Computação I - Python
Aula 6: Fatiamento e Manipulação de Listas
Atribuição à Fatias, Alias e Cópia de Listas

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

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

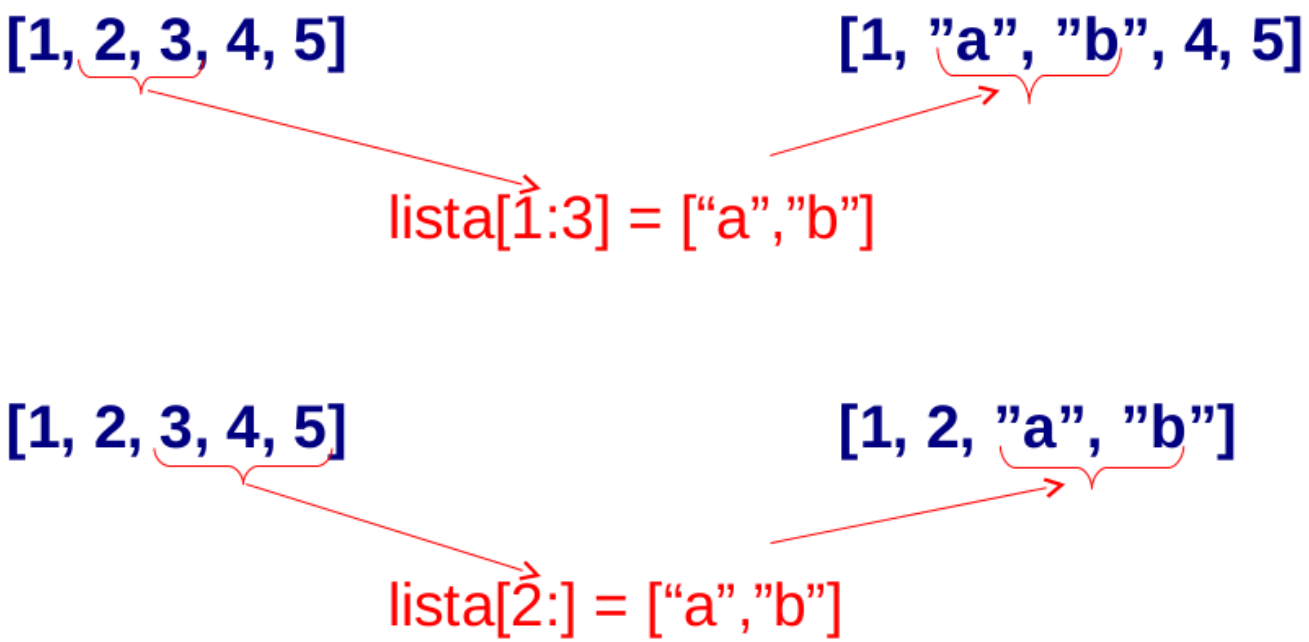
Listas - Atribuição a Fatias

Atribuição: ao atribuir uma sequência a uma fatia, os elementos desta devem ser substituídos pelos elementos daquela.

Listas - Atribuição a Fatias

Atribuição: ao atribuir uma sequência a uma fatia, os elementos desta devem ser substituídos pelos elementos daquela.

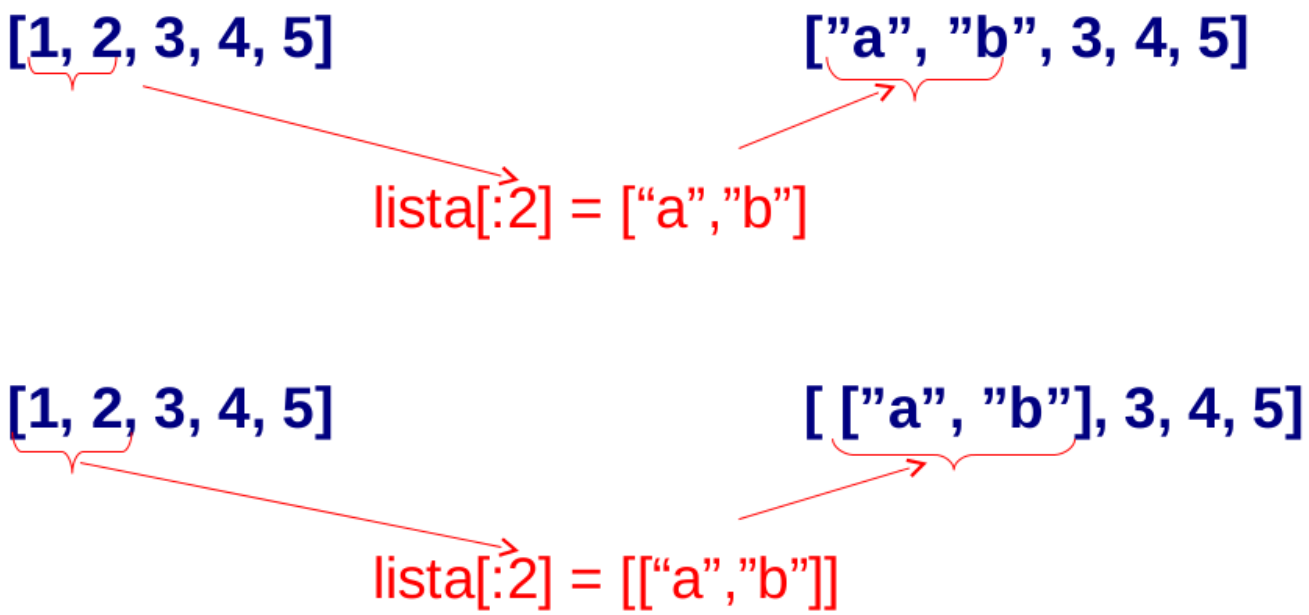
```
>>> lista = [1,2,3,4,5]
```



Listas - Atribuição a Fatias

Atribuição: ao atribuir uma sequência a uma fatia, os elementos desta devem ser substituídos pelos elementos daquela.

```
>>> lista = [1,2,3,4,5]
```

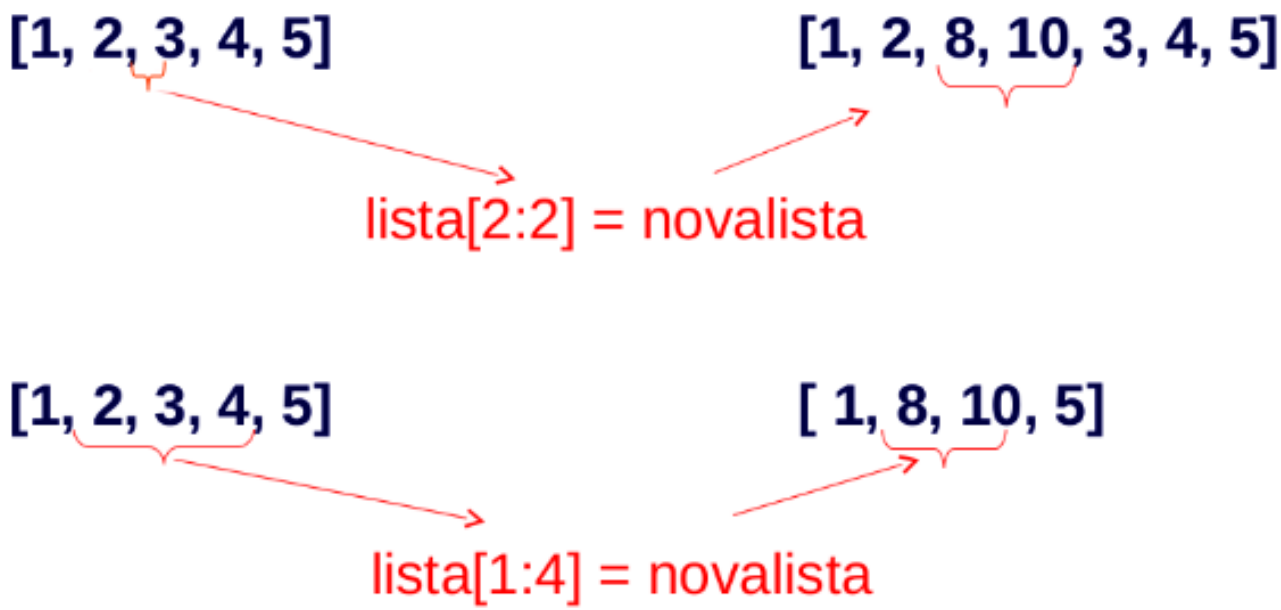


Listas - Atribuição a Fatias

Atribuição: ao atribuir uma sequência a uma fatia, os elementos desta devem ser substituídos pelos elementos daquela.

```
>>> lista = [1,2,3,4,5]

>>> novalista = [8,10]
```



Listas - Atribuição a Fatias

- O que ocorre ao se tentar atribuir algum elemento que não seja uma lista à uma fatia?

Atribuição de um Iterável: O iterável será implicitamente convertido para uma lista, onde cada elemento referenciado por um índice se tornará um elemento dessa lista.

```
>>> lista = [1,2,3,4,5]
>>> lista[0:0] = "ola"
['o', 'l', 'a', 1, 2, 3, 4, 5]
```

Atribuição de um Não-Iterável: Ocorrerá um erro de tipo na atribuição.

```
>>> lista = [1,2,3,4,5]
>>> lista[3:] = 3.14
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    lista[3:] = 3.14
TypeError: can only assign an iterable
```

Listas - Atribuição a Fatias

Atribuição a fatias com passo diferente de 1: Deve-se saber exatamente quantos elementos existem na fatia. A lista a ser atribuída deve conter exatamente a mesma quantidade de elementos da fatia.

Exemplo 1: Atribuindo ["a","b"] à fatia lista[::2], onde lista = [1,2,3,4,5]

```
>>> lista = [1,2,3,4,5]
>>> lista[::2] = ["a","b"]
Traceback (most recent call last):
  File "<pyshell#39>", line 1, in <module>
    lista[::2] = ["a","b"]
ValueError: attempt to assign sequence of size 2 to extended slice
    of size 3
```

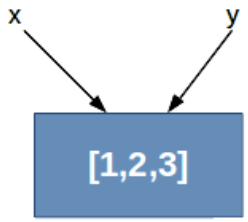
Exemplo 2: Atribuindo ["a","b","c"] à fatia lista[::2], onde lista = [1,2,3,4,5]

```
>>> lista = [1,2,3,4,5]
>>> lista[::2] = ["a","b","c"]
['a', 2, 'b', 4, 'c']
```

Alias e Tipos Mutáveis

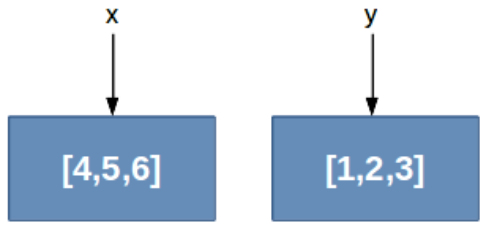
Relembrando - um *alias* acontece quando duas variáves se referem ao mesmo dado:

```
>>> y = [1,2,3]
>>> x = y
>>> x
[1,2,3]
>>> y
[1,2,3]
```



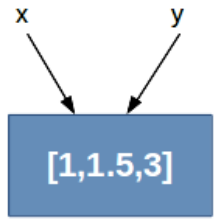
Qualquer nova atribuição feita a uma das variáveis quebrará o *alias* entre elas.

```
>>> x = [4,5,6]
>>> x
[4,5,6]
>>> y
[1, 2, 3]
```



Porém quando o dado é mutável (lista!!), o *alias* não é quebrado caso uma atribuição modifique uma parte do dado:

```
>>> y = [1,2,3]
>>> x = y
>>> x[1]=1.5
>>> x
[1, 1.5, 3]
>>> y
[1, 1.5, 3]
```



Listas - Cópias

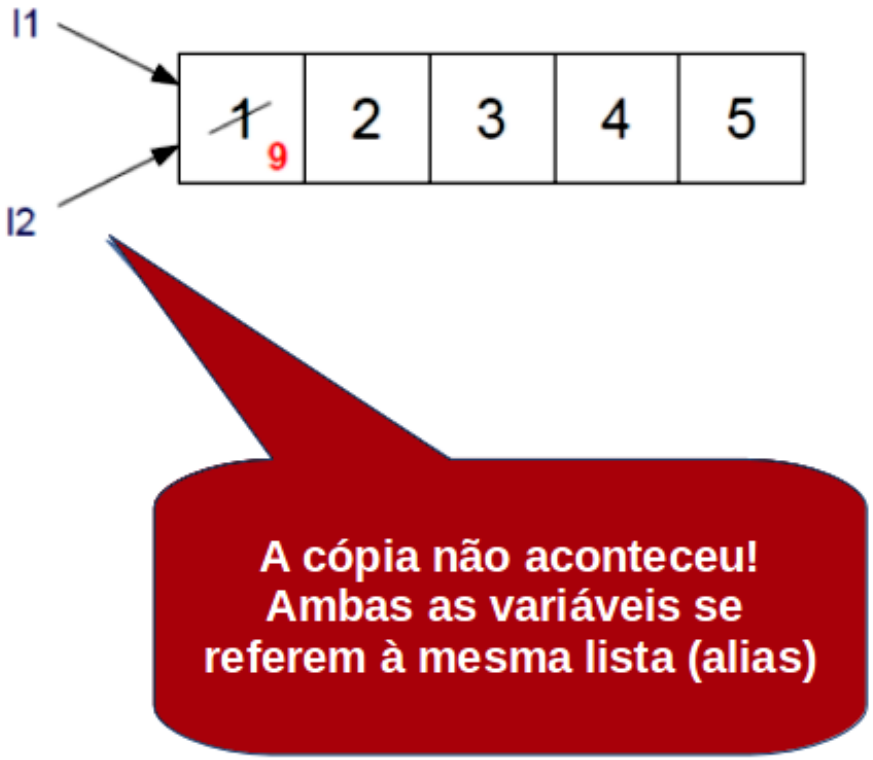
Cuidado quando fizer cópia de listas!

```
>>> l1 = [1,2,3,4,5]
>>> l2 = l1
>>> l1
[1,2,3,4,5]

>>> l2
[1,2,3,4,5]

>>> l2[0]=9
>>> l2
[9,2,3,4,5]

>>> l1
[9,2,3,4,5]
```



Listas - Cópias

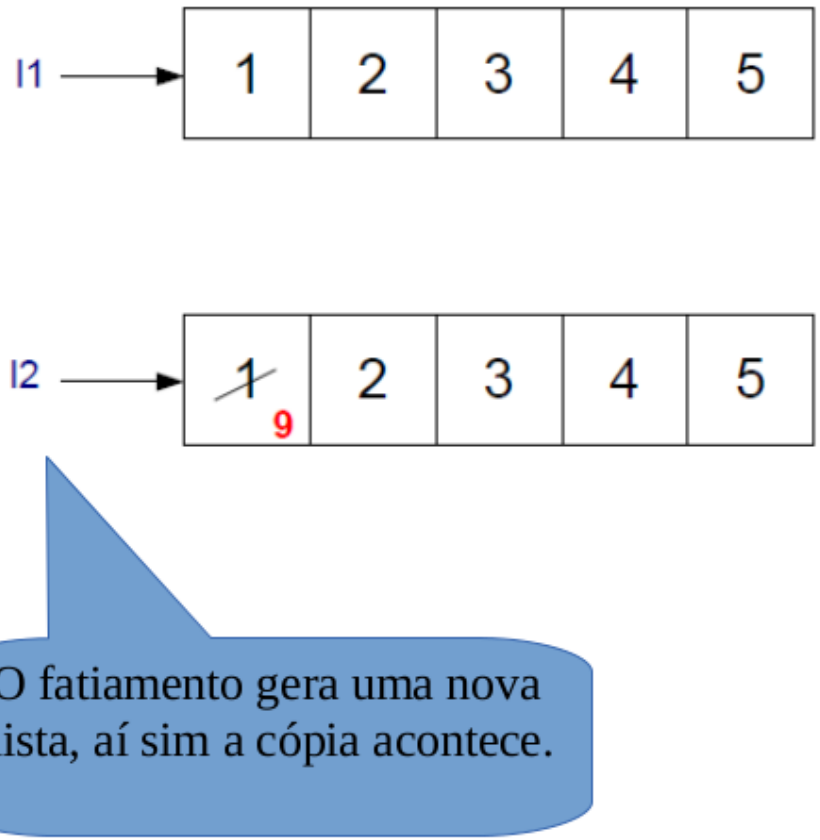
Cuidado quando fizer cópia de listas!

```
>>> l1 = [1,2,3,4,5]
>>> l2 = l1[:]
>>> l1
[1,2,3,4,5]

>>> l2
[1,2,3,4,5]

>>> l2[0]=9
>>> l2
[9,2,3,4,5]

>>> l1
[1,2,3,4,5]
```



Listas - Cópias

Outra forma de se fazer a cópia de uma lista:

- **Função COPY**: Retorna uma cópia da lista recebida.
  Utilização: **list.copy(lista)**

Listas - Cópias

Outra forma de se fazer a cópia de uma lista:

- **Função COPY**: Retorna uma cópia da lista recebida.
  Utilização: **list.copy(lista)**

```
>>> l1 = [1, 2, 3, 4, 5]
>>> l2 = l1[:]
>>> l3 = list.copy(l1)
>>> l1[0] = 'teste'
>>> l1
['teste', 2, 3, 4, 5]
>>> l2
[1, 2, 3, 4, 5]
>>> l3
[1, 2, 3, 4, 5]
```

Gerar uma cópia de uma lista utilizando a função copy
ou a partir de uma fatia inteira da lista gera o mesmo resultado.

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
- Fabrício Firmino de Faria ▸ Lattes

Computação I - Python
Aula 6: Fatiamento e Manipulação de Listas
Atribuição à Fatias, Alias e Cópia de Listas

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

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