



# Aprenda a programar

Dia 4 – Aprendendo a programar (parte 2)

# O que veremos

## Retomando a aula passada

### Funções

**Estruturas de repetição:**  
a magia de processar e iterar

—

O que é iterar?

ForLoops

Iteração por métodos

# Retomando a aula passada

## Retomando a aula passada

Uma variável é como se fosse pegar um dado e colocar em uma caixinha que demos um nome, assim podemos chamar essa caixinha pelo nome quando for necessário.



As variáveis podem receber dados dos **tipos**:

- **Strings (texto)** - "Instrutor Tiago"
- **Number (número)** - 12
- **Booleans** - podem ser **true** ou **false**
- **Arrays (Listas)** - Listas de qualquer tipo de variável (inclusive listas de listas)
- **Mapas** - pares de chave e valor\*



*\* calma a gente vai ver juntos*

# Retomando a aula passada

2 tipos **muito** importantes são listas e mapas (objetos)

## Lista

```
let lista = ['Tiago', 'Jô', 'Thales']
```

posição	valor
0	Tiago
1	Jô
2	Thales

# Retomando a aula passada

2 tipos **muito** importantes são listas e mapas (objetos)

## Objeto (ou mapa)

```
let objeto = {  
  nome: 'Tiago',  
  cargo: 'Engenheiro Backend'  
}
```

chave	valor
nome	Tiago
cargo	Engenheiro Backend

# Retomando a aula passada

2 tipos **muito** importantes são listas e mapas (objetos)

## Lista

posição	valor
0	Tiago
1	Jô
2	Thales

```
lista[0] // retorna Tiago
```

## Objeto (ou mapa)

chave	valor
nome	Tiago
cargo	Engenheiro Backend

```
objeto['nome'] // retorna Tiago
```

```
objeto.nome // retorna Tiago
```



## Retomando a aula passada

**Se (idade > 18)**  
*é verdade?*

**Pode jogar**

*sim*

**Não pode jogar**

*não*

# Retomando a aula passada

**Vamos resolver um problema juntos?**

Começar abrindo o VSCode

# Funções

# Funções

Uma função é um bloco de código que podemos reutilizar

# Funções

Uma função é um bloco de código que podemos reutilizar  
Lembre-se que um programa é uma série de comandos



Commando A



Commando B

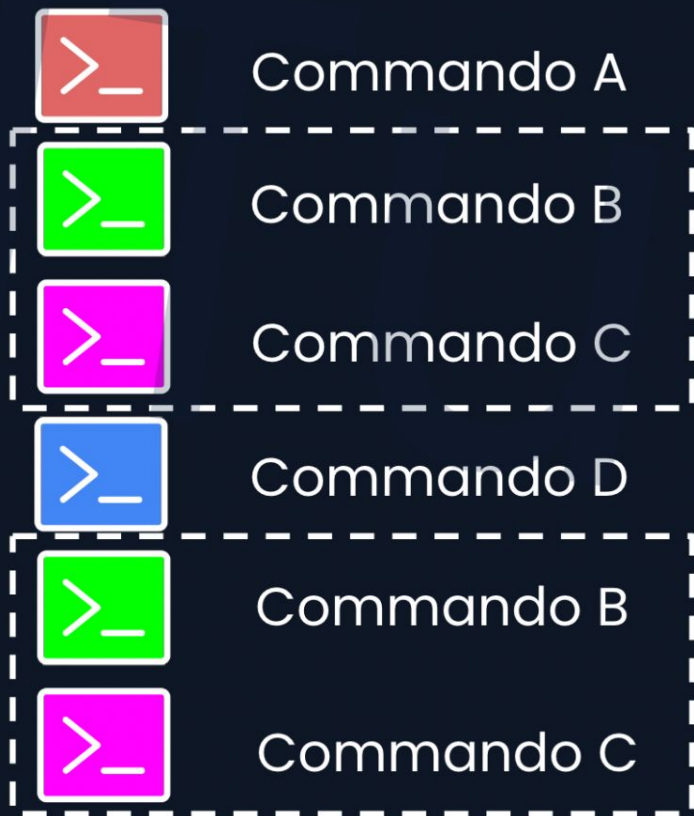


Commando C



Commando D

# Funções



As vezes, em um programa temos uma sequência de comandos que **fica se repetindo**.

# Funções

Para não ficar repetindo código, podemos criar uma função contendo aqueles comandos

## Função X



Commando B



Commando C

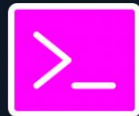
# Funções

E assim podemos chamar a função no código para executar aqueles comandos.

## Função X



Commando B



Commando C



Commando A



Commando B



Commando C



Commando D



Commando B



Commando C



# Funções

E assim podemos chamar a função no código para executar aqueles comandos.



# Funções

Mas como usar funções?

```
function nomeDaFuncao(parametro1, parametro2) {  
    // comandos dentro da função  
}
```

**Vamos para o VSCode!**

# **Estruturas de Repetição:** a magia de processar e iterar

**O que é iterar?**

# O que é iterar?

Iterar é **repetir** estruturas de código.

*Mas porque precisamos repetir?*



**Jogo de luta:** enquanto o adversário estiver “vivo”, vou bater nele.



**Fazer um ovo mexido:** enquanto o ovo não estiver misturado e pronto, vou mexer ele.



**Um cavalo comendo capim:** enquanto não estiver saciado, vou comer capim.

# O que é iterar?

Iterar é **repetir** estruturas de código.

*Mas porque precisamos repetir?*

Ou quando temos uma lista, e queremos repetir o comando para cada item dela enquanto ela não acabar.



Usando o comando **for**

# Usando o comando **for**

No comando **for**, temos uma variável que será um contador. O comando vai se repetir até que o contador chegue em um número desejado

```
for (let contador = 0; contador < 10; contador ++)
```

DE

ATÉ



Aumentando 1 por vez o contador



# Usando o comando `for`

Mas o JavaScript permite um jeito mais fácil  
focado em listas

```
for (let itemDaLista of lista)
```

O comando vai se repetir para cada item da lista, e podemos nos referir ao item que estamos passando com a variável declarada. No exemplo, a variável `itemDaLista`