

# **ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II**

**Tipos Mutáveis e Não Mutáveis  
em Python**

# **Gerenciamento de memória**

**Até o bimestre passado, aprendemos os conceitos básicos de programação.**

**O funcionamento interno das operações não foi apresentado em maiores detalhes.**

**Neste bimestre, aprofundaremos nosso estudo em programação, começando com o gerenciamento de memória na linguagem Python.**

# Gerenciamento de memória

Quando fazemos uma atribuição:

```
>>> a = 3
```

o objeto int com valor 3 e o nome a são criados.

Python mantém em uma tabela todos os nomes de variáveis criadas pelo programa, os quais 'apontam' para os objetos alocados na memória.

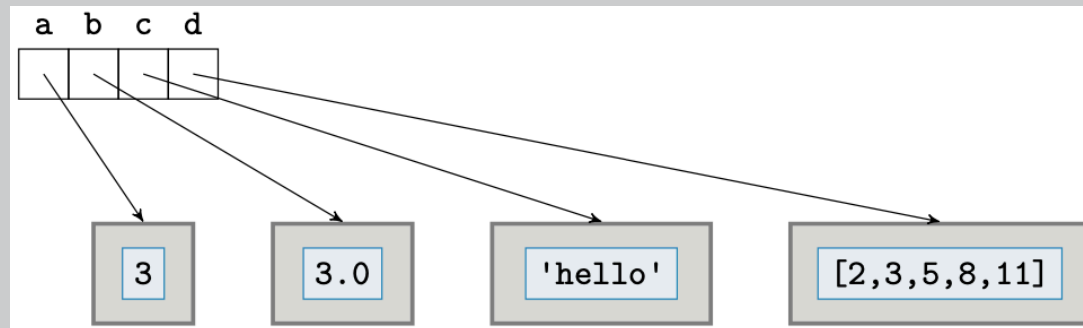
# Gerenciamento de memória

```
>>> a = 3
```

```
>>> b = 3.0
```

```
>>> c = 'hello'
```

```
>>> d = [2, 3, 5, 8, 11]
```

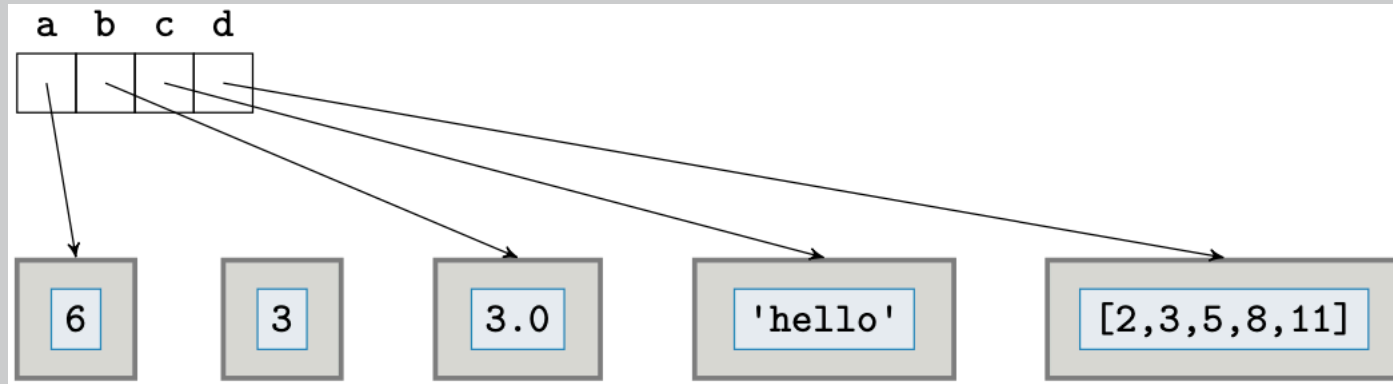


Fonte: Perkovic, 2015

# Gerenciamento de memória

O que acontece quando alteramos a variável a?

```
>>> a = 6
```



Fonte: Perkovic, 2015

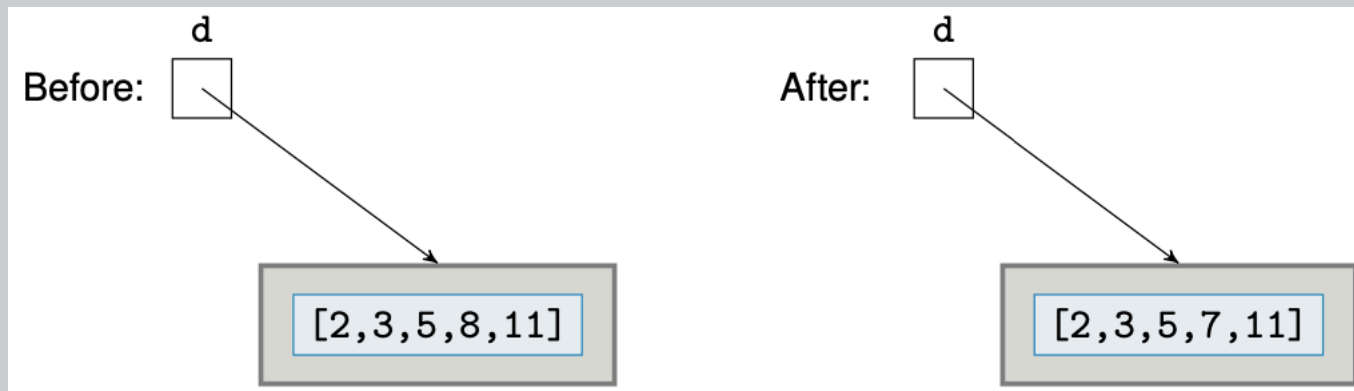
→ int, bool, float, str e complex são imutáveis!

# Gerenciamento de memória

Já no caso de lista, seu valor pode alterar:

```
>>> d = [2, 3, 5, 8, 11]
```

```
>>> d[3] = 7
```



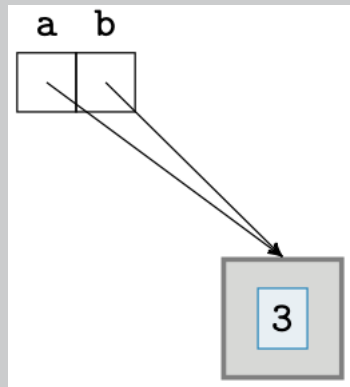
Fonte: Perkovic, 2015

# Gerenciamento de memória

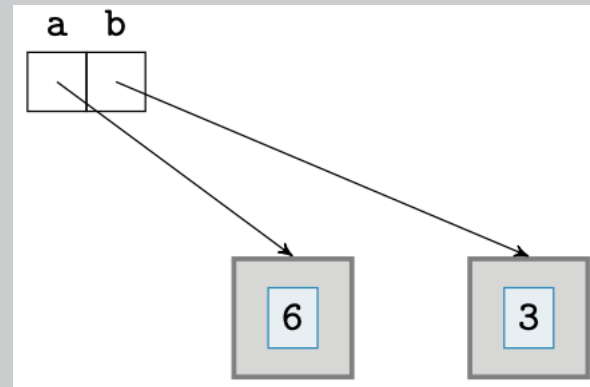
Como a mutabilidade de objetos afeta a atribuição de variáveis?

```
>>> a = 3
```

```
>>> b = a
```



```
>>> a = 6
```



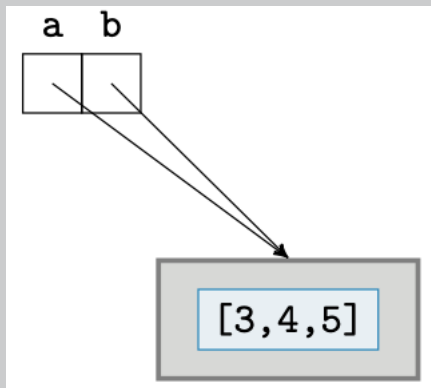
Fonte: Perkovic, 2015

# Gerenciamento de memória

Como a mutabilidade de objetos afeta a atribuição de variáveis?

```
>>> a = [3, 4 5]
```

```
>>> b = a
```



```
>>> b[1] = 8
```

```
>>> b
```

```
[3, 8, 5]
```

```
>>> a
```

```
[3, 8, 5]
```

Fonte: Perkovic, 2015



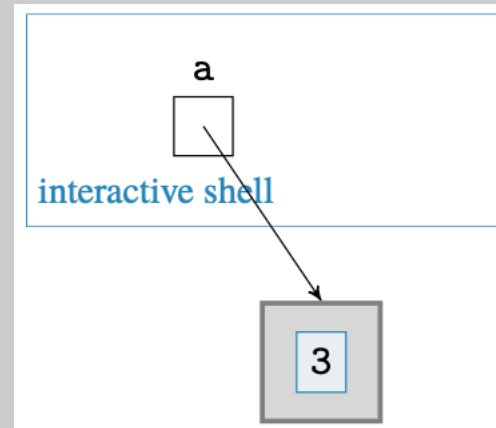
# Gerenciamento de memória

Como a mutabilidade de objetos afeta a passagem de parâmetros para funções?

```
def g(x):
```

```
    x = 5
```

```
>>> a = 3
```



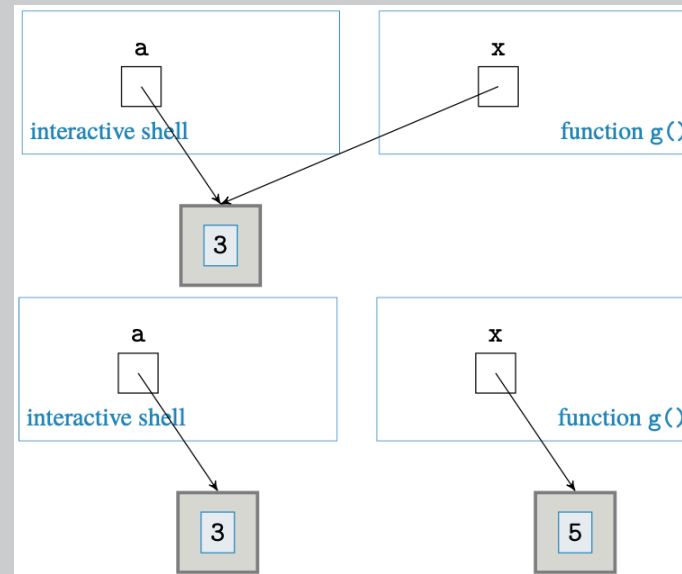
# Gerenciamento de memória

Como a mutabilidade de objetos afeta a passagem de parâmetros para funções?

```
def g(x):  
    x = 5
```

```
>>> a = 3
```

```
>>> g(a)
```



Fonte: Perkovic, 2015

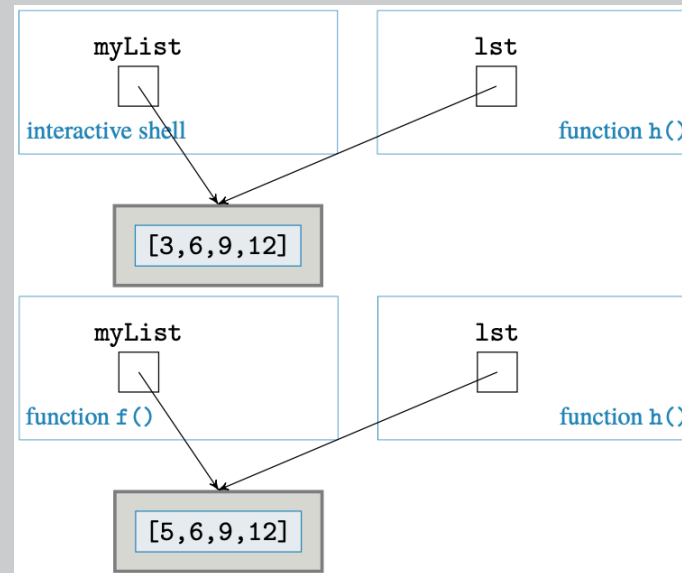
# Gerenciamento de memória

Como a mutabilidade de objetos afeta a passagem de parâmetros para funções?

```
def h(lst):  
    lst[0] = 5
```

```
>>> myList = [3, 6, 9, 12]
```

```
>>> h(myList)
```



Fonte: Perkovic, 2015

# **ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II**

**Tipos Mutáveis e Não Mutáveis  
em Python**