

# 06

Introdução às  
Tecnologias Web

## JavaScript (1/3)

The JavaScript logo, consisting of the letters 'JS' in a bold, black, sans-serif font, centered on a bright yellow square background.

### Perguntas

**Padding e Margin?**

**Como centro uma “caixa”?**

**Posicionamento relativo vs absoluto?**

**Posicionamento flutuante?**

**Como centro uma imagem?**

## Qual o significado?

```
1 p {  
2   padding: 10px;  
3   border-bottom-color: red;  
4   border-bottom-style: solid;  
5 }  
6 img.cc {  
7   display: block;  
8   margin: 2px auto;  
9   border: 10px dashed blue;  
10 }  
11 p.top {  
12   position: fixed;  
13   top: 10px;  
14   left: 0px;  
15 }  
16
```

## Qual o significado?

```
1 p {  
2   padding: 10px;  
3   border-bottom-color: red; ←  
4   border-bottom-style: solid;  
5 }  
6 img.cc {  
7   display: block; ←  
8   margin: 2px auto;  
9   border: 10px dashed blue;  
10 }  
11 p.top {  
12   position: fixed;  
13   top: 10px;  
14   left: 0px; ←  
15 }  
16
```

## Resumo Aula Anterior

### **Caixas e Bordas**

margin, padding, border  
top, right, bottom, left

### **Posicionamento**

Normal, relativo, absoluto, fixo, flutuante

### **Tamanhos de Ecrãs**

Design Responsive

### **Formatação de Imagens**

Tamanho, alinhamento, fundos

## Sumário

### **Introdução ao JavaScript**

### **Conceitos Nucleares**

### **Tipos, Valores e Variáveis**

### **Decisões e Funções**

# 01

## INTRODUÇÃO AO JAVASCRIPT

### JavaScript

**Linguagem de programação dinâmica**

**Permite executar scripts no cliente (browser)**

**Torna as páginas HTML interativas**

HTML – Conteúdo

CSS – Apresentação

**JavaScript - Comportamento**

## JavaScript

### **JavaScript NÃO É Java**

Nome foi escolhido para aproveitar a fama do Java

### **Linguagem Baseada em objectos**

(mas não em classes)

### **Independente da plataforma**

Windows, MacOS X, Linux, Android, iOS, etc.

## Permite Alterar

### **Conteúdo HTML**

### **Atributos dos elementos HTML**

### **Estilo de um elemento HTML (CSS)**

### **Validar dados (de um formulário)**

### **Reagir a eventos**

## Como inserir numa página HTML?

Colocando o código no **<head>**

Colocando o código no **<body>**

Ficheiro externo **.js**

Ou uma **combinação** dos três

## Código no <head>

```
<head>  
<script>  
  // código JavaScript  
</script>  
</head>  
<body>  
<h1>Heading H1</h1>  
</body>
```

Código carregado antes do conteúdo do body

## Código no <body>

```
<body>
<h1>Heading H1</h1>
<script>
  // código JavaScript
</script>
<h2>Heading H2</h2>
</body>
```

Código carregado em conjunto com o conteúdo do body

## Ficheiro .js externo

```
<head>
<script src="codigo.js"></script>
</head>
```

## Vantagens ficheiro externo

### Facilidade na manutenção

Código apenas num local  
Facilita edição ou correção do código

### Carregamento mais rápido da página

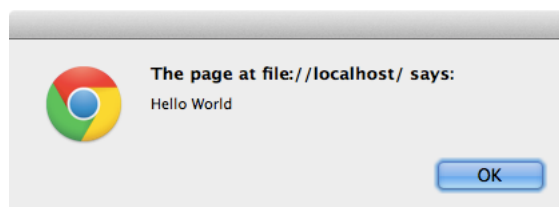
Ficheiro .js armazenado na cache  
Evita-se carregamento cada vez que a página é chamada

### Melhor organização

Separa **comportamento** (JavaScript)  
do **conteúdo** (HTML) da **apresentação** (CSS)

## Hello World em JavaScript

```
<head>
<script>
    alert("Hello World");
</script>
</head>
<body>
<h1>Heading H1</h1>
</body>
```





# 02

## CONCEITOS NUCLEARES

### Instruções em JavaScript

**Valores**

**Operadores**

**Expressões**

**Palavras Reservadas (*Keywords*)**

**Comentários**

# Valores

## Valores Fixos - Literais

Valores que aparecem diretamente no programa

Exemplos:

```
12           // o número 12
1.2 + 3      // a soma de dois números
"hello world" // A string de texto
'Hi'         // Outra string
true         // o valor booleano true
false        // O valor booleano false
```

## Valores Variáveis – Variáveis

```
var x;
var contador;
```

# Identificadores

## Identificadores = Nomes

Nomes de **variáveis** e de **funções**

## Nomes devem começar por **letra**, **\_** ou **\$**

Seguidos de letras, dígitos, **\_** ou **\$**

## Exemplos de identificadores válidos

```
i
my_var_name
v123
_tmp
$myStr
```

## Variáveis

### São Contentores para guardar valores

```
var x = 5;           // x guarda o valor 5
var y = 6;           // y guarda o valor 6
var z = x + y;       // z guarda o valor 11
```

### Declaradas (criadas) usando var

### Não têm tipo pré-definido

### Devem ter nomes únicos

### Podem ser usadas em expressões

## Operadores JavaScript

### Aritméticos

### Atribuição

### Strings (e números)

### Comparação

### Lógicos

## Operadores Aritméticos

| Operador | Nome          | Descrição                                |
|----------|---------------|--|
| +        | Soma          | Igual aos operadores matemáticos         |
| -        | Subtração     |  |
| *        | Multiplicação |  |
| /        | Divisão       |  |
| %        | Módulo        | Resto da divisão inteira (ex. 12%5 -> 2) |
| ++       | Incremento    | x++                                      |
| --       | Decremento    | x--                                      |

## Operadores de Atribuição

| Operador | Exemplo | Equivalente |
|----------|---------|-------------|
| =        | x = y;  | x = y;      |
| +=       | x += y; | x = x + y;  |
| -=       | x -= y; | x = x - y;  |
| *=       | x *= y; | x = x * y;  |
| /=       | x /= y; | x = x / y;  |
| %=       | x %= y; | x = x % y;  |

## Operadores de *Strings*

### + Junta duas strings (concatenação)

```
var firstName = "Manuel ";  
var lastName = "Fonseca";  
var fullName = firstName + lastName  
-> Manuel Fonseca
```

### + Junta uma string e um número

```
var firstNum = "007";  
var lastNum = 123;  
var z = firstNum + lastNum;  
-> 007123
```

## Operadores de Comparação

| Operador | Descrição                | Comparação                   | Resultado             |
|----------|--------------------------|------------------------------|-----------------------|
| ==       | igual a                  | x == 8<br>x == 5<br>x == "5" | false<br>true<br>true |
| ===      | valor igual e tipo igual | x === "5"<br>x === 5         | false<br>true         |
| !=       | diferente                | x != 8<br>x != "5"           | true<br>false         |
| !==      | tipo e valor diferentes  | x !== "5"<br>x !== 5         | true<br>false         |
| >        | maior                    | x > 8                        | false                 |
| <        | menor                    | x < 8                        | true                  |
| >=       | maior ou igual           | x >= 8                       | false                 |
| <=       | menor ou igual           | x <= 8                       | true                  |

Consideremos x = 5

## Operadores Lógicos

| Operador | Descrição | Exemplo              | Resultado |
|----------|-----------|----------------------|-----------|
| &&       | and       | (x < 10 && y > 1)    | true      |
|          | or        | (x === 5    y === 5) | false     |
| !        | not       | !(x === y)           | true      |

Consideremos x = 6 e y = 3

## Expressões

**Sequências válidas de operadores e operandos**

**Depois de avaliadas geram um valor primitivo ou objeto**

**Existem expressões:**

**Aritméticas** – geram um número

**Lógicas** – geram um booleano (true ou false)

**String** – geram uma string

**Especiais** – podem gerar objetos

## Palavras Reservadas

|          |         |            |           |              |
|----------|---------|------------|-----------|--------------|
| abstract | double  | goto       | native    | static       |
| boolean  | enum    | implements | package   | super        |
| byte     | export  | import     | private   | synchronized |
| char     | extends | int        | protected | throws       |
| class    | final   | interface  | public    | transient    |
| const    | float   | long       | short     | volatile     |

**Identificam ações a realizar**

**Não podem ser usadas como identificadores**

**Existem mais nomes não permitidos**

Variáveis globais pré-definidas

Funções

## Comentários JavaScript

**Dois tipos de comentários**

**Apenas uma linha**

`// Comentário JavaScript de uma linha`

**Múltiplas linhas**

`/* Comentário JavaScript que  
ocupa mais de uma linha */`

## Recomendações

### Case sensitive

teste diferente de Teste diferente de TESTE

### Nomes de Variáveis

*Camel Case:* firstName, lastName, myCounter

### Colocar ; no final de cada linha

Considerado boa prática de programação

Evita alguns erros

## Usar modo *Strict*

### Obriga a programar “corretamente”

### Evita problemas de programação

Enorme liberdade do JavaScript

Falta de verificação de algumas instruções

### Torna mais fácil fazer JS seguro

### Como usar?

Primeira linha de código JS

**"use strict";**

[http://www.w3schools.com/js/js\\_strict.asp](http://www.w3schools.com/js/js_strict.asp)



# 03

## TIPOS, VALORES E VARIÁVEIS

### Tipos de Dados

#### Tipos Primitivos

- Números
- Strings
- Booleanos

#### Tipos Especiais

- null e undefined
- Só têm um valor, igual ao tipo

#### Tipo Objeto

- O que não for primitivo ou especial
- Objeto = coleção de propriedades

## Tipos de Dados Primitivos

### Valores Numéricos

Só existe um tipo de números

ex. 234, 45.76, 3.1e34

NaN, Infinity, -Infinity

isFinite(x)

### Valores Booleanos

true e false

### Cadeias de caracteres (Strings)

"Isto é uma string."

'Isto é outra.'

## Tipos de Dados Especiais

### undefined

Valor possível: **undefined**

Conteúdo de variáveis não inicializadas  
(ex. var a;)

### null

Valor possível: **null**

Inexistência de valor associado à variável  
(ex: var a=null;)

## Valores Booleanos

**Tomam o valor true ou false**

**Qualquer valor pode ser convertido num booleano**

false (*falsy*): undefined, null, 0, NaN, ""

true (*truthy*): o resto

**Logo (são iguais):**

if (obj !== null) ...

if (obj) ...

## Scope das variáveis

### Locais

Criadas dentro das funções

Válidas apenas dentro da função onde foram criadas

### Globais

Criadas fora das funções

Válidas em todo o lado, incluindo dentro das funções

## Scope: Exemplo

Variáveis:  
**marca**

```
var marca;
```

Variáveis:  
**marca** e **matricula**

```
function regista () {
```

```
    var matricula;  
    matricula = "AA-00-00";
```

```
}
```

Variáveis:  
**marca**

## Sequências de Escape (\)

### O carácter \ tem um significado especial em JS

É usado em conjunto com o carácter seguinte

**\n** – nova linha

**\'** – apostrofe

**\t** – tab

**\uXXXX** – carácter unicode especificado

**\u03c0** (carácter Pi)

# 04

## ESTRUTURAS DE DECISÃO E FUNÇÕES

**if**

**Permite tomar decisões**

**Executa código de acordo com o resultado de  
uma expressão lógica**

```
if (expressão) {  
    // código a executar  
}
```

**Se expressão é *truthy*, executa o código**

## if else

Introduz um novo conjunto de código

```
if (expressão) {  
    // código 1  
}  
else {  
    // código 2  
}
```

Se expressão é *truthy*, executa o código 1

Se expressão é *falsy*, executa o código 2

## else if

Quando fazemos vários ifs encadeados

```
if (n === 1) {  
    // código 1  
}  
else if (n === 2){  
    // código 2  
}  
else if (n === 3){  
    // código 3  
}  
else {  
    // código 4  
}
```

## switch

```
switch (n) {  
  case 1: // executa este código se n==1  
    // código 1  
    break;  
  case 2: // executa este código se n==2  
    // código 2  
    break;  
  case 3: // executa este código se n==3  
    // código 3  
    break;  
  default: // executa este código se tudo o  
    resto falhar  
    // código 4  
    break;  
}
```

## Funções

Blocos de código para realizar uma tarefa específica

Código SÓ é executado quando a função é invocada

Código coloca-se entre { }

Função tem um nome

Pode ter parâmetros

```
function quadrado(x) {  
  return x*x; }
```

## Funções

### **Permitem reutilizar código**

Definida uma vez, usada várias

### **Parâmetros permitem usar a função com vários valores**

```
quadrado (3);  
quadrado (10);
```

### **Podem ser usadas em expressões**

```
var idade = 18;  
var z = idade + quadrado(idade);
```

# 05

## RESUMINDO



## Resumo

### **Introdução ao JavaScript**

Define comportamento

### **Conceitos Nucleares**

Operadores, expressões, comentários

### **Tipos, Valores e Variáveis**

Primitivos, scope das variáveis

### **Decisões e Funções**

if else, switch, function

## Próxima Aula



JS

