

Traccia: Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica. Dato il codice si richiede allo studente di:

- Capire cosa fa il programma senza eseguirlo.
- Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
- Individuare eventuali errori di sintassi / logici.
- Proporre una soluzione per ognuno di essi.

```
import datetime
```

```
def assistente_virtuale(comando):  
    if comando == "Qual è la data di oggi?":  
        oggi = datetime.date.today()  
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")  
    elif comando == "Che ore sono?":  
        ora_attuale = datetime.datetime.now().time()  
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")  
    elif comando == "Come ti chiami?":  
        risposta = "Mi chiamo Assistente Virtuale"  
    else:  
        risposta = "Non ho capito la tua domanda."  
    return risposta  
while True:  
    comando_utente = input("Cosa vuoi sapere? ")  
    if comando_utente.lower() == "esci":  
        print("Arrivederci!")  
        break  
    else:  
        print(assistente_virtuale(comando_utente))
```

1. Capire cosa fa il programma senza eseguirlo.

Il programma è un assistente virtuale , a mio parere molto limitata perchè non ha una risposta se l'utente scrive un'altro input diverso da quello indicato , rendendo il programma molto limitato e poco **user-friendly**.

risponde a tre domande:

- "Qual è la data di oggi?" — Restituisce la data corrente.
- "Che ore sono?" — Restituisce l'ora corrente.
- "Come ti chiami?" — Restituisce il nome dell'assistente virtuale.

2. Individuare casistiche non standard che il programma non gestisce

Mancanza di gestione di input vuoti o non validi: Il programma non verifica se l'input dell'utente è vuoto o contiene caratteri non validi.

Mancanza di gestione di errori durante la formattazione delle date e degli orari: Se si verificano errori durante la formattazione, il programma non li gestisce in modo corretto.

Ciclo while infinito: Il ciclo **while True** non ha una condizione di uscita chiara, portando a un loop infinito.

3. Individuare errori di sintassi

1. Indentazione errata:

- **Problema:** L'istruzione **return risposta** è indentata all'interno del blocco **if-elif-else**, facendo sì che la funzione termini prima dopo la prima condizione verificata.
- **Soluzione:** Dobbiamo spostare **return risposta** fuori dal blocco **if-elif-else** per assicurarti che la funzione restituisca sempre una risposta.

2. Ciclo **while** posizionato erroneamente:

- **Problema:** Il ciclo **while** è posizionato all'interno della funzione, il che significa che la funzione si eseguirebbe solo una volta.
- **Problema:** La riga **while True** deve terminare con i due punti per indicare l'inizio del blocco di codice.

- **Soluzione:** Una soluzione che possiamo adoperare è quella di spostare il ciclo `while` fuori dalla funzione, in modo che continui a chiedere all'utente l'input fino a quando non inserisce "esci".

3. La funzione `datetime.datetime.today()`:

- **Problema:** La funzione corretta per ottenere la data odierna è `datetime.date.today()`.

4. Gestione degli input vuoti:

- **Problema:** Se l'utente preme "invio" senza digitare nulla, il programma restituirà "Non ho capito la tua domanda." senza un messaggio utile.

5. Fuso Orario errato :

- **Problema:** se interroghiamo per sapere che ore sono il programma ci stampa un'orario con fuso orario diverso da quello dove mi trovo io geograficamente.
- **Soluzione:** Aggiungere il modulo `pytz` risolverebbe questo problema , modulo che permette di lavorare con i fuso orari su Python.(impostiamo sul codice direttamente il fuso orario italiano)

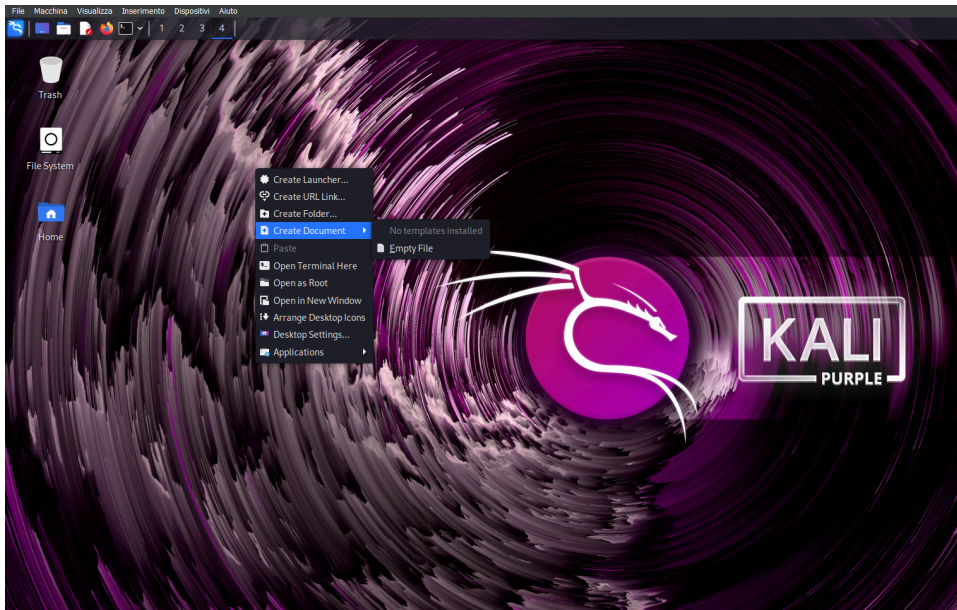
4)Proporre una soluzione per ognuno di essi

```
import datetime
import pytz # Importa il modulo pytz per gestire i fusi orari

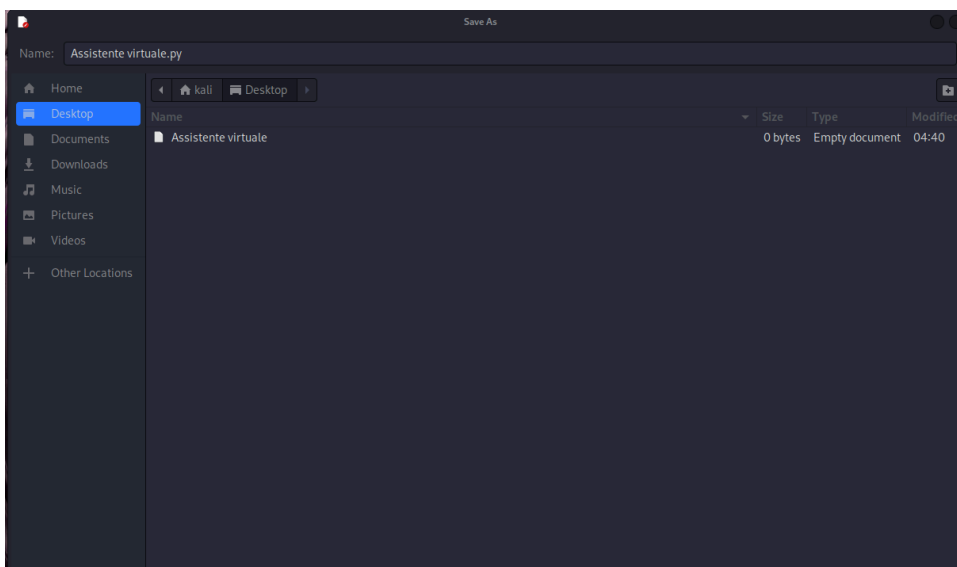
def assistente_virtuale(comando):
    comando = comando.strip() # Rimuove spazi vuoti prima e dopo il comando
    if comando == "Qual è la data di oggi?":
        oggi = datetime.date.today() # Ottieni la data di oggi
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando == "Che ore sono?":
        fuso_orario = pytz.timezone("Europe/Rome") # Imposta il fuso orario
        ora_attuale = datetime.datetime.now(fuso_orario) # Ottiene l'ora corrente nel fuso
        orario specificato
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif comando == "Come ti chiami?":
        risposta = "Mi chiamo Assistente Virtuale"
    else:
        risposta = "Non ho capito la tua domanda." # Risposta generica se il comando non è
        riconosciuto
    return risposta

# Ciclo while per continuare a ricevere comandi dall'utente
while True: # Aggiunti i due punti (:) alla fine della riga
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente.lower() == "esci": # Se l'utente digita "esci", il programma termina
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente)) # Stampa la risposta dell'assistente
```

Proviamo il codice riscritto e corretto , su kali linux



Per eseguire il codice che abbiamo corretto e implementato prima , dobbiamo recarci su kali (con VM o dal bootloader del pc se mai lo abbiamo installato) creare un documento , incollare il codice e salvare il file con l'estensione **.py** che contraddistingue i file python. Una volta salvato e denominato possiamo aprire il file per verificarne il contenuto .



```

~/Desktop/Assistente virtuale.py - Mousepad
File Edit Search View Document Help

1 import datetime
2 import pytz # Importa il modulo pytz per gestire i fusi orari
3
4 def assistente_virtuale(comando):
5     comando = comando.strip() # Rimuove spazi vuoti prima e dopo il comando
6     if comando == "Qual è la data di oggi?":
7         oggi = datetime.date.today() # Ottieni la data di oggi
8         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
9     elif comando == "Che ore sono?":
10        fuso_orario = pytz.timezone("Europe/Rome") # Imposta il fuso orario
11        ora_attuale = datetime.datetime.now(fuso_orario) # Ottiene l'ora corrente nel fuso orario specificato
12        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
13    elif comando == "Come ti chiami?":
14        risposta = "Mi chiamo Assistente Virtuale"
15    else:
16        risposta = "Non ho capito la tua domanda." # Risposta generica se il comando non è riconosciuto
17    return risposta
18
19 # Ciclo while per continuare a ricevere comandi dall'utente
20 while True: # Aggiunti i due punti (:) alla fine della riga
21     comando_utente = input("Cosa vuoi sapere? ")
22     if comando_utente.lower() == "esci": # Se l'utente digita "esci", il programma termina
23         print("Arrivederci!")
24         break
25     else:
26         print(assistente_virtuale(comando_utente)) # Stampa la risposta dell'assistente
27

```

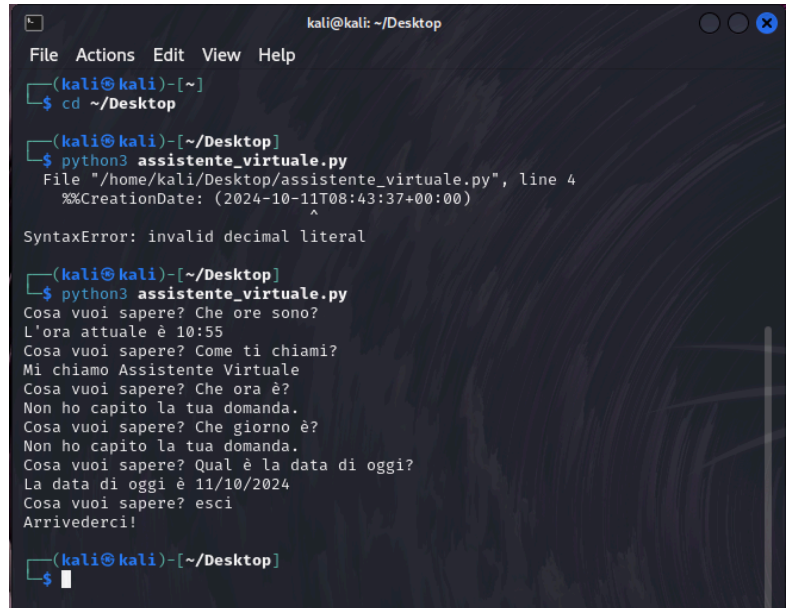
Dopo aver verificato che il contenuto sia quello che effettivamente abbiamo incollato o scritto possiamo procedere con l'apertura del terminale , Possiamo farlo cercando "Terminale" nel menu delle applicazioni o premendo

Ctrl + Alt + T

Una volta sul terminale dobbiamo comunicare al terminale in quale cartella cercare il nostro file `cd ~/Desktop`

Dopo di che apriamo il nostro file con il seguente comando
`python3 assistente_virtuale.py`

che ci permette successivamente di vedere il programma e di eseguirlo .



```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~]
$ cd ~/Desktop
(kali@kali)-[~/Desktop]
$ python3 assistente_virtuale.py
File "/home/kali/Desktop/assistente_virtuale.py", line 4
    %%CreationDate: (2024-10-11T08:43:37+00:00)
    ^
SyntaxError: invalid decimal literal

(kali@kali)-[~/Desktop]
$ python3 assistente_virtuale.py
Cosa vuoi sapere? Che ore sono?
L'ora attuale è 10:55
Cosa vuoi sapere? Come ti chiami?
Mi chiamo Assistente Virtuale
Cosa vuoi sapere? Che ora è?
Non ho capito la tua domanda.
Cosa vuoi sapere? Che giorno è?
Non ho capito la tua domanda.
Cosa vuoi sapere? Qual è la data di oggi?
La data di oggi è 11/10/2024
Cosa vuoi sapere? esci
Arrivederci!

(kali@kali)-[~/Desktop]
$
```