

Correzione Codice Python

```
1 import datetime
2   1
3 while True
4     comando_utente = input("Cosa vuoi sapere? ")
5     if comando_utente == "esci":
6       print("Arrivederci!")
7     2 break
8     else:
9       print(assistente_virtuale(comando_utente))
10    3
11 def assistente_virtuale(comando):
12   if comando == "Qual è la data di oggi?":
13     oggi = datetime.date.today() 4
14     risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
15   elif comando == "Che ore sono?":
16     ora_attuale = datetime.datetime.now().time() 5
17     risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
18   elif comando == "Come ti chiami?":
19     risposta = "Mi chiamo Assistente Virtuale"
20   else:
21     risposta = "Non ho capito la tua domanda."
22   return risposta
23
```

Il presente report analizza il codice di un assistente virtuale utilizzato che utilizza la libreria datetime.

Errori di Sintassi e Logici

Sono stati individuati i seguenti errori:

-Riga 3:

while True:

Sono stati omessi i : che permetto al comando di funzionare correttamente. Ho riportato la versione corretta e individuato nell'immagine precedente l'errore.

-Riga 7:

break

Il comando break, utilizzato per interrompere il ciclo while, è stato indentato erroneamente sotto il comando if. Per funzionare correttamente deve andare sotto il comando while, appunto.

-Riga 9:

Funzione def

La funzione def è stata definita successivamente rispetto al ciclo principale while. Ciò porta il codice a non funzionare correttamente, dunque c'è la necessità di spostare la funzione def nella sezione superiore rispetto al while.

-Riga 13:

datetime.datetime.today

La sintassi della funzione seguente è errata e non porta a compimento l'esecuzione del codice nel momento in cui l'utente utilizza l'opzione "Qual è la data di oggi?". Si riporta al di sopra la formulazione corretta da sostituire.

-Riga 16:

datetime.datetime.now()

Non è proprio classificabile come errore ma potrebbe essere tranquillamente eliminato il metodo .time() in quanto il codice verrebbe eseguito correttamente.

Casistiche Non Standard e Miglioramenti di Sicurezza

Questi elementi rappresentano comportamenti non gestiti che devono essere contemplati per rendere il programma robusto e sicuro.

-Il confronto dei comandi (if comando == "...") è sensibile alle maiuscole/minuscole. Se l'utente digita "ESCI" o "Che Ore Sono?", il programma risponde sempre "Non ho capito la tua domanda.".

Una possibile soluzione può essere quella di utilizzare il **metodo .lower()** in modo da ovviare al problema di minuscole e maiuscole posizionate non come programmato aggiungendo:

`comando_utente_normalizzato = comando_utente.lower()`
e confrontare: *if comando_utente_normalizzato == "esci".*

- Se l'utente inserisce il comando con degli spazi accidentali all'inizio o alla fine (es. " esci " o "Che ore sono? "), il codice fallisce.

Una possibile soluzione può essere quella di utilizzare il **metodo .strip()** per eliminare gli spazi bianchi come di seguito: `comando_utente_normalizzato = comando_utente.strip().lower()`.