

## Array Const()

tem-se prática comum declarar array com const.  
um array declarado com const não pode ser alterado, reatribuído.

↳ `const cars = ['porsche', 'volvo']`

ERRO ⇔ `cars = ['porsche', 'volvo', 'BMW']`

Podemos adicionar um novo item ao array usando o método `push()`

## Date Objects

`Date()` nos permite trabalhar com datas / hora do sistema

ano: 2023 → `getFullYear()`

mês: 8 → `getMonth()`

dia: 33 → `getDate()`

horas: 33 → `getHours()`

minutos: → `getMinutes()`

segundos: → `getSeconds()`

## Date formats

Há 3 formats:

ISO date — "2015-03-25" \* formato padrão

Short date — "03/25/2015"

long date — "mar 25 2015"

## new Date() Constructor

Em JavaScript os objetos de data são criados com

`new Date()`

`new Date()` retorna um objeto de data com a data e hora atuais.

```
const diaHoje = new Date()
```

```
diaHoje.getFullYear()
```

```
diaHoje.getMonth()
```

```
diaHoje.getDate()
```

```
diaHoje.getDay()
```

```
diaHoje.getHours()
```

```
diaHoje.getMinutes()
```

```
diaHoje.getSeconds()
```

```
diaHoje.getMilliseconds()
```

```
diaHoje.getTime()
```

## Set Date methods

O método set permite definir valores de datas para um objeto Date()

## Math Object

Este método permite que você execute operações matemáticas em números.

- métodos para ~~arredondar~~ arredondar:

`Math.round(x)` arredondando para o inteiro mais próximo

`Math.ceil(x)` arredondando para cima

`Math.floor(x)` arredondando para baixo

`math.pow(x,y)` retorna o número elevado

↳  $x^y$  potenciação

`math.pow(8,2)` → saída: 64

↳  $8^2 = 8 \cdot 8 = \underline{64}$

`math.sqrt()` retorna raíz quadrada

`math.sqrt(x)` → `math.sqrt(64)` saída: 8

`math.abs(x)` → retorna o valor absoluto (positivo) de  $x$ .

`math.abs(-4.7)` saída: 4.7

`math.random()` retorna um número aleatório entre 0 e 1.