

Aula 02 - Python: Listas



A estrutura de dados mais básica em Python é a **sequencia**. Cada elemento de uma sequencia recebe um número - sua posição ou índice. O primeiro índice é zero, o segundo índice é um, e assim por diante.

O Python possui seis tipos de sequências incorporados, mas os mais comuns são as **listas** e as **tuplas**, que veremos neste tutorial.

A **lista** é o tipo de dados mais versátil disponível no Python, que pode ser escrito como uma lista de valores (itens) separados por vírgulas entre colchetes.

Um dos pontos importante sobre uma lista é que os itens de uma lista não precisam ser do mesmo tipo.

Aula 02 - Python: Listas



```
In [9]: list01 = ['Ze Maria', 'Pedro Augusto', 'Larissa Magalhaes', 'Felisberta Novaes']
list02 = [1, 2, 3, 4, 5, 7]
list03 = [3.2, 3, 'Cachorrinho', ['Gato', 'Preto'], 4.5]

print (list01)
print (list02)
print (list03)

['Ze Maria', 'Pedro Augusto', 'Larissa Magalhaes', 'Felisberta Novaes']
[1, 2, 3, 4, 5, 7]
[3.2, 3, 'Cachorrinho', ['Gato', 'Preto'], 4.5]
```

Aula 02 - Python: Listas



```
In [12]: list01 = [1, 2, 3, 'Gatinho', 'Cachorrinho', 'Porquinho']  
print ('list01[0]', list01[0])  
list01[0] 1
```

```
In [13]: list01 = [1, 2, 3, 'Gatinho', 'Cachorrinho', 'Porquinho']  
print ('ultimo elemento de list01 ', list01[-1])  
ultimo elemento de list01 Porquinho
```

```
In [15]: list01 = [1, 2, 3, 'Gatinho', 'Cachorrinho', 'Porquinho']  
print ('os três elementos da lista são ', list01[-3:])  
os três elementos da lista são ['Gatinho', 'Cachorrinho', 'Porquinho']
```

```
In [19]: list01 = [1, 2, 3, 'Gatinho', 'Cachorrinho', 'Porquinho']  
print ('terceira posição até o penultimo elemento da lista ', list01[2:-1])  
terceira posição até o penultimo elemento da lista [3, 'Gatinho', 'Cachorrinho']
```

Aula 02 - Python: Listas

```
In [26]: nome = 'Ze Maria'
```

```
print (nome[3])
```

```
nome[3] = 'Z'
```

M

```
-----  
TypeError                                Traceback (most recent call last)
```

```
<ipython-input-26-9f1e5628e433> in <module>()
```

```
    3 print (nome[3])
```

```
    4
```

```
----> 5 nome[3] = 'Z'
```

```
TypeError: 'str' object does not support item assignment
```

```
In [23]: list01 = [1, 2, 3, 'Gatinho', 'Cachorrinho', 'Porquinho']
```

```
list01[3] = 3.14
```

```
print (list01)
```

```
[1, 2, 3, 3.14, 'Cachorrinho', 'Porquinho']
```

Aula 02 - Python: Listas



```
In [28]: list01 = []  
list01.append('Oi')  
list01.append('eu')  
list01.append('estou')  
list01.append('aqui')  
  
print (list01)  
  
['Oi', 'eu', 'estou', 'aqui']
```

```
In [31]: list01 = ['Oi', 'eu', 'estou', 'aqui']  
print (list01)  
  
del list01[2]  
  
print (list01)  
  
['Oi', 'eu', 'estou', 'aqui']  
['Oi', 'eu', 'aqui']
```

Aula 02 - Python: Listas

len - tamanho da lista

```
In [33]: lst = ['Vaca', 'Galinha', 'Cachorro', 'Gato', 'Pavão']  
         print (len(lst))  
5
```

concatenação de listas

```
In [35]: lst_fazenda = ['Vaca', 'Cavalo', 'Galinha']  
         lst_casa    = ['Gato', 'Cachorro']  
  
         lst_animais = lst_fazenda + lst_casa  
  
         print (len(lst_animais))  
         print (lst_animais)  
5  
['Vaca', 'Cavalo', 'Galinha', 'Gato', 'Cachorro']
```

Aula 02 - Python: Listas



Replicação

```
In [38]: lst = ['Ola'] * 5
print (len(lst))
print (lst)

5
['Ola', 'Ola', 'Ola', 'Ola', 'Ola']
```

```
In [43]: lst = ['Ola']
lst_a = ['a'] * 10
lst_b = lst + lst_a

print (len(lst_b))
print (lst)
print (lst_a)
print (lst_b)
print (lst[0]+lst_a[0])

11
['Ola']
['a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a']
['Ola', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a']
Olaa
```


Aula 02 - Python: Listas



```
In [44]: lst = ['Ola']
          lst_a = ['a'] * 10
          lst_b = lst + lst_a

          print (len(lst_b))
          print (lst)
          print (lst_a)
          print (lst_b)
          print (lst[0]+lst_a[0:])

11
['Ola']
['a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a']
['Ola', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a', 'a']

-----
TypeError                                 Traceback (most recent call last)
<ipython-input-44-33e9543905f5> in <module>()
      7 print (lst_a)
      8 print (lst_b)
----> 9 print (lst[0]+lst_a[0:])

TypeError: Can't convert 'list' object to str implicitly
```


Aula 02 - Python: Listas

Existe em

```
In [48]: lst = ['Vaca', 'Cavalo', 'Galinha', 'Gato', 'Cachorro']

if 'Vaca' in lst:
    print ('Tem vaca sim')
else:
    print ('Sem churrasco')

Tem vaca sim
```

for em uma lista

```
In [50]: lst = ['Vaca', 'Cavalo', 'Galinha', 'Gato', 'Cachorro']

for animal in lst:
    print (animal)

Vaca
Cavalo
Galinha
Gato
Cachorro
```

Aula 02 - Python: Tuplas

Uma **tupla** é uma sequência de objetos Python imutáveis. Os Tuples são sequências, assim como listas. A principal diferença entre as tuplas e as listas é que as tuplas não podem ser alteradas ao contrário das listas. Os Tuples usam parênteses, enquanto as listas usam colchetes.

Criar uma tupla é tão simples como colocar diferentes valores separados por vírgulas. Opcionalmente, você pode colocar esses valores separados por vírgulas entre parênteses também.

```
In [1]: tup1 = ('physics', 'chemistry', 1997, 2000)
        tup2 = (1, 2, 3, 4, 5)
        tup3 = "a", "b", "c", "d"
        tup4 = () # tupla vazia
        tup5 = (50,) # tupla com um unico valor

        print (tup1)
        print (tup2)
        print (tup3)
        print (tup4)
        print (tup5)

('physics', 'chemistry', 1997, 2000)
(1, 2, 3, 4, 5)
('a', 'b', 'c', 'd')
()
(50,)
```

Aula 02 - Python: Tuplas

```
In [4]: tup1 = ('fisica', 'quimica', 1997, 2000)
        tup2 = (1, 2, 3, 4, 5, 6, 7 )

        print ("tup1[0]: ", tup1[0])
        print ("tup2[1:5]: ", tup2[1:5])

        tup1[0]: fisica
        tup2[1:5]: (2, 3, 4, 5)
```

Os Tuples são **imutáveis**, o que significa que você não pode atualizar ou alterar os valores dos elementos da tupla. Você pode pegar porções das tuplas existentes para criar novas tuplas

```
In [6]: tup1 = (12, 34.56)
        tup2 = ('abc', 'xyz')

        tup3 = tup1 + tup2
        print (tup3)

        (12, 34.56, 'abc', 'xyz')
```

Aula 02 - Python: Tuplas



```
In [11]: tup = ('fisica', 'quimica', 1997, 2000);  
  
print (tup)  
del tup  
  
print (tup)  
  
('fisica', 'quimica', 1997, 2000)  
  
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-11-52f09bbb43bb> in <module>()  
      4 del tup  
      5  
> 6 print (tup)  
  
NameError: name 'tup' is not defined
```

Aula 02 - Python: Tuplas

Python Expression	Results	Description
<code>len((1, 2, 3))</code>	3	Length
<code>(1, 2, 3) + (4, 5, 6)</code>	<code>(1, 2, 3, 4, 5, 6)</code>	Concatenation
<code>('Hi!') * 4</code>	<code>('Hi!', 'Hi!', 'Hi!', 'Hi!')</code>	Repetition
<code>3 in (1, 2, 3)</code>	True	Membership
<code>for x in (1,2,3) : print (x, end = ' ')</code>	1 2 3	Iteration

https://www.tutorialspoint.com/python3/python_tuples.htm

Aula 02 - Python: Tuplas de/para Listas

```
In [15]: lst = ['physics', 'chemistry', 1997, 2000]
          tup = ('casa', 'sala', 'escola', 2017, 2011)

          lst[0] = 'fisica'
          lst[1] = 'quimica'

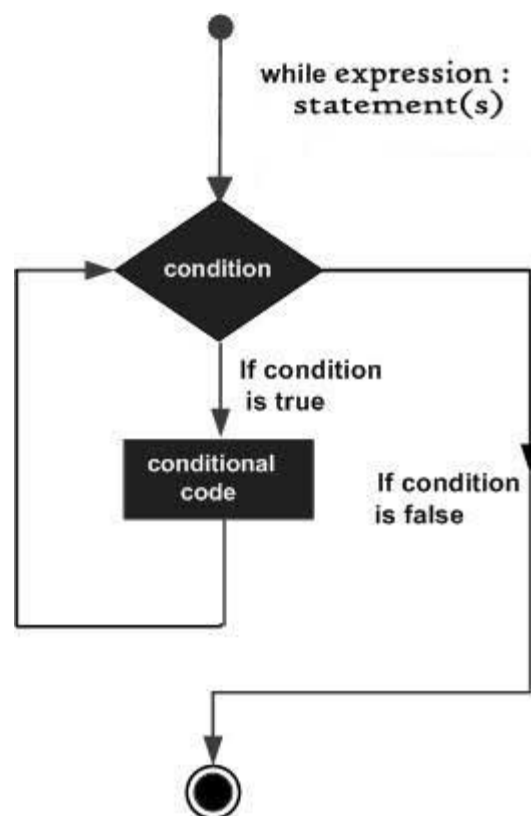
          print (lst)
          print (tup)

          tup2 = tup + tuple(lst)
          print (tup2)

          lst2 = lst + list(tup)
          print (lst2)

['fisica', 'quimica', 1997, 2000]
('casa', 'sala', 'escola', 2017, 2011)
('casa', 'sala', 'escola', 2017, 2011, 'fisica', 'quimica', 1997, 2000)
['fisica', 'quimica', 1997, 2000, 'casa', 'sala', 'escola', 2017, 2011]
```

Aula 02 - Python: While



```
In [1]: count = 0
while (count < 9):
    print ('o valor de count é...: %d'% count)
    count = count + 1

print (".| fim da laço")
```

o valor de count é...: 0
o valor de count é...: 1
o valor de count é...: 2
o valor de count é...: 3
o valor de count é...: 4
o valor de count é...: 5
o valor de count é...: 6
o valor de count é...: 7
o valor de count é...: 8
.| fim da laço

Aula 02 - Python: While

```
In [*]: var = 1
while var == 1 : # laço infinito
    num = int(input("informe número :"))
    print ("o número informado foi...: %d"% num)

print (".| fim do laço")
```

```
informe número :2
o número informado foi...: 2
informe número :3
o número informado foi...: 3
informe número :2
o número informado foi...: 2
informe número :0
o número informado foi...: 0
informe número :1
o número informado foi...: 1
informe número :1
o número informado foi...: 1
```

```
informe número : 
```

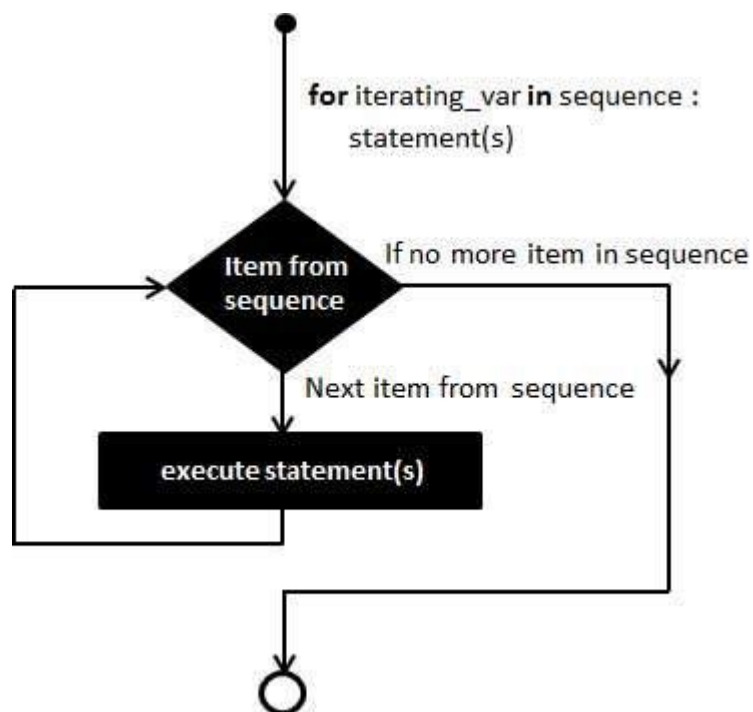
Aula 02 - Python: While



```
In [4]: count = 0
while count < 5:
    print ("%d é menor que 5" % count)
    count = count + 1
else:
    print ("%d não é menor que 5" % count)
```

```
0 é menor que 5
1 é menor que 5
2 é menor que 5
3 é menor que 5
4 é menor que 5
5 não é menor que 5
```

Aula 02 - Python: for



```
In [4]: for var in range(5):
        print (var)
```

```
0
1
2
3
4
```

Aula 02 - Python: for



```
In [9]: for letra in 'Python':  
        print ('letra atual do laco : %c' % letra)  
  
frutas = ['banana', 'maca', 'manga']  
for fruta in frutas:  
    print ('fruta corrente do laco :', fruta)
```

```
letra atual do laco : P  
letra atual do laco : y  
letra atual do laco : t  
letra atual do laco : h  
letra atual do laco : o  
letra atual do laco : n  
fruta corrente do laco : banana  
fruta corrente do laco : maca  
fruta corrente do laco : manga
```

Aula 02 - Python: for



```
In [14]: frutas = ['banana', 'maca', 'manga']

print ('len(frutas) %d ' % len(frutas))
print (range(len(frutas)))

for index in range(len(frutas)):
    print ('fruta corrente do laco :', frutas[index])

len(frutas) 3
range(0, 3)
fruta corrente do laco : banana
fruta corrente do laco : maca
fruta corrente do laco : manga
```


Linguagem de Programação Python

