

Programmazione in Python

variabili: approfondimento

Dario Pescini - Mirko Cesarini

Variabili: nome + oggetto

In Python le **variabili** sono il risultato dell'associazione di un **nome** ad un **oggetto**.

- Il **nome** è il riferimento astratto al contenuto della variabile, che permette l'astrazione e semplifica la stesura del programma.
- L'**oggetto** è il contenitore che isola il lato "tecnico" e lo nasconde al programmatore.

Variabili - Oggetti

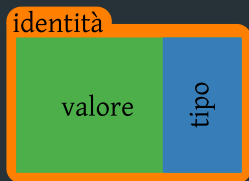
Quando dichiariamo una **variabile**, l'interprete aggiunge una coppia (**identità**, **nome_variabile**) allo **spazio dei nomi**.

spazio dei nomi

identità	nome
140506419992072	lato
140506419992654	area
...	...

Variabili: oggetto

In Python gli **oggetti** associati alle **variabili** sono composti da:



identità identificativo univoco dell'oggetto

tipo descrive come viene immagazzinata l'informazione

valore l'informazione stessa

Oggetto: tipo

Il tipo “di una variabile”:

- É associato alla **codifica** del contenuto nella memoria del calcolatore.
- **ne determina l'interpretazione del contenuto.**
- É associato **in maniera automatica al contenuto** della variabile al momento della sua dichiarazione.

```
>>> type(10)
<type 'int'>
>>> type(-10)
<type 'int'>
>>> type(10.0)
<type 'float'>
>>> type(-10.0)
<type 'float'>
>>> type('abcde18')
<type 'str'>
```

- É possibile **forzare** il tipo associato: `a = int(10.0)`

Variabili - Oggetti

É possibile richiamare

- lo **spazio dei nomi** tramite l'istruzione `dir()`

```
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__', 'lato']
```

- l'**identità** di una variabile tramite l'istruzione `id()`

```
>>> lato = 10.0
>>> id(lato)
140506419992072
```

- il **tipo** di una variabile tramite l'istruzione `type()`

```
>>> type(a)
<type 'float'>
>>> a = int(a)
>>> type(a)
<type 'int'>
>>> _
```

- il **contenuto** di una variabile tramite l'istruzione `print()`

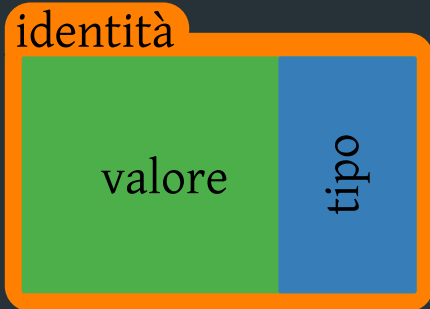
```
>>> print(lato)
10.0
>>> _
```

Variabili oggetto: quella corretta



spazio dei nomi

identità	nome
123456	pippo
987654	pappo
...	...



Dichiarazione variabile - Plasticità

In python una variabile può avere contenuti diversi (anche per tipo) nel tempo:

```
>>> a = 10
>>> type(a)
<type 'int'>
>>> a = 'pippo'
>>> type(a)
<type 'str'>
```


Dichiarazione variabile - Plasticità

In python una variabile può avere contenuti diversi (anche per tipo) nel tempo:

```
>>> a = 10
>>> type(a)
<type 'int'>
>>> a = 'pippo'
>>> type(a)
<type 'str'>
```

Python, ad ogni dichiarazione, crea una nuova coppia (identificativo, nome_variabile) nello spazio dei nomi

Dichiarazione variabile - Plasticità

In python una variabile può avere contenuti diversi (anche per tipo) nel tempo:

```
>>> a = 10
>>> type(a)
<type 'int'>
>>> a = 'pippo'
>>> type(a)
<type 'str'>
```

Python, ad ogni dichiarazione, crea una nuova coppia (identificativo, nome_variabile) nello spazio dei nomi

```
>>> a = 100
>>> id(a)
140605457971008
>>> a = 20
>>> id(a)
140605457968976
>>>
```

Variabili: attenzione nella copia

```
>>> a = 10
>>> b = a
>>> print(b)
10
>>> a = 100
>>> print(a)
100
>>> print(b)
10
>>> _
```

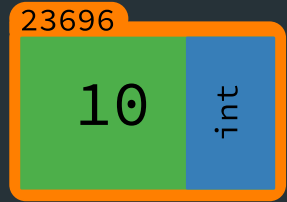
Variabili: attenzione nella copia

```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```

- `b = a` è da interpretarsi come “il nome `b` è associato allo stesso oggetto del nome `a`”
- ma `a = 100` associa un nuovo oggetto al nome `a`
- alla fine le due variabili hanno contenuto diverso

Variabili: attenzione nella copia

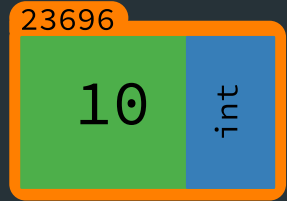
```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```



Variabili: attenzione nella copia

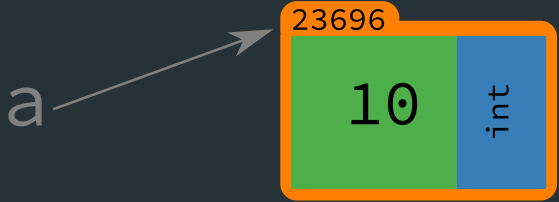
```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```

a



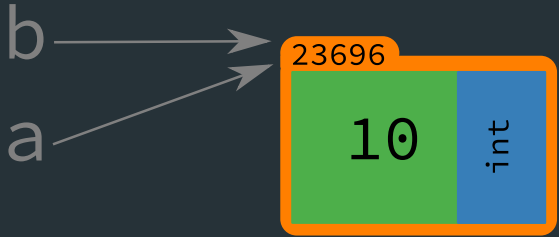
Variabili: attenzione nella copia

```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```



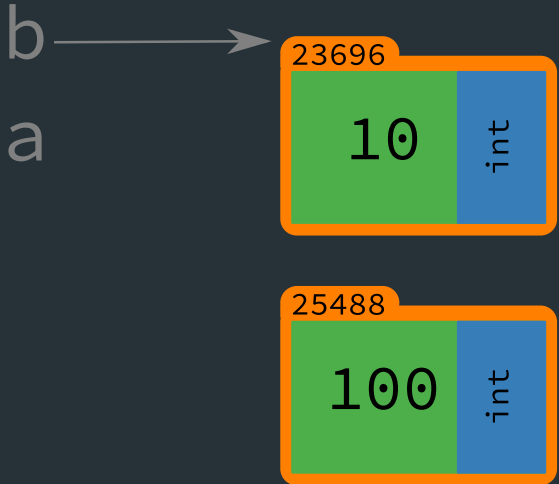
Variabili: attenzione nella copia

```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```



Variabili: attenzione nella copia

```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```



Variabili: attenzione nella copia

```
>>> a = 10
>>> id(a)
140506421023696
>>> b = a
>>> id(b)
140506421023696
>>> a = 100
>>> id(a)
140506421025488
>>> id(b)
140506421023696
>>> print a
100
>>> print b
10
>>> _
```

