

Computação I - Python

Aula 6: Fatiamento e Manipulação de Listas

Manipulação de Listas: Outras Funções

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



Manipulação de listas: Outras funções

- **Função COUNT:** Retorna quantas vezes um elemento aparece em uma lista.
Utilização: `list.count(lista, elemento)`

Manipulação de listas: Outras funções

- **Função COUNT:** Retorna quantas vezes um elemento aparece em uma lista.
Utilização: **list.count(lista, elemento)**

```
>>> L = [0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1]
>>> qtd_zero = list.count(L, 0)
>>> qtd_zero
9
>>> qtd_um = list.count(L, 1)
>>> qtd_um
8
>>> qtd_dois = list.count(L, 2)
>>> qtd_dois
0
```

Caso o elemento procurado não exista na lista, a função retorna 0.

Manipulação de listas: Outras funções

- **Função INDEX:** Retorna o índice da primeira ocorrência em que um elemento aparece em uma lista.
Utilização: **`list.index(lista, elemento)`** ou **`list.index(lista, elemento, start, stop)`**

Manipulação de listas: Outras funções

- **Função INDEX:** Retorna o índice da primeira ocorrência em que um elemento aparece em uma lista.

Utilização: `list.index(lista, elemento)` ou `list.index(lista, elemento, start, stop)`

```
>>> L = [0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1]
>>> list.index(L, 1)
1
>>> list.index(L, 1, 4)
6
>>> list.index(L, 1, 4, 6)
Traceback (most recent call last):
  File "<pyshell#55>", line 1, in <module>
    list.index(L, 1, 4, 6)
ValueError: 1 is not in list
```

Com apenas 2 argumentos, procura-se o elemento do início até o final da lista.

Com 3 argumentos, procura-se o elemento a partir da posição start, até o final da lista.

Com 4 argumentos, procura-se o elemento a partir da posição start, e até antes da posição stop.

Se o elemento procurado não existir na lista (ou fatia) procurada, ocorrerá um erro de valor.

Manipulação de listas: Outras funções

ATENÇÃO

Utilizar a função **index** para determinar se um elemento existe ou não em uma lista não é adequado, pois um erro ocorrerá caso o elemento não exista na lista. Para esta finalidade, recomenda-se o uso do operador **in**, conforme visto anteriormente.

Manipulação de listas: Outras funções

- **Função SORT:** Altera a lista recebida, reordenando os seus elementos.
Utilização: `list.sort(lista)` ou `list.sort(lista, reverse=True)`

Manipulação de listas: Outras funções

- **Função SORT:** Altera a lista recebida, reordenando os seus elementos.
Utilização: **list.sort(lista)** ou **list.sort(lista, reverse=True)**

```
>>> C = ['p', 'y', 't', 'h', 'o', 'n']
>>> list.sort(C)
>>> C
['h', 'n', 'o', 'p', 't', 'y']
>>> list.sort(C, reverse=True)
>>> C
['y', 't', 'p', 'o', 'n', 'h']
```

Ao se passar apenas a lista como argumento, a ordenação é ascendente, do menor para o maior.
Ao se passar a lista e o argumento True para o parâmetro reverse, a lista será ordenada de maneira descendente, do maior para o menor.

Manipulação de listas: Outras funções

- **Função REVERSE:** Altera a lista recebida, invertendo a ordem dos seus elementos.
Utilização: `list.reverse(lista)`

Manipulação de listas: Outras funções

- **Função REVERSE:** Altera a lista recebida, invertendo a ordem dos seus elementos.
Utilização: **list.reverse(lista)**

```
>>> C = ['p', 'y', 't', 'h', 'o', 'n']
>>> list.reverse(C)
>>> C
['n', 'o', 'h', 't', 'y', 'p']
```

A lista recebida é alterada, onde a ordem de seus elementos é invertida.

Manipulação de listas: Outras funções

RECAPITULANDO...

- **Função COUNT:** Retorna quantas vezes um elemento aparece em uma lista.
Utilização: `list.count(lista, elemento)`
- **Função INDEX:** Retorna o índice da primeira ocorrência em que um elemento aparece em uma lista.
Utilização: `list.index(lista, elemento)` ou `list.index(lista, elemento, start, stop)`
- **Função SORT:** Altera a lista recebida, reordenando os seus elementos.
Utilização: `list.sort(lista)` ou `list.sort(lista, reverse=True)`
- **Função REVERSE:** Altera a lista recebida, invertendo a ordem dos seus elementos.
Utilização: `list.reverse(lista)`

Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

Computação I - Python

Aula 6: Fatiamento e Manipulação de Listas

Manipulação de Listas: Outras Funções

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>

