

# Introduzione

Prima di iniziare è il caso di spendere qualche parola per spiegarti come è fatto questo libro e cosa contiene ogni capitolo.

## A chi si rivolge questo libro

A tutti quelli che vogliono imparare a programmare. Nessuno escluso. Dai 13 ai 99 anni con una doverosa premessa: alcuni concetti richiedono conoscenze e competenze, soprattutto di matematica, che normalmente sono acquisite in corsi di studio nella scuola primaria e secondaria di primo e secondo grado.

Se trovi qualche argomento che non conosci o che ti risulta difficile non scoraggiarti: affrontalo cercando di capire alcuni passaggi e l'obiettivo finale da raggiungere. Piano piano, con la lettura e soprattutto la pratica, alcune cose ti saranno più chiare. Tieni a mente che nessuno conosce e capisce tutto della programmazione o di Python. Infatti, stiamo parlando di concetti e tecnologie in continua evoluzione e, come diceva lo scrittore di fantascienza Arthur C. Clarke, "Ogni tecnologia sufficientemente avanzata è indistinguibile dalla magia".

## Requisiti per la lettura

Non ci sono requisiti particolari per la lettura di questo libro, oltre quello di avere un computer, essere collegati a Internet e aver già assaggiato almeno un po' di logica e "insalata di matematica" (aver guardato il cartone animato giapponese *Goldrake* o la serie televisiva *Monty Python's Flying Circus* non è necessario, ma aiuta). Sapere un po' di inglese è utile, ma non indispensabile.

Questo non è un manuale di Python. Ci siamo sforzati di trattare gli argomenti e i concetti esposti in maniera il più possibile rigorosa. Se qualche volta non ci siamo riusciti questo è dovuto a una scelta che mette al primo posto l'aspetto didattico oppure a limiti ed errori dell'autore, che ti chiediamo cortesemente di segnalare, e per i quali ci scusiamo in anticipo.

La connessione a Internet serve per scaricare e installare Python o, eventualmente, per lavorare con le versioni online del linguaggio.

## Struttura del libro

Il libro contiene figure che ti aiuteranno a capire come sono fatti i programmi in Python e cosa otterrai eseguendoli.

L'impostazione sarà per lo più basata sul *learning by doing* (imparare facendo), *learning by design* (imparare progettando) e *project-based learning* (apprendimento basato sull'affrontare la realizzazione di progetti).

L'idea generale è semplice e consiste in 4 passaggi:

- introduzione di un problema da risolvere o un progetto da costruire (obiettivo);
- presentazione di una possibile realizzazione, cioè del programma Python, che fornisce la soluzione al problema (implementazione);
- presentazione di elementi di Python, di nozioni teoriche della programmazione e di altri strumenti che servono a raggiungere l'obiettivo e a comprendere meglio come lo si è raggiunto (teoria, linguaggio e strumenti);
- proposta di alcuni spunti per migliorare o estendere quanto costruito.

I concetti teorici e gli strumenti sono pertanto affrontati in funzione della risoluzione di problemi specifici (*problem solving*). In qualche capitolo troverai alcune ripetizioni, utili comunque a riprendere, consolidare e approfondire le caratteristiche del linguaggio.

Il nostro obiettivo finale è esattamente quello del titolo: imparare a programmare con Python. Quindi, la trattazione sarà, almeno lo speriamo, semplice e lineare, lasciando a libri più avanzati una trattazione più completa e approfondita.

### Box

Troverai anche alcuni box (li riconosci dallo sfondo colorato) che contengono approfondimenti sul tema affrontato in un determinato punto del testo. Puoi leggerli o decidere di ignorarli se non ti interessa conoscere di più su quell'argomento.

Ci saranno qua e là anche delle note, più brevi dei box e accompagnate da una caratteristica icona: contengono suggerimenti e consigli che ti aiutano a capire meglio ed evitare gli errori più comuni. Per questo motivo è meglio non ignorarle.

NOTA

In fondo a ogni capitolo c'è un piccolo riassunto utile a monitorare i tuoi progressi.

## Presentazione dei capitoli

- **Capitolo 1 – Cos'è Python:** introduce la programmazione e presenta Python, la sua storia, la sua filosofia e la comunità di appassionati e programmatori che ruota attorno a questo linguaggio.
- **Capitolo 2 – Primi passi con Python:** spiega come installare Python e come usarlo online e offline. Descrive l'interfaccia base per iniziare a programmare e interagire con l'interprete Python.
- **Capitolo 3 – Una calcolatrice programmabile:** presenta gli operatori matematici di base e alcuni programmi che permettono di risolvere semplici problemi. Introduce alcune funzioni predefinite, i concetti di variabile e di tipo di dato e lo stile di programmazione.
- **Capitolo 4 – Sequenza, selezione, operatori relazionali e logici:** introduce alcuni programmi per la risoluzione di problemi, per lo più di natura matematica, attraverso le strutture di controllo sequenza e selezione e l'uso degli operatori di confronto e booleani.
- **Capitolo 5 – I cicli e la geometria della tartaruga:** utilizza la “geometria della tartaruga” per disegnare sullo schermo percorsi e figure geometriche, avvalendosi di cicli per ripetere blocchi di istruzioni un determinato numero di volte oppure fintanto che una condizione è vera.
- **Capitolo 6 – Moduli, funzioni e numeri (quasi) casuali:** spiega come scrivere codice che puoi riutilizzare, richiamandolo quando ti serve. Inoltre, mostra come la generazione di numeri casuali permetta la costruzione di programmi belli e interessanti.
- **Capitolo 7 – Stringhe, liste, tuple, insiemi e dizionari:** passa in rassegna i tipi di dato composti: stringhe, liste, tuple, set e dizionari, che aiutano, in particolare, a risolvere problemi legati ai testi e alla crittografia.
- **Capitolo 8 – Errori, test e debugging:** parla degli errori come di qualcosa con cui dobbiamo imparare a convivere al meglio nel mondo della programmazione. Spiega come gestire gli errori a *run time* e come si può scrivere codice che controlla altro codice.

- **Capitolo 9 – File, dati e statistica:** mostra come Python ben si presta alla risoluzione di problemi di tipo statistico e alla gestione di dati memorizzati su file.
- **Capitolo 10 – Grafica e immagini:** dopo un'introduzione sulla rappresentazione del colore e delle immagini in un computer, utilizza questi concetti per costruire programmi per l'elaborazione delle immagini col supporto di una libreria esterna. Inoltre, accenna alla creazione di applicazioni dotate di interfaccia grafica a finestre.
- **Capitolo 11 – Giochi e videogiochi:** ti guida nella programmazione di alcuni semplici videogiochi *arcade 2D*, che mettono assieme quanto appreso nel resto del testo, approfondendo, in particolare, il concetto di sprite, l'input tramite tastiera e mouse, la gestione di scenari e del punteggio.
- **Capitolo 12 – Verso l'infinito e oltre!:** elenca (solo) alcune delle estensioni e moduli per utilizzare funzionalità avanzate di Python.

## Programmi presentati nel libro

Tutti i programmi presentati nel testo e alcuni altri, per cui non c'è stato spazio nel libro, sono reperibili sul sito di Apogeo all'indirizzo <http://bit.ly/apo-mb-python>.

## Credits

L'idea per “Quanti giorni mancano a Natale?” nel Capitolo 6 e alcune elaborazioni con le immagini nel Capitolo 10 sono riprese da analoghi programmi di Andrea Manzini (<http://ilmanzo.github.io/>). I programmi “Sommatore grafico” del Capitolo 10 e “Occhio al fungo” del Capitolo 11 sono ripresi da analoghi programmi dei miei studenti di informatica Alessandro Marchioro, Leonardo Bonadimani e Matteo Guardini.

## Software

Python (<http://www.python.org/>), sviluppato da Guido van Rossum e gestito dalla *Python Software Foundation*.

Inkscape (<http://inkscape.org/>), editor per la grafica vettoriale gratuito, open source e multiplatforma.

GIMP (<http://gimp.org/>), programma per l'elaborazione immagini raster gratuito, open source e multiplatforma.

## Immagini e suoni

Laddove nel testo non viene espressamente indicata la fonte (in base ai termini di licenze come la *Creative Commons Attribution-Share Alike*, CC BY-SA, <https://creativecommons.org/licenses/by-sa/3.0/deed.it>), i file per le immagini e i suoni utilizzati nei progetti del libro sono di pubblico dominio (licenza CC0) e sono liberamente scaricabili da Internet da siti quali <https://pixabay.com/>, <https://opengameart.org/>, <https://www.vecteezy.com/>, <http://www.widgetworx.com/projects/s1.html>. Un riconoscimento speciale va all'arte e all'inventiva di Kenney (<http://kenney.nl/>) e di Karlisson M. Bezerra (<http://hacktoon.com/>).

## Ringraziamenti

Questo libro ha visto la luce grazie al contributo di molte persone, tra cui: Chiara, sposa che mi supporta e mi sopporta, mio figlio Lorenzo che ha fatto da *beta-tester* di alcuni programmi, mio fratello Luca Boscaini, mio cugino Massimiliano Masetti (qui ho terminato con la famiglia :-)), Ugo Solitro, Alessandro Marchioro, Davide Tonin e Alessia Marcolini che hanno collaborato con professionalità e generosità alla correzione del testo, il FabLab Verona (<http://www.veronafablab.it/>) per l'apertura e la disponibilità anche nell'organizzare corsi Python.

Nella conoscenza di Python i gradi di separazione e di ringraziamento tra l'autore e Guido van Rossum sono segnati da programmatori geniali e "open" come Stefano Scipioni e Alex Martelli.

Un ringraziamento finale va anche a Marco Beri e a tutta la fantasmagorica community di Python Italia.



**Figura I.1** Loghi del "Verona FabLab" e della Conferenza "Pycon 8" organizzata a Firenze da Python Italia nell'aprile 2017.