

Revisão do For em Python

1. Estrutura básica do `for`

Explicação

O `for` é usado para percorrer sequências como listas, strings, ranges e outros iteráveis. Ele executa um bloco de código uma vez para cada item da sequência.

Exemplo

```
for i in [1, 2, 3]:  
    print(i)
```

Exercícios

1. Imprima os números de 1 a 5 usando um `for`.
2. Percorra uma lista de nomes e exiba cada nome.

2. `for` com `range()`

Explicação

`range()` gera uma sequência de números. Pode ter 1, 2 ou 3 argumentos.

Exemplo

```
for i in range(5):  
    print(i)
```

Exercícios

1. Imprima os números de 0 a 10.
2. Imprima apenas números pares de 2 a 20.

3. Percorrendo listas

Explicação

Listas são coleções ordenadas.

Exemplo

```
frutas = ["maçã", "banana", "uva"]
for fruta in frutas:
    print(fruta)
```

Exercícios

1. Percorra uma lista de 10 itens e imprima todos.
2. Dada uma lista de números, imprima os maiores que 10.

4. Percorrendo strings

Explicação

Strings são sequências de caracteres.

Exemplo

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

Exercícios

1. Conte quantas vogais existem em uma palavra.
2. Imprima cada letra de uma frase, uma por linha.

5. Percorrendo dicionários

Explicação

Dicionários armazenam pares chave–valor.

Exemplo

```
dados = {"nome": "Gil", "idade": 40}
for chave, valor in dados.items():
    print(chave, valor)
```

Exercícios

1. Exiba todas as chaves de um dicionário.

2. Exiba todas as chaves e valores.

6. for com enumerate()

Explicação

`enumerate()` retorna índice e valor ao mesmo tempo.

Exemplo

```
lista = ["a", "b", "c"]
for indice, valor in enumerate(lista):
    print(indice, valor)
```

Exercícios

1. Exiba índice e nome em uma lista de nomes.
2. Exiba índice e valor de uma lista de números.

7. for com break e continue

Explicação

`break` interrompe o loop; `continue` pula para a próxima iteração.

Exemplo

```
for n in range(10):
    if n == 5:
        break
    print(n)
```

Exercícios

1. Use `continue` para pular números ímpares.
2. Use `break` ao encontrar o número 7.

8. List Comprehension

Explicação

Forma compacta de criar listas.

Exemplo

```
quadrados = [x*x for x in range(5)]
```

Exercícios

1. Gere uma lista com números pares até 20.
2. Gere uma lista com quadrados de 1 a 10.

9. For aninhado

Explicação

Um `for` dentro de outro permite gerar combinações.

Exemplo

```
for i in range(3):  
    for j in range(3):  
        print(i, j)
```

Exercícios

1. Gere uma matriz 3×3.
2. Imprima todas as combinações possíveis entre duas listas.