

Estrutura de Dados Lista

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! :-)

0

9

1 70

0

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
\overline{N} = 1
t = 1
while contador < N:
  print(t) # imprime o valor de t no momento
  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
```

```
while exemplo2 contador.py
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

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

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

1 2 3 4 5 6 7

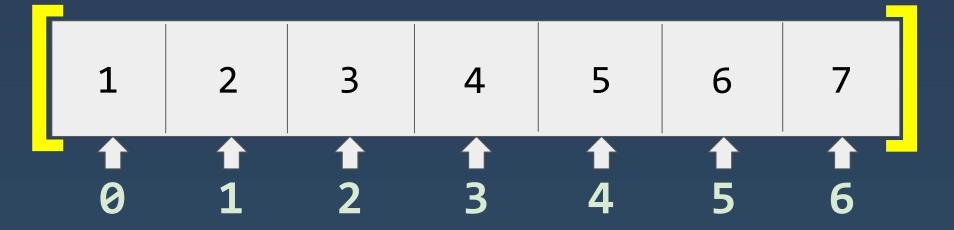
lista = [1, 2, 3, 4, 5, 6, -7]

1 2 3 4 5 6 7

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

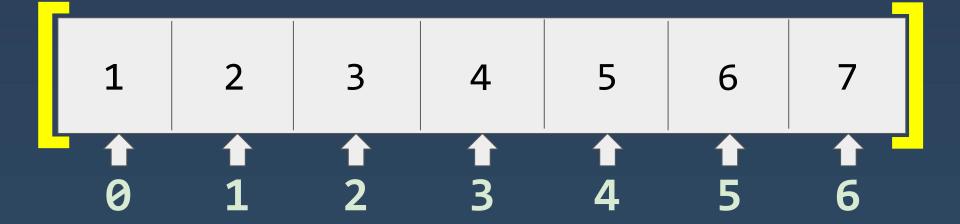
1 2 3 4 5 6 7

cada valor é acessado pelo índice da posição lista = [1, 2, 3, 4, 5, 6, -7]



cada valor é acessado pelo índice da posição

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
```

```
while exemplo2 contador.py
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)
```

```
while exemplo2 contador.py
# 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
```

```
while_exemplo2_contador.py
```

```
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
```

```
while exemplo2 contador.py
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
```

```
while exemplo2 contador.py
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

.append(val)

adiciona val no fim da lista

.insert(pos, val)

insere val na posição pos

.remove(val)

remove da lista, procurando val

.pop([i])

remove da lista pelo índice i, retornando o valor encontrado

.clear()

limpa a lista

.index(val)

retorna o índice do valor val

.count(val)

retorna a qtd de vezes que val aparece

.reverse()

inverte a ordem dos elementos

() ordena os	elementos da lista
ista) retorna o núr	nero de elementos da lista
ista) soma os	elementos de lista
ista) retorna o maio	or valor encontrado em lista
ista) retorna o meno	or valor encontrado em lista
le(lista)	embaralha a lista

slicing

```
slicing_arrays.py
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
                 # copia a lista inteira
a[:]
```

```
slicing_arrays_2.py
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