

Listas Python... Continuação!



Já vimos

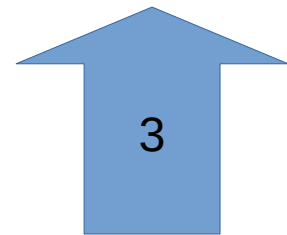
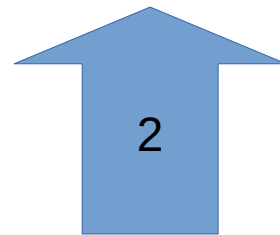
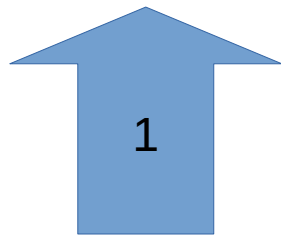
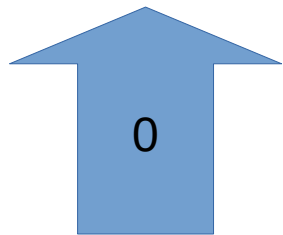
- Como criar listas:

```
lista = [ "Abacaxi", "Banana", "Detergente",  
"Pão" ]
```

Já vimos

- Como está ordenada a lista:

["Abacaxi", "Banana", "Detergente", "Pão"]



Já vimos

- Com acessar elementos da lista

Lista = ["Abacaxi", "Banana", "Detergente",
"Pão"]

- "Banana"
 - lista[1]

Já vimos

- Com acessar elementos da lista

Lista = ["Abacaxi", "Banana", "Detergente",
"Pão"]

- ["Banana", "Detergente"]
 - Lista[1:3]

O que podemos fazer com a lista

- Para trabalharmos com a lista precisamos conhecer algumas funções básicas e alguns métodos da lista.

Funções básicas

- `min()`
- `max()`
- `sum()`
- `len()`

Função min()

- A função min() retorna o menor valor da lista.

```
>>> lista = [ 5, 3, 8, 1, 9, 10 ]
```


Função min()

- A função min() retorna o menor valor da lista.

```
>>> lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> min( lista )
```

Função min()

- A função min() retorna o menor valor da lista.

```
>>> lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> min( lista )
```

```
1
```

Função min()

- Você pode guardar o valor retornado da função min() em uma variável

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> minimo = min( lista )
```

Função min()

- Você pode guardar o valor retornado da função min() em uma variável

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> minimo = min( lista )
```

```
>>> print( minimo )
```

```
1
```

Função max()

- A função max() retorna o maior valor da lista

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> maximo = max( lista )
```

Função max()

- A função max() retorna o maior valor da lista

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> maximo = max( lista )
```

```
>>> print( maximo )
```

Função max()

- A função max() retorna o maior valor da lista

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> maximo = max( lista )
```

```
>>> print( maximo )
```

```
10
```

Função sum()

- A função sum() irá retornar a soma de todos os elementos da lista

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> soma = sum( lista )
```


Função sum()

- A função sum() irá retornar a soma de todos os elementos da lista

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> soma = sum( lista )
```

```
>>> print( soma )
```

Função sum()

- A função sum() irá retornar a soma de todos os elementos da lista

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> soma = sum( lista )
```

```
>>> print( soma )
```

```
36
```

Função len()

- A função len() conta quantos elementos tem em uma lista.

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> quantidade = len( lista )
```

Função len()

- A função len() conta quantos elementos tem em uma lista.

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> quantidade = len( lista )
```

```
>>> print( quantidade )
```

Função len()

- A função len() conta quantos elementos tem em uma lista.

```
lista = [ 5, 3, 8, 1, 9, 10 ]
```

```
>>> quantidade = len( lista )
```

```
>>> print( quantidade )
```

```
6
```

E os métodos???

- Os métodos são funções da lista.
- Eles são acessados de outra forma.

Funções vs Métodos

- Função

`len()`

`len(lista)`

Método

`lista = [1, 2, 3]`

`lista.pop()`

Métodos da lista

- `.append()`
- `.insert()`
- `.pop()`
- `.remove()`
- `.sort()`
- `.reverse()`

Método .append()

- Insere um elemento no final da lista

```
>>> lista = [ ]
```

Método .append()

- Insere um elemento no final da lista

```
>>> lista = [ ]
```

```
>>> lista.append(  )
```

Método .append()

- Insere um elemento no final da lista

```
>>> lista = []
```

```
>>> lista.append( ■ )  
[ ■ ]
```

Método .append()

- Insere um elemento no final da lista

```
>>> lista = []
```

```
>>> lista.append( ■ )  
[ ■ ]
```

```
>>> lista.append( ◆ )
```

Método .append()

- Insere um elemento no final da lista

```
>>> lista = []
```

```
>>> lista.append(  )  
[  ]
```

```
>>> lista.append(  )  
[ ,  ]
```

Método .append()

- Insere um elemento no final da lista

```
>>> lista = []
```

```
>>> lista.append(  )  
[  ]
```

```
>>> lista.append(  )  
[ ,  ]
```

```
>>> lista.append(  )
```

Método .append()

- Insere um elemento no final da lista

```
>>> lista = []
```

```
>>> lista.append(  )  
[  ]
```

```
>>> lista.append(  )  
[ ,  ]
```

```
>>> lista.append(  )  
[ , ,  ]
```

Método .insert()

- Este método adiciona um elemento em uma posição específica da lista.
- Você determina qual o índice e o elemento a ser adicionado.
- `.insert(index, elemento)`

Método .insert()

```
>>> lista = [ , ,  ]
```

Método .insert()

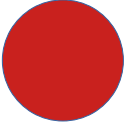
```
>>> lista = [ , ,  ]
```

- ```
>>> lista.insert (1, )
```

# Método .insert()

```
>>> lista = [, , ]
```

- ```
>>> lista.insert ( 1,  )
```

```
[ , , ,  ]
```

Método .pop()

- O método .pop() remove o ultimo elemento da lista e retorna para o programa

```
>>> lista = [ , , ,  ]
```

Método .pop()

- O método .pop() remove o ultimo elemento da lista e retorna para o programa

```
>>> lista = [ , , ,  ]
```

```
>>> ultimo = lista.pop( )
```

Método .pop()

- O método .pop() remove o ultimo elemento da lista e retorna para o programa

```
>>> lista = [ , , ,  ]
```

```
>>> ultimo = lista.pop( )
```

```
>>> print()
```


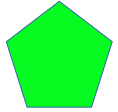
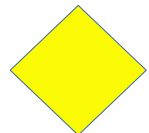
Método .pop()

- O método .pop() remove o ultimo elemento da lista e retorna para o programa

```
>>> lista = [ , , ,  ]
```

```
>>> ultimo = lista.pop( )
```

```
>>> print()
```

```
[ , ,  ]
```

Método .pop()

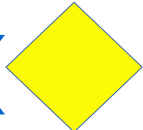
- Podemos também determinar um índice para o .pop() remover

```
>>> lista = [ , , , 
```


Método .pop()

- Podemos também determinar um índice para o .pop() remover

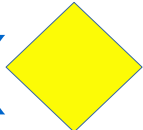
```
>>> lista = [ , , ,  ]
```

```
>>> elemento = lista.pop()
```

Método .pop()

- Podemos também determinar um índice para o .pop() remover

```
>>> lista = [ , , , 
```

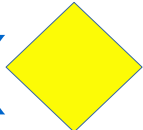
```
>>> elemento = lista.pop()
```

- ```
>>> print(elemento)
```

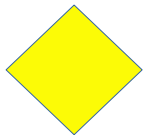
# Método .pop()

- Podemos também determinar um índice para o .pop() remover

```
>>> lista = [, , , ]
```

```
>>> elemento = lista.pop()
```

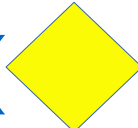
- ```
>>> print( elemento )
```



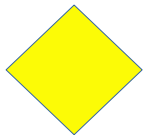
Método .pop()


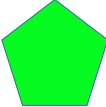

- Podemos também determinar um índice para o .pop() remover

```
>>> lista = [ , , , 
```

```
>>> elemento = lista.pop()
```

- ```
>>> print(elemento)
```


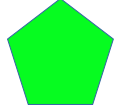
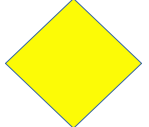
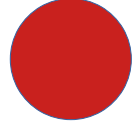
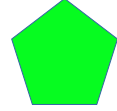


```
[, , 
```


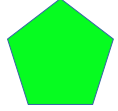
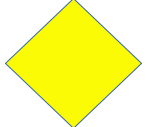
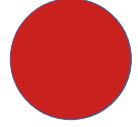
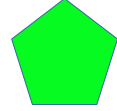
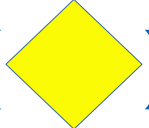
# Método .remove()

- O Método .remove( ) irá remover um elemento pelo valor dele.
- Este método não retorna o elemento removido.


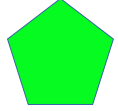
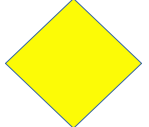
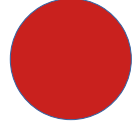
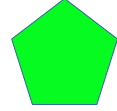
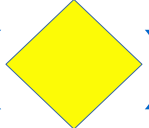
# Método .remove()

```
>>> lista = [, , , , ]
```

# Método .remove()

```
>>> lista = [, , , , ]
>>> lista.remove()
```

# Método .remove()

```
>>> lista = [, , , , ]
>>> lista.remove()
```


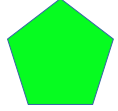
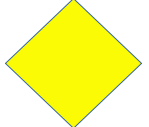
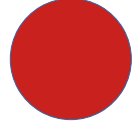
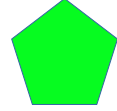
```
[, , , ]
```




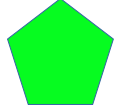
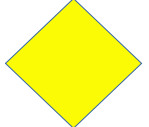
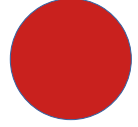
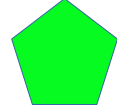
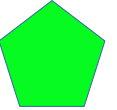
# Método .remove()

- Em caso de item repetitivo, o .remove() irá eliminar o primeiro elemento.



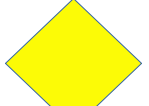

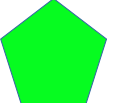
# Método .remove()

```
>>> lista = [, , , , ]
```


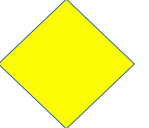


# Método .remove()

```
>>> lista = [, , , , ]
>>> lista.remove()
```

# Método .remove()

```
>>> lista = [, , , , ]
```

```
>>> lista.remove()
```

```
[, , , ]
```

# Método .sort()

- O método .sort() irá ordenar os elementos da lista.

```
>>> lista = [5, 3, 8, 1, 9, 10]
```

# Método .sort()

- O método .sort() irá ordenar os elementos da lista.

```
>>> lista = [5, 3, 8, 1, 9, 10]
```

```
>>> lista.sort()
```

# Método .sort()

- O método .sort() irá ordenar os elementos da lista.

```
>>> lista = [5, 3, 8, 1, 9, 10]
```

```
>>> lista.sort()
```

```
[1, 3, 5, 8, 9, 10]
```

# Método .reverse()

- O método .reverse() ele inverte os elementos da lista.

```
>>> lista = [5, 3, 8, 1, 9, 10]
```



# Método .reverse()

- O método .reverse() ele inverte os elementos da lista.

```
>>> lista = [5, 3, 8, 1, 9, 10]
```

```
>>> lista.reverse()
```

# Método .reverse()

- O método .reverse() ele inverte os elementos da lista.

```
>>> lista = [5, 3, 8, 1, 9, 10]
```

```
>>> lista.reverse()
```

```
[10, 9, 1, 8, 3, 5]
```