

Programação I

Repetição Condicional e Repetição Contável

Igor Machado Coelho

14/04/2025

1 Repetição Condicional

Section 1

Repetição Condicional

Comando While

Executa o bloco de instruções enquanto a condição for verdadeira

Portugol

enquanto CONDIÇÃO

faça

INSTRUÇÃO 1;

INSTRUÇÃO 2;

...

INSTRUÇÃO N;

...

Comando While

Executa o bloco de instruções enquanto a condição for verdadeira

Portugol

enquanto CONDIÇÃO

faça

INSTRUÇÃO 1;

INSTRUÇÃO 2;

...

INSTRUÇÃO N;

...

Python

while CONDIÇÃO:

INSTRUÇÃO 1;

INSTRUÇÃO 2;

...

INSTRUÇÃO N;

...

Comando While

Executa o bloco de instruções enquanto a condição for verdadeira

Portugol

enquanto CONDIÇÃO

faça

INSTRUÇÃO 1;

INSTRUÇÃO 2;

...

INSTRUÇÃO N;

...

Python

while CONDIÇÃO:

INSTRUÇÃO 1;

INSTRUÇÃO 2;

...

INSTRUÇÃO N;

...

- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções é delimitado por indentação
- Deve haver algum processo dentro do bloco de comandos que torne a condição falsa para que a repetição seja encerrada
- **Pergunta:** Qual fluxograma representa esses códigos?

Exemplos com While

```
numero = 1
while numero <= 100:
    print(numero)
    numero = numero + 1
```

Exemplos com While

```
numero = 1
while numero <= 100:
    print(numero)
    numero = numero + 1

n = int(input("Digite um número:"))
numero = 1
while numero <= n:
    print(numero)
    numero = numero + 1
```


Exemplos com While

```
numero = 1
while numero <= 100:
    print(numero)
    numero = numero + 1

n = int(input("Digite um número:"))
numero = 1
while numero <= n:
    print(numero)
    numero = numero + 1

a = 5
while a == a:
    a = a + 1
print(a)
```

Mais exemplos com While

```
num = 100
pares = 0
while num <= 200:
    if num % 2 == 0:
        pares = pares + 1
    num = num + 1
print(pares)
```

Mais exemplos com While

```
num = 100
pares = 0
while num <= 200:
    if num % 2 == 0:
        pares = pares + 1
    num = num + 1
print(pares)
```

```
num = 100
pares = 0
continua = True

while continua:
    if num % 2 == 0:
        pares = pares + 1
    num = num + 1
    if num > 200:
        continua = False
print(pares)
```

Padrão while(True) / break

Em algumas situações, é útil ter um laço infinito:

```
while True:  
    # faça algo útil!
```

Mas como escapar desse laço infinito?

Padrão while(True) / break

Em algumas situações, é útil ter um laço infinito:

```
while True:  
    # faça algo útil!
```

Mas como escapar desse laço infinito?

```
num = 100  
contador_pares = 0  
while True:  
    if num % 2 == 0:  
        contador_pares = contador_pares + 1  
    num = num + 1  
    if num > 200:  
        break  
print(contador_pares)
```

Padrão while(True) / break / continue

Em algumas situações, é útil ter um laço infinito:

```
num = 99
contador_pares = 0
while True:
    num = num + 1
    if num > 200:
        break
    if num % 2 != 0:
        continue # o que isso faz?
    contador_pares = contador_pares + 1
print(contador_pares)
```

Pergunta: Por que começamos com 99?

Padrão for-range em repetições contáveis

A repetição contável, ou laço tipo `for-range`, é muito utilizado em python. Esse conceito será melhor explicado no futuro, porém apresentamos brevemente a estrutura de um `for-range` básico (simplesmente, uma variável e um intervalo de continuação/incremento):

```
numero = 1
while numero <= 100:
    print(numero)
    numero = numero + 1
```

Padrão for-range em repetições contáveis

A repetição contável, ou laço tipo `for-range`, é muito utilizado em python. Esse conceito será melhor explicado no futuro, porém apresentamos brevemente a estrutura de um `for-range` básico (simplesmente, uma variável e um intervalo de continuação/incremento):

```
numero = 1
while numero <= 100:
    print(numero)
    numero = numero + 1

for numero in range(1, 101):
    print(numero)
```

A estrutura é bastante compacta e muito utilizada na prática! Porém, ela mascara alguns comportamentos internos do python que serão melhor discutidos no futuro.

Dica: Busque utilizar e praticar o laço do tipo `while`, pois ele permite construir *qualquer tipo* de repetição, não apenas contáveis como `for`.

Pergunta: O que acontece com `break` e `continue` em um laço `for`?
Pratique!

Laços Aninhados

Seguir com material do prof Yuri

Reprodução do material

Esses slides foram escritos utilizando pandoc, segundo o tutorial ilectures:

- <https://igormcoelho.github.io/ilectures-pandoc/>

Exceto expressamente mencionado (com as devidas ressalvas ao material cedido por colegas), a licença será Creative Commons.

Licença: CC-BY 4.0 2025

Igor Machado Coelho

This Slide Is Intentionally Blank (for goomit-mpx)