

AULA 4 - Operadores e Lógica de Programação

LÓGICA DE PROGRAMAÇÃO

- Chamamos esse passo-a-passo de instruções passadas para o computador de **Algoritmo**

- Um Algoritmo é, portanto, uma sequência de passos que visam atingir um objetivo bem definido

> Receita de um bolo, trocar o pneu de um carro

- O algoritmo vai representar o raciocínio lógico envolvido na resolução do problema

- Antes mesmo de começar a programar, você pode pensar em **COMO** resolver o seu problema

- Não se preocupe com o javascript nesse momento! Faça desenhos, diagramas, escreva em linguagem natural a sua ideia

- Temos algumas ferramentas para nos ajudar nesse momento de pensar sobre o problema:

- ☐ **Descrição textual**

- ☐ **Pseudocódigo**

- ☐ **Fluxogramas**

PSEUDOCÓDIGO

Esse é um passo entre a linguagem natural e o código

- É um texto estruturado de maneira mais próxima à linguagem de programação, mas você não precisa saber as palavras e sintaxes corretinhas

ler nome

let mensagem ← `Oi eu me chamo {nome}`

imprimir mensagem

FLUXOGRAMA

- Um fluxograma é um diagrama que nos ajuda a entender a ordem em que cada coisa acontece na nossa solução de uma forma visual

Operadores Aritméticos

- Como vimos na aula passada, podemos ter variáveis que são do tipo número
- E quando falamos de números, lembramos da nossa querida matemática!

SOMA

```
const primeiroValor = 10
```

```
const segundoValor = 20
```

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

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

SUBTRAÇÃO

```
const primeiroValor = 10
```

```
const segundoValor = 20
```

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

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

MULTIPLICAÇÃO

```
const primeiroValor = 10
```

```
const segundoValor = 25
```

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

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

DIVISÃO

```
const primeiroValor = 345
```

```
const segundoValor = 10
```

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

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

● 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

■ $20 = 3 \times 6 + 2$

>>Resto da Divisão

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

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

>> 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 minhaldade = 26 minhaldade = minhaldade + 1
```

• 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
```

Fixação

> Operadores Aritméticos

+

-

*

/

%

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 <=

● 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**.

Operadores Lógicos

● São operadores especiais usados entre booleanos

● Retornam um valor **booleano**

● Existem 3 importantes:

○ Operador E: &&

○ Operador Ou: ||

○ Operador Não/Negação: !

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

true && true // resultado = **true**

true && false // resultado = false

false && true // resultado = false

false && false // resultado = false

● **Operador OU (||):** Retorna **false** se, e somente se, todos booleanos envolvidos também

forem **false**

true || true // resultado = true

true || false // resultado = true

false || true // resultado = true

false || false // resultado = **false**

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

!true // resultado = false

!false // resultado = true

● **Numbers**

> Operadores Aritméticos: +, -, *, / e %

○ **Comparadores:** ==, !=, > e >=, < e <=