

Disciplina: IPOO - Introdução à Programação Orientada a Objetos

Prof. Luiz Cláudio Dalmolin - luiz.dalmolin@udesc.br - 2023/1

Listas em Python

```
In [10]: # Acesso a uma Lista:  
z = [15,8,9]  
#z[0]  
print(z[2])  
print(z[0])
```

9
15

```
In [12]: # Modificação de uma Lista:  
z = [15,8,9]  
z[0]  
z[0] = 7  
# z[0]  
print(z)
```

[7, 8, 9]

```
In [15]: # Exemplo, cálculo da média:  
notas = [6,7,10,8,9,100]  
soma = 0  
i = 0  
while i < 6:  
    soma += notas[i]  
    i += 1  
print("Média: %.2f" % (soma/i))  
print("Qtde de valores: %d" % i)
```

Média: 23.33
Qtde de valores: 6

```
In [16]: # 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 # i = i + 1  
print("Média: %.2f" % (soma/i))
```

Nota 1: 10
Nota 2: 8.8
Nota 3: 7
Nota 4: 9
Nota 5: 9.5
Média: 8.86

In [4]: *# 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
print("")
while i < 5:
    print("Nota %d: %.2f" % (i+1), notas[i])
    i += 1
print("Média: %.2f" % (soma/i))
```

Nota 1: 4
Nota 2: 8
Nota 3: 6
Nota 4: 7.8
Nota 5: 9

Nota 1: 4.00
Nota 2: 8.00
Nota 3: 6.00
Nota 4: 7.80
Nota 5: 9.00
Média: 6.96

In [20]: *# Exemplo, apresentação de números:*

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

Número 1: 8
Número 2: 4
Número 3: 5
Número 4: 6
Número 5: 4
Que posição quer imprimir '-1' para sair): 4
Você escolheu o número: 6
Que posição quer imprimir '-1' para sair): -1

In [21]: *### 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 [6]: *# 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:*

```
# [inicio : fim] # vai do índice 'início' até o 'fim'-1
# [inicio : ]   # 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 [22]: *# número de elementos de uma lista*

```
L = [1,2,3,4,5,55,8,3,44,4,5,66,7,3,22,44,5]
print(len (L))

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

```
17
0
```

In [23]: *# Exemplo de repetição com tamanho de lista usando len*

```
L = [5,9,88,54,99,66,7,88,3,21]
x = 0
while x < len(L):
    print(L[x])
    x += 1
```

5
9
88
54
99
66
7
88
3
21

In [24]: *# Exemplo de adição de elementos à lista:*

```
L = []  
L.append("a")  
print(L)  
L.append("x")  
L.append("b")  
L.append(33)  
L.append("c")  
print(L)  
len(L)
```

```
['a']  
['a', 'x', 'b', 33, 'c']
```

Out[24]: 5

In [25]: *# Exemplo de adição de elementos à lista:*

```
L = []  
while True:  
    n = float(input("Digite uma nota (-1.0 para sair): "))  
    if n == -1.0:  
        break  
    L.append(n)  
x = 0  
soma = 0  
while x < len(L):  
    print(L[x])  
    soma = soma + L[x]  
    x += 1  
print("Quantidade de notas digitadas: %d" % x)  
print("Media = %.2f" % (soma/x))
```

```
Digite uma nota (-1.0 para sair): 9  
Digite uma nota (-1.0 para sair): 8.5  
Digite uma nota (-1.0 para sair): 7.7  
Digite uma nota (-1.0 para sair): 10  
Digite uma nota (-1.0 para sair): 8  
Digite uma nota (-1.0 para sair): -1  
9.0  
8.5  
7.7  
10.0  
8.0  
Quantidade de notas digitadas: 5  
Media = 8.64
```

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 [8]: # 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[8]: [0, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]
```

```
In [26]: # pesquisa de elementos na lista
L = [15,7,27,39,6,88,100,1,6,7,33]
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: 7
7 achado na posição 1
```

```
In [31]: # Exemplo da impressão de todos os elementos da lista:
L = [8,9,15,8]
for e in L:
    print(e)
# impressão direta da lista:
#L
```

```
8
9
15
8
```

```
In [33]: # Exemplo de pesquisa usando for
L = [7,9,10,12,6,4]
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: 6
6 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 v  
# 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 [34]: # 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 digitada  
soma=0  
for x in alu:  
    soma=soma+ x  
qtde = len(alu)  
media = soma/qtde  
print("Media = ", media)
```

Digite a quantidade de Notas: 5
Digite a 1a. nota: 10
Digite a 2a. nota: 9
Digite a 3a. nota: 8.7
Digite a 4a. nota: 5
Digite a 5a. nota: 7.5
[10.0, 9.0, 8.7, 5.0, 7.5]
Media = 8.0400000000000001

```
In [38]: # Ler 5 inteiros do teclado e armazenar numa Lista.  
# Depois percorrer esta lista mostrando os números ímpares.
```

```

L = []
for i in range(0,5,1):
    L.append(int(input("Digite um número: ")))

for x in L:
    if x % 2 != 0:
        print(x)

```

```

Digite um número: 6
Digite um número: 3
Digite um número: 4
Digite um número: 55
Digite um número: 2
3
55

```

In [2]: *# Ler 5 inteiros do teclado e armazenar numa lista.
Depois percorrer esta lista mostrando os números ímpares.*

```

L = []
n = 0
q = int(input("Quantos números serão digitados? "))
while n < q:
    L.append(int(input("Digite um número: ")))
    n += 1
n = 0
while n < len(L):
    if L[n] % 2 != 0:
        print ("ímpar -> ", L[n])
    n += 1

```

```

Quantos números serão digitados? 4
Digite um número: 55
Digite um número: 10
Digite um número: 8
Digite um número: 7
ímpar -> 55
ímpar -> 7

```

In [1]: *# Ler números inteiros do teclado e armazenar numa lista
(encerra com '0').
Depois percorrer esta lista mostrando o maior número digitado.*

```

L = []
n = 0
while True:
    numero = int(input("Digite um número ('0' para terminar): "))
    if numero == 0:
        break
    L.append(numero)
    # n += 1
maior = L[0]
n = 1
while n < len(L):
    if L[n] > maior:
        maior = L[n]
    n += 1
print("O maior número digitado foi %d" % maior)

```

```
Digite um número ('0' para terminar): 4
Digite um número ('0' para terminar): 6
Digite um número ('0' para terminar): 23
Digite um número ('0' para terminar): 4
Digite um número ('0' para terminar): 1
Digite um número ('0' para terminar): 0
O maior número digitado foi 23
```

```
In [8]: Lista = []
Lista.append("Carlos")
Lista.append("Ana")
Lista.append("Pedro")
Lista.append("José")
compr = len(Lista)
while compr > 0:
    print(Lista[compr-1])
    compr -= 1
```

```
José
Pedro
Ana
Carlos
```

```
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 v
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)
```

```
1
```

```
In [26]: # Imagine uma situação na qual precisa se selecionar os elementos de uma lista a
# 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 is 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 [19]: *# Ordenação de Lista (Bubble sort) e função SORT.*
versão com limite final variável -> <https://www.youtube.com/watch?v=JP5KkzdUE>
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
###
#L = [7,4,3,12,1,8]
#L.sort()
print("ordenado", L)

```

ordenado [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 [ ]: 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 ("ímpar -> ", L[n])  
    n +=1
```

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 ($C[i]=A[i]+B[i]$).

```
In [2]: A=[10, 11, 12, 13, 14, 15, 16,17]
B=[1, 2, 3,4, 5, 6, 7, 8]
C=[]
i = 0
while i < len(A):
    C.append(A[i]+B[i])
    i += 1
print(C)
```

```
[11, 13, 15, 17, 19, 21, 23, 25]
```

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

```
In [3]: L=[]
for i in range(5):
    L.append(float(input("Digite um número: ")))
while i >= 0:
    print(L[i])
    i -= 1
```

```
Digite um número: 3
Digite um número: 4
Digite um número: 1
Digite um número: 6
Digite um número: 4
4.0
6.0
1.0
4.0
3.0
```

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

```
In [5]: L1 = []
L2 = []
L3 = []
while True:
    n = int(input("Digite um número para lista 1('0' p/ fim): "))
    if n == 0:
        break
    L1.append(n)
while True:
    n = int(input("Digite um número para lista 2('0' p/ fim): "))
    if n == 0:
```

```

        break
    L2.append(n)
i=0
while i < len(L1):
    L3.append(L1[i])
    i += 1
i=0
while i < len(L2):
    L3.append(L2[i])
    i += 1
print(L3)

```

```

Digite um número para lista 1('0' p/ fim): 1
Digite um número para lista 1('0' p/ fim): 3
Digite um número para lista 1('0' p/ fim): 5
Digite um número para lista 1('0' p/ fim): 7
Digite um número para lista 1('0' p/ fim): 0
Digite um número para lista 2('0' p/ fim): 2
Digite um número para lista 2('0' p/ fim): 4
Digite um número para lista 2('0' p/ fim): 6
Digite um número para lista 2('0' p/ fim): 0
[1, 3, 5, 7, 2, 4, 6]

```

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

In [10]:

```

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

L = [15,7,27,39,8,7]
p = int(input("Digite o primeiro valor a procurar: "))
v = int(input("Digite o segundo valor a procurar: "))
achouP = False
achouV = False
x = 0
while x < len(L):
    if L[x] is p:
        achouP = True
        posiP=x
    if L[x] is v:
        achouV = True
        posiV=x
    if achouP and achouV:
        break
    x += 1
if achouP and achouV:
    if posiP < posiV:
        print("Acho primeiro o %d na posição %d" % (p,posiP))
    else:
        print("Acho primeiro o %d na posição %d" % (v,posiV))
if achouP:
    print("%d achado na posição %d" % (p,posiP))
if achouV:
    print("%d achado na posição %d" % (v,posiV))
if not achouP and not achouV:
    print("%d e %d não encontrados" % (p,v))

```

Digite o primeiro valor a procurar: 4
Digite o segundo valor a procurar: 3
4 e 3 não encontrados

In [13]: *# 12) Modifique o exercício anterior de forma a pesquisar p e v em toda a lista
ao usuário a posição onde p e a posição onde v foram encontrados.*

```
L = [15,7,27,39,8,7]
p = int(input("Digite o primeiro valor a procurar: "))
v = int(input("Digite o segundo valor a procurar: "))
achouP = False
achouV = False
x = 0
while x < len(L):
    if L[x] is p:
        achouP = True
        print("%d achado na posição %d" % (p,x))
    if L[x] is v:
        achouV = True
        print("%d achado na posição %d" % (v,x))
    x += 1
if not achouP and not achouV:
    print("%d e %d não encontrados" % (p,v))
```

Digite o primeiro valor a procurar: 32
Digite o segundo valor a procurar: 12
32 e 12 não encontrados

In [14]: *# 13) Faça um programa para imprimir o menor elemento de uma lista.*

```
L = [1, 15, 9, 13]
menor = L[0]
for e in L:
    if e < menor:
        menor = e
print(menor)
```

1

In [18]: *# 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
assim como a temperatura média.*

```
T = [ 10, 8, 0, 1, 2, 3, 2, 4]
menor = T[0]
maior = T[0]
soma = 0
for e in T:
    if e < menor:
        menor = e
    if e > maior:
        maior = e
    soma += e
media = soma/len(T)
print("Menor temperatura registrada: %d graus" % menor)
print("Maior temperatura registrada: %d graus" % maior)
print("Menor temperatura registrada: %.2f graus" % media)
```

Menor temperatura registrada: 0 graus
Maior temperatura registrada: 10 graus
Menor temperatura registrada: 3.75 graus

In [20]: *#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].*

```
L = [2,1,5,4,3]
```

```

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
print("ordenado", L)

```

ordenado [5, 4, 3, 2, 1]

In [22]: *# 16) Faça um programa que Leia N valores inteiros, e depois apresente os em ordem*

```

L = []
while True:
    n = int(input("Digite um valor ('0' p/ fim): "))
    if n == 0:
        break
    L.append(n)

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
print("ordenado", L)

```

```

Digite um valor ('0' p/ fim): 6
Digite um valor ('0' p/ fim): 2
Digite um valor ('0' p/ fim): 1
Digite um valor ('0' p/ fim): 5
Digite um valor ('0' p/ fim): 0
ordenado [1, 2, 5, 6]

```

In [18]: *# unindo listas*

```

lojas = ['Rio de Janeiro', 'São Paulo', 'Curitiba']
vendas =[10000,20000,30000]
resultados=[]

for i in range(3):
    resultados.append((lojas[i],vendas[i]))
print(resultados)
print()
for i in range(len(resultados)):
    print("Cidade: %s - Vendas R$ %.2f" % (resultados[i][0],resultados[i][1]))

```

```
[('Rio de Janeiro', 10000), ('São Paulo', 20000), ('Curitiba', 30000)]
```

Cidade: Rio de Janeiro - Vendas R\$ 10000.00

Cidade: São Paulo - Vendas R\$ 20000.00

Cidade: Curitiba - Vendas R\$ 30000.00

```
In [2]: # relação de alunos e notas
notas_ipoo=[]
while True:
    nome=input("Nome do aluno ('fim' p/ encerrar): ")
    if nome == 'fim':
        break
    nota=float(input("nota: "))
    notas_ipoo.append((nome,nota))
print()
soma=0
for i in range(len(notas_ipoo)):
    print("Nome: %s - Nota: %.2f" % (notas_ipoo[i][0],notas_ipoo[i][1]))
    soma+=notas_ipoo[i][1]
print("Quantidade de alunos: %d" % len(notas_ipoo))
print("Média = %.2f" % (soma/len(notas_ipoo)))
```

```
Nome do aluno ('fim' p/ encerrar): r
nota: 6
Nome do aluno ('fim' p/ encerrar): t
nota: 8
Nome do aluno ('fim' p/ encerrar): j
nota: 7
Nome do aluno ('fim' p/ encerrar): fim
```

```
Nome: r - Nota: 6.00
Nome: t - Nota: 8.00
Nome: j - Nota: 7.00
Quantidade de alunos: 3
Média = 7.00
```

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 [19]: 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: Universidade

```

U
Un
Uni
Univ
Unive
Univer
Univers
Universi
Universid
Universida
Universidad
Universidade
niversidade
iversidade
versidade
ersidade
rsidade
sidade
idade
dade
ade
de
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 pa
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
```