

# Aplicações para a Internet I

## JavaScript

# JavaScript

- **JavaScript** é uma linguagem de programação interpretada
- Linguagem para programação *client-side* - ou seja, em contextos normais de desenvolvimento web, é interpretada pelo cliente (web browser) e não pelo servidor.
  - Exceção para programação server-side com javascript como Node.js.
  - Isto irá permitir:
    - Alterar conteúdo da página (adicionar, remover, alterar, etc..);
    - Alterar estilos;
    - Efetuar validações;
    - Executar lógica sem a necessidade de efetuar pedidos ao servidor;
    - Efetuar pedidos Ajax;
    - Entre outros..

Ver slides em anexo

# JavaScript- Sintaxe

- JavaScript é case-sensitive – por isso var1 é diferente de Var1.
- As variáveis JavaScript são declaradas com a keyword "var" `var variavel1;`
- As variáveis não são definidas por tipo, ou seja, até a sua definição, não têm tipo (não são string ou inteiro ou qualquer outro tipo de dado);

```
> var var1;  
< undefined  
  
> (typeof var1) //código para obter o tipo de variável  
< "undefined"  
  
> var1 = 0;  
< 0  
  
> (typeof var1) //código para obter o tipo de variável  
< "number"  
  
> var1 = "texto";  
< "texto"  
  
> (typeof var1) //código para obter o tipo de variável  
< "string"
```

# JavaScript - Sintaxe

- As variáveis têm de ter um identificador único e devem seguir as seguintes regras:
  - Nome da variável pode conter letras, números, "\$" e "\_";
  - Nome da variável tem de começar com uma letra, "\$" ou "\_";
  - Nomes das variáveis são case-sensitive;
  - Existem palavras reservadas em JavaScript que não podem ser utilizadas como nome de variáveis.

# JavaScript - Sintaxe

- Podemos definir comentários em JavaScript da seguinte forma:
  - Entre "//" e o fim dessa mesma linha de código;
  - Entre "/\*" (início) e "\*/" (fim) - funciona para linhas múltiplas;
  - JavaScript reconhece também o início de comentários HTML "<!--" mas não interpreta como início de bloco, mas sim como interpreta o "//";
  - JavaScript não reconhece o final de comentários HTML "-->".

```
// Change heading:
document.getElementById("myH")
// Change paragraph:
document.getElementById("myP")
```

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

# JavaScript - Operadores

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

## JavaScript (head, body ou externo)

```
<!DOCTYPE html>
<html>

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

<body>

<h1>My Web Page</h1>

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

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

</body>
</html>
```

## JavaScript (head, body ou externo)

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

<h1>My Web Page</h1>

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

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

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

</body>
</html>
```



## JavaScript (head, body ou externo)

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

# JavaScript

O JavaScript pode alterar os conteúdos do HTML

`getElementById()`

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

<h1>What Can JavaScript Do?</h1>

<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?

Hello JavaScript!

Click Me!

# JavaScript

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

<h1>JavaScript Can Change Images</h1>

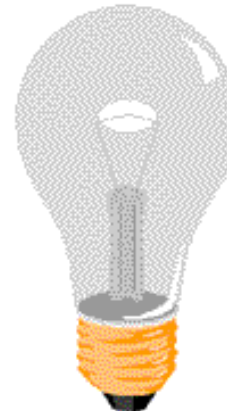


<p>Click the light bulb to turn on/off the light.</p>

<script>
function changeImage() {
    var image = document.getElementById('myImage');
    if (image.src.match("bulbon")) {
        image.src = "pic_bulboff.gif";
    } else {
        image.src = "pic_bulbon.gif";
    }
}
</script>

</body>
```

## JavaScript Can Change Images



Click the light bulb to turn on/off the light.

# JavaScript

## What Can JavaScript Do?

JavaScript can change the style of an HTML element.

Click Me!

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

<h1>What Can JavaScript Do?</h1>

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

<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>

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

</body>
</html>
```

# JavaScript

```
<input id="numb" type="number">

<button type="button" onclick="myFunction()">Submit</button>

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

<script>
function myFunction() {
    var x, text;

    // Get the value of the input field with id="numb"
    x = document.getElementById("numb").value;

    // If x is Not a Number or less than one or greater than 10
    if (isNaN(x) || x < 1 || x > 10) {
        text = "Input not valid";
    } else {
        text = "Input OK";
    }
    document.getElementById("demo").innerHTML = text;
}
</script>
```

## JavaScript Can Validate Input

Please input a number between 1 and 10:

# JavaScript (Métodos de Strings)

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

<p id="p1">Please locate where 'locate' occurs!.</p>

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

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

<script>
function myFunction() {
    var str = document.getElementById("p1").innerHTML;
    var pos = str.indexOf("where");
    document.getElementById("demo").innerHTML = pos;
}
</script>

</body>
</html>
```

Please locate where 'locate' occurs!.

Try it

# JavaScript (Metodos de Strings)

```
var str = "Apple, Banana, Kiwi";  
var res = str.slice(7,13);
```

Banana

```
str = "Please visit Microsoft!";  
var n = str.replace("Microsoft","W3Schools");
```

Please visit W3Schools!

```
var text1 = "Hello World!";  
var text2 = text1.toUpperCase();
```

HELLO WORLD!

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

Hello World!

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

# JavaScript (Datas)

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

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

```
<script>  
var d = new Date("October 13, 2014 11:13:00");  
document.getElementById("demo").innerHTML = d;  
</script>
```

```
var d = new Date("2015-03-25");
```



## JavaScript (If...Else)

```
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  
}
```

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

# JavaScript (Switch)

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

```
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";  
        break;  
}
```

# JavaScript (Loop)

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

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

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

```
for (i = 0; i < 5; i++) {  
    text += "The number is " + i + "<br>";  
}
```

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

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

# JavaScript (output)

JavaScript can "display" data in different ways:

- Writing into an alert box, using **window.alert()**.
- Writing into the HTML output using **document.write()**.
- Writing into an HTML element, using **innerHTML**.
- Writing into the browser console, using **console.log()**.

## JavaScript (output)

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

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

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

</body>
</html>
```

## JavaScript (output)

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

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

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

</body>
</html>
```

## JavaScript (output)

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

<h1>My First Web Page</h1>
<p>My First Paragraph</p>

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

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

</body>
</html>
```

## JavaScript (output)

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

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

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

</body>
</html>
```



# JavaScript - Operadores

Operator	Example	Same As
=	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

# JavaScript – Tipo de variáveis

- Podemos ter diferentes tipos de variáveis em JavaScript sem para isso ser necessário definirmos o tipo na declaração.

```
var length = 16;           // Number
var lastName = "Johnson"; // String
var cars = ["Saab", "Volvo", "BMW"]; // Array
var x = {firstName:"John", lastName:"Doe"}; // Object
```

# JavaScript (variáveis)

```
var price1 = 5;  
var price2 = 6;  
var total = price1 + price2;
```

Todas as variáveis em JavaScript devem ser únicas

```
var pi = 3.14;  
var person = "John Doe";  
var answer = 'Yes I am!';
```

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

## JavaScript (variáveis)

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

<h1>JavaScript Variables</h1>

<p>Add "John" + " " + "Doe":</p>

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

<script>
var x = "John" + " " + "Doe";
document.getElementById("demo").innerHTML = x;
</script>

</body>
</html>
```

## JavaScript Variables

Add "John" + " " + "Doe":

John Doe

# JavaScript - Funções

- Uma função em JavaScript é definida pela keyword "function", seguido do nome da função e "()".
- Dentro dos parêntesis podem estar definidos parâmetros da função.
- O nome das funções segue as mesmas regras das variáveis.
- O código das funções tem de ser definido entre "{" e "}".

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

# JavaScript - Funções

```
function myFunction(p1, p2) {  
    return p1 * p2;  
}
```

```
var x = myFunction(4, 3);
```

```
function myFunction(a, b) {  
    return a * b;  
}
```

```
function toCelsius(fahrenheit) {  
    return (5/9) * (fahrenheit-32);  
}  
document.getElementById("demo").innerHTML = toCelsius(77);
```

# JavaScript - Funções

- JavaScript pode "apresentar" ao utilizador informação de diferentes formas:
  - Em janela de alerta: `window.alert("texto");`
  - Diretamente no HTML: `document.write("texto");`
  - Dentro de um elemento HTML: `$(elemento).innerHTML = "texto";`
  - Na consola do browser: `console.log("texto");`

# JavaScript - Funções

- As funções JavaScript podem ser invocadas da seguinte forma:
  - Quando ocorre um evento;
  - Quando é invocada diretamente no código JavaScript;
  - Automaticamente (auto invocada). Ex.: `(function(){...})();`
- As funções JavaScript podem ser terminadas quando chegam a um *return*, podendo da mesma forma retornar valores.
- As funções em JavaScript são muito importantes porque permitem reutilização de código.

Ver slides em anexo



# JavaScript (Eventos)

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

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