



IN

Javascript
Aula 01 - Introdução

Apresentação do Módulo

Javascript

Objetivo do Módulo:

- 1. Desenvolver competências para programação em Javascript;**
- 2. Cronograma das aulas;**
- 3. Práticas: Páginas e elementos dinâmicos, calculadora, jogos, validações;**

Introdução

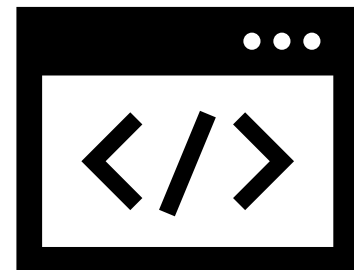
Aula 01

Os objetivos da aula de hoje são:

- 1. Entender o funcionamento básico da linguagem**
- 2. Compreender os elementos iniciais, como variáveis, operadores, tipos de dados e funções básicas**

O que é Javascript?

- JavaScript é uma linguagem de programação client-side.
 - Ela é utilizada para controlar o HTML e o CSS e para manipular comportamentos na página.
 - Mas lembre-se: JavaScript **NÃO É** Java!
- Comportamentos?
 - Imagine três camadas básicas no desenvolvimento para Web:
 - A informação, que fica no HTML
 - A formatação, que fica no CSS
 - O comportamento, que fica no JavaScript



Explorando uma página HTML básica

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título da página</title>
  </head>

  <body>
    <script type="text/javascript">
      alert('Alô Mundo!');
    </script>
  </body>
</html>
```

O que mais eu posso fazer

- JavaScript consegue mudar aspectos de um página em tempo de execução.

```
<body>
    <div id="demo"></div>
    <script>
document.getElementById("demo").innerHTML = "Hello JavaScript";
    </script>
</body>
```

– https://www.w3schools.com/js/js_intro.asp

Onde eu coloco meu código?

- E onde eu coloco meu código? 3 lugares:

```
<head>
    <script>
        alert("Javascript no head!");
    </script>
    <script src="myScript.js"></script>
</head>
<body>
    <script>
        alert("Javascript no body!");
    </script>
</body>
```

– https://www.w3schools.com/js/js_where.asp

Output, escrevendo com Javascript

- Exibindo informações:

```
<p id="demo"></p>
<script>
    //1
    document.getElementById("demo").innerHTML = 5 + 6;
    //2
    document.write(5 + 6);
    //3
    alert(5+6)
    //4
    console.log(5 + 6);
</script>
<!--5 -->
<button type="button" onclick="document.write(5 + 6)">Somar</button>
https://www.w3schools.com/js/js\_output.asp
```


1.1 Prática: Cartão de visitas

- Monte um cartão de visitas online contendo:
 - Seu nome
 - Um botão que, quando pressionado, revela sua profissão

Dica: – https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst



Sintaxe Básica

- Variáveis, números, strings e comentários

```
// Criando variáveis:
```

```
var x;
```

```
let y;
```

```
// Usando variáveis:
```

```
x = 5;
```

```
y = 6.5;
```

```
let z = x + y;
```

```
let nome = "Murilo";
```

```
let sobrenome = 'Freire';
```

```
/* Exemplo de  
Comentário de múltiplas linhas*/
```

https://www.w3schools.com/js/js_syntax.asp

https://www.w3schools.com/js/js_comments.asp

Sintaxe Básica

- Regras de identificadores

/ Identificadores são nomes;*

As regras para Javascript são semelhantes a outras linguagens de programação;

Deve começar por letra, underscore _ ou cifrão \$

*Os caracteres seguintes podem ser letras, dígitos, underscores ou cifrões */*

```
var teste;
```

```
var _teste;
```

```
var $teste;
```

//Javascript é case sensitive

```
let lastname, lastName;
```

```
lastName = "Doe";
```

```
lastname = "Peterson";
```

Sintaxe Básica

- Operadores e expressões

(+ - * /)

(5 + 6) * 10 != 5 + 6 * 10

- Operador de atribuição =

var x = 10; // Em x coloque 10

- Outros operadores

– https://www.w3schools.com/js/js_arithmetic.asp

Sintaxe Básica

- Operadores de string

```
let text1 = "John";  
let text2 = "Doe";  
let text3 = text1 + " " + text2; // "John Doe"
```

/* Somando números e letras: **cast** para **string** */

```
let x = 5 + 5; // 10  
let y = "5" + 5; // "55"  
let z = "Oi" + 5; // "Oi5"
```

Sintaxe Básica

- Operadores de comparação

`==` igualdade // `x == y`

`===` mesmo valor e mesmo tipo // `5 === "5"`

`!=` diferente

`!==` diferente no valor OU diferente no tipo

`>` maior que

`<` menor que

`>=` maior igual

`<=` menor igual

`?` Operador ternário // Veremos em comandos condicionais

https://www.w3schools.com/js/js_operators.asp

Sintaxe Básica

- Operadores lógicos

&& and // **var1 && var2**

|| or // **var1 || var2**

! not // **!var1**

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X	<u>X</u>
0	1
1	0

Sintaxe Básica

- Palavras Reservadas (Keywords)

`var, let, function`

https://www.w3schools.com/js/js_syntax.asp

- Let, const e var
 - Variáveis declaradas com **let** não podem ser redeclaradas.
 - Variáveis devem ser declaradas antes de serem usadas.
 - Variáveis definidas com **let** tem escopo local (block scope).

```
let x = "John Doe";  
let x = 0; //erro
```

– https://www.w3schools.com/js/js_let.asp

Sintaxe Básica

- Let, const e var
 - **let** tem escopo local, **var** tem escopo global

```
var x = 10; // Aqui x é 10
let y = 10; // Aqui y é 10
```

```
{
var x = 2; // Aqui x é 2
let x = 2; // Aqui y é 2
}
```

```
// Aqui x é 2
// Aqui y é 10
```

– https://www.w3schools.com/js/js_let.asp

Sintaxe Básica

- Let, const e var
 - Javascript tem 3 escopos: **Global, Function e Block**
 - Variáveis declaradas com a palavra reservada **var** não podem ter escopo local (Block Scope)
 - Variables declaradas entre **{ }** podem ser acessadas fora desse bloco.

```
{  
var x = 2; // Aqui x é 2  
let x = 2; // Aqui y é 2  
}
```

```
// x pode ser usado aqui  
// y não pode ser usado aqui
```

– https://www.w3schools.com/js/js_let.asp

Sintaxe Básica

- Let, const e var
 - Uma variável **const** não pode ser reatribuída;
 - Estas variáveis devem receber um valor quando são declaradas
 - **const** tem Block Scope.

```
const PI = 3.141592653589793;
```

```
PI = 3.14;           // This will give an error
```

```
PI = PI + 10;        // This will also give an error
```

– https://www.w3schools.com/js/js_const.asp

Sintaxe Básica

- Atribuição de valores

```
let x = 10;
```

```
x += 5; //x = x + 5;
```

```
x -= 5; //x = x - 5;
```

```
x *= 5;
```

```
x /= 5;
```

```
x %= 5;
```

```
x++; //x = x + 1;
```

```
x--; //x = x - 1;
```

```
let text1 = "Um dia muito ";
```

```
text1 += "bonito"; //"Um dia muito bonito"
```

https://www.w3schools.com/js/js_assignment.asp

Sintaxe Básica

- Aprendendo um pouco mais sobre strings e números

https://www.w3schools.com/js/js_strings.asp

https://www.w3schools.com/js/js_numbers.asp



Funções - introdução

- Bloco de código designado a realizar uma tarefa específica

```
function myFunction(p1, p2) {  
    return p1 * p2;    // Retorna o produto de p1 e p2  
}
```

```
var x = myFunction(4, 3);  
Console.log(x);
```

https://www.w3schools.com/js/js_functions.asp

Funções - introdução

- Funções

- https://www.w3schools.com/js/js_functions.asp

- Parâmetros

- https://www.w3schools.com/js/js_function_parameters.asp



1.2 Prática: Atualizando o cartão de visitas

- Altere seu cartão de visitas online:
 - Retire o botão de data de nascimento
 - Crie uma função chamada “exibeDadosPess()” para exibir seus dados pessoais
 - Crie uma função chamada “exibeDadosProf()” para exibir seus dados profissionais
 - Crie dois novos botões para chamar essas funções
- Dica:
 - https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst
 - https://www.w3schools.com/js/js_functions.asp

Arrays

- Arrays são um tipo especial de variável, que podem conter mais de um valor;

```
const carros = ["Saab", "Volvo", "BMW"]; // Criando um array
```

```
carros[3] = "Uno"; // Adicionando um element ao fim do array
```

```
let carro = carros[0]; // Índice começa em 0
```

```
carros[1]; = "Opala"; // Trocamos "Volvo" por "Opala"
```

```
carros.push("Corolla"); // Outro jeito de adicionar ao fim do array
```

```
cars.length // Retorna o número de elementos do array
```

```
cars.sort() // Ordena o array
```

https://www.w3schools.com/js/js_arrays.asp

Datas e Funções Matemáticas

- Datas
 - https://www.w3schools.com/js/js_dates.asp
- Funções matemáticas
 - https://www.w3schools.com/js/js_math.asp



1.2 Prática: Calculadora

- Estude o código da calculadora JS fornecido pelo professor.
- Você consegue alterar este código para criar uma calculadora que realiza as quatro operações aritméticas básicas?

Lembre-se: o usuário terá que informar qual a operação (soma, subtração, multiplicação ou divisão) que ele deseja executar, além dos números que serão utilizados nessa operação.

O código fornecido pelo professor está no ambiente EAD do curso e/ou em:

- <https://github.com/EduSSXavier/Infinity-JS>

“ Referências”

1. <https://www.w3schools.com>