

# Dynamic typing

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

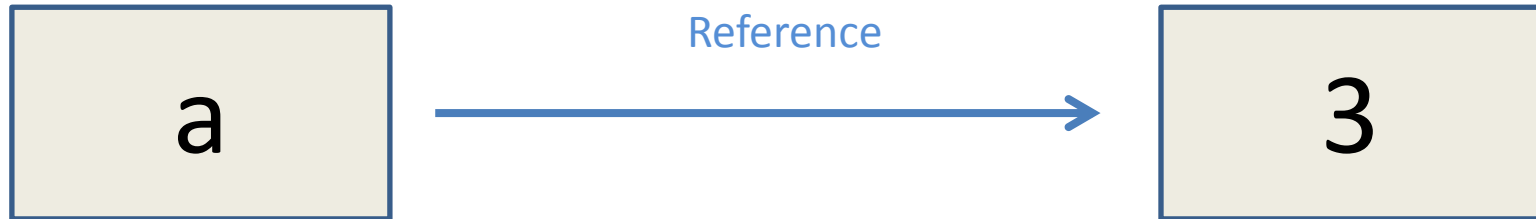
Quando digito **a=3** Python esegue le seguenti operazioni:

1. Crea un oggetto intero che rappresenta il numero 3.
2. Crea la variabile (il nome) **a**, se ancora non esiste.
3. Connette il nome **a** **all'oggetto che rappresenta il numero 3**

# Dynamic typing

Nome (variabile)

Oggetto

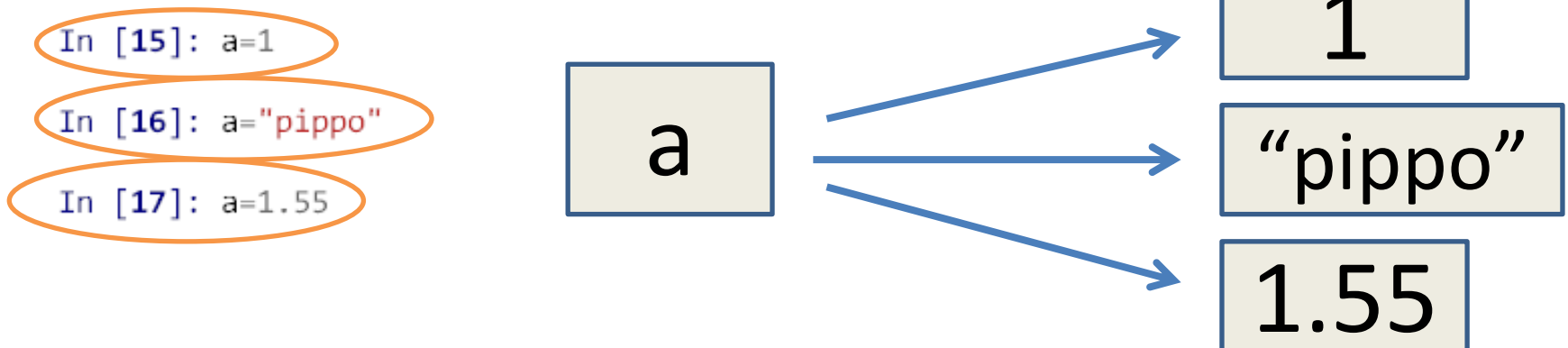


Posso usare lo stesso nome per riferirmi ad oggetti con tipi diversi (non sto cambiando tipo ad un oggetto ma sto cambiando riferimento a una variabile):

Nome (variabile)

Reference

Oggetto

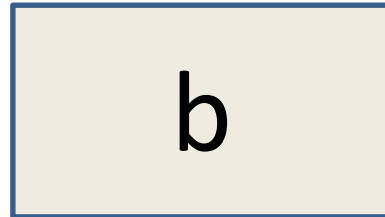
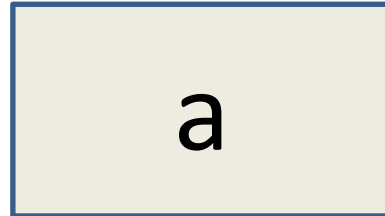


# Shared references (immutable objs)

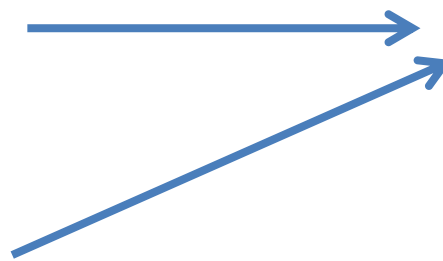
```
In [3]: a=5
```

```
In [4]: b=a
```

Nome (variabile)



Reference



Oggetto



```
In [10]: b=7
```

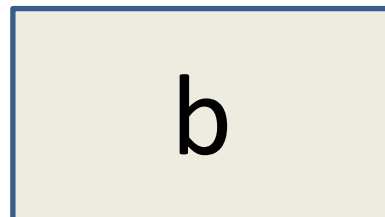
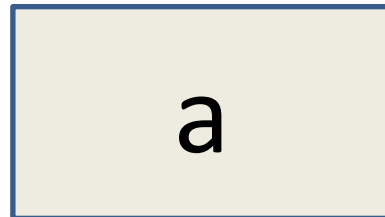
```
In [11]: a
```

```
Out[11]: 5
```

```
In [12]: b
```

```
Out[12]: 7
```

Nome (variabile)



Reference



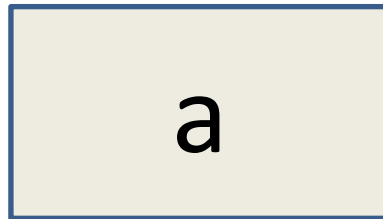
Oggetto



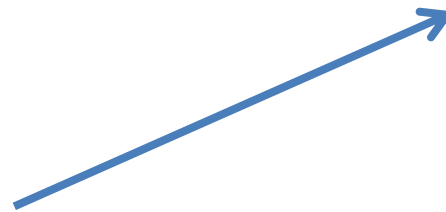
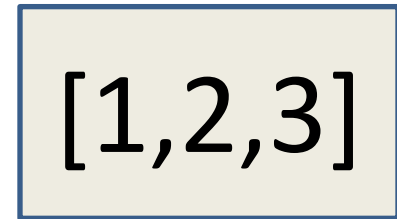
# Shared references (mutable obj)

Nome (variabile)

Oggetto

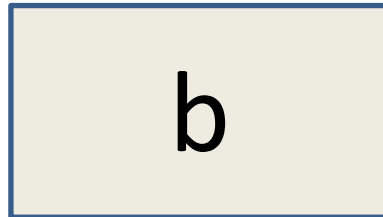


Reference



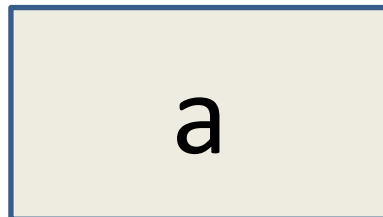
```
In [25]: a=[1,2,3]
```

```
In [26]: b=a
```

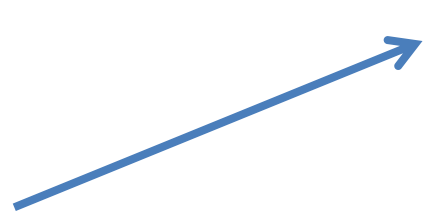
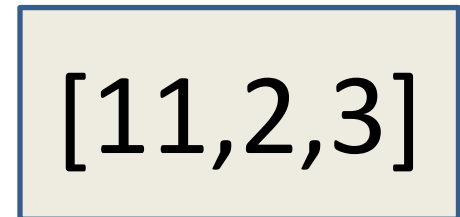


Nome (variabile)

Oggetto



Reference



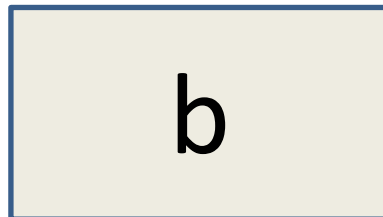
```
In [27]: a[0]=11
```

```
In [28]: a
```

```
Out[28]: [11, 2, 3]
```

```
In [29]: b
```

```
Out[29]: [11, 2, 3]
```



# Shared references (mutable obj)

## copy

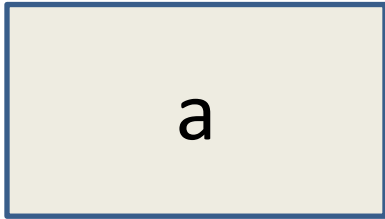
```
In [18]: a=[1,2,3]
```

```
In [19]: b=a[:]
```

```
In [20]: b=copy(a)
```

```
In [21]: b=list(a)
```

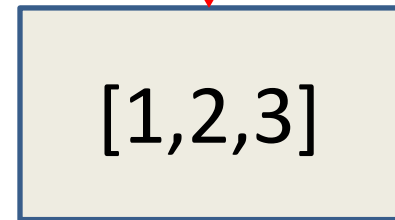
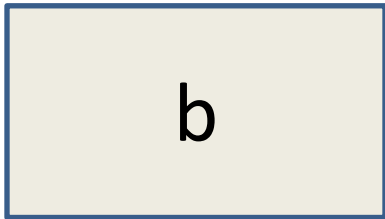
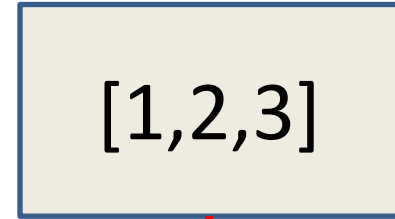
Nome (variabile)



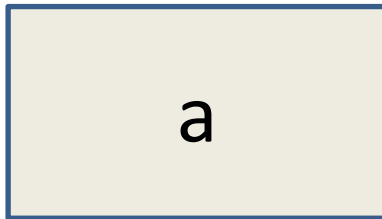
Reference



Oggetto



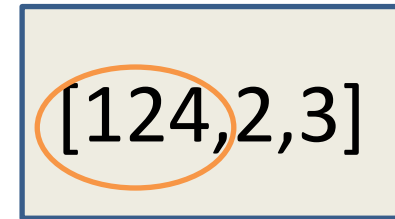
Nome (variabile)



Reference



Oggetto



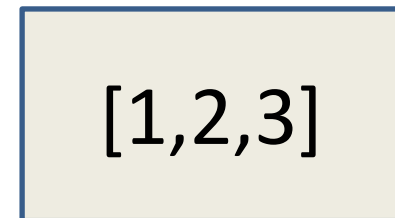
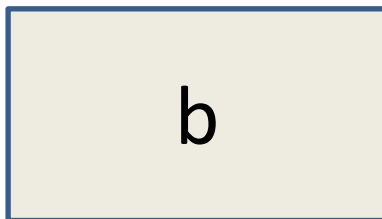
```
In [22]: a[0]=124
```

```
In [23]: a
```

```
Out[23]: [124, 2, 3]
```

```
In [24]: b
```

```
Out[24]: [1, 2, 3]
```



# Shared reference e Uguaglianza

```
In [25]: a=[1,2,3]
```

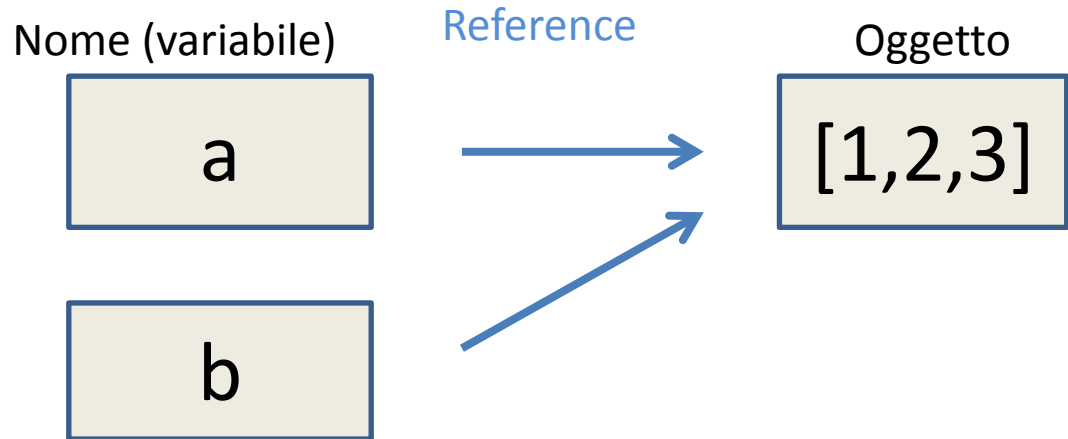
```
In [26]: b=a
```

```
In [27]: a==b
```

```
Out[27]: True
```

```
In [28]: a is b
```

```
Out[28]: True
```



**==** controlla l'uguaglianza di valore

**is** controlla se due references si riferiscono allo stesso oggetto

```
In [29]: b=[1,2,3]
```

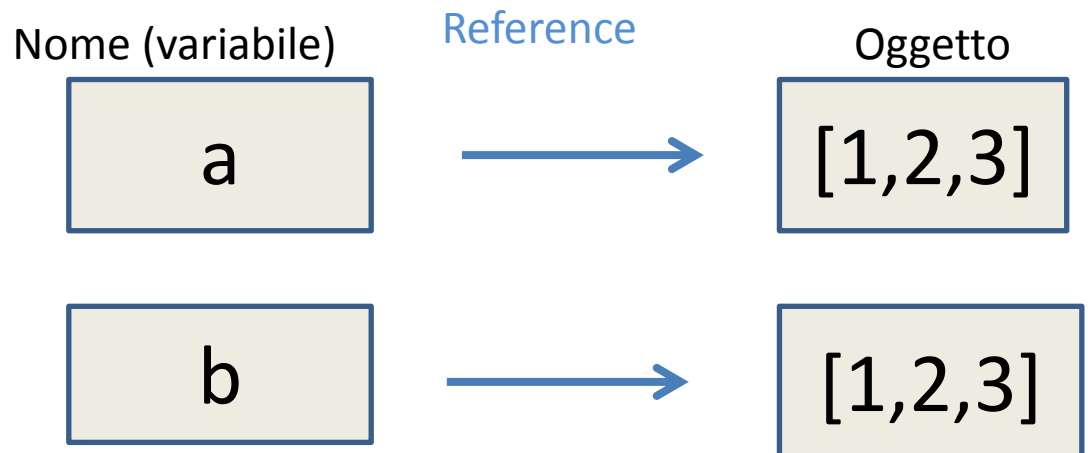
```
In [30]: a=[1,2,3]
```

```
In [31]: a==b
```

```
Out[31]: True
```

```
In [32]: a is b
```

```
Out[32]: False
```



# Note su Dynamic typing

- Le variabili si riferiscono sempre ad oggetti e non ad altre variabili.
- Distinguere tra reference a tipi mutable e non-mutable
- Il tipo è un'informazione attaccata all'oggetto e non alla variabile.