MÓDULO 4: JAVASCRIPT

INTRODUÇÃO

FACULDADE ENGENHARIA
Departamento de Informática

JAVASCRIPT

Tecnologia web do lado do cliente (client-side scripting);

É uma linguagem de programação usada para programar o comportamento das páginas *web*, para as tornar dinâmicas

Com Javascript podemos:

- Mudar o conteúdo de elementos HTML
- Mudar os atributos de elementos HTML
- Mudar os estilos dos elementos HTML (propriedades CSS)
- Esconder/Mostrar elementos HTML

Não precisa de compilação, corre em qualquer *browser*

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

A TAG <SCRIPT>

O código Javascript pode ser incluido no `<head>` e/ou `<body>` dos documentos HTML

Tem de ser inserido entre as tags `<script>` e `</script>`

Alternativamente, pode ser escrito em ficheiros externos e referenciado no ficheiro HTML em questão

- *À semelhança do CSS (`<style>` e `<link>`)*
- *Só no Javascript fazemos tudo com o `<script>`*

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

JAVASCRIPT INTERNO (NO HEAD)

```
<!DOCTYPE html>
<html>
  <head>

    <script>
        function mudaTexto() {
            document.getElementById("paragrafo1").innerHTML = “Escrevi outro texto.";
        }
    </script>

  </head>

  <body>
    <h1>A minha página</h1>
    <p id=“paragrafo1">Um texto qualquer.</p>
    <button type="button" onclick=“mudaTexto()">Mudar o texto! </button>

  </body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

JAVASCRIPT INTERNO (NO BODY)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo</title>
  </head>

  <body>

    <h1>A minha página</h1>
    <p id=“paragrafo1">Um texto qualquer.</p>
    <button type="button" onclick=“mudaTexto()">Mudar o texto! </button>

    <script>
        function mudaTexto() {
           document.getElementById("paragrafo1").innerHTML = “Escrevi outro texto.";
        }
    </script>

  </body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

JAVASCRIPT INTERNO (NO PRÓPRIO ELEMENTO HTML)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo</title>
  </head>

  <body>

    <h1>A minha página</h1>
    <p id=“paragrafo1">Um texto qualquer.</p>
    <button type="button" onclick=“document.getElementById(‘paragrafo1’).innerHTML = ‘Escrevi
outro texto’;”>Mudar o texto! </button>

  </body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

JAVASCRIPT EXTERNO

No documento HTML o ficheiro javascript é incluido dentro de <script> e </script>

```
<!DOCTYPE html>
<html>
  <body>
    <script src="ficheiro.js"></script>
</body>
</html>
```

ficheiro.js

```
function mudaTexto() {
   document.getElementById("paragrafo1").innerHTML = "Escrevi outro texto.";
}
```

O ficheiro externo deve gravar-se com extensão .js

No ficheiro javascript externo não é necessário colocar as tags <script> e </script> a envolver o código

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

OUTPUT E COMENTÁRIOS

FACULDADE ENGENHARIA
Departamento de Informática

OUTPUT

```
window.alert()
```

```
document.write()
```

```
innerHTML
```

```
console.log()
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

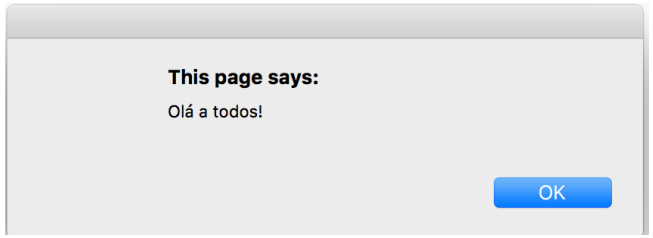
window.alert()

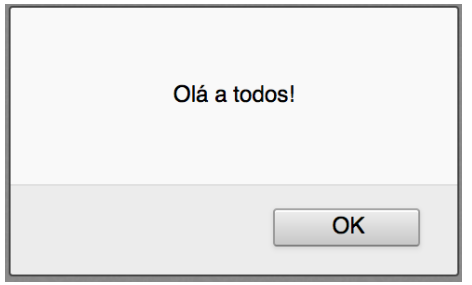
```
<!DOCTYPE html>
<html>
  <body>

    <h1>A minha página</h1>

    <script>
    window.alert("Olá a todos!");
    </script>

  </body>
</html>
```





O alert() é aliás muito utilizado mostrar mensagens ao utilizador em resposta a certas acções (ex: clicar num botão)

```
<button onclick="alert('Olá a todos!')">Diz olá!</button>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

document.write()

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

    <h1>A minha página</h1>

    <script>
    document.write("Olá  a todos!");
    </script>

  </body>
</html>
```

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

innerHTML

É possível escrever especificamente num elemento HTML

- É necessário identificar esse elemento com um `id`
- Com Javascript, *apanhamos* esse elemento pelo `id`: **`document.getElementById(id_do_elemento)`**
- O **`innerHTML`** define o conteúdo desse elemento

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

    <h1 id="titulo"></h1>

    <script>
         document.getElementById("titulo").innerHTML = "A minha página fantástica!";
    </script>

  </body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

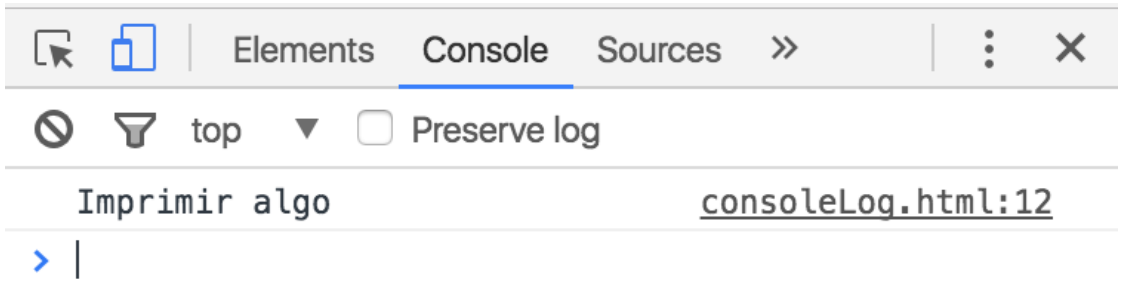
console.log()

Não se vê nada na página, temos de ir à consola:

No Chrome: View - Developer - Javascript Console

Imprime para a consola (bom para fazer debugging)

```
<!DOCTYPE html>
<html>
  <body>
    <script>
      console.log("Imprimir algo");
    </script>
  </body>
</html>
```



SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

COMENTÁRIOS

```
// Comentário de apenas uma linha


/* Comentário de várias linhas.
Muito útil quando queremos adicionar
muita informação nos comentários */
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

SINTAXE E PROGRAMAÇÃO BÁSICA

FACULDADE ENGENHARIA
Departamento de Informática

VARIÁVEIS

Declaradas usando a palavra reservada opcional `var`

```
var x;
var y, z;
var a=2, b=2;
x=5;
var c=a+b;
```

O Javascript é case-sensitive portanto `x ≠ X; last_name ≠ lastName`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

TIPOS DE VARIÁVEIS

```
var a;                                                    //undefined
a=2;                                                     //Número Inteiro (integer)
var a=2.67;                                              //Número Decimal (float)
var a=123e5;                                            //Notação Científica
var nome="Zé Miguel";                                  //String
var equipas=["Benfica", "Sporting", "Porto"];    //Array
var pessoa={nome:"Miguel", apelido:"Sousa"};    //Objecto
var a=true;                                            //Boolean
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

TIPOS DE VARIÁVEIS

Para saber o tipo de uma variável, usar typeof:

```
typeof 3.14                          // number
                                     (não faz distinção entre integer e
float)
typeof “Manuel”                      // string
typeof false                         // boolean
typeof [1,2,3,4]                     // object
                                     (no JS, os arrays são objects!)
typeof {nome:'Zé', idade:34}         // object
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

VARIÁVEIS

```
// É possível declarar várias variáveis diferentes de uma só vez:
var a = 2, nome = "Zé Miguel", carros = ["Volvo", "Audi"];

// É possível realizar operações aritméticas com variáveis em Javascript
var x = 5 + 2 + 3;  // 10

// É possível "adicionar strings": concatenação;
var x = "Maria" + " " + "Rita";  // "Maria Rita"

// Se um número for posto entre aspas, os seguintes são vistos como
strings e concatenados:
var x = "5" + 2 + 3;  // 523
var x = 2 + 3 + "5";  // 55
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES

- O Javascript define vários tipos de operadores:
  - Operadores aritméticos
  - Operadores de incremento e decremento
  - Operadores de atribuição
  - Operadores relacionais
  - Operadores lógicos
  - Operadores de cadeias (*Strings*)

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES ARITMÉTICOS

Operador	Descrição	Exemplo
+	Adição	`var x = 3, y = 2, z = x + y;  // z=5`
-	Subtração	`var x = 3, y = 2, z = x - y;  // z=1`
*	Multiplicação	`var x = 3, y = 2, z = x * y;  // z=6`
/	Divisão	`var x = 3, y = 2, z = x / y;  // z=1.5`
%	Resto da divisão inteira	`var x = 3, y = 2, z = x % y;  // z=1`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES DE INCREMENTO E DECREMENTO

Operador	Descrição	Exemplo

++	Incremento	`var x = 5;`
		`x++;`
		`var z = x;  // z=6`

--	Decremento	`var x = 5;`
		`x--;`
		`var z = x;  // z=4`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES DE ATRIBUIÇÃO

Operador	Exemplo	Equivale a:
=	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

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES RELACIONAIS

Operador	Descrição	Exemplo
==	Igual a	`2 == 1 //false`
!=	Diferente de	`2 != 1 //true`
>	Maior do que	`2 > 1 //true`
<	Menor do que	`2 < 1 //false`
>=	Maior ou igual que	`2 >= 1 //true`
<=	Menor ou igual que	`2 <= 1 //false`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES LÓGICOS

Operador	Descrição	Exemplo
!	Negação	x = 1 y = 2 !(x==y) //true
&&	Conjunção	x = 1 y = 5 (x<3 && y<3) //false
||	Disjunção	x = 1 y = 5 (x<3 || y<3) //true

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

OPERADORES DE CADEIAS *(STRINGS)*

Operador	Descrição	Exemplo

+	Concatenação	`string1 = “Bom”;`
		`string2 = “ dia”;`
		`str = string1+ string2;  // “Bom dia”`

```
// A soma de um número com uma string devolve sempre uma string
x = 5 + 5;                    // 10 (número)
y = "5" + 5;                  // 55 (string)
z = "Hello" + 5;              // Hello5 (string)
k = 5 + 5 + “Hello";          // 10Hello (string)
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

INSTRUÇÕES CONDICIONAIS

FACULDADE ENGENHARIA
Departamento de Informática

INSTRUÇÕES CONDICIONAIS

Em Javascript definem-se as seguintes instruções condicionais:

- `if` (*condição*) instrução
- `else` instrução
- `else if` (condição) instrução
- `switch - case`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

if

```
if (condição) {
    código a ser executado caso a condição seja verdadeira
}
```

```
if (idade < 5) {
    mensagem = “Ainda és um bebé”;
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

else

```
if (condição) {
    código a ser executado caso a condição seja verdadeira
} else {
    código a ser executado caso a condição seja falsa
}
```

```
if (idade < 5) {
    mensagem = “Ainda é um bebé";
} else {
    mensagem = “Já não és nenhum bebé";
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

else if

```
if (condição1) {
    código a ser executado caso a condição1 seja verdadeira
} else if (condição2){
    código a ser executado caso a condição1 seja falsa e a condição2 seja verdadeira
} else {
    código a ser executado caso ambas as condições sejam falsas
}
```

```
if (idade < 5) {
    mensagem = “Ainda é um bebé";
} else if (idade < 18) {
    mensagem = “És um adolescente";
} else {
    mensagem = “És um adulto!”;
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

switch

```
switch(expressão) {
    case n:
        bloco de código
        break;
    case n:
        bloco de código
        break;
    default:
        bloco de código default

}
```

```
var diaSemana = 1;

switch (diaSemana) {
    case 1:
        dia = "Segunda";
        break;
    case 2:
        day = "Terça";
        break;
    default:
        day = "Quarta";
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

switch

Se o valor da variável não coincidir com nenhum dos case, usa-se o valor default para indicar as instruções que se devem executar.

```
var diaSemana = 5;

switch (diaSemana) {
    case 1:
      msg = “Primeiro dia da semana!”;
      break;
    case 6:
    case 7:
       msg = “Fim de semana!”;
       break;
    default:
        msg = “À espera do fim de semana…”;
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

INSTRUÇÕES ITERATIVAS

FACULDADE ENGENHARIA
Departamento de Informática

INSTRUÇÕES ITERATIVAS

Em Javascript definem-se as seguintes instruções iterativas:

- `for`
- `for/in`
- `while`
- `do-while`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

O CICLO `for`

```
for (start; end; step) {
    código a ser executado
}
```

```
for (i = 1; i < 5; i++) {
    document.write("Número: " + i + "<br>");
}
```

Número: 1
Número: 2
Número: 3
Número: 4

```
var dias = ["Seg", "Ter", "Qua", "Qui", "Sex"];
for (var i = 0; i < 5; i++) {
    alert(dias[i]);
}
```

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

O CICLO `for/in`

Especificamente usado para percorrer as propriedades de um objecto:

```
var pessoa = {nome:"João", apelido:"Santos", idade:28};

for (var prop in pessoa) {
    document.write(pessoa[prop] + "<br>");
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

O CICLO `while`

```
while (condição) {
    código a ser executado
}
```

```
while (i < 5) {
    document.write("Número: " + i + "<br>");
    i++;
}
```

Nota: Não esquecer de incrementar/decrementar/alterar a variável, caso o contrário o loop nunca vai parar! (crasha o browser)

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

O CICLO do-while

```
do {
    código a ser executado
}
while (condição);
```

Número: 6
Número: 7
Número: 8
Número: 9

```
var i = 6;
do {
    document.write("Número: " + i + "<br>");
    i++;
}
while (i < 10);
```

Número: 6

```
var i = 6;
do {
    document.write("Número: " + i + "<br>");
    i++;
}
while (i < 5);
```

No do-while, o loop é sempre executado uma vez, mesmo que a condição seja falsa, porque o código é executado antes da condição ser testada.

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

INSTRUÇÕES BREAK AND CONTINUE

FACULDADE ENGENHARIA
Departamento de Informática

A INSTRUÇÃO `break`

A instrução break interrompe um ciclo passando o controlo à primeira instrução a seguir ao ciclo.

```
for (i = 0; i < 5; i++) {
    if (i == 3) {
       break;
    }
  document.write("Número: " + i + "<br>");
}

document.write("Terminei!");
```

Número: 0
Número: 1
Número: 2
Terminei!

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

A INSTRUÇÃO `continue`

A instrução continue interrompe o ciclo passando para iteração seguinte:

```
for (i = 0; i < 5; i++) {
    if (i == 3) {
       continue;
    }
  document.write("Número: " + i + "<br>");
}

document.write("Terminei!");
```

Número: 0
Número: 1
Número: 2
Número: 4
Terminei!

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

FUNÇÕES

FACULDADE ENGENHARIA
Departamento de Informática

FUNÇÕES

Uma função é uma sequência de instruções com o objectivo de desempenhar uma tarefa específica.

Pode ser invocada em qualquer ponto do programa, e quando é invocada (*chamada*), ela executa essas instruções.

O Javascript tem algumas funções predefinidas, mas nós podemos criar as nossas.

As funções têm argumentos que são passados na chamada da função e usados pela função como variáveis locais.

As funções podem devolver um valor ao programa de chamada mediante a utilização da instrução return.

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

FUNÇÕES PREDEFINIDAS DO JAVASCRIPT (*STRINGS*) (1/2)

length	Número de caracteres	`"Ola".length  //3`
concat()	Concatenação	`"Ola".concat(" pessoas"); //"Ola pessoas"`
toUpperCase()	Maiúsculas	`"Ola".toUpperCase(); //"OLA"`
toLowerCase()	Minúsculas	`"Ola".toLowerCase(); //"ola"`
charAt(i)	Caracter na posição i	`"Ola".charAt(2); //"a"`
indexOf(char)	Posição do caracter char	`"Ola".indexOf('O'); //0` `// Se não existe, devolve -1`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

FUNÇÕES PREDEFINIDAS DO JAVASCRIPT (*STRINGS*) (2/2)

substring(inicio, final)	Extrai string entre os indices início e final	"Palavra".substring(2,4); //la
		// O inicio está incluido
		// O final não

split(separador)	Separa as strings pelo separador e converte-os num array	var frase = "Sou uma frase comprida";
		var palavras = frase.split(" ");
		// palavras = ["Sou", "uma", "frase", "comprida"];

		var palavra = "Ola";
		var letras = palavra.split("");
		// letras = ["O", "l", "a"];

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

FUNÇÕES PREDEFINIDAS DO JAVASCRIPT (*ARRAYS*) (1/2)

length	Número de elementos	`[1,2,3,4].length;` `//4`
concat()	Concatenação	`[1,2,3,4].concat([5,6]);` `//[1,2,3,4,5,6]`

join(separado)	Une elementos por separador, formando uma string	`var array = [“Ola”, “amigos”];`
		`var msg = array.join(“”); //“Olaamigos”`
		`var msg = array.join(“ ”); //“Ola amigos”`

pop()	Retira o último elemento do array e devolve-o	`var array = [1,2,3,4];`
		`var ultimo = array.pop();`
		`//ultimo = 3, array = [1,2,3]`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

FUNÇÕES PREDEFINIDAS DO JAVASCRIPT (*ARRAYS*) (2/2)

push()	Adiciona um elemento no fim do array	`var array = [1,2,3,4];`
		`array.push(5);`
		`//array = [1,2,3,4,5]`

shift()	Retira o primeiro elemento do array e devolve-o	`var array = [1,2,3,4];`
		`var primeiro = array.shift();`
		`//primeiro = 1, array = [2,3,4]`

unshift()	Adiciona um elemento ao início do array	`var array = [1,2,3,4];`
		`array.unshift(0);`
		`//array = [0,1,2,3,4]`

reverse()	Inverte a ordem dos elementos do array	`var array = [1,2,3,4];`
		`array.reverse();`
		`//array = [4,3,2,1]`

FUNÇÕES PREDEFINIDAS DO JAVASCRIPT (*NÚMEROS*)

isNaN	Verifica se um valor é um número ilegal (Not-a-Number)	`isNaN(123); //false`
		`isNaN(0/0); //true`

toFixed(n)	Converte o número numa string apenas com n casas decimais	`var numero1 = 123.3456;`
		`var numero2 = toFixed(0);`
		`//numero2 = "123"`

parseFloat()	Faz o parse de uma string e devolve um float	`var a = parseFloat("10.33");` `//a = 10.33;`
parseInt()	Faz o parse de uma string e devolve um integer	`var a = parseInt("10.33");` `//a = 10;`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

CRIAÇÃO DE FUNÇÕES

```
function nome_função(parametro1, parametro2, parametro3,…) {
    código a ser executado
}
```

```
function soma(a, b) {
    return a + b;
}

var x = soma(3,2);  // x = 5
```

parâmetros

argumentos

invocação da função

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

CRIAÇÃO DE FUNÇÕES (RETURN)

As funções param de executar assim que surgir a expressão return (“*voltar*”, “*devolver*”).

Normalmente, as funções devolvem valores de retorno (*return values*):

```
function soma(a, b) {
    return a + b;
}

var x = soma(3,2);  // x = 5
document.write("O valor de x é: " + x);
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

FUNÇÕES

As funções mais simples não precisam de argumentos nem de retornar valores

```
function dizOla() {
    alert
(“Olá!”);
}

dizOla();
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

DOM - DOCUMENT OBJECT MODEL

FACULDADE ENGENHARIA
Departamento de Informática

DOCUMENT OBJECT MODEL (DOM)

O DOM define uma estrutura de representação das páginas web, tratando todos os elementos da página como objectos.

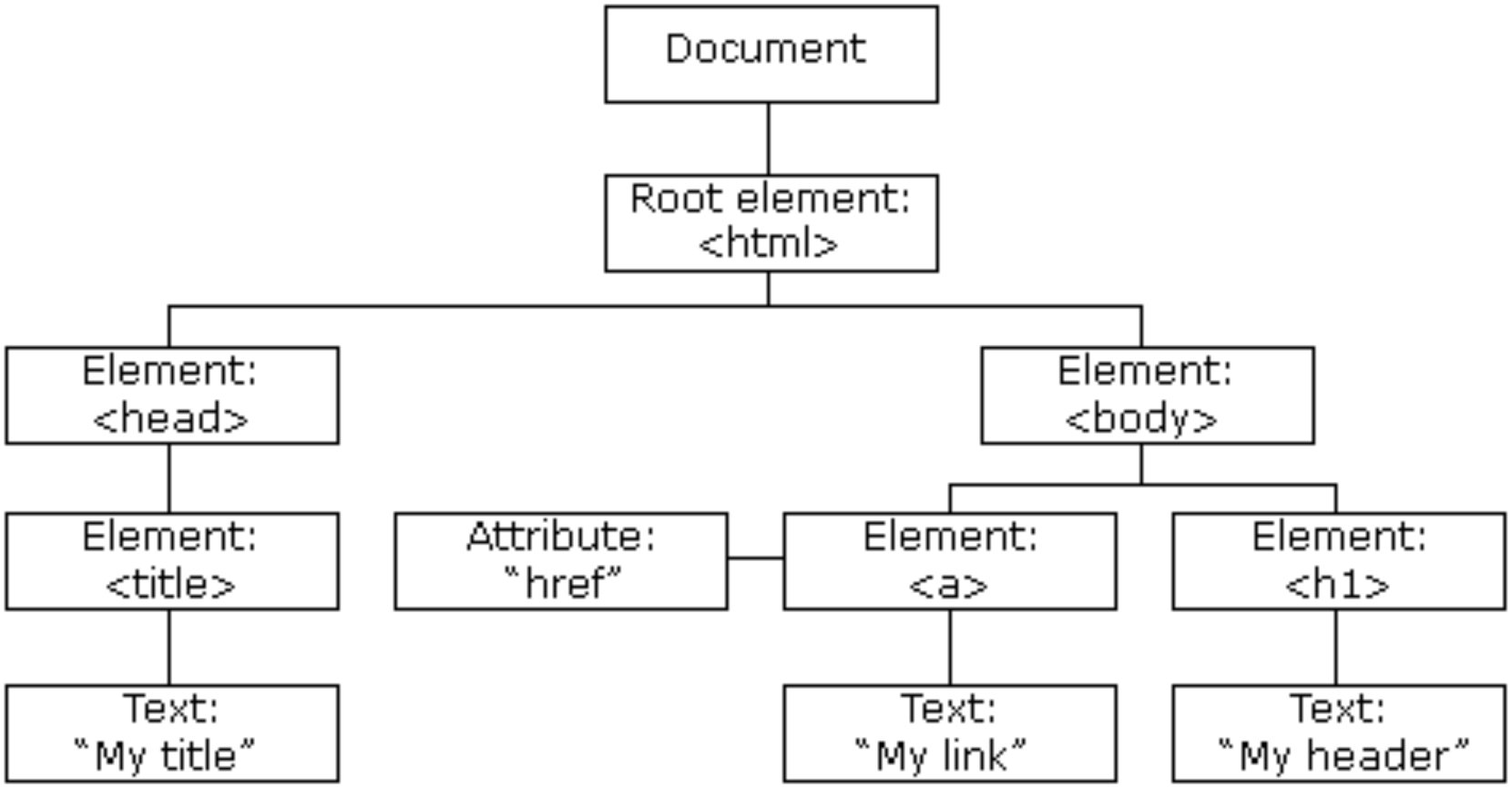
Permite aceder aos elementos de um documento HTML e ler, modificar, apagar ou adicionar esses elementos.

O Javascript permite aceder dinamicamente aos nós e atributos da árvore que representa o DOM, e ler e alterar os seus valores.

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

DOCUMENT OBJECT MODEL (DOM)



SEBASTIAO@DI.UBI.PT

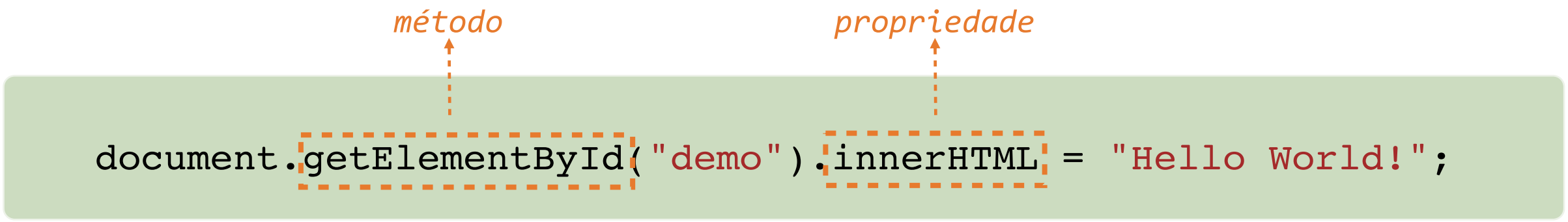
FACULDADE ENGENHARIA
Departamento de Informática

DOCUMENT OBJECT MODEL (DOM)

No DOM, os elementos HTML são definidos como objectos. Esses "objectos" têm:

Propriedades: valores que podemos obter ou alterar (ex: mudar o conteúdo de um elemento HTML)

Métodos: acções que podemos realizar (ex: adicionar ou apagar um elemento HTML)



SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

ACEDER E MANIPULAR ELEMENTOS HTML

O objecto document representa a página web. Para aceder a qualquer elemento HTML na página, precisamos de aceder primeiro ao document.

Método	Descrição
`document.getElementById(id)`	Encontrar elementos pelo seu id
`document.getElementsByTagName(name)`	Encontrar elementos pela sua tag
`document.getElementsByClassName(name)`	Encontrar elementos pela sua classe

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

ALTERAR ELEMENTOS HTML

Método	Exemplo
elemento.**innerHTML** = *novo conteúdo HTML*	document.getElementById("demo").**innerHTML** = "new demo";
elemento.atributo = novo valor	document.getElementById("myImage").src = "desert.jpg";
elemento.**setAttribute** = *(atributo, valor)*	document.getElementsByTagName("h1")[0].**setAttribute**("class", "especial);
elemento.**style**.*propriedade = novo estilo*	document.getElementById("demo").**style**.color = "red";

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

MÓDULO 4: JAVASCRIPT

EVENTOS

FACULDADE ENGENHARIA
Departamento de Informática

EVENTOS

O Javascript é uma linguagem orientada a eventos

Podem ser produzidos pelo sistema:

- Carregamento de uma página web

Podem resultar de acções directas do visitante:

- Premir um botão

Com Javascript podemos "reagir" a esses eventos: despoletar acções (executar código) quando esses eventos ocorrem

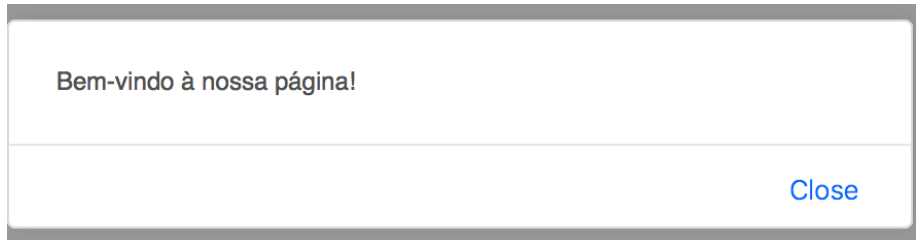
SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

RESPOSTA A EVENTOS DE SISTEMA (EXEMPLO)

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function entrar() {
          alert("Bem-vindo à nossa página");
      }
    </script>

  <body onload="entrar();">
    <h1>A nossa página</h1>
  </body>
</html>
```



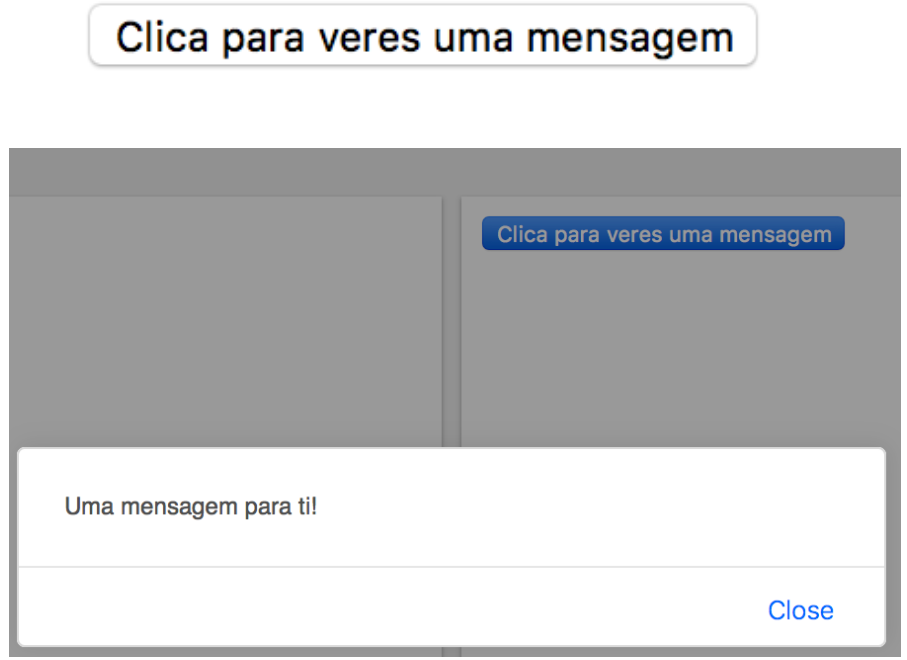
SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

RESPOSTA A EVENTOS DE UTILIZADOR (EXEMPLO)

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function mostraMsg() {
          alert("Uma mensagem para ti!");
      }
    </script>

  <body>
    <button onclick="mostraMsg();">
       Clica para veres uma mensagem
    </button>
  </body>
</html>
```



SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

EVENTOS COMUNS

Evento	Descrição	Elementos para os quais está definido
onclick	Clicar num elemento	Todos os elementos
onmouseover	O cursor "entra" num elemento	Todos os elementos
onmouseout	O cursor "sai" de uma elemento	Todos os elementos
onchange	Desselecionar um elemento que se modificou entretanto	<input>, <select>, <textarea>
onkeypress	Clicar num tecla	Elementos de formulario e <body>
onunload	Quando o browser carrega a página	<body>

Nota: Existem muitos eventos. **Consultar**: http://www.w3schools.com/jsref/dom_obj_event.asp

SEBASTIAO@DI.UBI.PT