

Linguagem de Programação Python

Listas e Tuplas

Professor: Ritormar Torquato

1

Listas e Tuplas

Objetivos: Funções e Métodos de Listas; Carregar e Imprimir; Inverter; Contar elementos; Menor, maior e soma;

2

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

3

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]
```

4

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']
```

6

## 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
```

7

## 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
```

8

## Tamanho de listas

A função `len()` retorna a quantidade de elementos em uma lista.

### Tamanho da Lista

```
>>> L = [12, 9, 5]
>>> len(L)
3
>>> V = []
>>> len(V)
0
```



### Repetição sem len()

```
>>> L = [1, 2, 3]
>>> i = 0
>>> while i < 3:
>>>     print(L[i])
>>>     i += 1
```

### Repetição com len()

```
>>> L = [1, 2, 3]
>>> i = 0
>>> while i < len(L):
>>>     print(L[i])
>>>     i += 1
```

9

## 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]
```



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(l)

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

11

## 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=' ')
```

12

## Aplicações com Índices

Nas listas cada valor é acessado por seu índice.

Nomes

0	1	2	3	4	5	6
Carla	Ana	Bia	Hugo	Dani	Ana	Ísis

Notas

0	1	2	3	4	5	6
8.6	7.8	7.5	8.5	8.9	9.2	9.0

Quem tirou a maior e a menor nota?



13

## Aplicações com Índices

Nas listas cada valor é acessado por seu índice.

```
nomes = ['Carla', 'Ana', 'Bia', 'Hugo', 'Dani', 'Ana', 'Ísis']  
notas = [8.6, 7.8, 7.5, 8.5, 8.9, 9.2, 9.0]
```

```
indice_maior_nota = notas.index(max(notas))  
indice_menor_nota = notas.index(min(notas))
```

```
print(f'A maior nota foi de {nomes[indice_maior_nota]}')  
print(f'A menor nota foi de {nomes[indice_menor_nota]}')
```

Quem tirou a maior e a menor nota?



14