

# JavaScript

Variáveis

Operadores Aritméticos

# Variáveis

## Objetos - Acessar, Atualizar, Criar

Definido	<pre>var Pessoa = {   Nome:"NOME",   Sobrenome:"SB",   Idade:25 };</pre>
Adicionado	<pre>Pessoa.CorDoOlho = "COR";</pre>
Resultado	<pre>Pessoa = {   Nome:"NOME",   Sobrenome:"SB",   Idade:25,   CorDoOlho:"COR" };</pre>
Substituído	<pre>Pessoa.Idade = 20;</pre>
Resultado	<pre>Pessoa = {   Nome:"NOME",   Sobrenome:"SB",   Idade:20,   CorDoOlho:"COR" };</pre>

Definido	<pre>var VETOR = [1, 5, 10];</pre>
Adicionado	<pre>VETOR [3] = 15;</pre>
Resultado	<pre>VETOR = [1, 5, 10, 15]</pre>
Substituído	<pre>VETOR [3] = 2;</pre>
Resultado	<pre>VETOR = [1, 5, 10, 2]</pre>

# Variáveis - Informações extras

Diferença de `null` e `undefined`.

- Muitos erros ocorrem pelo o tipo observado da variável
  - Ao utilizar o `null` , apaga-se todo o conteúdo da variável, mas mantém-se o tipo dela.
  - Ao utilizar o `undefined`, apaga-se todo o conteúdo da variável, inclusive o tipo.

Definido      `var Pessoa = {`  
                  `Nome:"NOME",`  
                  `Sobrenome:"SB",`  
                  `Idade:25`  
                  `};`

Novo Valor    `Pessoa = null`

Resultado     `Pessoa = {`  
                  `};`

Resultado     `typeof (Pessoa)`  
                  `object`

Novo Valor    `Pessoa = undefined`

Resultado     `Pessoa = ;`

Resultado     `typeof (Pessoa)`  
                  `undefined`

# Variáveis - Informações extras

Ao trabalhar com textos (strings) e se necessita utilizar aspas únicas ou duplas dentro de uma variável, pode se utilizar a barra invertida (escape char) ( \ ).

Ela faz com que o próximo caracter especial em frente dela seja tratada como string. Assim, se precisar de aspas duplas, se usa `\ "Texto\"`.

Por ser strings, elas também possuem propriedades extras, tal como a propriedade `length`. Ela retorna o tamanho de caracteres. Ideal para saber se uma senha tem no mínimo 8 caracteres por exemplo.

Outras propriedades específicas podem ser encontradas em:

[https://www.w3schools.com/js/js\\_string\\_methods.asp](https://www.w3schools.com/js/js_string_methods.asp)

# Variáveis - Informações extras

Igualmente aos strings, os *numbers* e *arrays* também tem propriedades específicas.

As de *Numbers* estão em:

[https://www.w3schools.com/jsref/jsref\\_obj\\_number.asp](https://www.w3schools.com/jsref/jsref_obj_number.asp)

As de *Arrays* estão em:

[https://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](https://www.w3schools.com/jsref/jsref_obj_array.asp)

## JavaScript: operadores aritméticos

+	adição ou operador de concatenar strings
	<pre>var a = 5 var b = 10 console.log(a+b) - vai mostrar o valor da soma console.log(5+10) - vai mostrar o valor da soma var c = a+b console.log(c) - vai mostrar o valor da soma</pre>
-	subtração
*	multiplicação
**	potência
/	divisão

%	resto da divisão <pre>var a = 10 var b = 3 console.log(a%b)</pre>
++	incremento
--	decremento
	Adiciona ou diminui +1 ou -1 a uma variável numérica
	<pre>var a = 10, b = 3; a++; console.log(a) (vai mostrar o valor 11);</pre>
	<pre>var a = 10, b = 3; a--; console.log(a) (vai mostrar o valor 9);</pre>
	Pode ser usado direto no console log porém tem que ser escrito antes da variável
	<pre>console.log(++a)</pre>

# Variáveis

## Concatenar

- Para o JavaScript identificar se o “+” é soma ou concatenação, depende das variáveis no processo.
- Ele realiza qualquer operação matemática anterior ao primeiro string de texto

```
var x = 16 + "Volvo";
```

Irá mostrar 16Volvo

```
var x = 8 + 8 + "Volvo";
```

Irá mostrar 16Volvo

```
var x = "8" + 8 + "Volvo";
```

Irá mostrar 88Volvo

```
var x = "Volvo" + 8 + 8 ;
```

Irá mostrar Volvo88