

20 FEBBRAIO 2025

Benvenuti

Chi sono

Nicolò Trebino

Diplomato nel 2021 al Liceo scientifico Da Vigo Nicoloso

Laureato nel 2024 in ingegneria informatica a Genova

Studente magistrale in Computer Engineering a Genova



Contatti



nicolo.trebino@gmail.com



nikytrebino



nicolotrebino

Parlatemi un po' di voi

- Classe e indirizzo
- Perché avete scelto questo corso
- Idee su cosa fare dopo la scuola
- Curiosità su università
- Attuali conoscenze di programmazione

Moodle

<https://www.davigonicoloso.edu.it/moodle/course/view.php?id=164>

Informatica Rapallo - TrebinoN

Dashboard / I miei corsi / INFORA2425

Attiva modifica

Navigazione

- › Dashboard
 - Home del sito
 - > Pagine del sito
- › I miei corsi
 - › INFORA2425
 - > Partecipanti
 - Badge
 - Competenze
 - Valutazioni
 - › Generale
 - Annunci
 - > Lezione 1
 - > Lezione 2
 - > Lezione 3
 - > Lezione 4
 - > Lezione 5
 - > Lezione 6

› Generale

Minimizza tutto

Benvenuti al corso di programmazione Python

Orario e Argomenti

Orario	Data	Argomento
15:00 - 17:00	Giovedì 20/02	Introduzione all'informatica e a Python
15:00 - 17:00	Lunedì 24/02	Variabili e operatori
15:00 - 17:00	Giovedì 06/03	Condizioni if else, tipi complessi
15:00 - 17:00	Lunedì 10/03	Ciclo while e ciclo for
15:00 - 17:00	Giovedì 20/03	Dizionari, funzioni e import
15:00 - 17:00	Lunedì 24/03	Progetto

Il corso si deve rivolgere a studenti senza
alcuna conoscenza informatica
e puntare sullo sviluppo del pensiero
algoritmico.

Cosa faremo

Introduzione all'informatica e alla programmazione

Il corso mira a introdurre gli studenti ai concetti base dell'informatica, partendo dalle fondamenta dei linguaggi di programmazione e del pensiero algoritmico.

Attraverso attività pratiche e interattive, si guideranno i partecipanti nella comprensione di come strutturare problemi e sviluppare soluzioni logiche, utilizzando linguaggi di programmazione semplici e accessibili.

L'obiettivo è stimolare curiosità e creatività, aprendo gli studenti alle opportunità offerte dall'era digitale e fornendo loro strumenti per affrontare con consapevolezza e spirito critico il mondo tecnologico in continua evoluzione.

Cosa faremo

Struttura del corso

01

Lezione 1
Giovedì 20/02

- Presentazioni
- Introduzione all'informatica
- Python
- Setup e introduzione a Jupyter Notebook

02

Lezione 2
Lunedì 24/02

- Variabili
- Tipi di variabile
- Funzioni print() e input()
- Operatori

03

Lezione 3
Giovedì 06/03

- Condizioni if else
- Tipi complessi (liste)

Struttura del corso

04

Lezione 4
Lunedì 10/03

- Ciclo while
- Ciclo for

05

Lezione 5
Giovedì 20/03

- Dizionari
- Funzioni
- Import

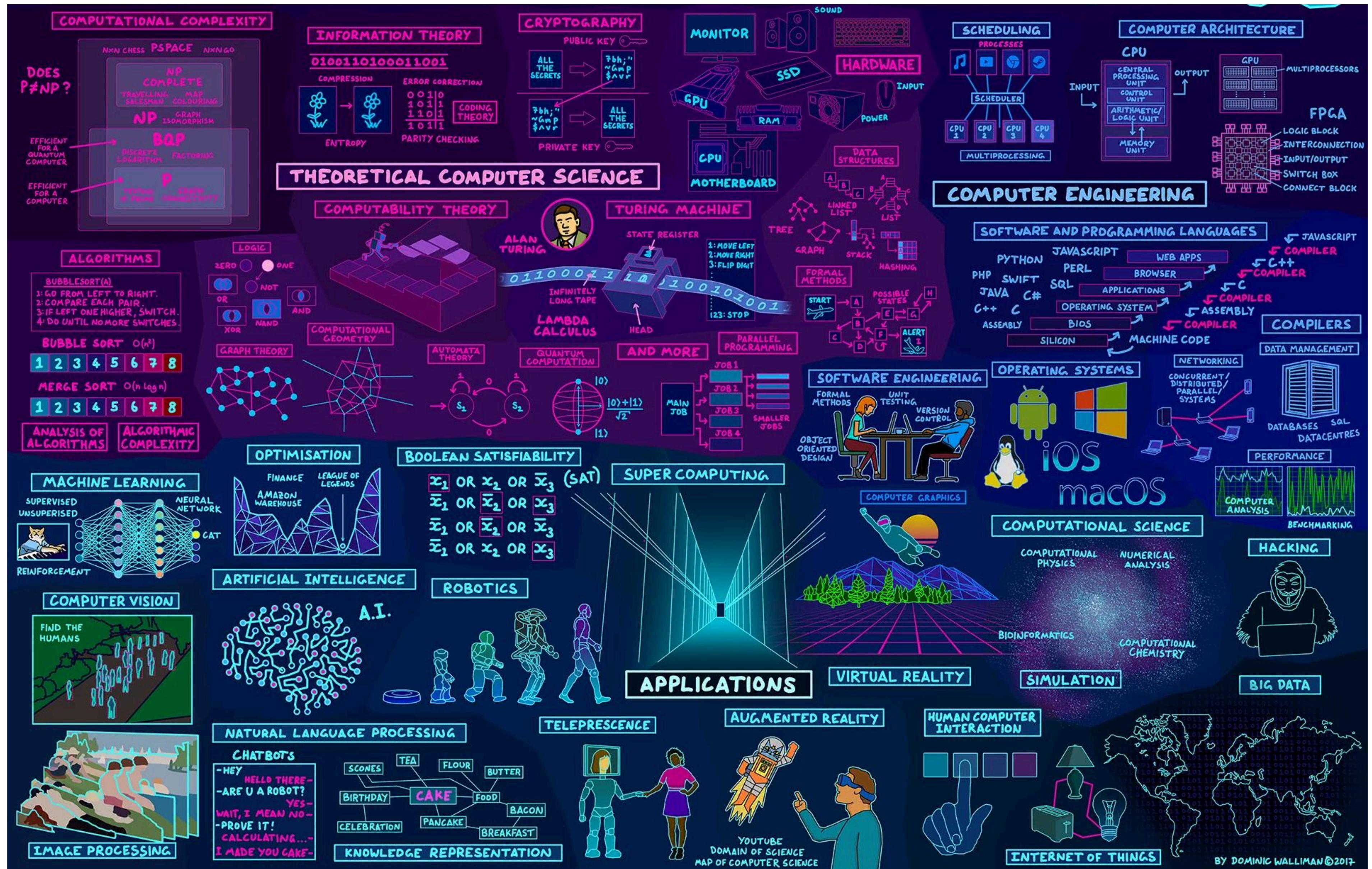
06

Lezione 6
Lunedì 24/03

- Implementazione di un progetto completo in Python con tutte le istruzioni, strumenti e strutture analizzate nel corso

Cosa vi viene in mente quando
pensate alla parola informatica?

Introduzione all'informatica



Mappa dell'informatica

Informatica

La parola informatica deriva da due termini:

INFORmazione autoMATICA.

Disciplina scientifica che si occupa dello studio dell'elaborazione delle informazioni e delle sue applicazioni pratiche.

Più nel dettaglio, essa si concentra sulla rappresentazione, organizzazione e gestione automatica delle informazioni.

<https://it.wikipedia.org/wiki/Informatica>

Informatica

La parola informatica deriva da due termini:
INFORmazione autoMATICA.

Disciplina scientifica che si occupa dello studio dell'elaborazione delle informazioni e delle sue applicazioni pratiche.

Più nel dettaglio, essa si concentra sulla rappresentazione, organizzazione e gestione automatica delle informazioni.

<https://it.wikipedia.org/wiki/Informatica>

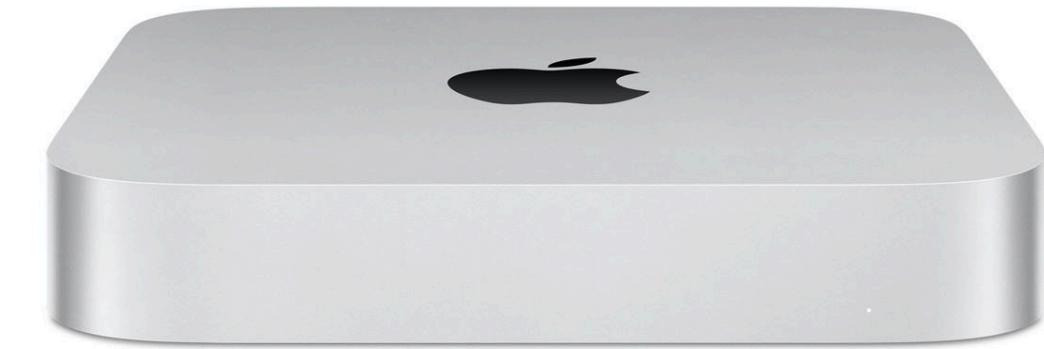


Informatica

La parola informatica deriva da due termini:
INFORmazione autoMATICA.



Personal computer (PC)



Mini computer (MAC mini)



Super computer (LEONARDO)

Cosa è un computer?

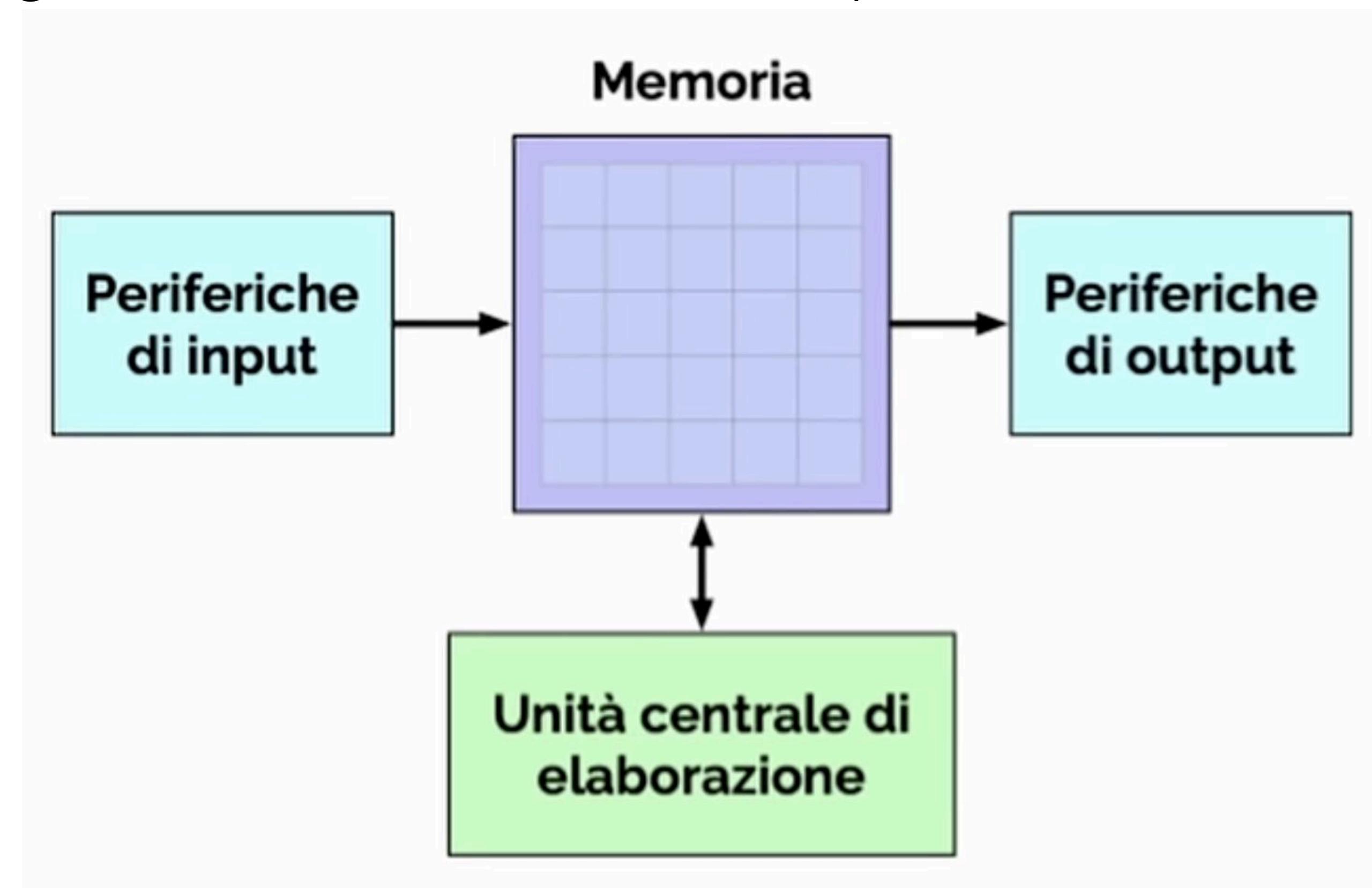
Elaboratore elettronico digitale

- **Elaboratore:** in grado di immagazzinare ed elaborare dati in base a una serie di istruzioni, il programma
- **Elettronico:** evidentemente utilizza componenti elettronici :-)
- **Digitale:** elabora informazioni convertendole in segnali digitali basati sul sistema binario (che non approfondiamo)

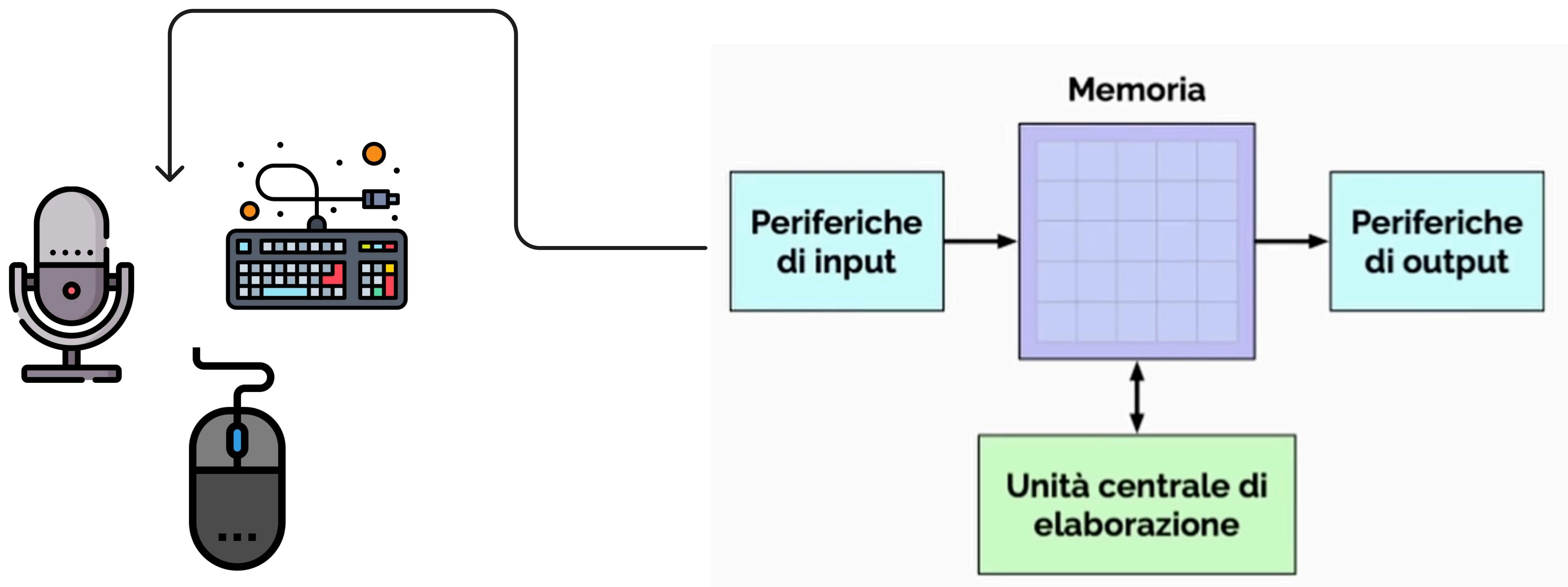
Architettura (Hardware)

1945, Von Neumann

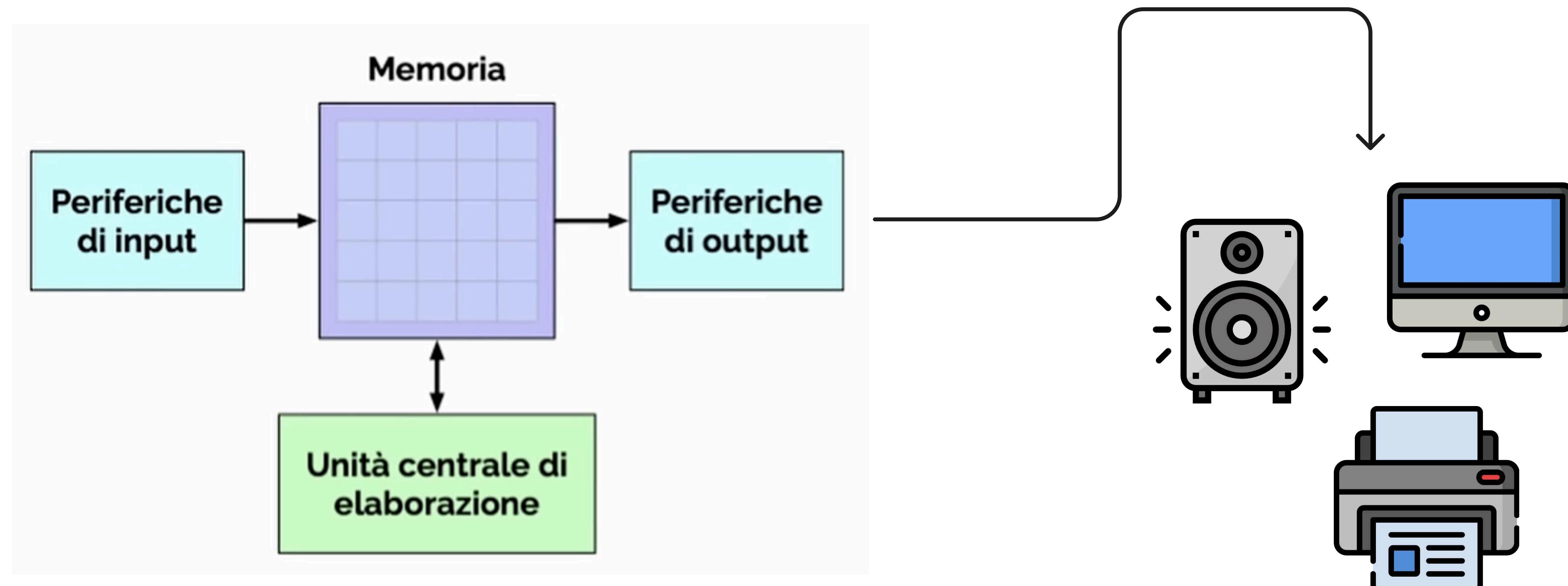
I dati e le istruzioni del programma sono memorizzati nello stesso spazio di memoria.



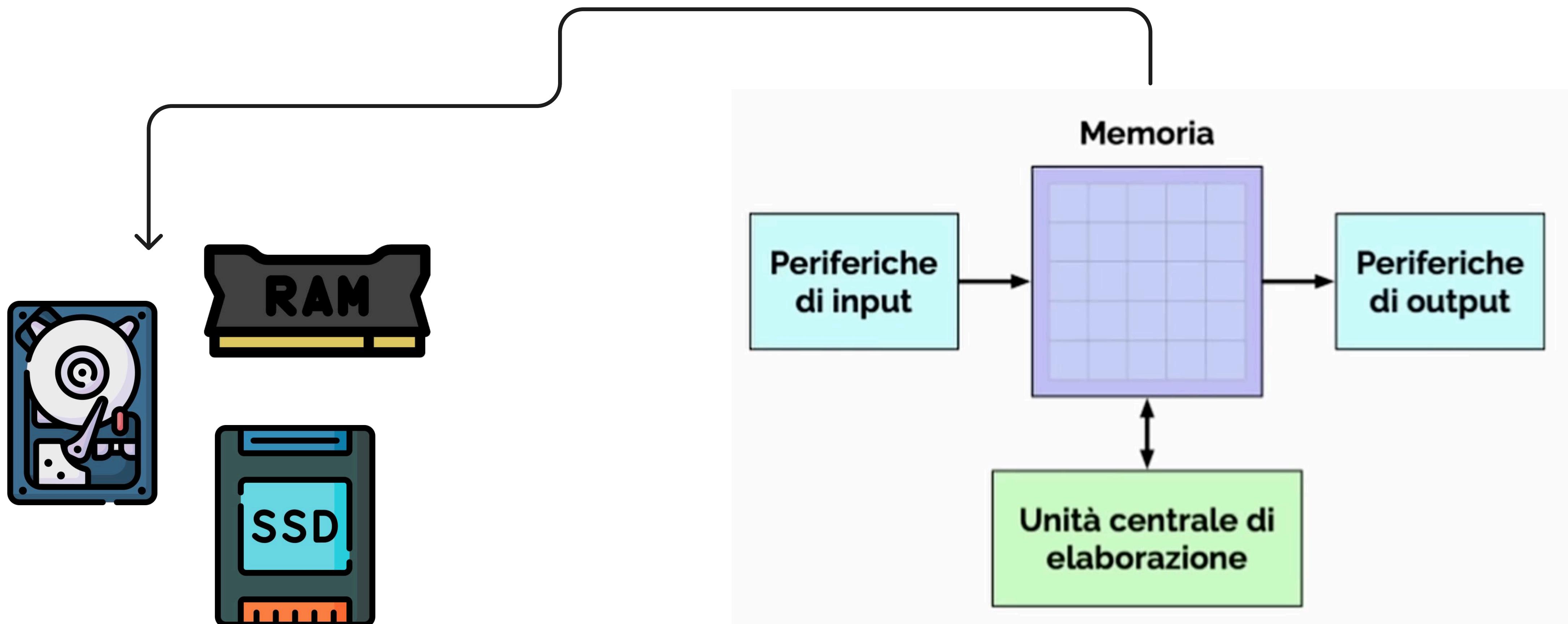
Architettura (Hardware)



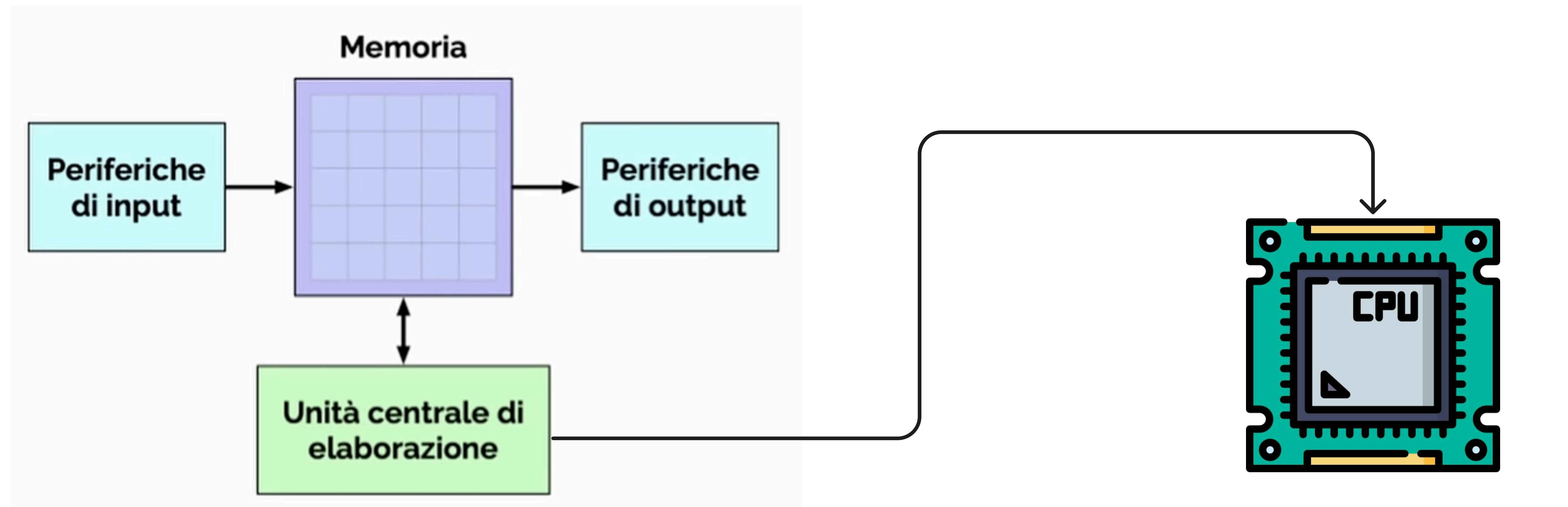
Architettura (Hardware)



Architettura (Hardware)



Architettura (Hardware)



Linguaggi

Gli elaboratori ragionano diversamente rispetto a noi.

Sono **rigorosi** e **formali** ed è necessario utilizzare un linguaggio appropriato per impartire loro istruzioni.

Noi comunichiamo sfruttando il **linguaggio naturale** (italiano, inglese, giapponese ecc.).

Per impartire istruzioni agli elaboratori bisogna utilizzare un **linguaggio formale**.

Tutti i **linguaggi di programmazione** sono **linguaggi formali**.

Linguaggi

Linguaggio naturale

“Metti il libro sul tavolo vicino alla porta.”

“Nella gara è arrivato ultimo, è proprio una lumaca.”

- La prima frase è interpretabile in due modi diversi.
- La seconda frase possiede una metafora.

Il **linguaggio naturale** quindi è **soggetto ad interpretazioni**: condizione non auspicabile se vogliamo comunicare con una macchina!

Linguaggi

Linguaggio formale

$$x = 10$$

Il **linguaggio formale** ha uno schema e regole rigide, non lascia spazio a interpretazioni ma definisce univocamente il significato di ogni istruzione.

Le macchine possono così *leggerlo* (**interpretarlo, compilarlo, eseguirlo**) con precisione.

Programma (Software)

Quindi programmare significa **impartire alla macchina delle istruzioni in un linguaggio formale per risolvere determinati problemi**.

Un **algoritmo** è un insieme di operazioni necessarie per risolvere un problema (le istruzioni di un LEGO, una ricetta).

Un **programma** è l'implementazione di un algoritmo in un linguaggio formale.
Un algoritmo ha infinite implementazioni in diversi programmi.

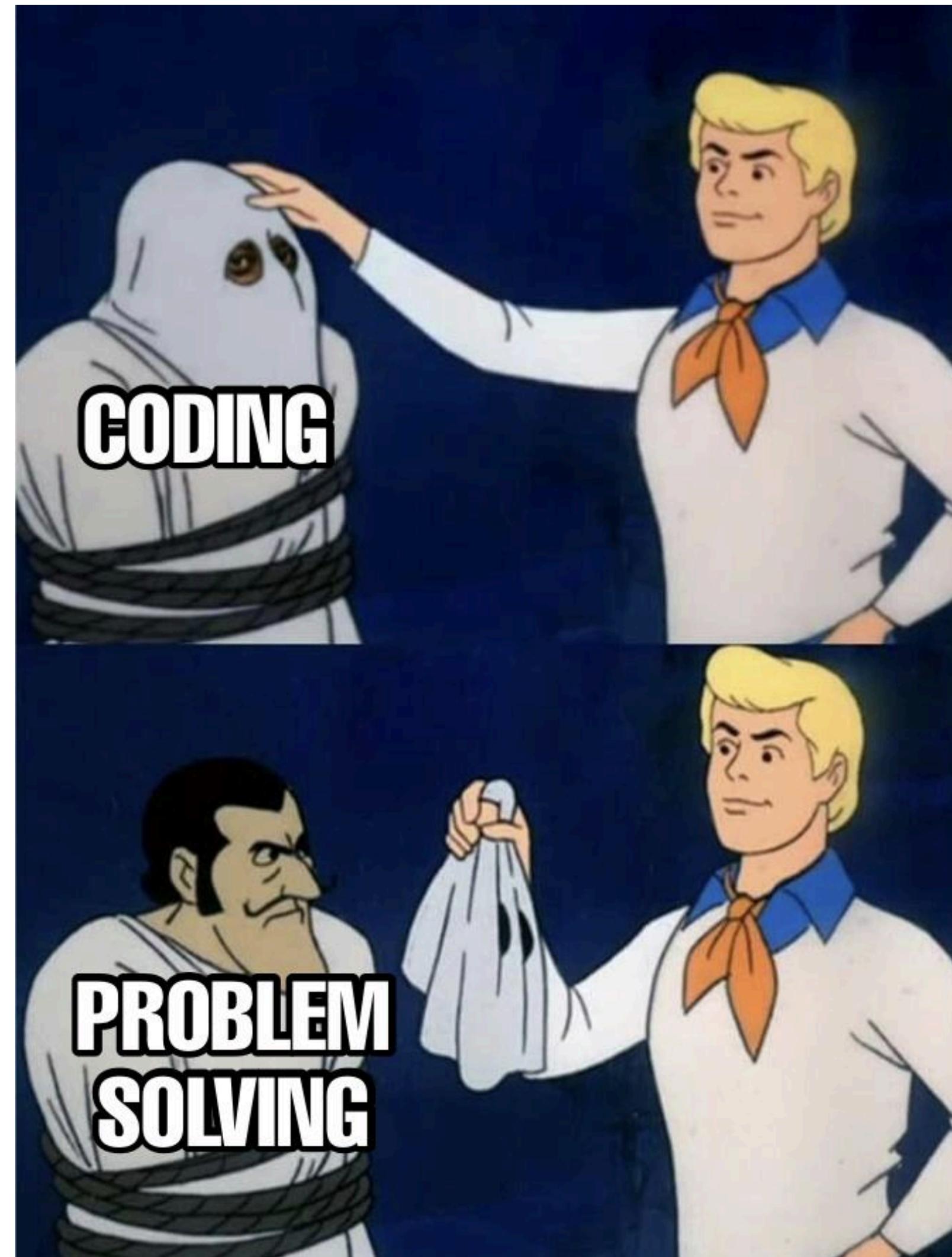
Perché programmare?

Programmare è una **skill importante**.

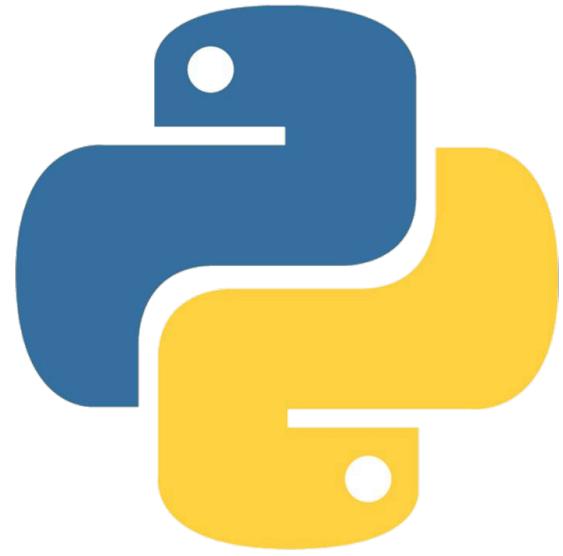
Utile, aiuta a creare applicazioni o automatizzare processi della vita reale.

Aiuta ad acquisire una mentalità **rigorosa** e si impara a **risolvere problemi**.

È come fare la settimana enigmistica ma con il computer!

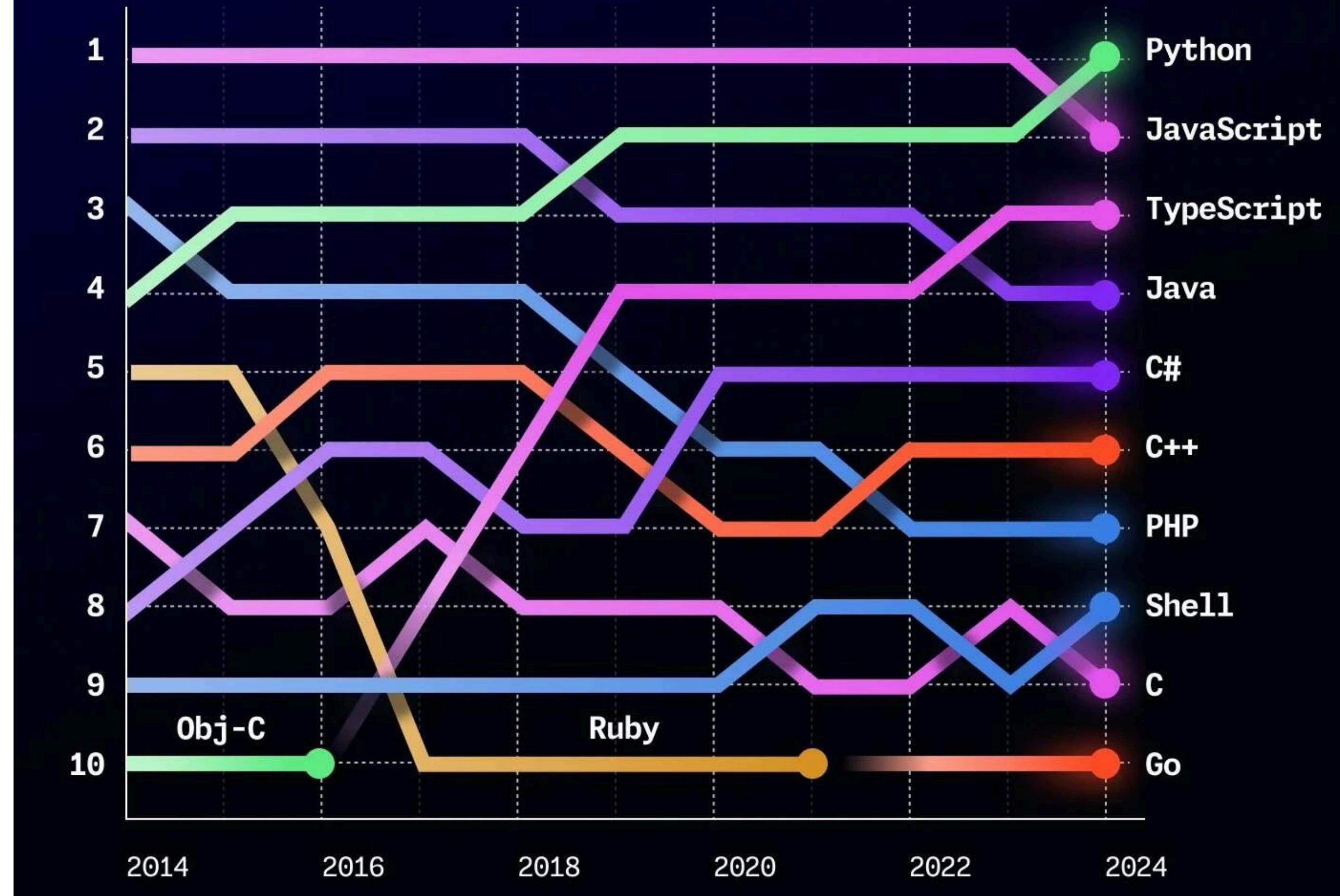


Python



Top programming languages on GitHub

RANKED BY COUNT OF DISTINCT USERS CONTRIBUTING TO PROJECTS OF EACH LANGUAGE.



Perché Python?

Python

Python For Everything

Python + Pandas = Data Manipulation

Python + Scikit-Learn = Machine Learning

Python + Tensorflow = Deep Learning

Python + Matplotlib = Data Visualization

Python + Seaborn = Adv. Visualization

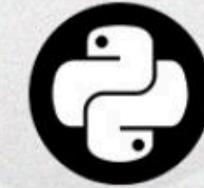
Python + Flask = Web Development

Python + Pygame = Game Development

Python + Kivy = Mobile App Development

Perché in 10 anni è diventato così famoso e utilizzato?

Python



Python For Everything

Python	+	Pandas	=	Data Manipulation
Python	+	Scikit-Learn	=	Machine Learning
Python	+	Tensorflow	=	Deep Learning
Python	+	Matplotlib	=	Data Visualization
Python	+	Seaborn	=	Adv. Visualization
Python	+	Flask	=	Web Development
Python	+	Pygame	=	Game Development
Python	+	Kivy	=	Mobile App Development

Perché in 10 anni è diventato così famoso e utilizzato?

Python

Python

Caratteristiche principali

- Semplice da imparare e usare
- Ottimo come primo linguaggio (quasi come parlare in inglese)

```
// C++

#include <iostream> // Per input/output

int main() {
    int somma = 0; // Variabile per la somma

    for (int i = 1; i <= 10; i++) { // Genera i numeri da 1 a 10
        somma += i; // Aggiungi i alla somma
    }

    std::cout << "La somma dei numeri da 1 a 10 è: " << somma << std::endl;

    return 0; // Indica che il programma è terminato correttamente
}
```

```
# PYTHON

somma = 0 # Variabile per la somma

for i in range(1, 11): # range(1, 11) genera i numeri da 1 a 10
    somma += i # Aggiungi i alla somma

print("La somma dei numeri da 1 a 10 è: {somma}")
```

Python

Hello World!

```
1 // C++
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     cout << "Hello world!";
8     return 0;
9 }
```

Python

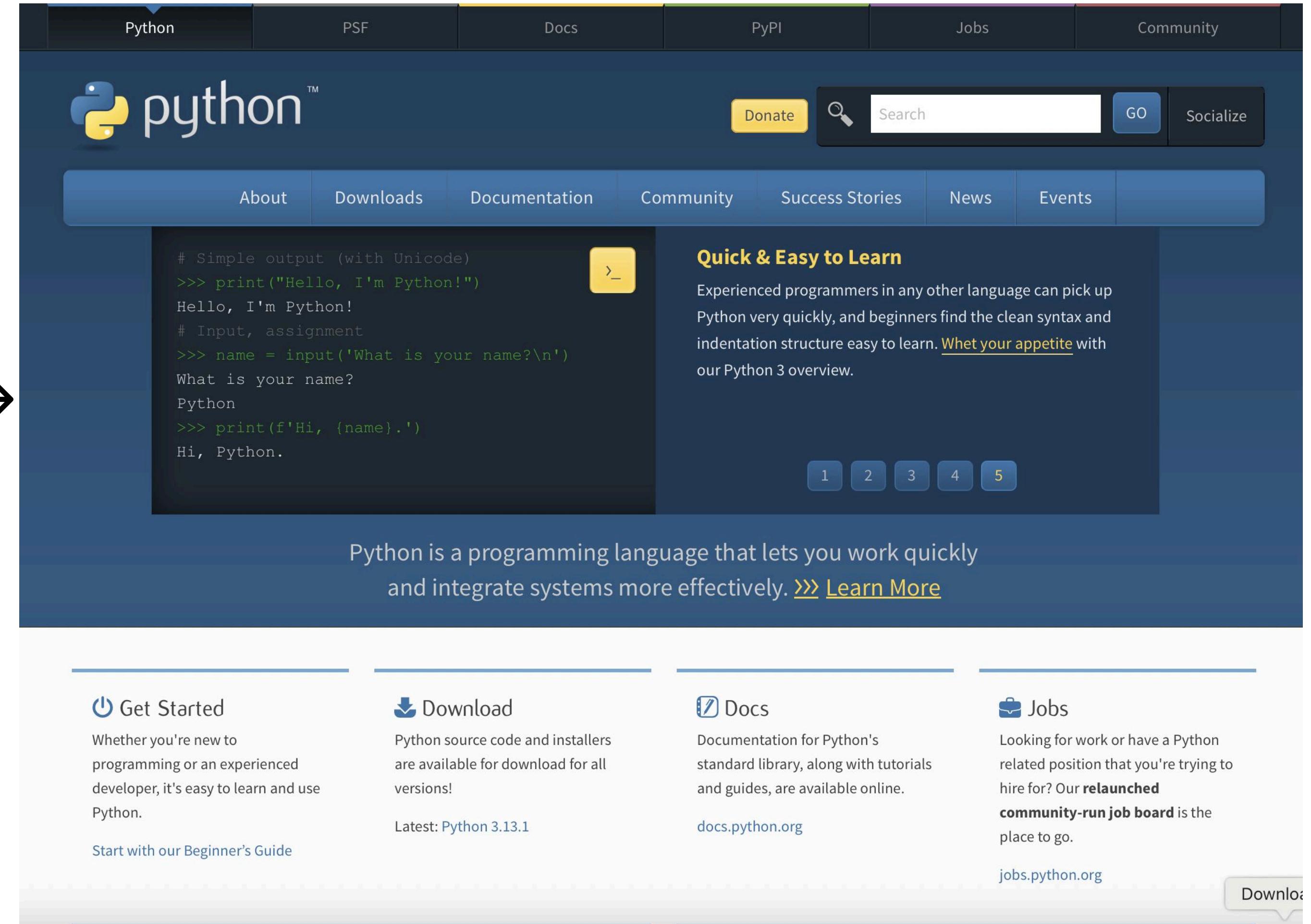
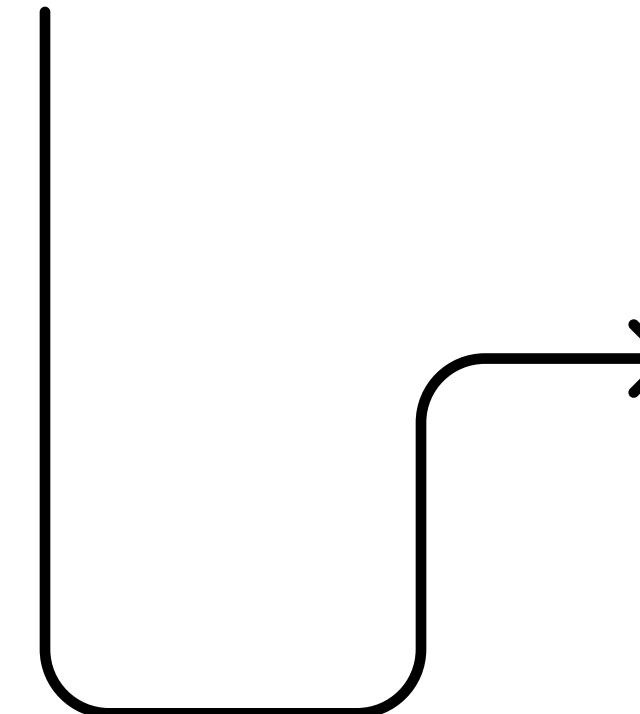
Hello World!

```
1 # PYTHON
2
3 print("Hello world!")
4
```



Caratteristiche principali

- È **open source** (www.python.org)



The screenshot shows the official Python website (www.python.org). The header features the Python logo and the word "python" in white. The navigation bar includes links for Python, PSF, Docs (which is highlighted), PyPI, Jobs, and Community. Below the header is a search bar and a "Quick & Easy to Learn" section with a code snippet demonstrating basic Python syntax. The main content area contains a "About" section with a code example:

```
# Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!
# Input, assignment
>>> name = input('What is your name?\n')
What is your name?
Python
>>> print(f'Hi, {name}.')
Hi, Python.
```

Below this, a banner states: "Python is a programming language that lets you work quickly and integrate systems more effectively. [» Learn More](#)". The footer contains sections for Get Started, Download, Docs, and Jobs, each with a brief description and a link to further information.

Python

Caratteristiche principali

- È un linguaggio **interpretato**

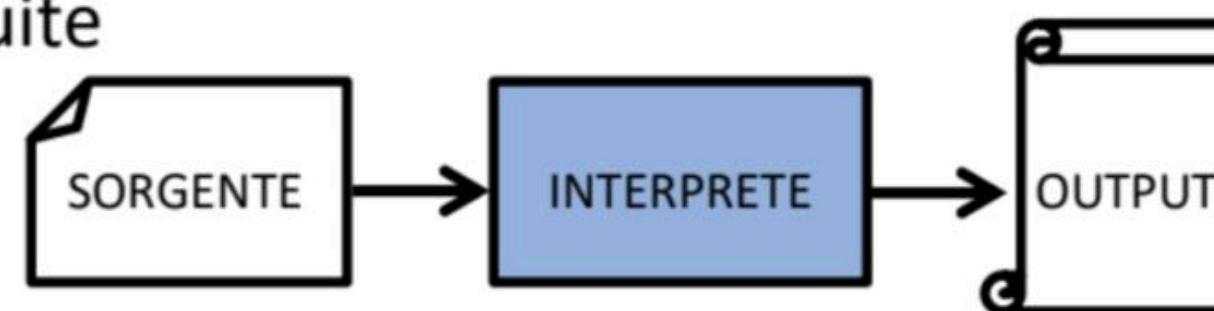
Linguaggi compilati: es. C, C++

- Il programmatore scrive il **codice sorgente** in un file di testo
- Il **compilatore** traduce il codice sorgente e produce un **codice eseguibile** (linguaggio macchina)
- L'esecutore carica il codice eseguibile nella memoria del computer e lo esegue



Linguaggi interpretati: es. Python

- Il programmatore scrive il **codice sorgente** in un file di testo
- L' **interprete** traduce (ad ogni esecuzione) il codice sorgente, trasformandolo in istruzioni del linguaggio macchina che vengono eseguite

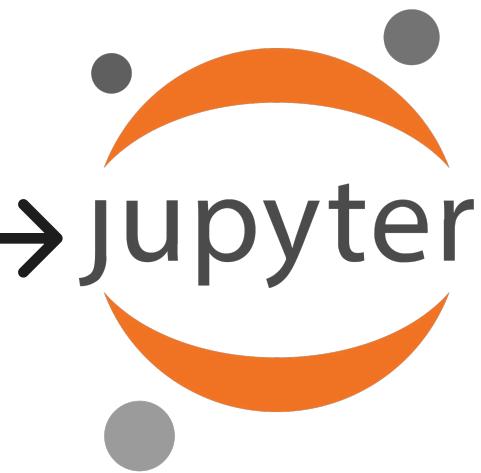


Python

2 modalità

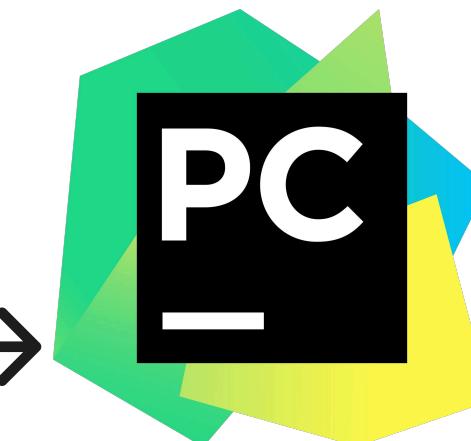
- **IPython**, offre un quaderno (notebook) interattivo che permette di scrivere testo e codice interpretato.

I file hanno estensione **.ipynb**



- **Python scripts**, utile per creare veri e propri progetti ed eseguire molte linee di codice tutte insieme.

I file hanno estensione **.py**



Setup

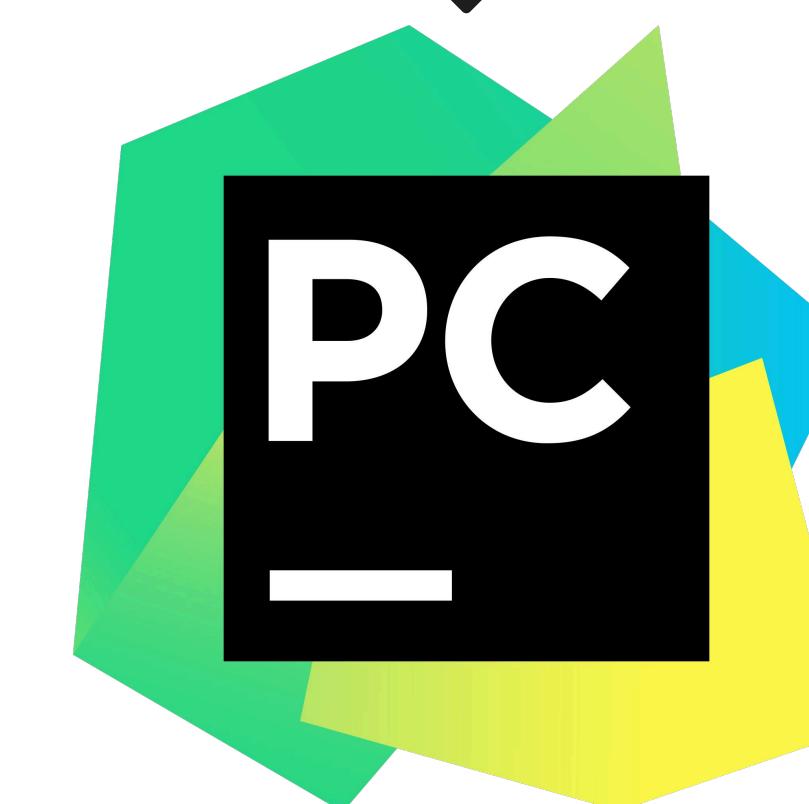
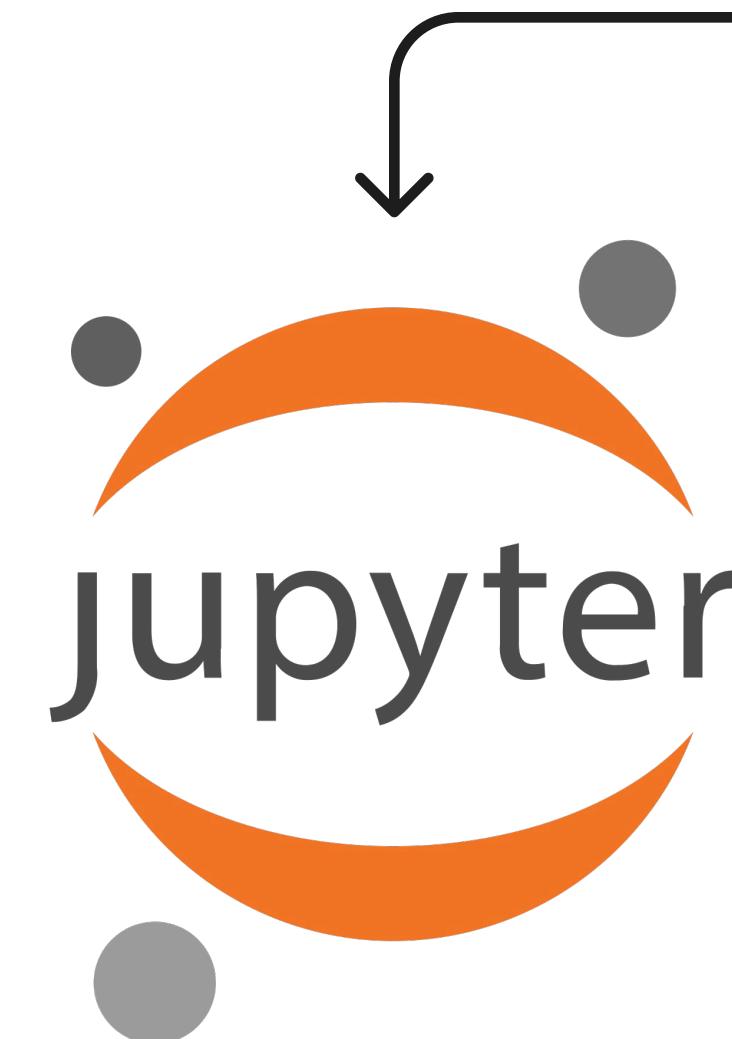


Setup



- Mette a disposizione degli utenti una distribuzione open source di Python
- È un gestore di pacchetti, librerie e ambienti di sviluppo per Python
- Possiede un'interfaccia grafica
- È disponibile per tutti i sistemi operativi (Windows, Mac, Linux)

Setup



Anaconda download



Products Solutions Resources Partners Company

Distribution

Free Download*

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, AI Assistant, Learning and more.

- ✓ Easily search and install thousands of data science, machine learning, and AI packages
- ✓ Manage packages and environments from a desktop application or work from the command line
- ✓ Deploy across hardware and software platforms
- ✓ Distribution installation on Windows, MacOS, or Linux

*Use of Anaconda's Offerings at an organization of more than 200 employees requires a Business or Enterprise license. [See Pricing](#)

Free Download Sign Up Sign In

Provide email to download Distribution

Email Address:

Agree to receive communication from Anaconda regarding relevant content, products, and services. I understand that I can revoke this consent [here](#) at any time.

By continuing, I agree to Anaconda's [Privacy Policy](#) and [Terms of Service](#).

Submit >

Skip registration

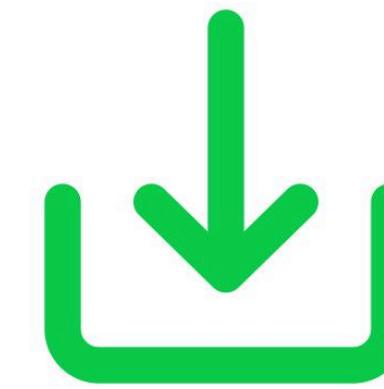
<https://www.anaconda.com/download>

Anaconda download

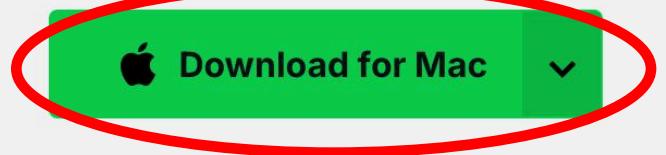
Download Now

For installation assistance, refer to [Troubleshooting](#).

Download Anaconda Distribution or [Miniconda](#) by choosing the proper installer for your machine. Learn the difference from our [Documentation](#).



Anaconda Installers

[Download for Mac](#) 



Windows

Python 3.12

 64-Bit Graphical Installer (912.3M)



Mac

Python 3.12

-  64-Bit (Apple silicon) Graphical Installer (704.7M)
-  64-Bit (Apple silicon) Command Line Installer (707.3M)
-  64-Bit (Intel chip) Graphical Installer (734.7M)
-  64-Bit (Intel chip) Command Line Installer (731.2M)



Linux

Python 3.12

-  64-Bit (x86) Installer (1007.9M)
-  64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)
-  64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)

<https://www.anaconda.com/download>

Aprire Anaconda

Connected to Cloud [Connect ▾](#)

ANACONDA NAVIGATOR

All applications on base (root) Channels

The screenshot shows the Anaconda Navigator application interface. On the left is a sidebar with links for Home, Environments, Learning, Community, Documentation, and Anaconda Blog, along with social media icons for X, GitHub, YouTube, and LinkedIn. The main area displays a grid of application cards. Two specific applications are highlighted with red circles: 'PyCharm Professional' (version 2024.3.1.1) and 'Jupyter Notebook' (version 7.2.2). Both cards include a 'Launch' button. Other visible applications include Anaconda AI Navigator, Anaconda Toolbox, Anaconda Assistant, Anaconda Cloud Notebooks, JupyterLab, VS Code, EduBlocks, Watsonx, Oracle Cloud Infrastructure, IBM Watsonx, Oracle Data Science Service, PvScript, and PythonAnywhere.

Application	Description	Version	Action
PyCharm Professional	The Python IDE for data science. It combines the interactivity of Jupyter notebooks with intelligent Python coding assistance, Anaconda support, and scientific libraries.	2024.3.1.1	Launch
Anaconda AI Navigator	Access various large language models (LLMs) curated by Anaconda, and start leveraging secure local AI today.		Install
Anaconda Toolbox	JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.	4.1.0	Launch
Anaconda Cloud Notebooks	Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.		Launch
JupyterLab	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	4.2.5	Launch
Jupyter Notebook	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	7.2.2	Launch
VS Code	Streamlined code editor with support for development operations like debugging, task running and version control.	1.96.2	Launch
EduBlocks	Web-based coding platform from Anaconda designed for students. Learn Python coding through an interactive, block-based visual environment.		Launch
watsonx™			
ORACLE Cloud Infrastructure			
IBM Watsonx			
Oracle Data Science Service			
PvScript			
PythonAnywhere			

Aprire Jupyter Notebook

Tramite
Anaconda

Connected to Cloud Connect ▾

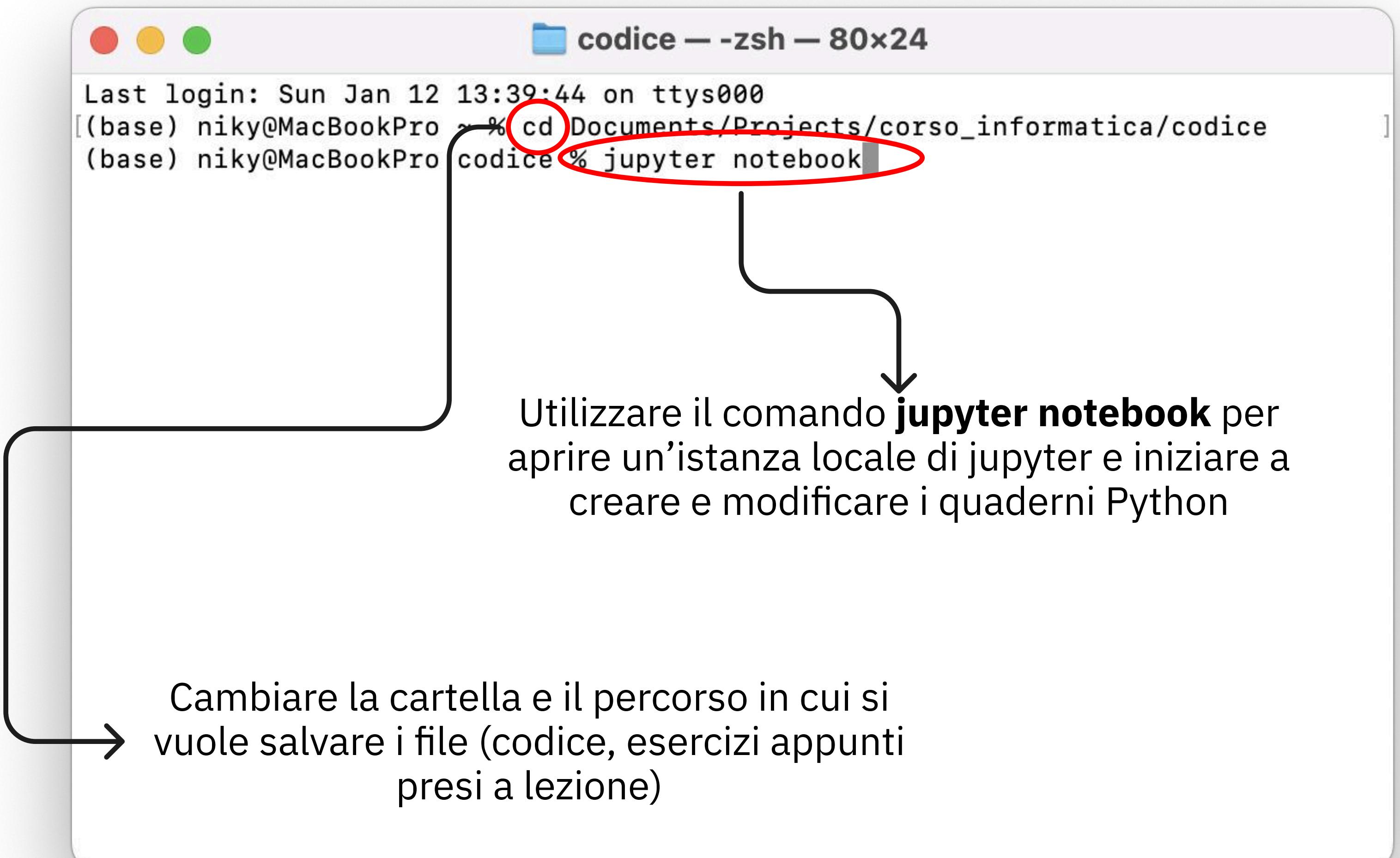
ANACONDA NAVIGATOR

All applications on base (root) Channels

Icon	Name	Version	Description	Action
PC	PyCharm Professional	2024.3.1.1	The Python IDE for data science. It combines the interactivity of Jupyter notebooks with intelligent Python coding assistance, Anaconda support, and scientific libraries.	Launch
AI	Anaconda AI Navigator		Access various large language models (LLMs) curated by Anaconda, and start leveraging secure local AI today.	Install
O	Anaconda Toolbox	4.1.0	Anaconda Assistant JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.	Launch
O	Anaconda Cloud Notebooks		Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.	Launch
lab	JupyterLab	4.2.5	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	Launch
jupyter	Notebook	7.2.2	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	Launch
VS	VS Code	1.96.2	Streamlined code editor with support for development operations like debugging, task running and version control.	Launch
edu	EduBlocks		Web-based coding platform from Anaconda designed for students. Learn Python coding through an interactive, block-based visual environment.	Launch
watsonx	IBM watsonx			
ORACLE	Oracle Data Science Service			
PvScript	PvScript			
PythonAnywhere	PythonAnywhere			

Aprire Jupyter Notebook

Tramite terminale



```
Last login: Sun Jan 12 13:39:44 on ttys000
(base) niky@MacBookPro ~ % cd Documents/Projects/corso_informatica/codice
(base) niky@MacBookPro codice % jupyter notebook
```

Cambiare la cartella e il percorso in cui si vuole salvare i file (codice, esercizi appunti presi a lezione)

Utilizzare il comando **jupyter notebook** per aprire un'istanza locale di jupyter e iniziare a creare e modificare i quaderni Python

Creazione del Notebook

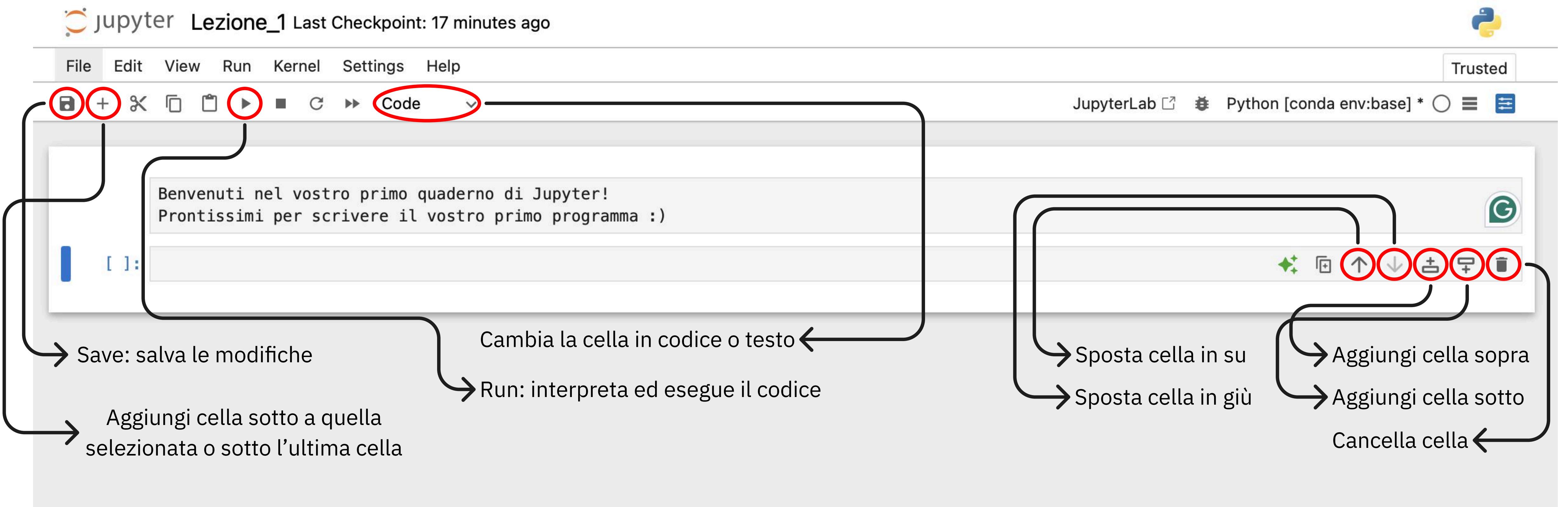
The image shows two screenshots of the Jupyter Notebook interface. The top screenshot displays a file browser window with a red circle around the path '/Documents/Projects/corso_informatica/'. The bottom screenshot shows the 'File' menu open, with the 'Notebook' option highlighted by a red circle. A large black arrow points from the 'File' menu towards the file browser, indicating the selection of a save location. Another black arrow points from the 'Notebook' option in the menu towards the 'Create un nuovo notebook all'interno del percorso scelto' text, which is enclosed in a black box.

Selezionare il percorso dove inserire tutti i file creati a lezione: codice, esercizi, appunti

Creare un nuovo **notebook** all'interno ← del percorso scelto

Aprire il Notebook

- Doppio click sul notebook appena creato per aprirlo



[Jupyter Notebook Tips And Shortcuts](#)