



Introduzione

Python

- Python è stato sviluppato intorno al 1991 da Guido van Rossum che ne ha derivato semplicità di sintassi e facilità d'uso in gran parte da ABC, un linguaggio dedicato all'insegnamento sviluppato negli anni '80.
- Oltre che per questo specifico contesto, Python è stato creato per risolvere problemi reali, dimostrando di possedere un'ampia varietà di caratteristiche tipiche di linguaggi di programmazione quali C++, Java, Modula-3 e Scheme.
- "Perché Python?": Python permette un ottimo equilibrio tra l'aspetto pratico e quello concettuale



Riferimenti

- Il sito principale del progetto: www.python.org
- Il sito ufficiale della community italiana Python: www.python.it
- Diversi tutorial. Ad esempio:
 - Per non-programmatori:
<https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>
 - Per programmatori:
<https://wiki.python.org/moin/BeginnersGuide/Programmers>
- E tanti libri....



“A short explanation of what Python is”

- Python is a clear and powerful object-oriented programming language, comparable to Perl, Ruby, Scheme, or Java.
- Some of Python's notable features:
 - Uses an elegant syntax, making the programs you write easier to read.
 - Is an easy-to-use language that makes it simple to get your program working. This makes Python ideal for prototype development and other ad-hoc programming tasks, without compromising maintainability.
 - Comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, reading and modifying files.
 - Python's interactive mode makes it easy to test short snippets of code. There's also a bundled development environment called IDLE.
 - Is easily extended by adding new modules implemented in a compiled language such as C or C++.
 - Can also be embedded into an application to provide a programmable interface.
 - Runs on many different computers and operating systems: Windows, MacOS, many brands of Unix, OS/2, ...
 - Is free software in two senses. It doesn't cost anything to download or use Python, or to include it in your application. Python can also be freely modified and re-distributed, because while the language is copyrighted it's available under [an open source license](#).

“A short explanation of what Python is”

- Some programming-language features of Python are:
 - A variety of basic data types are available: numbers (floating point, complex, and unlimited-length long integers), strings (both ASCII and Unicode), lists, and dictionaries.
 - Python supports object-oriented programming with classes and multiple inheritance.
 - Code can be grouped into modules and packages.
 - The language supports raising and catching exceptions, resulting in cleaner error handling.
 - Data types are strongly and dynamically typed. Mixing incompatible types (e.g. attempting to add a string and a number) causes an exception to be raised, so errors are caught sooner.
 - Python contains advanced programming features such as generators and list comprehensions.
 - Python's automatic memory management frees you from having to manually allocate and free memory in your code.

Installazione

- Python è un linguaggio interpretato e per funzionare è necessario installare l'interprete Python.
- GNU/Linux
 - l'interprete Python è generalmente preinstallato nelle distribuzioni recenti
 - dalla shell, si esegue l'interprete Python digitando il comando:
Python
 - nel caso non sia installato, può essere installato da sorgenti o tramite un gestore di pacchetti (a seconda della distribuzione GNU/Linux).
- Mac Os X
 - L'interprete Python è preinstallato.
 - dalla shell, si esegue l'interprete Python digitando il comando:
Python
 - Nel caso non sia installato, scaricate il programma di installazione da <http://www.python.org/download/>

Installazione

- Microsoft Windows
 - Windows non fornisce alcun interprete Python preinstallato.
 - Scaricate il programma di installazione da <http://www.python.org/download/>
 - Avviate l'interprete Python selezionandolo dal menu dei programmi

Interprete

- Ci sono due modi di usare l'interprete:
 - a linea di comando o modalità interattiva
 - modalità script: (estensione .py)
- L'interprete di comandi Python permette di eseguire interattivamente comandi Python e visualizzarne il risultato
- Una volta avviato, l'interprete presenta un cursore in cui inserire comandi
- Scrivendo un comando Python e premendo invio, l'interprete esegue il comando e ne visualizza il risultato
- In pratica l'interprete stampa su schermo una stringa che rappresenta l'output del comando inserito

Interprete - modalità interattiva

```
giuseppe$ python
Python 2.7.6 (v2.7.6:3a1db0d2747e, Nov 10 2013, 00:42:54)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more
information.
>>> 7 * 8
56
>>>
```

- L'uso interattivo dell'interprete permette di verificare velocemente l'output di un comando
 - E' un modo rapido ed efficace di testare codice Python
 - E' un modo rapido di accedere alla documentazione di comandi e funzioni
 - E' un modo rapido di verificare il contenuto di librerie di codice Python
- L'interprete ricorda (history) i comandi inseriti durante una sessione (fino alla chiusura dell'interprete), e possono essere recuperati per rieseguirli

Interprete – modalità script

- Le funzioni di editing di riga dell'interprete di solito non sono molto sofisticate. L'interprete opera all'incirca come una shell Unix: quando viene lanciato con lo standard input connesso ad un terminale legge ed esegue interattivamente dei comandi.
- Quando viene invocato con il nome di un file come argomento o con un file come standard input legge ed esegue uno script da quel file.
- A differenza dell'uso interattivo, in questo caso l'interprete non stampa automaticamente l'output dei comandi (usare esplicitamente `print` a questo scopo)
- Quando noti all'interprete, il nome dello script e gli argomenti addizionali sono passati allo script tramite la variabile `sys.argv`, che è una lista di stringhe. La sua lunghezza minima è uno. Quando non vengono forniti né gli script né argomenti, `sys.argv[0]` è una stringa vuota.

Interprete – modalità script

- Esempio: programma che stampa il risultato della moltiplicazione $7 * 8$.

- **EDITING**: Si crea, con un qualunque editor di testo, un file che chiamiamo, per esempio, prova.py e che contiene il seguente codice sorgente:

```
print 7 * 8
```

- **ESECUZIONE**: esecuzione del file tramite l'interprete dalla shell del sistema operativo

```
$giuseppe python prova.py
```

```
56
```