



**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO SUL
Câmpus Feliz

Linguagem JavaScript

Prof. Moser Fagundes

Programação III
ADS

Sumário

- Introdução
- Sintaxe do JavaScript
- Variáveis
- Operadores
- Estruturas de controle
- Arrays

O que é o JavaScript ?

- Amplamente usada na Web (navegadores).
- É uma linguagem **interpretada**.
- A sua orientação a objetos é **baseada em protótipos** (não possui o conceito de classes).
- Tipagem **fraca e dinâmica**
- **Funções de primeira classe** (são tratadas como se fossem um valor, ou seja, podemos atribuir funções às variáveis).

Onde inserir o JavaScript no HTML?

Pode ser inserido em dois locais no código HTML:

Dentro do body

```
<body>  
  <script> ... </script>  
</body>
```

→ ver **Exemplo_Slide_04A.html**

Dentro do head

```
<head>  
  <script> ... </script>  
</head>
```

→ ver **Exemplo_Slide_04B.html**

Escrevendo no console (*debugging*)

- Você pode usar o **console** para exibir dados com JavaScript (*debugging*).

- **Exemplo (sintaxe):**

```
console.log("Passou por aqui!");
```

- No **Firefox**, abra a página **Exemplo_Slide_05.html**, e então abra o console clicando em “Ferramentas”, “Desenvolvedor Web” e “Console da Web”. Depois, clique no botão da página **Exemplo_Slide_05.html**.

Desabilitando / Habilitando o JavaScript

Chrome

- Configurações
- Mostrar configurações avançadas
- Privacidade e configuração de conteúdo
- Na aba JavaScript marque "*Não permitir ...*"

Firefox

- Digite na barra de endereço: **about:config**
- Procure por **JavaScript.enabled**
- Coloque a propriedade como **false**

Note que o JavaScript é executado no cliente, não no servidor.

JavaScript e os elementos HTML

Conforme vimos no exemplo anterior, o JavaScript pode ser usado para **modificar elementos da página HTML**:

- O JavaScript pode acessar e modificar o conteúdo de páginas HTML através do **DOM**.
- **DOM** = Documento Object Model
 - É o padrão W3C para acessar elementos HTML.

JavaScript e os elementos HTML

- Para acessar os elementos HTML podemos usar os seus identificadores **id**:

```
document.getElementById("id1")
```

- Usamos o atributo **innerHTML** para modificar o seu conteúdo:

```
document.getElementById("id1").innerHTML
```

→ ver **Exemplo_Slide_08.html**

JavaScript e os elementos HTML

Com JavaScript também podemos:

- Acessar atributos HTML
→ ver **Exemplo_Slide_09A.html**
- Modificar o estilo de elementos HTML (CSS)
→ ver **Exemplo_Slide_09B.html**
- Validar dados
→ ver **Exemplo_Slide_09C.html**

Sintaxe do JavaScript

- Literais numéricos podem ser escritos com ou sem casas decimais:

1001

3.1415

- Literais string podem ser especificados com aspas duplas ou simples:

"Feliz"

'Capital da Cerveja Artesanal'

- Literais array (entre **colchetes**):

[3, 55, 62, 4, 45]

Variáveis

- As variáveis **são** declaradas do seguinte modo:

```
var x, z;  
x = 5;  
z = 6;
```

- O nome das variáveis deve começar por letra, \$ ou _ (underline).
- É case-sensitive.

- Se usa a palavra chave **var** para criar uma nova variável.

- As variáveis que não foram inicializadas terão valor **undefined**.

→ ver **Exemplo_Slide_11.html**

Variáveis

- As variáveis não são **fortemente tipadas**.
- Temos apenas um único tipo de número.

```
var size = 16;           // number
var points = x * 10;     // number
var a=10, b=20, c=30;
```

```
var lastName = "Smith";  // string
var cars = ["GM", "Volvo", "BMW"]; // object
```

```
// também temos boolean e undefined
```

Variáveis Numéricas

- Casos particulares:

// CUIDADO

```
> 3 / 0  
Infinity
```

```
> "ADS" * 10  
NaN
```

→ ver **Exemplo_Slide_13.html**

Variáveis Numéricas

- **Math API** (similar ao Java):
 - **Math.abs**
 - **Math.floor**
 - **Math.min**
 - **Math.max**
 - **Math.pow**
 - **Math.random**
 - **Math.round**
 - **Math.sqrt**

Sintaxe do JavaScript

- **Strings:**

```
var nome = "Feliz";
```

```
console.log(nome.charAt(2));           // 1  
console.log(nome.concat("!"));         // Feliz!  
console.log(nome.indexOf("z"));        // 4  
console.log(nome.replace("liz", "io")); // Feio
```

```
var nomes = "Pedro;Maria;Lucas;Juliana";  
var nomesArray = nomes.split(";");  
console.log(nomesArray[0]);
```

→ ver **Exemplo_Slide_15.html**

Sintaxe do JavaScript

- **Boolean:**

- Os seguintes valores são avaliados como **false**:

false	// Falso
0	// Zero
""	// String vazia
' '	// String vazia
NaN	// Not a Number
undefined	// Indefinido
null	// Nulo

- Todos os outros são **true** por padrão.

→ ver **Exemplo_Slide_16.html**

Operadores

Principais operadores:

x = y + 2; // soma, operador +

x = y - 2; // subtração, operador -

x = y * 2; // multiplicação, operador *

x = y / 2; // divisão, operador /

x = y % 2; // módulo, operador %

x++; // incremento unitário, operador ++

x--; // decremento unitário, operador --

Exercício

Crie um programa JavaScript com uma variável **tempo** contendo um número inteiro correspondente à idade de uma pessoa em dias.

- Por exemplo, se a pessoas tem 400 dias de vida (**tempo=400**) o programa deve escrever no Console:
1 ano, 1 mês e 5 dias
- Para facilitar o cálculo, considere todo ano com 365 dias e todo mês com 30 dias.
- Use a função **Math.floor** para obter valores inteiros nas divisões.

Operadores

Cuidado com os operadores de comparação!

```
0 == ""           // true
```

```
1 == "1"          // true
```

```
1 == true          // true
```

```
0 === ""           // false
```

```
1 === "1"          // false
```

```
1 === true          // false
```

Sintaxe do JavaScript

- Criando funções em JavaScript:

```
<script>
```

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

```
</script>
```

→ **ver Exemplo_Slide_20.html**

Condicional

```
if ( (hora>6) && (hora < 20) ) {  
    mensagem = "Bom dia!";  
} else {  
    mensagem = "Boa noite!";  
}
```

→ ver [Exemplo_slide_21.html](#)

Exercício

- Crie uma função que receba dois números como parâmetros, e que retorne apenas maior deles.

Escreva o número retornado no Console.

Switch

```
switch (new Date().getDay()) {  
    case 0:  
        day = "Domingo";  
        break;  
    case 6:  
        day = "Sábado";  
        break;  
}
```

→ ver [Exemplo_Slide_23.html](#)

Exercício

- Crie uma função que receba um número como parâmetro. Caso este número esteja entre 0 (inclusive) e 3 (inclusive), escreva o nome do número por extenso. Use um switch para implementar o código da função.

While / Do While

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

```
do {  
    text += " " + i;  
    i--;  
} while (i > 0);
```

→ ver **Exemplo_Slide_25.html**

Exercício

- Crie uma função que receba dois números n e d como parâmetros. Se n for maior que zero, esta função deve imprimir todos os números entre 0 e n (inclusive) que sejam divisíveis por d.

Use um laço while para resolver o problema.

Arrays

- Criando um array em JavaScript:

```
var carros = ["GM", "Volvo", "BMW"];
```

ou

```
var carros = new Array("GM", "Volvo", "BMW");
```

- Acessando e modificando um elemento do array:

```
var nome = carros[0];  
carros[0] = "Ferrari";
```

Arrays

- Percorrendo os elementos de um array:

```
var i;  
var frutas = ["Pera", "Banana", "Manga"];  
  
for (i = 0; i < frutas.length; i++) {  
    texto += frutas[i];  
}
```

→ ver [Exemplo_Slide_28.html](#)

Arrays

- Removendo elemento com **pop** e **shift**:

```
// pop() remove o último elemento  
var frutas = ["Pera", "Banana", "Manga"];  
frutas.pop();
```

```
// shift() remove o primeiro elemento  
var frutas = ["Pera", "Banana", "Manga"];  
frutas.shift();
```

→ ver **Exemplo_Slide_29_30.html**

Arrays

- Inserindo elemento com **push** e **unshift**:

```
// push() insere novo elemento no final  
var frutas = ["Pera", "Banana", "Manga"];  
frutas.push("Kiwi");
```

```
// unshift() insere novo elemento no inicio  
var frutas = ["Pera", "Banana", "Manga"];  
frutas.unshift("Tomate");
```

→ ver **Exemplo_Slide_29_30.html**

Arquivo Externo JS

Dentro do body

```
<body>  
  <script src="meuScript.js"></script>  
</body>
```

→ ver **Exemplo_Slide_31A.html**

Dentro do head

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

→ ver **Exemplo_Slide_31B.html**