

Professor Allyson Souza

Variáveis

let Declara variáveis locais, restrita ao escopo em que estão

```
let idade  
let nome
```

var Declara variáveis globais, que não se restringem ao escopo em que estão

```
var matricula  
var curso
```

const Declara constantes, variáveis que não podem ter seus valores iniciais modificados

Por convenção, usa-se letras maiúsculas para definição de nomes de constantes.

```
const PI = 3.14  
const DIAS_DA_SEMANA = 7
```

Atribuição de valor

Variáveis podem ser declaradas e depois terem um valor atribuído:

```
let idade  
idade = 25
```

Ou então podem já ter um valor inicial em sua criação.

```
let idade = 25
```

Tipos de dados

string Representa dados textuais

```
let nome = "Antonia"  
let mensagem = "Olá, estudante!"  
let placa = "FJG3A75"
```

number Representa dados numéricos, inteiros e decimais, positivos e negativos

```
let idade = 27  
let saldo = 110.57  
let temperatura = -2
```

boolean Representa um dado lógico, que pode ter apenas um de dois valores: **true** (verdadeiro) ou **false** (falso).

```
let maiorDeIdade = true  
let matriculado = true  
let milionario = false
```

undefined É o valor padrão atribuído pelo JavaScript a uma variável que foi declarada (criada), mas que ainda não recebeu nenhum valor.

null É a ausência intencional de qualquer valor.

```
let nome = null  
let idade = null  
let endereco = null
```

Operadores aritméticos

+	Soma	let resultado = 4 + 2	6
-	Subtração	let resultado = 10 - 7	3
*	Multiplicação	let resultado = 5 * 5	25
/	Divisão	let resultado = 90 / 9	9
^	Potenciação	let resultado = 2 ^ 3	8
%	Resto da divisão	let resultado = 11 % 2	1

Instruções básicas

alert()	Exibe mensagem em janela pop-up/caixa de diálogo.
console.log()	Exibe mensagem no console do navegador.
prompt()	Exibe caixa de entrada de dados em janela pop-up/caixa de diálogo.
typeof	Retorna o tipo de dado armazenado em uma variável

Concatenação de strings

O operador de **+**, quando utilizado com strings, sejam elas valores literais ou armazenados em variáveis, faz a concatenação delas, ou seja, sua junção.

```
let nome = "José"  
alert("Olá, " + nome)
```

Olá, José

```
let preco = 19.99  
alert("R$ " + preco)
```

R\$ 19.99

Professor Allyson Souza

Conversão de dados

Number()	Converte uma string em um número, inteiro ou decimal.
parseInt()	Converte uma string ou um número decimal, em inteiro.
parseFloat()	Converte um string ou um número inteiro, em decimal.
parseFloat()	Converte um número em uma string.

Condicionais

	condição <pre>if(idade >= 18) { console.log("É maior de idade.") } else { console.log("É menor de idade.") }</pre>
--	--

Operadores de comparação

==	Verifica a igualdade entre dois valores ou variáveis
!=	Verifica a diferença entre dois valores ou variáveis
>	Verifica se um valor ou variável é maior que outro
<	Verifica se um valor ou variável é menor que outro
>=	Verifica se um valor ou variável é maior ou igual à outra
<=	Verifica se um valor ou variável é menor ou igual à outra

Operadores lógicos

&&	Operador E: Verifica se uma condição E outra são verdadeiras. Retorna true se ambas as condições forem verdadeiras.
	Operador OU: Verifica se uma condição OU outra são verdadeiras. Retorna true se ao menos uma delas forem verdadeiras.
!	Operador de Negação: transforma uma condição ou valor verdadeiro (true) em falso (false), e vice-versa.

while

A declaração **while** cria um laço que executa uma rotina específica enquanto a condição de teste for avaliada como verdadeira. A condição é avaliada antes da execução da rotina.

```
let cont = 0;  
  
while ( cont < 10 ) {  
  cont++;  
  console.log(cont)  
}
```

do ... while

A declaração **do...while** cria um laço que executa uma declaração até que o teste da condição for falsa (false).

A condição é avaliada depois que o bloco de código é executado, resultando que a declaração seja executada pelo menos uma vez.

```
let cont = 0;  
  
do {  
  cont++;  
  console.log(cont)  
} while ( cont < 10 )
```

for

A instrução **for** cria um loop que consiste em três expressões opcionais, dentro de parênteses e separadas por ponto e vírgula, seguidas por uma declaração ou uma sequência de declarações executadas em sequência.

```
for ( let cont = 0; cont < 20; cont++ ) {  
  console.log(cont)  
}
```