



DTECH

Módulo 7

Listas

Algoritmos e Estruturas de Dados I

Python

(Rone Ilídio)



Conteúdo

- Listas
- Listas e o laço for
- Funções de listas
- Acessando parte da lista
- Listas de listas
- Obtendo os índices dos elementos
- Empacotar parâmetros em funções



Tipos Compostos

- Lista
 - Sequência de elementos acessados por índices
 - A ordem é importante
 - Declarada com [] ou list()
- Tupla
 - Semelhante a lista, mas não pode ser alterada após sua criação
 - Declarada com () ou uma sequência separada por ','

Tipos Compostos

- Dicionário
 - Mapeamentos no padrão chave:valor
 - A chave faz a função do índice nas listas
- Conjunto
 - A ordem não faz diferença, não possui índices
 - Não possui elementos repetidos
 - Declarado por {} ou set()



Listas



Universidade Federal
de São João del-Rei

Listas

- Listas são variáveis que armazenam mais de um valor
- Cada elemento é acessado pelo nome da lista e um número inteiro colocado entre colchetes
 - Esse número inteiro é chamado índice
 - Inicia sempre em 0
- São criadas atribuindo-se [] ao nome de uma variável ou a função *list()*
- Os valores podem ser de tipos diferentes



Listas

- Exemplo:

```
notas = [5.6, 10, 3.14, 7.8, 9.8, 9.9]
print(notas[0])
print(notas[1])
print(notas[2])
print(notas[3])
print(notas[4])
print(notas[5])
```



Listas

- Exemplo:

```
lista = []      #lista = list() ←  
for a in range(3):  
    x = input('Digite um nome: ')  
    lista.append(x)  
print(lista[0])  
print(lista[1])  
print(lista[2])
```

A lista também pode
ser criada dessa forma

Listas e o laço for



Listas e o laço for

- O comando for é utilizado para acessar cada uma dos elementos da lista
- **Exemplo**

```
for i in [5.6, 10, 3.14, 7.8, 9.8, 9.9]:  
    print(i)
```

Listas e o laço for

- **Exemplo**

```
lista = []
for i in range(5):
    lista.append(input('Informe uma fruta: '))
print('Lista de frutas informadas:')
for i in lista:
    print(i)
```

Funções de Listas



Funções de Listas

- Listas são objetos -> trataremos orientação a objetos mais a frente
- Uma lista possui várias funções predefinidas (principais):
 - `lista.append(x)` - insere x na lista
 - `lista.insert(i,x)` - insere x na posição i e desloca os demais para frente
 - `lista.remove(x)` - remove o primeiro elemento com valor igual a x
 - `lista.count(x)` - retorna o número de vezes que x aparece na lista
 - `lista.sort()` - ordena a lista
 - `lista.sort(reverse=True)` - ordena em ordem decrescente
 - `lista.pop()` - retorna o último elemento e o remove da lista



Funções de listas

- Exemplo

```
lista = []  
for i in range(3):  
    lista.append(int(input('Informe um número: ')))  
lista.insert(1,14)  
print(lista)
```

← **Insere 14 na posição 1**

Funções de Listas

- Exemplo

```
lista = [1, 2, 3, 4, 5]
lista.remove(4)
print(lista)
```

Atenção: se o 4 não existir ocorre uma exceção

Funções de Listas

```
lista = []
while(True):
    n = input('Digite um nome (sair para exibir): ')
    if n == 'sair':
        break
    lista.append(n)
lista.sort()
print(lista)
print('Tamanho da lista: ', len(lista))
```

Acessando somente parte de uma lista



Acessando somente parte de uma lista

- A sintaxe lista [x : y] acessa a lista a a partir da posição x até o elemento anterior à posição y
- Importante: a primeira posição é 0
- Exemplo 1

```
lista = ['a', 'b', 'c', 'd', 'e', 'f']
```

```
print(lista[2:5])
```

← Exibe os elementos da posição 2 até a 4: ['c', 'd', 'e']

Acessando somente parte de uma lista

- Exemplo:

```
lista = ['a', 'b', 'c', 'd', 'e', 'f']  
print(lista[3:])
```

Imprime da posição 3 até o final da lista: ['d', 'e', 'f']

Acessando somente parte de uma lista

- Exemplo:

```
lista = []
```

```
while(True):
```

```
    cor = input('Digite uma cor (fim para sair): ')
```

```
    if cor == 'fim':
```

```
        break
```

```
    lista.append(cor)
```

```
i = int(input('Informe o primeiro elemento: '))
```

```
print(lista[i:len(lista)])
```

← Imprime da posição i até o final



Comando del para apagar elementos da lista



Comando del para apagar elementos da lista

- Apaga um elemento ou um conjunto de elementos em sequência em uma lista
- Exemplo:

```
lista = ['aaa', 'bbb', 'ccc', 'ddd', 'eee']
```

```
print('Lista inteira:', lista)
```

```
del lista[1]
```



Elimina 'bbb'

```
print('Lista após remoção:', lista)
```



Comando del para apagar elementos da lista

- Exemplo:

```
lista = ['aaa', 'bbb', 'ccc', 'ddd', 'eee']  
print('Lista inteira:', lista)  
del lista[1:3] ←  
print('Lista após remoção:', lista)
```

Elimina 'bbb' e 'ccc',
posições 1 e 2

Criando listas com *

- O * é utilizado repetir o elemento inicial na criação da lista
- Exemplo

```
a = [0]*5  
print(a)
```

Resultado: uma lista formada por 5 zeros [0,0,0,0,0]

Lista de Listas



Universidade Federal
de São João del-Rei

Lista de Listas

- Cada elemento de uma lista pode ser outra lista
- Python não possui matrizes como em C e C++
- Uma matriz é criada como uma lista de listas



Lista de Listas

```
semestre = []
while True:
    nome = input('Nome do aluno (sair para finalizar):')
    if nome == 'sair':
        break
    prova = float(input('Nota da prova:'))
    trab = float(input('Nota do trabalho:'))
    semestre.append([nome, [prova, trab]])
print(semestre)
```

```
semestre = []
while True:
    nome = input('Nome do aluno (sair para finalizar):')
    if nome == 'sair':
        break
    prova = float(input('Nota da prova:'))
    trab = float(input('Nota do trabalho:'))
    semestre.append([nome,[prova, trab]])
print('\nResultado:')
for aluno in semestre:
    print('Nome:', aluno[0], 'Prova:', aluno[1][0], 'Trabalho:', aluno[1][1])
```

Lista de Lista

Exemplo: equivalente ao anterior com uma exibição melhor

Lista de Lista

```
semestre = []
while True:
    nome = input('Nome do aluno (sair para
finalizar):')
    if nome == 'sair':
        break
    prova = float(input('Nota da prova:'))
    trab = float(input('Nota do trabalho:'))
    semestre.append([nome, [prova, trab]])
print('\nResultado:')
for a,b in semestre:
    print('Nome:', a, 'Prova:', b[0], 'Trabalho:', b[1])
```

Exemplo: equivalente ao anterior, mas acesa a nota de forma diferente

Matrizes: lista de lista

- Pode ser considerada uma lista multidimensional
- Exemplo:

```
matriz = [[1,2,3],[4,5,6]]  
print(matriz)
```

- Esse exemplo imprime [[1,2,3],[4,5,6]], o que equivale à uma matriz com o seguinte formato

	0	1	2
0	1	2	3
1	4	5	6

Matrizes: lista de lista

- Para acessar cada elemento da lista, utilize dois colchetes
- Exemplo

```
matriz = [[1, 2, 3], [4, 5, 6]]
```

```
matriz[0][1] = 8
```

```
matriz[1][2] = 9
```

```
print(matriz)
```

Muda o elemento da linha 0,
coluna 1 para 8

Muda o elemento da linha 1,
coluna 2 para 9

- Resultado

```
[[1, 8, 3], [4, 5, 9]]
```

	0	1	2
0	1	8	3
1	4	5	9



Matrizes: lista de lista

- Utilize dois comandos *for* para acessar cada elemento da lista
- Exemplo (recebe e exibe uma matriz)

```
matriz = [[0,0,0],[0,0,0],[0,0,0]]
for l in range(0,3):
    for c in range(0,3):
        matriz[l][c] = input('Digite:')
print(matriz)
```

Matrizes: lista de lista

- Duas formas para declaração de uma matriz 3 x 3 (como no exemplo anterior)

```
matriz = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
```

ou

```
matriz = [[0]*3, [0]*3, [0]*3]
```

Matrizes: lista de lista

- Exemplo (recebe e exibe uma matriz)

```
matriz = [[0]*3, [0]*3, [0]*3]
for l in range(0,3):
    for c in range(0,3):
        matriz[l][c] = input(f'Digite a posição [{l}][{c}]:')
print(matriz)
```

Matrizes: lista de lista

- Exemplo (exibe separadamente cada elemento da matriz)

```
matriz = [[0]*3, [0]*3, [0]*3]
for l in range(0,3):
    for c in range(0,3):
        matriz[l][c] = int(input(f'Posição [{l}][{c}]:'))
for l in range(0,3):
    for c in range(0,3):
        print(matriz[l][c], ' ', end='')
```

Matrizes: lista de lista

- Exemplo (exibe a soma dos valores de cada linha)

```
matriz = [[0]*3, [0]*3, [0]*3]
```

```
for l in range(0,3):
```

```
    for c in range(0,3):
```

```
        matriz[l][c] = int(input(f'Posição [{l}][{c}]:'))
```

```
for linha in matriz:
```

```
    soma = 0
```

```
    for coluna in linha:
```

```
        soma += coluna
```

```
    print(soma)
```

Veja como os elementos da matriz foram acessados



Obtendo os Índices dos Elementos



Obtendo os Índices dos Elementos

- Com a função `enumerate()` pode-se obter os índices dos elementos de uma lista
- Também pode ser utilizada para tuplas, dicionários e conjuntos

Obtendo os Índices dos Elementos

- Exemplo

```
x = ['primeiro', 'segundo', 'terceiro']  
for i, v in enumerate(x):  
    print(i, v)
```

Resultado:
0 primeiro
1 segundo
3 terceiro

Empacotar Parâmetros



Empacotar Parâmetros

- Definição de funções que não possuem um número definido de parâmetros
- A chamada pode receber um número variado de parâmetros
- Na declaração do parâmetro no cabeçalho da função, coloca-se o *

Empacotar Parâmetros

- Exemplo

```
def maior(* p):
```

← * define que o número de parâmetros pode variar

```
    m = p[0]
```

← p é tratado como uma lista (na verdade uma tupla)

```
    for i in p:
```

```
        if(i>m):
```

```
            m = i
```

```
    return m
```

```
print(maior(124, 25, 4, 45, 5, 36))
```

Teste com números diferentes de parâmetros



Exercícios



Universidade Federal
de São João del-Rei

Exercícios

1. Crie uma lista com 5 nomes fornecidos pelo usuário. Exiba-os na ordem inversa da entrada.
2. Receba nomes de cores e insira em uma lista, finalize com sair. Exiba a lista em duas linhas, cada uma com metade dos elementos e em ordem alfabética.
3. Crie uma lista equivalente a uma matriz 4x4. Preencha com números informados pelo usuário. Exiba separadamente os elementos da diagonal principal, os que estão acima e os que estão abaixo.
4. Faça o usuário preencher uma lista de funcionários (número indefinido). Cada funcionário possui um nome e uma lista com os nomes de seus filhos. Ao final, pergunte um funcionário e exiba seus filhos. Repita essa operação.

Exercício 1

```
lista = []  
for i in range(5):  
    lista.append(input('Digite um nome:'))  
for i in range(4,-1,-1):  
    print(lista[i],end=' ')
```



Exercício 2

```
lista = list()
while True:
    cor = input('Digite uma cor:')
    if cor.lower() == 'sair':
        break
    lista.append(cor)
lista.sort()
t = len(lista)//2
print(lista[:t])
print(lista[t:])
```

```

lista = [[0]*4,[0]*4,[0]*4,[0]*4,]
for a in range(4):
    for b in range(4):
        lista[a][b] = int(input('Digite um número:'))
print('Diagonal principal:')
for a in range(4):
    print(lista[a][a], end=' ')
print('\nSuperior:')
for a in range(4):
    for b in range(4):
        if a < b:
            print(lista[a][b], end=' ')
        else:
            print(' ', end=' ')
print()

```

Exercício 3

1/2

```
print('\nInferior:')
for a in range(4):
    for b in range(4):
        if a > b:
            print(lista[a][b], end=' ')
        else:
            print(' ', end=' ')
    print()
```

Exercício 3

2/2

```
lista = []
while True:
    func = input('Nome do funcionário (fim para sair):')
    if func == 'fim':
        break
    filhos = []
    while True:
        filho = input('Nome do filho (fim para sair):')
        if filho == 'fim':
            break
        filhos.append(filho)
    lista.append([func, filhos])
```

Exercício 4

1/2

```
while True:
    procurado = input('Nome procurado (fim para sair):')
    if procurado == 'fim':
        break
    encontrou = False
    for i, u in lista:
        if i == procurado:
            print(u)
            encontrou = True
            break
    if not encontrou:
        print('Não encontrado')
```

Exercício 4

2/2