



\_

```
Função, método e palavra
reservada da linguagem

>>> lista = [1, 2, 3, 4, 5]
>>> # len() é uma função
>>> len(lista)
5
>>> lista
[1, 2, 3, 4, 5]
>>> # .append() é um método do objeto lista
>>> lista
[1, 2, 3, 4, 5, 6]
>>> # del é uma palavra reservada do Python
>>> del lista[0]
>>> lista
[2, 3, 4, 5, 6]
```

```
Função, método e palavra
reservada da linguagem

>>> lista = [3, 2, 6, 4, 7, 1, 9, 5, 0, 8]
>>> max(lista)
9
>>> print(min(lista))
0
>>> print(lista.sort())
None
>>> lista
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

3

```
Função, método e palavra reservada da linguagem

>>> lista
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> del lista[3]
>>> lista
[0, 1, 2, 4, 5, 6, 7, 8, 9]
>>> del lista[3:5]
>>> lista
[0, 1, 2, 6, 7, 8, 9]
```

5

```
Função, método e palavra reservada da linguagem

>>> lista
[0, 1, 2, 6, 7, 8, 9]
>>> lista.append('X')
>>> lista.insert(0, 'X')
>>> lista.insert(5, 'X')
>>> lista
['X', 0, 1, 2, 6, 'X', 7, 8, 9, 'X']
```

1

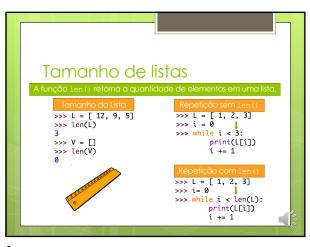
```
Função, método e palavra reservada da linguagem

>>> lista
['X', 0, 1, 2, 6, 'X', 7, 8, 9, 'X']
>>> lista.index('X')
0
>>> lista.index('X', 1)
5
>>> lista.index('X', 6, len(lista))
9
```

```
Função, método e palavra
reservada da linguagem

>>> lista.index('Y')
Traceback (most recent call last):
File "<pyshell#114>", line 1, in <module>
lista.index('Y')
ValueError: 'Y' is not in list
>>> 'Y' in lista
False
```

7



```
Inverter Lista

O método reverse () retorna a lista na ordem invertida.

>>> lista = [1, 2, 3, 4, 5]
>>> lista
[1, 2, 3, 4, 5]
>>> lista.reverse ()
>>> lista
[5, 4, 3, 2, 1]
>>> print (lista.reverse ())
None
>>> lista
[1, 2, 3, 4, 5]

Palíndromos
Padra de la para recersa (1, 2, 3, 4, 5)

None
>>> lista
[1, 2, 3, 4, 5]
```

9 10

```
Carregando uma Lista
from random import randrange, seed
seed()

def carrega lista(qtd_elementos):
    minha lista = []
    for i in range(qtd_elementos):
        minha_lista.append(randrange(1000))
    return minha_lista

def main():
    l = carrega_lista(5)
    print(1)

if __name__ == '__main__':
    main()
```

```
Percorrer Lista

def contar(lista):
   total = 0
   for elemento in lista:
        total += 1
   return total

def imprimir(lista):
   for indice in range(len(lista)):
        print(lista[indice], end=' ')
```

11 12

2





13 14