
Elementos da Linguagem JavaScript

Bem-vindo ao **segundo módulo** do nosso curso!

Nele você aprenderá como incluir programas, ou *scripts*, em suas páginas Web.

Este módulo será todo dedicado à linguagem **JavaScript**.

Objetivos da aula:

- ❖ Nessa aula você aprenderá o **que é JavaScript**;
 - ❖ O que se pode fazer com **Javascript ?**
 - ❖ Como **incluir-la** nas suas páginas HTML.
 - ❖ E entenderá:
 - Os diferentes tipos da linguagem;
 - A utilizar e armazenar valores
 - As constantes, variáveis e
 - a concatenação de strings;
 - ❖ Verá ainda alguns **scripts simples**
-

O que é JavaScript ?

- ❖ É uma **linguagem de script** (interpretada) criada por um convênio entre a Netscape-Sun
 - ❖ Nela o código fonte fica **embutido** na página HTML e é **interpretado pelo navegador**.
 - ❖ Ela tem características rudimentares de **Orientação a objetos**, OO
 - ❖ E permite uma **programação dirigida por eventos**
 - ❖ Apesar do nome, ela tem pouco a ver com **Java**. Apenas **semelhança sintática** e do fato de que, ambas podem fornecer um **conteúdo executável** pelos navegadores Web, a **principal relação** entre as duas linguagens vem de sua **origem comum**: as linguagens **C e C++**.
-

O que se pode fazer com Javascript ?

Com JavaScript é possível:

- ❖ **manipular imagens** embutidas na página;
 - ❖ ler e escrever o estado do usuário em pequena quantidade de dados armazenados no sistema do cliente - *Cookies*;
 - ❖ **interagir com o usuário**;
 - ❖ **Validar e controlar o conteúdo de formulários HTML**
 - ❖ **Detectar versão do navegador**
 - ❖ **Detetar necessidade de instalação de plug-ins**
 - ❖ **Manipular o conteúdo da página**, isso é alterar características do documento e controlar o comportamento do navegador fazendo este:
 - **carregar** um novo documento;
 - **retornar** para a página anteriormente visitada;
 - **exibir páginas diferentes de acordo com o browser do usuário**;
 - **Abrir e fechar janelas**.
-

JavaScript não possibilita:

- ❖ escrever ou ler arquivos;
- ❖ não possui capacidade de desenho;
- ❖ não suporta estabelecer conexões em rede,
- ❖ nem oferece possibilidades de processamento em segundo plano ou multitarefa.

Tipos de JavaScript

Existem **variantes da linguagem** dependendo do contexto onde se **embute o código** JavaScript, estas são:

❖ **Client-Side JavaScript (ou "no lado do cliente")**

- código é **embutido no HTML**
- precisa ser **interpretado pelo browser**
- o **interpretador** se encontra **embutido no navegador Web**
- é a variante mais comum
- nesse curso abordaremos apenas essa forma

❖ **Server-Side JavaScript:(ou "no lado do servidor")**

- pode ser usado com Netscape Server
 - alternativa para scripts CGI
 - nesta forma, diferente da anterior, é possível ler e escrever arquivos e bancos de dados no servidor
-

Observações:

- ❖ Como em HTML, há diferenças na linguagem JavaScript em função do **tipo de navegador** onde o programa será interpretado.
 - ❖ O nome **JavaScript** é propriedade da **Netscape**
 - ❖ A implementação da **Microsoft** da linguagem é oficialmente denominada **JScript**.
-

Incluindo JavaScript numa página :

- ❖ Há três formas de incluir um código JavaScript em uma página:
 - através da **tag** `<SCRIPT> ... </SCRIPT>`.
isso é entre `<SCRIPT>` e `</SCRIPT>`
 - através da **seleção de um link**.
No atributo **HREF** da tag `<A> ... `
 - através da **associação a um evento**.
-

Uso na tag **<SCRIPT>**

- ❖ Na primeira forma, a tag `<script>` pode aparecer em qualquer lugar da página. As linhas abaixo exemplificam esta forma:

```
<script language="Javascript">
```

```
<!--
```

```
...//codigo Javascript ...
```

```
-->
```

```
</script>
```

- ❖ Nessa forma não é possível fazer referência a elementos HTML ainda não criados
-

Uso em um link e em eventos:

- ❖ Na segunda forma, o link ao ser selecionado promove a execução do programa:

```
<a href="javascript: ...">
```

```
....
```

```
</a>
```

- ❖ Quando associada a um evento a sintaxe usada é:

```
<tag-html onEvento=" ... ">
```

Estrutura da linguagem

- ❖ Letras MAIÚSCULAS \neq letras minúsculas

por exemplo: **if** é uma palavra reservada, mas **IF** e **If** não, e cada uma considerada diferente será diferente

- ❖ Espaços, tabulações e linhas em branco são ignorados

- ❖ O caracter **ponto e vírgula** (;) é usado como limitadores, após cada comando.:

```
a = 3 ; b = 5 ;
```

- ❖ Mas podem ser omitidos se os comandos se encontrarem em linhas diferentes, por exemplo

```
a = 3
```

```
b = 5 ;
```

Como incluir comentários:

A linguagem JavaScript possui mais de uma forma:

- ❖ **As barras duplas, //**, em qualquer ponto da linha fazem com que o interpretador ignore todos os demais caracteres até o fim da linha.
- ❖ **A seqüência: /*** faz com que o interpretador ignore todos os caracteres **até encontrar a seqüência oposta: */**
- ❖ na forma **client-side** pode-se usar a sequência de **abertura de comentários de HTML: <!--**, que é tratada como **//**

Tipos de dados

- ❖ JavaScript tem 3 tipos primitivos de dados: números, strings e booleanos.
- ❖ Quanto a forma de utilização o dado pode ser literal (ou constante) ou ter um valor variável
- ❖ Os literais tem seu tipo associado a forma como aparecem no programa, por exemplo:

“João”

String delimitada por aspas

‘Linguagem JavaScript’

String delimitada por plicas

8, -12

Inteiros na base 10

047, -0123

Inteiros na base 8 - Octais

0x5F, -0x10

Inteiros na base 16 - Hexadecimal

3.1415, 0.25

Números de ponto flutuante - reais

true, false

Valores booleanos

Variáveis

- ❖ Uma **variável** é um nome associado a um dado ou valor, ou seja que "armazena" um valor.
- ❖ Elas permitem que se **manipule dados** no programa.
- ❖ **Nomes possíveis de variáveis:**
 - Letras, números, '_' e '\$'
 - **Não pode começar com número**
 - **Não pode ser igual as *palavras reservadas da linguagem*.**
Ex: **true, var, if**, etc.
- ❖ Exemplos válidos:

```
var i;           // Cria a variável i com valor undefined
i = 10;         // Guarda o valor 10 dentro de i
var i = 2;      // Faz as duas coisas ao mesmo tempo
i = 11;         // A palavra var é opcional
```
- ❖ Os tipos de dados podem ser **assumidos e convertidos dinamicamente** pela linguagem

```
i = 11;         i = "onze";
```

Números

- ❖ é o tipo mais básico;
- ❖ JavaScript não faz distinção entre valores inteiros e reais (pontos flutuantes). Todos são representados como valores de **ponto flutuante em 8 bytes**.
- ❖ Todos os inteiros entre -2^{53} ($= -9007199254740992$) e 2^{53} ($= 9007199254740992$), inclusive estes, podem ser representados exatamente.
- ❖ A linguagem tem alguns **valores numéricos especiais**:

`Number.MAX_VALUE`

o maior valor possível

`Number.MIN_VALUE`

o menor valor possível

`Number.POSITIVE_INFINITY`

numero muito grande

`Number.NEGATIVE_INFINITY`

numero muito grande negativo

`Number.NaN`

o valor especial not-a-number

concatenação de strings

- ❖ Como JavaScript é usada para produzir páginas Web, você provavelmente usará muitas **strings** em seus **scripts**.
- ❖ Se você usar o símbolo **+** (mais) com **strings**, você as juntará simplesmente unindo ambas. Isso é chamado de **concatenação**.

Por exemplo:

```
"2 dividido por " + " 0 " + " é " +  
"matematicamente indefinido!"
```

junta todo o texto em uma string única:

```
"2 dividido por 0 é matematicamente  
indefinido!"
```


duas formas de escrever textos nas páginas html

- ❖ **os métodos abaixo são usados para escrever textos dinamicamente nas páginas html:**

`document . write`
`document . writeln`

`document . write (“ Minha Página
 ”) ;`

`document . writeln (“ Minha Página
 ”) ;`

- ❖ **Exemplo**
-

finalizando....

Depois disso faça **os exercícios sugeridos** e sua **auto-avaliação**.

Não deixe de rever essa aula ou reler o material se ainda não se sentir seguro!

Até a próxima aula, onde conheceremos os **COMANDOS** da linguagem!
