



UNIVERSITÀ
di **VERONA**

Linguaggio di Programmazione Python

28/03/2022

Giulio Mazzi

giulio.mazzi@univr.it



Orari e Contatti

Lezioni:

- Aula Alfa (Ca' Vignal 2) + zoom (<https://univr.zoom.us/j/82177234592?pwd=Y015ZU1TNUFzUDZKYkwyGJoY2FtQT09>).
- Le lezioni saranno registrate.
- Lunedì dalle 11:30 alle 13:30.
- Martedì dalle 15:30 alle 17:30.

Sono a disposizione per dubbi, chiarimenti e domande.

Date e orari possono essere concordati via e-mail
(giulio.mazzi@univr.it)

Programma delle lezioni – Linguaggio

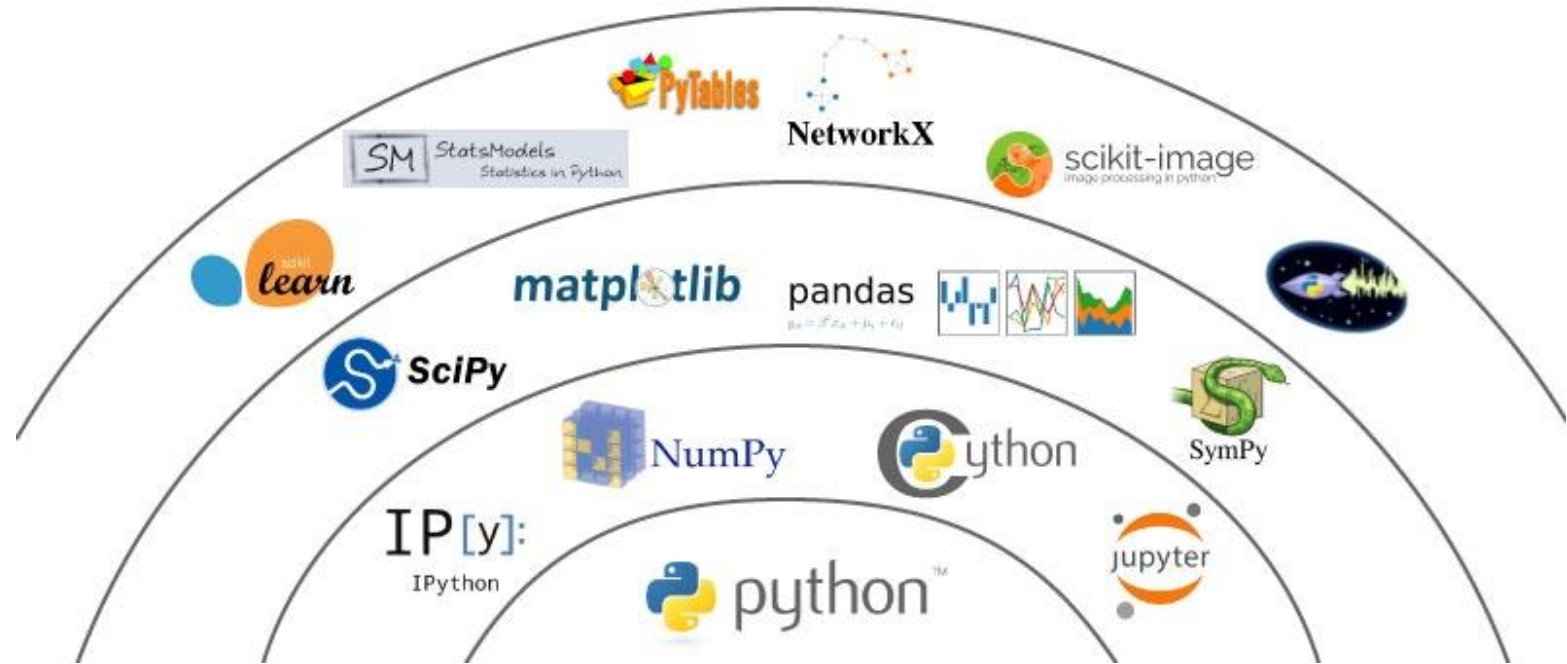
- Introduzione al Linguaggio Python
- Sviluppare in un linguaggio interpretato: ambienti e strumenti
- Cenni storici
- Il linguaggio:
 - Variabili, Espressioni ed Istruzioni
 - Funzioni
 - Istruzioni condizionali e Cicli
 - Manipolazione di Stringhe
 - Liste, Dizionari, Tuple
 - Gestione dei File



Programma delle lezioni - Librerie

Uso di Librerie avanzate:

- Numpy: computazione efficiente con matrici
- Matplotlib: plot e grafici
- Pandas: data analysis e data science
- NetworkX: algoritmi su grafi e networks
- BeautifulSoup: web scraping e manipolazione dati web



Modalità d'esame

- L'esame sarà svolto al PC.
- NIENTE VOTO! → Idonei/non idonei
- 3 esercizi in totale, bisogna risolverne correttamente almeno due per passare l'esame

Materiale di riferimento

Testo di riferimento: Think Python (2nd ed.), Allen Downey

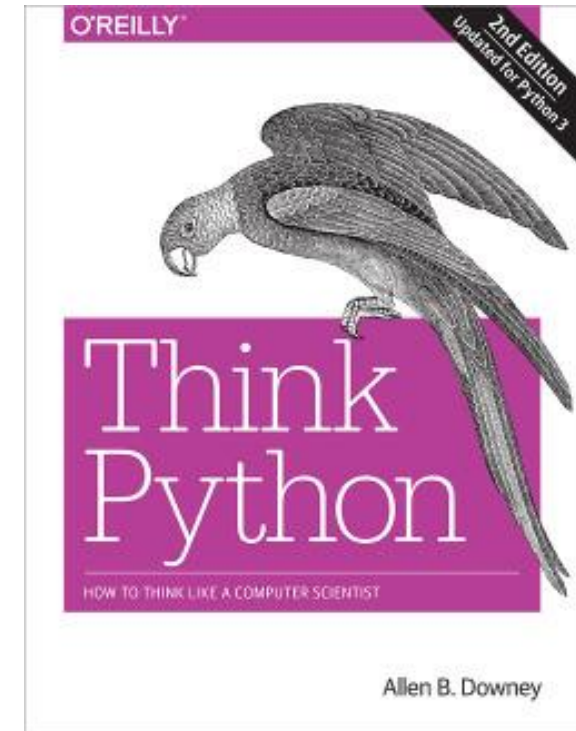
<https://greenteapress.com/wp/think-python-2e>

Documentazione ufficiale:

<https://docs.python.org/3/download.html>

Altro materiale utile:

- Automate the Boring Stuff with Python, Al Sweigart (libro gratuito con casi d'uso interessante)
- Esercizi utili:
 - LeetCode (<https://leetcode.com/>)
 - Advent of Code (<https://adventofcode.com/>)
 - Hacker Rank (<https://www.hackerrank.com>)





Perché python?

- Python è un linguaggio di scripting di alto livello.
- General purpose, sintassi snella, interpretato.
- È il linguaggio "più popolare" al mondo (marzo 2022):
 - TIOBE <https://www.tiobe.com/tiobe-index/>
 - PYPL <https://pypl.github.io/PYPL.html>
- Inventato da Guido van Rossum (https://en.wikipedia.org/wiki/Guido_van_Rossum) negli anni 80.

Caratteristiche del linguaggio



- **Linguaggio minimalista**: le parentesi vengono usate raramente, l'indentazione ha significato semantico
- **Facile**: sintassi vicina alla lingua inglese parlata, uso minimo di *boilerplate*
- **Free and Open source**
- **Alto livello**: I dettagli implementativi sono poco rilevanti. La memoria è gestita in automatico (garbage collection)
- **Portable**: usa bytecode indipendente dalla CPU
- **Multi-Paradigma**: imperativo, funzionale, a oggetti...
- **Duck Typing**: il sistema di tipaggio è dinamico
- **Extensible**: facile integrare codice scritto in c (o altro)
- **Embeddable**: facile da integrare in altri software
- **Batterie incluse**: grande libreria standard + enorme ecosistema di librerie di terze parti
- **Buone prestazioni**: sfrutta codice scritto in c

Linguaggi compilati vs. interpretati

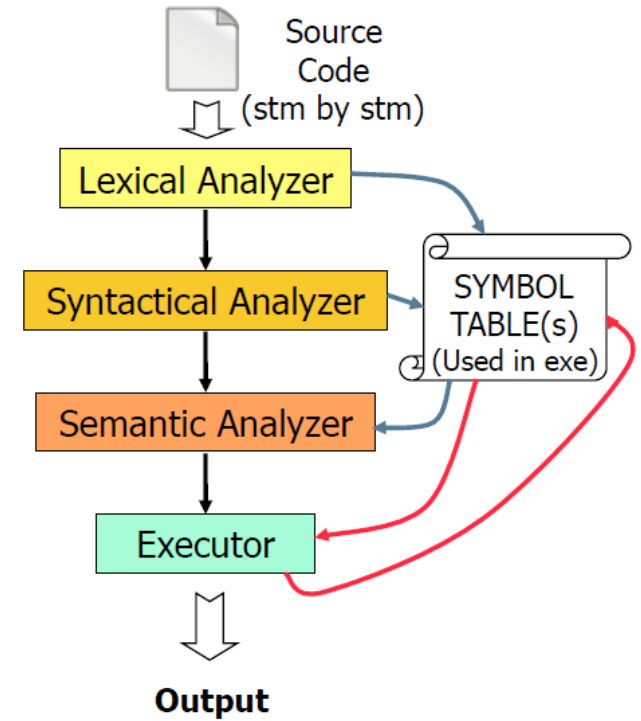
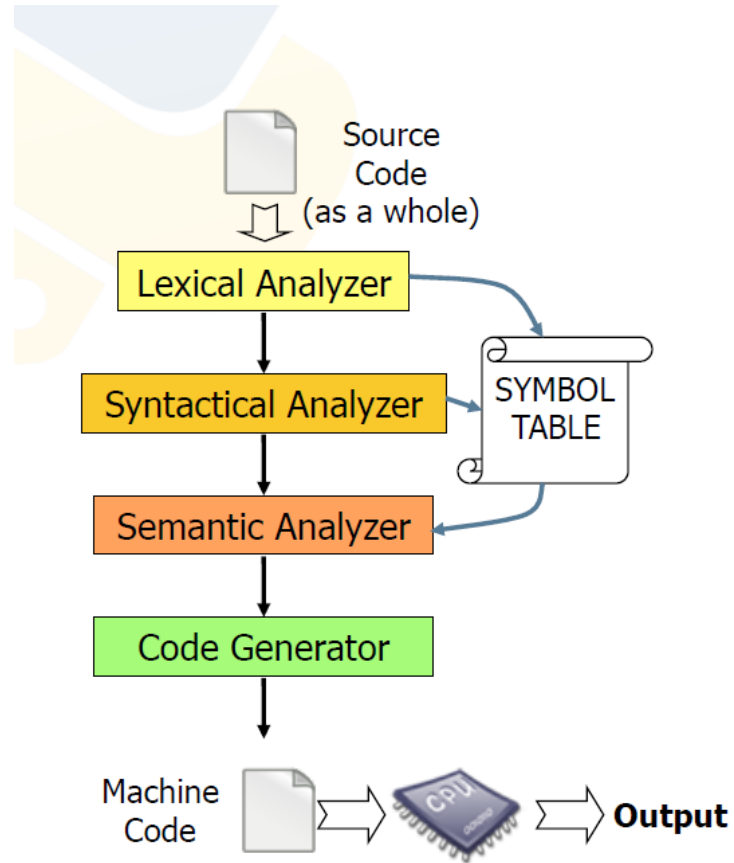
- L'esecuzione di un programma di alto livello si può ottenere attraverso la compilazione o l'interpretazione (oppure usando un linguaggio intermedio)
- Compilazione: Il codice ad alto livello viene processato e trasformato in codice macchina (in genere, avendo come target una specifica CPU). Il programma che esegue questa trasformazione si chiama compilatore



- Interpretazione: Le istruzioni ad alto livello vengono lette ed eseguite una alla volta da un interpreter



Linguaggi compilati vs. interpretati



Esempio: compilazione vs interpretazione

La differenza tra linguaggi compilati e interpretati è particolarmente significativa in caso di errori, vediamo un esempio:

C

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Ciao Mondo");
5     printf("%d", 3);
6     printf(a);
7     return 0;
8 }
```

Python

```
1 print("Ciao Mondo")
2 print(3)
3 print(a)
```

Dove avviene l'errore?

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Ciao Mondo");
5     printf("%d", 3);
6     printf(a);
7     return 0;
8 }
```



```
$ gcc esempio.c
esempio.c: In function 'main':
esempio.c:6:12: error: 'a' undeclared (first use in this function)
    6 |     printf(a);
      |           ^
```

```
1 print("Ciao Mondo")
2 print(3)
3 print(a)
```



```
$ python3 esempio.py
Ciao Mondo
3
Traceback (most recent call last):
  File "esempio.py", line 3, in <module>
    print(a)
NameError: name 'a' is not defined
```

In python
eseguo le due
istruzioni prima
dell'errore!

Strumenti & ambienti di sviluppo

shell



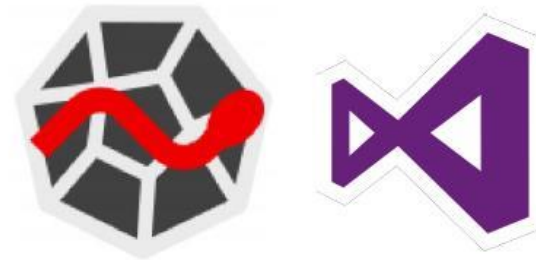
editor



notebook



ide



Eseguire uno script

Ci sono molti modi di eseguire codice python:

- Usare direttamente l'interprete:

```
$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("ciao")
ciao
>>> a = 1 + 2
>>> b = a * 7
>>> print(b)
21
```

- Eseguire uno script:

```
1 print('ciao')
2 a = 1 + 2
3 b = a * 7
4 print(b)
```

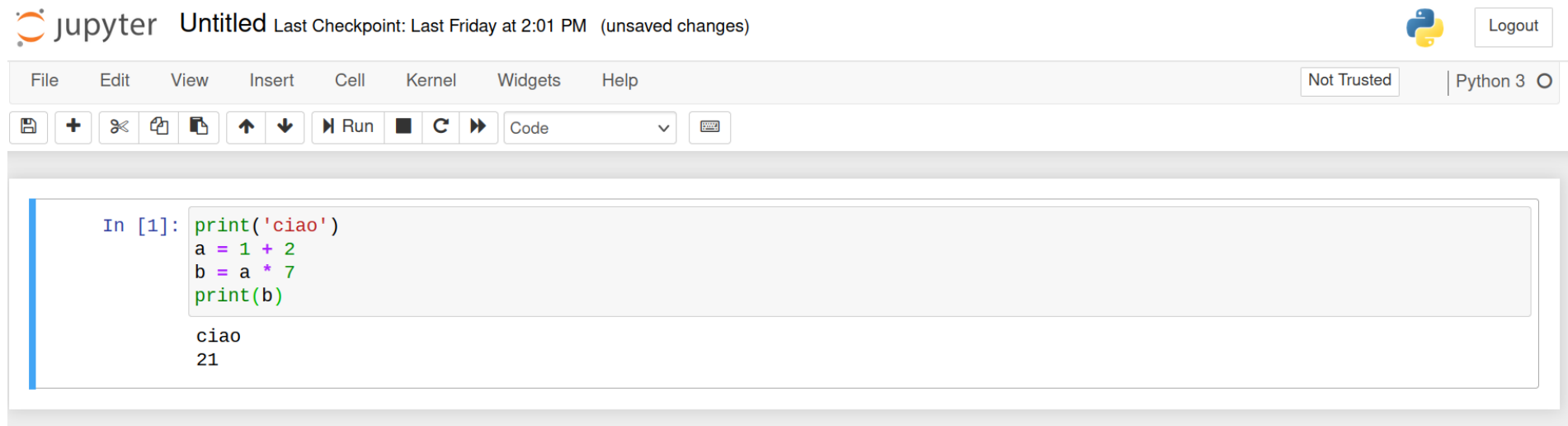
test.py



```
$ python3 test.py
ciao
21
```


Eseguire uno script - Notebook

Jupyter notebook ci permetterà di mescolare codice e testo, utile per analizzare dati e presentare risultati



Potete installarlo in locale o provarlo online: <https://jupyter.org/try>

Python 2 vs 3

In questo corso useremo **python 3**. Ci sono differenze sintattiche e semantiche tra le due versioni. Python 2 non è più supportato ufficialmente, ma il passaggio tra le due versioni è stato travagliato (potreste incontrare ancora del python 2!)

```
$ python2
Python 2.7.18 (default, Mar  8 2021, 13:02:45)
[GCC 9.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Ciao mondo"
Ciao mondo
>>> 5/2
2
```

```
$ python3
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Ciao Mondo")
Ciao Mondo
>>> 5/2
2.5
>>> █
```

Un linguaggio, tanti interpreti

Sono disponibili diversi interpreter:

- CPython, implementazione di riferimento scritto in c
- Jython, java
- IronPython, c#, .net
- PyPy, scritto in python!
- Cython, compilatore che mischia c e python
- ...