



LINGUAGEM JAVASCRIPT

SUMÁRIO

Introdução

- ◆ Contexto histórico da linguagem
- ◆ Vantagens da sua utilização

Desenvolvimento

- ◆ Variáveis
- ◆ Tipos de dados
- ◆ Estruturas de seleção
- ◆ Estruturas de repetição
- ◆ Eventos
- ◆ Funções
- ◆ Arrays
- ◆ Exercício exemplificativo

INTRODUÇÃO

Contexto histórico da linguagem

- ◆ Desenvolvido por Brendan Eich na Netscape.
- ◆ Teve vários nomes antes de ser conhecido como JavaScript em 1995.

INTRODUÇÃO

Vantagens da sua utilização

- ◆ *Suporta interatividade do utilizador*
- ◆ *Suporta imagens como links*
- ◆ *Suporta validações do lado do cliente*
- ◆ *Suporta identificação do navegador*
- ◆ *Suporta deteção de plug-ins*
- ◆ *Suporta simplificação de programas*

DESENVOLVIMENTO

Variáveis

Em JavaScript, uma variável contém um valor, como "hello" ou 5. Quando se usa a variável, faz-se referência aos dados que ela representa.

Usamos variáveis para armazenar, recuperar e manipular valores que aparecem no código. Devem-se dar nomes significativos às variáveis, para que as outras pessoas possam compreender melhor a finalidade do código.

Devem-se declarar as variáveis antes de as usar. Para fazer isso, usa-se a palavra-chave **var**.

Exs.:

```
<script>  
var num = 1          // Inteiro  
var dia = "segunda" // String  
</script>
```

DESENVOLVIMENTO

Tipos de dados

Em JavaScript, os tipos de dados podem ser divididos em dois grupos:

◆ Primitivos

- ★ Strings
- ★ Numbers
- ★ Booleans
- ★ Undefined

◆ Complexos

- ★ Objects

DESENVOLVIMENTO

Tipos de dados - Primitivos

◆ Strings

Uma string é uma cadeia de caracteres como “Jonh Doe”.
As strings escrevem-se entre aspas. Podem-se usar plicas ou aspas.

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

<h2>JavaScript Strings</h2>

<p>Strings are written with quotes. You can use single or double quotes:</p>

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

<script>
var carName1 = "Volvo XC60";
var carName2 = 'Volvo XC60';

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

</body>
</html>
```

JavaScript Strings

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

Volvo XC60
Volvo XC60

DESENVOLVIMENTO

Tipos de dados - Primitivos

◆ Numbers

O JavaScript tem apenas um tipo de números.
Os números podem ser escritos com ou sem casas decimais.

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

<h2>JavaScript Numbers</h2>

<p>Numbers can be written with, or without decimals:</p>

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

<script>
var x1 = 34.00;
var x2 = 34;
var x3 = 3.14;

document.getElementById("demo").innerHTML =
x1 + "<br>" + x2 + "<br>" + x3;
</script>

</body>
</html>
```

JavaScript Numbers

Numbers can be written with, or without decimals:

34
34
3.14

DESENVOLVIMENTO

Tipos de dados - Primitivos

◆ Booleans

Os Booleans só podem ter dois valores: verdadeiro ou falso.

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

<h2>JavaScript Booleans</h2>

<p>Booleans can have two values: true or false:</p>

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

<script>
var x = 5;
var y = 5;
var z = 6;
document.getElementById("demo").innerHTML =
(x == y) + "<br>" + (x == z);
</script>

</body>
</html>
```

JavaScript Booleans

Booleans can have two values: true or false:

true
false

DESENVOLVIMENTO

Tipos de dados - Primitivos

◆ Undefined

Em JavaScript, uma variável sem um valor, tem o valor **indefinido** . O tipo também é **indefinido** .

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

<h2>JavaScript</h2>

<p>The value (and the data type) of a variable with no value is
<b>undefined</b>.</p>

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

<script>
var car;
document.getElementById("demo").innerHTML =
car + "<br>" + typeof car;
</script>

</body>
</html>
```

JavaScript

The value (and the data type) of a variable with no value is **undefined**.

undefined
undefined

DESENVOLVIMENTO

Tipos de dados - Complexos

◆ Objects

Os *objects* JavaScript são escritos com chaves.

As propriedades do *object* são escritas como nome: pares de valores, separados por vírgulas.

```
<!DOCTYPE html>
<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.

O *object* (pessoa) no exemplo acima tem 4 propriedades: nome, apelido, idade e cor dos olhos.

DESENVOLVIMENTO

Estruturas de seleção

◆ if

Usa-se o **if** para especificar um bloco de código JavaScript a ser executado se a condição for verdadeira.

```
if (condition) {
    block of code to be executed if the condition is true
}
```

DESENVOLVIMENTO

Estruturas de seleção

◆ if

Exemplo:

```
<!DOCTYPE html>
<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!

DESENVOLVIMENTO

Estruturas de seleção

◆ if ... else

Usa-se o **if ... else** para especificar um bloco de código a ser executado se a condição for falsa.

```
if (condition) {
    block of code to be executed if the condition is true
} else {
    block of code to be executed if the condition is false
}
```

DESENVOLVIMENTO

Estruturas de seleção

◆ if ... else

Exemplo:

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

<p>Click the button to display a time-based greeting:</p>

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

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

<script>
function myFunction() {
  var hour = new Date().getHours();
  var greeting;
  if (hour < 18) {
    greeting = "Good day";
  } else {
    greeting = "Good evening";
  }
  document.getElementById("demo").innerHTML = greeting;
}
</script>

</body>
</html>
```

Click the button to display a time-based greeting:

Try it

Good day

DESENVOLVIMENTO

Estruturas de seleção

◆ else ... if

Usa-se o **else ... if** para especificar uma nova condição, se a primeira condição é falsa.

```
if (condition1) {
  block of code to be executed if condition1 is true
} else if (condition2) {
  block of code to be executed if the condition1 is false and condition2 is true
} else {
  block of code to be executed if the condition1 is false and condition2 is false
}
```


DESENVOLVIMENTO

Estruturas de seleção

◆ else ... if

Exemplo:

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

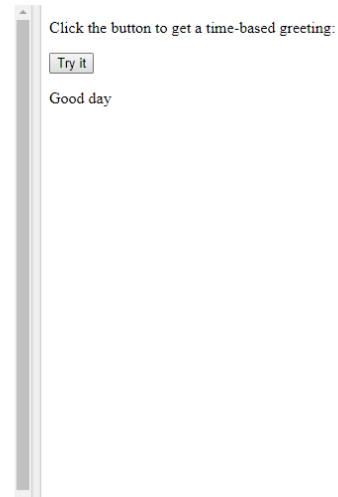
<p>Click the button to get a time-based greeting:</p>

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

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

<script>
function myFunction() {
  var greeting;
  var time = new Date().getHours();
  if (time < 10) {
    greeting = "Good morning";
  } else if (time < 20) {
    greeting = "Good day";
  } else {
    greeting = "Good evening";
  }
  document.getElementById("demo").innerHTML = greeting;
}
</script>

</body>
</html>
```



DESENVOLVIMENTO

Estruturas de seleção

◆ switch

Usa-se o **switch** para selecionar um dos muitos blocos de código a ser executado.

```
switch(expression) {
  case n:
    code block
    break;
  case n:
    code block
    break;
  default:
    code block
}
```

Como funciona:

- A expressão chave é avaliada uma vez.
- O valor da expressão é comparado com os valores de cada caso.
- Se houver uma correspondência, o bloco associado de código é executado.

DESENVOLVIMENTO

Estruturas de seleção

◆ switch

Exemplo:

```
<!DOCTYPE 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>
```



DESENVOLVIMENTO

Estruturas de repetição

◆ while

Enquanto uma condição específica for verdadeira, o bloco de código vai ser executado.

```
while (condition) {
  code block to be executed
}
```

DESENVOLVIMENTO

Estruturas de repetição

◆ while

Exemplo:

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

<h2>JavaScript while</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

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

DESENVOLVIMENTO

Estruturas de repetição

◆ do ... while

O bloco de código é executado uma vez antes de se verificar se a condição é verdadeira. Este ciclo vai-se repetir enquanto a condição for verdadeira.

```
do {
    code block to be executed
}
while (condition);
```

DESENVOLVIMENTO

Estruturas de repetição

◆ do ... while

Exemplo:

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

<h2>JavaScript do ... while</h2>

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

<script>
var text = ""
var i = 0;

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

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

</body>
</html>
```

JavaScript do ... while

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

DESENVOLVIMENTO

Estruturas de repetição

◆ for

O **for** é uma das ferramentas utilizadas para criar ciclos.

```
for (statement 1; statement 2; statement 3) {
  code block to be executed
}
```

DESENVOLVIMENTO

Estruturas de repetição

◆ for

Exemplo:

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

<h2>JavaScript Loops</h2>

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

<script>
var text = "";
var i;
for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

JavaScript Loops

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4

DESENVOLVIMENTO

Eventos

Eventos HTML são "**coisas**" que acontecem a elementos HTML.
Quando o JavaScript é usado em páginas HTML, pode "**reagir**" sobre esses eventos.

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

DESENVOLVIMENTO

Funções

Uma função JavaScript é um bloco de código projetado para executar uma tarefa específica.

Uma função JavaScript é executada quando "algo" a invoca (chama).

```
<!DOCTYPE html>
<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>
function myFunction(p1, p2) {
  return p1 * p2;
}
document.getElementById("demo").innerHTML = myFunction(4, 3);
</script>

</body>
</html>
```

JavaScript Functions

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

12

DESENVOLVIMENTO

Arrays

Um **array** é uma variável especial, que pode conter mais de um valor de cada vez. Os valores destas variáveis são escritos entre aspas, separados por vírgulas e dentro de parênteses retos.

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

<h2>JavaScript Arrays</h2>

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

<script>
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>
```

JavaScript Arrays

Saab, Volvo, BMW

DESENVOLVIMENTO

Exercício (if ... else):

```
if-else_JavaScript.html
1  <!DOCTYPE html>
2  <html lang="en" dir="ltr">
3    <head>
4      <meta charset="utf-8">
5      <title></title>
6    </head>
7    <body>
8
9      <p>Apresenta "Good Day", se a hora for inferior a 20:00:</p>
10
11     <button onclick="apresenta()">Try it</button>
12
13     <p id="hora"></p>
14
15     <script>
16     function apresenta(){
17       if (new Date().getHours() < 20)
18       {
19         document.getElementById("hora").innerHTML = "Good Day";
20       }
21       else
22       {
23         document.getElementById("hora").innerHTML = "Good Evening";
24       }
25     }
26     </script>
27
28   </body>
29 </html>
30
```

Apresenta "Good Day", se a hora for inferior a 20:00:

Try it

Good Evening