

Capitolo 2

Come installare Python

Per poter installare Python, dovremo scaricare la versione adeguata alla nostra piattaforma e, contestualmente, troveremo le istruzioni più aggiornate per l'installazione. Per questo motivo non dedicheremo troppo tempo a questa fase, ma cercheremo più che altro di dare le indicazioni utili per essere poi autonomi nella ricerca delle soluzioni ai problemi che potremmo incontrare. Come recitava un famoso proverbio inglese "dai dello *spam* a un uomo e lo sfamerai per un giorno, insegnagli a macinare un maiale e lo sfamerai per sempre".

Per chi se lo fosse già dimenticato, lo ripetiamo: lo *spam* è la carne in scatola resa famosa dai Monty Python. Scrivere un libro su Python senza far riferimento ogni tanto allo *spam* è considerato politicamente scorretto dalla comunità *pythonista*.

Windows

Per installare Python su Windows è sufficiente accedere al sito ufficiale <http://www.python.org> e fare clic sulla voce *Download*, normalmente presente nel menu a sinistra.

A questo punto arriveremo a una pagina dove troveremo un elenco di versioni di Python: scegliamo la versione 3.1.1.

Naturalmente, al posto della versione 3.1.1 potremmo trovare una nuova versione, come la 3.1.2 o la 3.2: scegliete pure l'ultima versione che trovate.

Se non dovessimo riuscire a trovare l'elenco, possiamo accedere direttamente alla pagina che ci serve digitandone l'indirizzo URL:

<http://www.python.org/download/releases/3.1.1/>



Facciamo attenzione all'eventuale indicazione "has been superseded" (è stata rimpiazzata), che indica il rilascio di una nuova versione.

Giunti alla pagina dedicata alla versione che ci interessa, possiamo fare clic sul link *Windows x86 MSI Installer* (oppure *Windows AMD64 MSI Installer* se abbiamo una macchina AMD64). Subito dopo, il nostro browser ci chiederà di salvare un file con un nome più o meno simile a questo: `python-3.1.1.msi`.

A partire dalla versione 2.4, Python è distribuito come file in formato Microsoft Installer, con estensione `.msi`. Per scoprire se la nostra versione di Windows supporta tale formato ci basterà fare clic sul file. Se partirà una schermata di installazione simile a quella rappresentata nella Figura 2.1, non dovremo fare altro che seguire le solite istruzioni che incontriamo quando installiamo un programma per Windows, accettando, senza preoccuparci troppo, tutte le opzioni di default.

Se invece il nostro sistema operativo non riconosce il formato `.msi`, ci apparirà una schermata simile a quella rappresentata nella Figura 2.2.

In questo caso, non perdiamoci assolutamente d'animo, facciamo clic su *Annulla* e andiamo sul sito Microsoft dove potremo cercare "Windows Installer" e scaricare gratuitamente (da non crederci...) la versione adatta al nostro sistema operativo.



Figura 2.1 Schermata di installazione iniziale di Python in formato Microsoft Installer (file con estensione `.msi`).

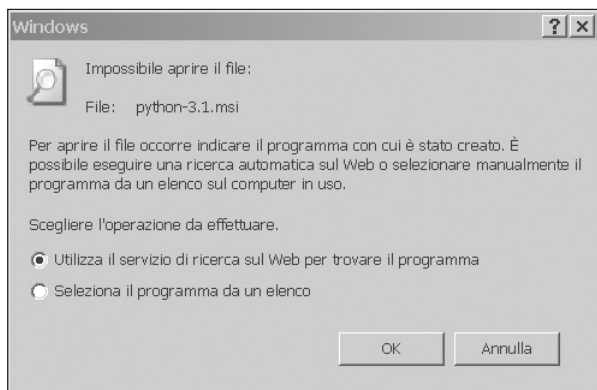


Figura 2.2 Il sistema operativo non riconosce il formato Microsoft Installer.

Ne esiste una versione per Windows 95/98 (il cui file di installazione si chiama `InstMsiA.exe`) e una per Windows NT 4.0/2000 (`InstMsiW.exe`). Windows XP e seguenti sono invece in grado di comprendere direttamente il formato `.msi`. Se non abbiamo Windows XP ma siamo riusciti comunque a installare Python, nessun mistero: molti programmi Microsoft, per esempio Word o Excel, portano in dote Microsoft Installer.

Arrivati a questo punto, anche i più sfortunati avranno avuto la possibilità di iniziare l'installazione. Completati i passi richiesti, nel menu dei programmi installati sul sistema troveremo una voce *Python 3.1*. Apriamola e scegliamo *Python (command line)*. Evviva! Ecco la finestra dell'interprete di Python dove finalmente possiamo provare il nostro primo programma:

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



Un'ultima indicazione utile per un uso più comodo di Python in Windows: se non riusciamo a eseguire direttamente l'interprete da una finestra del prompt dei comandi (il comando `cmd` per intenderci) è probabile che nella variabile di sistema `PATH` non sia presente la directory di installazione. Nel caso della versione 3.1.1, se non abbiamo scelto una directory diversa, tale directory sarà `C:\python31`. Possiamo aggiungerla manualmente dal Pannello di controllo, scegliendo Sistema, quindi la scheda Avanzate e infine il pulsante Variabili d'ambiente.

Linux

È estremamente probabile che sul nostro sistema Linux, Python sia già presente, ma nella versione 2.6 o 2.7. Per verificarlo è sufficiente aprire una finestra Terminale e provare a eseguire direttamente il comando `python`: la prima riga visualizzata ci indica la versione di Python installata.

Se non abbiamo la versione 3, abbiamo due alternative: cercare una versione binaria di Python in un formato compatibile per la nostra distribuzione di Linux (`.RPM` per RedHat, `.deb` per Debian o Ubuntu e così via) oppure usare un programma di installazione dei pacchetti (`apt-get` su Debian o `Synaptic Package Manager` su Ubuntu).

In realtà abbiamo una terza alternativa: scaricare e compilare Python partendo direttamente dal codice sorgente. A questa opzione è dedicato un intero paragrafo di questo capitolo.

Mac OS X

Anche nel caso di Mac OS X è molto probabile che Python 2.6 o 2.7 sia già installato sul nostro sistema. Di nuovo, per appurarlo, ci basta aprire una finestra Terminale ed eseguire il comando `python`.

Se Python 3 non dovesse essere presente sul nostro sistema, avremo comunque vita facile, visto che la versione binaria per l'installazione su Mac OS X è già pronta e disponibile sul sito di Python in formato `.dmg` (disk image) compatibile sia per Mac su PowerPC sia per Mac su Intel.

La possiamo scaricare direttamente dalla pagina dedicata alla versione che ci interessa. Per installarla basta fare clic sul file appena scaricato e anche con Mac OS X siamo pronti a partire per l'avventura *pythoniana*!

Compilazione dal codice sorgente

Per compilare Python a partire dal codice sorgente, dobbiamo innanzitutto recuperare proprio i file di codice sorgente. Visitiamo quindi la pagina della versione che ci interessa. Abbiamo già visto più volte in dettaglio questa operazione nei paragrafi precedenti, per cui la ripetiamo in breve. In fondo è facile supporre che chi voglia compilare Python abbia una certa dimestichezza con il proprio sistema operativo e con l'informatica in generale.

Andiamo alla seguente pagina per scegliere la versione:

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

Oppure digitiamo direttamente l'indirizzo URL:

<http://www.python.org/download/releases/3.1.1/>

Ora cerchiamo il link *Gzipped source tar ball* (oppure *Bzipped source tar ball* a seconda del formato di compressione che preferiamo) che ci permette di scaricare il file `Python-3.1.1.tgz`. Una volta scaricato il file dobbiamo espandere i file in esso contenuti. Il formato `.tgz` dovrebbe essere noto a tutti gli utenti Linux e Mac, un po' meno agli utenti Windows. Per recuperare un'utilità in grado di espandere tale formato, basta cercare su Google le stringhe `tgz` e `windows` e, voilà, ecco trovato l'indirizzo URL che ci serviva:

<http://www.gzip.org>

Espandiamo i file di codice sorgente in una directory a nostro piacimento, apriamo una finestra Terminale (o il prompt dei comandi se siamo in Windows) e posizioniamoci nella directory in questione.

Ora dobbiamo fare una distinzione. La compilazione dei file di codice sorgente in Linux o Mac OS X è semplice: basta eseguire la "solita danza" dei comandi `gunzip`, `tar`, `configure`, `make` e `make install`. Il comando `gunzip` espande il file `.tgz` in un file `.tar` da cui il comando `tar` estrae i file di codice sorgente. Il comando `configure` verifica la presenza dei programmi necessari alla compilazione (e le relative versioni), creando un apposito `makefile`. Il comando `make` compila i file di codice sorgente, creando i file per l'installazione. Infine il comando `make install` copia i file nelle directory di destinazione corrette. Se qualcosa non dovesse funzionare è probabile che il problema riguardi il comando `configure`, per cui possiamo leggere il file `config.log` alla ricerca della possibile soluzione del problema. Prima di riprovare la "danza" dei comandi, assicuriamoci di eseguire il comando `make clean`.

In Windows le cose sono diverse. Innanzitutto dobbiamo accertarci che sul sistema sia installato Microsoft Visual C++ 2008. Quindi dobbiamo aprire il `workspace pcbuild.sln` presente nella directory `PCBuild`. Solo a questo punto possiamo lanciare il comando per la creazione degli eseguibili.



Se, durante la compilazione dei file di codice sorgente, ci siamo scontrati con qualche problema insormontabile, un'insostituibile fonte di informazioni è <http://groups.google.com>. Aggiungendo alla ricerca la stringa `group:comp.lang.python`, saremo sicuri di trovare informazioni relative al nostro linguaggio preferito.