

# TIPOS DE DADOS DO JAVASCRIPT

---

Introdução aos principais tipos de dados do Javascript

## Como funciona a tipagem no Javascript?

---

O Javascript, como toda linguagem de programação, tem sua própria estrutura de dados embutida. A tipagem no Javascript funciona de forma dinâmica, ou seja, não é necessário declarar o tipo de uma variável antes de atribuir um valor para ela. O tipo é definido quando o programa é processado.

Além disso, é possível reatribuir uma mesma variável com um tipo diferente.

Exemplo:

```
let address = 42;    // address é um Number agora
address = "bar";    // address é um String agora
address = true;     // address é um Boolean agora
```

## Tipos primitivos

---

Estes são os principais `valores primitivos` do Javascript. Diferente dos objetos, que possuem referência, ao alterar um dado primitivo você gera um novo valor, por isso são imutáveis.

São estes os mais comuns: `Boolean`, `undefined`, `Number` e `String`.

## Boolean

---

Dados do tipo **Boolean** representam uma entidade lógica e podem ser `true` (verdadeiro) ou `false` (falso).

Exemplo:

```
let weekend = false
```

## undefined

---

O valor **undefined** é atribuído automaticamente para uma variável que não tem atribuição no programa.

Exemplo:

```
let appointment  
//como nenhum valor for atribuído, appointment recebe undefined
```

# Number

---

O **Number** é um dado do tipo número. Ele pode ser **inteiro** ou **real (float)**:

```
let age = 12
// age é um número inteiro

let age = 12
// age é um número inteiro
```

**NaN - Not a Number** - usado para representar um dado que não é número. Exemplo:

```
let notANumber = (25 * "casa")
// notANumber = NaN, pois é impossível multiplicar um número por uma string alfanumérica
```

**Infinity** é um valor simbólico que representa o infinito. Precisa ser declarado com o 'I' maiúsculo.

```
let numberInfinity = (42 / 0)
// numberInfinity = Infinity
```

# String

---

A **String** é uma cadeia de caracteres, ou seja, um texto.

Para atribuir uma string a uma variável, você pode usar:

```
' ' // aspas simples  
" " // aspas duplas  
` ` // template literals, para incluir outras variáveis e expressões dentro da própria string
```

Você pode declarar da seguinte forma:

```
let name = "Mayk Brito"  
let name = 'Mayk Brito'  
let name = `Mayk ${ 1 + 1 }` // aqui o resultado é Mayk 2
```

## Tipos estruturais

---

Dados do `tipo estrutural` são diferentes dos tipos primitivos, pois apresentam uma “estrutura” com atributos e métodos.

São estes os mais comuns: `Object` e `Array`.



# Object

---

Dados do tipo **Object** são objetos e possuem atributos (propriedades) e métodos (funcionalidades).

Object é declarado da seguinte forma:

```
// aqui criamos um Object copo
const copo = {
  material: "vidro", //atributo String
  capacidade: 250, // atributo Number
  cheio: function() {
    // aqui podemos declarar uma função/método que será atribuído ao Object
  }
}
```

# Array

---

Um dado do tipo **Array** representa um agrupamento de dados em forma de lista.

Um Array pode ser declarado da seguinte forma:

```
// aqui criamos um Array shoppingList  
const shoppingList = [  
  "leite",  
  "ovos",  
  "chocolate",  
  2,  
  3,  
]
```

## Tipo primitivo estrutural

---

Dados do tipo **null** são `primitivos estruturais`, pois tem um valor (vazio) que são imutáveis e, também, podem representar uma estrutura vazia.

Exemplo:

```
let appointment = null
```

## Referências bibliográficas

---

**Estrutura de dados do Javascript**, MDN web docs. Acesso em 25 de julho de 2022. Disponível em <  
[https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Data\\_structures](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Data_structures) >