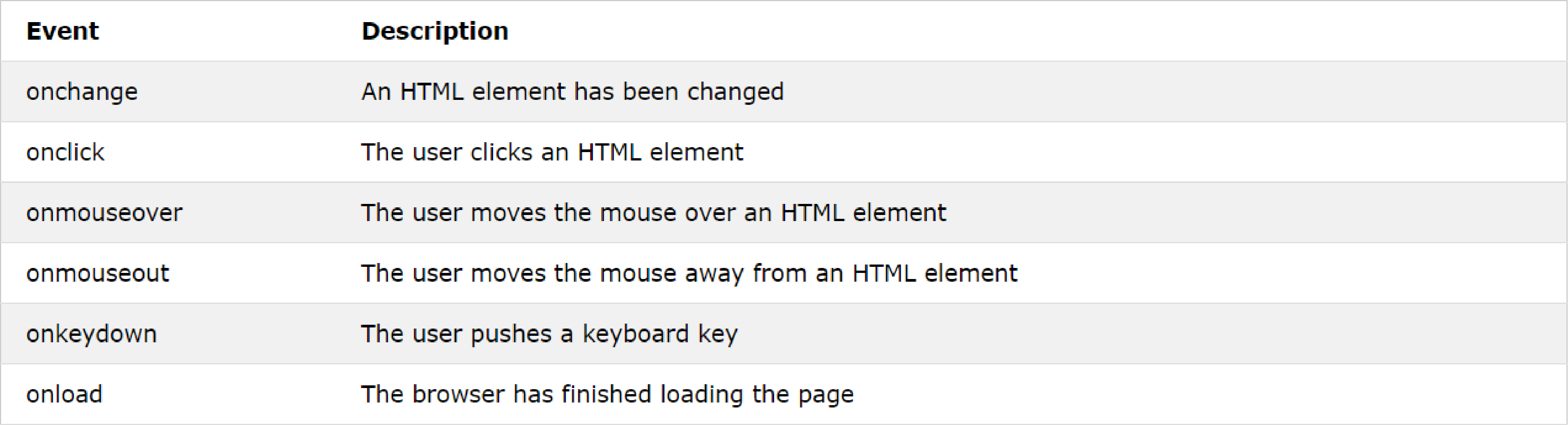
JavaScript Functions

This example calls a function which performs a calculation and returns the result:

12

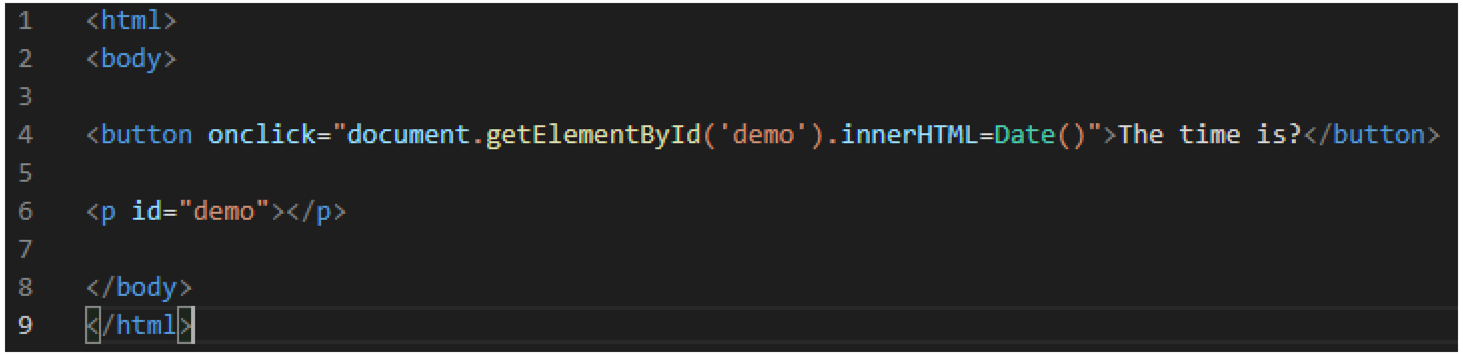
Eventos no JavaScript

- São “coisas” que acontecem nos elementos HTML.
- Eventos comuns do HTML:

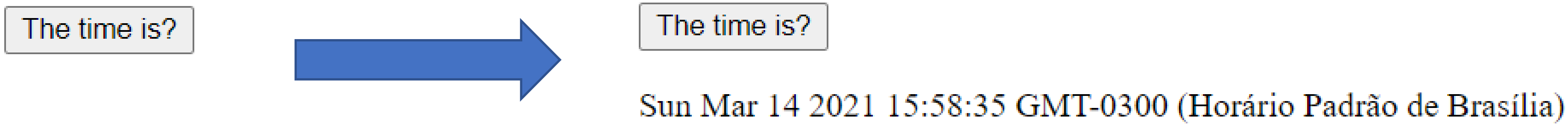


Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Eventos

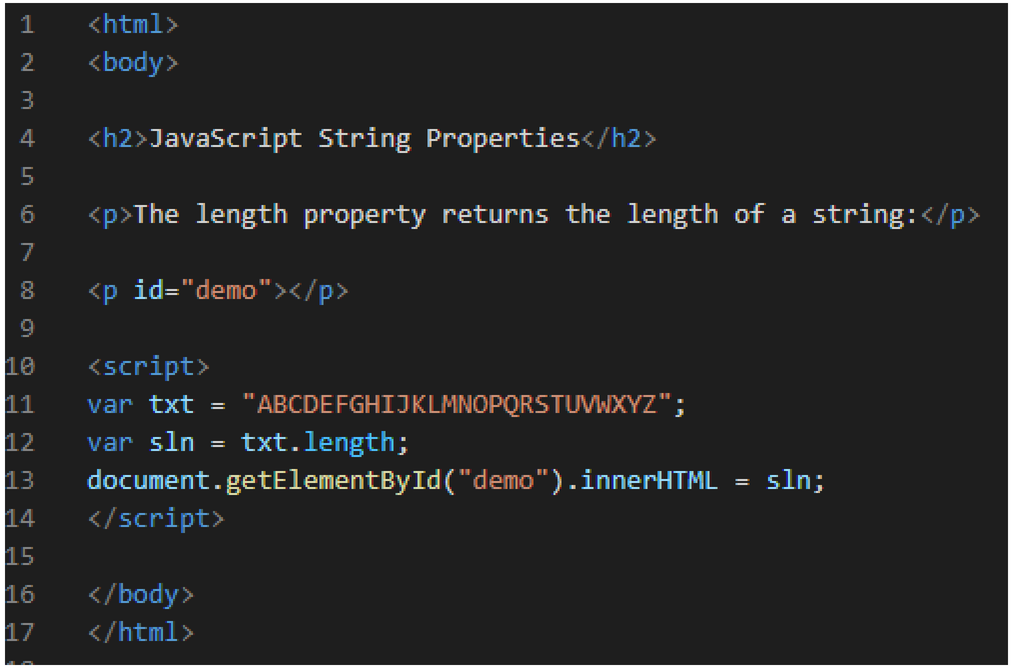


```html
<html>
<body>

<button onclick="document.getElementById('demo').innerHTML=Date()">The time is?</button>

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

</body>
</html>
```

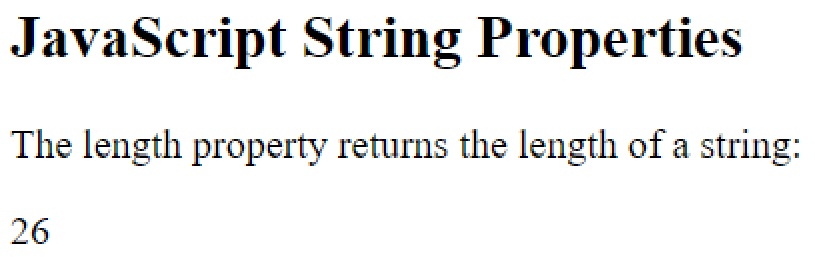


```
Sun Mar 14 2021 15:58:35 GMT-0300 (Horário Padrão de Brasília)
```

String – Length



```
<html>
<body>

<h2>JavaScript String Properties</h2>

<p>The length property returns the length of a string:</p>

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

<script>
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length;
document.getElementById("demo").innerHTML = sln;
</script>

</body>
</html>
```

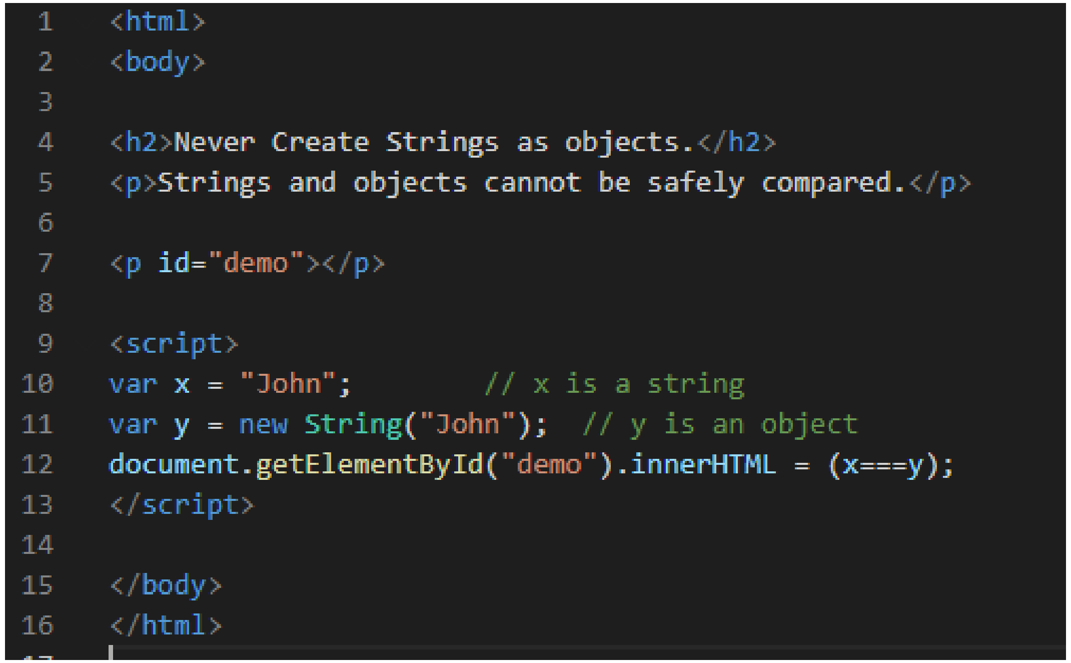


JavaScript String Properties

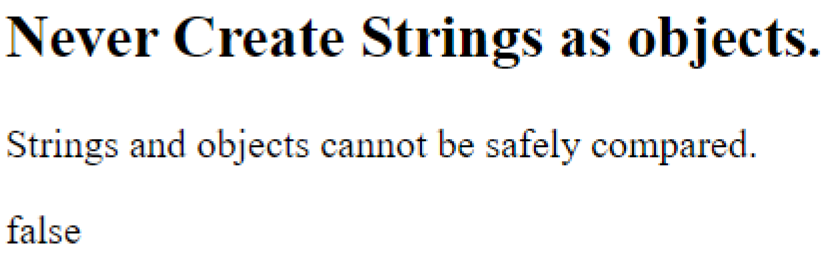
The length property returns the length of a string:

26

Objeto String x valor String



```
<html>
<body>

<h2>Never Create Strings as objects.</h2>
<p>Strings and objects cannot be safely compared.</p>

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

<script>
var x = "John";          // x is a string
var y = new String("John");  // y is an object
document.getElementById("demo").innerHTML = (x===y);
</script>

</body>
</html>
```



Never Create Strings as objects.

Strings and objects cannot be safely compared.

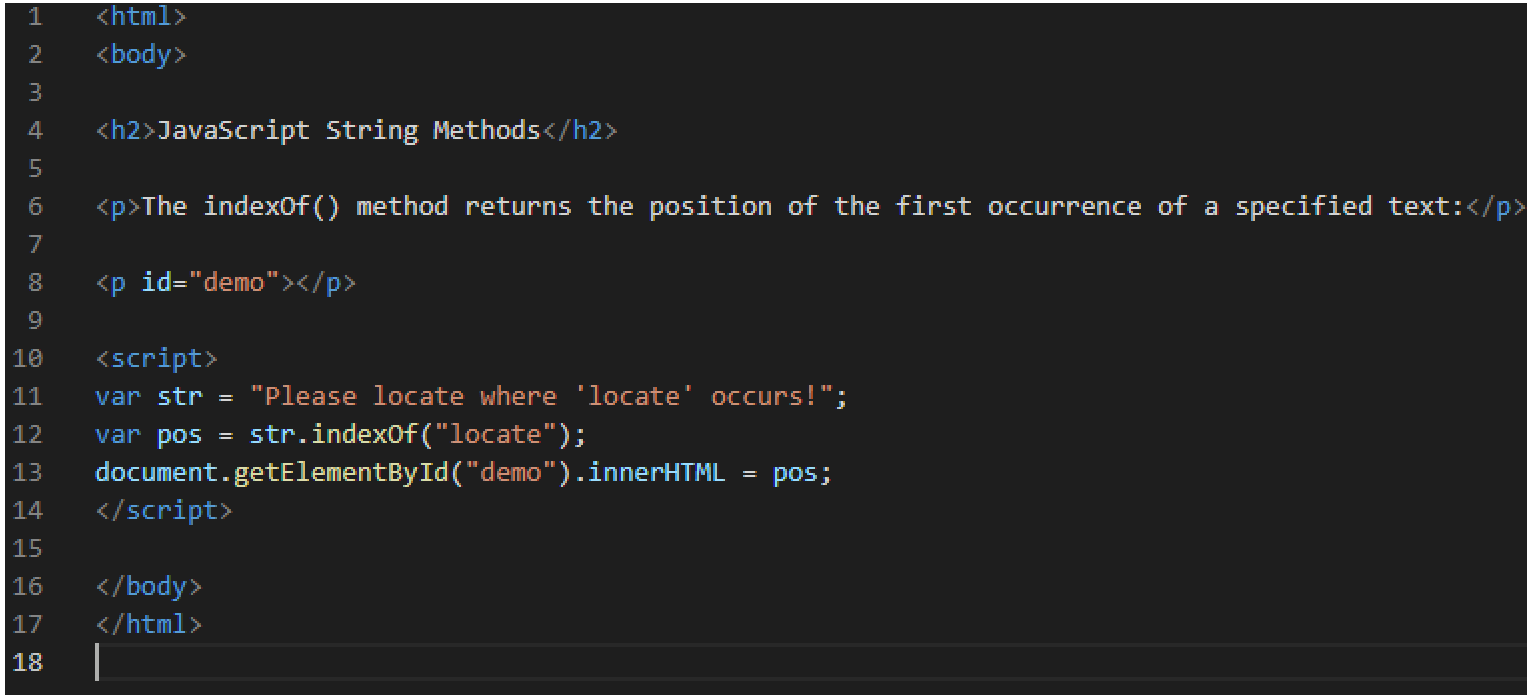
false

==   Quando os valores são iguais

=== Quando os valores e tipos de dados são iguais

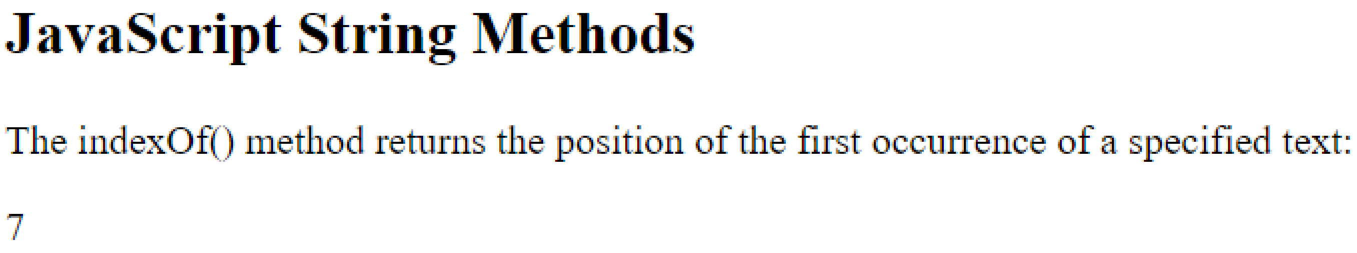
```
var x = new String("John");
var y = new String("John");

// (x == y) é falso, porque x e y são objetos
```

Funções (String) – `indexOf e lastIndexOf`



```html
<html>
<body>

<h2>JavaScript String Methods</h2>

<p>The indexOf() method returns the position of the first occurrence of a specified text:</p>

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

<script>
var str = "Please locate where 'locate' occurs!";
var pos = str.indexOf("locate");
document.getElementById("demo").innerHTML = pos;
</script>

</body>
</html>

```

*Primeiro elemento encontrado.



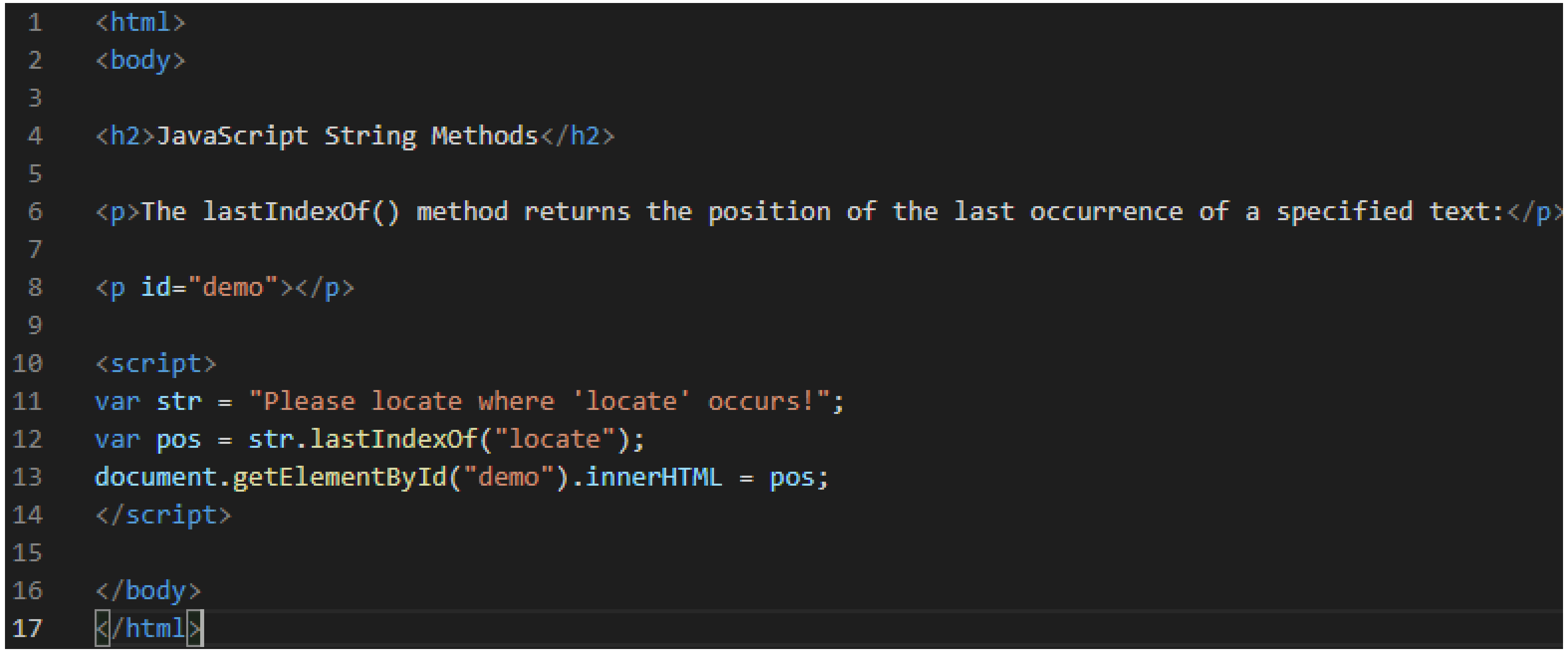
JavaScript String Methods

The indexOf() method returns the position of the first occurrence of a specified text:

7

É possível definir a posição inicial para a busca:

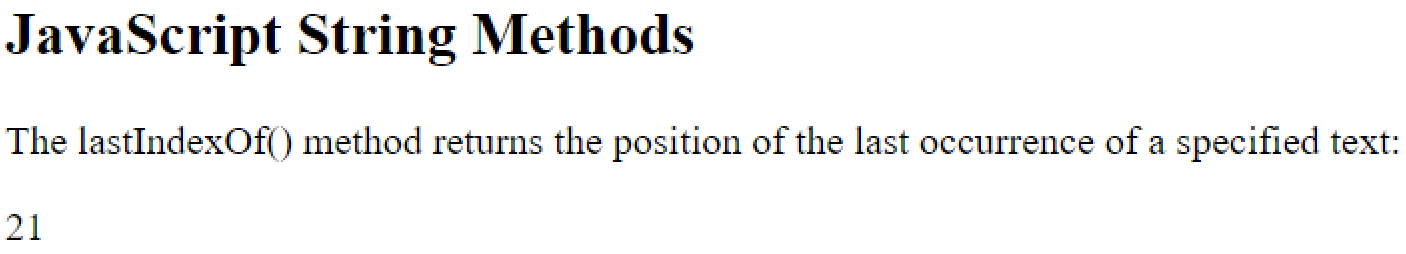
var pos = str.indexOf("locate",15);



```
<html>
<body>

<h2>JavaScript String Methods</h2>

<p>The lastIndexOf() method returns the position of the last occurrence of a specified text:</p>

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

<script>
var str = "Please locate where 'locate' occurs!";
var pos = str.lastIndexOf("locate");
document.getElementById("demo").innerHTML = pos;
</script>

</body>
</html>
```

*Último elemento encontrado.

*Se o elemento não for encontrado é retornado o valor -1

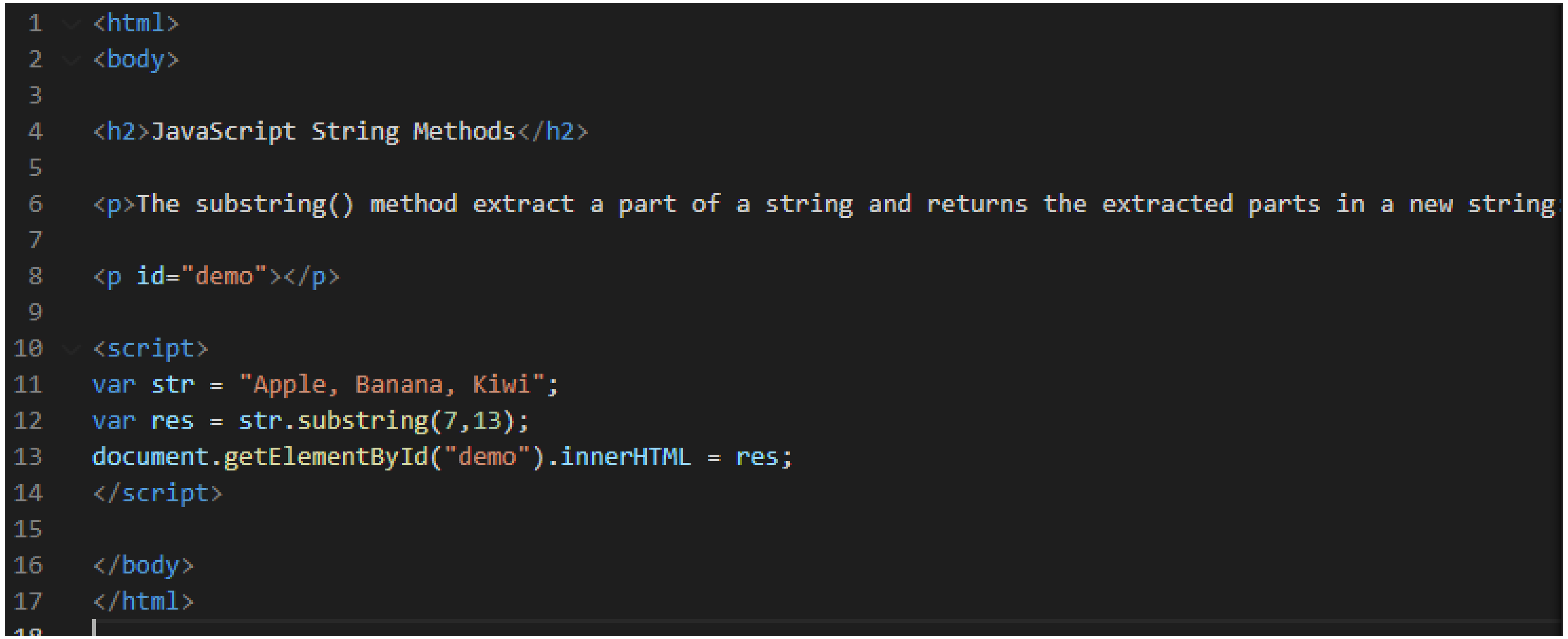


JavaScript String Methods

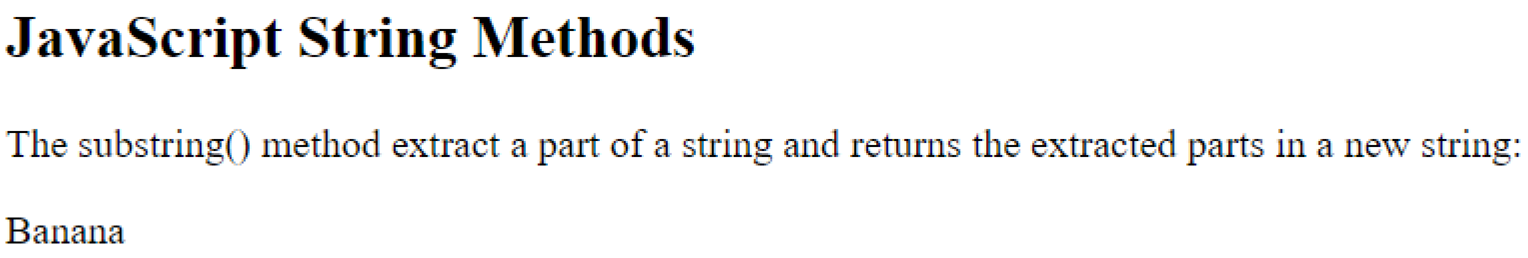
The lastIndexOf() method returns the position of the last occurrence of a specified text:

21

Substring()



```
<html>
<body>

<h2>JavaScript String Methods</h2>

<p>The substring() method extract a part of a string and returns the extracted parts in a new string

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

<script>
var str = "Apple, Banana, Kiwi";
var res = str.substring(7,13);
document.getElementById("demo").innerHTML = res;
</script>

</body>
</html>
```

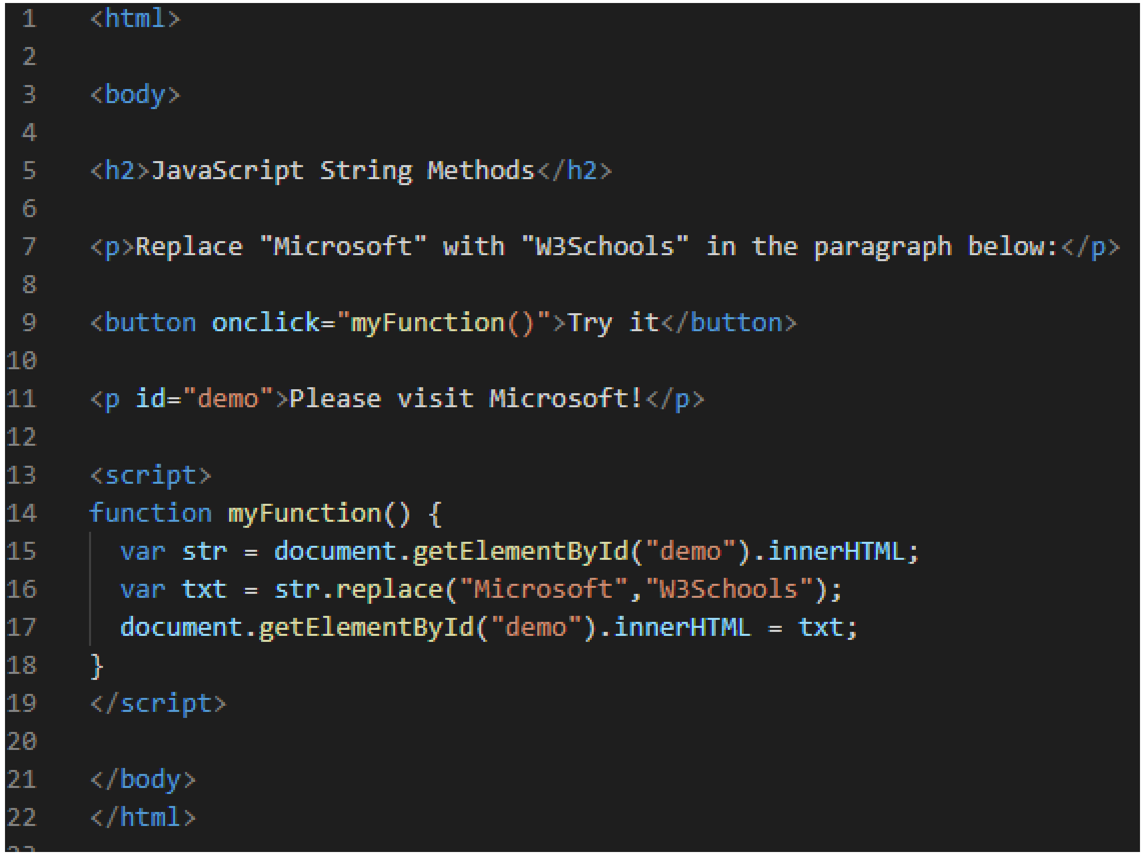


JavaScript String Methods

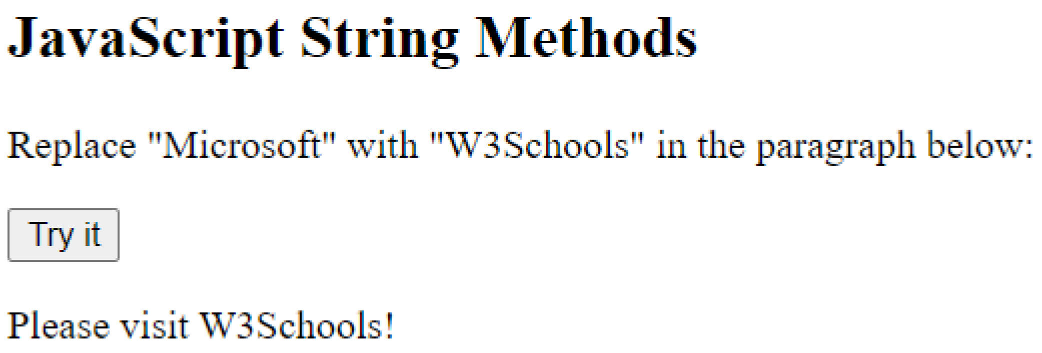
The substring() method extract a part of a string and returns the extracted parts in a new string:

Banana

Replace()



```
<html>

<body>

<h2>JavaScript String Methods</h2>

<p>Replace "Microsoft" with "W3Schools" in the paragraph below:</p>

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

<p id="demo">Please visit Microsoft!</p>

<script>
function myFunction() {
  var str = document.getElementById("demo").innerHTML;
  var txt = str.replace("Microsoft","W3Schools");
  document.getElementById("demo").innerHTML = txt;
}
</script>

</body>
</html>
```



JavaScript String Methods

Replace "Microsoft" with "W3Schools" in the paragraph below:

Please visit W3Schools!

Funções (String)

```
var text1 = "Hello World!";       // String
var text2 = text1.toUpperCase();  // text2 is text1 converted to upper


var text1 = "Hello World!";       // String
var text2 = text1.toLowerCase();  // text2 is text1 converted to lower


var text1 = "Hello";
var text2 = "World";
var text3 = text1.concat(" ", text2);

var str = "       Hello World!        ";
alert(str.trim());
```

```
var str = "5";
str = str.padStart(4,0);  // result is 0005
```

```
var str = "5";
str = str.padEnd(4,0);    // result is 5000
```

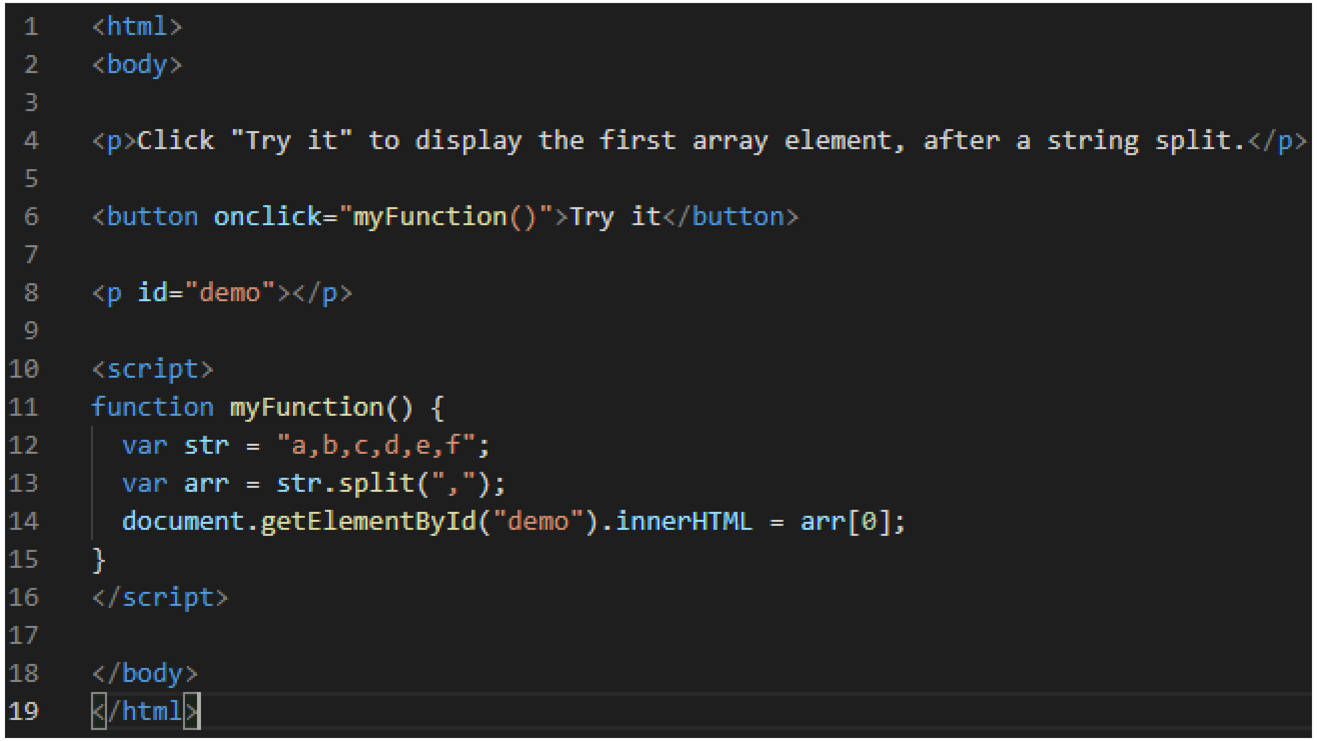
```
var str = "HELLO WORLD";
str.charAt(0);            // returns H
```

```
var str = "HELLO WORLD";
str.charCodeAt(0);        // returns 72 (UTF-16 code)
```

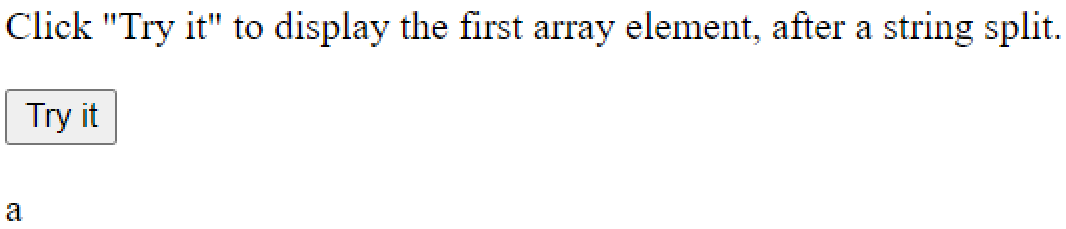
```
var str = "HELLO WORLD";
str[0];                   // returns H
```

```
var str = "HELLO WORLD";
str[0] = "A";             // Gives no error, but does not work
str[0];                   // returns H
```

Split

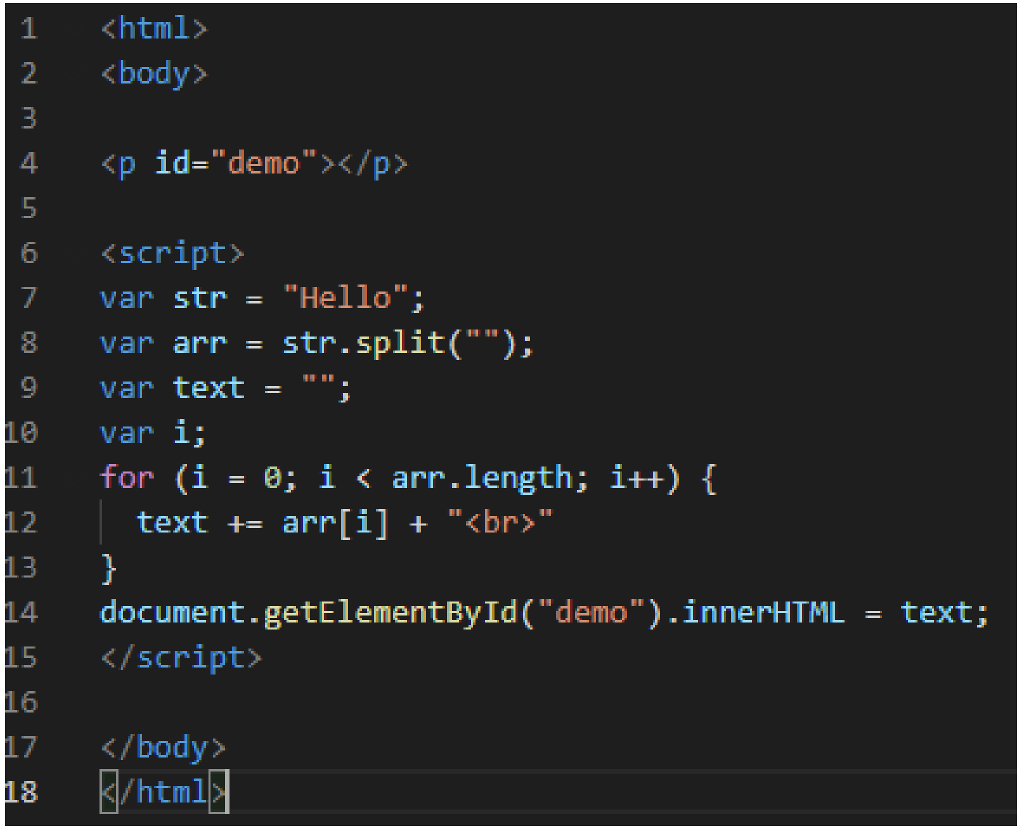


```
<html>
<body>

<p>Click "Try it" to display the first array element, after a string split.</p>

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

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

<script>
function myFunction() {
  var str = "a,b,c,d,e,f";
  var arr = str.split(",");
  document.getElementById("demo").innerHTML = arr[0];
}
</script>

</body>
</html>
```



Click "Try it" to display the first array element, after a string split.

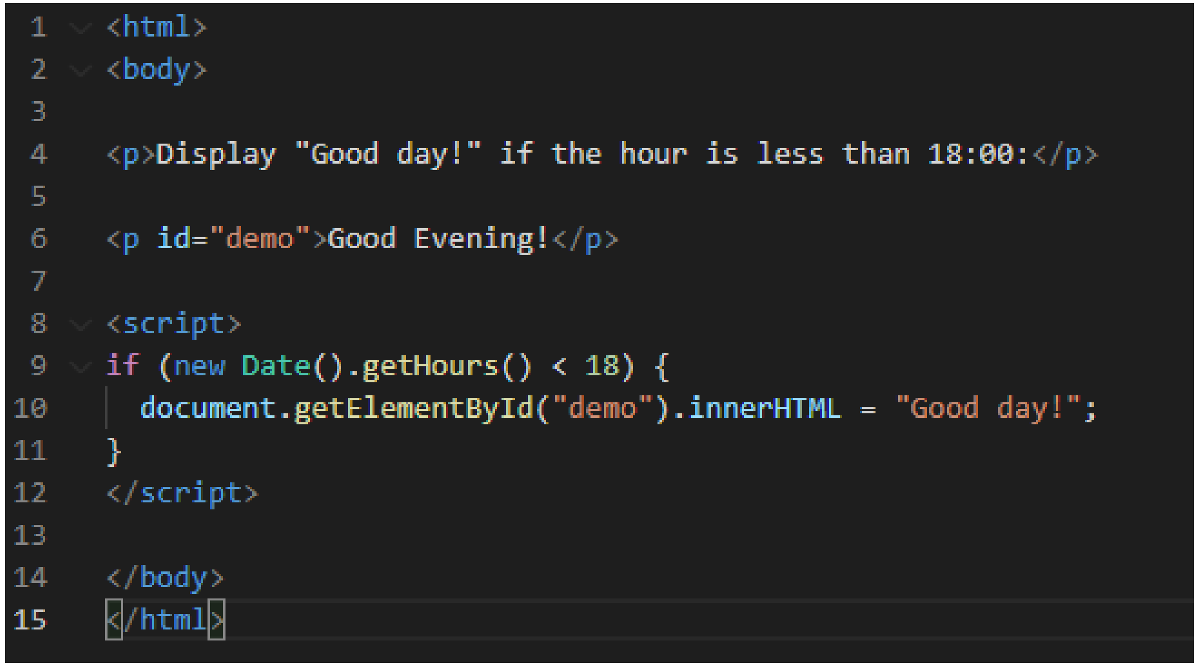
a



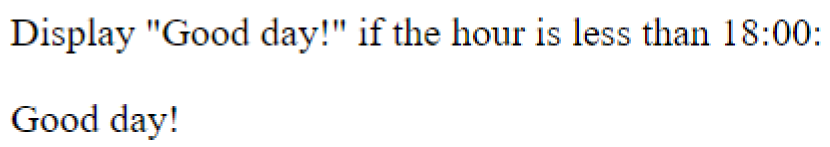
```html
<html>
<body>

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

<script>
var str = "Hello";
var arr = str.split("");
var text = "";
var i;
for (i = 0; i < arr.length; i++) {
  text += arr[i] + "<br>"
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

```
H
e
l
l
o
```

if else e else if



```
<html>
<body>

<p>Display "Good day!" if the hour is less than 18:00:</p>

<p id="demo">Good Evening!</p>

<script>
if (new Date().getHours() < 18) {
  document.getElementById("demo").innerHTML = "Good day!";
}
</script>

</body>
</html>
```

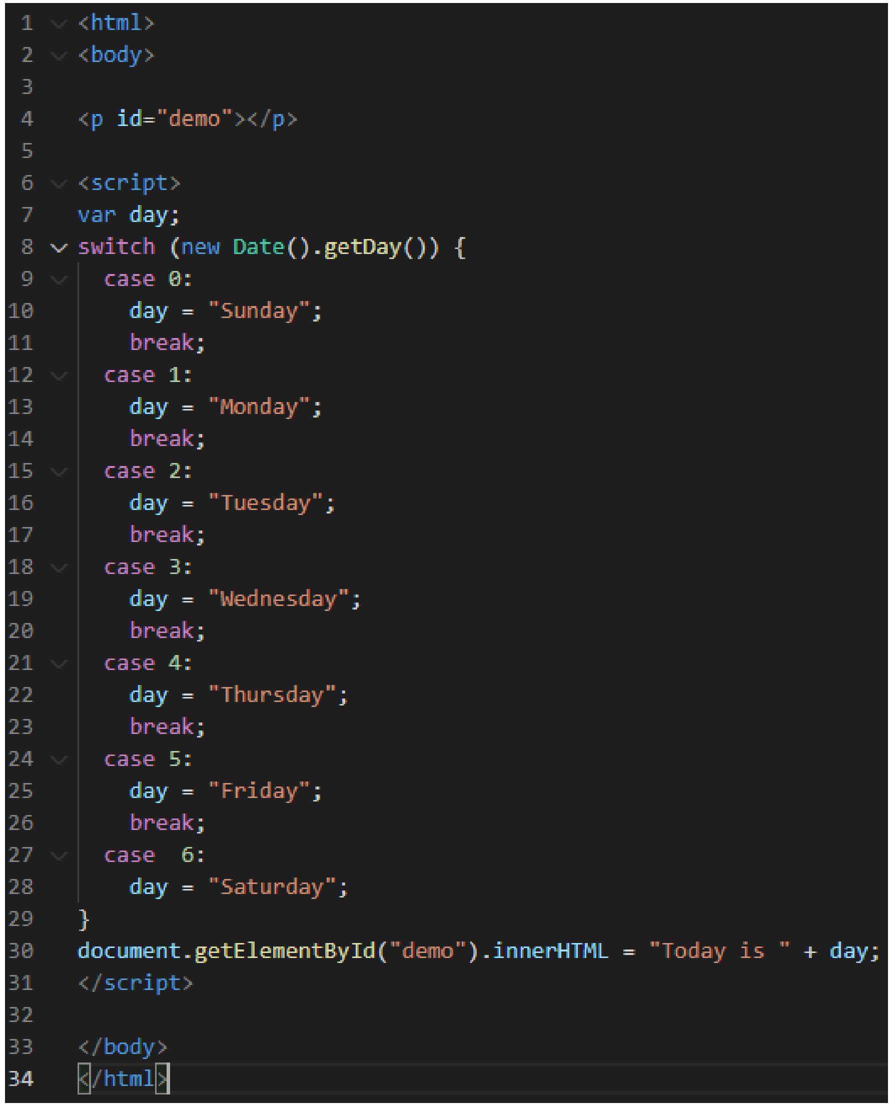


```
Display "Good day!" if the hour is less than 18:00:

Good day!
```

```
if (hour < 18) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

```
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

Switch

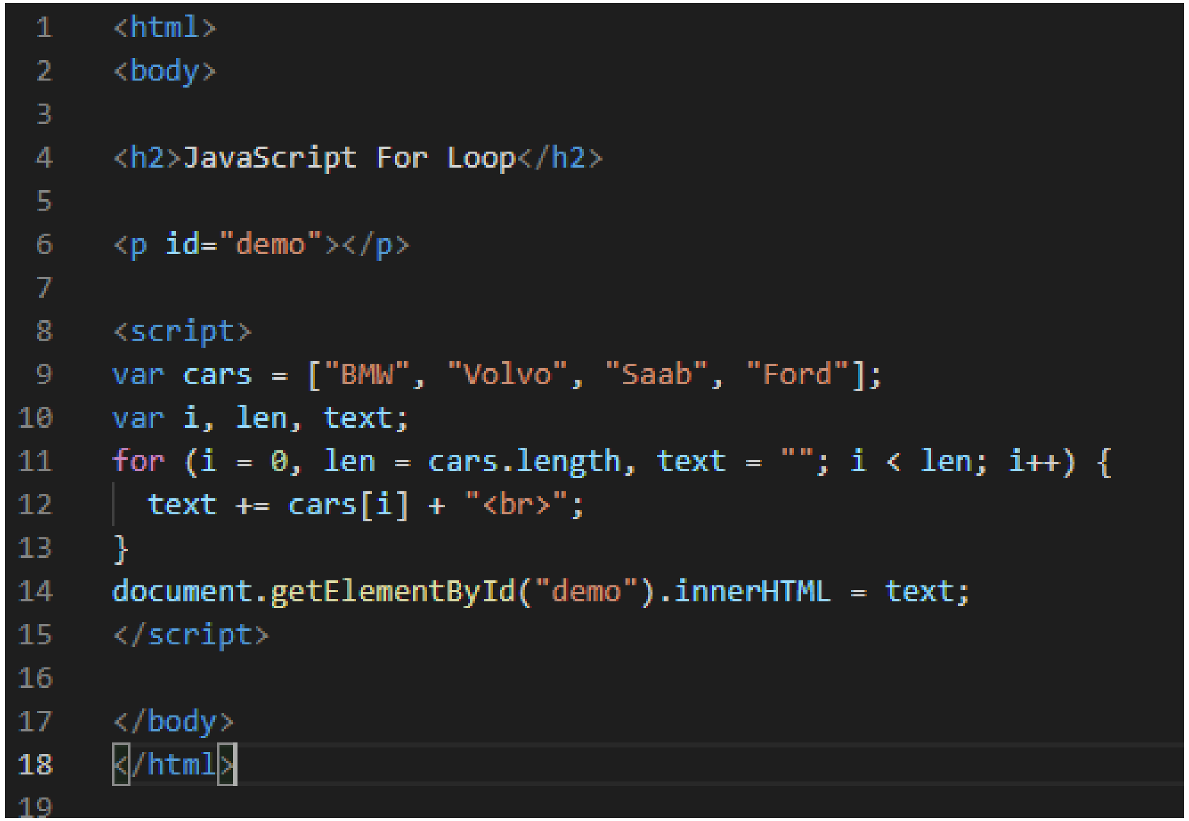


```html
<html>
<body>

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

<script>
var day;
switch (new Date().getDay()) {
  case 0:
    day = "Sunday";
    break;
  case 1:
    day = "Monday";
    break;
  case 2:
    day = "Tuesday";
    break;
  case 3:
    day = "Wednesday";
    break;
  case 4:
    day = "Thursday";
    break;
  case 5:
    day = "Friday";
    break;
  case  6:
    day = "Saturday";
}
document.getElementById("demo").innerHTML = "Today is " + day;
</script>

</body>
</html>
```

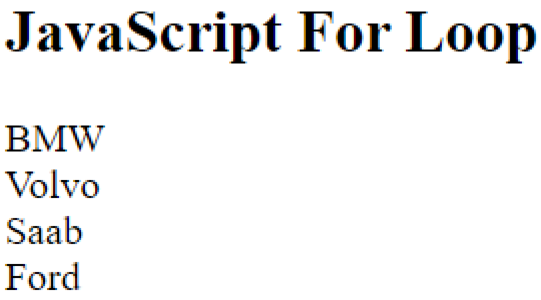
Today is Sunday

```javascript
switch (new Date().getDay()) {
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the Weekend";
}
```

For Loop

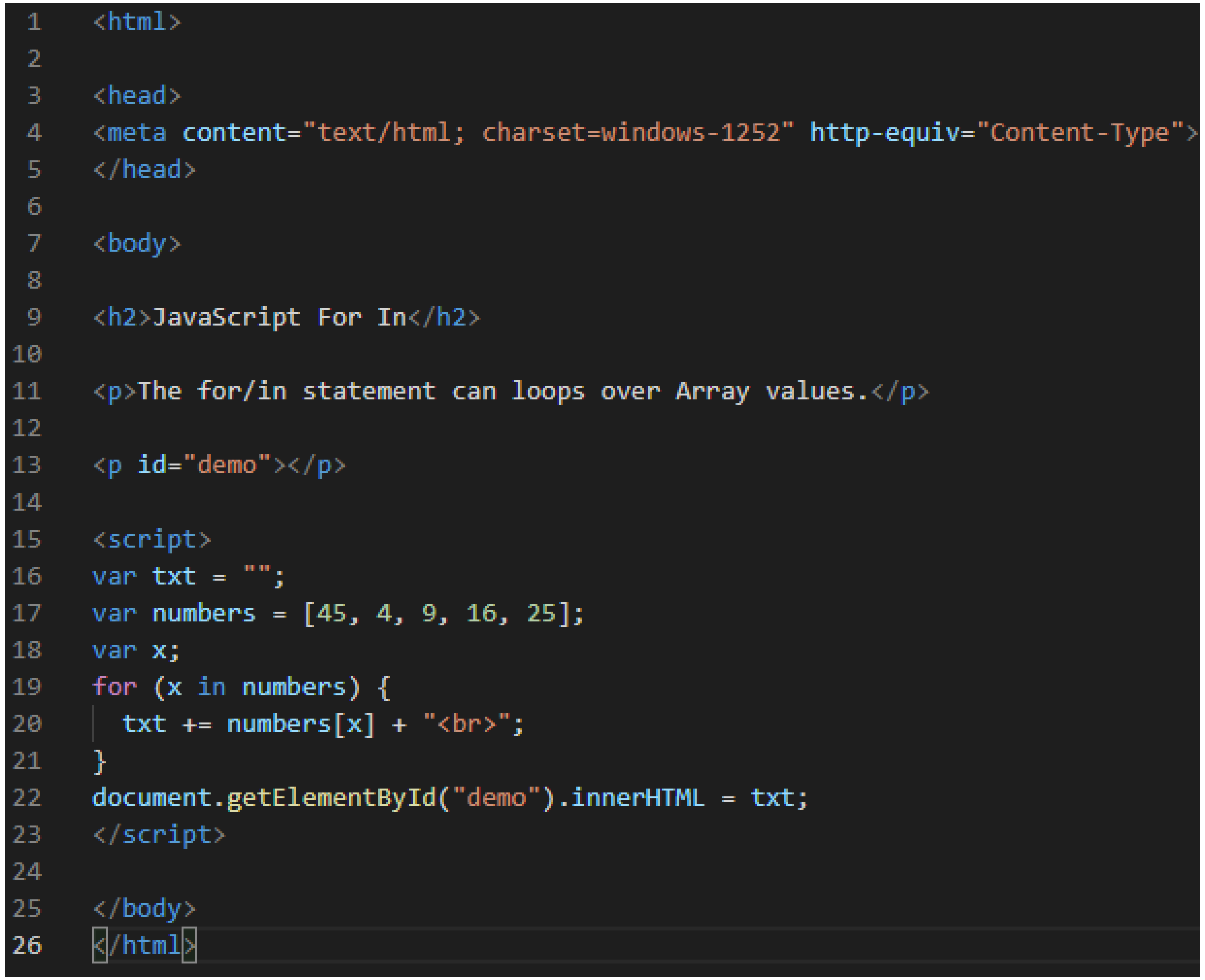


```
<html>
<body>

<h2>JavaScript For Loop</h2>

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

<script>
var cars = ["BMW", "Volvo", "Saab", "Ford"];
var i, len, text;
for (i = 0, len = cars.length, text = ""; i < len; i++) {
  text += cars[i] + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

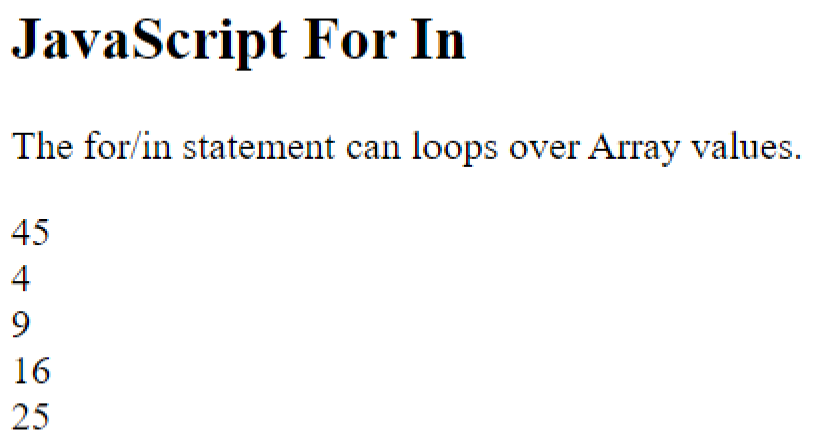


```
JavaScript For Loop

BMW
Volvo
Saab
Ford
```

For In



```
<html>

<head>
<meta content="text/html; charset=windows-1252" http-equiv="Content-Type">
</head>

<body>

<h2>JavaScript For In</h2>

<p>The for/in statement can loops over Array values.</p>

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

<script>
var txt = "";
var numbers = [45, 4, 9, 16, 25];
var x;
for (x in numbers) {
  txt += numbers[x] + "<br>";
}
document.getElementById("demo").innerHTML = txt;
</script>

</body>
</html>
```

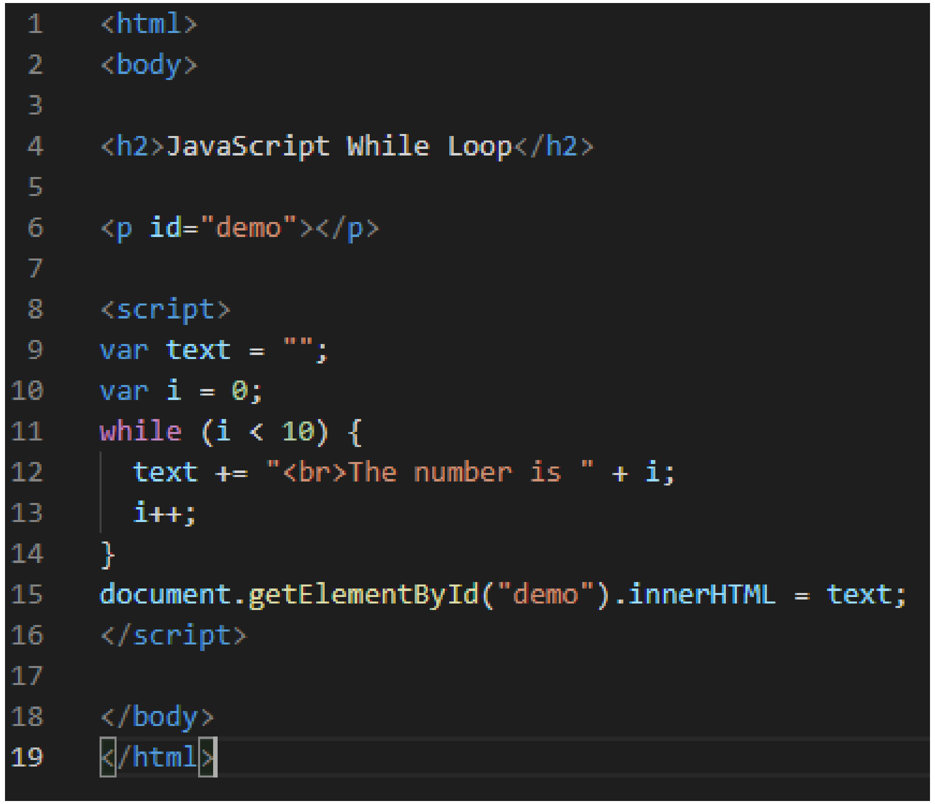


JavaScript For In

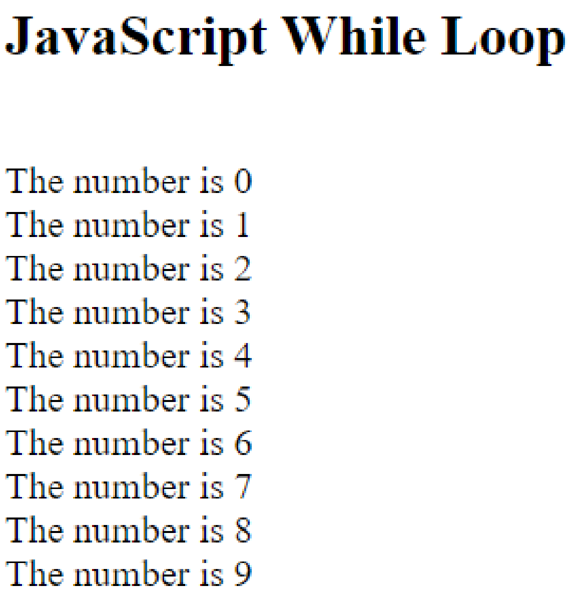
The for/in statement can loops over Array values.

```
45
4
9
16
25
```

While Loop



```
<html>
<body>

<h2>JavaScript While Loop</h2>

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

<script>
var text = "";
var i = 0;
while (i < 10) {
  text += "<br>The number is " + i;
  i++;
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```



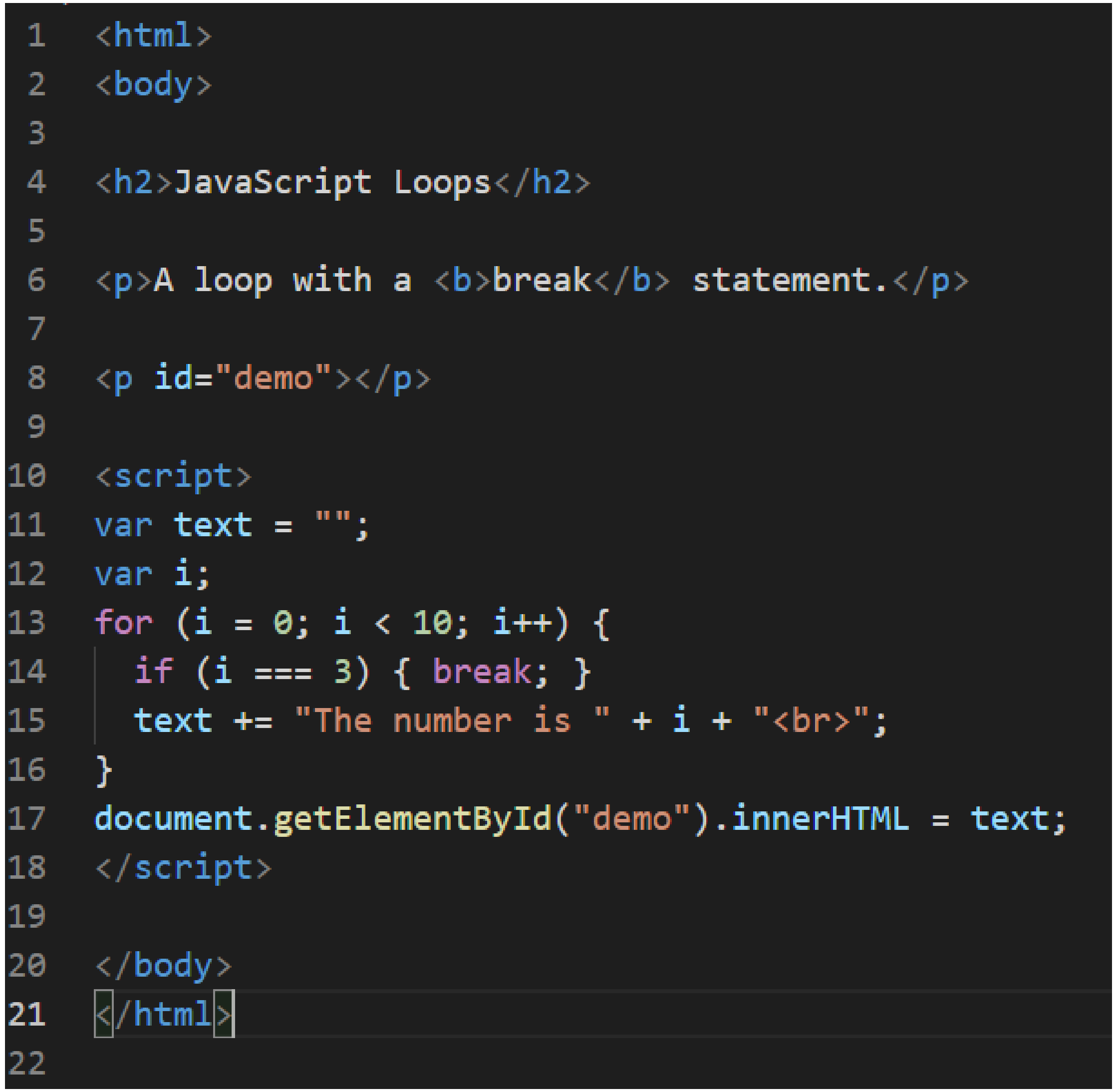
JavaScript While Loop

```
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
```

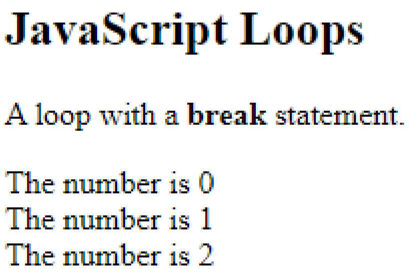
Do While -> Sempre executa pelo menos uma vez.

```
do {
  text += "The number is " + i;
  i++;
}
while (i < 10);
```

Break



```
<html>
<body>

<h2>JavaScript Loops</h2>

<p>A loop with a <b>break</b> statement.</p>

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

<script>
var text = "";
var i;
for (i = 0; i < 10; i++) {
  if (i === 3) { break; }
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>

```

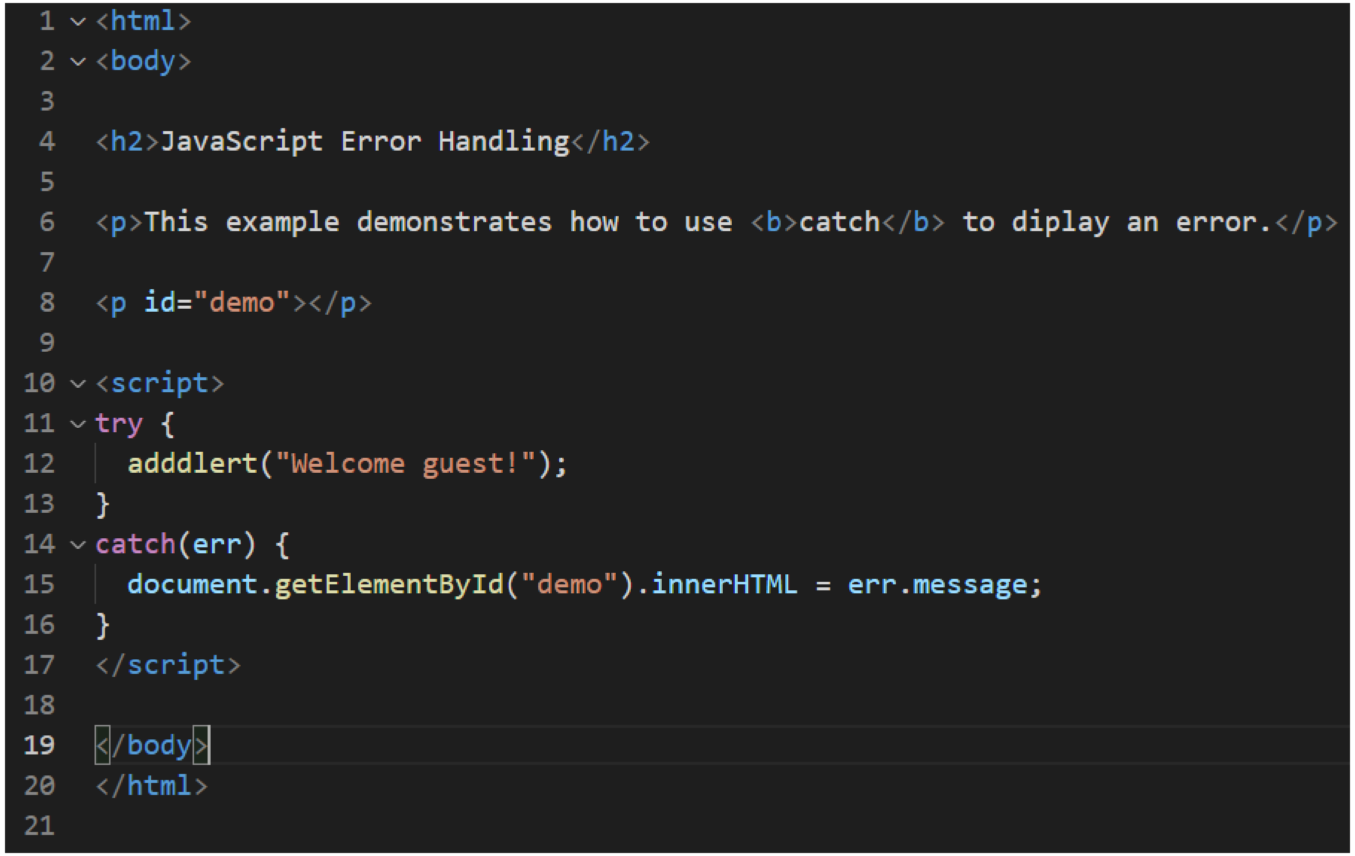


JavaScript Loops

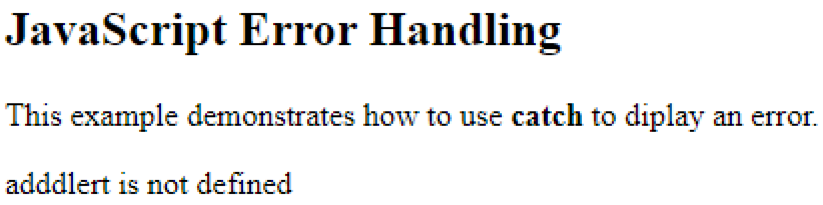
A loop with a **break** statement.

```
The number is 0
The number is 1
The number is 2
```

Tratamento de erros



```
<html>
<body>

<h2>JavaScript Error Handling</h2>

<p>This example demonstrates how to use <b>catch</b> to diplay an error.</p>

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

<script>
try {
  adddlert("Welcome guest!");
}
catch(err) {
  document.getElementById("demo").innerHTML = err.message;
}
</script>

</body>
</html>

```



JavaScript Error Handling

This example demonstrates how to use **catch** to diplay an error.

adddlert is not defined

CEP x Buscar dados

- Desenvolvimento