
Francesca Mazzia
Dipartimento di Matematica
Università di Bari

Python: lezione introduttiva

Python è un particolare linguaggio di programmazione di tipo interpretativo che consente di sviluppare le proprie applicazioni. È nato come linguaggio di programmazione per sviluppare applicazioni distribuite, scripting, e per il calcolo scientifico. Python può essere paragonato a Tcl, Perl, Java, JavaScript e Visual Basic. È reperibile gratuitamente dal sito: <http://www.python.org/>

Python può essere associato con una interfaccia grafica. Per windows suggerisco **python(x,y)** che può essere scaricato da <http://www.pythonxy.com/>. Per linux suggerisco spider che può essere scaricato da <http://spyderlib.googlecode.com/>. Python va poi integrato con tutti i pacchetti numerici che utilizzeremo, che vanno installati a parte (con python(x,y) molti pacchetti vengono installati automaticamente))

Lavorando usando la console interattiva di spider, al prompt **■>** ogni istruzione viene interpretata ed eseguita immediatamente.

```
>>>
```

il che significa che il calcolatore è pronto a ricevere le istruzioni ed ad eseguirle. Se vogliamo eseguire la somma 3+2 basta scrivere:

```
>>> 3 + 2
```

e premere il tasto invio. Otteniamo immediatamente la risposta. e il calcolatore è di nuovo pronto a ricevere un nuovo comando. Le variabili nei linguaggi di programmazione sono dei contenitori di valori. Se vogliamo assegnare il risultato dell'operazione alla variabile di nome var basta scrivere:

```
>>> var = 3 + 2 #questo e' un commento
```

Questa istruzione si chiama *istruzione di assegnazione*. Per visualizzare il risultato dell'istruzione eseguita si digita il nome della variabile

```
>>> var
5
```

Possiamo assegnare a una variabile il valore di una espressione più complicata. Per far questo dobbiamo sapere a quali simboli corrispondono le principali operazioni aritmetico-logiche. In R questi sono:

+	addizione
-	sottrazione
*	moltiplicazione
/	divisione
**	elevamento a potenza
&	and logico
	or logico
!	not logico
==	uguale
!=	diverso
<	minore
<=	minore uguale
>	maggiore
>=	maggiore uguale

Le espressioni aritmetiche e logiche seguono le classiche regole di precedenza, che è possibile modificare utilizzando le parentesi tonde.

Per avere informazioni su tutte le operazioni che possono essere eseguite in R si può digitare sul pulsante `?`. Compare la finestra di con l'elenco di tutte le informazioni su Python divise per capitoli.

Possiamo ora eseguire:

```
>>> var=5.0*3.0/(3.0+2.0)**2.0
>>> var
0.59999999999999998
>>>
```

```
>>> import numpy as np
>>> import scipy as sp
```

È anche possibile calcolare i valori delle principali funzioni. Ad esempio sono predefinite le funzioni in una variabile:

```
np.abs(x)  valore assoluto di x,
np.log(x)  logaritmo naturale,
np.exp(x)  esponenziale con base e,
np.sin(x)  seno,
np.cos(x)  coseno,
np.tan(x)  tangente.
```

È possibile quindi eseguire espressioni del tipo:

```
>>> np.cos((5 * 3/(3 + np.cos(5)))*np.log(2) * 100)
-0.74819904070879473
>>>
```

Oltre alle variabili, sono di particolare utilità anche le *costanti*. Le costanti sono dei contenitori di valori che non cambiano nel corso dell'esecuzione di un algoritmo. In R vi sono delle costanti predefinite, come, ad esempio, la costante `pi` che contiene il valore di π :

```
>>> pi
3.1415926535897931
>>>
```

```
[1] 3.142
```

Le variabili possono essere scalari oppure vettori o matrici. I vettori e le matrici sono variabili che contengono un certo numero di componenti. Ad esempio dopo l'istruzione:

```
>>> A = np.array([3., 5., 6.])
>>> A
array([ 3.,  5.,  6.])
>>>

[1] 3 5 6
```

la variabile A sarà un vettore di tre elementi con $A[0]=3$, $A[1]=5$, $A[2]=6$. Gli elementi di un vettore si indirizzano specificando l'indice tra parentesi quadre. Se vogliamo sapere la somma degli elementi di A possiamo eseguire:

```
> A[1] + A[2] + A[3]
```

```
[1] 14
```

oppure

```
> np.sum(A)
```

```
[1] 14
```

Per chiudere la sessione di lavoro digitiamo `quit()`.