

Disciplina: IPOO - Introdução à Programação Orientada a Objetos

Prof. Luiz Cláudio Dalmolin - luiz.dalmolin@udesc.br - 2021/2

Listas - (com a linguagem Python)

```
In [78]: # Acesso a uma lista:
z = [15,8,9]
#z[0]
z[1]
# z[2]
```

Out[78]: 8

```
In [2]: # Modificação de uma lista:
z = [15,8,9]
z[0]
# z[0] = 7
# z[0]
# z
```

Out[2]: 15

```
In [81]: # Exemplo, cálculo da média:
notas = [6,7,10,8,9]
soma = 0
i = 0
while i < 5:
    soma += notas[i]
    i += 1
print("Média: %.2f" % (soma/i))
```

Média: 8.00

```
In [84]: # Exemplo, cálculo da média com notas digitadas:
notas = [0,0,0,0,0]
soma = 0
i = 0
while i < 5:
    notas[i] = float(input("Nota %d: " % (i+1)))
    soma += notas[i]
    i += 1
print("Média: %.2f" % (soma/i))
```

Nota 1: 3
Nota 2: 3
Nota 3: 4
Nota 4: 3
Nota 5: 2
Média: 3.00

In [85]: *# Exemplo do cálculo da média com notas digitadas e impressão:*

```
notas = [0, 0, 0, 0, 0]
soma = 0
i = 0
while i < 5:
    notas[i] = float(input("Nota %d: " % (i+1)))
    soma += notas[i]
    i += 1
i = 0
while i < 5:
    print("Nota %d: %.2f" % (i+1), notas[i])
    i += 1
print("Média: %.2f" % (soma/i))
```

```
Nota 1: 20
Nota 2: 6
Nota 3: 8
Nota 4: 15
Nota 5: 4
Nota 1: 20.00
Nota 2: 6.00
Nota 3: 8.00
Nota 4: 15.00
Nota 5: 4.00
Média: 10.60
```

In [87]: *# Exemplo, apresentação de números:*

```
num = [0,0,0,0,0]
x = 0
while x < 5:
    num[x] = int (input("Número %d: " % x))
    x += 1
while True:
    escolhido= int(input("Que posição quer imprimir '-1' para sair): "))
    if escolhido is -1:
        break
    print("Você escolheu o número: %d" % (num[escolhido]))
```

```
Número 0: 4
Número 1: 2
Número 2: 5
Número 3: 7
Número 4: 8
Que posição quer imprimir '-1' para sair): 1
Você escolheu o número: 2
Que posição quer imprimir '-1' para sair): 4
Você escolheu o número: 8
Que posição quer imprimir '-1' para sair): 0
Você escolheu o número: 4
Que posição quer imprimir '-1' para sair): -1
```

In [93]: *# Exemplo, tentativa de copiar listas:*

```
L = [1,2,3,4,5]
V = L
print(L)
print (V)
V[0] = 6
print (V)
print (L)
```

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

In [94]: *# Exemplo para criar uma cópia independente:*

```
L = [1,2,3,4,5]
V = L[:]
V[0] = 6
print (V)
print (L)
```

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

In [95]: *# Fatias (slices) em listas:*

```
# [início : fim]      # vai do índice 'início' até o 'fim'-1
# [início : ]        # vai do índice 'início' até o final da lista
# [ : fim]           # vai do início da lista até índice 'fim'-1
# [ : ]              # copia a lista toda
```

Exemplos de slices:

```
L = [10,20,30,40,50]
print (L[0:4])      # elementos com índices 0 a <5
print (L[:5])        # elementos com índice < 5
print (L[:-1])       # menos o último elemento
print (L[1:4])       # elementos com índice 1 até <4
print (L[3:])         # elementos a partir do índice 3
print (L[:3])         # elementos com índices < 3
print (L[-2])         # penúltimo elemento
```

```
[10, 20, 30, 40]
[10, 20, 30, 40, 50]
[10, 20, 30, 40]
[20, 30, 40]
[40, 50]
[10, 20, 30]
40
```

In [96]: *# número de elementos de uma lista*

```
L = [1,2,3,4,5,55,8,3,44]
print(len (L))

V = []
print(len (V))
```

```
9
0
```

```
In [99]: # Exemplo de repetição com tamanho de lista usando len
L = [1,2,7,5]
x = 0
while x < len(L):
    print(L[x])
    x += 1
```

```
1
2
7
5
```

```
In [101]: # Exemplo de adição de elementos à lista:
L = []
L.append ("a")
print(L)
L.append ("x")
L.append ("b")
L.append ("c")
print(L)
len(L)
```

```
['a']
['a', 'x', 'b', 'c']
```

Out[101]: 4

```
In [104]: # Exemplo de adição de elementos à lista:
L = []
while True:
    n = int(input("Digite um número (0 para sair): "))
    if n is 0:
        break
    L.append(n)
x = 0
while x < len(L):
    print(L[x])
    x += 1
L
```

```
Digite um número (0 para sair): 54
Digite um número (0 para sair): 3
Digite um número (0 para sair): 2
Digite um número (0 para sair): 0
54
3
2
```

Out[104]: [54, 3, 2]

```
In [68]: # Remoção de elementos de uma lista com 'del'
L = ["a", "b", "c"]
print (L)

del L[1]
print (L)

del L[0]
print (L)

['a', 'b', 'c']
['a', 'c']
['c']
```

```
In [105]: # Pode-se apagar fatias inteiras de uma só vez.
L = list(range(40)) # range: gera uma lista de números.
print(L)
del L[1:30]
L

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
```

```
Out[105]: [0, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
```

```
In [108]: # pesquisa de elementos na lista
L = [15,7,27,39,6,88,100,1,6,33]
p = int(input("Digite o valor a procurar: "))

achou = False
x = 0
while x < len(L):
    if L[x] is p:
        achou = True
        break
    x += 1
if achou:
    print("%d achado na posição %d" % (p,x))
else:
    print("%d não encontrado" % p)
```

```
Digite o valor a procurar: 6
6 achado na posição 4
```

```
In [110]: # Exemplo da impressão de todos os elementos da lista:
L = [8,9,15]
for e in L:
    print(e)
# impressão direta da lista:
#L

8
9
15
```

```
In [113]: # Exemplo de pesquisa usando for
L = [7,9,10,12]
p = int(input("Digite um valor a procurar: "))
for e in L:
    if e is p:
        print("%d foi encontrado" % p)
        break
    else:
        print("%d não encontrado" % p)
```

Digite um valor a procurar: 9
9 foi encontrado

```
In [115]: # usando a função 'range' para gerar listas simples.
# Exemplo para imprimir de 0 a 9 na tela:
#for v in range(5):
#    print(v)

for v in range(5,8):
    print(v)
```

5
6
7

```
In [116]: # Usando 'range': se adicionar um terceiro elemento, terá como saltar entre os valores
          # gerados. Exemplo:
for t in range(3,17,3):
    print(t)
```

3
6
9
12
15

```
In [39]: # Pode se transformar o resultado de range em uma lista.
# Exemplo:
L=list(range(100,1100,50))
print(L)
```

[100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000, 1050]

```
In [20]: # Programa para ler N notas e imprimir ao final:
N = int (input("Digite a quantidade de Notas: "))
alu = []
qtde = 0
for i in range(0,N,1):
    alu.append(float(input("Digite a %da. nota: " % (qtde+1))))
    qtde += 1
print(alu)

# Altere o programa para imprimir ao final a média aritmética das notas digitadas.
soma=0
qtde=0
for x in alu:
    soma=soma+ alu[qtde]
    qtde +=1
media = soma/qtde
print("Media = ", media)
```

```
Digite a quantidade de Notas: 4
Digite a 1a. nota: 8
Digite a 2a. nota: 6
Digite a 3a. nota: 8
Digite a 4a. nota: 6
[8.0, 6.0, 8.0, 6.0]
Media = 7.0
```

```
In [22]: # Função 'enumerate' para ampliar as funcionalidades de 'for' .
# Exemplo de como imprimir uma lista, onde se tem o índice entre colchetes e o valor à
sua direita:
L = [5,9,13]
x = 0
for e in L:
    print ("[%d] %d" % (x,e))
    x += 1

# função enumerate() retorna uma tupla de dois elementos a cada iteração:
# um número sequencial e um item da sequência correspondente

L = [5,9,13]
for x, e in enumerate(L):      # o primeiro valor é o índice e o
    print ("[%d] %d" % (x, e)) # segundo é o elemento da lista
```

```
[0] 5
[1] 9
[2] 13
[0] 5
[1] 9
[2] 13
```

```
In [24]: # Exemplo para verificação do maior valor:
L = [1, 15, 9, 13]
maior = L[0]
for e in L:
    if e > maior:
        maior = e
print(maior)
```

```
In [26]: # Imagine uma situação na qual precisa se selecionar os elementos de uma lista de forma a copiá-los para outras duas listas.
# Para P, se forem pares; ou para I, se ímpares.
# Exemplo de cópia de elementos para outras listas:
V = [9,8,7,12,0,13,21,66,43,12,1]
P = []
I = []
for e in V:
    if e % 2 == 0:
        P.append(e)
    else:
        I.append(e)
print("Pares: ", P)
print("Ímpares: ", I)
```

```
Pares: [8, 12, 0, 66, 12]
Ímpares: [9, 7, 13, 21, 43, 1]
```

```
In [30]: # Ordenação de lista (Bubble sort)
# versão com limite final variável -> https://www.youtube.com/watch?v=JP5KkzdUEYI
# demonstração com limite fixo -> https://www.youtube.com/watch?v=Bi8xZLuydWY

L = [7,4,3,12,1,8]
fim = len(L)
while fim > 1:
    trocou = False
    x = 0
    while x < (fim-1):
        if L[x] > L[x+1]:
            temp = L[x]
            L[x] = L[x+1]
            L[x+1] = temp
            trocou = True
        x += 1
    if not trocou:
        break
    fim -= 1
#for e in L:
#    print(e)
L
```

```
Out[30]: [1, 3, 4, 7, 8, 12]
```

Exercícios Propostos

1) Inicializar um vetor de 10 inteiros com o valor zero e imprimir.

```
In [6]: L=[]
x=0
while x<10:
    L.append(0)
    x +=1
print(L)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

2) Inicializar um vetor com os valores 1,2,3,4,5,6,7,8,9,10 e depois imprimir.


```
In [32]: L=[]
x=0
while x<10:
    L.append(x+1)
    x +=1
print(L)
```

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

3) Ler 5 inteiros do teclado e armazenar num vetor. Depois percorrer este vetor mostrando os números ímpares.

```
In [33]: L = []
n = 0
while n<5:
    L.append(int(input("Digite um número: ")))
    n +=1
n = 0
while n < len(L):
    if L[n] % 2 != 0:
        print ("impar -> ", L[n])
    n +=1
```

Digite um número: 54

Digite um número: 45

Digite um número: 33

Digite um número: 22

Digite um número: 7

impar -> 45

impar -> 33

impar -> 7

4) Inicializar um vetor com valores 2, 4, 6, 8, 10. Calcular e imprimir a soma dos valores.

```
In [36]: L = []
n = 0
v = 2
while n<5:
    L.append(v)
    n +=1
    v +=2
n = 0
soma=0
while n < len(L):
    soma += L[n]
    n +=1
print("Soma= ", soma)
```

Soma= 30

Out[36]: [2, 4, 6, 8, 10]

5) Ler e armazenar num vetor 5 números float fornecidos pelo usuário e calcular a média dos valores.

```
In [38]: L = []
n = 0
while n<5:
    L.append(float(input("Digite um número: ")))
    n +=1
n = 0
soma=0
while n < len(L):
    soma += L[n]
    n +=1
print("Média= ", soma/5)
```

```
Digite um número: 10
Digite um número: 10
Digite um número: 10
Digite um número: 10
Digite um número: 10
Média= 10.0
```

6) Dado os vetores A=[10, 11, 12, 13, 14, 15, 16,17] e B=[1, 2, 3,4, 5, 6, 7, 8], fazer um programa para somar os dois vetores e armazenar o resultado em um vetor C ($C[i]=A[i]+B[i]$).

7) Escrever um programa que leia 5 números float e imprima os em ordem inversa.

8) Faça um programa que leia duas listas e que gere uma terceira com os elementos das duas primeiras.

9) Faça um programa que percorra duas listas e gere uma terceira sem elementos repetidos.

```
In [12]: L1=[10,15,3,8,5,3,40,20]
L2=[4,3,9,8,11,10]
L3=[]
x=0
y=0
while x<len(L1):
    if L1[x] not in L3:
        L3.append(L1[x])
    x +=1
x=0
while x<len(L2):
    if L2[x] not in L3:
        L3.append(L2[x])
    x +=1
print (L3)
```

```
[10, 15, 3, 8, 5, 40, 20, 4, 9, 11]
```

10) Faça um programa de forma a realizar a mesma tarefa do exemplo localizado no slide 21, mas sem utilizar a variável achou. Dica: observe a condição de saída do while

```
In [16]: # programa do slide 21 (original)
L = [15,7,27,39]
p = int(input("Digite o valor a procurar: "))
achou = False
x = 0
while x < len(L):
    if L[x] is p:
        achou = True
        break
    x += 1
if achou:
    print("%d achado na posição %d" % (p,x))
else:
    print("%d não encontrado" % p)
```

Digite o valor a procurar: 4
4 não encontrado

```
In [17]: # programa do slide 21 (MODIFICADO)
L = [15,7,27,39]
p = int(input("Digite o valor a procurar: "))
x = 0
while x < len(L):
    if L[x] is p:
        print("%d achado na posição %d" % (p,x))
        break
    x += 1
if x >= len(L):
    print("%d não encontrado" % p)
```

Digite o valor a procurar: 4
4 não encontrado

11) Modifique o exemplo (slide 21) para pesquisar dois valores. Em vez de apenas p, leia outro valor v que também será procurado. Na impressão, indique qual dos dois valores foi achado primeiro.

12) Modifique o exercício anterior de forma a pesquisar p e v em toda a lista e informando ao usuário a posição onde p e a posição onde v foram encontrados.

13) Faça um programa para imprimir o menor elemento de uma lista.

14) A lista de temperaturas de Mons, na Bélgica, foi armazenada na lista T = [10, 8, 0, 1, 2, 3, 2, 4]. Faça um programa que imprima a menor e a maior temperatura, assim como a temperatura média.

15) Modifique o programa do slide 32 para ordenar a lista em ordem decrescente. L =[2,1,5,4,3] deve ser ordenada como L=[5,4,3,2,1].

16) Faça um programa que leia N valores inteiros, e depois apresente os em ordem crescente.

Outros exemplos com LISTAS:

```
In [3]: # convertendo uma string em lista
test_str="Python"
list(test_str)
```

```
Out[3]: ['P', 'y', 't', 'h', 'o', 'n']
```

```
In [23]: s1 = input("Digite seu nome: ")
for x in s1:
    print (x)
```

```
Digite seu nome: UDESC
U
D
E
S
C
```

```
In [38]: s1 = input("Digite seu nome: ")
# s1[::-1]
i=0
for x in s1:
    print(s1[:i:])
    i+=1
i=0
for x in s1:
    print(s1[i::])
    i+=1
```

```
Digite seu nome: teste

t
te
tes
test
teste
este
ste
te
e
```

```
In [10]: # tipos de listas em Python
lista_simples_inteiro = [1, 2, 3, 8, 14, 4, 5]
lista_simples_string = ["Olá", "Mundo"]
lista_simples_mesclada = [1, 2, 3, "Olá, mundo", True, 1.5]

lista_simples_inteiro
#lista_simples_string
#lista_simples_mesclada
```

```
Out[10]: [1, 2, 3, 8, 14, 4, 5]
```

```
In [21]: # manipulação de listas
lista = [1, 2, 3, 8, 14, 4, 5, 5, 5]
print(lista)

# Append()
lista.append(6)
print(lista)

# Insert()
lista.insert(2, 100) # insere o valor 100 na posição 2, deslocando os valores para a d
ireita.
print(lista)

# Len()
print (len(lista)) # imprime o tamanho da lista

# Remove()
lista.remove(5) # remove um valor 5 da lista
print(lista)

# cortar uma lista
l2 = lista[0:4]
print(lista[0:4])
print(l2)

[1, 2, 3, 8, 14, 4, 5, 5, 5]
[1, 2, 3, 8, 14, 4, 5, 5, 5, 6]
[1, 2, 100, 3, 8, 14, 4, 5, 5, 5, 6]
11
[1, 2, 100, 3, 8, 14, 4, 5, 5, 6]
[1, 2, 100, 3]
[1, 2, 100, 3]
1
2
100
3
8
14
4
5
5
6
```