

# PROGRAMAÇÃO DE INTERNET

Prof. Richard Brosler

E-mail: [richard.brosler@sp.senai.br](mailto:richard.brosler@sp.senai.br)

Grupo WhatsApp: <https://bit.ly/3K8jFIg>





# AGENDA

- Recordando os conteúdos
  - Introdução ao Javascript
  - Tipos de dados
  - Variáveis e Constantes
  - Operadores matemáticos, relacionais, lógicos, bit-a-bit e ternário.
  - Condicionais: IF
- Condicionais: Switch
- Laços de repetição
  - While
  - Do while
  - For
- Procedimentos e funções
- Objetos, Arrays e operações com Array



# INTRODUÇÃO AO JAVASCRIPT

- Tipos de dados
- Variáveis
- Constantes
- Operadores matemáticos
- Operadores relacionais
- Operadores lógicos
- Operadores bit-a-bit
- Operador ternários
- Estrutura condicional (IF)



# INTRODUÇÃO AO JAVASCRIPT

## TIPOS DE DADOS

- String
- Number
- Boolean
- Null
- Undefined
- Symbol ([artigo](#) interessante sobre eles)





# INTRODUÇÃO AO JAVASCRIPT

## VARIÁVEIS E CONSTANTES

- Para definirmos uma variável no Javascript, podemos definir de 2 formas:

- Utilizando var

```
1 var valor = 10;
```

- Utilizando let

```
1 let valor = 10;
```

- Para definirmos uma constante, usamos o const

```
1 const PI = 3.1415;
```



# INTRODUÇÃO AO JAVASCRIPT

## VARIÁVEIS E CONSTANTES

- Diferenças do var e do let é o contexto.
- Vejamos o exemplo:

```
1  var podePassar = true;  
2  if (podePassar) {  
3      var valor=15;  
4  }  
5  console.log(valor);
```

- Use esse [site](#) para colocar o código





# INTRODUÇÃO AO JAVASCRIPT

## VARIÁVEIS E CONSTANTES

- Diferenças do var e do let é o contexto.
- Vejamos agora outro exemplo:

```
1 let podePassar = true;  
2 if (podePassar) {  
3     let valor=15;  
4 }  
5 console.log(valor);
```

- Use esse [site](#) para colocar o código
- Note que nesse exemplo vai dar erro de variável valor não definida.



# INTRODUÇÃO AO JAVASCRIPT

## VARIÁVEIS E CONSTANTES

- Tanto o `let` quanto o `const` possuem escopo de contexto ao ser utilizado
- O `var` possui escopo do código todo.
- Por esse motivo, cuidado ao utilizar. 😊
- Recomendações:
  - Evite definir variáveis no meio do código;
  - Centralize as definições no início do código ou início da função.





# INTRODUÇÃO AO JAVASCRIPT

## OPERADORES MATEMÁTICOS

- Soma ( + )
- Subtração ( - )
- Multiplicação ( \* )
- Divisão ( / )
- Exponenciação ( \*\* )
- Resto da divisão ( % )
- Incremento ( ++ ) (unário)
- Decremento ( -- ) (unário)



# INTRODUÇÃO AO JAVASCRIPT

## OPERADORES RELACIONAIS

- Maior que ( > )
- Menor que ( < )
- Maior igual à ( >= )
- Menor igual à ( <= )
- Igualdade ( == )
- Igualdade de valor e tipo de dados ( === )
- Diferente ( != )
- Diferente de valor e tipo de dados ( !== )





# INTRODUÇÃO AO JAVASCRIPT

## OPERADORES LÓGICOS

- Operador AND ( && )
- Operador OR ( || )
- Operador de negação ( ! ) (unário)



# INTRODUÇÃO AO JAVASCRIPT

## OPERADORES BIT-A-BIT E TERNÁRIO

- Operador AND ( & )
- Operador OR ( | )
- Operador XOR ( ^ )
- Operador NOT ( ~ )
- Deslocamento à esquerda ( << )
- Deslocamento à direita ( >> )
- Operador ternário ( (condição) ? Valor se verdadeiro : valor se falso )





# INTRODUÇÃO AO JAVASCRIPT

## ESTRUTURA CONDICIONAL IF

- O operador condicional `if`



```
1 //if com um comando somente
2 if (condição) comando;
3 //if com mais de um comando
4 if (condição) {
5     comando1;
6     comando2;
7 }
8 //if e else com um comando
9 if (condição)
10     comando;
11 else
12     comando2; //com else e um comando
13 //if e else com mais de um comando
14 if (condição) {
15     comando1;
16     comando2;
17 } else {
18     comando3;
19     comando4;
20 }
```

# ESTRUTURA CONDICIONAL SWITCH

- O operador condicional switch



```
1  var foo = 1;
2  var output = 'Output: ';
3  switch (foo) {
4      case 10:
5          output += 'Então ';
6      case 1:
7          output += 'Qual ';
8          output += 'É ';
9      case 2:
10         output += 'O Seu ';
11     case 3:
12         output += 'Nome';
13     case 4:
14         output += '?';
15         alert(output);
16         break;
17     case 5:
18         output += '!';
19         alert(output);
20         break;
21     default:
22         alert('Favor escolher um número de 0 à 6!');
23 }
```



# LAÇO DE REPETIÇÃO: WHILE

- Laço de repetição: [while](#)

JS teste.js > ...

```
1  var i = 0;  
2  while (i <= 10) {  
3      console.log(`5 x ${i} = ${i * 5}`);  
4      i++;  
5  }
```



# LAÇO DE REPETIÇÃO: DO .. WHILE

- Laço de repetição: [do .. while](#)

```
JS teste.js > ...  
1  var i = 0;  
2  do {  
3      console.log(`5 x ${i} = ${i * 5}`);  
4      i++;  
5  } while (i <= 10);
```





# LAÇO DE REPETIÇÃO: FOR

- Laço de repetição: for

JS teste.js > ...

```
1  for(let i=0; i<=10; i++){  
2    |  console.log(`5 x ${i} = ${i * 5}`);  
3  }
```



- Documentação sobre funções e procedimentos



```
JS teste.js > ...
1  /* Declarando uma função */
2  function soma(a, b) {
3      |   return a + b;
4      |
5  }
6  /* Declarando uma função */
7  const soma2 = (a,b) => {
8      |   return a + b;
9      |
10     }
11 /* Declarando um procedimento */
12 function helloWorld(){
13     |   console.log("Olá mundo!")
14     |
15 }
16 /* Declarando um procedimento */
17 const helloWorld2 = () => {
18     |   console.log("Olá mundo!")
19     |
20 }
```



# VETORES E MATRIZES

- Vetores/Matrizes em Javascript.



```
JS teste.js > ...
1  /* Declarando um vetor */
2  var vetor = [];
3  /* Declarando um vetor com valores */
4  var vetor = ['Valor1', 'valor2', 15, 50.1, false];
5  /* Declarando uma matriz de 2x2 */
6  var matriz = [['valor10', 'valor11', "valor12"],
7                ["valor20", 'valor21', "valor22"]];
```

# OBJETOS

- Objetos em Javascript.



```
JS teste.js > ...
1  /* Declarando um objeto */
2  var obj = {};
3  /* Declarando um objeto */
4  var obj2 = new Object();
5  /* Declarando um objeto com propriedades */
6  var obj3 = //notação JSON (JavaScript Object Notation)
7  {
8      "id" : 1,
9      "nome": "Roberval",
10     "endereco": "Rua das ruas, 12",
11     "fumante" : false,
12     "salario" : 7500.00
13 }
14 /* Acessando os dados do objeto */
15 console.log(obj3.nome);
```



# ACESSANDO ELEMENTOS HTML COM JAVASCRIPT

- Vamos ver agora como acessar elementos html com Javascript.



```
9   <body>
10   <form action="#" method="post">
11   |   <input type="text" id="Idade" name="nmIdade"
12   |   |   placeholder="Digite sua idade">
13   </form>
14   <button onclick="mostraConteudo()">Clique aqui</button>
15   <script src="scripts.js"></script>
```

```
JS scripts.js > mostraConteudo
1   function mostraConteudo() {
2   |   /* buscando um elemento html pelo id idade */
3   |   var campoIdade = document.getElementById("Idade");
4   |   console.log("valor do campo idade é ", campoIdade.value);
5   |   }
```

# EXERCÍCIOS

- Vamos fazer uma calculadora científica usando html, css e Javascript.