

Operadores e Lógica de Programação

Operadores Aritméticos

Operadores Aritméticos



- Soma

```
const primeiroValor = 10
```

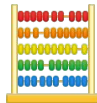
```
const segundoValor = 20
```

```
const resultado = primeiroValor + segundoValor + 4
```

```
console.log(resultado) // 34
```

Vamos ver na prática! 

Operadores Aritméticos



- Subtração

```
const primeiroValor = 10
```

```
const segundoValor = 20
```

```
const resultado = primeiroValor - segundoValor
```

```
console.log(resultado) // -10
```

Vamos ver na prática!



Operadores Aritméticos



- Multiplicação

```
const primeiroValor = 10
```

```
const segundoValor = 25
```

```
const resultado = primeiroValor * segundoValor
```

```
console.log(resultado) // 250
```

Vamos ver na prática!



Operadores Aritméticos



- Divisão

```
const primeiroValor = 345
```

```
const segundoValor = 10
```

```
const resultado = primeiroValor / segundoValor
```

```
console.log(resultado) // 34.5
```

Vamos ver na prática!



Operadores Aritméticos



- **Resto da Divisão**

- Existem casos em que a divisão não dá um número inteiro (sem vírgula)
- Quando acontece isso, dizemos que **há um resto na divisão**
- Por exemplo: dividir 20 por 3:
 - Dá o resultado 6, com resto 2
 - $20 = 3 \times 6 + \mathbf{2}$

Operadores Aritméticos

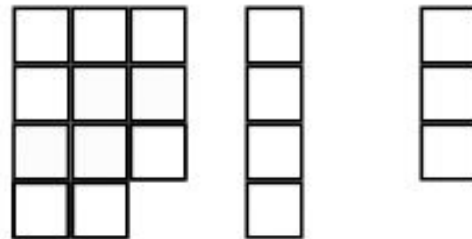


- Resto da Divisão

```
const restoDaDivisao = 11 % 4
```

```
console.log(restoDaDivisao) // 3
```

Modulo operation



$$11 \bmod 4 = 3$$

Vamos ver na prática!



Operadores Aritméticos



- Às vezes, podemos querer alterar o valor de uma variável fazendo alguma conta **com ela própria**
- Por exemplo: vamos supor que temos uma variável de idade e que a pessoa fez aniversário

```
let minhaIdade = 26
```

```
minhaIdade = minhaIdade + 1
```

Operadores Aritméticos



- Uma possível simplificação

```
let resultado = 100
```

```
resultado = resultado + 20    ➡    resultado += 20
```

```
resultado = resultado - 10    ➡    resultado -= 10
```

```
resultado = resultado * 5     ➡    resultado *= 5
```

```
resultado = resultado / 10    ➡    resultado /= 10
```

Vamos ver na prática! 

Comparadores

Comparadores 🧤

- Comparadores são operadores que permitem **comparar** o valor das variáveis entre si
- São eles:
 - Igual: `===`
 - Diferente: `!==`
 - Maior e maior igual: `>` e `>=`
 - Menor e menor igual: `<` e `<=`

Comparadores 🧤

- Quando fazemos uma comparação, o nosso resultado sempre vai ser verdadeiro ou falso
 - **Ex:** 1 é igual a 1? SIM ou NÃO? (verdadeiro ou falso)
- Por isso, resultado de uma comparação é sempre um **booleano** (true ou false)
- Quando a comparação for correta, o resultado é **true**. Caso contrário, é **false**.

Comparadores 🧤

- `===` verifica se o valor e o tipo são iguais

```
"1" === "2" // false, são diferentes  
"2" === "2" // true, são iguais  
2 === "2"   // false, tipo diferente
```

```
const condicao = 1 === 2  
// o valor que sai da comparação  
// pode ser guardado em uma variável  
// nesse caso, condicao === false
```

Vamos ver na prática! 🧐

Comparadores 🧤

- `!==` verifica se o valor e o tipo são diferentes

```
"1" !== "2" // true, são diferentes  
"2" !== "2" // false, são iguais  
2 !== "2" // true, tipos diferentes  
  
const condicao = 1 !== 2 // true
```

Vamos ver na prática! 🔬

Comparadores 🧤

- **> e >=**
 - Pode ser usado com **numbers**
 - **>=**: retorna **true** se os números envolvidos forem **iguais** ou se o primeiro for **maior** que o segundo
 - **>**: retorna **true só** se o primeiro for **maior** que o segundo

Comparadores 🧤

- `>` e `>=`

```
1 > 2    // false, porque 1 é menor que 2
2 > 2    // false, porque 2 é igual a 2
3 > 2    // true, porque 3 é maior que 2
```

```
1 >= 2   // false, porque 1 é menor que 2
2 >= 2   // true, porque 2 é igual a 2
3 >= 2   // true, porque 3 é maior que 2
```

Vamos ver na prática! 🔬

Comparadores 🧤

- **< e <=**
 - Também pode ser usado com **numbers**
 - **<=**: retorna **true** se os números envolvidos forem **iguais** ou se o primeiro é **menor** que o segundo
 - **<**: retorna **true só** se o primeiro é **menor** que o segundo

Comparadores 🧤

- < e <=

```
1 < 2 // true, porque 1 é menor que 2
2 < 2 // false, porque 2 é igual a 2
3 < 2 // false, porque 3 é maior que 2
```

```
1 <= 2 // true, porque 1 é menor que 2
2 <= 2 // true, porque 2 é igual a 2
3 <= 2 // false, porque 3 é maior que 2
```

Vamos ver na prática! 🔬

Operadores Lógicos

Operadores Lógicos

- São operadores especiais usados entre booleanos
- Retornam um valor booleano
- Existem 3 importantes:
 - Operador E: **&&**
 - Operador Ou: **||**
 - Operador Não/Negação: **!**

Operadores Lógicos

- **Operador E (&&):** retorna **true** se, e somente se, os **todos booleanos** envolvidos também forem **true**

a	b	a && b
V	V	V
V	F	F
F	V	F
F	F	F

```
true && true    // resultado = true
true && false   // resultado = false
false && true   // resultado = false
false && false  // resultado = false
```

Vamos ver na prática! 

Operadores Lógicos

- **Operador OU (||)**: retorna **false** se, e somente se, **todos** booleanos envolvidos também forem **false**

a	b	a b
V	V	V
V	F	V
F	V	V
F	F	F

```
true || true // resultado = true
true || false // resultado = true
false || true // resultado = true
false || false // resultado = false
```

Vamos ver na prática! 

Operadores Lógicos

- **Operador NÃO (!):** sempre retorna o booleano oposto

a	!a
V	F
F	V

```
!true // resultado = false
```

```
!false // resultado = true
```

Vamos ver na prática! 

Dúvidas?

