

# Jovem Programador

Vlademir Jean Machado



# Aula 05

Vlademir Jean Machado

# Revisão

# Estrutura de controle

# Estrutura de controle

As estruturas de controle permitem que você controle o fluxo de execução do seu programa. Isso inclui instruções condicionais, como o if-else, que executam diferentes ações com base em condições, e loops, como o for e while, que repetem a execução de um bloco de código.

# Estrutura de controle

## Instrução Condicional (if-elif-else):

A instrução condicional if permite executar um bloco de código somente se uma condição for verdadeira. A instrução elif (opcional) é usada para testar condições adicionais, e a instrução else (opcional) é executada se todas as condições anteriores forem falsas.

```
idade = 18
if idade < 18:
    print("Você é menor de idade.")
elif idade == 18:
    print("Você tem 18 anos.")
else:
    print("Você é maior de idade.")
```

```
idade = 18
if idade < 18:
    print("Você é menor de idade.")
elif idade > 18:
    print("Você tem 18 anos.")
else:
    print("Você é maior de idade.")
```

# Estrutura de controle

**Loop While:** A estrutura while permite repetir a execução de um bloco de código enquanto uma condição for verdadeira.

```
contador = 0
while contador < 5:
    print(contador)
    contador += 1
```



```
contador = 0  
while contador < 5:  
    print(contador)  
    contador += 1
```

# Estrutura de controle

## Instruções de Controle de Loop:

- A instrução **break** é usada para interromper a execução de um loop antes que a condição de término seja alcançada.
- A instrução **continue** é usada para pular para a próxima iteração do loop, ignorando o restante do bloco de código atual.

```
contador = 0
while contador < 10:
    contador += 1
    if contador == 5:
        continue
    print(contador)
    if contador == 8:
        break
```

## Estrutu

```
contador = 0
while contador < 10:
    contador += 1
    if contador == 5:
        continue
    print(contador)
    if contador == 8:
        break
```

# Estrutura de controle

```
frutas = ["maçã", "banana", "laranja"]  
for fruta in frutas:  
    print(fruta)
```

**Loop For:** A estrutura for permite iterar sobre uma sequência (como uma lista, uma string ou um intervalo numérico) e executar um bloco de código para cada elemento da sequência.

# Vamos praticar

insira sua idade. Em seguida, usamos uma estrutura de controle **if-elif-else** para verificar em qual faixa etária a pessoa se encontra.

- Se a idade for menor que 18, a mensagem "**Você é menor de idade**" deve ser exibida.
- Se a idade for maior ou igual a 18 e menor que 65, a mensagem "**Você é adulto**" deve ser exibida.
- Caso contrário, ou seja, se a idade for maior ou igual a 65, a mensagem "**Você é idoso**" deve ser exibida



```
# Atividade: Verificação de Faixa Etária

# Solicita ao usuário que informe a idade
idade = int(input("Digite a sua idade: "))

# Verifica em qual faixa etária a pessoa se encontra
if idade < 18:
    print("Você é menor de idade.")
elif idade >= 18 and idade < 65:
    print("Você é adulto.")
else:
    print("Você é idoso.")
```

# Métodos em Lista

# Lista

**Listas (list):** Listas são coleções ordenadas e mutáveis de elementos. Elas podem conter diferentes tipos de dados e permitem armazenar múltiplos valores em uma única variável.

```
# Exemplo de lista  
numeros = [1, 2, 3, 4, 5]  
print(numeros) # Saída: [1, 2, 3, 4, 5]
```



# Lista

## **append(valor)**

- Adiciona um item ao final da lista.

```
lista = [1, 2]  
lista.append(3)    # [1, 2, 3]
```

# Lista

## **extend(iterável)**

- Adiciona todos os elementos de um iterável à lista.

```
lista = [1, 2]  
lista.extend([3, 4]) # [1, 2, 3, 4]
```

## Lista

### **insert(posição, valor)**

- Insere um item em uma posição específica.

```
lista = [1, 3]
```

```
lista.insert(1, 2) # [1, 2, 3]
```

# Lista

## **remove(valor)**

- Remove a primeira ocorrência do valor.

```
lista = [1, 2, 3, 2]
```

```
lista.remove(2) # [1, 3, 2]
```

# Lista

## pop([índice])

- Remove e retorna o item na posição (ou o último se nenhum índice for informado).

```
lista = [1, 2, 3]
```

```
lista.pop()           # retorna 3, Lista fica [1, 2]
```

```
lista.pop(0)          # retorna 1, Lista fica [2]
```

# Lista

## `index(valor)`

- Retorna o índice da primeira ocorrência do valor.

```
lista = ['a', 'b', 'c']  
lista.index('b') # 1
```

# Lista

## count(valor)

- Conta quantas vezes um valor aparece na lista.

```
lista = [1, 2, 2, 3]  
lista.count(2)  # 2
```

# Lista

## sort()

- Ordena a lista (ordem crescente, por padrão).

```
lista = [3, 1, 2]
```

```
lista.sort() # [1, 2, 3]
```



# Lista reverse()

- Inverte a ordem dos elementos na lista.

```
lista = [1, 2, 3]
```

```
lista.reverse() # [3, 2, 1]
```

# Lista

## **copy()**

- Cria uma cópia da lista.

```
lista1 = [1, 2, 3]  
lista2 = lista1.copy()
```

# Lista

## clear()

- Remove todos os elementos da lista.

```
lista = [1, 2, 3]  
lista.clear() # []
```

# Identificação

# Identação

Em Python, a indentação é uma parte fundamental da sintaxe da linguagem. Diferentemente de outras linguagens de programação, como C ou Java, que utilizam chaves ({} ) para definir blocos de código, Python utiliza a indentação para delimitar a estrutura e a organização do código.

# Identação

Em Python o código é dividido em BLOCOS e NIVEL.

- ◆ **Bloco:** Conjunto de linhas de código que pertencem a uma mesma estrutura, como if, while, for, entre outras.
- ◆ **Nível:** Determina a **profundidade** do código dentro de um bloco, e é definido por **espaços** ou **tabulações**.

Nível 0

Nível 1

Nível 2

```
# NÍVEL 0
```

```
senha = "1234"
```

```
# NÍVEL 0 → Início do bloco IF
```

```
if senha == "1234":
```

```
    # NÍVEL 1 → Dentro do primeiro IF
```

```
    print("Senha correta.")
```

```
    confirma = "1234"
```

```
    if confirma == senha:
```

```
        # NÍVEL 2 → Dentro do segundo IF (bloco dentro de bloco)
```

```
        print("Confirmação bem-sucedida.")
```

```
        print("Acesso autorizado.")
```

```
# NÍVEL 0
```

```
print("Fim do programa.")
```

*# BLOCO 1: Condicional if*

```
idade = 17
```

```
if idade >= 18:
```

```
    print("É maior de idade.")    # BLOCO 1
```

```
    print("Pode tirar CNH.")      # BLOCO 1
```

Bloco 1

*# BLOCO 2: Estrutura else*

```
else:
```

```
    print("É menor de idade.")    # BLOCO 2
```

```
    print("Ainda não pode tirar CNH.") # BLOCO 2
```

Bloco 2

*# BLOCO 3: Laço de repetição for*

```
for i in range(3):
```

```
    print("Repetição número", i)  # BLOCO 3
```

Bloco 3



