PYTHON PER RAGAZZI

UN'INTRODUZIONE GIOCOSA ALLA PROGRAMMAZIONE

JASON R. BRIGGS

EDIZIONI LSWR

Hanno detto di

PYTHON PER RAGAZZI

"Un'esperienza eccellente per tutta la famiglia."
—Patrice Gans, *Education Week*

"È una buona introduzione anche per gli adulti che imparano a programmare."
—Matthew Humphries, Geek.com

"Jason Briggs riesce a descrivere la programmazione ai ragazzi senza rendere stupidi i contenuti. Le lezioni sono costruite bene e lasciano il lettore, alla fine di ogni capitolo, con la sensazione di aver raggiunto un traguardo."
—Marziah Karch, *GeekMom,* blog di Wired.com

"Ben scritto e coinvolgente. Non c'è nulla di più intrigante che vedere delle semplici righe di codice che creano qualcosa sullo schermo e sapere come e perché sia successo."
—Copil Yáñez, *Full Circle*

"Un'eccellente introduzione alla programmazione, per chiunque sia interessato a imparare a programmare, indipendentemente dalla sua età. Un grande risorsa, sia a casa che a scuola."
—Roy Wood, *GeekDad*

"Ho chiesto a mia figlia di otto anni che cosa pensasse di *Python per ragazzi*. Ha scelto cinque stelle… Mi fa molto piacere che mia figlia scriva dei veri programmi in Python seguendo le pagine di questo libro."
—Richard Bejtlich, CSO di Mandiant

"Un grande libro, per chiunque voglia entrare nel mondo della programmazione senza sensi di inadeguatezza."
—Sandra Henry-Stocker, *ITworld*

PYTHON PER RAGAZZI

UN'INTRODUZIONE GIOCOSA ALLA PROGRAMMAZIONE

JASON R. BRIGGS

EDIZIONI LSWR

no starch press

Titolo originale: *Python for Kids | A Playful Introduction to Programming*
ISBN: 978-1-59327-407-8
Published by No Starch Press, Inc.
245 8th Street, San Francisco, CA 94103
www.nostarch.com
Cover and Interior Design: Octopod Studios

Edizione italiana:
Python per ragazzi | Un'introduzione giocosa alla programmazione
Traduzione di: Virginio B. Sala
Editor in Chief: Marco Aleotti

ISBN: 978-88-6895-348-5

EDIZIONI
LSWR
Via G. Spadolini, 7
20141 Milano (MI)
Tel. 02 881841
www.edizionilswr.it
Printed in Italy

Finito di stampare nel mese di ottobre 2016 presso "LegoDigit" Srl, Lavis (TN)

(*) Edizioni Lswr è un marchio di La Tribuna Srl. La Tribuna Srl fa parte di LSWR GROUP.

INDICE IN BREVE

INDICE GENERALE

PARTE II: BOUNCE!

PARTE III: L'AVVENTUROSA FUGA DI MR. STICK MAN

L'AUTORE

Jason R. Briggs programma da quando aveva otto anni e ha imparato il BASIC su un Radio Shack TRS-80. Ha scritto software a livello professionale, come sviluppatore e architetto di sistema, e ha collaborato come redattore esterno al *Java Developer's Journal*. Suoi articoli sono stati pubblicati su *JavaWorld*, *ONJava*, *ONLamp*. *Python per ragazzi,* è il suo primo libro.

Potete raggiungere Jason sul sito *http://jasonrbriggs.com/* o via posta elettronica all'indirizzo *mail@jasonrbriggs.com*.

L'ILLUSTRATORE

Miran Lipovača è autore di *Learn You a Haskell for Great Good!*. Ama il pugilato, la chitarra basso e, ovviamente, disegnare. È affascinato dagli scheletri che ballano e dal numero 71 e, quando entra dalle porte automatiche, sostiene di aprirle con il potere della sua mente.

I REVISORI TECNICI

Diplomatosi da poco alla The Nueva School, Josh Pollock, 15 anni, è matricola alla Lick-Wilmerding High School di San Francisco. Ha iniziato a programmare in Scratch a nove anni, ha iniziato a usare il TI-BASIC quando ne aveva dieci e subito dopo è passato a Java e Python e infine a UnityScript. Oltre a programmare, ama suonare la tromba, sviluppare giochi per computer e parlare di scienza.

Maria Fernandez ha una laurea in linguistica applicata e si interessa di computer e tecnologia da oltre 20 anni. Ha insegnato inglese alle giovani rifugiate con il Global Village Project in Georgia e ora vive in California settentrionale, dove lavora per ETS (Educational Testing Service).

RINGRAZIAMENTI

Ci si deve sentire così, quando si sale sul palco per ricevere un premio e ci si rende conto di aver dimenticato l'elenco delle persone da ringraziare negli altri pantaloni. Di sicuro ti dimenticherai di qualcuno, e subito partirà la musica che segnala che è venuto il momento di scendere dal palco.

Detto questo, ecco l'elenco (sicuramente incompleto) delle persone verso cui ho un enorme debito di gratitudine per aver contribuito a fare di questo libro il buon libro che penso sia.

Grazie alla redazione di No Starch, in particolare a Bill Pollock, per aver usato una sana dose di "che cosa penserebbe un bambino" mentre lo redazionava. Quando si programma da molto tempo, è facile dimenticare quanto siano difficili certe cose per chi è alle prime armi, e Bill è stato prezioso nell'indicare le parti inutilmente complicate. Grazie anche a Serena Yang, straordinaria production manager, con la speranza che non abbia perso troppi capelli nell'attribuire i colori giusti a 300 e più pagine di codice.

Un grande grazie a Miran Lipovaca per le sue brillanti illustrazioni. Più che brillanti. Davvero! Se avessi fatto io i disegni, al massimo ci sarebbe qualche figura abborracciata che non assomiglierebbe a niente in particolare. È un orso…? O un cane…? No, aspetta… forse un albero?

Grazie ai revisori. Mi scuso se qualcuno dei vostri consigli alla fine non è stato realizzato. Probabilmente avevate ragione voi, ma la colpa è di qualche piccola mania personale. Un grazie particolare a Josh per alcuni suggerimenti davvero eccellenti. E le mie scuse a Maria se qualche volta la formattazione del codice lasciava un po' a desiderare.

Grazie a mia moglie e mia figlia, per aver sopportato un marito e padre con il naso sepolto ancora più spesso del solito in uno schermo di computer.

A mia madre, per gli infiniti incoraggiamenti nel corso degli anni.

Infine, grazie a mio padre per avermi acquistato un computer ancora negli anni Settanta e aver tollerato che volessi usarlo il più possibile. Nulla di tutto questo sarebbe stato possibile senza di lui.

INTRODUZIONE

Perché imparare a programmare?

La programmazione favorisce la creatività, il ragionamento e la capacità di risolvere problemi. Chi programma ha la possibilità di creare qualcosa dal nulla, di usare la logica per trasformare i costrutti della programmazione in una forma che un computer può eseguire e, quando le cose non vanno proprio come avrebbero dovuto, di usare la capacità di risolvere problemi per stabilire che cosa sia andato storto.

Programmare è un'attività divertente, a volte impegnativa (e ogni tanto anche un po' frustrante) e le capacità che si apprendono programmando possono essere utili sia a scuola sia nel lavoro… anche se quel che fate non ha nulla a che vedere con i computer.

Infine, se non altro, la programmazione è un modo eccellente per passare un pomeriggio quando fuori fa brutto tempo.

PERCHÉ PYTHON?

Python è un linguaggio di programmazione facile da imparare, ma con alcune caratteristiche davvero molto utili per chi è alle prime armi. Il codice è molto facile da leggere, rispetto a quello di altri linguaggi di programmazione, e ha una interfaccia interattiva (la "shell") in cui si possono scrivere i programmi e vederli girare. Oltre alla semplicità della struttura del linguaggio e alla shell interattiva con cui si può sperimentare, Python ha alcune altre caratteristiche che rafforzano di molto il processo di apprendimento e permette di creare semplici animazioni per creare i propri giochi. Uno di questi elementi è il modulo turtle, ispirato alla grafica della tartaruga (utilizzato dal linguaggio di programmazione Logo già negli anni Sessanta) e pensato per le applicazioni in campo educativo. Un altro è il modulo tkinter, un'interfaccia per Tk, un "toolkit", cioè un insieme di strumenti, per le interfacce grafiche, che mette a disposizione un metodo semplice per creare programmi con grafica e animazioni un po' più avanzate.

COME IMPARARE A SCRIVERE CODICE

Come sempre, quando si prova a fare qualcosa per la prima volta, è meglio iniziare dalle basi, perciò cominciate con i primi capitoli e non fatevi prendere dalla tentazione di saltare fino ai capitoli più avanzati. Nessuno è in grado di suonare una sinfonia la prima volta che prende in mano uno strumento musicale. Gli aspiranti piloti non fanno volare un aereo prima di aver capito i controlli fondamentali e i ginnasti (di solito) non sono capaci di fare un salto mortale

al primo tentativo. Se andate avanti troppo rapidamente, non solo non vi resteranno in testa le idee fondamentali, ma i contenuti dei capitoli successivi vi sembreranno più complicati di quel che sono in realtà.

Nel procedere, provate tutti gli esempi: così potrete vedere come funzionano. Alla fine della maggior parte dei capitoli ci sono anche dei rompicapo di programmazione che potete cercare di risolvere: vi aiuteranno a migliorare le vostre abilità di programmazione. Ricordate: quanto meglio assimilerete le basi, tanto più facile vi risulterà capire idee più complicate in seguito.

Quando incontrate qualcosa di frustrante o di troppo impegnativo, ecco alcune cose che io trovo utili:

1. Suddividete un problema in parti più piccole. Cercate di capire che cosa fa un piccolo frammento di codice, oppure ragionate solo su una piccola parte di un'idea difficile (concentratevi su un piccolo frammento di codice invece di cercare di capire il tutto in una volta sola).
2. Se questo ancora non vi è d'aiuto, a volte la cosa migliore è semplicemente non pensarci per un po'. Dormiteci sopra e riprendete il problema il giorno dopo. È un buon modo per risolvere molti problemi e può essere particolarmente utile per la programmazione.

CHI DOVREBBE LEGGERE QUESTO LIBRO

Questo libro è per chiunque sia interessato alla programmazione, bambino o adulto che a questo campo si avvicini per la prima volta. Se volete imparare a scrivere il vostro software anziché usare solamente i programmi scritti da altri, questo libro è un ottimo punto di partenza.

Nei capitoli seguenti, troverete le informazioni per installare Python, eseguire la shell e svolgere qualche calcolo, stampare il testo che appare sullo schermo, creare liste e svolgere alcune semplici operazioni di controllo del flusso utilizzando gli enunciati if e i cicli for (e scoprirete che cosa sono gli enunciati if e i cicli for!). Vedrete come riutilizzare il codice con le funzioni, gli elementi fondamentali di classi e oggetti e la descrizione di alcune funzioni e alcuni moduli, fra i molti che fanno parte di Python.

Troverete capitoli sulla grafica della tartaruga, a livello più semplice e più avanzato, e anche sull'uso del modulo `tkinter` per disegnare sullo schermo. Alla fine di molti capitoli vi sono rompicapo di programmazione di varia complessità, che vi aiuteranno a consolidare le conoscenze appena acquisite, con la possibilità di scrivere voi stessi piccoli programmi.

Una volta acquisita la conoscenza dei fondamenti della programmazione, imparerete come scrivere i vostri giochi. Svilupperemo due giochi grafici e scopriremo il rilevamento delle collisioni, gli eventi e varie tecniche di animazione.

La maggior parte degli esempi nel libro usa la shell IDLE (*Integrated DeveLopment Environment*, ambiente integrato di sviluppo) di Python. IDLE evidenzia la struttura sintattica, offre funzioni di copia e incolla (analoghe a quelle che usereste in altre applicazioni) e una finestra di editor in cui potete salvare il vostro codice per poterlo usare ancora in seguito, il che significa che IDLE funziona sia come un ambiente interattivo per la sperimentazione e anche un po' come un editor di testo. Gli esempi funzioneranno altrettanto bene con la console standard e un normale editor di testo, ma l'evidenziazione della sintassi e un ambiente un po' più amichevole come quello di IDLE possono essere di aiuto per capire meglio, perciò nel primo capitolo vedremo come impostarlo.

CHE COSA C'È IN QUESTO LIBRO

Ecco una breve panoramica di quello che troverete in ciascun capitolo.

Il **Capitolo 1** è un'introduzione alla programmazione, con le istruzioni per installare Python.

Il **Capitolo 2** introduce i calcoli fondamentali e le variabili, e il **Capitolo 3** descrive alcuni dei tipi fondamentali di Python, come le stringhe, le liste e le ennuple.

Con il **Capitolo 4** arriva il primo assaggio del modulo `turtle`. Passeremo dagli elementi base della programmazione a come far muovere una tartaruga (sotto forma di una freccia) in giro per lo schermo.

Il **Capitolo 5** tratta delle variazioni delle condizioni e degli enunciati `if` (condizionali); il **Capitolo 6** procede con i cicli `for` e `while`.

Nel **Capitolo 7** inizieremo a usare e creare funzioni, poi nel **Capitolo 8** parleremo di classi e oggetti. Esamineremo le idee base abbastanza a fondo per poter poi utilizzare qualcuna delle tecniche di programmazione necessarie, nei capitoli più avanti, per lo sviluppo di giochi. A questo punto il materiale comincia a diventare un po' più complicato.

Il **Capitolo 9** tratta la maggior parte delle funzioni interne a Python e il **Capitolo 10** continua affrontando alcuni moduli (fondamentalmente, blocchi di funzionalità utili), installati per impostazione predefinita insieme a Python.

Il **Capitolo 11** ritorna al modulo `turtle` per sperimentare qualche forma più complicata. Il **Capitolo 12** passa all'uso del modulo `tkinter` per creare qualche oggetto grafico più avanzato.

Nei **Capitoli 13** e **14** creeremo il nostro primo gioco, "Bounce!", che si fonda sulle conoscenze acquisite nei capitoli precedenti; nei **Capitoli 15-18** creeremo un altro gioco, "L'avventurosa fuga di Mr. Stick Man". I capitoli dedicati allo sviluppo dei giochi sono quelli in cui le cose possono farsi davvero serie. Se incontrate problemi che non riuscite a risolvere altrimenti, scaricate il codice dal sito web che accompagna il libro (*http://python-for-kids.com/*) e confrontate il vostro codice con questi esempi, che sicuramente funzionano.

Nella **Postfazione**, concludiamo dando uno sguardo a PyGame e ad alcuni altri linguaggi di programmazione molto diffusi.

Infine, nell'**Appendice**, potrete trovare le parole chiave di Python trattate in dettaglio e, nel **Glossario**, troverete le definizioni dei termini della programmazione utilizzati in tutto il libro.

IL SITO WEB DI ACCOMPAGNAMENTO

Se vi trovate nelle condizioni di aver bisogno di aiuto mentre leggete, provate a consultare il sito di accompagnamento, *http://python-for-kids.com/*(in inglese), dove potrete trovare tutti gli esempi del libro e altri rompicapo di programmazione.

Troverete anche le soluzioni di tutti i rompicapo nel libro, nel caso vi troviate in difficoltà o vogliate controllare il vostro lavoro.

BUON DIVERTIMENTO!

Ricordate, mentre procedete, che la programmazione può essere divertente. Non prendetela come un lavoro. Pensate la programmazione come un modo per creare giochi o applicazioni divertenti che potete condividere con gli amici o con altri.

Imparare a programmare è un meraviglioso esercizio mentale e i risultati possono essere molto gratificanti. Ma soprattutto, qualsiasi cosa facciate, divertitevi!

PARTE I

IMPARARE A PROGRAMMARE

1

NON TUTTI I SERPENTI STRISCIANO

Un programma per computer è un insieme di istruzioni grazie alle quali un computer svolge qualche attività. Non si tratta delle parti fisiche di una macchina (come i fili, i microchip, le schede, il disco fisso e altre cose simili) ma di quella materia impalpabile e nascosta che "gira" su quell'hardware. Un programma per computer, che chiamerò in genere semplicemente *programma*, è l'insieme dei comandi che dicono a quell'hardware "stupido" che cosa fare. *Software* è un insieme di programmi.

Senza programmi, quasi ogni dispositivo che usate quotidianamente smetterebbe di funzionare o sarebbe molto meno utile di quel che è. I programmi, in una forma o in un'altra, controllano non solo il vostro personal computer ma anche i sistemi per i videogiochi, i telefoni cellulari e il navigatore satellitare nell'automobile. Il software controlla anche apparecchi meno ovvi, come i televisori a LCD e i loro telecomandi, radio, lettori di DVD, forni e anche qualche frigorifero dei più recenti. Anche i motori delle automobili, i semafori, i lampioni, i segnali dei treni, i tabelloni elettronici e gli ascensori sono controllati da programmi.

I programmi sono un po' come pensieri. Se non aveste pensieri, probabilmente ve ne stareste seduti sul pavimento, con lo sguardo perso nel vuoto e la saliva che vi gocciola sulla maglietta. Il pensiero "alzati dal pavimento" è un'istruzione, o un comando, che dice al vostro corpo di alzarsi. Allo stesso modo, i programmi dicono ai computer che cosa fare.

Se sapete scrivere programmi per computer, potete fare ogni genere di cose utili. Certo, magari non sarete in grado di scrivere programmi che controllano auto, semafori o il frigorifero (o almeno, non inizialmente), ma potrete creare pagine web, scrivere i vostri giochi o addirittura un programma che vi sia d'aiuto nei compiti a casa.

QUALCHE PAROLA SUL LINGUAGGIO

Come gli esseri umani, i computer possono usare molte lingue per comunicare – in questo caso, linguaggi di programmazione. Un linguaggio di programmazione è solo un modo particolare di parlare a un computer – un modo per usare istruzioni che sia gli esseri umani sia i computer sono in grado di comprendere.

Esistono linguaggi di programmazione che prendono il nome da persone (come Ada e Pascal), quelli che hanno come nome semplici acronimi (come BASIC e FORTRAN) e anche qualcuno che prende il nome da trasmissioni televisive, come Python. Sì, il linguaggio di programmazione Python ha preso il nome dalla trasmissione televisiva inglese *Monty Python's Flying Circus*, non dal serpente (*python* è la parola inglese che significa "pitone", se non l'avevate ancora capito).

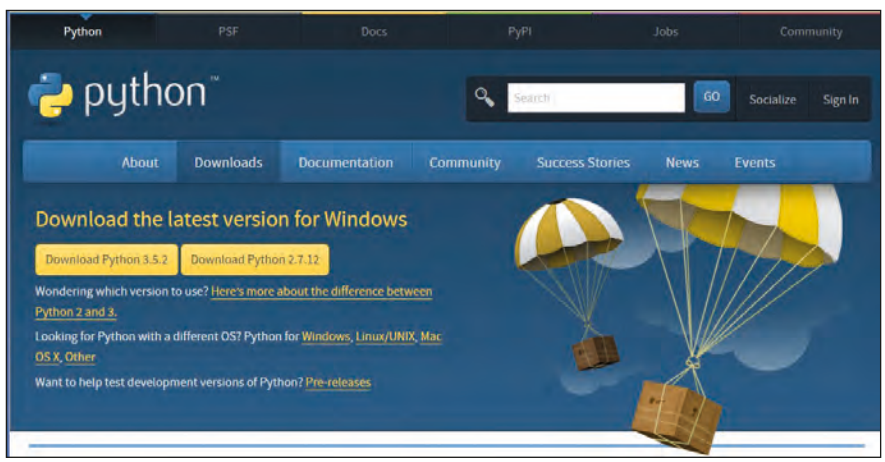
Parecchie caratteristiche fanno di Python un linguaggio di programmazione estremamente utile per chi è alle prime armi. Potete utilizzare Python per scrivere molto rapidamente programmi semplici ma molto efficaci. Non usa molti simboli complicati, a differenza di altri linguaggi di programmazione, ed è perciò più facile da leggere e molto più "amichevole" per chi è agli inizi. (Questo non vuol dire che Python non usi simboli – semplicemente il loro uso non è così esteso come in molti altri linguaggi.)

INSTALLARE PYTHON

L'installazione di Python è molto semplice. Qui vedremo come lo si installa in Windows 10, Mac OS X e Ubuntu. Installando Python, troverete anche un collegamento al programma IDLE (che significa Integrated DeveLopment Environment, ambiente di sviluppo integrato), che permette di scrivere programmi per Python. Se Python è già installato sul vostro computer, potete passare subito al paragrafo "Dopo aver installato Python" a pagina 10.

INSTALLARE PYTHON IN WINDOWS 10

Per installare Python per Microsoft Windows 10, usate il browser web, andate all'indirizzo *http://www.python.org/* e scaricate il programma di installazione più recente per Python 3. Nel menu del sito troverete una sezione intitolata **Downloads**. Un clic sul nome di questa sezione vi porterà a una finestra come questa:



NOTA

Quale sia esattamente la versione di Python che scaricate non è importante, basta che il suo numero inizi con il 3. Gli aggiornamenti sono frequenti: qui vedete le indicazioni per la versione 3.5.1, ma è probabile che quando leggerete queste pagine la versione più recente abbia un numero più alto.

Una volta scaricato il programma di installazione per Windows, fate un doppio clic sulla sua icona, poi seguite le istruzioni per installare Python nella posizione predefinita.

1. Selezionate la casella di controllo **Install launcher for all users** (se non è già selezionata).
2. Non cambiate la directory di destinazione predefinita, ma annotatevi il nome della directory (potrebbe essere C:\Python32, per esempio).
3. Ignorate la sezione *Customize installation* e fate clic su **Install now**.

Alla fine del procedimento, nel menu Start troverete, tra le nuove app, o fra le app installate di recente, alcune voci relative a Python.

Poi, seguite questi passi per aggiungere un collegamento a Python 3 al vostro desktop:

1. Fate clic destro in un punto vuoto del desktop, poi selezionate **Nuovo**, **Collegamento** dal menu di scelta rapida.
2. Nella casella sotto la scritta **Immettere il percorso per il collegamento**, scrivete (assicurandovi che il nome della directory sia quello che vi siete annotati in precedenza):

```
c:\Python\32\Lib\idlelib.idle.pyw -n
```

3. Fate clic su **Avanti** per passare alla finestra di dialogo successiva.
4. Inserite il nome come *IDLE* e fate clic su **Fine** per creare il collegamento.

Ora potete passare a "Dopo avere installato Python" a pagina 10 per iniziare.

INSTALLARE PYTHON SU MAC OS X

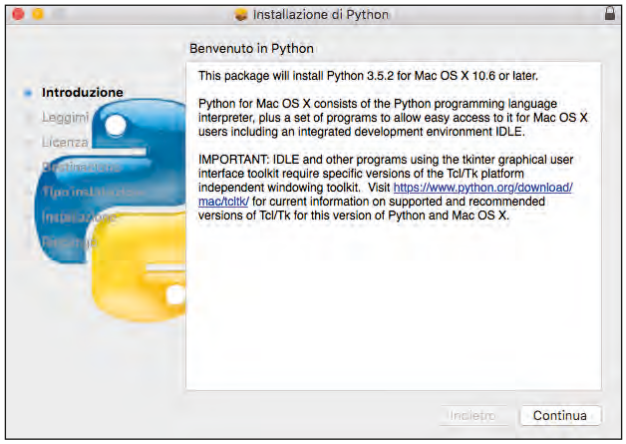
Se usate un Mac, dovreste avere una versione di Python già preinstallata, ma probabilmente si tratta di una versione più vecchia. Per essere sicuri di avere la più recente, aprite il browser e andate all'indirizzo *http://www.python.org/getit/* per scaricare il programma di installazione più recente per il Mac.

I programmi di installazione sono diversi: quello che dovete scaricare dipende dalla versione di Mac OS X che avete. (Se avete dubbi, potete scoprire il numero di versione del sistema operativo facendo clic sulla Mela nella barra dei menu in alto e scegliere Informazioni su questo Mac.)

Se il vostro computer ha una versione di Mac OS X compresa fra la 10.3 e la 10.6, scaricate la versione a 32 bit di Python 3 per i386/PPC.

Se il vostro computer ha una versione di Mac OS X 10.6 o superiore, scaricate la versione 64-bit/32-bit di Python 3 per x86-64.

Una volta scaricato il file (avrà un nome con estensione *.pkg* e lo troverete nella cartella Download), fate un doppio clic su di esso e partirà il processo guidato di installazione.

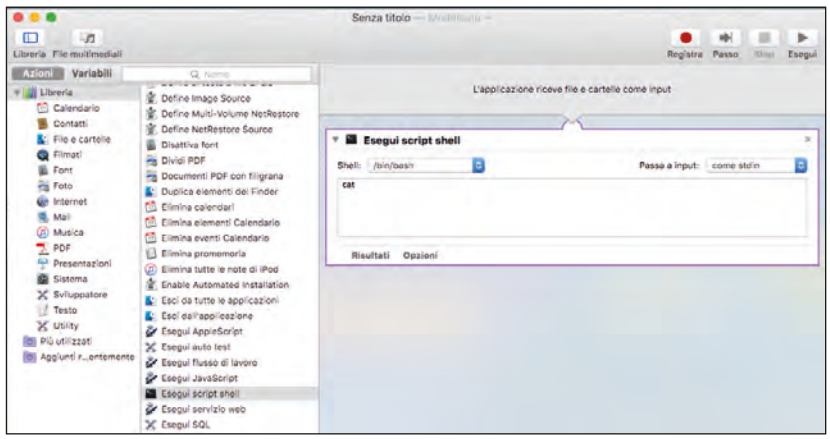


In questa finestra, fate doppio clic su *Python mpkg*, poi seguite le istruzioni per installare il software. Vi verrà chiesta la password di amministratore del vostro Mac. (Non conoscete la password di amministratore? Probabilmente dovrete chiedere ai vostri genitori di inserirla.)

Poi, dovete aggiungere al desktop uno script per lanciare l'applicazione IDLE di Python, in questo modo:

1. Fate clic sull'icona **Spotlight**, la lente di ingrandimento nell'angolo superiore destro della schermata.
2. Nella finestra che compare, scrivete *Automator*.
3. Fate clic sull'applicazione che ha l'aspetto di un robot, quando compare nel menu. Sarà nella sezione intitolata Più utilizzati o in **Applicazioni**.
4. Avviato Automator, selezionate il modello Applicazione:

5. Fate clic su **Scegli** per continuare.
6. Nell'elenco delle azioni, trovate **Esegui script shell** e trascinatelo nel riquadro vuoto a destra. Vedrete qualcosa di simile a questo:



7. Nella casella di testo, vedrete la parola *cat*. Selezionate questa parola e sostituitela con il testo seguente (tutto, da `open` a `-n`):

```
open -a "/Applications/Python 3.2/IDLE.app" -args -n
```

Può darsi che dobbiate cambiare la directory, a seconda della versione di Python che avete installato.

8. Selezionate **File**, **Salva** e inserite come nome *IDLE*.
9. Selezionate **Desktop** dalla finestra di dialogo Situato in, poi fate clic su **Salva**.

Ora potete passare a "Dopo avere installato Python" a pagina 10 per iniziare.

INSTALLARE PYTHON SU UBUNTU

Python è già preinstallato nella distribuzione Ubuntu di Linux, ma può darsi che non sia la versione più recente. Per installare Python 3 su Ubuntu procedete così:

1. Fate clic sul pulsante dell'Ubuntu Software Center nella Sidebar (è l'icona che ha l'aspetto di una borsa arancione – se non la vedete, potete sempre fare clic sull'icona Dash Home e inserire *Software* nella finestra di dialogo).

2. Inserite *Python* nella casella di ricerca (in alto al centro) del Software Center.
3. Nell'elenco del software che si presenta, selezionate la versione più recente di IDLE, che è, al momento in cui scrivo, *IDLE (using Python 3.5)*.
4. Fate clic su **Installa**.
5. Inserite la password di amministratore quando vi viene richiesta e procedete all'autenticazione. (Non conoscete la password di amministratore? Dovrete chiedere ai vostri genitori di inserirla per voi.)

NOTA *In qualche versione di Ubuntu, può darsi che vediate solo Python (v3.5) nel menu principale (anziché IDLE): potete installare quello.*

Con alcune versioni di Ubuntu la ricerca di Python attraverso il Software Center sembra non dare frutto. In tal caso, aprite una finestra di Terminale e al prompt inserite il comando seguente:

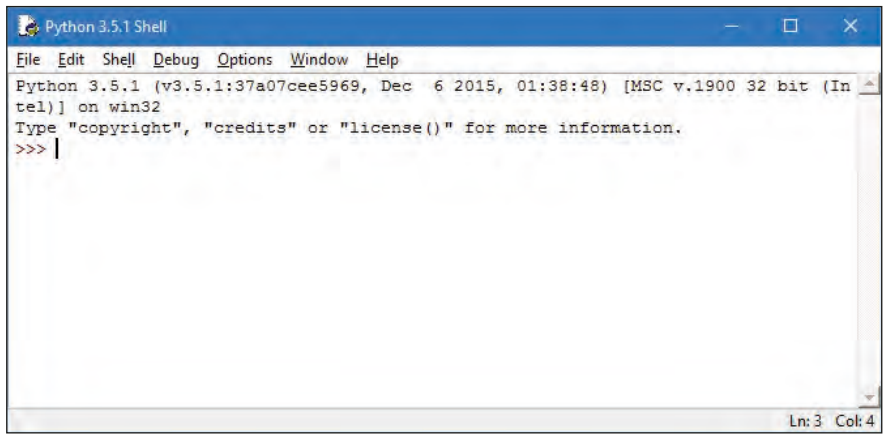
```
sudo apt-get install idle-python3.5
```

L'installazione è poi completamente automatica, tranne una richiesta di conferma (a cui dovete rispondere con un Sì). Al termine potrete trovare *IDLE (using Python 3.5)* nell'elenco del software installato nel Software Center, nella sezione Applicazioni. Se avete difficoltà a trovarlo, utilizzate la casella di ricerca di *Cerca sul computer* (è la prima icona in alto nella barra laterale).

DOPO AVER INSTALLATO PYTHON

Ora sul desktop del vostro computer (se avete una macchina Windows o un Mac OS X) dovreste avere un'icona **IDLE**. Se usate Ubuntu, a seconda della versione del sistema operativo, troverete l'icona nel menu **Applicazioni** oppure nel Software Center.

Fate un doppio clic sull'icona o scegliete l'opzione da menu e vi comparirà una finestra come questa:



Questa è la *shell di Python*, che fa parte dell'ambiente di sviluppo integrato. I tre simboli "maggiore di" (>>>) sono il cosiddetto *prompt*.

Inseriamo qualche comando al prompt, cominciando con questo:

```
>>> print("Ciao mondo")
```

Fate attenzione a non dimenticare i doppi apici (" "). Premete Invio sulla tastiera, quando avete finito di scrivere questa riga.

Se avete inserito correttamente il comando, vedrete qualcosa di analogo a questo:

```
>>> print("Ciao Mondo")
Ciao Mondo
>>>
```

Sarà ricomparso il prompt, per farvi sapere che la shell di Python è pronta ad accettare altri comandi.

Congratulazioni! Avete appena creato il vostro primo programma in Python. La parola `print` fa parte di quel tipo di comandi Python che sono chiamati *funzioni*, e "stampa" qualsiasi cosa si trovi all'interno delle parentesi.

In sostanza, avete dato al computer l'istruzione di visualizzare le parole "Ciao mondo" – una istruzione comprensibile sia a voi che al computer.

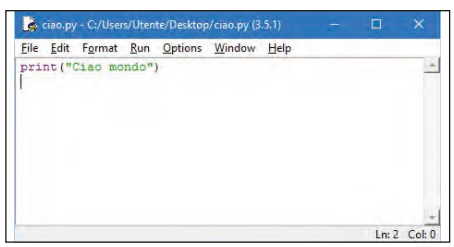
SALVARE I PROGRAMMI PYTHON

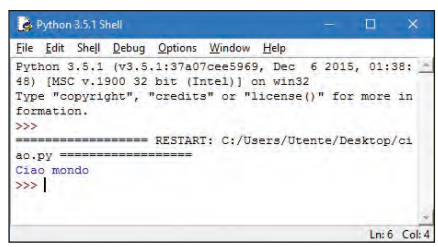
I programmi Python non sarebbero molto utili se fosse necessario riscriverli ogni volta che li si vuole usare, oppure visualizzare in modo da poterli consultare. Certo, un programma molto breve si potrebbe anche riscrivere, ma un programma di grandi dimensioni, come un elaboratore di testo, può essere costituito da milioni di righe di codice. Stampatele tutte e vi ritroverete magari con un malloppo da oltre 100.000 pagine. Difficile portarlo a casa sotto braccio – sperando che non arrivi una folata di vento.

Per fortuna, è possibile salvare i programmi per poterli riusare in futuro. Per salvare un nuovo programma, in IDLE scegliete **File**, **New File**. Si aprirà una nuova finestra, con il nome **Untitled** nella barra del titolo. Nella nuova finestra della shell scrivete questo codice:

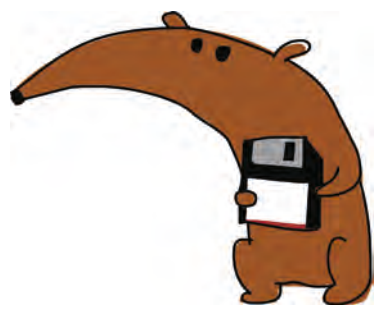
```
print("Ciao mondo")
```

Ora scegliete **File**, **Save**. Quando viene chiesto il nome del file, scrivete *ciao.py* e salvate il file sul desktop. Poi scegliete **Run**, **Run Module**. Se non avete commesso errori, il programma verrà eseguito, con questo risultato:





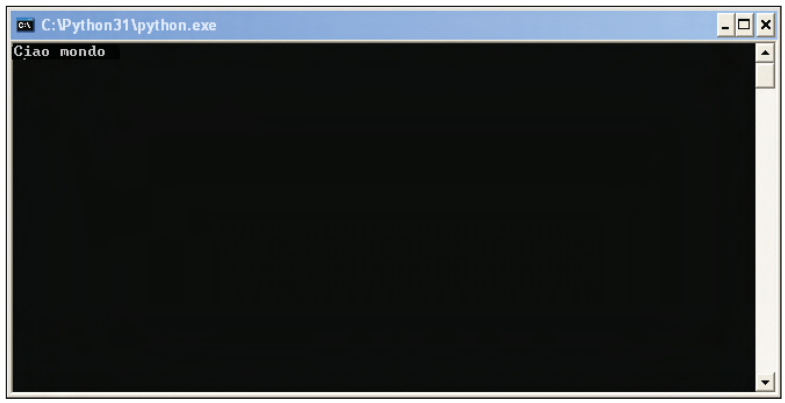
Ora, se chiudete la finestra della shell ma lasciate aperta la finestra ciao.py e scegliete di nuovo **Run**, **Run Module**, ricomparirà la shell di Python e il vostro programma verrà nuovamente eseguito.



(Per riaprire la shell di Python senza eseguire il programma, scegliete **Run**, **Python Shell**.

Dopo aver eseguito il codice, sul desktop troverete una nuova icona, con il nome *hello.py*. Se fate un doppio clic su quell'icona, apparirà brevemente una finestra con lo sfondo nero, e subito dopo svanirà. Che cosa è successo?

Avete visto la console da riga di comando di Python (analoga alla shell) che si avviava, stampava "Ciao mondo" e poi si chiudeva. Ecco quello che vi apparirebbe, se aveste una vista ultrarapida da supereroe e riusciste a vedere la finestra prima che si chiuda:



Oltre ai menu, potete usare i tasti di scelta rapida per creare una nuova finestra della shell, salvare un file ed eseguire un programma:

- In Windows e Ubuntu, usate Ctrl-N per creare una nuova finestra della shell, Ctrl-S per salvare il file dopo che avete finito di scriverlo, F5 per eseguire il programma.
- In Mac OS X, usate Opzione-N per creare una nuova finestra della shell, Opzione-S per salvare il file, mentre per eseguire il programma premete e tenuto premuto il tasto funzione (FN) e premete F5.

CHE COSA AVETE IMPARATO

Abbiamo iniziato in questo capitolo con un'applicazione Ciao mondo, il programma con cui praticamente tutti iniziano quando cominciano a studiare la programmazione. Nel prossimo capitolo, faremo qualche altra cosa utile con la shell di Python.