

Corso Pratico Python

[Modulo 1- 2026]

Linguaggio, sviluppo algoritmi,
applicazioni pratiche e analisi dati

Richiedente: Dott. Roberto Felici [CNR-ISM] • Insegnante: Dott. Scigè J. Liu [INAF-IAPS]



Obiettivi e Struttura

- **Obiettivo del Corso**

- Acquisire la capacità necessaria per la acquisizione e la sintesi di algoritmi in Python, aggiornato alle applicazioni moderne con esercitazioni pratiche ed esempi in presenza.

- **Durata**

- **Corso Base:** 20 ore + 5 ore (Recupero/Verifiche)

- **Richiedente:** Dott. Roberto Felici
[CNR-ISM]

- **Insegnante:** Dott. Scigè J. Liu
[INAF-IAPS]

- **Supplente:** Dott. Andrea Reale
[ENEA]





Logistica e Pianificazione

- **Periodo**

- Dal 16 Febbraio
al 5 Marzo 2026
- Dal 16 Marzo
al 26 Marzo 2026

- **Orari**

- **Lunedì:** 14:00 - 16:30
Giovedì: 10:00 - 12:30

- **Aula IB02/IB09**

- 5 ore a settimana
5 settimane

Disponibilità in remoto

Sarà possibile seguire da remoto.

Le lezioni saranno registrate, la frequenza asincrona è fortemente suggerita anche quale ripasso delle argomentazioni.

Domande e approfondimenti verranno discussi in due incontri di recupero.

Prerequisiti consigliati

Programma didattico in cooperazione con :

- UniToV : Prof. Dario Del Moro, et al.
- LaSapienza : Prof. Giancarlo de Gasperis

Ricetta per il successo nella programmazione (scientifica)

- **Pensare** molto **prima** di scrivere codice → creare/usare uno/più algoritmo/i
- **Usare** correttamente la **grammatica** e **sintassi** del linguaggio di programmazione (nel nostro caso Python) nello scrivere il codice sorgente
- **Usare** correttamente le regole del **linguaggio logico**
- **Ottimizzare** la procedura di calcolo adottata
- (minimizzare il numero di operazioni)

- **CONOSCENZA E CAPACITÀ DI COMPrensIONE:**
Conoscenza basilare della struttura di un calcolatore; Conoscenza degli operatori basilari nella computazione numerica; Aver acquisito conoscenze di base per la progettazione di algoritmi. Conoscere e comprendere la terminologia propria del calcolo numerico; Conoscenza di semplici algoritmi per Calcolo Matriciale, Ricerca di Radici Semplici ed Integrazione numerica
- **Rappresentazione** di numeri interi e reali, conversione tra basi numeriche, formati IEEE 754 per Floating Point, numeri macchina.
- Errori numerici di rappresentazione: assoluti e relativi.
- Cenni di struttura e funzionamento di un moderno calcolatore (CPU, RAM, ROM, hard-disk, OS, etc). Descrizione dell'ambiente UNIX e suoi principali comandi.
- Introduzione al linguaggio di programmazione. Variabili, tipi e cast, operatori aritmetici, incremento e decremento

Pianificazione:

• Settimane 1-2

Data	Giorno	Orario	Argomento Principale
16 Feb	Lunedì	14:00 - 16:30	Intro & Ambiente: Sintassi, Moduli, Cartelle.
19 Feb	Giovedì	10:00 - 12:30	Linguaggio Python: Variabili, Classi, Liste, Array.
23 Feb	Lunedì	14:00 - 16:30	Tipizzazione: Strutture dati, Oggetti di sistema.
26 Feb	Giovedì	10:00 - 12:30	Matplotlib: Plots, Esempi e casi d'uso.

• Settimane 3-4

Data	Giorno	Orario	Argomento Principale
02 Mar	Lunedì	14:00 - 16:30	I/O Dati: Lettura valori, Print, Formatting, DateTime.
04 Mar	Giovedì	10:00 - 12:30	Esercitazione: Esempi pratici e Seminari (1.5h).
16 Mar	Lunedì	14:00 - 16:30	Ambiente: Interpretare, Pip/Conda, Spyder, Jupyter, OS.
19 Mar	Giovedì	10:00 - 12:30	Files: CSV, TSV, Formati Strutturati (HDF, FITS).

• Settimane 3-4

Data	Giorno	Orario	Argomento Principale
23 Mar	Lunedì	14:00 - 16:30	Stack Scientifico: Numpy, Pandas, SciPy (Base).
26 Mar	Giovedì	10:00 - 12:30	Chiusura: Recupero + Verifica Finale.

Esercitazioni pratiche

Casi di sviluppo di difficoltà progressiva.

Verifica frequenza/formazione

Questionario a opzioni rigide.

- **Aggiunte Extracurricolari**
- **Codici Esempio:** script pronti all'uso.
- **Piattaforma Esercitazione:** Accesso a notebook interattivi.
- **Dispense:** Materiale PDF per approfondimenti verticali.
- **Seminari:** Sessioni extra su argomenti specifici.

Programma preliminare

1. Introduzione all'ambiente Python
 - 1.1. La sintassi : motivazioni al disegno del linguaggio
 - 1.2. Corpus, moduli, cartelle
2. Linguaggio Python
 - 2.1. Variabili
 - 2.1.1. Visibilità, Interprete
 - 2.1.2. Classi, Moduli
 - 2.1.3. Liste, Array, Dizionari, Tuple.
 - 2.1.4. Argomenti, KWargs,
 - 2.2. Tipizzazione
 - 2.2.1. Strutture dati del linguaggio
 - 2.2.2. Array
 - 2.2.3. Oggetti di sistema e convenzioni delle librerie
3. Matplotlib
 - 3.1. Plots
 - 3.2. Esempi e casi d'uso
4. Lettura e Visualizzazione di valori
 - 4.1.1. Print
 - 4.1.2. Formatting
 - 4.1.3. Date Time
5. Ambiente di esecuzione
 - 5.1. Inteprete, pre-compilatore,
 - 5.2. Ambiente Variabili, convenzioni sulle chiamate
 - 5.3. Environment, Pip, Conda
 - 5.4. Spyder
 - 5.5. Colab, Jupyter
 - 5.6. Os, sys
6. Accesso a File
 - 6.1. StdIO + Files
 - 6.1.1. CSV, TSV
 - 6.2. Tipi Strutturati
 - 6.2.1. HDF, NC, FITS
7. Numpy, Pandas, SciPy – Base
 - 7.1. Librerie e tipi stretti

1. Introduzione all'ambiente Python

- **Introduzione:** Sintassi, motivazioni al disegno del linguaggio, Corpus e Moduli.
- **Obiettivi del Linguaggio:** Operatori, controllo del flusso, formattazione (if, while, for)
- **Tipizzazione:** convenzioni dei dati (interi/reali, stringhe, oggetti)

```
setOfNumbers = []

print("How many random numbers do you want to generate?")
max = int(input())

for i in range (max):
    setOfNumbers.append(random.randrange(1,101,1))
setOfNumbers.sort()
print(setOfNumbers)

print("Which number do you want to find in the set of random numbers")
searchNumber = int(input())

firstPos = 0
lastPos = max-1
found = False

while (not found and firstPos <= lastPos):
    midPos = int((firstPos + lastPos)/2)

    if (searchNumber == setOfNumbers[midPos]) :
        found = True
    else :
        if (searchNumber < setOfNumbers[midPos]):
            lastPos = midPos - 1
        else :
            firstPos = midPos + 1

if (found) :
    print("Your item is in the list")
else :
    print("Your item is not in the list")
```



2. Fondamenti del Linguaggio

- **Variabili & Strutture:**
 - Visibilità, Interprete, Classi.
 - Liste, Array, Dizionari, Tuple.
 - Gestione argomenti (Args, Kwargs).
- **Tipizzazione:** Strutture dati, Array, Oggetti di sistema e convenzioni.

```
setOfNumbers = []

print("How many random numbers do you want to generate?")
max = int(input())

for i in range(max):
    setOfNumbers.append(random.randrange(1,101,1))
setOfNumbers.sort()
print(setOfNumbers)

print("Which number do you want to find in the set of random numbers")
searchNumber = int(input())

firstPos = 0
lastPos = max-1
found = False

while (not found and firstPos <= lastPos):
    midPos = int((firstPos + lastPos)/2)

    if (searchNumber == setOfNumbers[midPos]):
        found = True
    else :
        if (searchNumber < setOfNumbers[midPos]):
            lastPos = midPos - 1
        else :
            firstPos = midPos + 1

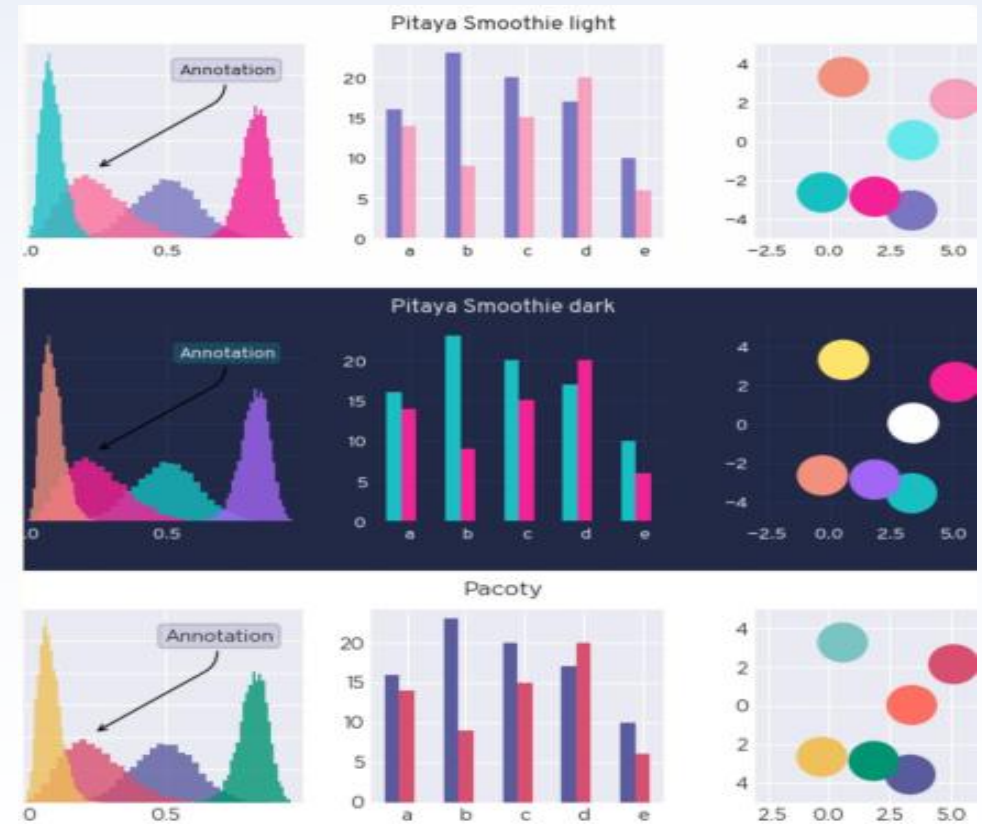
if (found) :
    print("Your item is in the list")
else :
    print("Your item is not in the list")
```





3. Matplotlib

- 3. Visualizzazione
- Matplotlib
- Creazione di Plots avanzati, analisi di esempi pratici e casi d'uso reali per la rappresentazione scientifica dei dati.





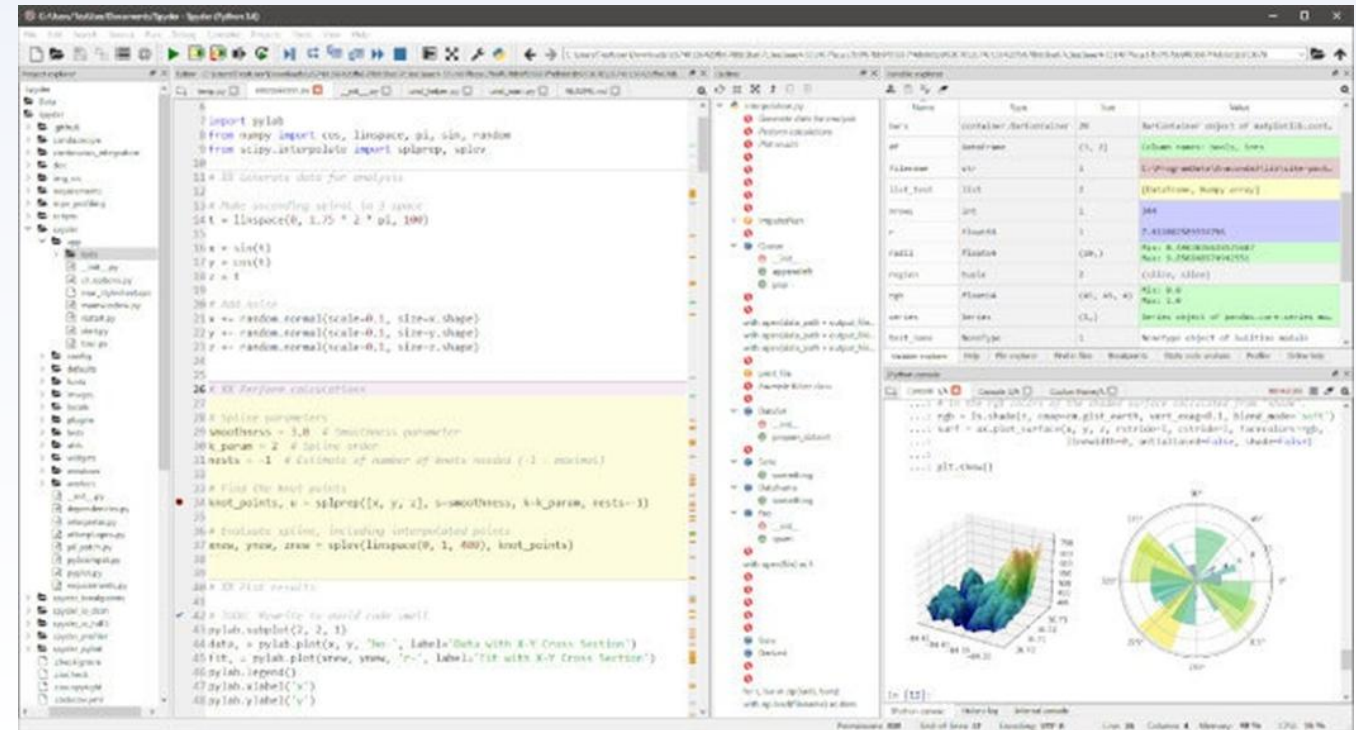
4. Lettura e Visualizzazione di valori

- **Lettura e Output**
- Funzioni di stampa e formatting.
- Gestione Date e Time.
- Visualizzazione formattata di valori.



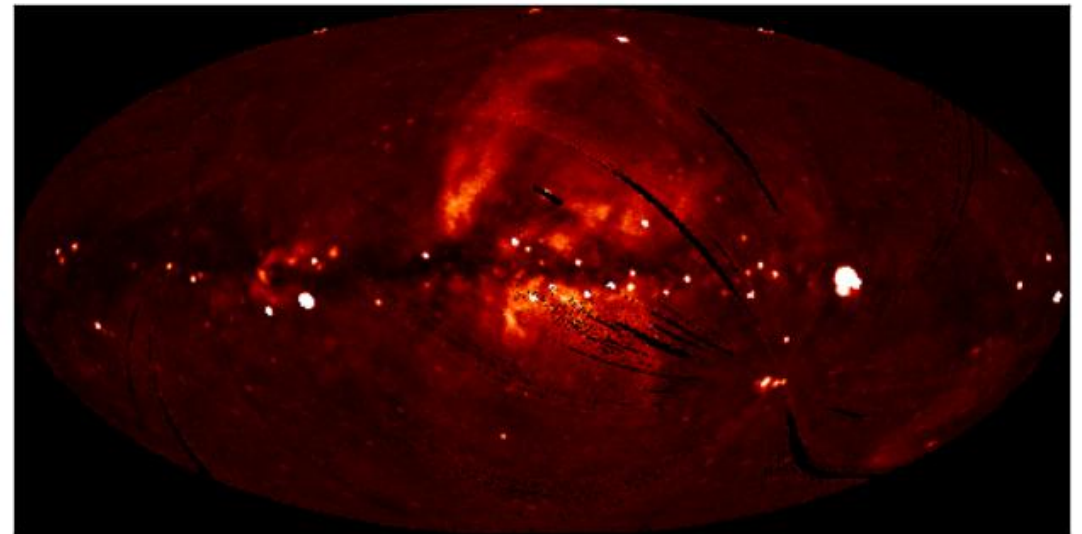
5. Ambiente di Esecuzione

- **Core**
- Interprete, Pre-compilatore
Variabili d'ambiente
OS & Sys
- **Gestione Pacchetti**
- Pip
Conda
Virtual Environments
- **IDE & Tools**
- Spyder
Jupyter Notebook
Google Colab



6. Accesso Dati e Stack Scientifico

- **Accesso a File**
- **Standard I/O:** Gestione file di testo.
- **Formati Tabellari:** CSV, TSV.
- **Tipi Strutturati:** HDF, NC, FITS (Formati scientifici per grandi moli di dati).





7. Dati e frameworks Scientifici

- **Librerie Scientifiche (Base)**
- Introduzione a **Numpy**, **Pandas**, **SciPy**: librerie e tipi stretti per il calcolo numerico.

