

Operadores e Lógica de Programação

Labenu_



O que vamos ver hoje?

- Lógica de Programação
- Operadores Aritméticos
- Comparadores
- Operadores Lógicos



Lógica de Programação

Labenu_



Lógica de Programação 💡

- O computador consegue fazer milhões de contas por segundo, mas não é capaz de compreender coisas da forma que o cérebro humano faz
- Para que algo aconteça da maneira que a gente quer em um programa, precisamos explicar CADA PASSO
- Precisamos, então, **organizar** nossas ideias e transformar em **passos** que o computador entenda



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



Lógica de Programação 💡

- 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
- Isso vai te ajudar a entender como o problema funciona e **traduzir** a solução para código será mais fácil



Lógica de Programação 💡

- Temos algumas ferramentas para nos ajudar nesse momento de pensar sobre o problema:
 - Descrição textual
 - Pseudocódigo
 - Fluxogramas



Descrição Textual

- A ideia aqui é simplesmente **escrever o que você está pensando** para conseguir resolver o problema
- Isso ajuda a ter mais **clareza** do que está se passando pela sua cabeça e **registrar** esses pensamentos, o que será útil quando chegar o momento de programar
- Não existe uma regra! Faça da maneira que achar melhor para você



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

Labenu_



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!**
- À seguir vamos ver como fazer algumas operações básicas da matemática no Javascript



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
 - $20 = 3 \times 6 + 2$



Operadores Aritméticos

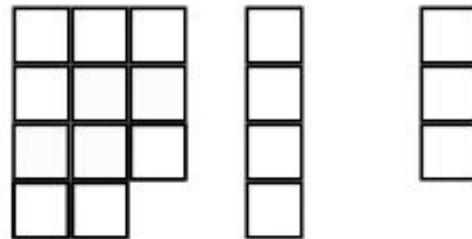


- Resto da Divisão

```
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!





Exercício 1

Faça as seguintes operações usando o computador:

1. Somar 3 com 4
2. Multiplicar 3 com 5 e dividir o resultado por 2
3. Subtrair 5 de 4 e multiplicar o resultado por -1
4. Determinar o resto da divisão de 234 por 5



Pausa para relaxar 🤪

5 min

- Operadores Aritméticos

- +
- -
- *
- /
- %



Comparadores

Labenu_



Comparadores 🧤

- Comparadores são operadores que permitem **comparar** 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! 🔬





Exercício 2

Crie duas variáveis que guardem dois números. Imprima na tela as seguintes mensagens:

1. O primeiro número é igual ao segundo? True/False
2. O primeiro número é diferente do segundo? True/False
3. O primeiro número é maior que o segundo? True/False
4. O primeiro número é menor que o segundo? True/False



Pausa para relaxar 🤔

10 min

- Comparadores

- `===`
- `!==`
- `> e >=`
- `< e <=`



Operadores Lógicos

Labenu_



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! 





Exercício 3

Antes de começar, crie 3 variáveis: a, b e c. Atribua os valores true, false e true, respectivamente

1. Realize a operação: **a && b**
2. Realize a operação: **b && c**
3. Realize a operação: **a && c**
4. Realize a operação: **a && b && c**



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! 





Exercício 4

Antes de começar, crie 3 variáveis: a, b e c. Atribua os valores true, false e true, respectivamente

a= true

b = false

c = true

1. Realize a operação: **a || b**
2. Realize a operação: **b || c**
3. Realize a operação: **a || c**
4. Realize a operação: **a || b || c**



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! 

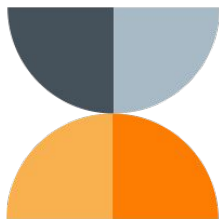


Pausa para relaxar 🧘

5 min

- Operadores Lógicos

- &&
- ||
- !





Exercício 5

Faça um programa que receba o nome, ano de nascimento de uma pessoa e o ano atual e mostre:

- O nome da pessoa com um olá
- A idade dessa pessoa
- Um true ou false que diz se ela é maior de idade
- Quantos anos ela terá em 2050

```
Nome: Laís  
Idade: 23  
É maior de idade? true  
Idade em 2050: 52
```



Resumo

Labenu_



Resumo

- **Numbers**

- Operadores Aritméticos:

- $+$, $-$, $*$, $/$ e $\%$

- Comparadores:

- $===$

- $!==$

- $> e \geq$

- $< e \leq$



Resumo

- **Operador E (&&):** Só devolve true se todos os booleanos forem true

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



Resumo

- **Operador Ou (||):** Só devolve false se todos forem false

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



Resumo

- **Operador !:** Devolve o oposto

a	!a
V	F
F	V



Dúvidas? 🧐

Labenu_





Obrigado(a)!