

Linguagem de Programação Python

Listas e Tuplas

Professor: Rítomar Torquato

1

Listas e Tuplas

Objetivos: Introdução às Listas; Índices; Cópia e fatiamento; Adição e remoção de elementos simples.


2

Introdução

Lista é um tipo de **variável** que permite o armazenar **vários valores** com **um único nome** e um **índice**.

Estrutura de Dados

O tamanho de uma lista é a quantidade de elementos que ela contém.



3

Introdução

Podemos imaginar uma lista como um edifício. O térreo é o andar zero, o primeiro andar é outro item da lista e assim por diante.



Em cada andar podemos armazenar "coisas". O prédio tem um único nome com andares (índices).

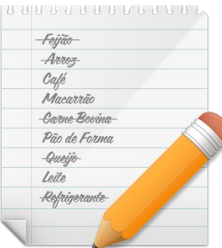
4

Introdução

Em uma lista de compras com nove itens teremos os índices variando de 0 a 8.

Se chamarmos nossa lista de compras por C, teremos C[0] feijão, C[1] arroz, C[2] café, continuando até o item C[8] refrigerante.

Listas são flexíveis e podem crescer ou diminuir com o tempo.



5

Introdução

Em Python temos:

Uma Lista Vazia.

```
>>> L = []
```

Cria uma lista chamada L que está vazia (zero itens). Os colchetes [] indicam que L é uma lista.



6

Introdução

Em Python temos:



Uma Lista com 3 elementos.

```
>>> Z = [ 15, 8, 9 ]
```

A lista Z foi criada com 3 elementos.
Dizemos que o tamanho de Z é 3.

7

Introdução

Em Python temos:



Uma Lista com 3 elementos.

```
>>> Z = [ 15, 8, 9 ]
>>> Z[0]
15
>>> Z[0] = 7
>>> Z[0]
7
>>> Z
[7, 8, 9]
```

Podemos mudar o valor de um elemento com uma atribuição.

8

Introdução

Um aluno tem 5 notas e desejamos calcular sua média aritmética.

```
notas = [6, 7, 5, 8, 9]
soma = 0
```

```
for i in range(5):
    soma += notas[i]

print(f'Média {soma/5:.2f}')
```



Todas as notas foram armazenadas na lista,
utilizando um índice para armazenar cada valor.

9

Trabalhando com índices

Um programa ler 5 números e depois solicita que o usuário escolha um número para mostrar.

```
números = [0, 0, 0, 0, 0]
```

```
for i in range(5):
    números[i] = int(input("Número %d: " % (i + 1))) A

while True:
    escolhido = int(input("Que posição você quer mostrar (0 para sair): "))
    if escolhido == 0:
        break
    print("Você escolheu o número %d" % números[escolhido-1]) B
```

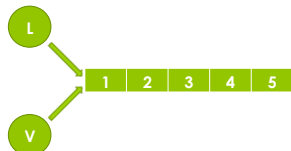
Em A adicionamos 1 para imprimir de 1 a 5, não é natural começar a contar de zero. Em B fizemos a operação inversa.

10

Cópia e fatiamento de listas

Cópia de Listas:

```
>>> L = [ 1, 2, 3, 4, 5]
>>> V = L
>>> L
[1, 2, 3, 4, 5]
>>> V[0] = 6
>>> V
[6, 2, 3, 4, 5]
>>> L
[6, 2, 3, 4, 5]
```



11

Cópia e fatiamento de listas

Cópia de Listas:

```
>>> L = [ 1, 2, 3, 4, 5]
>>> V = L[:] # Nova cópia de L
>>> V[0] = 6
>>> L
[1, 2, 3, 4, 5]
>>> V
[6, 2, 3, 4, 5]
```



12

Cópia e fatiamento de listas

Fatiamento de Listas:

```
>>> L = [1, 2, 3, 4, 5]
>>> L[0 : 5]
[1, 2, 3, 4, 5]
>>> L[ : 5]
[1, 2, 3, 4, 5]
>>> L[ : -1]
[1, 2, 3, 4]
>>> L[1 : 3]
[2, 3]
>>> L[1 : 3]
[2, 3]
>>> L[-2]
4
>>> L[-2]
4
```

Um índice negativo conta a partir de último elemento. Se L[0] é sempre o primeiro elemento L[-1] é o último, L[-2] o penúltimo...

13

Adição de elementos

Podemos adicionar elementos durante a execução.

```
L = []
while True:
    n = int(input("Digite um número (0 para sair): "))
    if n == 0:
        break
    L.append(n)
    append(5)
for i in L:
    print(i)
```

1 2 3 4 ...

Append adiciona um elemento no final da lista.

14

Adição de elementos

Podemos adicionar elementos durante a execução.

```
>>> L = [0, 1, 4, 6]
>>> L.insert(3, 5)
>>> L
[0, 1, 4, 5, 6]
>>> L.insert(2, 3)
>>> L
[0, 1, 3, 4, 5, 6]
>>> L.insert(2, 2)
>>> L
[0, 1, 2, 3, 4, 5, 6]
```

insert(3, 5)

0 1 4 ... 6

Insert adiciona um elemento na posição desejada da lista.

15

Remoção de elementos

Usamos del para remover elementos da Lista.

Remover fatias

```
>>> L = ["a", "b", "c"]
>>> del L[1]
>>> L
['a', 'c']
>>> del L[0]
>>> L
['c']
>>> L = list(range(101))
>>> del L[1:99]
>>> L
[0, 99, 100]
```



16

Aplicações

Lendo e imprimindo uma lista de compras.

```
compras = []
while True:
    produto = input("Produto ('fim' para terminar): ")
    if produto.upper() == "FIM":
        break
    compras.append(produto)
for p in compras:
    print(p)
```



17

exemplo

- Calcule a média aritmética de um aluno que possui 7 notas e mostre as notas maiores que a média. Use a estrutura de repetição mais adequada.



18

```
def ler_notas(n):
    notas = []
    for i in range(n):
        notas.append(float(input(f'Nota {i+1} de {n}:')))
    return notas

def media(notas):
    return sum(notas) / len(notas)

def maiores_que(media, notas):
    maiores = []
    for nota in notas:
        if nota > media: maiores.append(nota)
    return maiores

def main():
    notas = ler_notas(7)
    m = media(notas)
    print(maiores_que(m, notas))

if __name__ == '__main__':
    main()
```