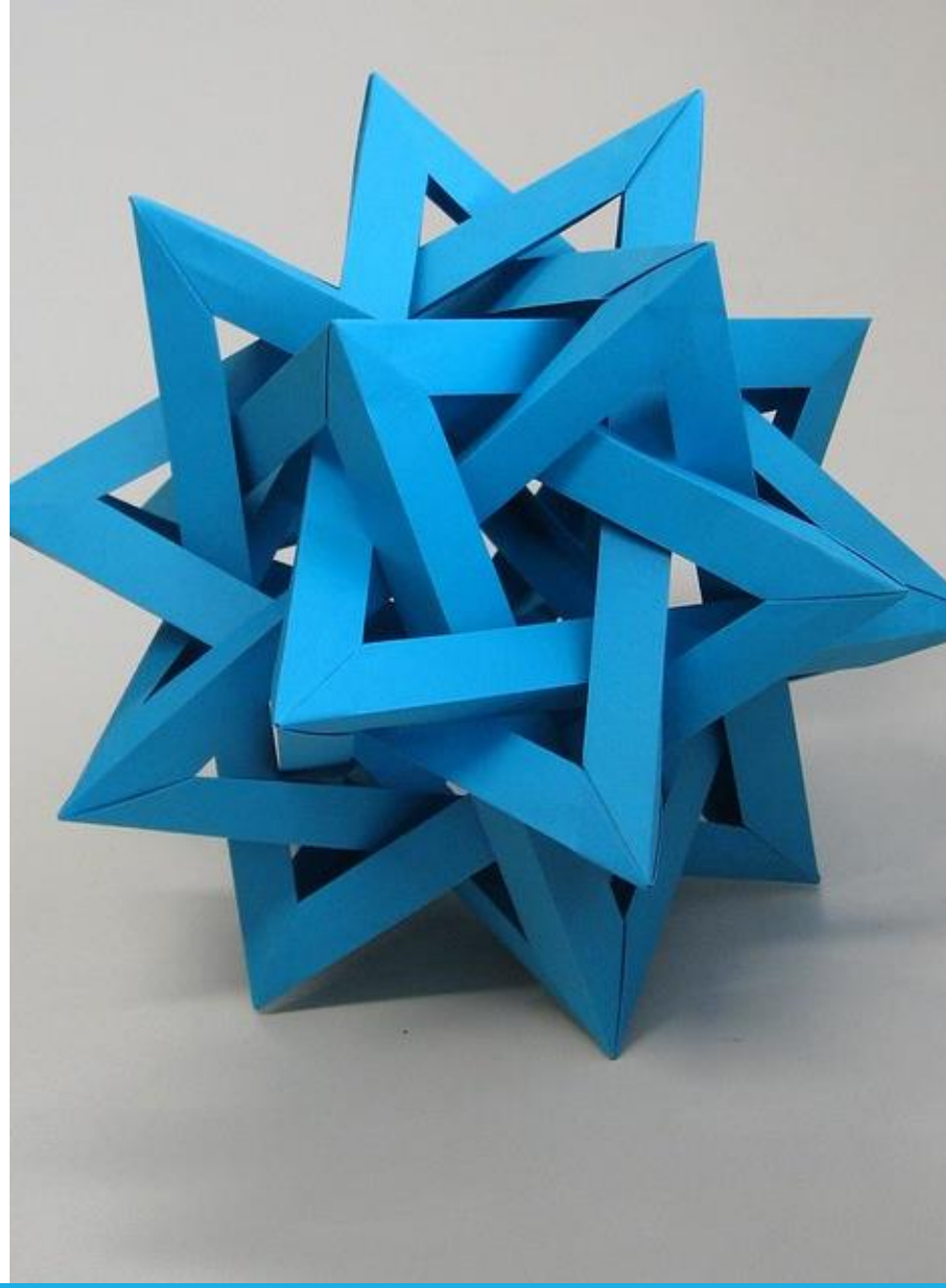
Politecnico di Torino
Dipartimento di Automatica e Informatica

14BHDxx Informatica

ING. INF/ELT/ENE/MTM/ELN/FIS/CIN – CORSO #9 (FAJ-KOR)

PROF. FULVIO CORNO



This Photo by Unknown Author is licensed under CC BY-SA

Welcome

Corso di **Informatica** (14BHD)

- Insegnamento obbligatorio, 8 crediti, 1° Semestre
- Corso n. 9
  - Studenti con cognomi compresi tra FAJ e KOR
  - Iscritti ai corsi di ing. Informatica, Elettrica, Energetica, Matematica, Elettronica, Fisica, Cinema
- Docenti:
  - **Fulvio Corno**
    (lezioni, esercitazioni in aula)
  - **Roberta Bardini**
    (lezioni, esercitazioni in aula)
  - **Lorenzo Martini**
    (esercitazioni in laboratorio)

Benvenuti ad Ingegneria

- Questo è il primo corso di Ingegneria che affronterete
- Ingegneria =
  - Saper progettare
  - Risolvere problemi
  - Trovare soluzioni
  - Soddisfare le specifiche
  - Nel rispetto dei vincoli
  - Con gli strumenti disponibili

La difficoltà maggiore è acquisire **la forma mentis** adatta all'approccio ingegneristico-progettuale

Benvenuti ad Ingegneria

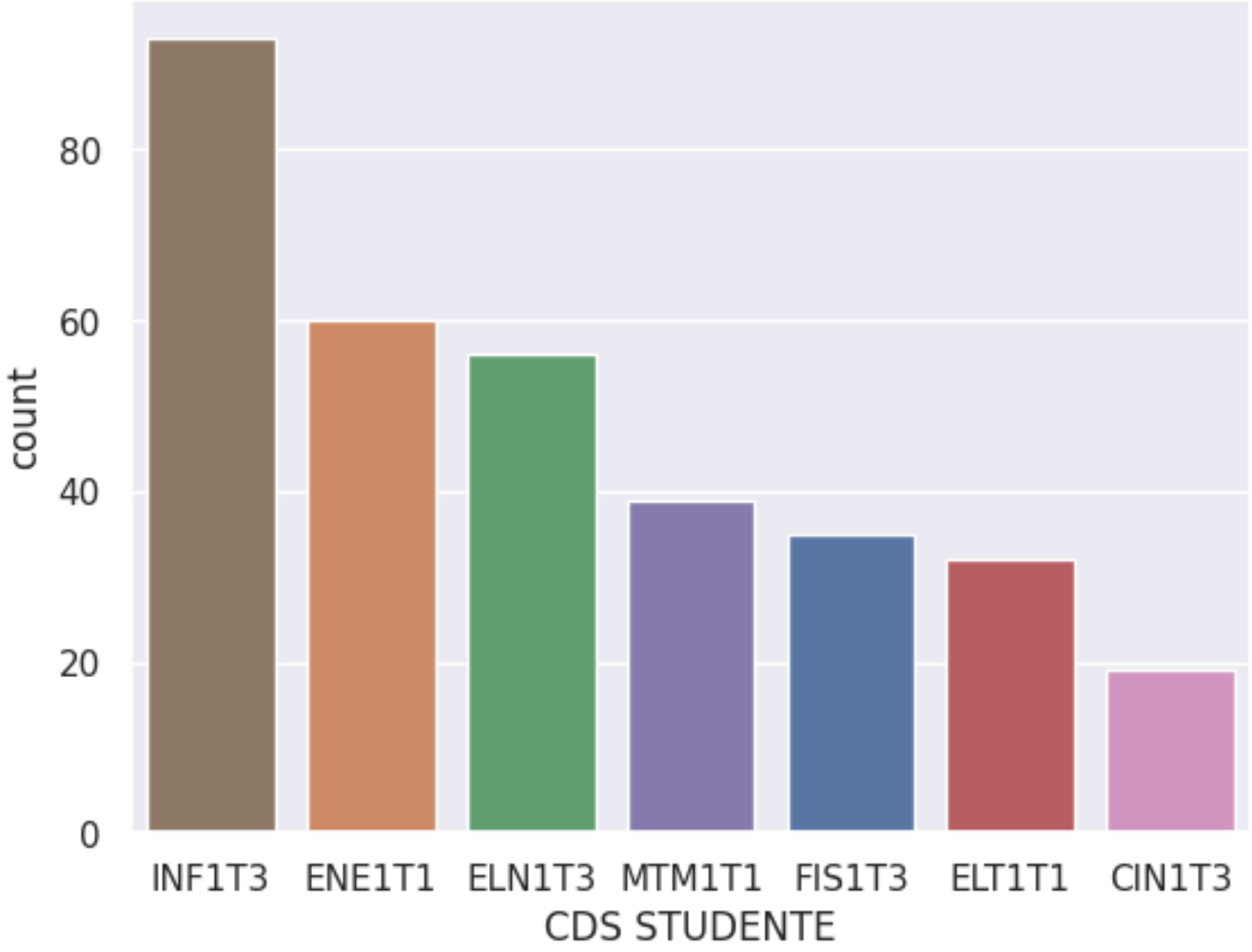
- Questo è il primo corso di Ingegneria che affronterete
- Ingegneria =
  - Saper progettare
  - Risolvere problemi
  - Trovare soluzioni
  - Soddisfare le specifiche
  - Nel rispetto dei vincoli
  - Con gli strumenti disponibili

La difficoltà maggiore è acquisire **la forma mentis** adatta all'approccio ingegneristico-progettuale

- Ingegneria informatica =
  - Problemi di ogni genere (calcolo, gestione dati, interazione, ...)
  - Lo strumento è il calcolatore

Statistiche studenti

```
import pandas as pd
import seaborn as sns
studenti = pd.read_csv('14BHDOA_2024.csv')
sns.countplot(data=studenti, x='CDS STUDENTE')
```



Linguaggi di programmazione

- In questo corso **si parte da zero**, non è richiesto conoscere già un linguaggio
- Le capacità di **problem solving** si trasferiscono facilmente da un linguaggio all'altro
- ⚠️ Talvolta, dovrete dis-imparare alcune cattive abitudini apprese in precedenza

Quale linguaggio di programmazione conosci e/o hai già utilizzato in passato?

Final results

Quale linguaggio di programmazione conosci e/o hai già utilizzato in passato?

Final results

Programma del corso

Link utili

- Sito del corso (ufficiale):
  - http://elite.polito.it/ → Teaching → Informatica (14BHD)
  - Link breve: http://bit.ly/polito-informatica
- Materiale, laboratori, esercizi
  - https://github.com/polito-informatica
- Gruppo Telegram
  - https://t.me/politoinfo2024
- Canale Telegram per avvisi e tutorati
  - https://t.me/+QttiF2nn0hFiNDg0

@POLITOINFO2024

Programma dell'insegnamento

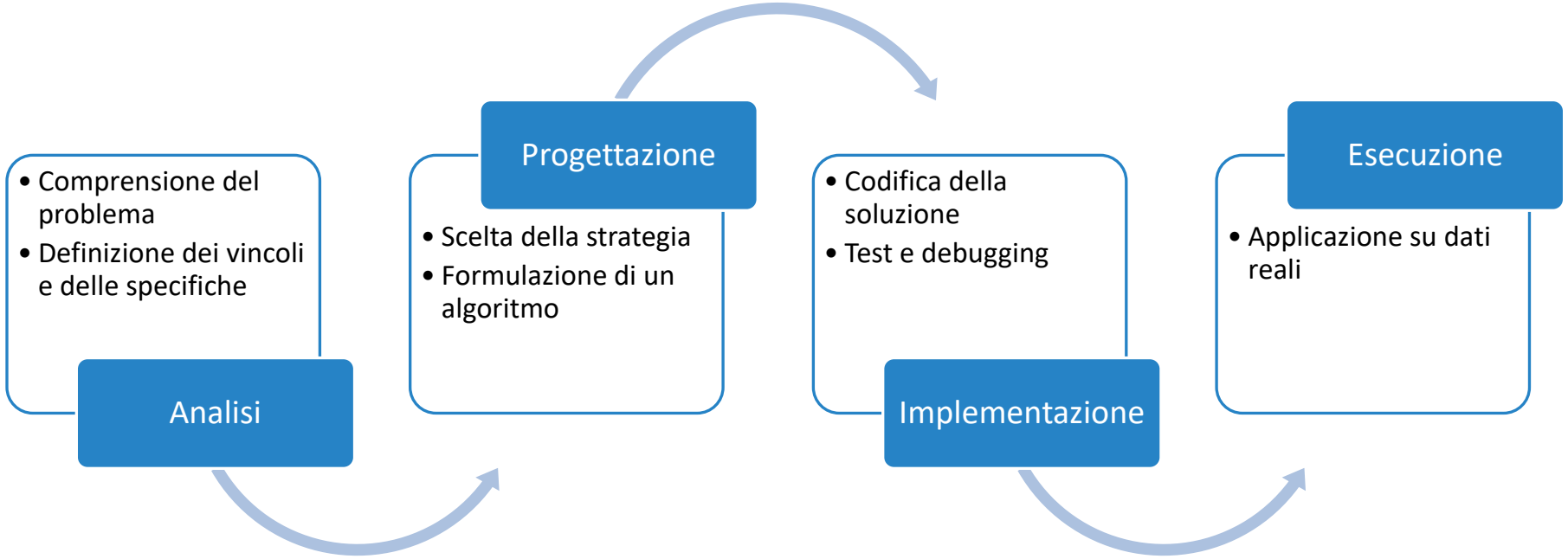
- Metodologie di *Problem Posing and Solving* (PPS)
  - Tecniche di analisi basate su flow-chart e pseudo-code
  - Implementazione attraverso programmi informatici
- Strutture dati e rappresentazione dell'informazione nel PPS
  - Numeri, Stringhe, Vettori, Sequenze, Liste, Insiemi, Dizionari, …
- Linguaggio di programmazione Python
  - Maggior semplicità sintattica e maggior potenza espressiva
  - Possibilità di affrontare esercizi con uno scopo applicativo più diretto
  - Ambiente di lavoro adeguato ai sistemi operativi moderni
  - Disponibilità di numerose librerie adatte a diversi campi applicativi (che potranno essere introdotte negli insegnamenti successivi).

Contenuti

- Teoria (9h)
  - Cenni di Informatica generale ed impatti dell'informatica e del digitale
  - Struttura ed architettura del calcolatore, linguaggi, applicazioni
  - Rappresentazione dell'Informazione
- Problem Solving (12h)
  - Approccio alla logica dei problemi
  - I passaggi del processo di Problem Solving
  - Tipologie di problemi e di approcci risolutivi
- Programmazione Python (41h)
  - Numeri e Stringhe
  - Decisioni
  - Cicli
  - Liste (vettori)
  - Insiemi e dizionari (array associativi)
  - File
  - Funzioni
- Laboratori (12 x 1,5 = 18h)

...e cioè cosa impariamo a fare?

- Quali sono i nomi più frequenti in quest'aula?

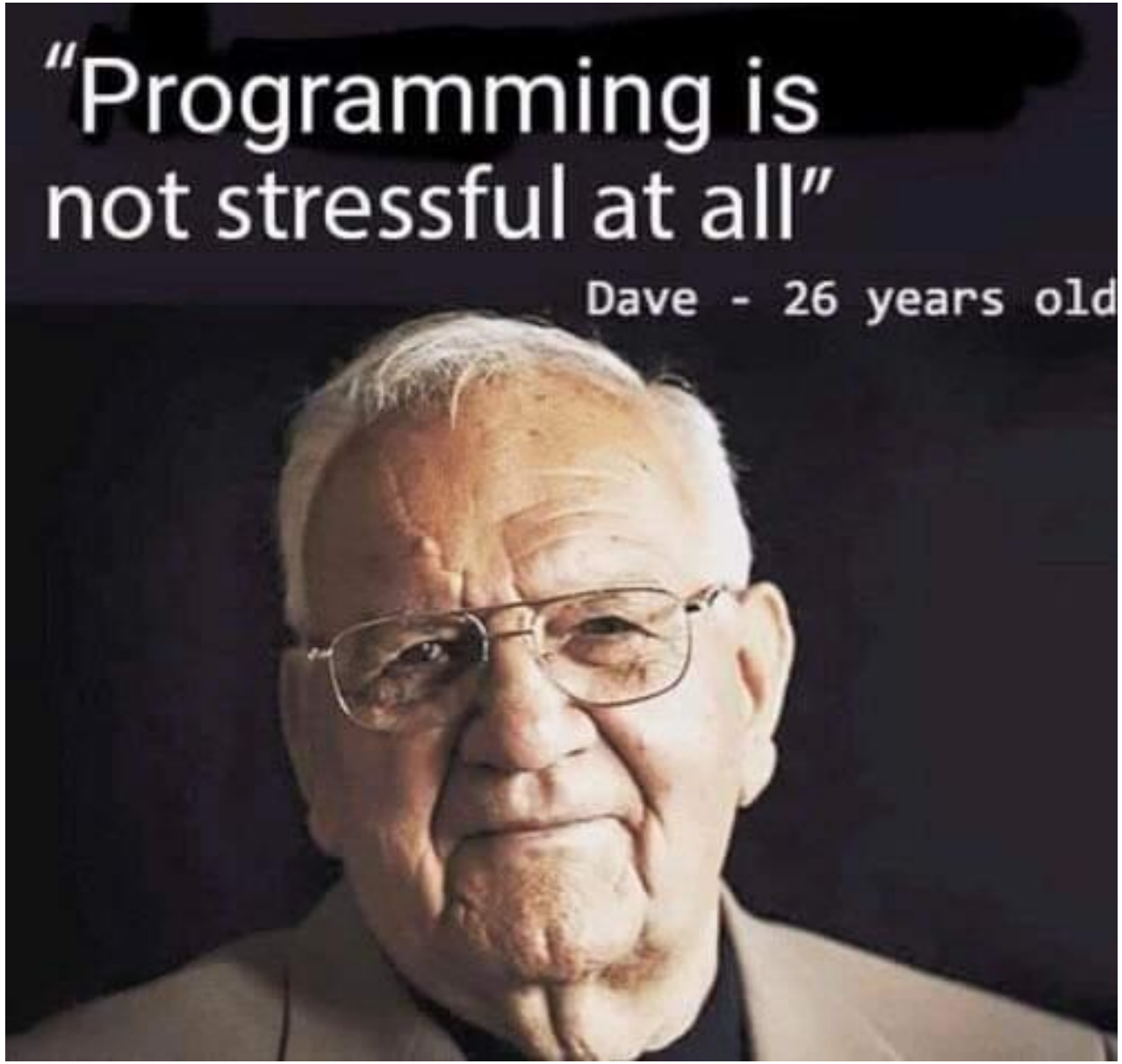


Una possibile soluzione… in Python

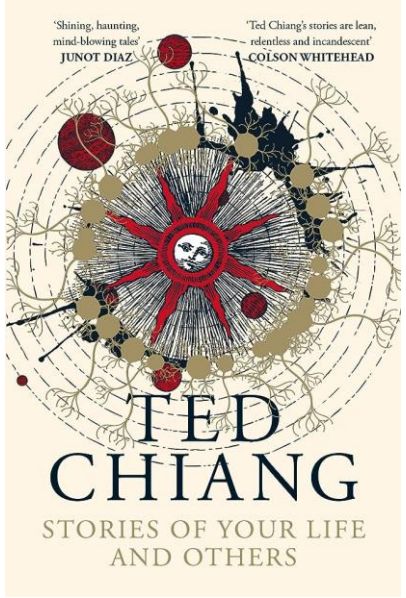
```python
import csv
from matplotlib import pyplot

# Leggi l'elenco degli studenti e salvalo in un'array
def leggi(nome_file):
    file = open(nome_file, 'r')
    reader = csv.reader(file)
    prima = True
    studenti = []
    for line in reader:
        if prima:  # skip first line (headers)
            prima = False
        else:
            studenti.append(line)
    file.close()
    return studenti

# estrai i nomi di battesimo da un elenco di studenti
def estrai_nomi(elenco):
    lista_nomi = []
    for riga in elenco:
        lista_nomi.append(riga[2])
    return lista_nomi

# Calcola le frequenze dei vari nomi presenti in un array
def frequenze(tokens):
    freq = {}
    for token in tokens:
        if token in freq:
            freq[token] = freq[token] + 1
        else:
            freq[token] = 1
    return freq
```

```python
# calcola il massimo valore presente nelle frequenze
def max_frequenza(freq):
    return max(freq.values())


def nomi_piu_frequenti(freq, max):
    return [nome for (nome, frequenza) in freq.items() if frequenza == max]


FILENAME = '01TXYOV_2020.csv'
def main():
    stud = leggi(FILENAME)
    nomi = estrai_nomi(stud)
    print(f"Nella classe ci sono {len(stud)} studenti")
    freq = frequenze(nomi)
    max_freq = max_frequenza(freq)
    print(f"Il nome più frequente compare {max_freq} volte")
    nomi_max = nomi_piu_frequenti(freq, max_freq)
    print(f"Si tratta di : {nomi_max}")
    # estrai solo i nomi che compaiono almeno 3 volte
    freq2 = {k: v for (k, v) in freq.items() if v >= 3}
    print(
        f"I nomi che compaiono più volte sono {', '.join(sorted(list(freq2.keys())))}."
    )

    pyplot.barh(list(freq2.keys()), freq2.values())
    pyplot.show()


main()
```

https://replit.com/@fulcorno/NomiFrequentiStudenti#main.py


"Programming is
not stressful at all"
Dave - 26 years old

A cosa serve imparare a programmare?



Dopo «Informatica»

	I anno	II anno	III anno
Ing. Informatica	Tecniche di Programmazione	Algoritmi e strutture dati Basi di dati Calcolatori elettronici	Sistemi operativi Programmazione a oggetti Reti di calcolatori Controlli automatici Insegnamento a scelta: Analisi e visualizzazione dati; Embedded Systems and IoT; Intro elettronica sistemi embedded; Introduzione alle Applicazioni Web
Ing. Cinema		Algoritmi e programmazione a oggetti	Basi di dati Computer Grafica Reti di calcolatori
Ing. Gestionale		Basi di dati (L8+L9)	Programmazione a oggetti Tecniche di Programmazione Progettazione di applicazioni Internet Reti telematiche e internet (solo L8)
Ing. Elettronica	Algoritmi e programmazione		
E.C.E.	Communication networks		Algorithms and Programming

Dopo «Dopo «Informatica»»

Laurea

- Ing. Informatica
- Ing. Elettronica
- Ing. Gestionale (L8)
- Ing. Cinema e Mezzi di Comunicazione
- Ing. Fisica
- Electronic and Communications Eng.
- Matematica per ing.
  - Corsi a scelta
- Altre Lauree

Senza debiti

Necessaria integrazione

- Percorso integrativo

Laurea Magistrale

- Ing. Informatica Computer Engineering
- Ing. del Cinema e dei Mezzi di Comunicazione
- Data Science and Engineering
- Mechatronic Engineering
- Cybersecurity
- Quantum Engineering

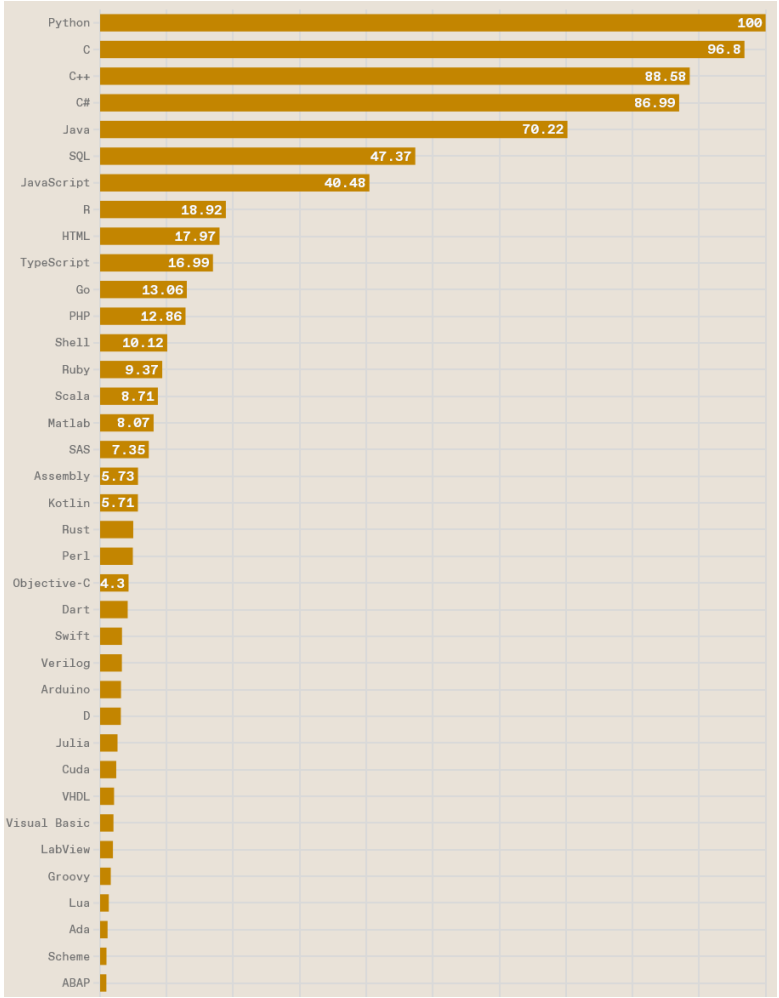
Uno sguardo a Python

VISIONE GENERALE DELL'ECOSISTEMA PYTHON

Il linguaggio Python

- Linguaggio gratuito ed open-source
- Disponibile per tutti i sistemi operativi
  - Windows, Mac OS X, Linux
  - Sistemi embedded, Raspberry PI, Android
- Progettato negli anni ’90 da Guido Van Rossum
  - Sintassi semplice, pulita, regolare
  - Approccio «batterie incluse»
    - Ampia libreria di funzioni standard
  - Basso gradino d’accesso
    - Linguaggio interpretato
- Sterminata documentazione on-line

Diffusione del linguaggio Python



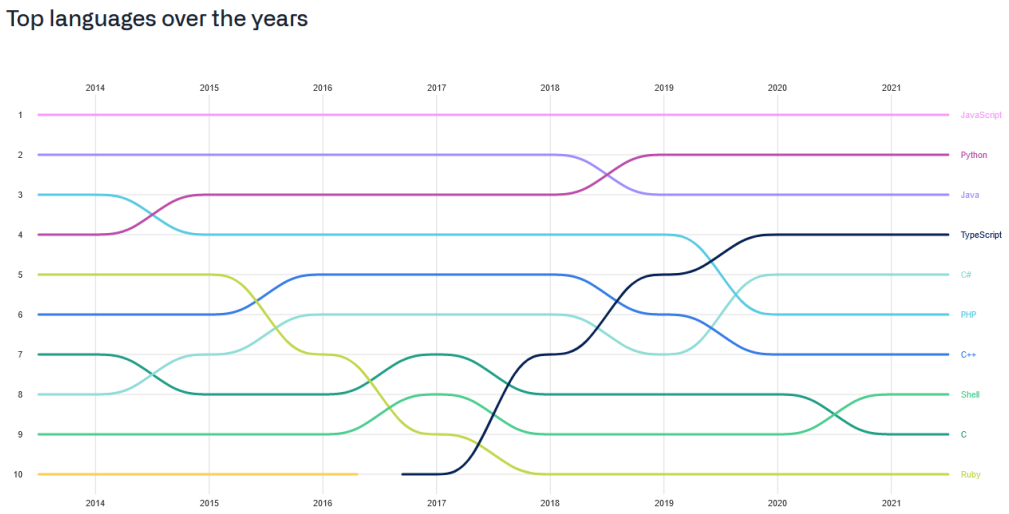
- IEEE Spectrum, 23 Aug 2022 https://spectrum.ieee.org/top-programming-languages-2022
  - Top Programming Languages 2022

- Altre statistiche, per i più curiosi:
  - https://www.tiobe.com/tiobe-index/
  - http://pypl.github.io/PYPL.html
  - https://octoverse.github.com/
  - …

Diffusione del linguaggio Python

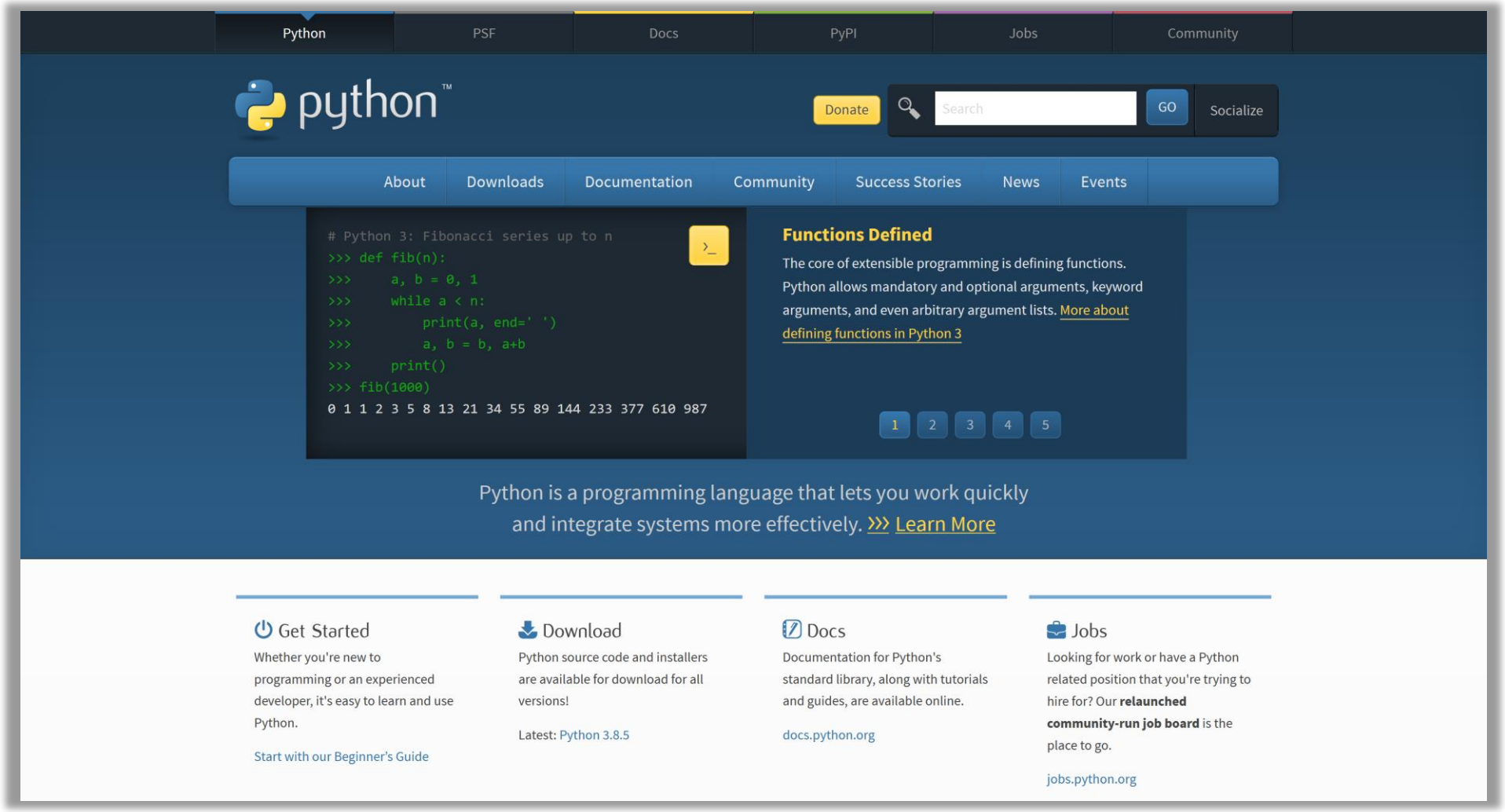
Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	^	Python	15.42%	+3.56%
2	1	v	C	14.59%	+2.03%
3	3		Java	12.40%	+1.96%
4	4		C++	10.17%	+2.81%
5	5		C#	5.59%	+0.45%
6	6		Visual Basic	4.99%	+0.33%
7	7		JavaScript	2.33%	-0.61%
8	9	^	Assembly language	2.17%	+0.14%
9	10	^	SQL	1.70%	+0.23%
10	8	v	PHP	1.39%	-0.80%
11	16	^^	Swift	1.27%	+0.30%
12	12		Classic Visual Basic	1.27%	+0.04%
13	22	^^	Delphi/Object Pascal	1.22%	+0.60%
14	23	^^	Objective-C	1.22%	+0.61%
15	18	^	Go	0.98%	+0.08%
16	14	v	R	0.92%	-0.13%
17	17		MATLAB	0.90%	-0.08%
18	15	v	Ruby	0.82%	-0.18%
19	13	vv	Fortran	0.81%	-0.32%
20	20		Perl	0.72%	-0.06%

https://www.tiobe.com/tiobe-index/

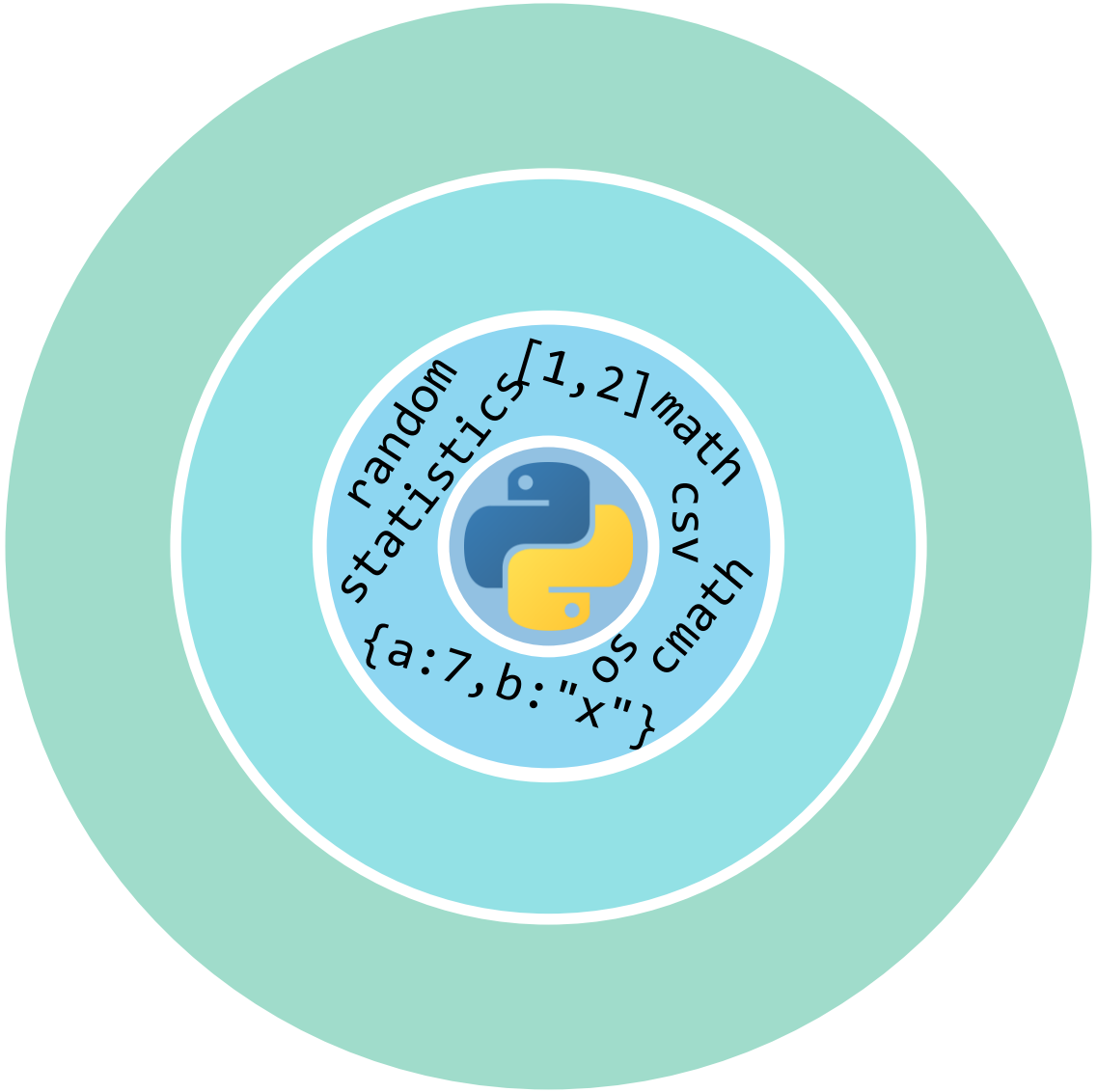


https://octoverse.github.com/

https://www.python.org/



Batterie incluse



- Tipi di dato fondamentali
  - boolean, int, float, complex, string, regexp
- Strutture dati fondamentali
  - liste/array/matrici, tuple, insiemi, dizionari/mappe/hash, file, …
- Orientato agli oggetti
  - Utilizzo semplice e diretto di oggetti predefiniti
  - Possibilità di creare classi ed oggetti personalizzati (avanzato)
- 200+ Moduli nella libreria standard

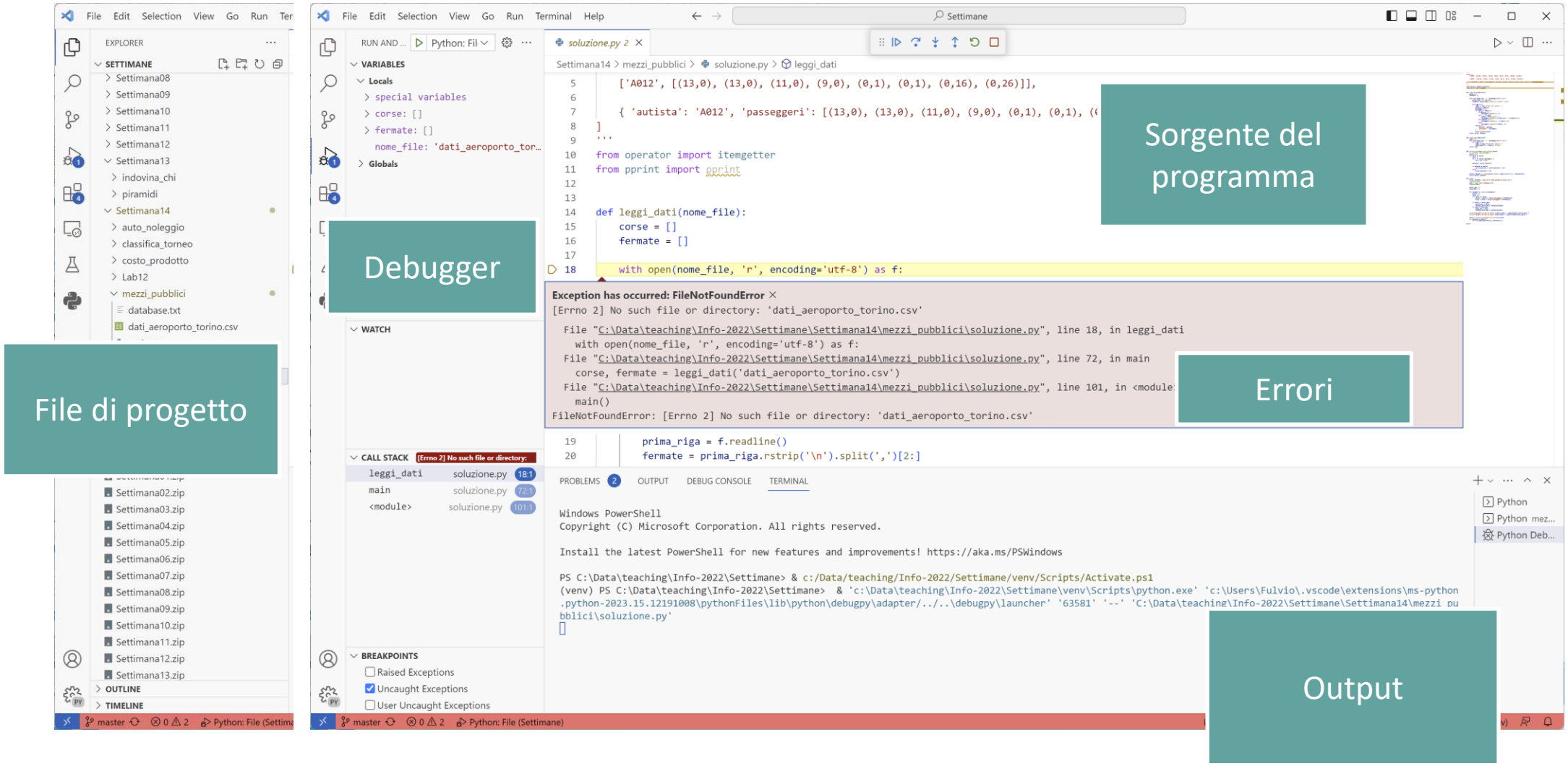
200 Moduli della libreria standard

abc	chunk	**decimal**	getpass	keyword	optparse	**queue**	sndhdr	telnetlib	**unittest**
aifc	**cmath**	difflib	gettext	linecache	**os**	quopri	socket	**tempfile**	urllib
argparse	cmd	dis	glob	locale	ossaudiodev (Linux, FreeBSD)	**random**	socketserver	termios (Unix)	uu
array	**codecs**	distutils	**graphlib**	**logging**	parser	**re**	spwd (Unix)	**test**	**uuid**
ast	codeop	doctest	grp (Unix)	lzma	**pathlib**	readline (Unix)	**sqlite3**	**textwrap**	venv
asynchat	**collections**	**email**	gzip	mailbox	pdb	reprlib	ssl	threading	warnings
asyncio	colorsys	encodings	**hashlib**	mailcap	pickle	resource (Unix)	stat	**time**	wave
asyncore	compileall	ensurepip	**heapq**	marshal	pickletools	rlcompleter	**statistics**	**timeit**	weakref
atexit	configparser	enum	hmac	**math**	pipes (Unix)	runpy	**string**	**tkinter**	**webbrowser**
audioop	contextlib	errno	**html**	mimetypes	pkgutil	sched	stringprep	token	winreg (Win)
base64	contextvars	faulthandler	**http**	mmap	platform	secrets	struct	tokenize	winsound (Win)
bdb	**copy**	fcntl (Unix)	imaplib	modulefinder	plistlib	select	subprocess	trace	wsgiref
binascii	copyreg	**filecmp**	imghdr	msilib (Windows)	poplib	selectors	sunau	traceback	xdrlib
binhex	crypt (Unix)	fileinput	imp	msvcrt (Windows)	**pprint**	shelve	symbol	tracemalloc	**xml**
bisect	**csv**	fnmatch	importlib	multiprocessing	profile	shlex	symtable	tty (Unix)	xmlrpc
builtins	ctypes	**fractions**	inspect	netrc	pstats	**shutil**	**sys**	**turtle**	**zipapp**
bz2	curses (Unix)	ftplib	**io**	nis (Unix)	pty (Linux)	signal	sysconfig	turtledemo	zipfile
calendar	**dataclasses**	functools	ipaddress	nntplib	pwd (Unix)	site	syslog (Unix)	types	zipimport
cgi	**datetime**	gc	**itertools**	**numbers**	pyclbr	smtpd	tabnanny	typing	**zlib**
cgitb	dbm	getopt	json	**operator**	pydoc	smtplib	tarfile	unicodedata	zoneinfo

Gli ambienti di lavoro

- Ambienti di sviluppo **tradizionali** (IDE)
  - IDLE, PyCharm, Visual Studio Code, Eclipse PyDev, …
- Ambienti di sviluppo **on-line**
  - Repl.it, PythonAnywhere, Python Tutor
- Ambienti per il calcolo **interattivo**
  - Spyder, IPython
- **Notebook** Computazionali
  - Jupyter, JupyterLab, Google Colab
- Ambienti per **l'apprendimento**
  - Mu, Thonny, Wing

L’IDE di Visual Studio Code



L’IDE di PyCharm

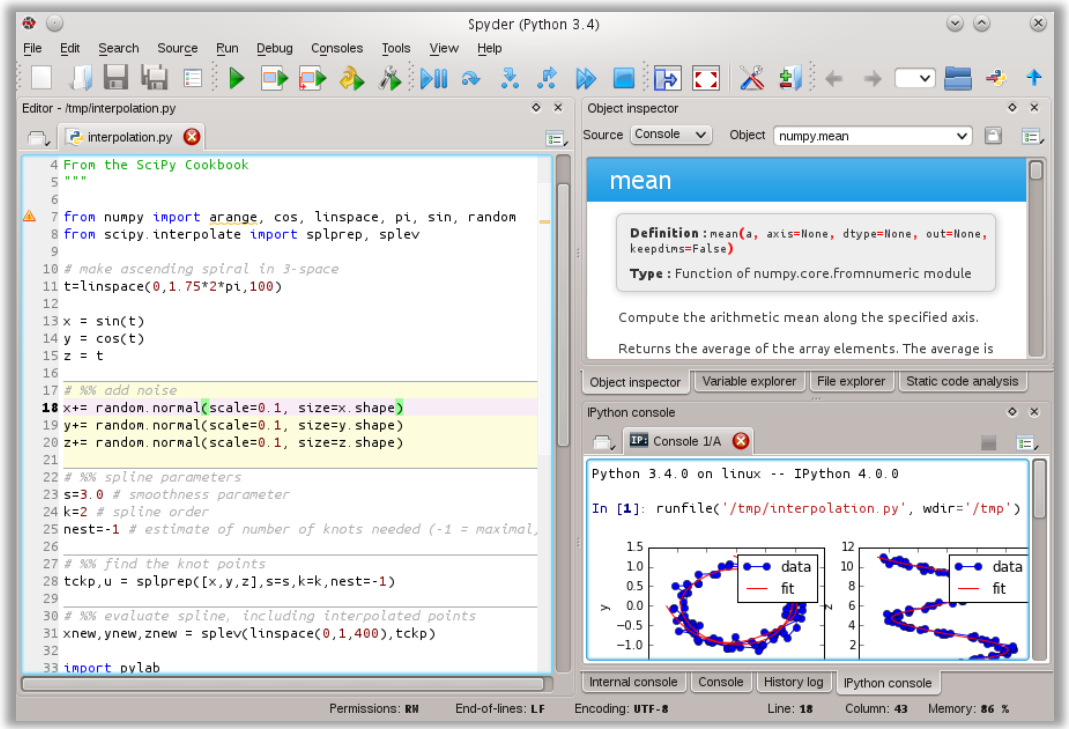
File di progetto

Sorgente del programma

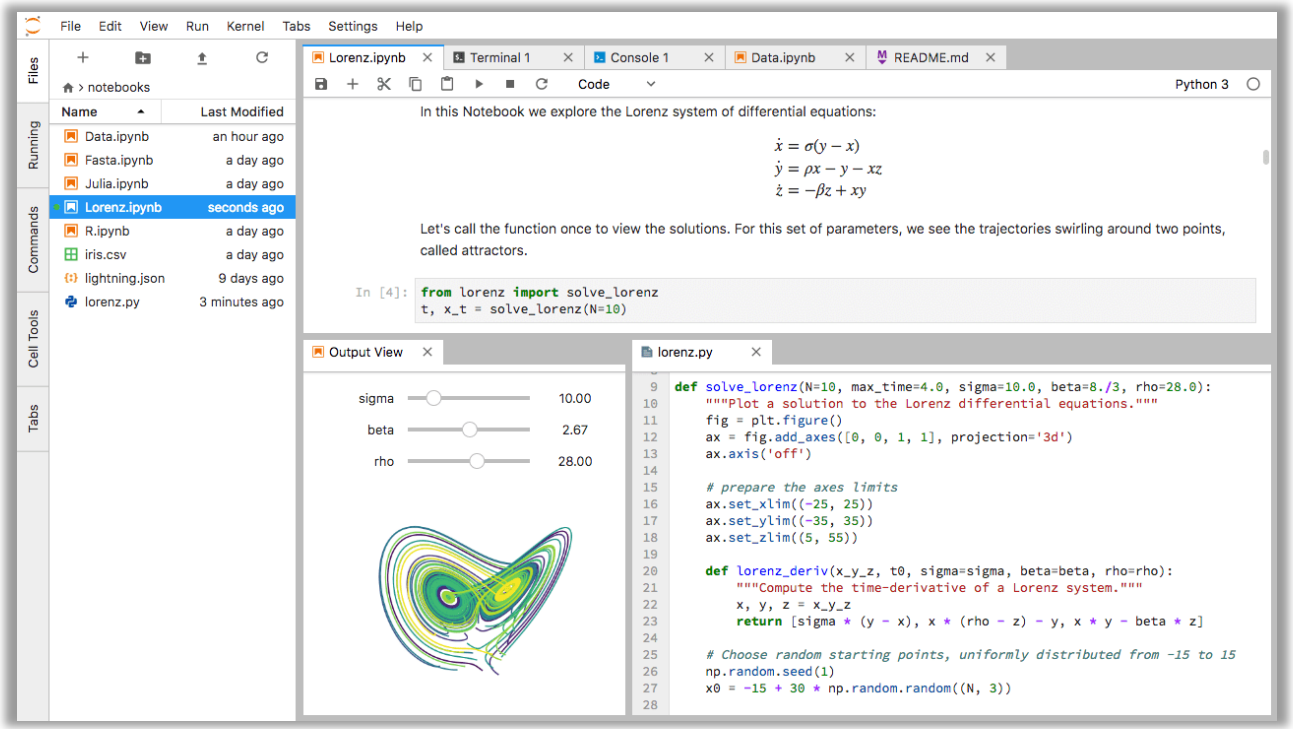
Debugger/Output

Ambienti scientifici interattivi

SPYDER



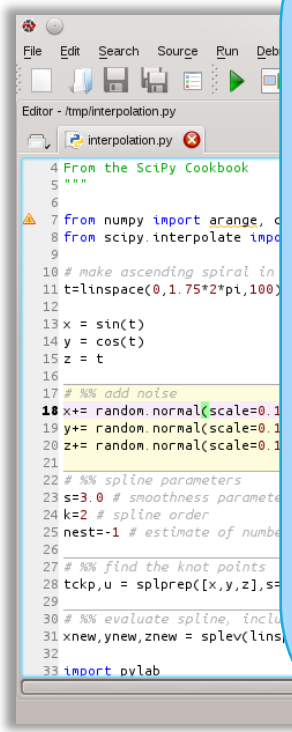
JUPYTERLAB (ANCHE ON-LINE), GOOGLE COLAB



Ambienti scientifici interattivi

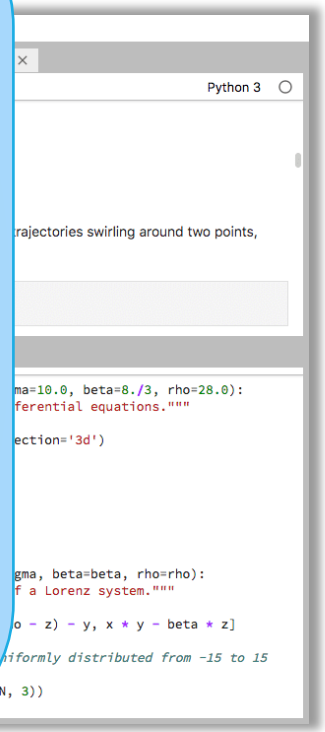
SPYDER

JUPYTERLAB (ANCHE ON-LINE), GOOGLE COLAB

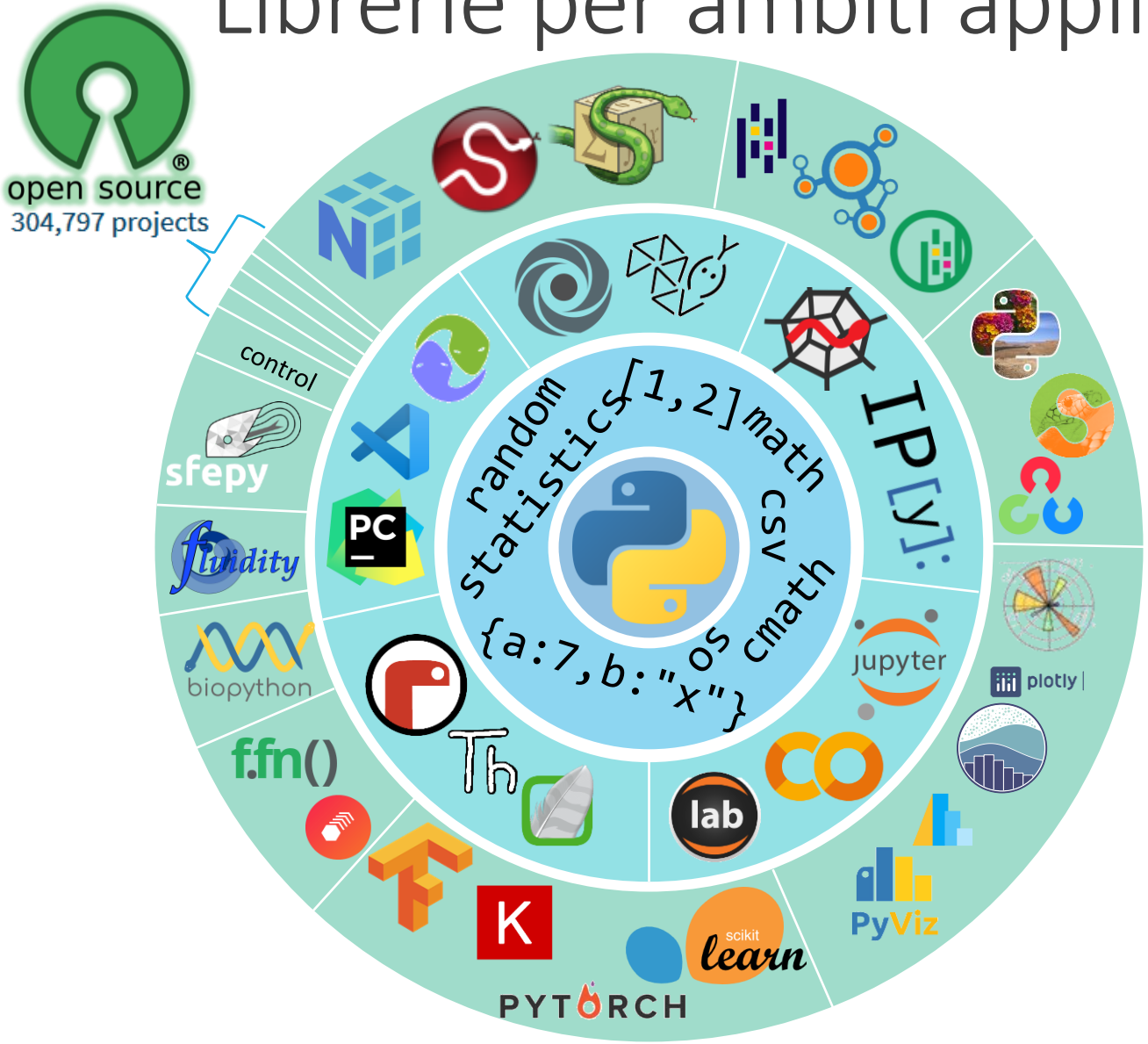


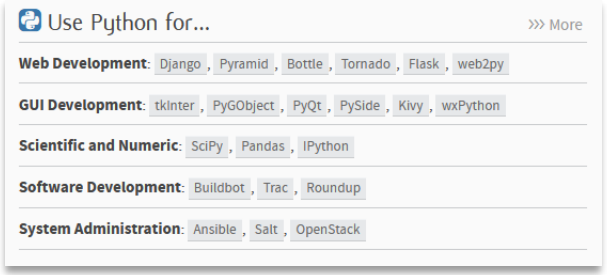
Nuove possibilità…

- Pubblicare on-line esercizi «interattivi» sotto forma di notebook
- Redigere le prime versioni di un articolo, inframmezzando il testo alle formule, con il [ri-]calcolo automatico di risultati e grafici



Librerie per ambiti applicativi





- Scientific computation
  - NumPy, SciPy, SymPy
- Data Analysis, Algoritmi, Grafi
  - Pandas, networkx, GeoPandas
- Image Processing
  - Pillow, scikit-image, OpenCV
- Visualization
  - Pyviz, matplotlib, plotly, seaborn, altair
- Machine Learning
  - Scikit-learn, tensorflow, pytorch, keras
- Fintech
  - f.fn, zipline, pyalgotrade
- Biology and Genome
  - Biopython
- Fluid Dynamics
  - Fluidity
- Finite Elements
  - Sfepy
- Control systems

python™ Package Index

Singoli moduli

ANACONDA®

Toolkit completo per data science

Calcolo scientifico

- NumPy
  - Array, vettori, algebra lineare
- SciPy
  - Package specializzati su diversi ambiti scientifici
- SymPy
  - Calcolo simbolico
- Pandas
  - Analisi e manipolazione dati

Subpackage	Description
cluster	Clustering algorithms
constants	Physical and mathematical constants
fftpack	Fast Fourier Transform routines
integrate	Integration and ordinary differential equation solvers
interpolate	Interpolation and smoothing splines
io	Input and Output
linalg	Linear algebra
ndimage	N-dimensional image processing
odr	Orthogonal distance regression
optimize	Optimization and root-finding routines
signal	Signal processing
sparse	Sparse matrices and associated routines
spatial	Spatial data structures and algorithms
special	Special functions
stats	Statistical distributions and functions

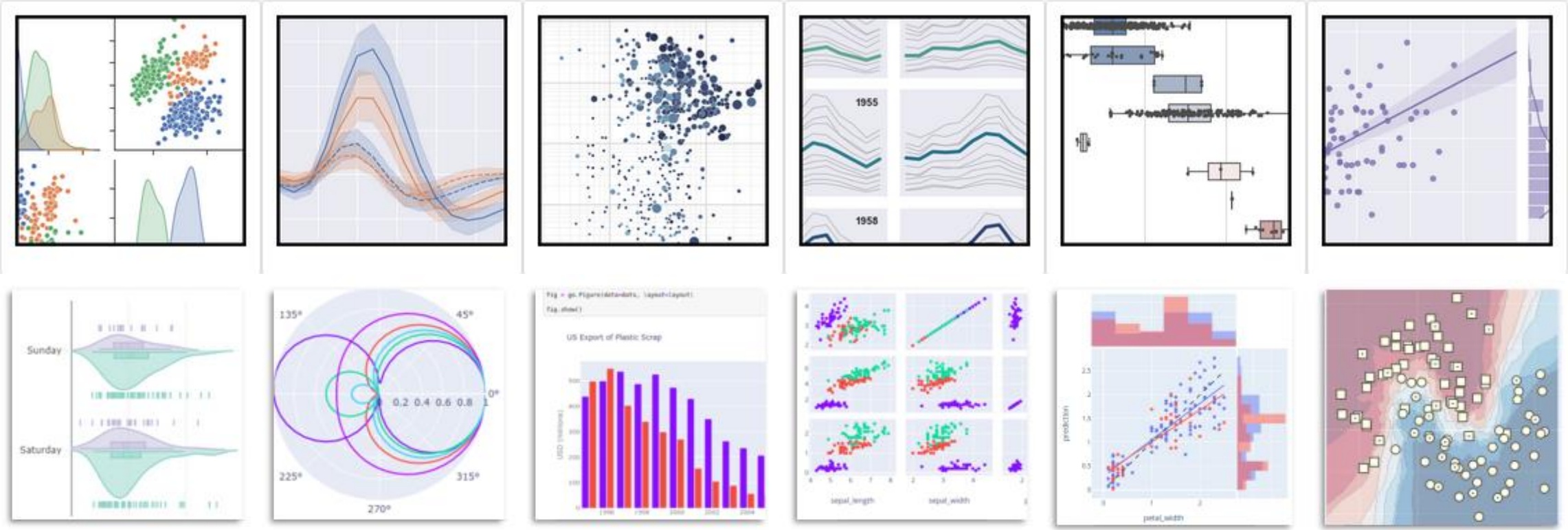
Calcolo scientifico

- NumPy
  - Array, vettori, algebra lineare
- SciPy
  - Package specializzati su diversi ambiti scientifici
- SymPy
  - Calcolo simbolico
- Pandas
  - Analisi e manipolazione dati

Features

- Core capabilities
- Polynomials
- Calculus
- Solving equations
- Combinatorics
- Discrete math
- Matrices
- Geometry
- Plotting
- Physics
- Statistics
- Cryptography
- Parsing
- Printing

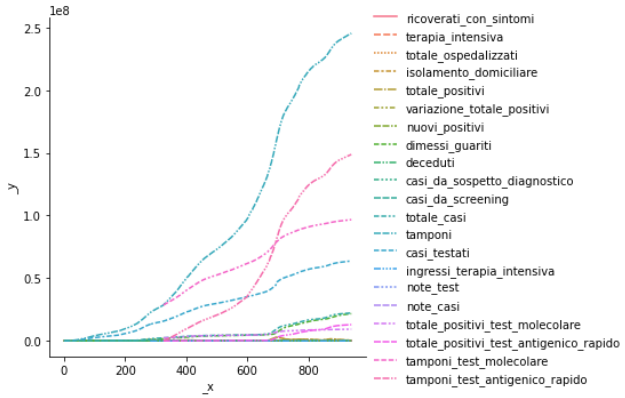
Visualizzazione

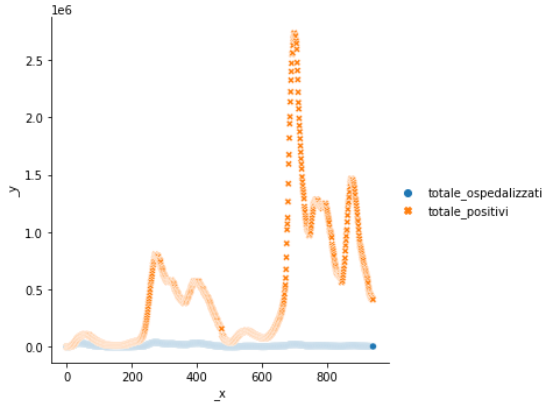


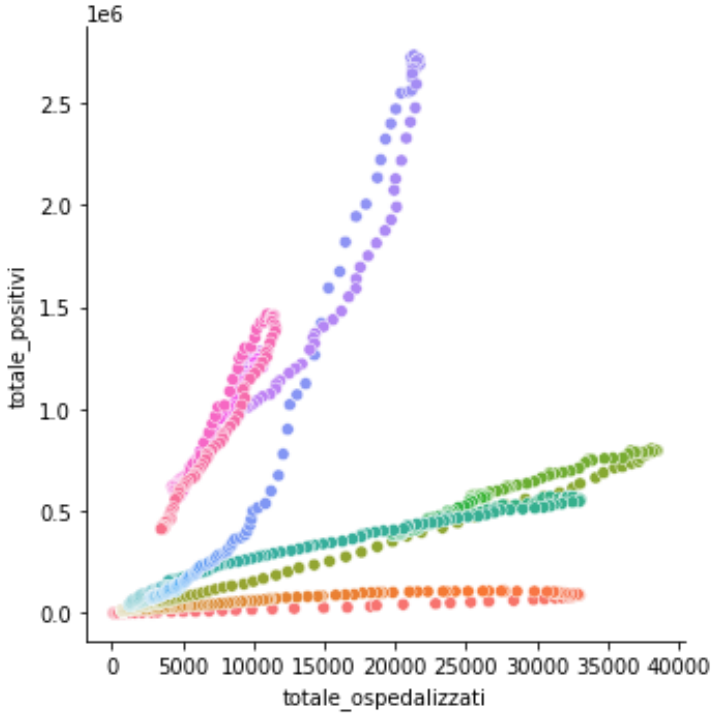
matplotlib, plotly, seaborn, …

Esempio: dati ufficiali Covid-19 in real-time

```
import pandas as pd
import seaborn as sns
sns.set_style("whitegrid")

# Leggi dati aggiornati
covid = pd.read_json(
path_or_buf='https://raw.githubusercontent.com/pcm-dpc/COVID-
19/master/dati-json/dpc-covid19-ita-andamento-nazionale.json',
convert_dates=['data'])

covid.set_index('data', inplace=True)

sns.relplot(data=covid, kind='line')

dati_utili = covid[['totale_ospedalizzati', 'totale_positivi' ]]

sns.relplot(data=dati_utili, kind='line')

sns.relplot(data=dati_utili, kind='scatter',
x='totale_ospedalizzati', y='totale_positivi', hue='data',
legend=False)
```

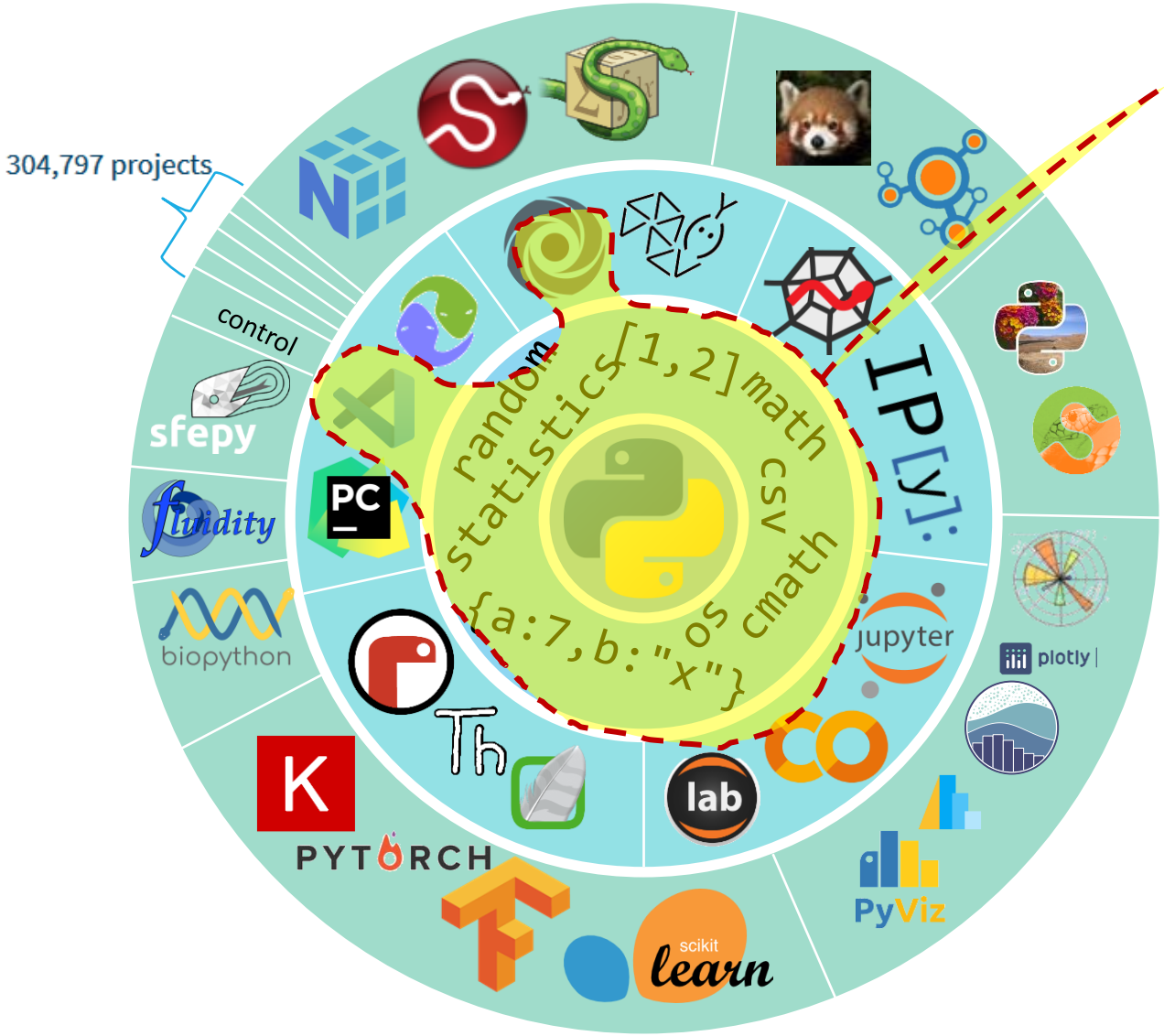






Try me on Google Colab

Dove arriviamo nel corso del primo anno?



- Programma del corso di Informatica
  - Conoscenza di base del linguaggio
  - Familiarità con gli ambienti di sviluppo più semplici
  - Capacità di analizzare un problema e formulare un algoritmo
- Le specializzazioni sulle varie aree non rientrano nel programma di Informatica
  - Possibile costruire negli insegnamenti successivi
  - Partendo da una base consolidata

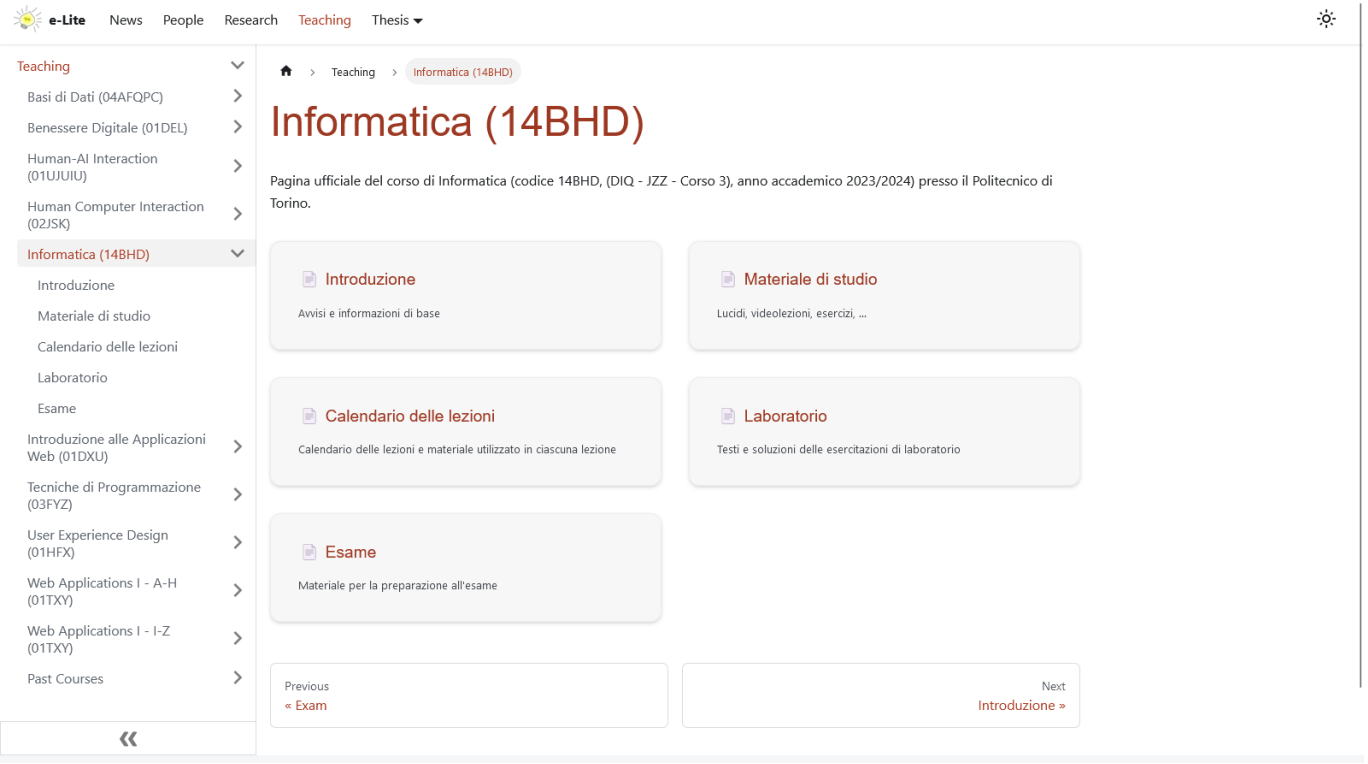
Organizzazione del corso

Sito del corso

Tutto il materiale sarà disponibile su questo sito

- Slide
- Laboratori
- Esempi svolti
- Video Lezioni
- Temi d’esame
- Calendario lezioni
- …

http://bit.ly/polito-informatica



Struttura del corso

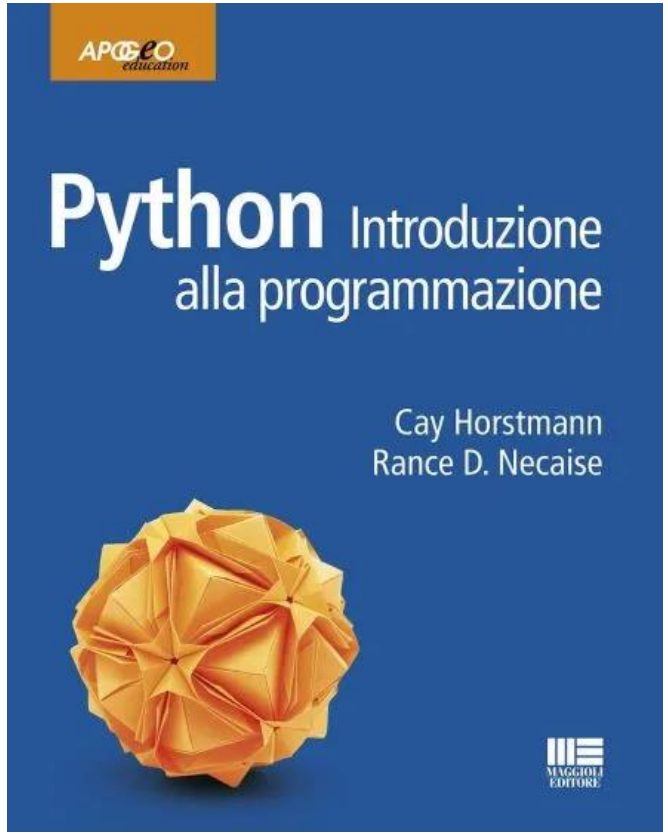
- Programmazione e Python : 41 ore
- Teoria: 9 ore
- Problem solving : 12 ore
- Laboratorio : 18 ore (x 3 squadre)

- Totale: 80 ore/studente

Libri di testo

TESTO COMPLETO IN ITALIANO

VERSIONE RIDOTTA (CAPITOLI 1-8)
CORRISPONDENTE AL PROGRAMMA SVOLTO



Libri di testo: Informazioni dettagliate

TESTO COMPLETO IN ITALIANO

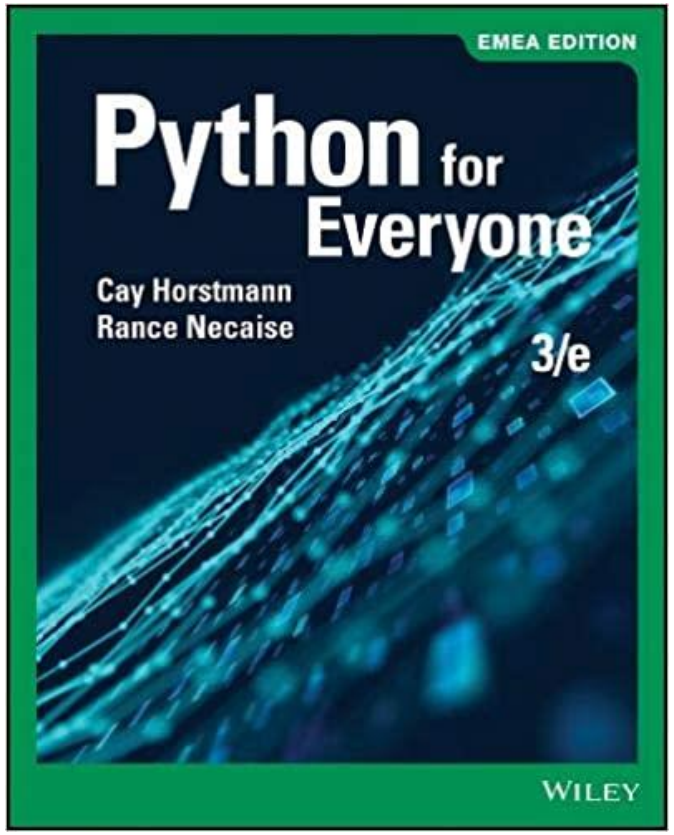
- Concetti di informatica e fondamenti di Python
- Seconda edizione - Giugno 2019 (II° Edizione)
- Cay Horstmann - Rance D. Necaise
- Maggioli Editore
- Giugno 2019
- ISBN 978-8891635433
- https://www.maggiolieditore.it/concetti-di-informatica-e-fondamenti-di-python.html

VERSIONE RIDOTTA (CAPITOLI 1-8) CORRISPONDENTE AL PROGRAMMA SVOLTO

- Python - Introduzione alla programmazione
- Versione ridotta della Seconda Edizione
- Cay S. Horstmann, Rance D. Necaise
- Maggioli Editore
- Luglio 2023
- ISBN: 978-8891663979
- https://www.maggiolieditore.it/python-introduzione-alla-programmazione.html

Libri di testo

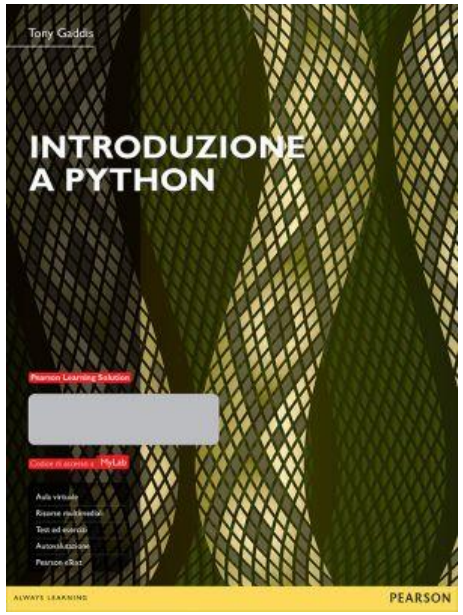
TESTO ORIGINALE IN LINGUA INGLESE



INFO DETTAGLIATE

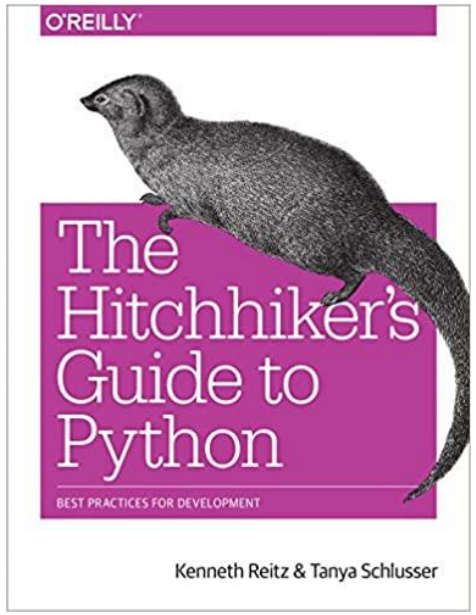
- Python For Everyone
- 3rd Edition
- Cay S. Horstmann, Rance D. Necaise
- Wiley
- ISBN: 978-1-119-49853-7 December 2018
- https://www.wiley.com/en-it/Python+For+Everyone,+3rd+Edition-p-9781119498537

Altre risorse…



Introduzione a Python
Tony Gaddis
Pearson - ISBN: 9788891900999

Introduzione a Python per l'informatica e la data science
Paul J. Deitel, Harvey M. Deitel, Pietro Codara, Carlo Mereghetti
Pearson - ISBN: 9788891915924



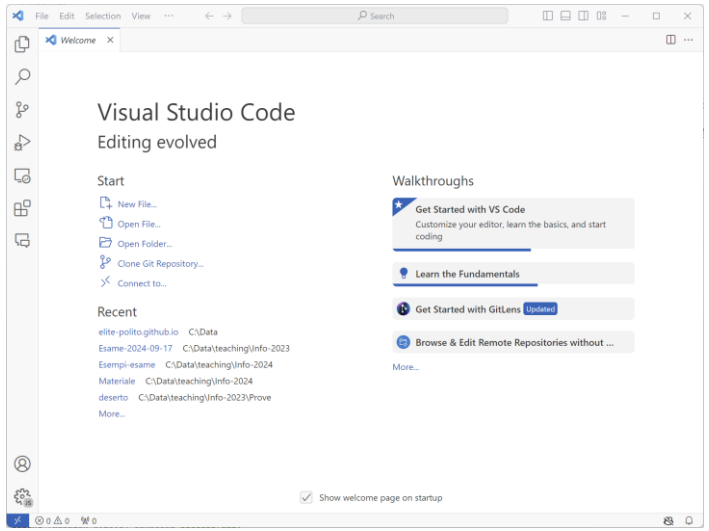
The Hitchhiker's Guide to Python: Best Practices for Development
Kenneth Reitz, Tanya Schlusser
O'Reilly Media - ISBN-13: 978-1491933176

Gratis su:
https://docs.python-guide.org/

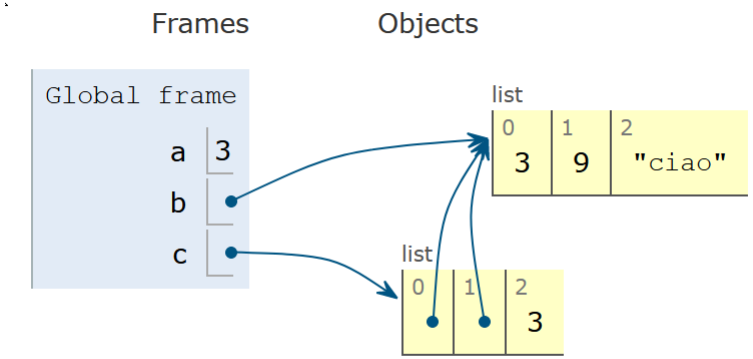
https://www.python.org/
https://docs.python.org/3/
https://docs.python.org/3/tutorial/

https://realpython.com/

Strumenti per programmare



https://code.visualstudio.com/



http://pythontutor.com

https://www.jetbrains.com/pycharm/

- Professional (gratis per docenti e studenti)
- Community (gratis per tutti)

Istruzioni di installazione

14BHDxx - Informatica

Installazione software (piattaforma Windows)

Sommario

FASE A: Installare l'interprete Python 2
Gli ambienti di sviluppo PyCharm 4
FASE B: Installazione di PyCharm Edu **(opzione consigliata)** 4
FASE C: Attivazione di PyCharm Edu 7
FASE D: Creazione di un nuovo progetto in PyCharm Edu 8
In alternativa (opzione avanzata): PyCharm Community o Professional 10
FASE B: Installazione di PyCharm Community o Professional 10
FASE C: Attivazione di PyCharm Community/Professional e della relativa licenza d'uso 12
FASE D: Creazione di un nuovo progetto in PyCharm Community/Professional 14

12BHDxx - Informatica

Installazione software (piattaforma macOS)

Sommario

FASE A: Installare l'interprete Python 2
FASE B: Installare l'ambiente di sviluppo PyCharm 4
FASE C: Attivazione di PyCharm e della relativa licenza d'uso 5
FASE D: Creazione di un nuovo progetto in PyCharm 8

http://bit.ly/polito-informatica

File Edit Selection View Go Run Terminal Help

Settimane

RUN AND DEBUG Python: Current File

nomi_frequenti_studenti.py 2 | launch.json 1, U

Settimana01 > nomi_frequenti_studenti.py > main

VARIABLES

Locals

```
freq: {'DANIELE': 4, 'MATILDE': 3, 'DOAA...
  > special variables
  > function variables
    'DANIELE': 4
    'MATILDE': 3
    'DOAA': 1
    'SARA': 3
    'LETIZIA': 2
    'NICOLO': 2
    'LEONARDO': 2
    'DILETTA': 1
    'SIMONE': 7
    'EDOARDO': 5
    'ANGELO': 1
```

int

WATCH

CALL STACK Paused on breakpoint

main	nomi_frequenti_studenti.py	60:1
<module>	nomi_frequenti_studenti.py	65:1

BREAKPOINTS

- [ ] Raised Exceptions
- [x] Uncaught Exceptions
- [ ] User Uncaught Exceptions
- [x] nomi_frequenti_studenti.py Settimana01 60

```
47      nomi = estrai_nomi(stud)
48      print(f"Nella classe ci sono {len(stud)} studenti")
49      freq = frequenze(nomi)
50      max_freq = max_frequenza(freq)
51      print(f"Il nome più frequente compare {max_freq} volte")
52      nomi_max = nomi_piu_frequenti(freq, max_freq)
53      print(f"Si tratta di : {', '.join(nomi_max)}")
54      # estrai solo i nomi che compaiono almeno 3 volte
55      freq2 = {k: v for (k, v) in freq.items() if v >= 3}
56      print(
57          f"I nomi che compaiono almeno 3 volte sono {', '.join(sorted(list(freq2.keys())))}."
58      )
59
60      hist = list(sorted(list(freq2.items()), key=operator.itemgetter(1), reverse=True))
61      pyplot.barh(y=[h[0] for h in hist], width=[h[1] for h in hist])
62      pyplot.show()
63
64
65  main()
66
```

PROBLEMS 3 | OUTPUT | DEBUG CONSOLE | TERMINAL

```
ine 294, in _get_code_from_file
    code = compile(f.read(), fname, 'exec')
  File "C:\Data\teaching\Info-2022\Settimane\.vscode\launch.json", line 9
    "purpose": ["debug-in-terminal"]
               ^^^^^^^^^^^^^^^^^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
(venv) PS C:\Data\teaching\Info-2022\Settimane\.vscode>  C:; cd 'C:\Data\teaching\Info-2022\Settimane\Settimana01'; &
ttimane\venv\Scripts\python.exe' 'c:\Users\Fulvio\.vscode\extensions\ms-python.python-2023.15.12301911\pythonFiles\lib
debugpy\launcher' '65381' '--' 'C:\Data\teaching\Info-2022\Settimane\Settimana01\nomi_frequenti_studenti.py'
Nella classe ci sono 280 studenti
Il nome più frequente compare 12 volte
Si tratta di : LORENZO
I nomi che compaiono almeno 3 volte sono ALBERTO, ALESSANDRO, ALESSIO, ANDREA, DANIELE, DAVIDE, EDOARDO, FEDERICO, FRA
IORGIA, GIOVANNI, GIULIA, JACOPO, LORENZO, LUCA, MARCO, MATILDE, MATTEO, MATTIA, PIETRO, SARA, SIMONE, SOFIA, STEFANO.
Backend TkAgg is interactive backend. Turning interactive mode on.
```

master* ⊗ 1 ⚠ 2 Python: Current File (Settimane) Ln 60, Col 1 Space

Visual Studio Code

Visual Studio Code

Version: 1.81.1 (user setup)
Commit: 6c3e3dba23e8fadc360aed75ce363ba185c49794
Date: 2023-08-09T22:22:42.175Z (1 wk ago)
Electron: 22.3.18
ElectronBuildId: 22689846
Chromium: 108.0.5359.215
Node.js: 16.17.1
V8: 10.8.168.25-electron.0
OS: Windows_NT x64 10.0.22621

Copy OK

Informazioni pratiche

Orario Settimanale

	Lunedì	Martedì	Mercoledì	Giovedì	Venerdì
08:30-10:00					
10:00-11:30					
11:30-13:00			**Lezione / Esercitazione** Aula 4	**Laboratorio** Squadra 1 Aula 3D	
13:00-14:30			**Lezione / Esercitazione** Aula 4	**Laboratorio** Squadra 2 Aula 3D	
14:30-16:00	**Lezione / Esercitazione** Aula 4				
16:00-17:30			**Laboratorio** Squadra 3 Aula 3D		
17:30-19:00					

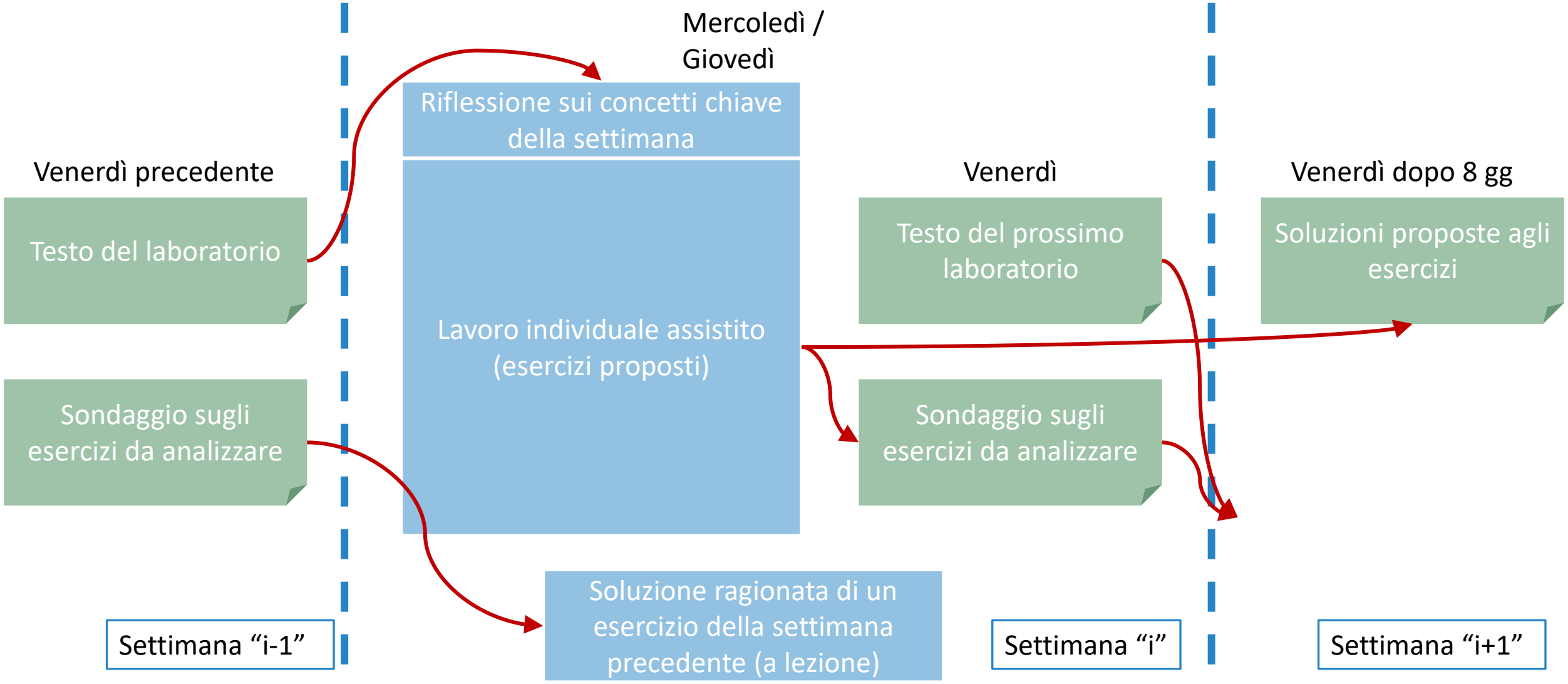
Laboratori

- La parte più importante del corso, in cui imparare a risolvere problemi e scrivere programmi
- Il corso è diviso in 3 squadre
  - Presso i Laboratori Informatici (LAIB)
- Inizio laboratori: **02-03/10/2024**
- Testo pubblicato sul sito del corso
- È necessario installare il software richiesto (*Python* e *Visual Studio Code*) - vedere le istruzioni di installazione sul portale

Suddivisione in squadre

- Squadra 1 (giovedì 11:30, Aula 3D): xxxxx
- Squadra 2 (giovedì 13:00, Aula 3D): xxxxx
- Squadra 3 (mercoledì 16:00, Aula 3D): xxxxx

Svolgimento dei laboratori



Tutorati

- Nel Corso del semestre saranno disponibili dei *tutor* (in presenza e on-line) per aiutarvi a rivedere esercizi e chiarire dubbi e difficoltà
- I tutorati sono un'opportunità da sfruttare!

- Per info sui tutorati ed altri avvisi generali unirsi al canale Telegram:
  - https://t.me/+QttiF2nn0hFiNDg0

Comunicazioni

- **Tutti** i contatti con i docenti avverranno sulla piattaforma **Telegram**
  - Non inviare e-mail ai docenti, ma utilizzare il **gruppo** Telegram
  - Non inviare messaggi privati, se non richiesti dai docenti stessi
- **Iscriversi (obbligatorio!) all'indirizzo** https://t.me/politoinfo2024

@POLITOINFO2024

Esame

Contenuti dell’esame

- 3 domande brevi sulla parte teorica del corso (6 punti totali)
- Un esercizio di programmazione (26 punti)
  - Con la possibilità di usare uno strumento di sviluppo per la scrittura del codice
  - Sarà consegnato il codice sorgente del programma sviluppato
  - Il codice verrà corretto manualmente (valuteremo la qualità della soluzione, e il rispetto delle richieste del problema, in maniera indipendente da eventuali errori sintattici e dall’effettivo funzionamento del programma)
- Nelle ultime 2 settimane del corso vi proporremo diversi esercizi di simulazione dell’esame, in modo da poter familiarizzare con le modalità di esame e con le conoscenze richieste per passarlo
- Temi d’esame: https://github.com/polito-informatica/Esempi-esame

Cosa serve per passare [bene] l'esame?

- Capacità logico-razionali di analisi e di sintesi
  - Comprendere i propri processi risolutivi e saperli formalizzare
- Svolgere tutti gli esercizi proposti
  - Davvero
  - Anche quelli [che sembrano] facili
  - Da soli
  - Su Personal Computer
  - Verificarli con dati diversi
  - Cercare di metterli in crisi
- Inventarsi nuovi problemi, o varianti di quelli proposti
  - E poi risolverli

Accertamento intermedio - NOVITÀ 2024/25

- **Sperimentazione di una <u>prova intermedia</u>**
  - Verso fine novembre, durante le ore di laboratorio
  - **4 esercizi (2 teoria + 2 programmazione)**
    a) **2 domande di "teoria" (simili a quelle dell'esame) + 1 esercizio di Python**
    b) **1 esercizio di Python**

a) **darà un punteggio da 0 a 6 punti (2x3) che vi ESONERA dallo svolgere gli esercizi di teoria durante l'esame**
   - Il punteggio verrà sommato al voto d'esame della parte di programmazione
   - **BONUS:** più tempo per fare il programma!

b) **darà un bonus compreso tra 0 e 3 punti da sommare al voto dell'esame**

VALIDITÀ SUI 4 APPELLI 2024/25

- **Gli aspetti logistici verranno chiariti più avanti (organizzazione, modalità)**

Contatti

Docenti

- Fulvio Corno
  - Dipartimento di Automatica e Informatica (3° piano)
  - fulvio.corno@polito.it
- Roberta Bardini
  - Dipartimento di Automatica e Informatica (2° piano)
  - roberta.bardini@polito.it
- Lorenzo Martini
  - Dipartimento di Automatica e Informatica (2° piano)
  - lorenzo.martini@polito.it

- Ci trovate su Telegram!

Link utili

- Sito del corso (ufficiale):
  - http://elite.polito.it/ → Teaching → Informatica (14BHD)
  - Link breve: http://bit.ly/polito-informatica
- Gruppo Telegram
  - https://t.me/politoinfo2023
- Materiale, laboratori, esercizi
  - https://github.com/polito-informatica

Licenza d’uso

- Queste diapositive sono distribuite con licenza Creative Commons “Attribuzione - Non commerciale - Condividi allo stesso modo 2.5 Italia (CC BY-NC-SA 2.5)”
- Sei libero:
  - di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera
  - di modificare quest'opera
- Alle seguenti condizioni:
  - **Attribuzione** — Devi attribuire la paternità dell'opera agli autori originali e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.
  - **Non commerciale** — Non puoi usare quest'opera per fini commerciali.
  - **Condividi allo stesso modo** — Se alteri o trasformi quest'opera, o se la usi per crearne un'altra, puoi distribuire l'opera risultante solo con una licenza identica o equivalente a questa.
- http://creativecommons.org/licenses/by-nc-sa/2.5/it/