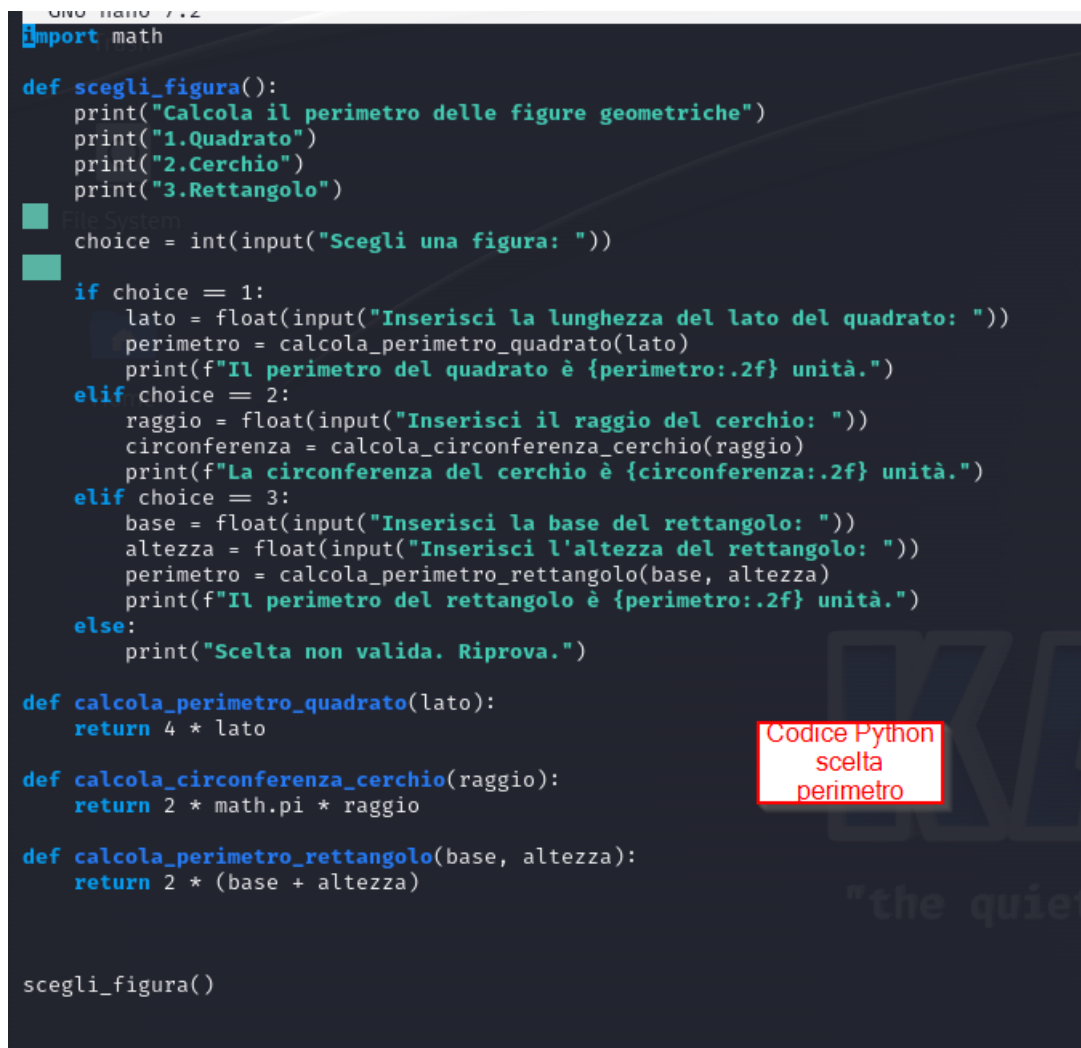


ESERCIZIO W6D4

Andiamo a svolgere un esercizio su con python sempre usando nano per scrivere il codice. La traccia è la seguente:

Si scriva un programma in Python che in base alla scelta dell'utente permetta di calcolare il perimetro di diverse figure geometriche (scegliete pure quelle che volete voi). Per la risoluzione dell'esercizio abbiamo scelto:

- Quadrato ($\text{perimetro} = \text{lato} * 4$)
- Cerchio ($\text{circonferenza} = 2 * \pi \text{ greco} * r$)
- Rettangolo ($\text{perimetro} = \text{base} * 2 + \text{altezza} * 2$)



```
import math

def scegli_figura():
    print("Calcola il perimetro delle figure geometriche")
    print("1.Quadrato")
    print("2.Cerchio")
    print("3.Rettangolo")
    choice = int(input("Scegli una figura: "))

    if choice == 1:
        lato = float(input("Inserisci la lunghezza del lato del quadrato: "))
        perimetro = calcola_perimetro_quadrato(lato)
        print(f"Il perimetro del quadrato è {perimetro:.2f} unità.")
    elif choice == 2:
        raggio = float(input("Inserisci il raggio del cerchio: "))
        circonferenza = calcola_circonferenza_cerchio(raggio)
        print(f"La circonferenza del cerchio è {circonferenza:.2f} unità.")
    elif choice == 3:
        base = float(input("Inserisci la base del rettangolo: "))
        altezza = float(input("Inserisci l'altezza del rettangolo: "))
        perimetro = calcola_perimetro Rettangolo(base, altezza)
        print(f"Il perimetro del rettangolo è {perimetro:.2f} unità.")
    else:
        print("Scelta non valida. Riprova.")

def calcola_perimetro_quadrato(lato):
    return 4 * lato

def calcola_circonferenza_cerchio(raggio):
    return 2 * math.pi * raggio

def calcola_perimetro Rettangolo(base, altezza):
    return 2 * (base + altezza)

scegli_figura()
```

Codice Python
scelta
perimetro

```
(kali㉿kali)-[~]  
$ cd Python  
  
(kali㉿kali)-[~/Python]  
$ nano scegli_perimetro.py  
  
(kali㉿kali)-[~/Python]  
$ nano scegli_perimetro.py  
  
(kali㉿kali)-[~/Python]  
$ python -m py_compile scegli_perimetro.py
```

Compilatore
ed
esecuzione