

Usare Python

Informatica@DSS 2019/2020 — Il canale

Massimo Lauria <massimo.lauria@uniroma1.it>
<http://massimolauria.net/courses/informatica2019/>

Contenuto di queste slide

- ▶ descrizione dell'ambiente di lavoro in laboratorio
- ▶ suggerimenti per l'ambiente di lavoro a casa
- ▶ puntatori a risorse e strumenti aggiuntivi

Lavorare in laboratorio



Sessione interattiva: terminale

(Dal menù) *Strumenti di sistema* → *LXTerminal*

Piccoli
esperimenti iniziali

```
massimo@lauriabox:~$ python3
Python 3.7.3 (default, Jun 17 2019, 12:55:33)
[GCC 7.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 7
>>> x - 2
5
>>> - 3*x
-21
>>> 2*y
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'y' is not defined
>>> 'Ciao a tutti!'
'Ciao a tutti!'
>>> 'Ciao a' + ' ' + 'tutti!'
'Ciao a tutti!'
>>> 'Ciao' + 3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
>>> █
```

Scrittura di programmi: terminale + editor

(Dal menù) *Programmazione* → *Geany*

- ▶ Scrittura di programmi
- ▶ Esecuzione da terminale
- ▶ Geany ha un terminale integrato
- ▶ Sintassi colorata

The screenshot displays the Geany IDE interface. The main editor window shows a file named `primoprogramma.py` with the following Python code:

```
1
2 x = 7
3
4 print(x - 7)
5
6 print('Ciao a' + ' ' + 'tutti!')
7
8
```

The code is color-coded: `x` is blue, `7` is green, `print` is purple, and the string literals are orange. Below the editor, the integrated terminal shows the execution of the script:

```
studente@debian9:~$ python3 primoprogramma.py
0
Ciao a tutti!
studente@debian9:~$
```

The terminal output shows the result of the program's execution, which is `0` followed by the string `Ciao a tutti!`.

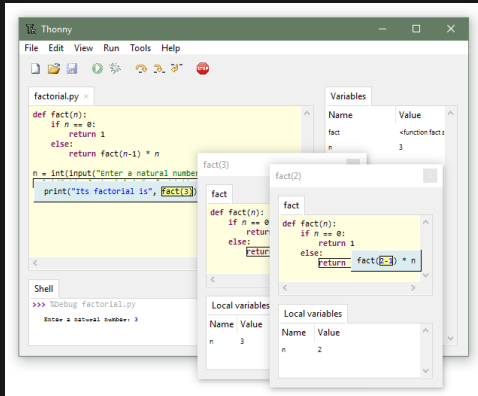
Scrittura/Debug/Interazione: Thonny

```
$ pip3 install thonny
```

(Installazione per utente)

```
$ python3 -m thonny
```

(Esecuzione)



<http://thonny.org>

Lavorare a casa



Impostare un ambiente di lavoro

Allen B. Downey, autore del libro di testo *Pensare in Python* ha un tutorial che aiuta ad iniziare a lavorare autonomamente.

<link al tutorial>

Purtroppo il tutorial è in inglese, ma un inglese molto semplice.

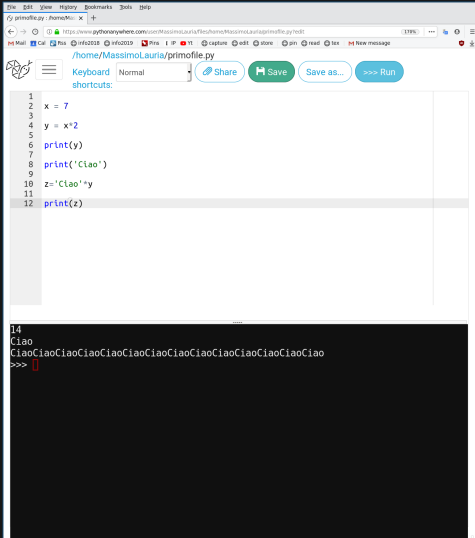
Lavorare online: Python anywhere

Python Anywhere è un ambiente python **online** che permette di lavorare in python **nel browser**, senza installare nulla sul proprio computer.

`https://www.pythonanywhere.com/`

- ▶ Serve solo il browser web
- ▶ Necessario creare un account
- ▶ Account gratuito più che sufficiente

Lavorare online: Python anywhere (II)



Modalità interattiva

- ▶ esattamente come Python da terminale

Modalità scrittura file

- ▶ editor di testo nel browser
- ▶ gestione file

Descritto nel tutorial

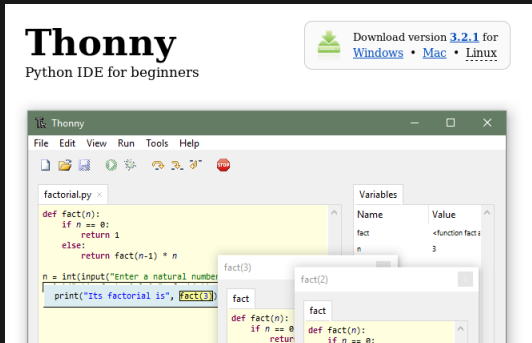
Installazione locale di Python

- ▶ Differenti procedure per Mac/Win/Linux
- ▶ Differenti installazioni/distribuzioni
- ▶ Differenti editor di testo

Aiutatevi tra voi, magari in gruppi di 4-5.

Thonny — <https://thonny.org>

- ▶ semplice da usare
- ▶ non richiede di installare Python3 a parte
- ▶ lo usiamo a lezione



Anaconda — <https://www.anaconda.com/>

- ▶ contiene funzionalità aggiuntive (troppe?)
- ▶ professionale
- ▶ istruzioni nel tutorial indicato precedentemente

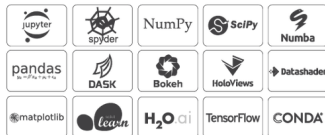
Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source **Anaconda Distribution** is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with **Conda**
- Develop and train machine learning and deep learning models with **scikit-learn**, **TensorFlow**, and **Theano**
- Analyze data with scalability and performance with **Dask**, **NumPy**, **pandas**, and **Numba**
- Visualize results with **Matplotlib**, **Bokeh**, **Datashader**, and **Holoviews**



Altro materiale utile

Python tutor — <http://pythontutor.com/>

- ▶ evoluzione delle variabili
- ▶ osservare i singoli passi di esecuzione
- ▶ andare avanti e indietro

Get live help!

Start private chat

(warning: chat service may crash at any time)

These Python Tutor users are asking for help right now. Please volunteer to help!

user_c9d from Petaling Jaya, Malaysia needs help with Python3 - 3 people chatting - [click to help](#) (active a minute ago, requested an hour ago)

user_91f from Singapore, Singapore needs help with Python3 - [click to help](#) (active a few seconds ago, requested a few seconds ago)

user_016 from Tallinn, Estonia needs help with Python3 - [click to help](#) (idle: last active 11 minutes ago, requested 11 minutes ago)

Python 3.6

```
1
2 A = [7, 'gatto', 3.5, 'cane', 12, 0.3]
3 B = []
4 while len(A)>0:
5     y=A.pop()
6     B.append(y)
```

[Edit this code](#)

→ line that has just executed
→ next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.

Step 10 of 21

Frames

Global frame

A	→	list
B	→	list
y	→	12

Objects

list	0	1	2	3
	7	"gatto"	3.5	"cane"

list	0	1
	0.3	12

Documentazione standard

`https://docs.python.org/3/`

- ▶ molto ricca e dettagliata
- ▶ richiede un po' di esperienza
- ▶ in inglese

Bibliografia web

Libro di testo:

https://github.com/AllenDowney/ThinkPythonItalian/raw/master/thinkpython_italian.pdf

Il linguaggio Python

- ▶ Pagina principale: <https://www.python.org/>
- ▶ Documentazione ufficiale Python: <https://docs.python.org/3/>

Ambienti di lavoro

- ▶ Tutorial: <http://www.allendowney.com/wp/books/think-python-2e/>
- ▶ Python Anywhere: <https://www.pythonanywhere.com/>
- ▶ Thonny: <https://thonny.org/>
- ▶ Anaconda: <https://www.anaconda.com/>

Altre risorse

- ▶ Python Tutor: <http://pythontutor.com/>
- ▶ Tutorial uso del terminale: https://tutorial.djangogirls.org/it/intro_to_command_line/