Manipulação de listas em Python - material auxiliar

Uma lista armazena vários valores em uma única variável. Cada elemento possui um índice, começando em zero.

```
notas = [7.5, 8.0, 6.5, 9.0]
print(notas[0])  # 7.5
print(notas[2])  # 6.5
```

1. Inserção, remoção e acesso

```
numeros = [10, 20, 30]
numeros.append(40)     # Acrescenta ao final
numeros.insert(1, 15)  # Insere no indice 1
numeros.remove(20)     # Remove a primeira ocorrencia de 20
numeros.pop(0)        # Remove pelo indice
print(numeros)         # [15, 30, 40]
```

Outros comandos: len(lista) conta os elementos; sum(lista) soma os valores numéricos; lista.count(5) conta quantas vezes o valor 5 aparece.

2. Soma dos elementos: acumulador

Um acumulador começa com zero e recebe o resultado parcial a cada repetição.

```
numeros = [4, 7, 3]
soma = 0
for n in numeros:
    soma = soma + n
print(soma)  # 14
```

Passos: soma = 0; depois 4; depois 11; ao final 14.

3. Contagem de pares e ímpares

O operador % calcula o resto da divisão. Um número inteiro é par quando n % 2 == 0.

```
numeros = [3, 8, 11, 14, 20, 25]
pares = 0
impares = 0
for n in numeros:
    if n % 2 == 0:
        pares = pares + 1
    else:
        impares = impares + 1
print(pares, impares)  # 3 3
```

4. Percorrendo listas com while

O while exige o controle do índice e sua atualização a cada repetição.

```
numeros = [10, 20, 30]
i = 0
while i < len(numeros):
    print(numeros[i])
    i = i + 1
```

5. Concatenação e repetição não são operações vetoriais

```
a = [2, 4, 6]
```

```
b = [1, 3, 5]
print(a + b)  # [2, 4, 6, 1, 3, 5]
print(a * 2)  # [2, 4, 6, 2, 4, 6]
```

Em Python, + junta duas listas e * com um inteiro repete a lista. Eles não fazem soma ou multiplicação matemática elemento a elemento.

6. Soma matemática de dois vetores

Para vetores do mesmo tamanho: C[i] = A[i] + B[i]. Exemplo: [2, 4, 6] + [1, 3, 5] = [3, 7, 11].

```
a = [2, 4, 6]
b = [1, 3, 5]
c = []
for i in range(len(a)):
    c.append(a[i] + b[i])
print(c)  # [3, 7, 11]
```

7. Multiplicação por escalar

Multiplicar um vetor por 3 significa multiplicar cada um de seus elementos por 3.

```
a = [2, 4, 6]
c = []
for n in a:
    c.append(n * 3)
print(c)  # [6, 12, 18]
```

8. Multiplicação elemento a elemento

Para vetores do mesmo tamanho, C[i] = A[i] * B[i].

```
a = [2, 4, 6]
b = [1, 3, 5]
c = []
for i in range(len(a)):
    c.append(a[i] * b[i])
print(c)  # [2, 12, 30]
```

9. Produto escalar

O produto escalar soma os produtos correspondentes. Ele produz um número: 2*1 + 4*3 + 6*5 = 44.

```
a = [2, 4, 6]
b = [1, 3, 5]
produto = 0
for i in range(len(a)):
    produto = produto + a[i] * b[i]
print(produto)  # 44
```

10. Atenção aos erros comuns

Os índices começam em zero; um índice inexistente causa erro; o while precisa atualizar o índice; remove(valor) é diferente de pop(indice); e a soma matemática entre vetores pressupõe tamanhos compatíveis.