



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS IPANGUAÇU

Aula 2 - Javascript

Comandos e Funções

Roberta Freitas
robertacynthia@gmail.com

Diferença entre let e var, const

- Se você tentar usar **let** antes dela ser declarada, dará um erro e ela não será inicializada. Já com **var**, ela é inicializada.
- **let** tem escopo de bloco. Mas o que é um bloco? Um bloco é uma porção de código cercado por **{}**.
- **var** tem escopo global ou de função.

Diferença entre let e var, const

- Variáveis do tipo **let** **não podem** ser redeclaradas. Já as tipo **var** podem ser declaradas.
- **const** não permite ser mudado.

```
let x = "Maria Joaquina";  
  
let x = 0;
```

Pode ser atualizado, mas
não redeclarado

```
var x = "Maria Joaquina";  
var x = 0;
```

Pode ser redeclarado

Variáveis

- Em JavaScript, uma variável sem valor algum tem seu valor definido como undefined.

```
<script>
  var x;
  alert(x);           //retorna undefined
  alert(typeof(x));  //retorna undefined
```

Operadores numéricos

Operador	Operação	Exemplo	Atribuição
+	Adição	<code>c = a + b</code>	<code>a += b</code>
-	Subtração	<code>c = a - b</code>	<code>a -= b</code>
*	Multiplicação	<code>c = a * b</code>	<code>a *= b</code>
/	Divisão	<code>c = a / b</code>	<code>a /= b</code>
%	Módulo da divisão ou Resto	<code>c = a % b</code>	<code>a %= b</code>
++	Incremento	<code>c = a++</code>	<code>a++</code>
--	Decremento	<code>c = a--</code>	<code>a--</code>

O único operador aritmético que pode ser utilizado com o tipo String é '+'

Strings

- O operador aritmético que pode ser usado em strings é o operador de +.
- Esse operador é usado para concatenação de valores de variáveis.

```
var nome = "Hermione";  
var nomeCompleto = nome + " Granger";  
alert(nomeCompleto);
```

Métodos de Strings

String length
String slice()
String substring()
String substr()
String replace()
String replaceAll()
String toUpperCase()
String toLowerCase()
String concat()

String trim()
String trimStart()
String trimEnd()
String padStart()
String padEnd()
String charAt()
String charCodeAt()
String split()

Números e Strings

- A avaliação das operações em Javascript é feita da esquerda para a direita.
- Adicionar dois números irá retornar um número.
- Adicionar um número e uma string irá retornar uma string.

► Exemplo:

```
a = 5 + 5;  
b = "5" + 5;  
c = "Olá" + 5;  
d = 5 + 5 + "Olá";  
e = "Olá" + 5 + 5;  
f = "5" + "5";
```

Resultado:

```
10  
55  
Olá5  
10Olá  
Olá55  
55
```


Operadores Relacionais

- Em JavaScript os operadores relacionais retornam um valor booleano (true ou false) e são:

Operador	Operação	Exemplo
<	Menor	<code>a < b</code>
>	Maior	<code>a > b</code>
==	Igual	<code>a == b</code>
!=	Diferente	<code>a != b</code>
>=	Maior ou igual	<code>a >= b</code>
<=	Menor ou igual	<code>a <= b</code>

Operadores Relacionais

- Em Javascript, assim como em outras linguagens de tipagem dinâmica, há diferença entre o operador `==` e o operador `===`

Operador	Descrição
<code>==</code>	Faz conversão de tipo e retorna verdadeiro se as variáveis possuem valores iguais.
<code>===</code>	Retorna verdadeiro apenas se as variáveis são do mesmo tipo e possuem valores iguais
<code>!=</code>	Faz conversão de tipo e retorna falso baseado apenas no valor das variáveis.
<code>!==</code>	Retorna verdadeiro se as variáveis são de tipos diferentes, mesmo que possuem diferentes iguais

Exemplo Prático

- Copie e cole esse código e veja sua execução.

HTML

```
<!DOCTYPE html>
<html>
<body>

<h3> O feitiço correto para acender uma luz é: </h3>
<input type="text" id="resposta"/><br/>
<button type="button" id="bresponder"
onclick="corrigir()">Responder</button><br/>
<script>
Colocar trecho Javascript
</script>

</body>
</html>
```

Javascript

```
<script>
function corrigir(){
var resposta =
document.getElementById("resposta").value;
if (resposta == "lumos"){
document.write("Parabéns! Você acendeu a luz.");
document.body.style.backgroundColor = "yellow";
}else{
document.write("Que pena! Você continua no
escuro.");
document.body.style.backgroundColor = "gray";
}
}
</script>
```

Exercício de Fixação

1. Acesse o site W3Schools e veja os métodos de números e de Strings https://www.w3schools.com/js/js_string_methods.asp.
2. Crie um código para testar os métodos com Strings:
 - a. Coloque todas as letras em maiúsculas.
 - b. Coloque todas as letras em minúsculas.
 - c. Conte o tamanho de uma string.
 - d. Declare uma frase e substitua uma palavra dessa frase por outra.
 - e. Crie duas strings e em seguida, junte essas strings utilizando um método.

Funções

- Função é um bloco de construção fundamental em JavaScript.
- Uma função é um conjunto de instruções que executa uma tarefa ou calcula um valor.
- Para usar uma função, você deve defini-la em algum lugar no escopo do qual você deseja chamar.

Exemplo de Funções

```
function quadrado(numero) {  
    return numero * numero;  
}
```

Exemplo de Função


HTML

```
<!DOCTYPE html>
<html>
<body>

<form>
  Digite o Número 1: <input type="text" id="n1"/><br/>
  Digite o Número 2: <input type="text" id="n2"/><br/>
  <button type="button" onclick="somar()">Somar</button><br/>
  R: <input type="text" id="resultado"/><br/>
</form>

<head>

</head>
</body>
</html>
```



Javascript

```
<script>
function somar(){
  var x = document.getElementById("n1").value;
  var y = document.getElementById("n2").value;
  var soma = parseInt(x) + parseInt(y);
  document.getElementById("resultado").value =
    soma;
}
</script>
```

Exemplo de Função


HTML

```
<!DOCTYPE html>
<html>
<body>

<form>
  Digite o Número 1: <input type="text" id="n1"/><br/>
  Digite o Número 2: <input type="text" id="n2"/><br/>
  <button type="button" onclick="somar()">Somar</button><br/>
  R: <input type="text" id="resultado"/><br/>
</form>

<head>

</head>
</body>
</html>
```



Javascript

```
<script>
function somar(){
  var x = document.getElementById("n1").value;
  var y = document.getElementById("n2").value;
  var soma = x + y;
  document.getElementById("resultado").value =
    soma;
}
</script>
```


Como chamar uma função?

- Definir uma função não a executa.
- Chamar a função executa realmente as ações especificadas com os parâmetros indicados.

**Chamando uma
função:**

quadrado(5);

Funções

Uma função pode chamar ela mesma?

Funções

Sim.

**Um exemplo clássico
é a função de fatorial.**

```
function fatorial(n) {  
  if (n == 0 || n == 1) return 1;  
  else return n * fatorial(n - 1);  
}
```

Funções

JS



```
var a, b, c, d, e;  
a = fatorial(1); // a recebe o valor 1  
b = fatorial(2); // b recebe o valor 2  
c = fatorial(3); // c recebe o valor 6  
d = fatorial(4); // d recebe o valor 24  
e = fatorial(5); // e recebe o valor 120
```

Dúvidas



Exercício de Fixação

1. Faça uma calculadora, a partir do código mostrado anteriormente com o botão Soma, adicionando as outras operações:
 - a. Subtração
 - b. Multiplicação
 - c. Divisão
2. Crie uma função para definir que se a pessoa tiver uma idade menor que 18 anos não pode votar e se for maior, deve votar.

Exercício 2

qual a sua idade?

Vota.

Próximas Aulas

- Estruturas de controle de fluxo, de repetição e de sequência.

Bibliografia

Javascript. Disponível em: <https://www.w3schools.com/js>. Acesso em 05/09/2023.