

Aula 04

Estrutura de repetição - While



Tópicos

- Estruturas de repetição
- Comando de repetição “While”
- Estruturas de repetição aninhadas
- Comandos break e continue

Estrutura de Repetição

As estruturas de repetição

- Estruturas de repetição são usadas para executar um bloco de código várias vezes. Em Python, existem dois tipos principais de laços de repetição:
 - for e
 - while.

```
[5] for i in range(5):  
    print("Contador é:", i)
```

```
⇒ Contador é: 0  
   Contador é: 1  
   Contador é: 2  
   Contador é: 3  
   Contador é: 4
```

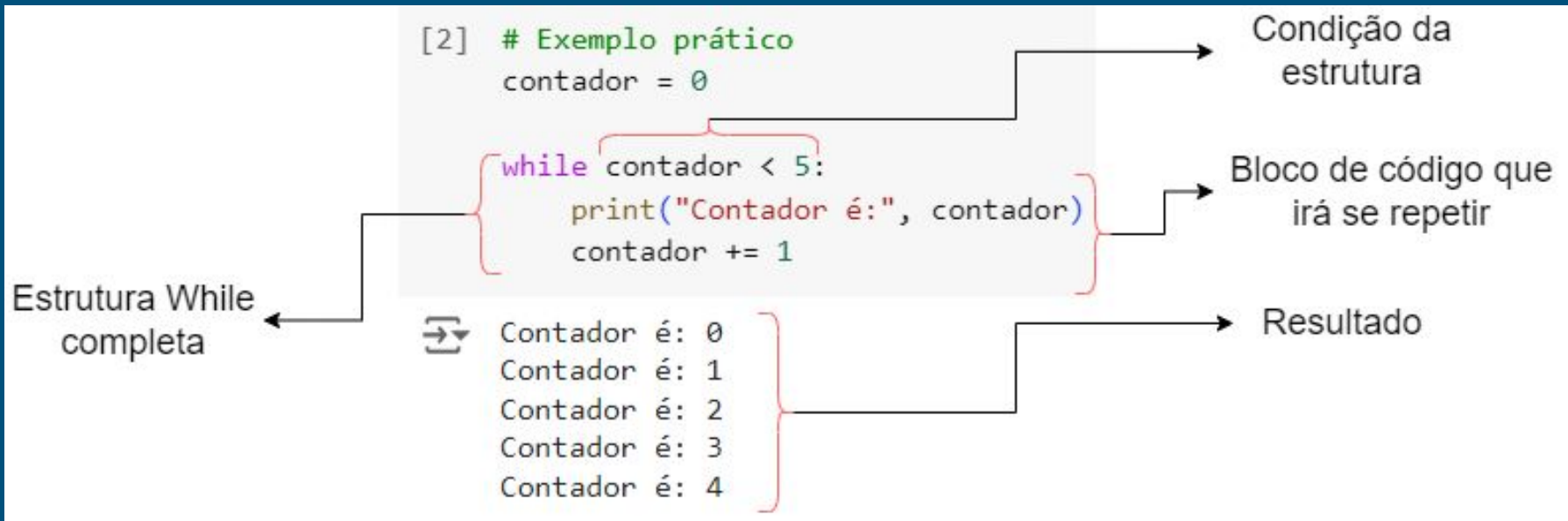
```
[2] # Exemplo prático  
    contador = 0  
  
    while contador < 5:  
        print("Contador é:", contador)  
        contador += 1
```

```
⇒ Contador é: 0  
   Contador é: 1  
   Contador é: 2  
   Contador é: 3  
   Contador é: 4
```

A estrutura While

- O laço while executa um bloco de código enquanto uma condição é verdadeira.
- Se a condição nunca se tornar falsa, o while continuará executando indefinidamente, resultando em um laço infinito.
- O comando “break” pode ser usado para sair do laço while antecipadamente, independentemente da condição.
- O comando “continue” pode ser usado para pular o restante do bloco de código atual e iniciar a próxima iteração do laço.

A estrutura While



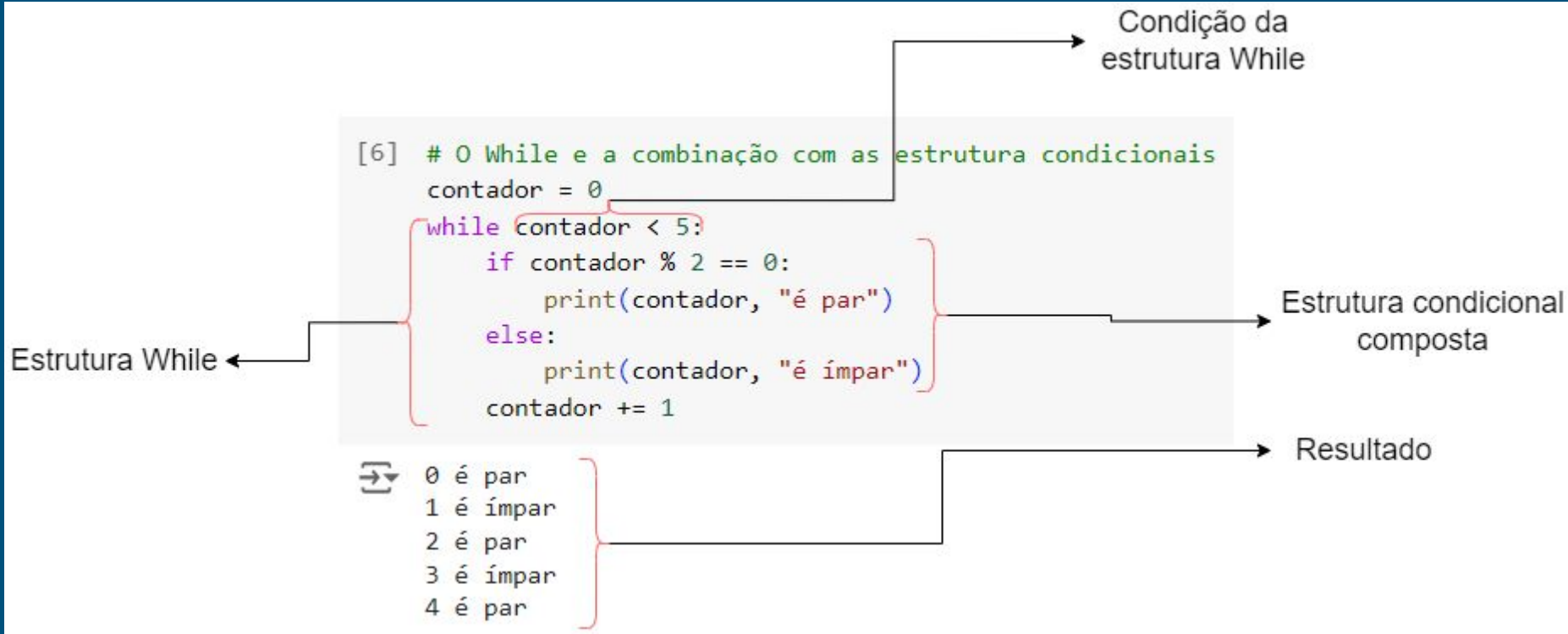
O while e a estrutura condicional

- O laço while pode ser combinado com estruturas condicionais (if, elif, else) para criar fluxos de controle mais complexos.

```
[6] # O While e a combinação com as estrutura condicionais
    contador = 0
    while contador < 5:
        if contador % 2 == 0:
            print(contador, "é par")
        else:
            print(contador, "é ímpar")
        contador += 1
```

```
⇒ 0 é par
   1 é ímpar
   2 é par
   3 é ímpar
   4 é par
```

Entendendo a estrutura



Estrutura “While” com “else”

- Laços while com else
- Um bloco else pode ser associado ao laço while. O bloco else é executado quando a condição do while se torna falsa.



```
# Estrutura While com else
contador = 0
while contador < 5:
    print("Contador é:", contador)
    contador += 1
else:
    print("Laço while terminou.")
```



```
Contador é: 0
Contador é: 1
Contador é: 2
Contador é: 3
Contador é: 4
Laço while terminou.
```

Estruturas de repetição aninhadas

O “while” aninhado

- O laço “while” podem ser aninhado dentro de outro laço “while” ou “for”, permitindo a criação de estruturas de repetição mais complexas.



```
# Trabalhando a estrutura While de maneira aninhada
i = 1
while i <= 3:
    j = 1
    while j <= 3:
        print(f"i = {i}, j = {j}")
        j += 1
    i += 1
```



```
i = 1, j = 1
i = 1, j = 2
i = 1, j = 3
i = 2, j = 1
i = 2, j = 2
i = 2, j = 3
i = 3, j = 1
i = 3, j = 2
i = 3, j = 3
```

O “while” e o “for” aninhados

- Este exemplo mostra como combinar um laço “for” dentro de um laço “while” para realizar uma operação específica. Aqui, para cada número na lista números, o laço “while” controla a iteração sobre a lista, enquanto o laço “for” interno realiza uma contagem decrescente a partir do número atual.

```
# Lista de números
numeros = [3, 2, 1]

# Laço while externo
indice = 0
while indice < len(numeros):
    num = numeros[indice]

    # Laço for interno
    for i in range(num, 0, -1):
        print(f"Contando a partir de {num}: {i}")

    indice += 1
```

```
Contando a partir de 3: 3
Contando a partir de 3: 2
Contando a partir de 3: 1
Contando a partir de 2: 2
Contando a partir de 2: 1
Contando a partir de 1: 1
```

Comandos break e continue

Comandos break e continue

- Ambos os comandos são úteis para controlar o fluxo nos laços “while” e “for”, permitindo a implementação de lógica condicional para saída antecipada ou para pular partes do código dentro do laço.

Trabalhando com a estrutura “while”

```
[18] # Sai do laço quando contador for 5
      contador = 0
      while contador < 10:
          if contador == 5:
              break
          print(contador)
          contador += 1
```

⇒ 0
1
2
3
4

```
[19] # Pula a iteração atual se contador for par
      contador = 0
      while contador < 10:
          contador += 1
          if contador % 2 == 0:
              continue
          print(contador)
```

⇒ 1
3
5
7
9

Vamos praticar!
[Link](#)



Vamos exercitar!

[Link](#)

