Programação em Python

Listas de dados (ou vetores)

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

- 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]
```

Operações com Listas

- Função para obter o **tamanho** da lista **(len)**:

```
x = [20, 40, 60, 80] #lista de 4 elementos
print(len(x))
4
```

- Acesso aos **elementos** da lista:
  - **lista[índice]**
  - O índice começa em zero:

```
x = [20, 40, 60, 80]
print(x[0])
20
print(x[1])
40
```

Operações com Listas

- O índice pode ser negativo:

```
x = [20, 40, 60, 80]
print(x[-1]) #índice negativo acessa do último para o primeiro
80
print(x[-2])
60
```

- Subconjunto de listas:
  - **x[i:j]**, retornando o intervalo de **i** até **j-1**:

```
x = [20, 40, 60, 80]
print(x[1:3])
[40, 60]
print(x[:2]) #retorna o intervalo de 0 até o índice-1
[20, 40]
print(x[2:]) #retorna o intervalo de 2 até o fim da lista
[60, 80]
```

Operações com Listas

- Atualizar elementos da lista:

```
x = [20, 40, 60, 80]
x[1] = 45
print(x)
[20, 45, 60, 80]
```

- Concatenar listas:
  - Utilizar operador **+**:

```
x = [20, 40, 60, 80]
y = [100, 120]
z = x + y
print(z)
[20, 40, 60, 80, 100, 120]
```

Operações com Listas

- Adicionar elementos:

```
x = []   #declaração de uma lista vazia
x.append(20) #insere o novo elemento no fim da lista
x.append(30)
print(x)
[20, 30]

x.insert(1, 25) #insere o novo elemento no índice 1
print(x)
[20, 25, 30]
```

Operações com Listas

- Remover elementos:

```
x = [20, 40, 60, 80, 100, 120]
a = x.pop() #remove o último elemento da lista, retornando-o
print(a)
120

a = x.pop(1) #remove o elemento do índice 1, retornando-o
print(a)
40

x.remove(80) #remove o elemento 80 (pelo valor)
print(x)
[20, 60, 100]
```

Operações com Listas

- Ordenar elementos:

```
x = [18, 3, 27, 9, 1]
x.sort()
print(x)
[1, 3, 9, 18, 27]
```

- Reverter ordem:

```
x = [18, 3, 27, 9, 1]
x.reverse()
print(x)
[1, 9, 27, 3, 18]
```

Operações com Listas

- Outras operações para listas:

Método	Funcionalidade
lista.clear()	Remove todos os elementos da lista
lista.index(val)	Retorna o índice do primeiro item encontrado igual a “val”
lista.count(val)	Retorna quantos valores igual a “val” existem na lista
lista.copy()	Retorna um cópia da lista

Repetições com listas

- Comando **for:**
  - Executa um bloco de comandos **para cada** elemento de uma lista:

```
for variavel in lista:
    # bloco de comandos
    # a repetir
```

Repetições com listas

- Mostrar na tela os elementos da lista;
- Variável **i** recebe um valor diferente a cada iteração do laço;
- Exemplo:

```
x = [20, 30, 40]
for i in x:
    print(i)
```

- Saída:

```
20
30
40
```

Exemplo: repetições com lista

- Mostrar a tabuada de multiplicação do número 6 por números de 0 a 10:
  - Criar lista com números de 0 a 10;
  - Para cada elemento da lista, multiplicá-lo por 6;

- Exemplo:

```
numeros = [0,1,2,3,4,5,6,7,8,9,10]
for i in numeros:
    print(i * 6)
```

- Saída:

```
0
6
12
18
24
30
36
42
48
54
60
```

Exemplo: repetições com lista

- Mostrar a tabuada de multiplicação do número 6 por números de 0 a 10:
  - Utilizar **range** para criar a lista;
  - Para cada elemento da lista, multiplicá-lo por 6;

- Exemplo:

```
for i in range(11): #repete de 0 a 10
    print(i * 6)
```

- Saída:

```
0
6
12
18
24
30
36
42
48
54
60
```

Gerar uma lista

- Método **range** pode ser utilizado para **gerar** uma lista com uma sequência de números:
  - **range** gera apenas um iterador para o **for**;
  - Para gerar a lista, é preciso fazer o *cast* (**list**):

```
x = range(11) #x não é uma lista, mas um iterador
print(x)
range(0, 11)
x = list(range(11))
print(x)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
x = list(range(2,10,2))
print(x)
[2, 4, 6, 8]
x = list(range(5, 0, -1))
print(x)
[5, 4, 3, 2, 1]
```

Exercícios

1) Dado a lista de números abaixo, faça um programa que calcule a média desses números.

```
x = [5, 6, 7, 8.8, 5, 6, 7.8, 9.8, 7]
```

2) Faça um programa que lê cinco (5) nomes digitados pelo usuário e no final imprime esses nomes ordenados. Guarde os nomes em uma lista para serem ordenados no final.