

# I Python









Appunti della Prof.ssa Carolina Pellecchia

3C/3J/4J

*Istituto di istruzione superiore i.i.c di vittorio - i.i.i. lattanzio*

# Lezione n° 1

## Contenuti del Corso

-  Intro
-  Introduzione al Python
-  Installazione di Python
-  Hello World In Python
-  Variabili In Python
-  Stringhe In Python

## Cos'è Python?

Python è un linguaggio di programmazione semplice, veloce e intuitivo, adatto ai principianti e non solo.

Le sue caratteristiche di integrabilità con altri linguaggi sono elevatissime e numerose sono anche le librerie con funzionalità già pronte per questo linguaggio.

E' un linguaggio ad alto livello interpretato; cioè per eseguire un programma scritto in python è necessario aver installato un interprete python.

**Interpretato** significa che per eseguire le linee di codice di un programma python, un interprete legge linea per linea e le trasforma temporaneamente in codice “macchina”, ovvero codice immediatamente comprensibile dal processore presente sulla macchina.

Osserva: potrai imparare a programmare anche guardando il codice delle librerie di python

**Iniziamo, hai già l'interprete Python installato**

## sulla tua macchina?

Il modo più semplice per verificare, è provare a eseguire il comando python dalla linea di comando.

### In Windows

1. aprire il prompt dei comandi  
→
2. digitare Python

Se è già installato, si otterrà il seguente messaggio:

```
C:\Users\carol>python
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38)
```

```
[MSC v.1929 64 bit (AMD64)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```

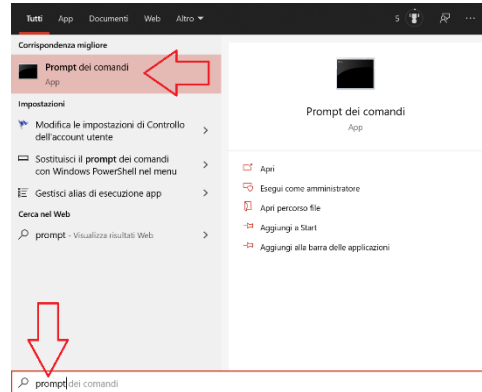
```
Prompt dei comandi - python
Microsoft Windows [Versione 10.0.19043.1237]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\carol>python
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

se non è presente, sarà restituito il messaggio seguente:

***"Python" non è riconosciuto come comando interno o esterno, un programma eseguibile o un file batch.***

Per installare Python scarica la versione più recente dal sito ufficiale: <https://www.python.org/downloads/>



Per editare il codice usiamo Visual Studio Code che puoi scaricare gratuitamente dal sito seguente:

<https://code.visualstudio.com/download>

## What is REPL in Python?

The Python standard shell, or REPL (Read-Eval-Print Loop), allows you to run Python code interactively while working on a project or learning the language. This tool is available in every Python installation, so you can use it at any moment.



e apri il tuo terminale e scrivi python, vedrai una schermata come questa:

```
flavio — python /Users/flavio — Python — 78x11
→ ~ python
Python 3.9.0 (v3.9.0:9cf6752276, Oct 5 2020, 11:29:23)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

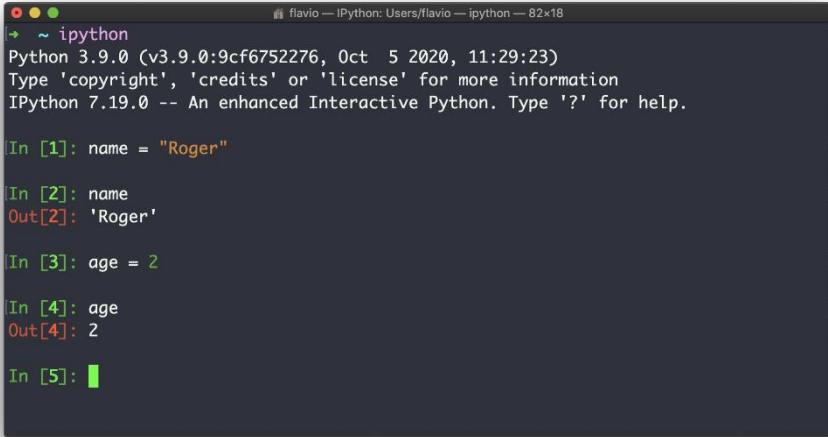
Potrebbe essere più conveniente da usare perché poi muoverti con il mouse e fare copia & incolla più facilmente che nel terminale.

Tutto ciò è integrato in Python di default. Tuttavia, consiglio di installare IPython, probabilmente la miglior applicazione riga di comando REPL che puoi trovare.

Installala con:

*pip install ipython*

Assicurati che i file binari di pip siano nel tuo percorso, poi esegui ipython:

A screenshot of a terminal window titled "flavio — IPython: Users/flavio — ipython — 82x18". The prompt is "~ ipython". The terminal shows the following output:

```
Python 3.9.0 (v3.9.0:9cf6752276, Oct 5 2020, 11:29:23)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.19.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: name = "Roger"

In [2]: name
Out[2]: 'Roger'

In [3]: age = 2

In [4]: age
Out[4]: 2

In [5]: █
```

Inizieremo, questo corso, direttamente programmando. Vedremo molti esercizi per entrare in confidenza con il linguaggio

### ❖ Hello World In Python

Iniziamo con un classico primo programma di esempio HelloWorld.py

```
print("hello word")
```

oppure

```
>>print('hello word') https://onlinegdb.com/WZndzlwna
```

proviamo a stampare un numero

```
print('hello word: ',2021)
```

## ❖ Variabili In Python

### Dynamic typing

Il tipo di una variabile in python è determinato durante l'esecuzione del programma; per questo le variabili non sono associate ad un particolare tipo e non devono essere dichiarate.

Programmiamo <https://onlinegdb.com/BojYq3vBf>

- # Questo è un commento i = 1 # Ma anche questo è un commento
- Concatenazione di stringhe nella stampa  
<https://onlinegdb.com/EKT76EuRp>

```
print(Name+' ciao ciao')
print(Name,' ciao ciao')
```

output: Carol ciao ciao  
Carol ciao ciao

- Concatenazione tra stringa e intero  
[https://onlinegdb.com/\\_qN-sDARsS](https://onlinegdb.com/_qN-sDARsS)
- Nuova linea '\n' [https://onlinegdb.com/reqP6da\\_s](https://onlinegdb.com/reqP6da_s)  

```
print("ciao,\n sono molto felice")
```

## ❖ Stringhe In Python

- Le stringhe sono array di caratteri e il primo carattere si trova alla posizione 0

<https://onlinegdb.com/z8oDWZ8fN>

```
hello.py > ...
1 nome = 'Carolina'
2 print(nome[0])
3 print(nome[3])
```

Output: C

o

- Convertire una stringa in maiuscolo o minuscolo

<https://onlinegdb.com/YQ0tuEyA9x>

```
Nome ='Carolina'
```

```
print(nome.lower())
```

```
nome.isupper() #restituisce true se tutti i caratteri della stringa nome sono in maiuscolo
```

```
print(len(nome)) #stampa il numero di caratteri della stringa nome 8
```

- Sostituzione di una stringa <https://onlinegdb.com/bS0BPZ8Ei>

```
print(nome.replace('o','O'))
```

```
print(nome.replace('na','O'))
```

- **Stampare** l'indice di un carattere all'interno della stringa







```
Print(name.index('r'))
```

- Definizione di una funzione

```
def yell(text):
```

```
    return text.upper() + '!'
```

# Lezione n° 2

-  Numeri In Python
-  Prendere l'input dall'utente
-  Esercizio di sostituzione delle parole
-  Liste In Python
-  Metodi delle liste
-  Soluzione esercizi per casa
  - ★ Stampa i caratteri di una stringa
  - ★ Somma dei soli numeri dispari
  - ★ Prodotto (nome, prezzo) esempio di Dictionary

carolina.pellecchia@gmail.com



# I Numeri In Python

- Stampare numeri e variabili contenenti numeri

```
num=34
```

```
print(67)
```

```
print(num)
```

<https://onlinegdb.com/yGWfOyYt0E>

sui numeri si possono applicare le operazioni aritmetiche di base

```
print(67+2.37) #somma
```

```
print(67-2.37) #sottrazione
```

```
print(67*2.37)#moltiplicazione
```

```
print(67/2.37)#divisione restituisce il quoziente della divisione tra 67 e 2.37
```

```
print(67%2.37)#modulo restituisce il resto della divisione tra 67 e 2.37
```

- Trasformare un numero in stringa

```
num=67
```

```
num1=str(num)# num1 contiene la stringa '67'
```

<https://onlinegdb.com/dsaXaqxc1>

- Valore assoluto del numero

```
print(abs(-34))
```

<https://onlinegdb.com/3DUKYA9wf>

- Min e Max

```
print(max(4,6,7,1))# posso mettere una lista di numeri di lunghezza a piacere
```

```
print(min(4,6,7,1))# posso mettere una lista di numeri di lunghezza a piacere
```

<https://onlinegdb.com/tTwyvvmz2>

- Arrotondamento di un numero

```
print(round(15.5)) →out 16
```

```
print(round(15.4)) →out 15
```

<https://onlinegdb.com/e2y7TsO2x>

- Convertire un numero in binario  
`print(bin(6))`
- Importare la libreria matematica con tutte le sue classi e funzioni (\*)

Questa libreria contiene tutte le operazioni per lavorare con i numeri

```
from math import *
```

```
print(sqrt(45)) #ora possiamo trovare anche la radice quadrata
```

<https://onlinegdb.com/LAZ6fuq9r>

# Prendere l'input dall'utente

- La funzione è `input()`  
    `nome = input('inserisci il tuo nome: ')`  
    `num = int(input('inserisci un numero: '))`  
    `print("il tuo nome: "+nome+" il numero: ",num)`  
<https://onlinegdb.com/5cHP74l6r>

# Esercizio di sostituzione delle parole

- Semplice programma che prende in input una frase e sostituisce una parola

```
frase = input('inserisci la frase: ')
```

```
print( ' la frase che hai inserito: '+frase)
```

```
parola1 = input('inserisci la parola da sostituire: ')
```

```
parola2= input('inserisci la parola con cui sostituire: ')
```

```
print(frase.replace(parola1,parola2))
```

<https://onlinegdb.com/0rnf7g8ak>

# Liste in python

Le liste in Python rappresentano delle strutture dati sequenziali. Queste memorizzano gli elementi in un determinato ordine, ammettono duplicati e permettono di modificare gli oggetti che vi sono contenuti.

Esempi di liste:

```
['lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom']  
[67, 'mar', 'mer', 'gio', true, 'sab', 0.9, 'b']
```

- Creare ed inserire i dati in una lista  
giorni=['lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom']
- possiamo usare anche il suo costruttore  
Giorni = list(('lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom'))  
print(giorni) # stampa la lista  
print(giorni[0]) # ogni elemento della lista ha un indice che parte da 0  
print(giorni[4]) → ven  
print(giorni[0][1]) → u
- Stampare gli elementi della lista a partire da un indice  
print(giorni[1:]) → ['mar', 'mer', 'gio', 'ven', 'sab', 'dom']  
print(giorni[2:]) → ['mer', 'gio', 'ven', 'sab', 'dom']
- Specifichiamo un intervallo  
print(giorni[1:3]) → ['mar', 'mer', 'gio']
- Stampare il tipo  
print(type(giorni))# della lista → <class 'list'>  
num = 89  
print(type(num))# del numero → <class 'int'>
- Sostituire un elemento della lista  
giorni[2] = 'ciao'  
print(giorni) →  
['lun', 'mar', 'mer', 'gio', 'ven', 'sab', 'dom']
- L'ultimo elemento della lista ha indice -1

```
print(giorni[-1])→'dom'
```

```
print(giorni[-2])→'sab'
```

```
print(giorni[-7])→'lun'
```

- Inversione di una stringa
- Numero di elementi in una lista  

```
print(len(giorni))→7
```
- La lista può contenere un misto di tipi di dati  

```
giorni=[67,'mar','mer','gio',True,'sab',0.9]
```

```
print(giorni)
```

```
print(type(giorni[0]))→ <class 'int'>
```

```
print(type(giorni[6]))→ <class 'float'>
```

```
print(type(giorni[4]))→ <class 'bool'>
```

# Metodi della lista

- Congiunzione di due liste  
giorni=['lun','mar','mer','gio','ven','sab','dom']  
numeri=list(1,2,3,4,5,6,7)  
giorni.extend(numeri)  
print(giorni)
- Aggiungere un elemento alla lista  
giorni.append('ciao')  
print(giorni)  
print(len(giorni))
- Inserire un elemento nella lista  
giorni.insert(2,'ciao')  
print(giorni)→  
['lun','mar','ciao','mer','gio','ven','sab','dom']
- Eliminiamo un particolare elemento dalla lista  
giorni.remove('ciao')  
print(giorni)→  
['lun','mar','mer','gio','ven','sab','dom']
- Eliminare tutti gli elementi dalla lista  
giorni.clear()  
print(giorni)→[] empty lista
- Eliminare completamente la lista  
del giorni  
print(giorni)→lista eliminata
- Contare quante volte appare un elemento nella lista  
numeri=[1,4,1,45,7,1,2]  
print(numeri.count(1))→3  
giorni = list('lun','mar','dom','gio','ven','sab','dom')  
print(giorni.count('dom'))→2
- Ordinare una lista  
numeri=[1,4,1,45,7,1,2]

```
numeri.sort()
```

```
print(numeri)
```

- Invertire una lista

```
numeri=[1,4,1,45,7,1,2]
```

```
numeri.reverse()
```

```
print(numeri)
```

- Copiare una lista

```
numeri=[1,4,1,45,7,1,2]
```

```
n=numeri.copy()
```

- Eliminare l'ultimo elemento di una lista

```
numeri.pop()
```

```
print(numeri)
```

- Eliminare un elemento mediante l'indice

- ```
numeri.remove(2)
```

```
print(numeri)
```

- ```
del numeri[2]
```

```
print(numeri)
```



# Soluzione esercizi per casa

★ Stampa i caratteri di una stringa

<https://onlinegdb.com/aASrM9UAO>

★ Somma dei soli numeri dispari




<https://onlinegdb.com/ZMMftGIRxU>

★ Prodotto (nome, prezzo) esempio di Dictionary

[https://onlinegdb.com/H8t\\_DrPML](https://onlinegdb.com/H8t_DrPML)

carolina.pellecchia@gmail.com

# Lezione n° 3

-  (12:10:00) Soluzione esercizi per casa
    - ★ Popolare una lista con n numeri pari
    - ★ Popolare una lista con i multipli di 5
  -  (12:30:00) Generare un numero random
  -  (12:10:00) Assegno LaBor@torio
    1. Indovina il numero
    2. Esercizio Professor (Random)
    3. Esercizio Estrazione numero Casuale
  - ★ Assegno per casa
- Proseguire gli esercizi di laboratorio

# Soluzione esercizi per casa

★ Popolare una lista con n numeri pari

```
...  
Popolare una lista di n elementi con i primi n numeri pari.  
Dopo averli inseriti  
visualizzare in output i valori memorizzati nella lista e la  
loro posizione.  
...  
  
def InserisciNumeri(mylist,n):  
    for i in range (0,2*n,2):  
        mylist.append(i)  
  
def VisualizzaElemPos(mylist):  
    for l in mylist:  
        print("valore: ",l," posizione: ",mylist.index(l))  
  
# programma principale  
  
mylist=list()  
n=int(input("inserisci in numero di elementi che vuoi  
inserire: "))  
print("Popolare una lista di n elementi con i primi n numeri  
pari")  
InserisciNumeri(mylist,n)  
  
print(mylist)  
print("visualizzare in output i valori memorizzati nella  
lista e la loro posizione")  
VisualizzaElemPos(mylist)
```

## ★ Popolare una lista con i multipli di 5

...

Popolare una lista di n elementi con i primi n multipli di 5 escludendo lo zero.

Dopo aver terminato l'inserimento visualizzare in output i valori della lista e il relativo indice.

...

```
def InserisciMultipli5(mylist,n):
    for i in range (1,n+1):
        mylist.append(i*5)
        print(i*5)

def VisualizzaElemPos(mylist):
    for l in mylist:
        print("valore: ",l," posizione: ",mylist.index(l))

# programma principale
mylist=list()
n=int(input("inserisci in numero di elementi che vuoi
inserire: "))
print("Popolare una lista di n elementi con i primi n
multipli di 5 escludendo lo zero")
InserisciMultipli5(mylist,n)

print("visualizzare in output i valori memorizzati nella
lista e la loro posizione")
VisualizzaElemPos(mylist)
```

# Generare un numero random

Python non ha una funzione random. Per la generazione dei numeri casuali devi importare il modulo random.

```
import random
```

Nel modulo random ci sono diverse funzioni per generare uno o più numeri casuali.

Le funzioni più usate e comuni sono le seguenti:

- **random.random()** genera un numero casuale reale da 0 a 1.
- **random.randint(x,y)** genera un numero casuale intero da x a y.

Generazione dei numeri casuali reali

Esempio 1

Questo script genera un numero casuale da 0 a 1

1. import random
2. x=random.random()
3. print("numero casuale reale:",x)

```
Output>> numero casuale reale: 0,45454545999
```

Generazione dei numeri casuali interi

Esempio 2

Questo script genera un numero random intero da 1 a 10

1. import random
2. x=random.randint(1,10)
3. print("numero casuale intero",x)

```
Output>> numero casuale intero: 10
```

# QUIZZ

ABC in python  
14 Domande

NOME : \_\_\_\_\_

CLASSE : \_\_\_\_\_

DATA : \_\_\_\_\_

1. Cos'è Python?(dare 2 risposte)

- |  |   |
|--|---|
| <input type="checkbox"/> A un linguaggio di programmazione | <input type="checkbox"/> B un interprete                                      |
| <input type="checkbox"/> C un serpente per computer        | <input type="checkbox"/> D un linguaggio per chiedere informazioni all'utente |

2. a cosa serve Visual Studio Code?

- |  |  |
|--|--|
| <input type="checkbox"/> A a scrivere poesie | <input type="checkbox"/> B ad eseguire programmi |
| <input type="checkbox"/> C ad editare codice | <input type="checkbox"/> D a scrivere            |

3. Come si chiede all'utente di inserire un numero?

- |   |
|---|
| <input type="checkbox"/> A num=input('inserisci un numero: ') <input type="checkbox"/> B num:print('inserisci un numero: ')             |
| <input type="checkbox"/> C num=int(input('inserisci un numero: ')) <input type="checkbox"/> D num=int(println('inserisci un numero: ')) |

4. Come si stampa un numero intero dopo una stringa?

- |  |  |
|--|--|
| <input type="checkbox"/> A print('il numero è'+89) | <input type="checkbox"/> B println('il numero è'+89) |
| <input type="checkbox"/> C print('il numero è':89) | <input type="checkbox"/> D print('il numero è',89)   |

5. Come si concatenano le stringe?

- |  |   |
|--|---|
| <input type="checkbox"/> A str1 ='Pino Rossi'+67 | <input type="checkbox"/> B str1 = str1+"Pino Rossi" |
| <input type="checkbox"/> C str1 ='Pino Rossi'    | <input type="checkbox"/> D str1 = str1:"Pino Rossi" |

6. Le stringhe sono array di caratteri. consideriamo la stringa  
--str = "LA CLASSE 1C"-- cosa stampa l'istruzione print(str[3])?

- |                                   |                              |
|-----------------------------------|------------------------------|
| <input type="checkbox"/> A C      | <input type="checkbox"/> B A |
| <input type="checkbox"/> C spazio | <input type="checkbox"/> D L |

7. Indica la posizione del primo e dell'ultimo carattere in una stringa.

- |                            |      |                            |      |
|----------------------------|------|----------------------------|------|
| <input type="checkbox"/> A | 0,-2 | <input type="checkbox"/> B | 0,-1 |
| <input type="checkbox"/> C | 1,n  | <input type="checkbox"/> D | 0,n  |

8. Come si converte una stringa in minuscolo?

- |                            |                                  |                            |                                    |
|----------------------------|----------------------------------|----------------------------|------------------------------------|
| <input type="checkbox"/> A | <code>print(nome.lower())</code> | <input type="checkbox"/> B | <code>print(nome.low())</code>     |
| <input type="checkbox"/> C | <code>print(nome.lower())</code> | <input type="checkbox"/> D | <code>println(nome.lower())</code> |

9. Come si sostituisce la parola pasta della frase `"oggi si mangia la pasta"` con la parola `"pizza"`?

- |                            |  |                            |   |
|----------------------------|--|----------------------------|---|
| <input type="checkbox"/> A | <code>print(frase.replace('pizza','pasta'))</code> | <input type="checkbox"/> B | <code>print(nome.replace('pizza','pasta'))</code> |
| <input type="checkbox"/> C | <code>print(frase.replace('pasta','pizza'))</code> | <input type="checkbox"/> D | <code>print(nome.replace('o','0'))</code>         |

10. Come si chiede ad un utente di inserire una frase?

- |                            |   |                            |   |
|----------------------------|---|----------------------------|---|
| <input type="checkbox"/> A | <code>frase = print('inserisci una frase: ')</code> | <input type="checkbox"/> B | <code>frase = println('inserisci una frase: ')</code> |
| <input type="checkbox"/> C | <code>frase = input('inserisci una frase: ')</code> | <input type="checkbox"/> D | <code>frase = inputln('inserisci una frase: ')</code> |

11. Come si stampa il primo elemento della lista seguente?  
`frutta = ['mela','pera','uva']`

- |                            |                               |                            |                               |
|----------------------------|-------------------------------|----------------------------|-------------------------------|
| <input type="checkbox"/> A | <code>print(frutta)</code>    | <input type="checkbox"/> B | <code>print(frutta[0])</code> |
| <input type="checkbox"/> C | <code>print(frutta[1])</code> | <input type="checkbox"/> D | <code>print(frutta)</code>    |

12. Come si stampa la lettera 'p' di pera?  
`frutta = ['mela','pera','uva']`

- |                            |                                  |                            |                                  |
|----------------------------|----------------------------------|----------------------------|----------------------------------|
| <input type="checkbox"/> A | <code>print(frutta[1][0])</code> | <input type="checkbox"/> B | <code>print(frutta[1])</code>    |
| <input type="checkbox"/> C | <code>print(frutta[2][1])</code> | <input type="checkbox"/> D | <code>print(frutta[0][3])</code> |

13. Come si chiede ad un utente di inserire un numero?

- |                            |  |                            |   |
|----------------------------|--|----------------------------|---|
| <input type="checkbox"/> A | <code>num = int(print('inserisci numero:'))</code> | <input type="checkbox"/> B | <code>num = input('inserisci numero:')</code> |
| <input type="checkbox"/> C | <code>num = int(input('inserisci numero:'))</code> | <input type="checkbox"/> D | <code>num = print('inserisci numero:')</code> |





14. Dynamic typing cosa significa?

- A Il tipo di una variabile in python non esiste
- C Il tipo di una variabile in python non è determinata

- B Il tipo di una variabile in python è determinato durante l'esecuzione del programma;
- D Il tipo di una variabile in python è determinato durante la scrittura del programma;



# Lezione n° 4

-  Tuples In Python
-  Functions In Python
-  The Return Keyword
-  IF Statements In Python *Assegno per casa*

[carolina.pellecchia@gmail.com](mailto:carolina.pellecchia@gmail.com)

**Esercizio 1**

```
#Calcolare quadrato e cubo di un numero intero fornito in input
print("Quadrato e cubo di un numero")
numero = int(input("Numero : "))
quadrato=numero*numero;
cubo=quadrato*numero;
print("Quadrato di "+str(numero)+" = " + str(quadrato))
print("Cubo di "+str(numero)+" = " + str(cubo))
```

```
Quadrato e cubo di un numero
Numero : 45
Quadrato di 45 = 2025
Cubo di 45 = 91125
```

**Esercizio 2**

```
# Calcolare, usando una funzione, la somma dei primi n numeri interi
# con n fornito in input
def sommanpriminumeri(n):
    somma = 0
    for i in range(1,n+1):
        somma += i
    print('La somma dei primi ' + str(n) + ' numeri vale ' + str(somma))
num=int(input("Valore di n: "))
sommanpriminumeri(num)
```

```
Valore di n: 100
La somma dei primi 100 numeri vale 5050
```

**Esercizio 3**

```
# Calcolare l'ipotenusa di un triangolo rettangolo con i due cateti forniti in
# input. Ripetere il calcolo quante volte si vuole.
import math
print("Teorema di Pitagora")
continui = "S"
while (continui != "N" and continui != "n"):
    cateto1 = float(input("Cateto 1 : "))
    cateto2 = float(input("Cateto 2 : "))
    ipotenusa = math.sqrt(math.pow(2, cateto1) + math.pow(cateto2, 2))
    ipotenusa = round(ipotenusa, 2)
    # oppure: ipotenusa=((int)(ipotenusa*100))/100
    print("Ipotenusa del rettangolo = " + str(ipotenusa))
    continui = str(input("Altri calcoli ? "))
```

```
Teorema di Pitagora
Cateto 1 : 4
Cateto 2 : 3
Ipotenusa del rettangolo = 5.0
Altri calcoli ?
```

**Esercizio 4**

```
# Inserire un numero intero e positivo e verificare se è un numero primo
# Controllare che in input si accetti esclusivamente un numero intero e positivo
```

```
def seprimo(n):
    for i in range(2, int(n / 2) + 1):
        if n % i == 0:
            print("Numero non primo")
            return
    print("Numero primo")

while True: # per ripetere il calcolo
    while True: # per validare l'input
        stringa = input("Numero (X per terminare) ")
        if stringa == "X" or stringa == "x":
            print("Ciao")
            exit()
        if stringa.isdigit() and stringa[0]!="-":
            numero=int(stringa)
            break
        else:
            print("input non valido")
    seprimo(numero)
```

```
Numero (X per terminare) 43.76
input non valido
Numero (X per terminare) 31
Numero primo
Numero (X per terminare) 12
Numero non primo
Numero (X per terminare) x
Ciao
```

**Esercizio 5**

```
# Visualizzare data e ora corrente
```

```
import datetime
adesso = datetime.datetime.now()
print("Data e ora corrente")
print(adesso.strftime("%d/%m/%Y %H:%M:%S"))
print("Anno corrente: ", end='') # end='' per evitare una nuova linea...
print(adesso.year)
```

```
Data e ora corrente
27/04/2020 17:27:37
Anno corrente: 2020
```

**Esercizio 6**

```
# Esercizio sulle stringhe: contare il numero di caratteri, lettere e
# altri simboli contenuti in due stringhe fornite in input.
def analizzastringa(s):
    ncar = len(s)
    nlet = 0
    nnum = 0
    for c in s:
        if c.islower() or c.isupper():
            nlet += 1
        if c.isdigit():
            nnum += 1
    nsim= ncar - nlet - nnum
    return ("Totale Caratteri: "+str(ncar)+" Lettere: "+str(nlet)+ " Numeri
: "+str(nnum)+" Altri Simboli: "+str(nsim))

print("Contare il numero totale di caratteri, il numero di lettere, di numeri e
di altri simboli")
stringa1 = input("Stringa1 : ")
stringa2 = input("Stringa2 : ")
print("Stringa1: "+analizzastringa(stringa1))
print("Stringa2: "+analizzastringa(stringa2))
```

```
Contare il numero totale di caratteri, il numero di lettere, di numeri e di altri simboli
Stringa1 : Teramo 27/04/2020!
Stringa2 : Roma 753AC
Stringa1: Totale Caratteri: 18 Lettere: 6 Numeri: 8 Altri Simboli: 4
Stringa2: Totale Caratteri: 10 Lettere: 6 Numeri: 3 Altri Simboli: 1
```

**Esercizio 7**

```
# Utilizzo di liste e della funzione sort() di Python
import random
# Genero una lista di n numeri interi random compresi tra n1 ed n2
# e una seconda lista di città
n=20
n1=0
n2=100
numerirandom=[]
for i in range(n):
    numerirandom.append(random.randint(n1, n2))
print(str(n)+" numeri random compresi tra "+str(n1)+" e "+str(n2)+":
"+str(numerirandom))
numerirandom.sort()
print("Numeri random ordinati: "+str(numerirandom))
print()
città=["Milano","Bologna","Palermo","Roma","Torino","Teramo","Napoli","Bari","Pescara"]
print("Città: "+str(città))
città.sort()
print("Città in ordine alfabetico: "+str(città))
città.sort(reverse=True)
print("Città in ordine alfabetico decrescente: "+str(città))
```

```
20 numeri random compresi tra 0 e 100: [12, 14, 67, 95, 81, 17, 59, 91, 1, 85, 22, 20, 53, 75, 76, 62, 48, 48, 30, 58]
Numeri random ordinati: [1, 12, 14, 17, 20, 22, 30, 48, 48, 53, 58, 59, 62, 67, 75, 76, 81, 85, 91, 95]
```

```
Città: ['Milano', 'Bologna', 'Palermo', 'Roma', 'Torino', 'Teramo', 'Napoli', 'Bari', 'Pescara']
Città in ordine alfabetico: ['Bari', 'Bologna', 'Milano', 'Napoli', 'Palermo', 'Pescara', 'Roma', 'Teramo', 'Torino']
Città in ordine alfabetico decrescente: ['Torino', 'Teramo', 'Roma', 'Pescara', 'Palermo', 'Napoli', 'Milano', 'Bologna', 'Bari']
```

**Esercizio 8**

```
# Generare 10000 numeri interi random compresi tra 0 e 1000000
# stampare il numero generato più grande,
# il numero generato più piccolo e la media dei numeri generati
import random
# Genero una lista di n numeri interi random compresi tra n1 e n2
n = 10000
n1 = 0
n2 = 1000000
nmax=n1
nmin=n2
somma=0
for i in range(n):
    numerogenerato=random.randint(n1, n2)
    if numerogenerato>nmax:
        nmax=numerogenerato
    if numerogenerato<nmin:
        nmin=numerogenerato
    somma+=numerogenerato
print("Numero generato più grande = "+str(nmax))
print("Numero generato più piccolo = "+str(nmin))
print("Media dei numeri generati = "+str(round(somma/n,2)))
```

```
Numero generato più grande = 999712
Numero generato più piccolo = 81
Media dei numeri generati = 500708.59
```

**Esercizio 9**

```
# Verificare se due stringhe fornite in input sono una l'anagramma dell'altra
def seanagramma(s1, s2):
    if len(s1) != len(s2): # se la lunghezza non è la stessa ...
        return False
    s1sorted = sorted(s1) # Ordiniamo le stringhe e le confrontiamo
    s2sorted = sorted(s2)
    if s1sorted == s2sorted:
        return True
    else:
        return False

stringa1 = input("Stringa1: ")
stringa2 = input("Stringa2: ")
if seanagramma(stringa1, stringa2):
    print("Le due stringhe sono una l'anagramma dell'altra")
else:
    print("Le due stringhe non sono una l'anagramma dell'altra")
```

```
Stringa1: incerta
Stringa2: trincea
Le due stringhe sono una l'anagramma dell'altra
Stringa1: vita incerta
Stringa2: trincea vate
Le due stringhe non sono una l'anagramma dell'altra
```

**Esercizio 10**

```
# Calcolare il numero di giorni trascorsi da una data fornita in input
# Controllare che la data inserita sia valida e non superiore alla data corrente
```

```
import datetime
#
adesso = datetime.datetime.now()
annocorrente = adesso.year
mesecorrente = adesso.month
giornocorrente = adesso.day
datacorrente = datetime.date(annocorrente, mesecorrente, giornocorrente)
s_datacorrente=str(giornocorrente)+"-"+str(mesecorrente)+"-"+str(annocorrente)
```

```
def sedatavalida(g,m,a):
    if a > annocorrente:
        return False
    if (m == 4 or m == 6 or m == 9 or m == 11) and g > 30:
        print("qui")
        return False
    if m == 2 and a % 4 == 0 and g > 29:
        return False
    if m == 2 and a % 4 > 0 and g > 28:
        return False
    if datetime.date(a, m, g) > datacorrente:
        return False
    return True
```

```
def giornitrascorsi(g,m,a):
    datainiziale=datetime.date(a, m, g)
    return (datacorrente-datainiziale).days
```

```
# inizio programma-----
print("Giorni trascorsi. Inserire la data" )
while True: # per validare l'input del giorno
    s_giorno = input("Giorno [1-31]: ")
    if s_giorno.isdigit() and int(s_giorno)>0 and int(s_giorno)<32:
        giorno= int(s_giorno)
        break
    else:
        print("giorno non valido")
while True: # per validare l'input del mese
    s_mese = input("Mese [1-12]: ")
    if s_mese.isdigit() and int(s_mese) > 0 and int(s_mese) < 13:
        mese = int(s_mese)
        break
    else:
        print("mese non valido")
while True: # per validare l'input dell'anno
    s_anno = input("Anno [1000-2050]: ")
    if s_anno.isdigit() and int(s_anno) > 999 and int(s_anno) < 2050:
        anno = int(s_anno)
        break
    else:
        print("anno non valido")
# controllo che la data sia corretta
datainserita=s_giorno+"-"+s_mese+"-"+s_anno
```

```
# se la data è corretta stampo i giorni trascorsi
if sedatavalida(giorno,mese,anno):
    print("Dalla data "+datainserita+" alla data di oggi "+ s_datacorrente+" sono trascorsi "+str(giornitrascorsi(giorno,mese,anno))+" giorni!")
else:
    print(" la data inserita: "+datainserita+" non è valida")
```

Giorni trascorsi. Inserire la data

Giorno [1-31]: 4

Mese [1-12]: 4

Anno [1000-2050]: 2020

Dalla data 4-4-2020 alla data di oggi 28-4-2020 sono trascorsi 24 giorni!