****

**Material Extra**

Variáveis, Tipos e Operadores

>\_ Índice

[**>\_** O que são variáveis?](#_heading=h.1fob9te)

[**>\_** Declaração e tipos de variáveis](#_heading=h.1fob9te)

[**>\_** O que são tipos de dados?](#_heading=h.3znysh7)

[**>\_** Diferentes tipos de dados](#_heading=h.2et92p0)

[**>\_** Tipos de dados especiais](#_heading=h.tyjcwt)

[**>\_** O que são operadores?](#_heading=h.3dy6vkm)

[**>\_** Tipos de operadores](#_heading=h.1t3h5sf)

**>\_**

# O que são variáveis?



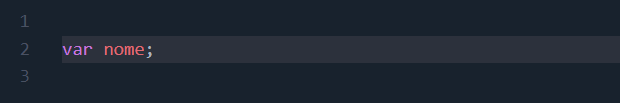
As **variáveis** são recursos das linguagens de programação que nos **permitem armazenar** informações. Podemos pensar em variáveis como se fossem gavetas: cada uma delas funciona como uma variável diferente, onde é possível guardar **coisas** dentro, e utilizar, quando necessário.

**>\_**

# Declaração e tipos de variáveis

Podemos observar uma **declaração** de variável no exemplo abaixo:

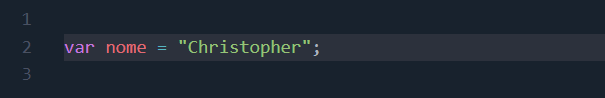




Para iniciar a declaração, **utilizamos** a palavra reservada **var**, em seguida atribuímos o identificador **nome**,que usaremos quando quisermos referenciar a variável.

Também é possível **atribuir valores**, como no exemplo abaixo:

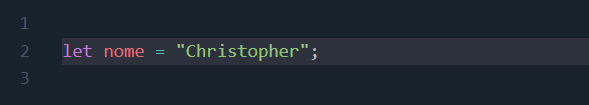




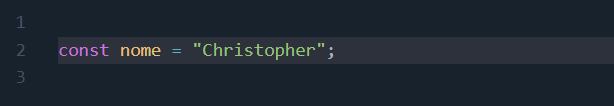
Nesta segunda declaração, **atribuímos** um valor **à variável**, diferente da primeira declaração. Com isso, a variável **nome** possui **uma informação** – um texto – que poderíamos acessar a qualquer momento.

No Javascript também há **outra forma** de declarar variáveis, **utilizando** a palavra reservada **let** ao invés de **var**. Em geral, a utilização de **let** **possui** mais **benefícios** em comparação ao **var**, por isso, devemos utilizar, **por padrão**, **let** para declarar variáveis.





Também é possível declarar **variáveis** com valores **imutáveis** – que não mudam – utilizando **constantes**. Esse recurso é muito utilizado, e considerado uma boa prática, pois dessa forma podemos assegurar o valor da variável, **impedindo** qualquer **alteração**.



Com a utilização de constantes, **caso** o valor seja **alterado**, o compilador **irá gerar** um **erro**, impedindo a execução do algoritmo. Com isso tornamos nosso código mais seguro.

**>\_**

# O que são tipos de dados?

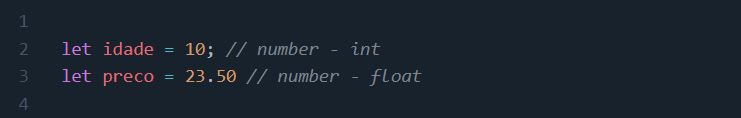
Na programação trabalhamos com **diferentes tipos de informações**. Devido a isso, existem diferentes **tipos de dados** para lidar com cada uma delas. Usar o tipo de dado correto **para** lidar com **certa informação** é essencial para um bom funcionamento do código.

**>\_**

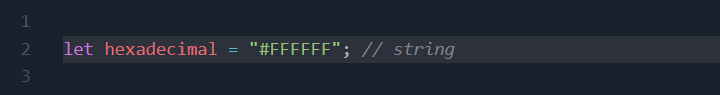
# Diferentes tipos de dados

Existem três tipos de dados **mais comuns** na programação, que são o **Number**, **String** e **Boolean**. Abaixo veremos uma descrição sobre cada um deles:

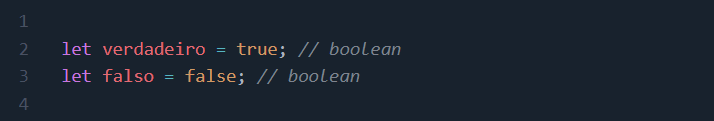
* **Number**: Tipo de dado numérico, muito utilizado para cálculos. Em algumas linguagens existem subdivisões dentro desse tipo, como **Int** indicando números inteiros, e **Float** indicando números com ponto flutuante (reais).



* **String**: Tipo de dado **textual**, que pode armazenar **qualquer** tipo de **caractere**, inclusive números. É importante ressaltar que por ser do tipo String, mesmo que possua números, **não** é recomendado **realizar cálculos**, pois o resultado pode acabar sendo uma concatenação;



* **Boolean**: Tipo de dado lógico, sendo **possível** apenas dois valores, **true (verdadeiro)** ou **false (falso)**. É utilizado para armazenar valores lógicos oriundos de comparações de quaisquer tipos de dados.

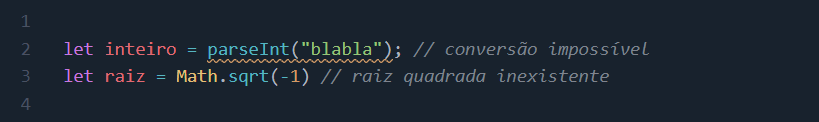


**>\_**

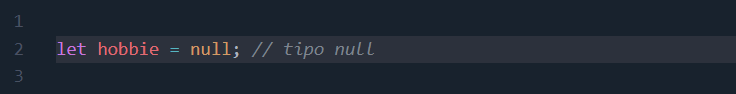
# Tipos de dados especiais

Também existem outros **tipos** de dados, chamados **especiais**. São eles:

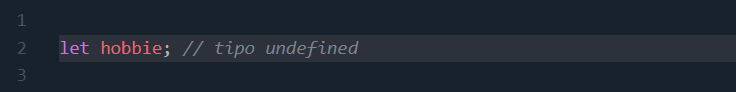
* **NaN**: **Not a Number (Não é um número) -** Esse tipo de dado **indica** uma operação em que o valor resultante **não** pode ser passado como **um número**;



* **Null**: **Indica** um tipo **vazio** – um elemento **sem** qualquer **valor** –, porém uma variável com **Null** no Javascript é considerada com valor, no caso um valor nulo;



* **Undefined**: Diferente do tipo **Null**, **Undefined** significa uma variável sem valor definido – com **ausência de valor** – e normalmente ocorre ao declarar uma variável sem **atribuição**.



**>\_**

# O que são operadores?

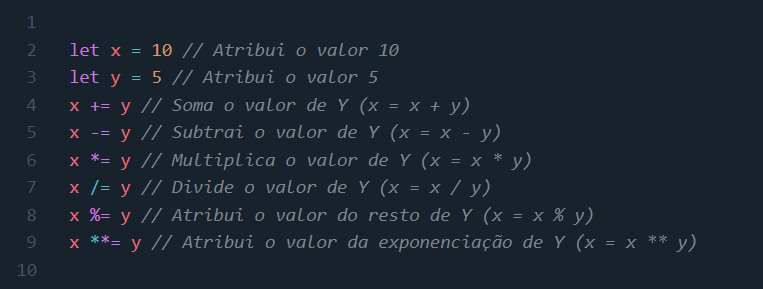
Os **operadores** são recursos que **nos** **permitem** manipular variáveis, **realizar** comparações e **operações** com seus valores.

**>\_**

# Tipos de Operadores

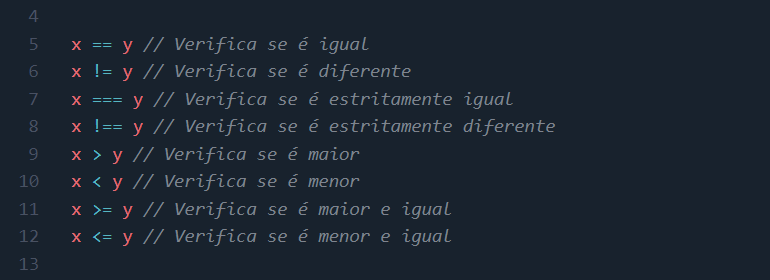
Existem diversos tipos de operadores, os mais comuns são os de **Atribuição**, **Comparação**, **Aritméticos**, **Lógicos** e **Concatenação**.

* **Atribuição**: **Atribui** valores às variáveis;

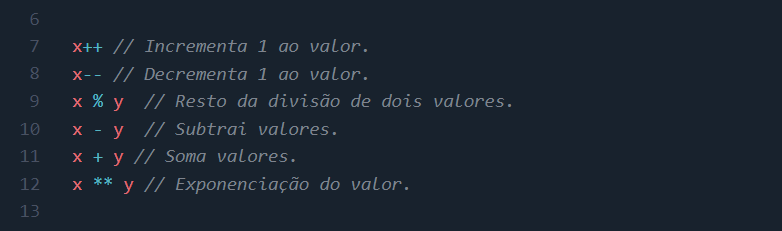


* **Comparação**: Realizam **comparações** que retornam **true** ou **false;**

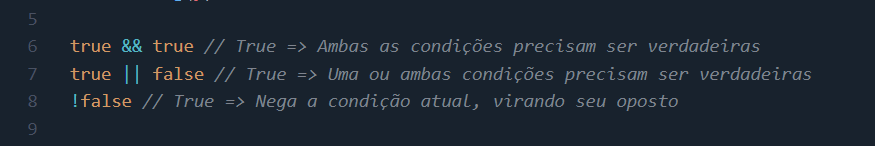




* **Aritméticos**: Realizam operações como **Soma**, **Subtração** etc;



* **Lógicos**: Verificam comparações e retornam **true** ou **false**. São utilizados os sinais **&&** (Ambos valores verdadeiros), **||** (Ambos ou um valor verdadeiro) e **!** (Nega a condição);



* **Concatenação: Concatena** textos – juntam suas partes – **em** apenas **uma**.



