

# Tipos de dados

No JavaScript e em qualquer outra linguagem de programação temos tipos de dados que podem ser usados, embora o JavaScript seja muito flexível em relação aos tipos, eles existem, e você deve saber quais são.

No JavaScript os tipos não são informados, ele entende sozinho o tipo de um dado, isso traz mais flexibilidade e rapidez, porém requer maior atenção em informar corretamente os dados nos momentos de informá-los.

Você deve ter lembrado que no português é preciso informar o tipo de uma variável criada, no JavaScript isso não ocorre, porém ainda sim os dados possuem tipos, vamos ver quais são aqui:

```
// Tipo String
const nome = "Lucas";
const sobrenome = 'Santos';
const cidade = `Jundiaí`;
// String com variáveis
const apresentacao = `Meu nome é ${nome} ${sobrenome} e eu moro em ${cidade}`;

// Tipo Number
let idade = 23;
let altura = 1.80;
let saldoBancario = -500;
let IDHJundiai = 0.8771;

// Tipo Boolean
const intrutor = true;
const triste = false;

// String com variáveis
const apresentacao2 = `Meu nome é ${nome} ${sobrenome}, tenho ${idade} anos e moro em ${cidade}`;

console.log(apresentacao);
console.log(apresentacao2);
```

*Dica: para não precisar abrir o navegador e ver os console.log você pode instalar a extensão **Code Runner** no vscode, após a instalação basta usar o comando **Ctrl+Alt+N** que os consoles serão exibidos no Output integrado do vscode.*

## Code Runner - Visual Studio Marketplace

Extension for Visual Studio Code - Run C, C++, Java, JS, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, CMD, BASH, F#, C#, VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml, R, AppleScript, Elixir, VB.NET, Clojure, Haxe, Obj-C, Rust, Racket, Scheme, AutoHotkey, AutoIt, Kotlin, Dart, Pascal, Haskell, Nim, <https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner>



No exemplo acima temos 3 tipos de dados: String, Number e Boolean, vamos falar um pouco sobre cada um desses três tipos.

Para verificar o tipo de um dado no JavaScript usamos o **typeof(dado)**.

Exemplo:

```
const nome = "Lucas";

console.log(typeof(nome));

// no console -> string
```

## String

Podemos usar strings no JavaScript com "aspas duplas", 'aspas simples' e com `esses acentos graves`, todas as formas são aceitas, porém a forma mais especial é essa última com acento grave, pois usando essa forma nós podemos inserir variáveis através do **\${ }**, além disso podemos escrever de forma mais livre no vscode dando ENTER sem problemas:

```
const stringExemplo = `
  Veja como podemos organizar
  melhor uma string no editor,
  isso não seria possível usando
  aspas duplas ou simples
`;
```

---

Então lembre-se que tanto faz você utilizar "aspas duplas" ou 'simples', o JavaScript irá entender, porém se quiser escrever uma grande string e organizar melhor você pode utilizar `esses acentos`, que chamamos de **template strings**, e ainda você pode usar variáveis através de **`${variável}`**.

---

## Number

Os números no JavaScript são o tipo Number, ao contrário de outras linguagens que possuem double, int, float e coisas do tipo, no JavaScript tudo é Number:

```
let idade = 23;
let altura = 1.80;
let saldoBancario = -500;
let IDHJundiai = 0.8771;

console.log(typeof (idade)); // no console -> number
console.log(typeof (altura)); // no console -> number
console.log(typeof (saldoBancario)); // no console -> number
console.log(typeof (IDHJundiai)); // no console -> number
```

---

Agora veremos um ponto de atenção ao JavaScript, por ele ter "tipagem fraca", observe o que acontece

```
let cidade = 'Jundiaí';
console.log(typeof (cidade)); // aqui no console irá aparecer string

cidade = 300;
console.log(typeof (cidade)); // aqui no console irá aparecer number
```

O que houve aí é que criamos uma variável como string mas em seguida pudemos atribuir um número a essa variável, por isso também é indicado usarmos **const** sempre que possível para as variáveis.

---

## Boolean

O tipo booleano tem apenas 2 possibilidades: true ou false, verdadeiro ou falso, podemos criar variáveis booleanas ou então verificar se algo é verdadeiro ou falso, veja exemplos:

```
console.log(3 > 5); // no console será mostrado false
```

Ou seja, essa é uma situação falsa, e esse tipo de coisa será muito útil nas condicionais, o if do JavaScript que veremos em breve.

---

## null e undefined

Esses tipos de dados são os tipos que não queremos muito ver em nossas aplicações, pois null se trata de algo realmente nulo e undefined significa geralmente que uma variável ainda não recebeu nenhum valor, observe um exemplo:

```
let qualquerCoisa;
console.log(qualquerCoisa); // no terminal -> undefined
```

Tivemos undefined como retorno pois a variável foi criada porém sem nenhum valor atribuído a ela.