

Lezione S2/L4

Linguaggio python(2)

La Lezione di oggi ha richiesto lo sviluppo di un programma utilizzando il linguaggio di programmazione python. in questo programma dovevamo:

- Scrivere un programma in Python che in base alla scelta dell'utente permetta di calcolare il perimetro di diverse figure geometriche. Per la risoluzione dell'esercizio abbiamo scelto:
 - Quadrato
 - Rettangolo
 - Triangolo
 - Cerchio

```
import math

def calcola_perimetro():
    print("Scegli la figura geometrica per calcolare il perimetro:")
    print("1. Quadrato")
    print("2. Rettangolo")
    print("3. Triangolo")
    print("4. Cerchio")
    scelta = input("Inserisci il numero corrispondente alla figura: ")

    if scelta == "1":
        lato = float(input("Inserisci la lunghezza del lato del quadrato: "))
        perimetro = 4 * lato
        print(f"il perimetro del quadrato è: {perimetro}")

    elif scelta == "2":
        base = float(input("Inserisci la lunghezza della base del rettangolo: "))
        altezza = float(input("Inserisci l'altezza del rettangolo: "))
        perimetro = 2 * (base + altezza)
        print(f"il perimetro del rettangolo è: {perimetro}")

    elif scelta == "3":
        lato1 = float(input("Inserisci la lunghezza del primo lato del triangolo: "))
        lato2 = float(input("Inserisci la lunghezza del secondo lato del triangolo: "))
        lato3 = float(input("Inserisci la lunghezza del terzo lato del triangolo: "))
        perimetro = lato1 + lato2 + lato3
        print(f"il perimetro del triangolo è: {perimetro}")

    elif scelta == "4":
        raggio = float(input("Inserisci la lunghezza del raggio del cerchio: "))
        perimetro = 2 * math.pi * raggio
        print(f"La circonferenza del cerchio è: {perimetro}")

    else:
        print("Scelta non valida. Riprova!")

# Chiamata alla funzione
calcola_perimetro()
```

figura 1: Il codice

Analizziamo ora le varie parti del codice e capiamo come siamo arrivati a questo risultato. Personalmente ho svolto il programma su terminale mac, i comandi sono praticamente uguali e il procedimento è esattamente uguale.

Innanzitutto dobbiamo creare il file "perimetro.py" per fare questo apriamo il terminale, viaggiamo nella directory dove vogliamo lavorare ed utilizziamo il comando nano come nell'esercizio precedente.

il codice si divide in 3 parti (figura 2):

INPUT : il codice inizia con la definizione della funzione "calcola_perimetro", questa chiede all'utente di scegliere un'opzione tra 1,2,3,4 per scegliere la tipologia di figura di cui vuole calcolare il perimetro stampando anche una legenda rispetto ai valori e alla scelta.

SCELTA MULTIPLA: qui facciamo un check con una condizione **if-else** della scelta fatta dall'utente, in base al valore selezionato chiediamo poi in input la lunghezza dei lati (solo i dati necessari), per il cerchio e il quadrato basta 1 dato, per il rettangolo ne servono due e invece per il triangolo occorrono tutti e 3.

Notare come usiamo sempre la variabile `perimetro` per ogni `if`, questo perché ad ogni iterazione del codice viene sovrascritta da 1 sola delle condizioni `if` le altre vengono saltate e quindi non si hanno problemi di dichiarazione multipla.

Ho anche aggiunto un check alla fine verificando che la scelta dell'utente sia valida.

CHIAMATA: alla fine tutto ciò che dobbiamo fare è semplicemente scrivere una chiamata alla funzione che abbiamo appena definito senza necessità di argomenti in entrata.

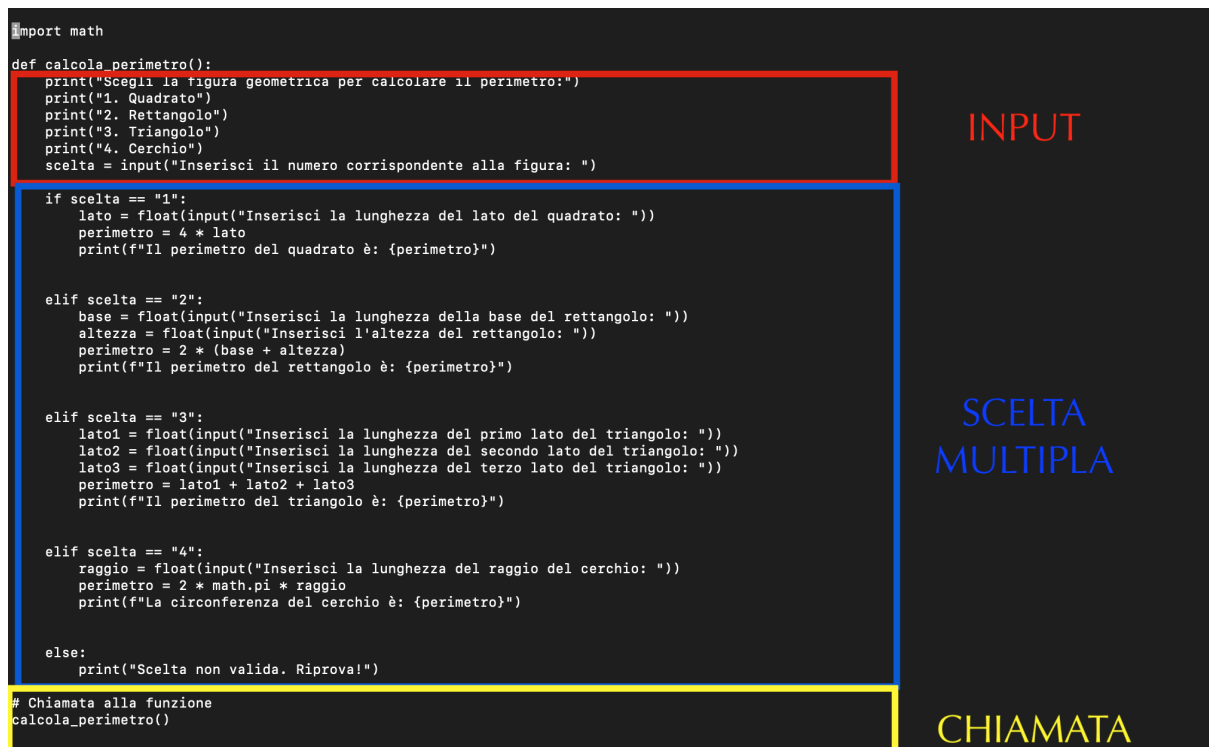


figura 2: breakdown del codice

Dopo aver salvato il programma con **“ctrl +O”** possiamo uscire. Per poter svolgere il codice dobbiamo ora prima cambiare i permessi tramite **chmod** per renderlo eseguibile e poi senza necessità di creare un eseguibile digitiamo, **python3 ./perimetro.py**.

python3 in questo caso sostituisce lo **shebang** che aveva implementato nell'esercizio precedente, entrambe le tecniche hanno risultato simile.

```
((base) matteobosio@MacBookdiMatteo Desktop % nano perimetro.py
(base) matteobosio@MacBookdiMatteo Desktop % python3 ./perimetro.py
Scegli la figura geometrica per calcolare il perimetro:
1. Quadrato
2. Rettangolo
3. Triangolo
4. Cerchio
Inserisci il numero corrispondente alla figura: 1
Inserisci la lunghezza del lato del quadrato: 4
Il perimetro del quadrato è: 16.0
(base) matteobosio@MacBookdiMatteo Desktop % python3 ./perimetro.py
Scegli la figura geometrica per calcolare il perimetro:
1. Quadrato
2. Rettangolo
3. Triangolo
4. Cerchio
Inserisci il numero corrispondente alla figura: 2
Inserisci la lunghezza della base del rettangolo: 10
Inserisci l'altezza del rettangolo: 5
Il perimetro del rettangolo è: 30.0
(base) matteobosio@MacBookdiMatteo Desktop % python3 ./perimetro.py
Scegli la figura geometrica per calcolare il perimetro:
1. Quadrato
2. Rettangolo
3. Triangolo
4. Cerchio
Inserisci il numero corrispondente alla figura: 4
Inserisci la lunghezza del raggio del cerchio: 2
La circonferenza del cerchio è: 12.566370614359172
(base) matteobosio@MacBookdiMatteo Desktop % python3 ./perimetro.py
Scegli la figura geometrica per calcolare il perimetro:
1. Quadrato
2. Rettangolo
3. Triangolo
4. Cerchio
Inserisci il numero corrispondente alla figura: 3
Inserisci la lunghezza del primo lato del triangolo: 6
Inserisci la lunghezza del secondo lato del triangolo: 5
Inserisci la lunghezza del terzo lato del triangolo: 7
Il perimetro del triangolo è: 18.0
```

figura 4: Esecuzione del programma

come vediamo il programma viene eseguito correttamente, possibili migliorie sarebbero:

- Se l'**utente sbaglia** mandare il programma in loop e richiedere ancora l'input
- Aggiungere un **check** al perimetro del triangolo verificando che le misure possano effettivamente comporre un triangolo
- Aggiungere altre **figure** implementando ulteriori condizioni if.
- Aggiungere un **calcolatore dell'area**.