



# Estrutura de Dados

## Lista

prof. Guilherme Rey

# Revisão

Programação

Python

Variáveis

Condicionais

While

## operadores.py

```
'''
```

Operadores Aritméticos:

+, -, /, \*, %, \*\*

Operadores para comparação:

==, <, <=, >, >=, !=

Operadores lógicos:

and, or, not

Operadores de atribuição:

=, +=, -=, \*=, /=, %=, \*\*=, //=

```
'''
```

comments.py

```
# Comentário
# Comentário 2
# Comentário 3

''' Comentário
Comentário 2
Comentário 3
'''
```



**Guido van Rossum**  @gvanrossum · Sep 10, 2011

Replying to [@BSUCSClub](#)

[@BSUCSClub](#) Python tip: You can use multi-line strings as multi-line comments. Unless used as docstrings, they generate no code! :-)

 9

 70

 130



variaveis.py

```
# Principais tipos  
texto = 'Sou um texto'  
inteiro = 20      # integer  
decimal = 3.14    # float  
verdade = True    # booleano (True/False)
```

input\_output.py

```
mensagem = input('Qual mensagem?')  
  
print(mensagem)
```

while\_com\_condicionais.py

```
while True:
    area = input('Qual é a área mais legal?')
    if area == 'Computação':
        print('Acertou! Está livre!')
        break
    else:
        print('Poxa, tenta de novo!')
```



while\_exemplo2\_contador.py

```
contador = 0
N = 1
t = 1
while contador < N:
    print(t)      # imprime o valor de t no momento
    t *= 2        # equivalente a t = t * 2
    contador += 1
```

exercicio\_revisao.py

```
# Escrever um programa que receba N  
# notas, até que o usuário insira  
# 'compute'. Depois de receber as  
# notas, imprimir a média delas
```

```
total = 0
contador = 0
while True:
    inp = input('Digite uma nota (ou compute) ')
    if inp == 'compute':
        break
    valor = float(inp)
    total += valor
    contador += 1
print(total / contador)
```

# Estruturas de dados

Como guardar vários valores  
em uma única variável?

# Lista

declaracao\_listas.py

```
lista_vazia = []  
alguns_numeros = [1, 2, 3, 4, 5]  
algumas_palavras = ['oi', 'td', 'bl']
```

declaracao\_listas.py

```
algumas_coisas = [1, 'b', 3, 'd']
```



Uma lista é uma coleção de  
valores

entendendo\_conceito.py

```
lista = [1, 2, 3, 4, 5, 6, -7]
```

entendendo\_conceito.py

```
lista = [1, 2, 3, 4, 5, 6, -7]
```

1

2

3

4

5

6

7

entendendo\_conceito.py

```
lista = [1, 2, 3, 4, 5, 6, -7]
```

1

2

3

4

5

6

7

entendendo\_conceito.py

```
lista = [1, 2, 3, 4, 5, 6, -7]
```



*cada valor é acessado  
pelo índice da posição*

entendendo\_conceito.py

```
lista = [1, 2, 3, 4, 5, 6, -7]
```



*cada valor é acessado  
pelo índice da posição*

entendendo\_conceito.py

```
lista = [1, 2, 3, 4, 5, 6, -7]
```



*cada valor é acessado  
pelo índice da posição*

```
lista[0] # é 1
```

# Operações com listas



operacoes\_listas.py

```
algumas_coisas = [1, 'b', 3, 'd']  
# Operações com a lista  
print(algumas_coisas)  
print(algumas_coisas * 2)
```

# Problema

operacoes\_listas.py

```
# Escrever um programa que receba N  
# notas, até que o usuário insira  
# 'compute'. Depois de receber as  
# notas, imprimir a média delas
```

```
total = 0
contador = 0
while True:
    inp = input('Digite uma nota (ou compute)')
    if inp == 'compute':
        break
    valor = float(inp)
    total += valor
    contador += 1
print(total / contador)
```

```
# declarar uma lista de notas
while True:
    inp = input('Digite uma nota (ou compute)')
    if inp == 'compute':
        break
    valor = float(inp)
    # adicionar valor na lista de notas
# imprimir soma dos valores divididos pela
quantidade de notas
```

```
notas = []  
while True:  
    inp = input('Digite uma nota (ou compute)')  
    if inp == 'compute':  
        break  
    valor = float(inp)  
    # adicionar valor na lista de notas  
# imprimir soma dos valores divididos pela  
quantidade de notas
```

```
notas = []  
while True:  
    inp = input('Digite uma nota (ou compute)')  
    if inp == 'compute':  
        break  
    valor = float(inp)  
    notas.append(valor)  
# imprimir soma dos valores divididos pela  
quantidade de notas
```

```
notas = []  
while True:  
    inp = input('Digite uma nota (ou compute)')  
    if inp == 'compute':  
        break  
    valor = float(inp)  
    notas.append(valor)  
print(sum(notas) / len(notas))
```



# Métodos de lista

<code>.append(val)</code>	adiciona <b>val</b> no fim da lista
<code>.insert(pos, val)</code>	insere <b>val</b> na posição <b>pos</b>
<code>.remove(val)</code>	remove da lista, procurando <b>val</b>
<code>.pop([i])</code>	remove da lista pelo índice <b>i</b> , retornando o valor encontrado
<code>.clear()</code>	limpa a lista
<code>.index(val)</code>	retorna o índice do valor <b>val</b>
<code>.count(val)</code>	retorna a qtd de vezes que <b>val</b> aparece
<code>.reverse()</code>	inverte a ordem dos elementos

<code>.sort()</code>	ordena os elementos da lista
<code>len(lista)</code>	retorna o número de elementos da lista
<code>sum(lista)</code>	soma os elementos de lista
<code>max(lista)</code>	retorna o maior valor encontrado em lista
<code>min(lista)</code>	retorna o menor valor encontrado em lista
<code>shuffle(lista)</code>	embaralha a lista

# slicing

```
a[inicio:fim]      # itens começando em inicio até fim - 1
a[inicio:]         # itens começando em inicio até o final
a[:fim]           # itens do começo até fim - 1
a[:]              # copia a lista inteira
```

```
a[inicio:fim:passo] # inicio por passo, até fim - 1
```

```
a[-1]      # último item na lista
```

```
a[-2:]     # dois últimos itens na lista
```

```
a[:-2]     # tudo, menos os dois últimos itens da lista
```

```
a[::-1]    # todos os itens da lista, ao contrário
```

# Loop, de novo!

loop\_2.py

```
# Escrever um programa que imprima  
# os 10 primeiros números inteiros
```



loop\_2.py

```
i = 0
while i < 10:
    print(i)
    i += 1
```

loop\_2.py

```
for i in range(10):  
    print(i)
```

loop\_2.py

```
amigos = ['Joey', 'Ross', 'Monica',  
'Rachel', 'Phoebe', 'Chandler']  
for amigo in amigos:  
    print(amigo + ' pegou um café')
```

`aula += 1`