

Loops Aninhados

Fundamentos de Programação em Python

Ciência de Dados e Inteligência Artificial

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Objetivos de Aprendizado

- Compreender o conceito de loops aninhados .
- Compreender iterações complexas e sua aplicação na resolução de problemas.

Introdução

- **Definição:** Um loop aninhado é um loop dentro de outro loop. Permite a execução de código repetidamente com base em múltiplas condições ou estruturas de dados.
- **Analogia:** Pense em uma grade (ex: tabuleiro de xadrez) – você precisa iterar sobre cada célula, e isso pode envolver iterar sobre linhas e colunas.
- **Importância:** Essenciais para trabalhar com dados estruturados como matrizes, listas de listas, etc. (tema de aula futura)

Conceitos Fundamentais

- **Loop Externo:** O loop principal que controla a iteração do loop interno.
- **Loop Interno:** O loop que é executado dentro do loop externo para cada iteração do loop externo.

```
for i in range(3): # Loop externo
    for j in range(2): # Loop interno
        print(f"i={i}, j={j}")
```

Escopo dos loops

- **Escopo de Variáveis:** Variáveis definidas dentro de um loop aninhado geralmente têm escopo local (visíveis apenas dentro desse loop).
- No exemplo abaixo, a variável `total` pode ser acessada pelo loop **interno**, já que este faz parte do escopo do loop **externo**.

```
for i in range(1, 6):
    total = 0
    for j in range(1, 6):
        total += i * j
        print(f"i={i}, j={j}, total={total}") # Imprime o valor de cada variável
```

Exercício 1: Produto Cartesiano

Escreva programa em Python que calcula o produto cartesiano entre os números presentes em duas strings. Cada string contém números de 0 a 9 separados por ponto e vírgula (";"). Você deve ignorar dígitos que não sejam números entre 0 e 9.

```
string1 = "a;1;3;5;"
string2 = "2;4;6;8;b;"

# 0 resultado esperado seria:
# (1, 2), (1, 4), (1, 6), (1, 8),
# (3, 2), (3, 4), (3, 6), (3, 8),
# (5, 2), (5, 4), (5, 6), (5, 8)
```

Produto cartesiano: é como fazer um cruzamento entre dois conjuntos (listas, grupos, etc.), ou seja, todas as combinações possíveis entre elementos do primeiro e do segundo conjuntos.

Exercício 2: Tabuleiro Jogo da Velha

Objetivo: Imprima um tabuleiro de jogo da velha (tic-tac-toe) usando apenas operações de strings e loops aninhados.

Restrições:

- Você deve representar o tabuleiro da velha como uma string única.
- O tabuleiro terá 3x3 posições.
- A saída deve ser o tabuleiro da velha formatado de forma legível.

Exercício 2: Tabuleiro Jogo da Velha (cont.)

O tabuleiro será representado como uma string, onde cada caractere representa uma posição no tabuleiro:

```
1 | 2 | 3
-----
4 | 5 | 6
-----
7 | 8 | 9
```

Dicas:

- Use loops aninhados para iterar sobre as linhas e colunas do tabuleiro.
- Use operações de strings (slicing, concatenação) para manipular a string do tabuleiro.

Exercício 3: Limpeza de Dados

Você tem uma string multilinhas que representa dados de usuários em um sistema. Cada linha contém informações separadas por vírgulas (nome, idade, e-mail). Sua tarefa é usar loops aninhados para extrair e imprimir a idade de cada usuário na string.

```
dados = """Nome,Idade,Email
Alice,25,alice@example.com
Bob,30,bob@example.com
#
Nome,Idade,Email
David,35,david@example.com
Eve,22,eve@example.com
#
Nome,Idade,Email
Charlie,28,charlie@example.com"""
```

Dica: A função `split(',')` é fundamental para dividir a string em campos individuais.

Dúvidas e Discussão
