



## 4 brevi lezioni di Python

### Guida all'installazione del materiale necessario - 2019

Francesco Maio - Damiano Lucarelli - Giulio Carotta

Lo scopo di questo piccolo corso è quello di assistere gli studenti alle prime armi nell'approccio al mondo della programmazione, con particolare enfasi sull'utilizzo che se ne fa in ambito scientifico: analisi dei dati, produzione di grafici, procedure di best fit, simulazioni numeriche e tanto altro. Partendo dalle basi, vedremo come si effettua ciascuna di queste operazioni utilizzando il linguaggio di programmazione Python, adottato da buona parte della comunità scientifica. Se non avete idea di cosa sia davvero un linguaggio di programmazione, non preoccupatevi, ne parleremo. Le lezioni saranno il più possibile interattive e cercheremo di illustrare i concetti principali della materia attraverso una moltitudine di esempi pratici, piuttosto che concentrandosi sugli aspetti teorici. Lo scopo sarà, infatti, quello di rendervi in grado di produrre il prima possibile codice pulito, funzionale e funzionante ogniquale volta ne avrete bisogno, partendo chiaramente dalle esperienze in laboratorio che dovrete affrontare a breve! A seconda dei vostri gusti, vi può far comodo o meno portare con voi il vostro computer personale. Fate come più vi aggrada, ma, in ogni caso, è fondamentale che i vostri apparecchi siano in grado di eseguire i codici che andremo ad analizzare. Scriviamo questo documento per aiutare chi ha bisogno nell'installazione e nella configurazione di tutto ciò che serve.

## 1 Installazione dell'ambiente: Pyzo

Il primo passo è procurarsi l'ambiente software tramite il quale è possibile scrivere, gestire e compilare il codice. La scelta su quale ambiente utilizzare è chiaramente arbitraria e soggetta al gusto del singolo. Noi raccomandiamo l'utilizzo di Pyzo, che tra l'altro è l'ambiente tradizionalmente installato sui computer dei laboratori dell'università. Alla pagina <https://pyzo.org/start.html> è possibile trovare i link per scaricare l'opportuno installer a seconda del sistema operativo che si usa (quelli indicati sotto lo Step 1). Si faccia anche attenzione alla differenza tra gli installer per sistemi a 32 o 64 bit<sup>1</sup>. Nel caso in cui vi piaccia smanettare con i sistemi Linux, consigliamo come procedura alternativa (e più immediata) accedere al terminale e digitare i seguenti comandi:

---

<sup>1</sup>Al seguente link potete trovare informazioni per scoprire, nel caso in cui non lo sapeste, se l'architettura del vostro computer è a 32 o 64 bit: <https://support.microsoft.com/it-it/help/15056/windows-7-32-64-bit-faq>

```
1 $ sudo apt-get install python3-pip python3-pyqt5
2 $ sudo python3 -m pip install pyzo --upgrade
3 $ pyzo
```

Tramite l'ultimo comando si accede alla schermata dell'ambiente Pyzo. A seconda della distribuzione che si utilizza potrebbe essere necessario utilizzare il comando yum al posto di apt-get, in particolare se utilizzate Fedora e derivati invece di Debian/Ubuntu.

## 2 Installazione dell'interprete: Anaconda

Python è un linguaggio di programmazione che viene prevalentemente utilizzato in forma interpretata. A lezione vedremo brevemente ciò cosa vuol dire, ma, in sostanza, è necessario un programma in grado di tradurre il codice da noi scritto, riga per riga, in una forma che sia comprensibile al processore del calcolatore. Tale programma è detto interprete e va necessariamente affiancato a Pyzo. Tra i tanti, noi raccomandiamo Anaconda, che porta in automatico tutti i pacchetti necessari per il lavoro scientifico. Esso è reperibile al seguente indirizzo: <https://www.anaconda.com/download/>. Allo scopo di mantenere la compatibilità con il sistema Pyzo si raccomanda di scaricare la versione corrispondente a Python 3 e non Python 2. Alternativamente è possibile procurarsi Miniconda, che è una versione ridotta e più leggera di Anaconda che arriva con molti meno pacchetti, ma occupa chiaramente meno spazio in memoria. Esso è reperibile al seguente indirizzo: <https://conda.io/miniconda.html>. È fortemente consigliato installare l'interprete nella cartella di default, in modo da rendere più semplice il lavoro di riconoscimento del programma da parte di Pyzo. Una volta installato l'interprete, aprendo Pyzo dovrete essere in grado di riconoscere sulla sinistra un editor di testo e sulla destra, uno sopra l'altro, una console per l'inserimento dei comandi e un file browser per accedere in modo più immediato alle cartelle del computer. Python, come tanti altri linguaggi di programmazione, dispone di pacchetti di funzioni già pronte e direttamente utilizzabili da parte del programmatore. Nel caso in cui abbiate optato per Miniconda, è probabile che abbiate bisogno di scaricare alcuni pacchetti aggiuntivi. L'operazione può essere effettuata accedendo alla console di Pyzo e digitando semplicemente:

```
1 $ install <nome_del_pacchetto>
```

oppure:

```
1 $ pip install <nome_del_pacchetto>
```

Nel corso ci interesseranno, praticamente, soltanto i pacchetti scaricabili attraverso il seguente comando:

```
1 $ install numpy scipy matplotlib math
```

Se eseguendo questo comando vi appare un errore riguardante uno di questi pacchetti, non scoraggiatevi: probabilmente è già presente nella vostra versione di Python. Per verificare ciò provate ad eseguire questo comando:

```
1 $ import <nome_del_pacchetto>
```

Questo è un comando che vedremo spesso a lezione. Se non appaiono altri errori, potete stare tranquilli.

Con questo dovrete essere a posto. Non esitate a contattarci in caso di qualsiasi problema e/o suggerimento. Ci potete trovare Venerdì 26 Ottobre, dove saremo a disposizione per aiutarvi in caso di problemi durante l'installazione. A presto!<sup>2</sup>

---

<sup>2</sup>Autori originali: Antonio D'Abbruzzo e Maria Domenica Galati