

# Estruturas de Repetição no Python

`while`, `for`, `do`  
`while`, aninhamento e  
listas

# O que são estruturas de repetição?

Blocos de código que se repetem várias vezes.

Evitam repetições manuais de comandos.

Tipos principais no Python:

- while
- for
- do while (simulação)

# Estrutura while

- `while` condição:  
    # bloco de código
- Repete enquanto a condição for verdadeira.
- Cuidado com **loop infinito**.

# While Exemplo.

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

-  Saída:
- 1
- 2
- 3
- 4
- 5

# Estrutura for

- `for item in sequência:`  
    `# bloco de código`
- Percorre elementos de uma **sequência** (lista, string, range...).

# Exemplo for

- ```
for i in range(1, 6):  
    print(i)
```

-  Saída:

- 1  
2  
3  
4  
5

# Simulando do while

- Python não tem do while, mas pode ser simulado:
- ```
while True:  
    entrada = input("Digite algo (x  
para sair): ")  
    if entrada == 'x':  
        break
```
- Executa **pelo menos uma vez**.

# COMPARANDO ESTRUTURAS

| Tipo                  | Quando usar                                      |
|-----------------------|--------------------------------------------------|
| <code>while</code>    | Quando não se sabe quantas vezes irá repetir     |
| <code>for</code>      | Quando se sabe quantas vezes ou há uma sequência |
| <code>do while</code> | Executar ao menos uma vez antes da verificação   |



# Aninhando estruturas (for dentro de while)

- ```
x = 0
while x < 3:
    for y in range(2):
        print(f"x={x}, y={y}")
    x += 1
```
- Resultado: combinações de x e y.

# Aninhando while dentro de for

- ```
for i in range(3):  
    j = 0  
    while j < 2:  
        print(f"i={i}, j={j}")  
        j += 1
```

- **For com "do while"**

- ```
for i in range(2):  
    while True:  
        entrada = input("Digite 'ok' para  
continuar: ")  
        if entrada == 'ok':  
            break
```

- Força o usuário a digitar "ok" antes de prosseguir com o for.

# Introdução às Listas

- `frutas = ["maçã", "banana", "laranja"]`
- Lista = coleção de elementos
- Acesso por índice: `frutas[0]`

# While com lista

- ```
frutas = ["maçã", "banana", "laranja"]  
i = 0  
while i < len(frutas):  
    print(frutas[i])  
    i += 1
```

-  Saída:

- maçã  
banana  
laranja

# Usando for com lista

- ```
for fruta in frutas:  
    print(fruta)
```

# While + for com lista

```
• listas = ["a", "b"], ["c", "d"]  
i = 0  
while i < len(listas):  
    for item in listas[i]:  
        print(item)  
    i += 1
```

# Recapitulando

- ✓ `while`: enquanto condição for verdadeira
- ✓ `for`: percorre sequência
- ✓ `do while`: simulado com `while True` e `break`
- ✓ Aninhamento possível entre todas as estruturas
- ✓ Listas tornam as repetições mais poderosas



