APONTAMENTOS

*JavaScript* – Introdução ao JavaScript

O JavaScript é uma linguagem vocacionada para a web. Usada em biliões de páginas web, acrescenta funcionalidades, valida *forms*, comunica com servidores e muito mais.

Considere os elementos numerados da imagem. Identifica cada elemento numa comunicação cliente/servidor.

5

4



3

2

1

# Como apresentar informação num Output?

O *javascript* pode enviar os dados para um *output* de diferentes formas:

* uma janela alert: **window.alert();**
* um documento HTML: **document.write();**
* um elemento HTML: **innerHTML();**
* a consola do browser: **console.log();**

# Cuidados a ter com browsers antigos!

Muitos *browsers* antigos mostram o código JavaScript no *browser*. Como podes adivinhar não é esse o objetivo. Nesses casos existe uma forma de esconder o código JavaScript.

<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Primeiro exemplo</title>

</head>

<body>

<script>

**<!--**

document.write("Olá a todos!...");

**// -->**

</script>

</body>

</html>

Como podes observar basta colocar os comentários conforme mostrado no exemplo anterior.

# Como está estruturado um documento HTML?

O DOM (***D****ocument* ***O****bject* ***M****odel*) é um *standard* oficial W3C, independente de qualquer linguagem de programação, para acesso e manipulação de documentos HTML. Um exemplo de uma árvore:



Cada elemento da árvore representa um nó, que por sua vez, estabelece uma relação com outro nó. O acesso aos nós pode ser efetuado por *JavaScript*.

O HTML DOM define os objetos e propriedades de todos os elementos HTML e os métodos de acesso.

# Manipulação de elementos HTML

As instruções *JavaScript* são executadas enquanto se processa o carregamento da página. O acesso aos elementos HTML pode realizar-se recorrendo aos métodos:

* ***document.getElementById(id)*** , onde o ***id*** representa o identificador do elemento HTML.
* ***document.getElementsByClassName("classname")*** , onde ***class*** representa a classe elemento HTML. O método ***getElementsByClassName()*** devolve todos os elementos do documento que tenham a classe “classname”, como um objeto NodeList. Um objeto NodeList representa um conjunto de nós (nodes). Os nós são acessíveis por um índice numérico, que inicia com o valor 0. Pode utilizar a propriedade length do objeto NodeList que devolve o número de elementos que possuem a classe definida.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

</head>

<body>

<h1 class="cabecalho">Uma página web</h1>

<h2 class="cabecalho">Classes</h2>

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

<script>

classes = **document.getElementsByClassName**("cabecalho");

var num = **classes.length**;

**classes[0].innerHTML** = "O primerio cabeçalho foi alterado!";

classes[1].innerHTML = "O segundo cabeçalho foi alterado!";

x = **document.getElementById**("demo"); // seleciona o identificador demo

x.innerHTML= num; // altera o conteúdo

</script>

</body>

</html>

O javascript ignora espaços em branco.

Quando o texto é longo pode-se distribuir o texto por várias linhas:

**document.write("<h1>Isto \**

**\é um**

**\cabeçalho</h1>");**

# Como determinar o tipo de uma variável?

O comando ***typeof*** permite avaliar o tipo de dados de uma variável.

**document.write("resultado : <b>", typeof resultado, "</b><br />");**

# Como converter uma string num número inteiro?

var num1 = **parseInt**("1234blue");

# Como converter uma string num número real?

var num4 = **parseFloat**(“22.5”);

# Como criar uma constante?

**const altura** = 1.74;

# Como converter um número numa string?

var num = 10, numero;

numero = **num.toString()**;

# Como procurar uma string dentro de outra?

var msg = "Escola Secundária Emídio Navarro - Viseu";

var substring = "Emídio";

**var posicao = msg.search(substring);**

document.write(posicao);

Se a string a pesquisar não existir então o valor devolvido será -1. O primeiro elemento da string tem índice zero (0).

# Como atualizar uma string?

var msg = "Escola Secundária Emídio Navarro - Viseu";

var pesquisa = "Viseu";

var muda = "Jardim";

**var resultado = msg.replace(pesquisa, muda);**

document.write("<b>Mensagem final:</b> " + resultado);

# Como fazer *debugging?*

Enquanto se programa é importante analisar os erros cometidos durante o processo de programação. Apresenta-se uma tabela que ilustra o acesso aos erros nos diferentes browsers:

|  |  |
| --- | --- |
| **Browser** | **Como aceder às mensagens de erro** |
| Google Chrome | **Menu > Ferramentas > Consola JavaScript** ou Ctrl - Shift - J |
| Internet Explorer | **Ferramentas > Ferramentas de programação** (F12) |
| Firefox | **Ferramentas > Desenvolvedor web >** escolher ferramenta ou *Ctrl - Shift - J* |
| Opera | **Página > Ferramentas de desenvolvimento > Opera DragonFly** |

# Quais são os operadores lógicos?

|  |  |
| --- | --- |
| **operador** | **descrição** |
| && | E |
| || | OU |
| ! | NEGAÇÃO |

# Como tratar as datas e horas?

Date() é um objeto que permite trabalhar com datas e horas. Neste momento precisas saber que a instrução **var hoje = new Date()** cria um objeto que vai ser manipulado por funções.

# Como fixar o número de casas decimais?

num = **Math.pow(1.73, 2.5)** – a variável num guarda o resultado do valor 1.732.5

num1 = **num.toFixed(2)** – a variável num1 guarda o valor num com 2 casas decimais.

# Como controlar o fluxo nos ciclos de repetição?

|  |  |
| --- | --- |
| **Instrução** | **Descrição** |
| Continue | Pode ser utilizada para reiniciar uma ação nos ciclos while, do while ou for, se ocorrer uma determinada condição. |
| Break | Pode ser utilizada para interromper um while, do while, if ou switch. Termina e sai da estrutura, transferindo o controlo do programa à instrução seguinte. |