

# Ficha Rápida – Aula 4: Laços de Repetição (for)

## Objetivo da aula

Ensinar como **repetir instruções** em Python usando o laço **for**. O laço for permite executar blocos de código várias vezes de forma automática, percorrendo sequências numéricas ou coleções como listas.

## O que você aprendeu

- A estrutura básica do laço for em Python: for item in colecao:.
- A função **range()** gera uma sequência de números para iterar.
- O range possui três parâmetros: início, fim (não incluso) e passo.
- Listas podem ser percorridas com for para acessar cada elemento.
- Podemos combinar **for** com **if** para processar elementos condicionalmente.
- Listas podem conter diferentes tipos de dados (int, float, str, bool).

## Exemplo com range()

```
# Imprimir de 0 a 4
for i in range(5):
    print(i)

# Imprimir de 1 a 10
for i in range(1, 11):
    print(i)

# Imprimir de 2 a 10 pulando de 2 em 2
for i in range(2, 11, 2):
    print(i)
```

## Exemplo percorrendo listas

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

# Lista com tipos mistos
dados = [1, "Python", True, 2.5]
for dado in dados:
    print(dado)
```

## Exemplo combinando for + if

```
idades = [13, 15, 18, 30, 50, 75]

for idade in idades:
    if idade >= 18:
        print(f"{idade} é maior de idade")
    else:
        print(f"{idade} é menor de idade")
```

## Boas práticas com for

- Use nomes de variáveis no singular para o item atual e plural para a coleção (ex.: for idade in idades).
- Evite modificar a lista original dentro do for se não for necessário.
- Lembre-se que **range** não inclui o número final; use +1 se necessário.

Referência rápida criada com base no conteúdo da Aula 4. Use esta ficha para revisar loops for e percorrer coleções em Python.