



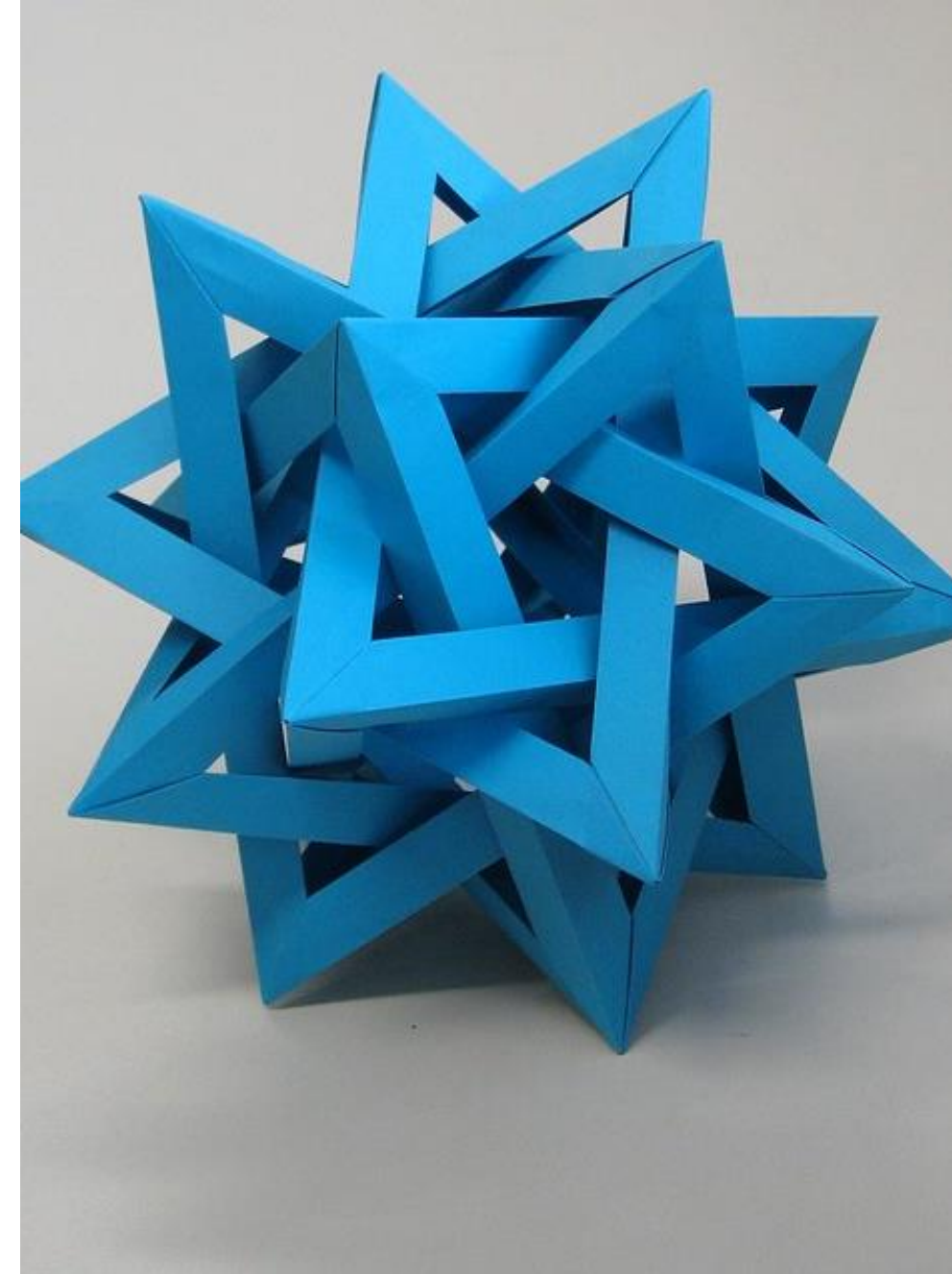
Politecnico  
di Torino

Dipartimento  
di Automatica e Informatica

# 14BHDxx Informatica

ING. INF/ELT/ENE/MTM/ELN/FIS/CIN – CORSO #3 (DET-HEZ)

PROF. FULVIO CORNO



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

# Welcome

---

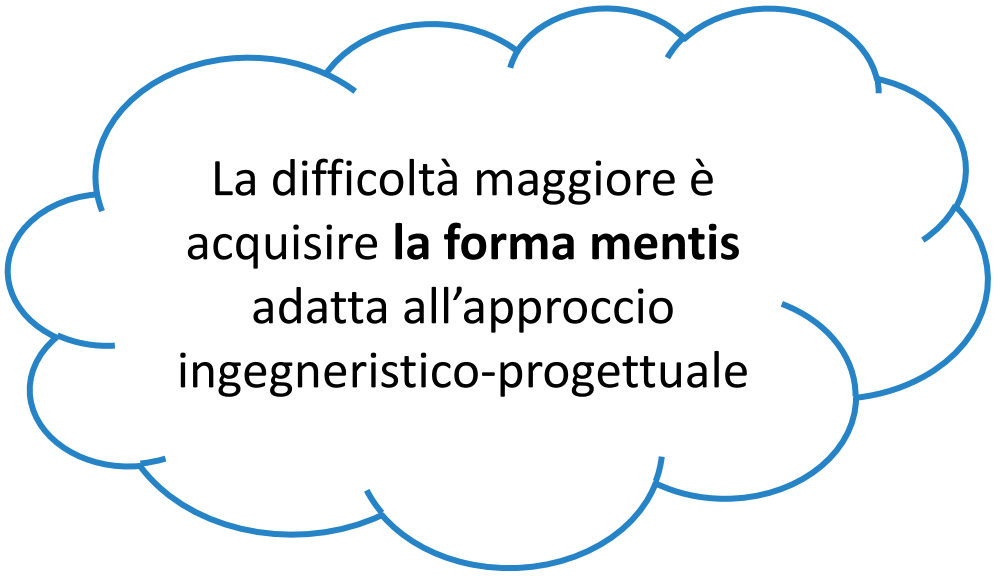
# Corso di Informatica (14BHD)

- Insegnamento obbligatorio, 8 crediti, 1° Semestre
- Corso n. 3
  - Studenti con cognomi compresi tra DET e HEZ
  - Iscritti ai corsi di ing. Informatica, Elettrica, Energetica, Matematica, Elettronica, Fisica, Cinema
- Docenti:
  - **Fulvio Corno**  
(lezioni, esercitazioni in aula)
  - **Roberta Bardini**  
(esercitazioni in laboratorio, esercitazioni in aula)



# Benvenuti ad Ingegneria

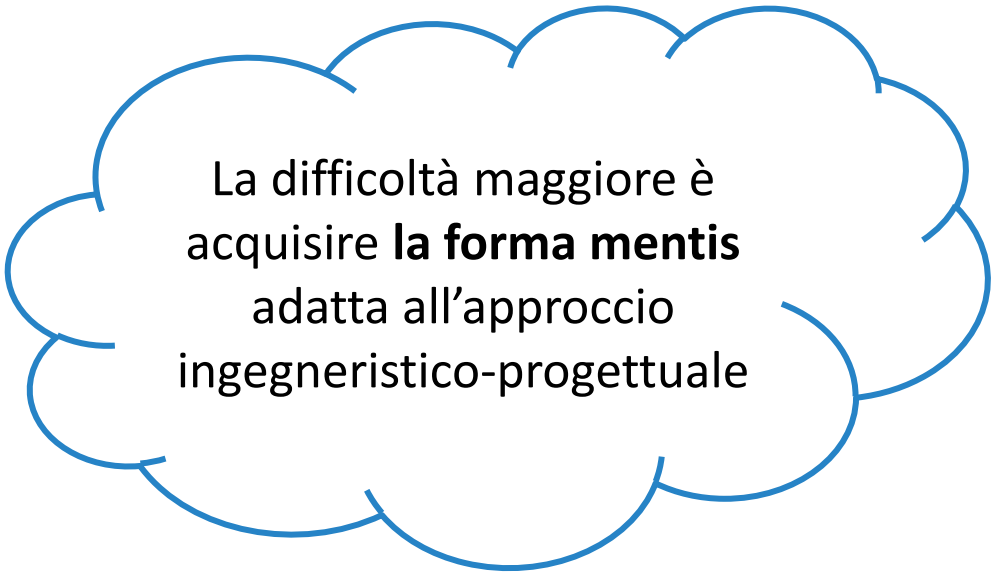
- Questo è il **primo corso di Ingegneria** che affronterete
- Ingegneria =
  - Saper progettare
  - Risolvere problemi
  - Trovare soluzioni
  - Soddisfare le specifiche
  - Nel rispetto dei vincoli
  - Con gli strumenti disponibili



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



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

# Programma del corso

---

# 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).

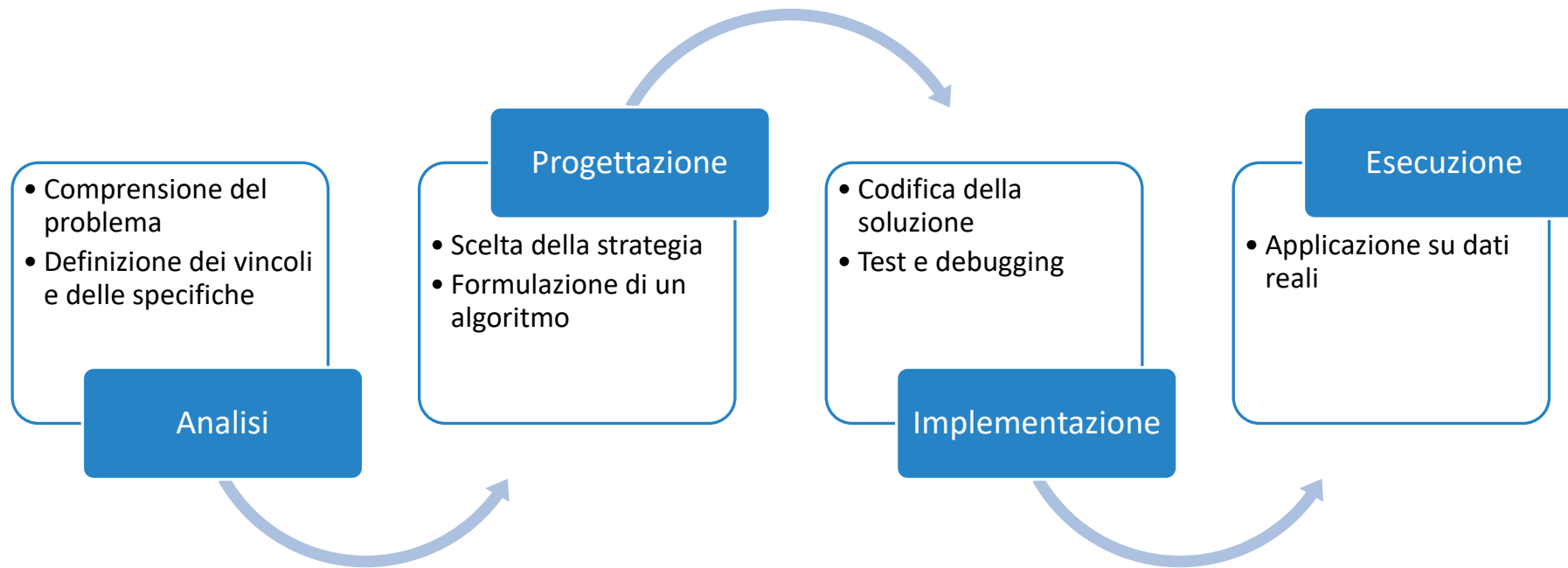
# Contenuti

- 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...

```
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
```

```
# 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 = '01TXYOYV_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>

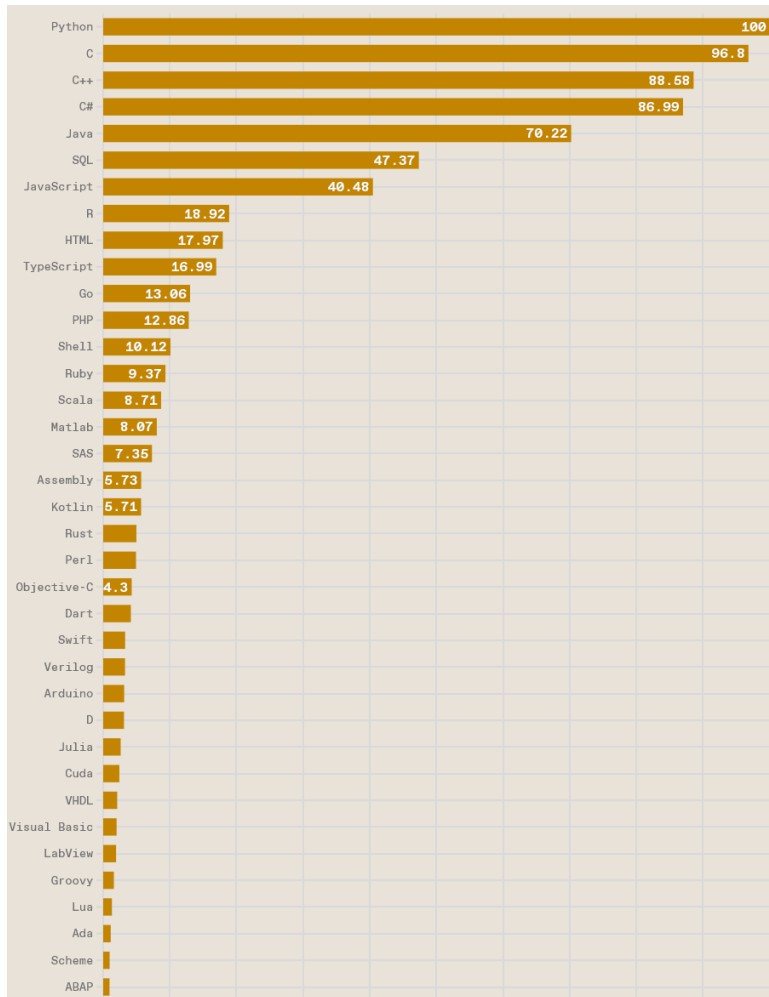
# Il linguaggio Python



- 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



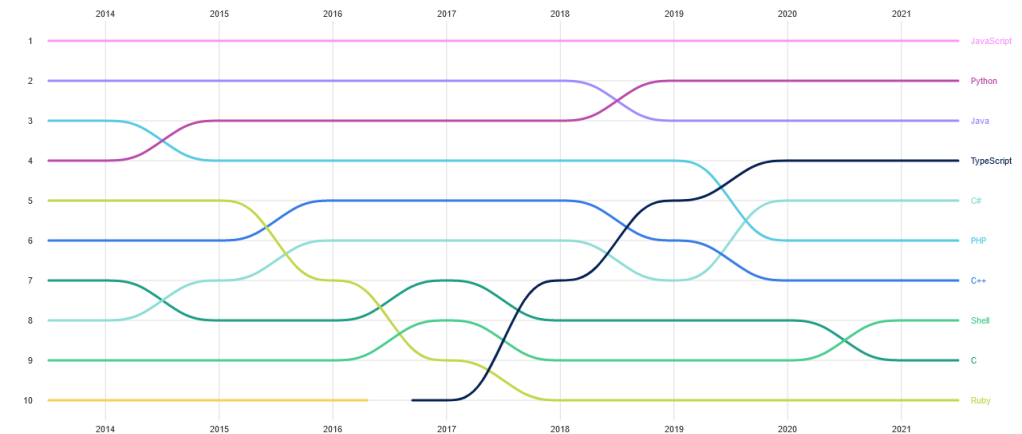
- IEEE Spectrum, 23 Aug 2022  
<https://spectrum.ieee.org/top-programming-languages-2022>
  - Top Programming Languages 2022
- Altre statistiche, per i più curiosi:
  - <https://www.tiobe.com/tiobe-index/>
  - <http://pypl.github.io/PYPL.html>
  - <https://octoverse.github.com/>
  - ...

# Diffusione del linguaggio Python

Aug 2022	Aug 2021	Change	Programming Language	Ratings	Change
1	2	▲	 Python	15.42%	+3.56%
2	1	▼	 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	▼	 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	▼	 R	0.92%	-0.13%
17	17		 MATLAB	0.90%	-0.08%
18	15	▼	 Ruby	0.82%	-0.18%
19	13	▼	 Fortran	0.81%	-0.32%
20	20		 Perl	0.72%	-0.06%

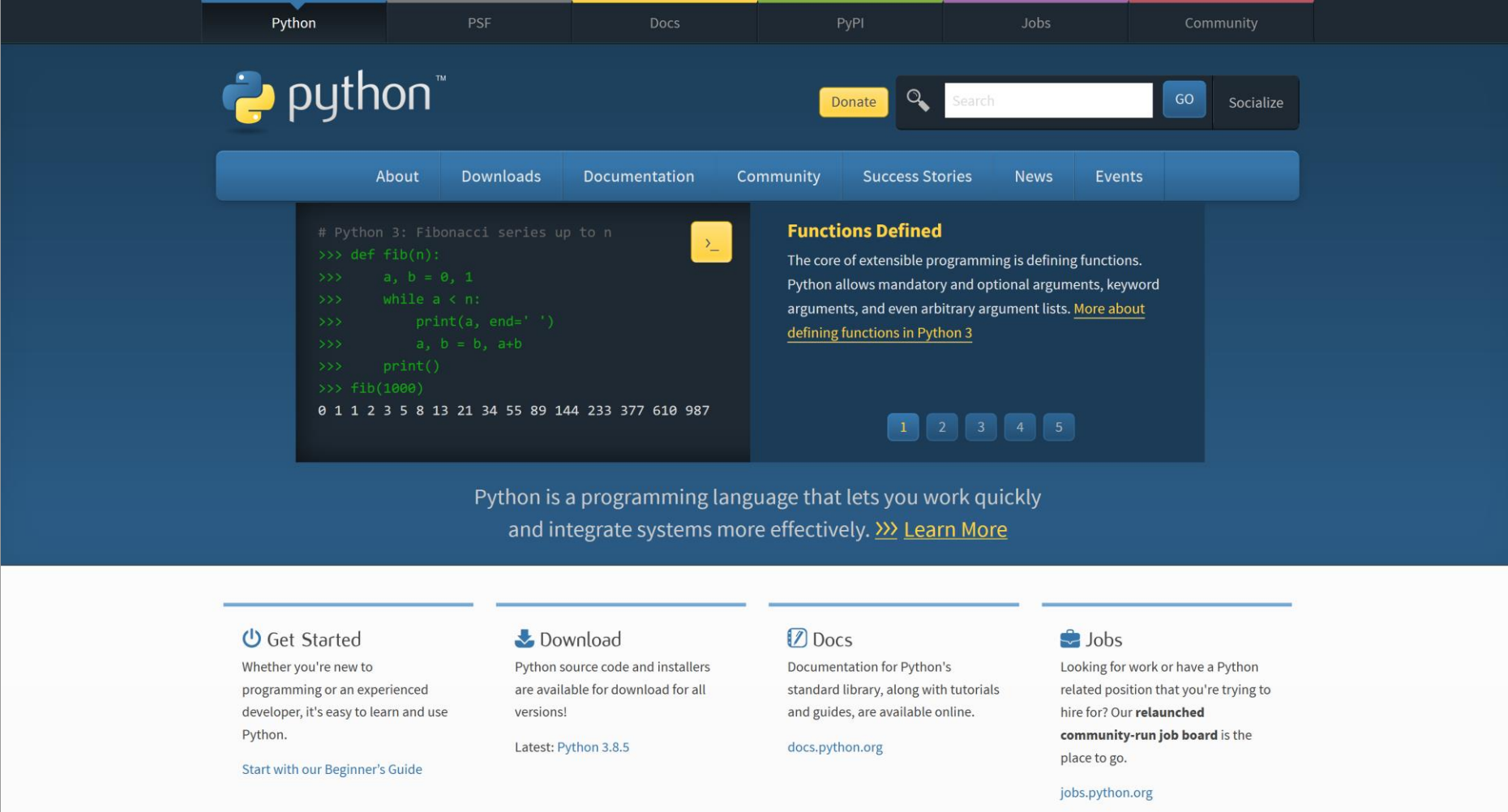
<https://www.tiobe.com/tiobe-index/>

Top languages over the years



<https://octoverse.github.com/>

# <https://www.python.org/>



The screenshot shows the Python.org homepage with a dark blue header and a white footer. The header contains navigation links: Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is a search bar with a magnifying glass icon and a 'GO' button. The main content area features a large blue box with a code editor showing a Fibonacci function and its output. To the right of the code editor is a section titled 'Functions Defined' with a brief description of Python's extensibility. Below the blue box is a white section with four columns: 'Get Started', 'Download', 'Docs', and 'Jobs'. Each column contains a brief description and a link to the relevant resource.

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

**Functions Defined**

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

**Get Started**

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

[Start with our Beginner's Guide](#)

**Download**

Python source code and installers are available for download for all versions!

Latest: Python 3.8.5

**Docs**

Documentation for Python's standard library, along with tutorials and guides, are available online.

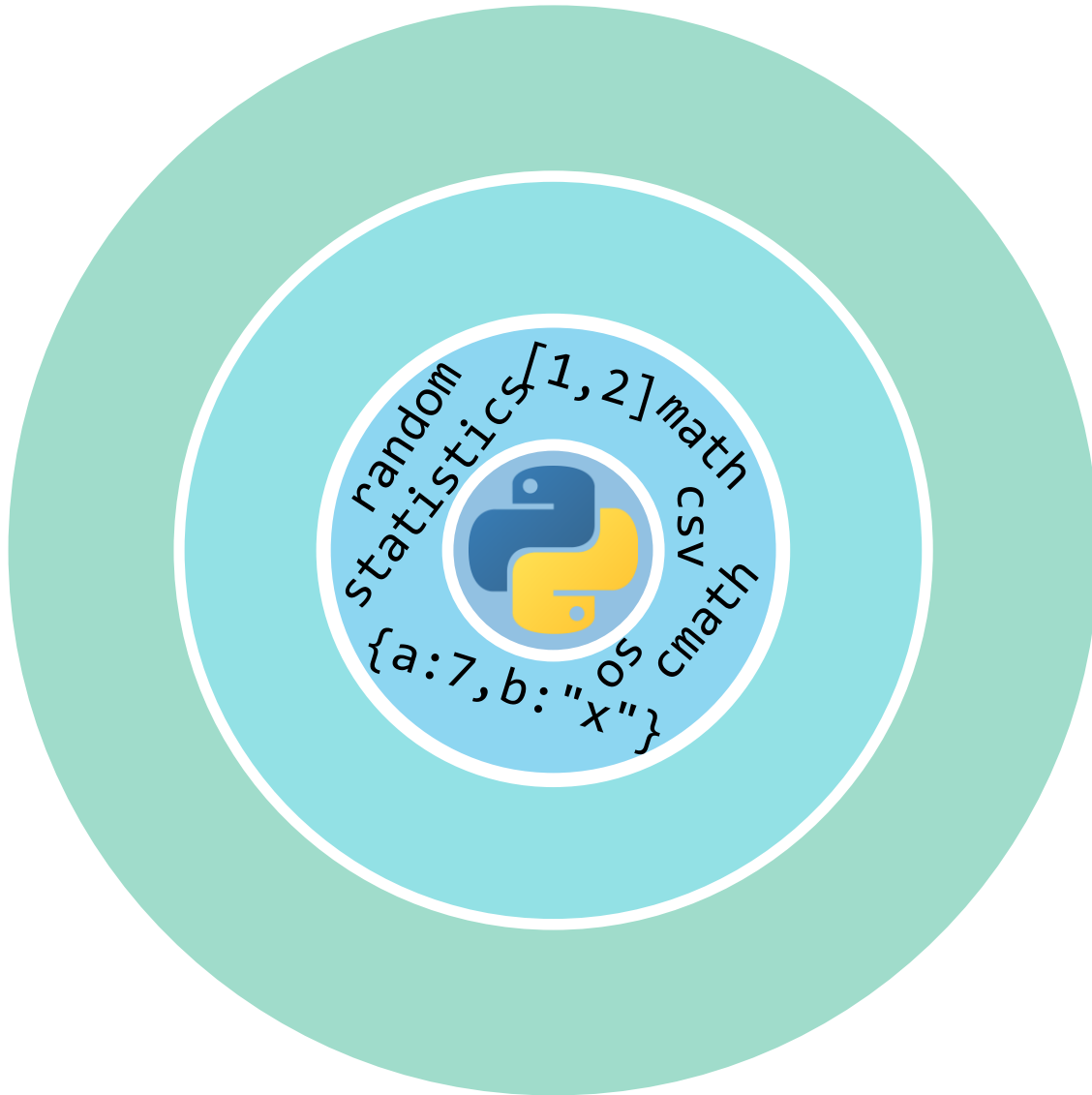
[docs.python.org](https://docs.python.org)

**Jobs**

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

[jobs.python.org](https://jobs.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

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



# Gli ambienti di lavoro

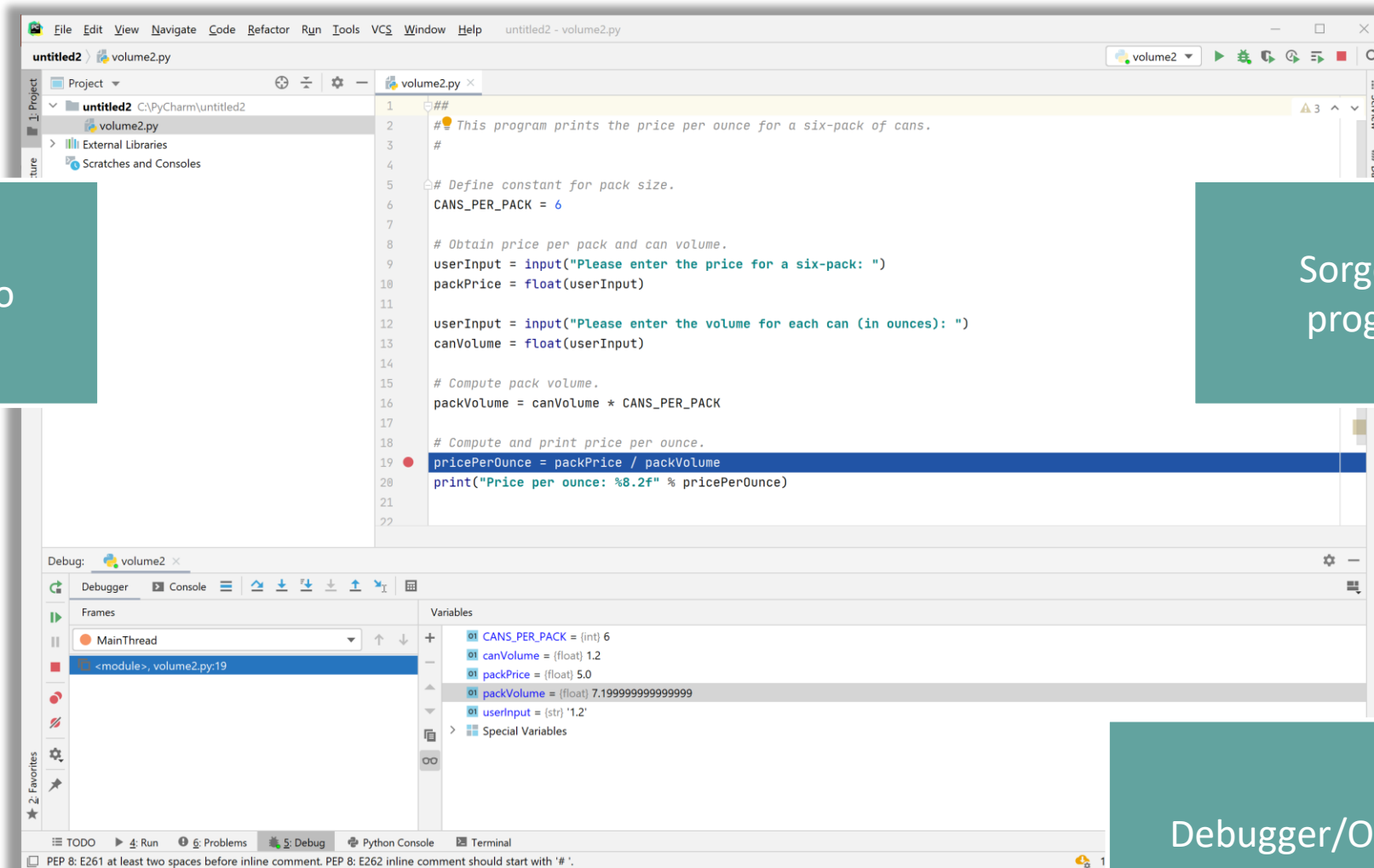


- Ambienti di sviluppo **tradizionali** (IDE)
  - IDLE, PyCharm, Visual Studio Code, Eclipse PyDev, ...
- Ambienti di sviluppo **on-line**
  - Repl.it, PythonAnywhere, Python Tutor
- Ambienti per il calcolo **interattivo**
  - Spyder, IPython
- **Notebook** Computazionali
  - Jupyter, JupyterLab, Google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

# L'IDE di PyCharm

File di progetto

Sorgente del programma



Debugger/Output

# IDE On-line : <https://replit.com/>

The screenshot shows the Replit IDE interface. At the top, the user is logged in as @anonymous / GentleGiftedGeneric. The interface is divided into three main sections: a left sidebar with file management icons, a central code editor, and a right sidebar for output and console. The code editor displays a file named main.py with the following code:

```
1 # My first Python program
2 print("Hello, world")
```

The output section on the right shows the result of running the code: "Hello, world".

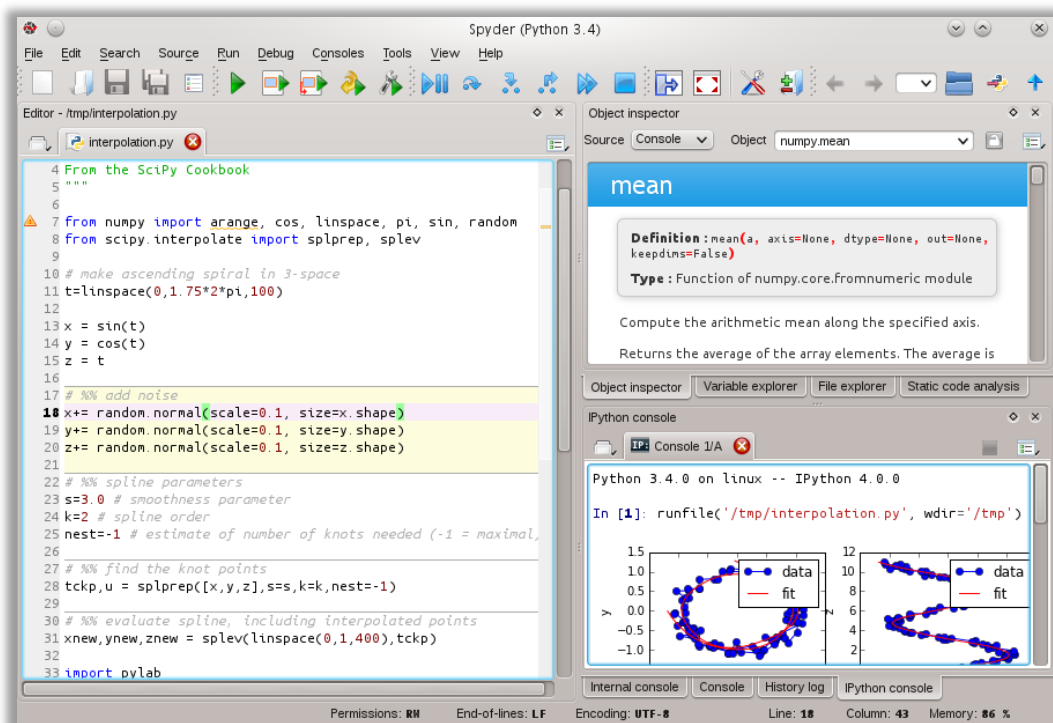
Annotations in blue boxes point to specific features:

- File di progetto** points to the file management icons in the left sidebar.
- Sorgente del programma** points to the code editor area.
- Output del programma e console interattiva** points to the output section on the right.

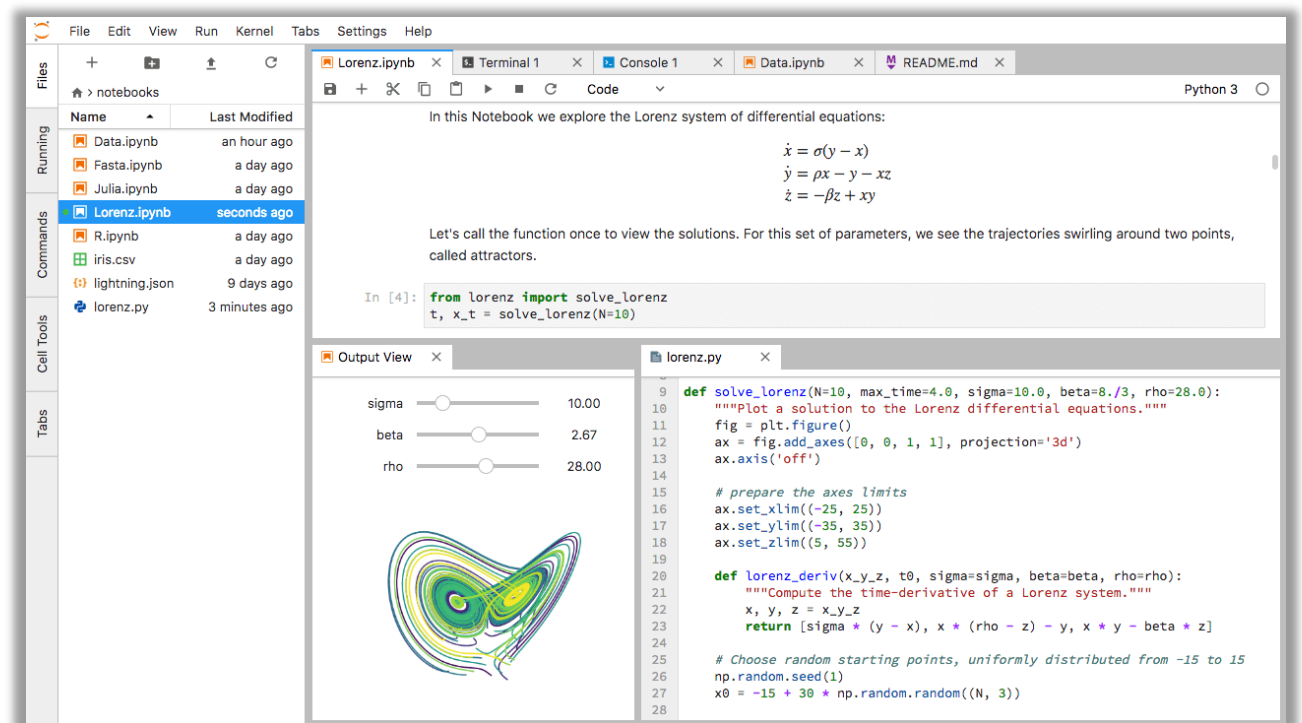
A green box at the bottom left contains the text: "Opzione fantastica per esempi rapidi, per testare frammenti di programma, per evitare di creare un intero Progetto per una piccola parte di codice,, ...".

# Ambienti scientifici interattivi

## SPYDER

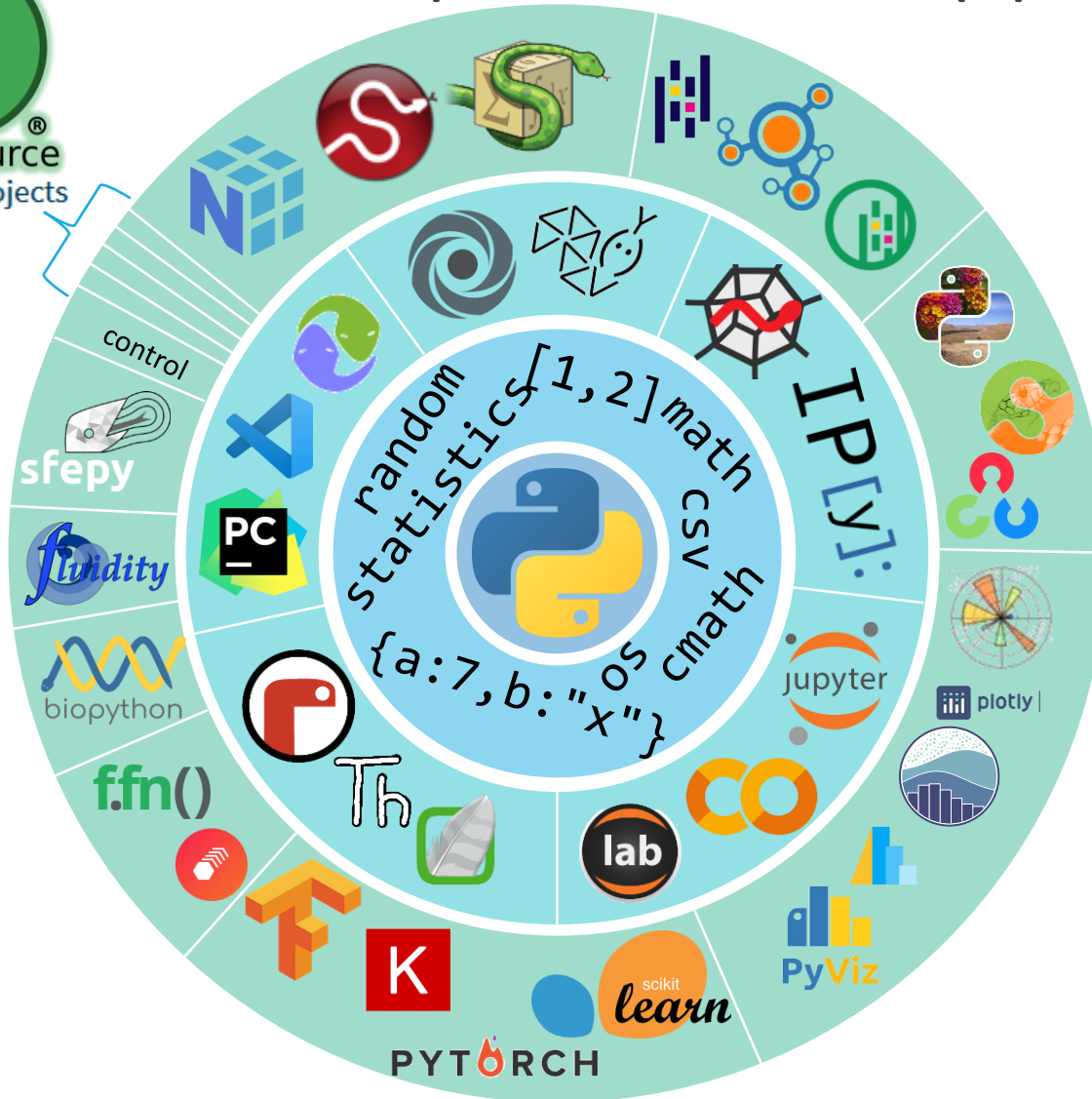


## JUPYTERLAB (ANCHE ON-LINE), GOOGLE COLAB

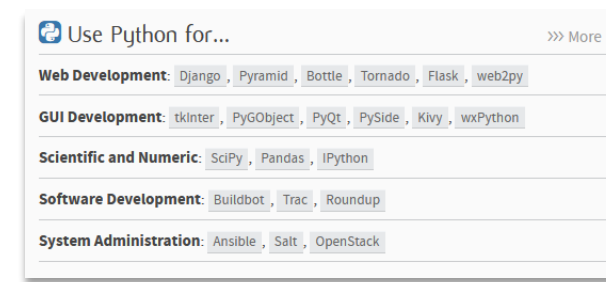


# Librerie per ambiti applicativi

  
open source  
304,797 projects



- 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



*Singoli moduli*



*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

**cluster**

**constants**

**fftpack**

**integrate**

**interpolate**

**io**

**linalg**

**ndimage**

**odr**

**optimize**

**signal**

**sparse**

**spatial**

**special**

**stats**

## Description

Clustering algorithms

Physical and mathematical constants

Fast Fourier Transform routines

Integration and ordinary differential equation solvers

Interpolation and smoothing splines

Input and Output

Linear algebra

N-dimensional image processing

Orthogonal distance regression

Optimization and root-finding routines

Signal processing

Sparse matrices and associated routines

Spatial data structures and algorithms

Special functions

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

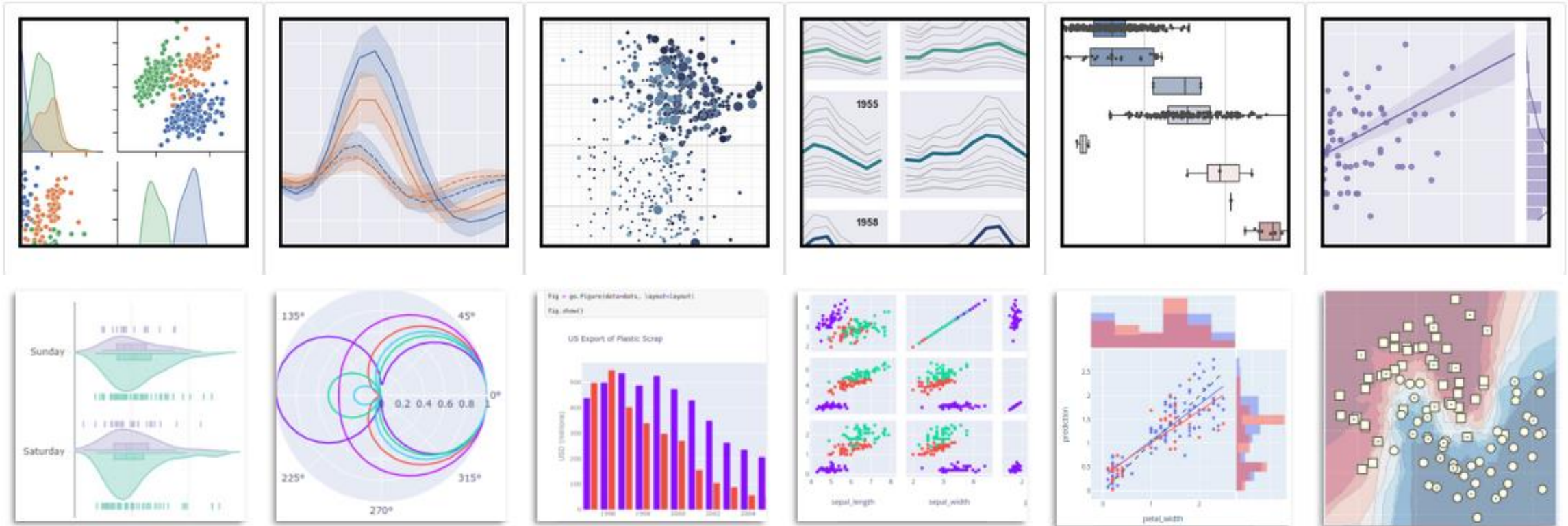
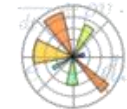
#### ✓ Cryptography

#### ✓ 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'])

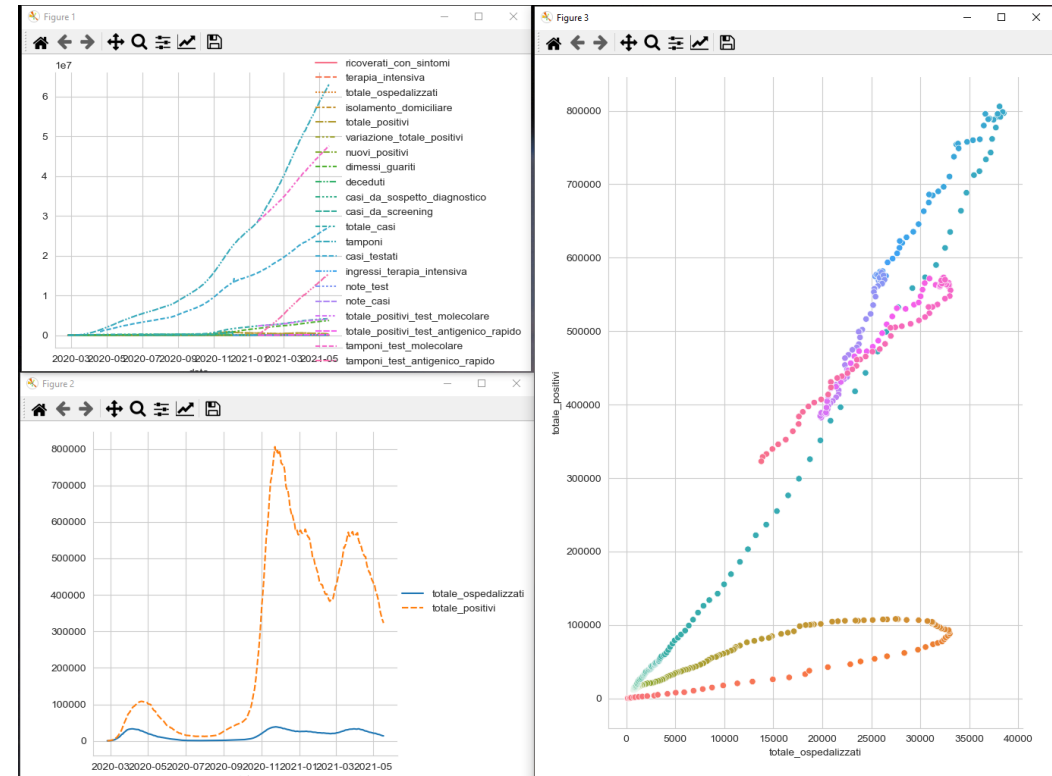
covid.set_index('data', inplace=True)

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

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

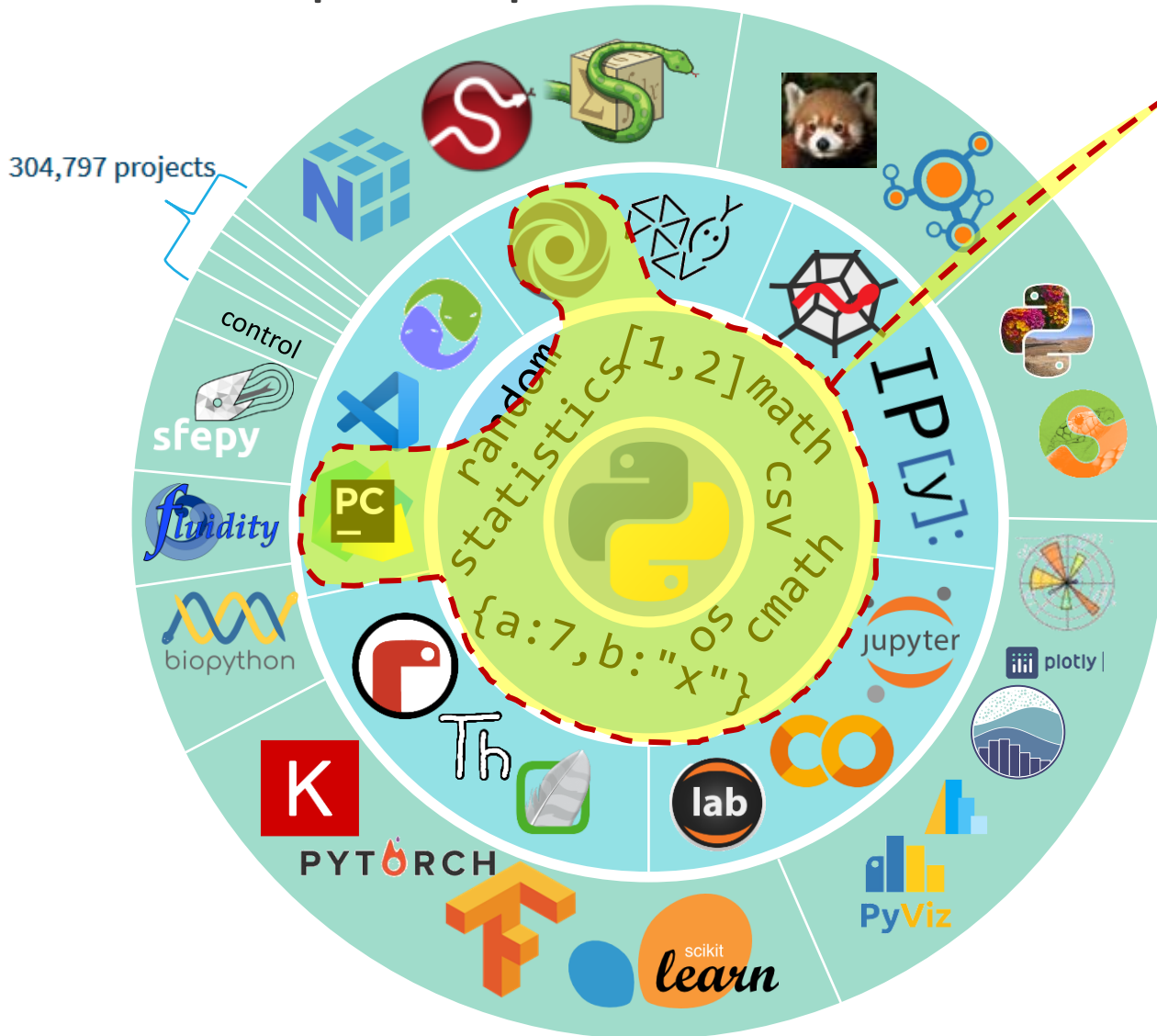
sns.relplot(data=dati_utili, kind='line')

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



Try me on Google Colab

# Dopo il primo anno...



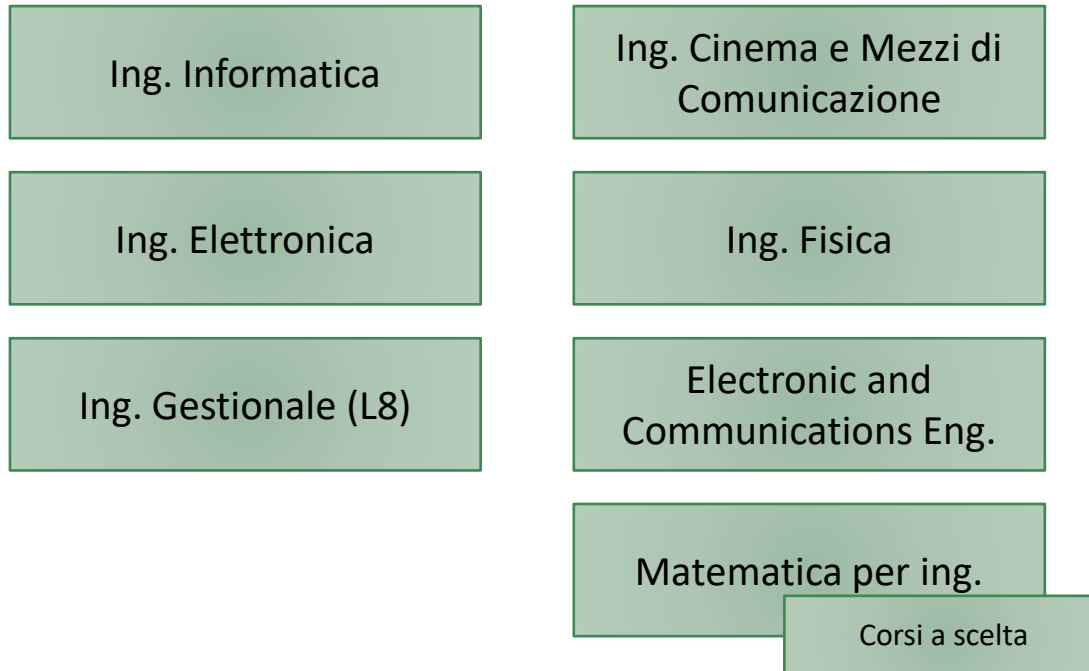
- Programma del corso di Informatica
  - Conoscenza di base del linguaggio
  - Familiarità con gli ambienti di sviluppo più semplici
  - Capacità di analizzare un problema e formulare un algoritmo
- Le specializzazioni sulle varie aree non rientrano nel programma di Informatica
  - Possibile costruire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

# Dopo «Informatica»

Ing. Informatica	Tecniche di Programmazione	Algoritmi e strutture dati	Calcolatori elettronici	Sistemi operativi	Reti di calcolatori
		Basi di dati		Programmazione a oggetti	Controlli automatici
Ing. Cinema		Algoritmi e programmazione a oggetti		Basi di dati	Reti di calcolatori
				Computer Grafica	
Ing. Gestionale		Basi di dati		Programmazione a oggetti	Progettazione di servizi web e reti di calcolatori
				Tecniche di Programmazione	(solo L8)
Ing. Elettronica	Algoritmi e programmazione				
E.C.E.				Algorithms and Programming	
	I anno	II anno		III anno	

# Dopo «Dopo «Informatica»»

## Laurea



Altre Lauree

Senza debiti

Necessaria integrazione

Percorso integrativo

## Laurea Magistrale



# Organizzazione del corso

---

<http://bit.ly/polito-informatica>

# Sito del corso

Tutto il materiale sarà disponibile su questo sito

- Slide
- Laboratori
- Esempi svolti
- Video Lezioni
- Temi d'esame
- Calendario lezioni
- ...

The screenshot shows the 'e-Lite' website interface. At the top, there is a navigation bar with links: HOME, NEWS, PEOPLE, RESEARCH, TEACHING (highlighted), THESIS, and JOBS. A search bar is located on the right. Below the navigation bar, a breadcrumb trail reads: HOME > TEACHING > CURRENT COURSES > 14BHD - INFORMATICA (CORSO 10) 2021/2022. The main heading is '14BHD - INFORMATICA (CORSO 10) 2021/2022', with a subtext 'Last Updated: 23 September 2021'. Below this, it says 'Page 1 of 7' and 'Pagina ufficiale del corso di Informatica (codice 14BHD, corso 10, cognomi GUL-LIZ, anno accademico 2021/2022) presso il Politecnico di Torino.' A link is provided: 'Link breve a questa pagina: <http://bit.ly/polito-informatica>'. The 'AVVISI' section contains a list of announcements, including dates and instructions for Slack, Zoom, and Python installation. On the right, an 'ARTICLE INDEX' sidebar lists: 14BHD - Informatica (corso 10) 2021/2022, Registro delle lezioni, Materiale, Laboratorio, Contatto con i docenti, Esame, from python import...more, and All Pages.

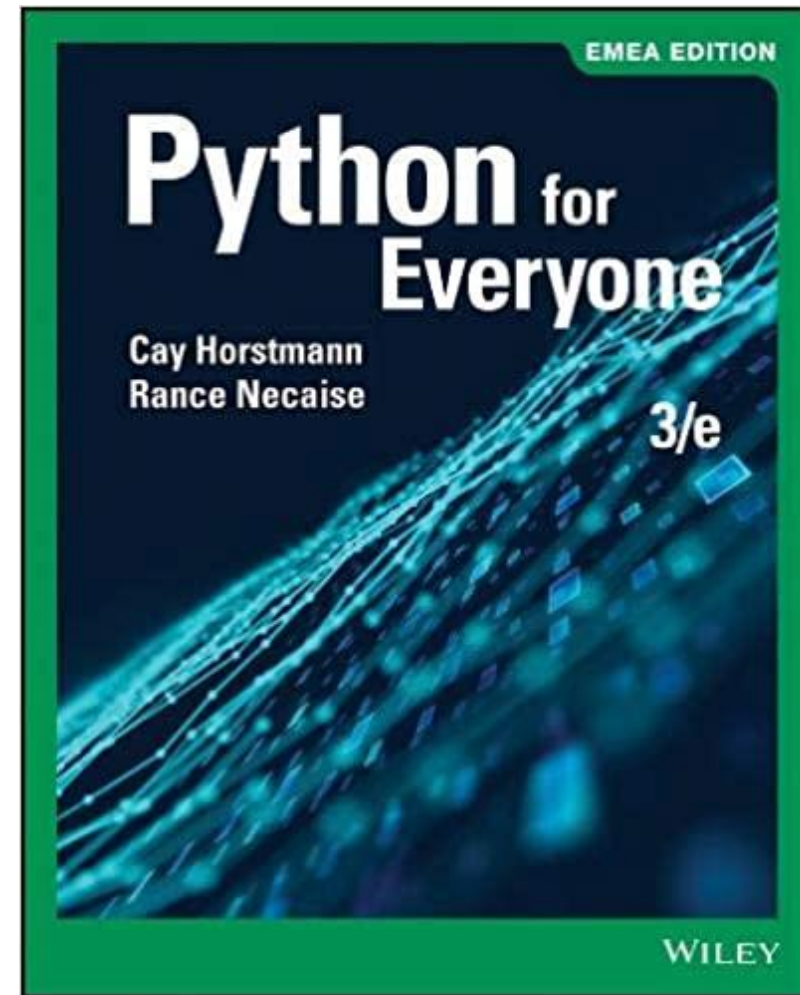
# Struttura del corso

- Programmazione e Python : 41 ore
- Teoria: 9 ore
- Problem solving : 12 ore
- Laboratorio : 18 ore (x 3 squadre)
- Totale: 80 ore/studente

The screenshot shows the e-Lite website interface for the course 14BHD - INFORMATICA (CORSO 4) 2020/2021. The page is titled "REGISTRO DELLE LEZIONI" and includes a sidebar with navigation links like "Home", "News", "People", "Research", "Teaching", "These", and "Jobs". The main content area displays a table of lessons with columns for Date, Title, Type, Argument, Materials, and Document. The table lists various topics including Introduction to Python, Problem Solving, and Laboratory exercises. The table is organized into three main sections: "Introduction to Python", "Problem Solving", and "Laboratory". Each section contains a list of lessons with their respective dates, titles, types, arguments, materials, and documents. The table is paginated, showing 10 lessons per page.

Date	Title	Type	Argument	Materials	Document
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course
2020/03/03	2020/03/03	Laboratory	Introduction to Python		Public Course

# Libri di testo

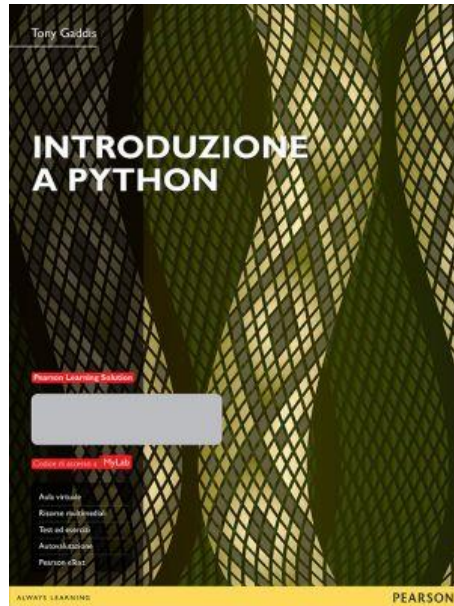




# Libri: Informazioni dettagliate

- Concetti di informatica e fondamenti di Python
- Seconda edizione - Giugno 2019 (II° Edizione)
- Cay Horstmann - Rance D. Necaise
- Maggioli Editore
- ISBN 9788891635433
- <http://www.apogeoeducation.com/concetti-di-informatica-e-fondamenti-di-python.html>
- 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>

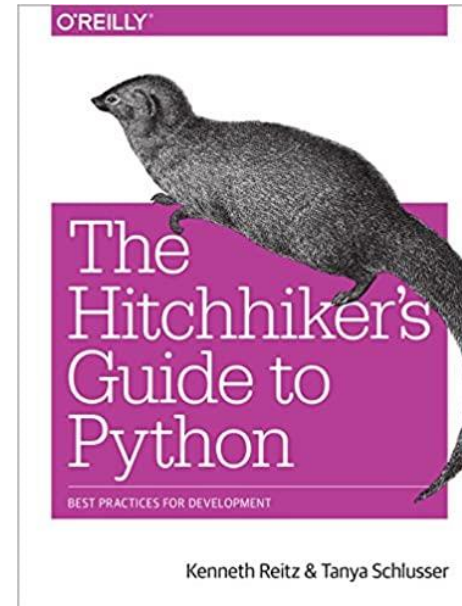
# Altre risorse



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/>



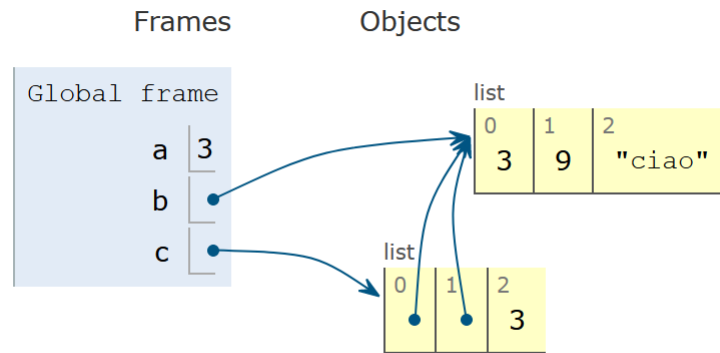
<https://realpython.com/>

# Strumenti per programmare

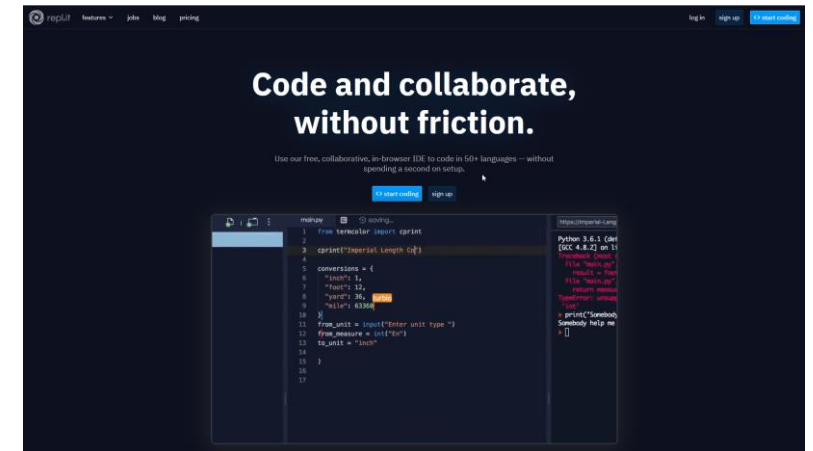


<https://www.jetbrains.com/pycharm/>

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



<http://pythontutor.com>



<https://replit.com>  
(free online IDE)

# Istruzioni di installazione

148HDxx - 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 ( <b>opzione consigliata</b> ) .....	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

128HDxx - Informatica

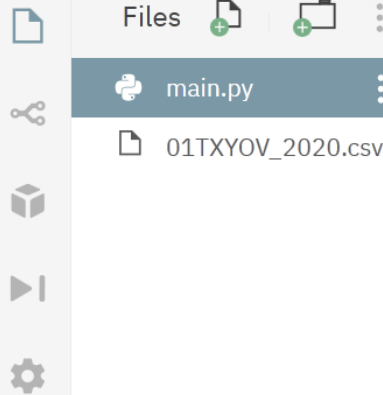
## Installazione software (piattaforma macOS)

### 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>





```

1 import csv
2 # from matplotlib import pyplot
3
4 FILENAME = '01TXYOY_2020.csv'
5
6 # Leggi l'elenco degli studenti e salvalo in un array
7 def leggi(nomefile):
8     file = open(nomefile, 'r')
9     reader = csv.reader(file)
10    prima = True
11    studenti = []
12    for line in reader:
13        if prima: #skip first line (headers)
14            prima = False
15        else:
16            studenti.append(line)
17    file.close()
18    return studenti
19
20 # estrai i nomi di battesimo da un elenco di studenti
21 def nomi(elenco):
22     nomi = []
23     for riga in elenco:
24         nomi.append(riga[2])
25     return nomi
26
27 # Calcola le frequenze dei vari nomi presenti in un
array
28 def frequenze(tokens):
29     freq = {}
30     for token in tokens:
31         if token in freq:
32             freq[token]= freq[token]+1
33         else:
34             freq[token] = 1

```

```
Nella classe ci sono 180 studenti
Il nome più frequente compare 9 volte
Si tratta di : ['ALESSANDRO']
I nomi che compaiono più di una volta sono ALESSANDRO, ANDREA
, CLAUDIO, DAVIDE, ENRICO, ETTORE, FEDERICA, FEDERICO, FRANCE
SCA, FRANCESCO, GABRIELE, GIANLUCA, GIOVANNI, GIUSEPPE, LOREN
ZO, LUCA, MARCO, MARTINA, MATTEO, MATTIA, MICHELE, PIETRO.
```

<https://replit.com>  
(free online IDE)

# Informazioni pratiche

---

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

# Orario Settimanale

---



# 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: **05/10/2022**
- Testo pubblicato sul sito del corso
- È necessario installare il software richiesto (Python e PyCharm) - vedere le istruzioni di installazione sul portale

# Suddivisione in squadre

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

# 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/+WikfXJT94b44ZDQ0>

# Esame

---

# Contenuti dell'esame

- Alcune domande brevi sulla parte teorica del corso
- Un esercizio di programmazione
  - 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

# 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

# Contatti

---

# Docenti

- Fulvio Corno

- Dipartimento di Automatica e Informatica (3° piano)
- [fulvio.corno@polito.it](mailto:fulvio.corno@polito.it)

- Roberta Bardini

- Dipartimento di Automatica e Informatica (2° piano)
- [roberta.bardini@polito.it](mailto:roberta.bardini@polito.it)

- Ci trovate su Telegram!



# Link utili



- Sito del corso (ufficiale):

- <http://elite.polito.it/> → Teaching → Current Courses → 14BHD - Informatica
- Link breve: <http://bit.ly/polito-informatica>

- Gruppo Telegram

- <https://t.me/+WikfXJT94b44ZDQ0>

- Materiale, laboratori, esercizi

- <https://github.com/polito-info-2022>

# 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/>

