

# ESTRUTURAS DE REPETIÇÃO

IFBA – Campus Ilhéus  
Curso: Back End com Python

# Objetivo da Aula

- Apresentar as formas de repetição para executar blocos de código diversas vezes.

# Laço while

- Executa um bloco de código enquanto uma condição for verdadeira.

- Sintaxe:

while condição:

    # bloco de código

# Laço while

- Exemplo:

```
contador = 1
```

```
while contador <= 5:
```

```
    print(f"Contador: {contador}")
```

```
    contador += 1
```

- Cuidado com loops infinitos!*

# Laço while

- Exemplo com condição controlada pelo usuário:

```
resposta = 's'
```

```
while resposta == 's':
```

```
    nome = input('Digite seu nome: ')
```

```
    print(f'Olá, {nome}!')
```

```
    resposta = input('Deseja continuar? (s/n):')
```

**Exercícios while**

# Laço for

- Utilizado para coisas finitas, quantidade previsível.
- É usado para percorrer uma **sequência de elementos iteráveis** (como listas, strings,...)
  - A cada volta do laço, ele pega um elemento da sequência e executa o bloco de código com ele.
- Sintaxe:  
for variável in sequência:  
    #bloco de código que será repetido

# Laço for

- Exemplo com string:

```
for letra in "Python":  
    print(letra)
```

- Exemplo com lista:

```
nomes = ["Maria", "Aline", "Ana"]  
for nome in nomes:  
    print("Olá, ", nome)
```

- Exemplo com range:

```
for i in range(5):  
    print("Repetição número: ", i)
```

# Função range()

- Cria uma sequência de números inteiros.

- Formas:

- `range(fim)` → de 0 até fim-1
- `range(início, fim)` → de início até fim-1
- `range(início, fim, passo)` → define o salto

- Exemplo:

```
print(list(range(5)))      # [0, 1, 2, 3, 4]
```

```
print(list(range(2, 6)))   # [2, 3, 4, 5]
```

```
print(list(range(10, 0, -2))) # [10, 8, 6, 4, 2]
```



# Laço for com range

- Com início e fim definidos:

```
for i in range(2, 6):  
    print(i)
```

- Com passo:

```
for i in range(10, 0, -2):  
    print(i)
```

# Laços Aninhados

- São usados quando é necessário percorrer estruturas mais complexas, como matrizes ou combinar itens de duas listas.
- Exemplo:

```
for i in range(1, 4):  
    for j in range(1, 4):  
        print(f"i={i}, j={j}")
```
- Isso vai gerar todas as combinações entre 1 e 3 para i e j.

# Comandos break e continue

- `break` → Interrompe o laço imediatamente.

```
while True:
```

```
    nome = input("Digite seu nome (ou 'sair'): ")
```

```
    if nome == "sair":
```

```
        break
```

```
    print(f"Olá, {nome}")
```

# Comandos break e continue

- ⦿ continue → Pula para a próxima repetição.

```
for i in range(1, 6):  
    if i == 3:  
        continue # pula o número 3  
    print(i)
```

# Exercícios while

1. Crie um programa que solicite um número e conte de 1 até esse número usando while.
2. Crie um programa que peça números ao usuário até que ele digite 0. Ao final, mostre a média dos números digitados.

**Voltar**

# Exercícios for

3. Faça um programa que imprima os múltiplos de 5, de 0 até 25.
4. Crie um programa que imprima a tabuada do número 3.
5. Usando for e range com passo 1, mostre apenas os números pares de 1 a 10, ignorando os ímpares com continue.

# Exercícios laços aninhados

6. Crie um programa que imprima uma tabela de multiplicação de 1 a 5 usando dois laços for aninhados (cada bloco de multiplicação corresponde a uma “tabuada” de um número de 1 a 5.).

$$1 \times 1 = 1$$

$$1 \times 2 = 2$$

$$1 \times 3 = 3$$

$$1 \times 4 = 4$$

$$1 \times 5 = 5$$

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$3 \times 1 = 3$$

$$3 \times 2 = 6$$

$$3 \times 3 = 9$$

$$3 \times 4 = 12$$

$$3 \times 5 = 15$$

$$4 \times 1 = 4$$

$$4 \times 2 = 8$$

$$4 \times 3 = 12$$

$$4 \times 4 = 16$$

$$4 \times 5 = 20$$

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$