

INDICE

- [Presentazione del corso](#)
 - [Introduttivo perché:](#)
 - [Mirato all'ambito scientifico perché:](#)
- [Cosa è Python](#)
 - [In che senso free?](#)
 - [Linguaggi interpretati e compilati](#)
 - [multiplatforma](#)
- [Perché uso Python](#)
- [Perché ho scelto di usare Python3 per il corso](#)
- [Installazione dei pacchetti necessari allo svolgimento del corso \(windows\)](#)
 - [Installazione dell'interprete \(versione 3.2.3\)](#)
 - [Numpy \(librerie di base per calcolo numerico\)](#)
 - [matplotlib \(librerie per la grafica 2D\)](#)
- [Installazione da sorgente](#)

Presentazione del corso

Si tratta di un corso:

- Introduttivo
- Mirato all'ambito scientifico

Introduttivo perché:

- porremo grande attenzione ai tipi ed ai costrutti base del linguaggio che occuperanno la maggior parte delle lezioni
- si cercherà di non dare per scontato nulla

Mirato all'ambito scientifico perché:

- dedicheremo particolare attenzione alla rappresentazione dei tipi numerici
- illustreremo il funzionamento di alcune librerie di base di questo ambito (numpy e matplotlib)
- trascureremo alcuni aspetti che, pur fondamentali per un generico linguaggio di programmazione, non sono strettamente necessari per la programmazione in ambito scientifico.

Cosa è Python

da www.python.org

Python è un linguaggio di programmazione dinamico orientato agli oggetti utilizzabile per molti tipi di sviluppo software. Offre un forte supporto all'integrazione con altri linguaggi e programmi, è fornito di una estesa libreria standard e può essere imparato in pochi giorni.

Molti programmatori Python possono confermare un sostanziale aumento di produttività e ritengono che il linguaggio incoraggi allo sviluppo di codice di qualità e manutenibilità superiori.

Python gira su Windows, Linux/Unix, Mac OS X, OS/2, Amiga, palmari Palm e cellulari Nokia; è stato anche portato sulle macchine virtuali Java e .NET. Python è distribuito con licenza Open-Source approvata dalla OSI: il suo utilizzo è gratuito e libero

anche per prodotti commerciali.

In sintesi:

- E' un vero linguaggio di programmazione general-purpose
- E' semplice da imparare ed usare.
- E' multiplatforma
- E' free!!!

Definiamo meglio cosa significano gli ultimi due punti.

In che senso free?

Python è distribuito con una licenza che è ['GPL compatible'](#)

Le licenze di Python, a differenza della GPL, consentono di distribuire una versione modificata del software senza rendere open source le modifiche.

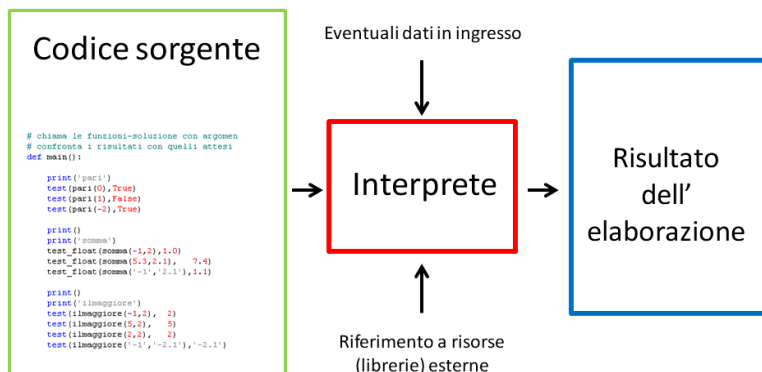
Linguaggi interpretati e compilati

Python è un linguaggio di programmazione interpretato.

Ciò vuol dire che con il termine 'Python' si indicano sostanzialmente due cose distinte:

- I. Il linguaggio inteso come insieme di regole formali che comprendono lessico, sintassi e semantica ('linguaggio' o 'Python' nel seguito. In questo corso parleremo soprattutto di questo).
- II. Un programma ('l'interprete' nel seguito) che interpreta il 'codice sorgente' scritto nel linguaggio di cui sopra.

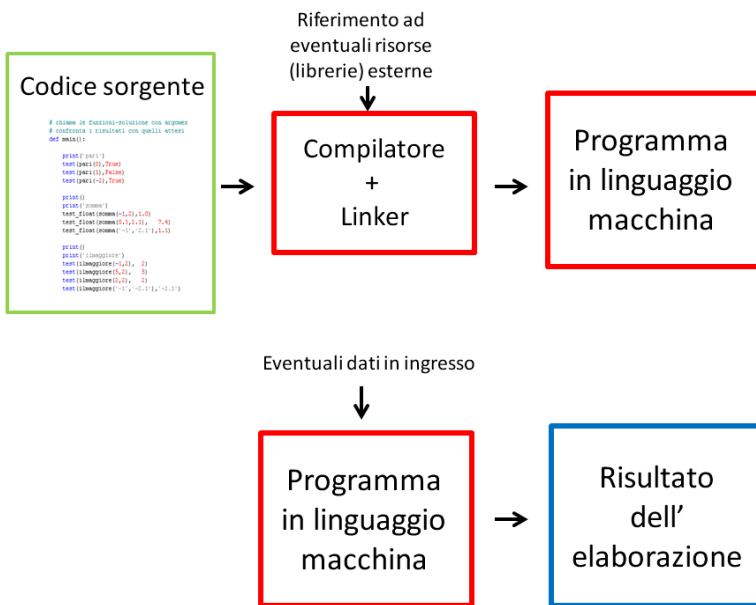
Lo schema di principio è quindi il seguente:



L'interprete, che come si è già detto è un programma, dato un certo set di istruzioni in linguaggio python dovrà:

- Interpretare quali operazioni è necessario effettuare e su quali dati in base alla sintassi del linguaggio.
- Eseguire le operazioni di cui sopra, utilizzando sia le funzionalità integrate nell'interprete, sia eventuali risorse esterne che in genere sono implementate (CPython) mediante moduli 'compilati', cioè da parti del programma che sono state tradotte una volta per tutte in linguaggio macchina.

Nel caso si utilizzi un linguaggio compilato ci si rifà ad uno schema di principio come quello sotto:



multiplatforma

Dai precedenti schemi è facile comprendere che una volta che l'interprete e le eventuali librerie aggiuntive sono disponibili per una piattaforma (e python lo è per tutte le principali), se un codice sorgente non fa riferimento a risorse specifiche ad una particolare architettura questo girerà su tutte le piattaforme supportate dal linguaggio.

Utilizzando un linguaggio come il C invece il codice sorgente di ogni programma che vogliamo utilizzare deve essere compilato sulla piattaforma che ci interessa.

Per chi vuole approfondire la questione sulla distinzione tra linguaggi compilati e interpretati può partire da [qui](#).

Perché uso Python

- perché non costa nulla!!
- perché si trovano liberamente tantissime risorse, sia in termini di documentazione sia di moduli aggiuntivi
- perché lo sviluppo di prototipi di applicazioni risulta molto più rapido che con gli altri linguaggi che ho usato
- perché è un vero linguaggio di programmazione con caratteristiche anche molto avanzate
- perché tali caratteristiche avanzate sono trasparenti per chi vuole usare il linguaggio in modo semplice
- perché ha una sintassi pulita e i programmi risultano in genere molto leggibili
- perché in genere gli stessi script girano sia su linux sia su windows
- perché c'è una comunità molto viva ed è un linguaggio molto utilizzato (forum, mailing-list, newsletter)
- è uno strumento modulare per cui è possibile ridurre al minimo l'installazione (*)
- perché è facilmente integrabile con programmi scritti in C/C++ (fortran)

di contro ho sperimentato anche alcuni limiti:

- non sempre è facile orientarsi nella documentazione disponibile
- (*) la modularità si paga con il fatto che l'installazione base non comprende un ambiente di sviluppo potente come quello di matlab o di altri ambienti integrati.

Perché ho scelto di usare Python3 per il corso

Python 2.0 è stato reso disponibile nel 2000, ed ha avuto un grande successo. Attualmente siamo alla versione 2.7 e non si

prevedono nuovi sviluppi ma solo eventuali bug-fixes. Si prevede comunque che il linguaggio sarà mantenuto per qualche altro anno.

[Python 3](#) è stato rilasciato nel 2008 ed è stato introdotto per vari motivi:

- per correggere alcuni errori di progettazione (divisione int, print statement ...)
- per stare al passo con i tempi e le risorse disponibili (str/unicode, long/int)
- per introdurre nuovi paradigmi di programmazione
- “Open source needs to move or die” (creatore di Ruby)

Python 3 ed è un linguaggio oramai maturo, tuttavia non tutti i moduli che hanno decretato il successo di Python sono stati tradotti da Python 2 a Python 3 (il web-framework [django](#) è un esempio).

Nonostante in ambito scientifico, Python 2 (2.7) sia ancora lo standard ho deciso di usare Python 3 perché:

- i moduli necessari ad un corso introduttivo sono già stati tradotti
- non volevo creare un corso già vecchio in partenza
- era anche per me un'occasione per passare a Python 3 ;-)

Installazione dei pacchetti necessari allo svolgimento del corso (windows)

La seguente procedura si riferisce ad una installazione su piattaforma windows.

Infatti il corso è stato sviluppato e gli esempi testati su piattaforma windows a 32 bit.

Anche se gli esempi non sono stati testati su piattaforma a 64 bit le risorse segnalate mettono a disposizione i pacchetti per win64 (vedi qui per valutare [i vantaggi](#)).

In particolare ci si limita ai 3 pacchetti strettamente necessari allo svolgimento del corso che sono

- l'interprete (versione 3.2.3)
- le librerie per il calcolo numerico(numpy)
- le librerie per la grafica 2D (matplotlib)

Installazione dell'interprete (versione 3.2.3)

Per prima cosa installare l'interprete Python.

In questo corso useremo la versione 3.2.3 che può essere scaricata al link qui sotto.

<http://www.python.org/download/releases/3.2.3/>

Numpy (librerie di base per calcolo numerico)

Le numpy per python 3.2 e windows si trovano qui:

<http://www.lfd.uci.edu/~gohlke/pythonlibs/#numpy>

in particolare scegliere la versione per python 3.2 e win 32
numpy-MKL-1.6.2.win32-py3.2. exe

é possibile testare la installazione del pacchetto aprendo una shell di python e quindi digitando nella finestra dell'interprete:

```
In [1]: #import numpy
        #numpy.test()
```

matplotlib (librerie per la grafica 2D)

Scaricare il pacchetto autoinstallante qui:

<http://www.lfd.uci.edu/~gohlke/pythonlibs/#matplotlib>

in particolare scegliere
matplotlib-1.2.0.win32-py3.2. exe

Installazione da sorgente

alcuni riferimenti utili:

- qui si trova il codice sorgente (ma anche i pacchetti pacchetti dmg per mac OS) per l'interprete: <http://www.python.org/download/releases/3.2/>
- codice sorgente (ma anche pacchetti pre-compilati) per numpy <http://pypi.python.org/pypi/numpy>
- codice sorgente per matplotlib <http://pypi.python.org/pypi/matplotlib>