

Programação em Python

Listas de dados (vetores e matrizes)

Prof. Daniel Di Domenico

https://github.com/danidomenico/gex003_algprog

Slides cortesia da profa. Andrea Charão (UFSM) e do prof. João V. F. Lima (UFSM)

Listas (recapitulação)

- São **coleções** de dados;
- Variável de tipo lista pode conter valores de todos os outros tipos (**int**, **float**, **str** ou **outras listas**);
- Exemplos:

```
x = [20, 40, 60, 80]
```

```
nomes = ["Fulano", "Beltrano"]
```

```
notas = ["Fulano", 8.5, 9.1]
```

Listas aninhadas

- São **coleções** de dados dentro de outras **coleções** de dados;
- Exemplo:

Nome	Nota
Fulano	6.5
Beltrano	8.6
Sicrano	7

Listas aninhadas

- Representação da tabela de alunos com uma lista em Python:

```
alunos= [ ["Fulano", 6.5],  
          ["Beltrano", 8.6],  
          ["Sicrano", 7] ]
```

```
for x in alunos: #x também é uma lista  
    print("Nome: {} - Nota: {:.1f}".format(x[0], x[1]))
```

Nome: Fulano - Nota: 6.5
Nome: Beltrano - Nota: 8.6
Nome: Sicrano - Nota: 7.0

Listas aninhadas

- Caso todas as sub-listas possuam o mesmo número de elementos, então tem-se uma **matriz**:

```
#alunos é uma matriz 3x2
alunos= [ ["Fulano", 6.5],
          ["Beltrano", 8.6],
          ["Sicrano", 7] ]
```

```
for aluno in alunos:
    for campo in aluno:
        print(campo, end=";")
    print()
```

```
Fulano;6.5;
Beltrano;8.6;
Sicrano;7;
```

Listas aninhadas

- **ATENÇÃO:** listas aninhadas podem não ser uma matriz:

```
#dados não é uma matriz
dados= [ ["Dado 1", 6.5, 4.2, 6.6], #4 colunas
         ["Dado 2", 8.6, 3.1, 2.1, 7.3], #5 colunas
         ["Dado 3", 7, 4.1] ] #3 colunas
```

Operações com Listas aninhadas

- Acesso aos **elementos** da matriz:
 - **matriz[índice][índice]**
 - Lembrando: o índice começa em zero

```
mat = [ ["Fulano", 6.5, 8.2, 9.3],  
        ["Beltrano", 8.6, 8.1, 3.2],  
        ["Sicrano", 7, 2.4, 9.2] ]
```

```
print(mat[0])
```

```
['Fulano', 6.5, 8.2, 9.3]
```

```
print(mat[0][1])
```

```
6.5
```

```
print(mat[1][2])
```

```
8.1
```

Operações com Listas aninhadas

- Acesso aos **elementos** da matriz com índices negativos:

```
mat = [ ["Fulano", 6.5, 8.2, 9.3],  
        ["Beltrano", 8.6, 8.1, 3.2],  
        ["Sicrano", 7, 2.4, 9.2] ]
```

```
print(mat[-1])
```

```
['Sicrano', 7, 2.4, 9.2]
```

```
print(mat[-1][-2])
```

```
2.4
```

```
print(mat[-2][-3])
```

```
8.6
```


Operações com Listas aninhadas

- Adicionar elementos em uma matriz:

```
mat = []
mat.append([1, 2, 3]) #append da lista inteira

cols = []
cols.append(4) #append dos elementos da lista cols
cols.append(5)
cols.append(6)
mat.append(cols) #append da lista 'cols' na lista 'mat'
print(mat)
[[1, 2, 3], [4, 5, 6]]

mat.append([]) #append da lista vazia
mat[len(mat)-1].append(7) #append dos elementos
mat[len(mat)-1].append(8)
mat[len(mat)-1].append(9)
print(mat)
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]

mat.append(list(range(10, 13))) #append da lista gerada
print(mat) #com range
[[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
```

Operações com Listas aninhadas

- Adicionar elementos através de concatenação (+):

```
mat = [ ["Fulano", 6.5, 8.2, 9.3],  
        ["Beltrano", 8.6, 8.1, 3.2],  
        ["Sicrano", 7, 2.4, 9.2] ]  
  
mat = mat + [ ["Catiano", 5.1, 7.3, 6.1]]
```

```
print(mat)  
[ ['Fulano', 6.5, 8.2, 9.3],  
  ['Beltrano', 8.6, 8.1, 3.2],  
  ['Sicrano', 7, 2.4, 9.2],  
  ['Catiano', 5.1, 7.3, 6.1]]
```

Operações com Listas aninhadas

- Alterar elementos de uma matriz:

```
mat = [ ["Fulano", 6.5, 8.2, 9.3],  
        ["Beltrano", 8.6, 8.1, 3.2],  
        ["Sicrano", 7, 2.4, 9.2] ]
```

```
linha = mat[0]  
linha[1] = 7.2  
print(linha)  
['Fulano', 7.2, 8.2, 9.3]
```

```
mat[2] = ["Catiano", 5.1, 7.3, 6.1]  
mat[1][3] = 8.9  
print(mat)  
[ ['Fulano', 7.2, 8.2, 9.3],  
  ['Beltrano', 8.6, 8.1, 8.9],  
  ['Catiano', 5.1, 7.3, 6.1] ]
```

Operações com Listas aninhadas

- Remover elementos em uma matriz (linha):

```
mat = [ ["Fulano", 6.5, 8.2, 9.3],  
        ["Beltrano", 8.6, 8.1, 3.2],  
        ["Sicrano", 7, 2.4, 9.2] ]
```

```
mat.pop(1)  
print(mat)
```

```
[ ['Fulano', 7.2, 8.2, 9.3],  
  ['Sicrano', 7, 2.4, 9.2] ]
```

Operações com Listas aninhadas

- Remover elementos em uma matriz (coluna):
 - Deve-se remover o elemento de **todas as linhas**, pois senão a matriz deixa de ser uma matriz:

```
mat = [ ["Fulano", 6.5, 8.2, 9.3],  
        ["Beltrano", 8.6, 8.1, 3.2],  
        ["Sicrano", 7, 2.4, 9.2] ]
```

```
for linha in mat:  
    linha.pop(1)
```

```
print(mat)  
[ ['Fulano', 8.2, 9.3],  
  ['Beltrano', 8.1, 3.2],  
  ['Sicrano', 2.4, 9.2] ]
```

Exercícios

- 1) Faça um programa que leia números inteiros para uma matriz 4x4. Em seguida, calcule a média dos elementos dessa matriz.
- 2) Faça um programa igual ao anterior, porém escreva os elementos da diagonal principal e em seguida, imprima a soma deles.