

Calcolo numerico in Python

Mattia Penati, Marco Restelli

13 febbraio 2014

Indice

Indice	i
I Elementi del linguaggio Python	1
1 Ambiente di lavoro	3
1.1 Il linguaggio Python	3
1.2 L'esecuzione di un programma Python	4
2 Esecuzione di programmi .py	5
II Esercizi risolti	7
Bibliografia	9
Indice analitico	11

Parte I

Elementi del linguaggio Python

Capitolo 1

Ambiente di lavoro

Con il termine Python si fa spesso riferimento a due oggetti che in realtà sono tra loro distinti:

- il linguaggio di programmazione Python
- il software utilizzato per eseguire un programma scritto in linguaggio Python.

L'ambiente di lavoro cui si fa riferimento in queste note, oltre a Python, include i seguenti pacchetti specifici per le applicazioni di calcolo numerico:

- NumPy: una libreria Python con gli strumenti base del calcolo scientifico, quali vettori, matrici, funzioni di algebra lineare, trasformate di Fourier, analisi statistiche ecc. La libreria è disponibile all'indirizzo <http://numpy.scipy.org/>.
- SciPy: una libreria Python che estende NumPy fornendo funzioni di più alto livello, quali matrici sparse, interpolazione, risoluzione di equazioni alle derivate ordinarie ecc. La libreria è disponibile all'indirizzo <http://www.scipy.org/>.
- IPython: una shell interattiva con molti strumenti utili, tra cui l'evidenziamento della sintassi. È disponibile all'indirizzo <http://ipython.org/>.

1.1 Il linguaggio Python

Citando dal sito www.python.it: “Python è un linguaggio di programmazione ad alto livello, rilasciato pubblicamente per la prima volta nel 1991 dal suo creatore Guido van Rossum, programmatore olandese attualmente operativo in Google. [...] Attualmente, lo sviluppo di Python (grazie e soprattutto all'enorme e dinamica comunità internazionale di sviluppatori) viene gestito dall'organizzazione no-profit Python Software Foundation [PSF].”

Sono attualmente in uso due versioni del linguaggio: 2.7 e 3.3. Il fatto che le versioni 2.x siano ancora utilizzate è dovuto alla mancanza di compatibilità all'indietro delle

versioni 3.x, che risultano quindi incompatibili con un gran numero di programmi attualmente in uso.

La documentazione ufficiale è disponibile in rete all'indirizzo <http://docs.python.org/>.

1.2 L'esecuzione di un programma Python

Semplificando un po' il problema, Python è un linguaggio interpretato, il che significa che un programma Python viene passato ad un interprete Python che lo esegue. L'interprete più utilizzato è CPython ed è scritto in C.

Usando CPython, un programma Python, contenuto in un file con estensione *.py* viene dapprima compilato in *byte code*, ottenendo un file con estensione *.pyc*, e successivamente eseguito. Il *byte code* è una rappresentazione di più basso livello rispetto al sorgente Python originale ma di più alto livello rispetto al codice macchina, ed è indipendente dall'hardware considerato.

Una modalità di esecuzione alternativa è quella tramite interprete di comandi in linea (ad esempio, usando IPython), nella quale ciascuna istruzione Python viene immediatamente interpretata ed eseguita dopo essere stata immessa (per maggiori dettagli si rimanda al Capitolo 2).

Capitolo 2

Esecuzione di programmi *.py*

Indichiamo con il termine “programma” l’insieme delle istruzioni Python contenute in un file con estensione *.py*. Per eseguire tale programma, vi sono varie possibilità, anche in funzione dell’ambiente di lavoro adottato. Consideriamo qui le alternative seguenti:

- esecuzione tramite l’interprete Python,
- esecuzione da un interprete di comandi in linea.

L’esecuzione tramite interprete Python avviene tipicamente invocando tale interprete da linea di comando, ad esempio con

```
$ python file.py
```

oppure, in ambiente Windows, “cliccando” sull’icona del file *file.py*.

Nel caso si volesse specificare la versione dell’interprete, tipicamente si userà

```
$ /usr/bin/python2.7 file.py
```

ovvero

```
$ /usr/bin/python3.2 file.py
```

L’interprete di comandi in linea viene invece attivato con

```
$ python
```

oppure, volendo usare IPython,

```
$ ipython
```

Una volta attivato tale interprete, possiamo “eseguire” il file *.py* con le istruzioni `import` e `reload`¹. Per motivi di efficienza, `import` esegue il file *.py* solo la prima

¹Propriamente parliamo di *parola chiave* per `import` e *funzione* per `reload`. In queste note, il termine *istruzione* è utilizzato come termine generico.

volta che viene invocata. Se il file viene successivamente modificato e si desidera eseguirlo nuovamente è necessario utilizzare `reload`. D'altra parte, non è possibile utilizzare `reload` se prima non si è importato il file almeno una volta.

Come primo esempio, possiamo provare le varie modalità di esecuzione utilizzando il seguente file *esempio1.py*

```
print("Un primo esempio di programma Python")
x = 10
y = 3
z = x*y
print("z = x+y = {0}".format(z))
```

Nei capitoli successivi si discutono i vari aspetti del linguaggio Python, che sono indipendenti dal fatto che le corrispondenti istruzioni vengano raccolte in un file *.py* o inserite utilizzando la linea di comando dell'interprete Python.

Parte II

Esercizi risolti

Bibliografia

- [Fro07] M. Frontini. *Fondamenti di calcolo numerico*. Maggioli Editore, 2007. ISBN: 978-88387-4176-X.
- [Lut09] M. Lutz. *Learning python*. O'Reilly, 2009. ISBN: 978-0-596-15806-4.
- [PSF] *Python Software Foundation*. URL: <http://www.python.org/psf/>.
- [QSS08] A. Quarteroni, R. Sacco e F. Saleri. *Matematica numerica*. Springer, 2008. ISBN: 978-88-470-0782-6.

Indice analitico

byte code, 4

import, 5

reload, 5