

# Programmazione in Python: dal prompt al file sorgente

Dario Pescini - Mirko Cesarini

Università degli Studi di Milano-Bicocca

Dipartimento di Statistica e Metodi Quantitativi

# Come si programma in Python

Esistono tre forme di interazione con l'interprete:

- **da linea di comando** utile per testare porzioni di codice o come calcolatrice avanzata
- **con codice sorgente** utile per programmi strutturati
- **da notebook** utile per analisi esplorative dei dati o prototipizzazione

# Come si programma in Python

- da linea di comando
  - 1 avvio dell'interprete dei comandi `python`
  - 2 introduzione dell'istruzione
  - 3 output
  - 4 introduzione dell'istruzione
  - 5 output
  - 6 ...

# Come si programma in Python

- con codice sorgente

- ① scrittura del file sorgente

Creazione, con un editor di testo, del file `sorgente.py` che contiene le istruzioni nel linguaggio di programmazione Python.

- ② esecuzione del programma

Il calcolatore interpreta ed esegue la sequenza di istruzioni codificate all'interno del programma.

`python sorgente.py`

## da linea di comando: python

Prompt >>> dove inserire il comando

```
darío@vulcano: python
Python 2.7.10 (default, May 25 2015, 13:06:17)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.56)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> latoA=3
>>> latoB=5
>>> area=latoA*latoB
>>> print area
15
>>> _
```

per uscire dal prompt digitare: `exit()` oppure `ctrl + D`

# da linea di comando: ipython

Prompt `In[ ]`: dove inserire il comando

```
dario@vulcano: ipython
Python 2.7.10 (default, Oct 5 2015, 09:41:46)
Type "copyright", "credits" or "license" for more information.

IPython 4.0.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: latoA = 3

In [2]: latoB = 5

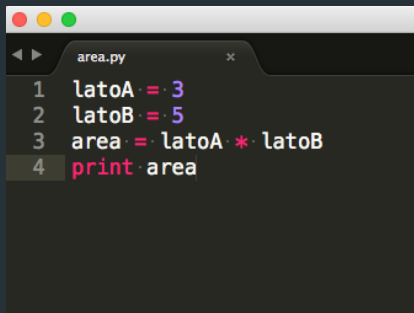
In [3]: area = latoA * latoB

In [4]: print area
15

In [5]: _
```

per uscire dal prompt digitare: `exit()` oppure `ctrl + D`

## file sorgente



```
1 latoA = 3
2 latoB = 5
3 area = latoA * latoB
4 print area
```

```
dario@vulcano: python area.py
```

```
15
```

```
dario@vulcano: _
```

# notebook

localhost:8888/notebooks/Untitled.ipynb?kernel\_name=python2

## jupyter Untitled

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: latoA = 5
```

```
In [2]: latoB = 7
```

```
In [3]: area = latoA * latoB
```

```
In [4]: print area
```

35

```
In [ ]:
```



# Struttura di un file Sorgente

Un file sorgente si suddivide in:

- importazione di librerie
- funzioni
- sequenza di istruzioni

# Elementi di un file sorgente

## ① commenti: `# commento`

servono a rendere più leggibile il codice per un suo riutilizzo.

## ② istruzioni Python: `istruzione`

```
istruzione1  
istruzione2
```

...

sono i “comandi” veri e propri; possono essere separate in gruppi logici per mezzo di un linea vuota.

## ③ indentazioni: `....4 spazi o multipli`

```
istruzione1  
....istruzione2  
....istruzione3  
istruzione4
```

servono a creare blocchi di istruzioni per il controllo del flusso dell'informazione.

# Elementi di un file sorgente

```
percorso.py
1 import math
2
3
4 # questo programma calcola la
5 # lunghezza del percorso che unisce
6 # più città.
7 # Viene terminato tramite l'inserimento di
8 # una coordinata negativa.
9
10 haltFlag = 1 # inizializzo flag
11 lunghezza = 0 #inizializzo a 0 per successiva sommatoria
12
13 print "inserire le coordinate delle città:"
14 print "un valore negativo termina l'inserimento"
15 xOld = input("ascissa ")
16 yOld = input("ordinata ")
17 while (haltFlag > 0):
18     x = input("ascissa ")
19     y = input("ordinata ")
20     if(x < 0 or y < 0):
21         haltFlag = 0
22     else :
23         deltaX = x - xOld
24         deltaY = y - yOld
25         lunghezza += (deltaX**2 + deltaY**2)
26 lunghezza = math.sqrt(lunghezza)
27 print "la lunghezza del percorso e' ", lunghezza
28
```

importazione modulo

commento su più linee

indentazione

4 spazi

blocco di istruzioni

# Attenzione!

- *Case Sensitive* `pippo`, `Pippo`, `Pipp0`, `PiPp0` sono diversi in Python.
- *Parole riservate* Esiste una insieme di identificatori propri del linguaggio che non possono essere ridefiniti:

|         |          |        |        |
|---------|----------|--------|--------|
| and     | as       | assert | break  |
| class   | continue | def    | del    |
| elif    | else     | except | exec   |
| finally | for      | from   | global |
| if      | import   | in     | is     |
| lambda  | not      | or     | pass   |
| print   | raise    | return | try    |
| while   | with     | yield  |        |

## Primo esempio di istruzioni

- `print(variable)` stampa a video una variabile

scrittura

```
print("HelloWorld!")
```

- `input('Stringa')` stampa la stringa a video e legge da tastiera quanto viene digitato

lettura di una variabile a

```
a = input("inserire il contenuto della  
variabile a ")
```

# Esempio I/O

# Dal prompt al sorgente