

## S2L5

- 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.

● Questo programma rappresenta un semplice assistente virtuale che risponde a tre domande a ciclo infinito: la data corrente, l'ora corrente e il proprio nome. l'utente può continuare a interagire con l'assistente tramite video fino a quando non decidono di uscire, inserendo "esci".

```
1 import datetime
2 def assistente_virtuale(comando):
3     if comando= "Qual è la data di oggi?":
4         oggi = datetime.datetime.today() #1 Primo Errore
5         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
6     elif comando = "Che ore sono?":
7         ora_attuale = datetime.datetime.now().time() #2 Secondo Errore
8         risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
9     elif comando = "Come ti chiami?":
10        risposta = "Mi chiamo Assistente Virtuale"
11    else:|
12        risposta = "Non ho capito la tua domanda." <<<<<
13    return risposta
14 while True #3 Terzo errore
15     comando_utente = input("Cosa vuoi sapere? ")
16     if comando_utente.lower() = "esci":
17         print("Arrivederci!")
18         break
19     else:
20         print(assistente_virtuale(comando_utente))
21
22
23
24
25
26
```

### Program explanation

Il programma come impostato sopra non funziona, ha due problemi principali.

**datetime.datetime.today()** non esiste, la forma corretta è **datetime.date.today()** altrimenti la funzione non può funzionare. **datetime** è una **libreria** standard usata per le date e gli orari, con determinate classi e moduli. Il secondo problema principale è la sintassi di while True (mancano i due punti :). Senza la corretta **sintassi ":"** python non capisce come associare la struttura di controllo come **while,if,for** o anche una funzione e lo script segnerà un errore (**syntax error line 14**). Una volta risolti questi problemi il programma funzionerà correttamente anche se ha delle carenze logiche. Se l'utente sbaglia a scrivere o inserisce un qualcosa di sbagliato non c'è nessuna gestione degli input non validi o errori imprevisti ed inoltre Il menu non è user friendly e non presenta nessuna opzione video oltre a **"cosa vuoi chiedere?"** lasciando l'User senza sapere cosa o meno può fare. A finire **.strip** per evitare spazi all'inizio o alla fine da input sbagliati.

### In-depth

Con **import datetime**, importiamo la libreria con funzioni per lavorare con date e orari.

La funzione **assistente\_virtuale** accetta il parametro chiamato comando.

ha 3 condizioni con risultati diversi, Per la data per l'ora e per il nome.

Viene chiesto un input da parte dell'utente su video e se la risposte sono: **"Qual è la data di oggi?"** chiamerà **datetime.date.today()** (Formattandola con **%d/%m/%Y**) . Invece

per l'orario (con input diverso da parte dell'utente) chiamerà

**datetime.datetime.now().time()** (formattandola con **%H:%M**) e per il nome risponderà

con una stringa "Mi chiamo Assistente Virtuale" I risultati vengono poi mostrati a video,

uguale per l'errore generico se l'utente scriverà qualunque altro input ripetendosi da capo.

### Bug List+Suggestions

**#1** Nel primo errore abbiamo la forma `datetime.datetoday()` errata, bisogna correggerlo con `datetime.date.today`

**#2** Nel secondo "errore" dobbiamo ottenere solo l'ora e `datetime.datetime.now()` fa al caso nostro, poi per formattarla uno usa `%H%M`

**#3** Il terzo errore è una formattazione sbagliata, alla file di `while true` c'è bisogno di `:` per indicare l'inizio di un blocco di codice.

**#4** Non è presente alcuna gestione delle eccezioni per gestire input non validi o errori imprevisti. O un menu a video user friendly.

**#5** Mancanza di un'istruzione di uscita "chiara". Sebbene ci sia un comando per uscire, non è gestito in modo da fornire un feedback all'utente in caso di input non riconosciuto.

**#6** `.strip` per evitare mis-input da parte dell'utente per gli spazi.

# Improving

always

```
1 import datetime
2
3 def assistente_virtuale(comando):
4     comando = comando.strip()
5     if comando == "Qual è la data di oggi?" or comando == "1":
6         oggi = datetime.date.today()
7         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
8     elif comando == "Che ore sono?" or comando == "2":
9         ora_attuale = datetime.datetime.now()
10        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
11    elif comando == "Come ti chiami?" or comando == "3":
12        risposta = "Mi chiamo Assistente Virtuale"
13    else:
14        risposta = "Devi inserire una domanda valida.\nPuoi anche selezionare da 1 a 4."
15    return risposta
16
17 def mostra_menu():
18     print("Benvenuto nel tuo Assistente Virtuale!")
19     print("Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.")
20     print("Attenzione: Case Sensitive")
21     print("1. Qual è la data di oggi?")
22     print("2. Che ore sono?")
23     print("3. Come ti chiami?")
24     print("4. Esci o 4 per terminare il programma")
25
26 while True:
27     mostra_menu()
28     comando_utente = input("Cosa vuoi sapere? ")
29     if comando_utente.strip() == "4" or comando_utente.lower() == "esci":
30         print("Arrivederci!")
31         break
32     else:
33         try:
34             risposta = assistente_virtuale(comando_utente)
35             print(risposta)
36         except Exception as e:
37             print("Si è verificato un errore, riprova:", e)
38
```

## Program Explanation and Improvements

In questa versione aggiornata il programma accetta anche opzioni numeriche oltre alle domande stesse ( con possibilità di aggiungere altre opzioni). Il menu è user friendly e spiega le opzioni disponibili su cosa e non si possa fare. I messaggi di errore sono più chiari e è stato aggiunto un blocco try \_except per gestire gli errori e fornire un feedback utile all'Utente. In poche parole è più interattivo e chiaro su cosa o non fare nel programma.

# Password Generator (EXTRA)

```
1 import random
2 import string
3
4 # psw len
5 lunghezza = int(input("Inserisci la lunghezza della password"))
6
7 # Cosa posso o meno inserire nella password
8 caratteri = string.ascii_letters + string.digits # Tutto
9
10 # Genera la password casuale
11 password = ''.join(random.choice(caratteri) for _ in range(lunghezza)) #specificato dall'utente
12
13 # Mostra la password
14 print("La tua password generata è:", password)
15 |
```

Programma (basic) per una creazione di una password con l'utilizzo di due librerie.

Con `lunghezza = int(input` l'utente inserirà la lunghezza della password, mentre con `string.ascii_letters` e `digits` la password uscirà con qualunque Carattere normale e non o numeri. `for _ in range` per specificare la lunghezza da usare.  
alla fine verrà mostrata la password generata in maniera casuale, all'utente.

File Actions Edit View Help

(rosky@vbox)-[~/Desktop]

\$ python pswgen.py

Inserisci la lunghezza della password10

La tua password generata è: qQWMt5ZRxZ

(rosky@vbox)-[~/Desktop]

\$

(rosky@vbox)-[~/Desktop]

\$ python S2L5IMP.py

Benvenuto nel tuo Assistente Virtuale!

Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.

Attenzione: Case Sensitive

1. Qual è la data di oggi?

2. Che ore sono?

3. Come ti chiami?

4. Esci o 4 per terminare il programma

Cosa vuoi sapere? 1

La data di oggi è 11/10/2024

Benvenuto nel tuo Assistente Virtuale!

Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.

Attenzione: Case Sensitive

1. Qual è la data di oggi?

2. Che ore sono?

3. Come ti chiami?

4. Esci o 4 per terminare il programma

Cosa vuoi sapere? 2

L'ora attuale è 09:19

Benvenuto nel tuo Assistente Virtuale!

Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.

Attenzione: Case Sensitive

1. Qual è la data di oggi?

2. Che ore sono?

3. Come ti chiami?

4. Esci o 4 per terminare il programma

Cosa vuoi sapere? 3

Mi chiamo Assistente Virtuale

Benvenuto nel tuo Assistente Virtuale!

Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.

Attenzione: Case Sensitive

1. Qual è la data di oggi?

2. Che ore sono?

3. Come ti chiami?

4. Esci o 4 per terminare il programma

Cosa vuoi sapere? 4

Arrivederci!

```
1 import datetime
2 def assistente_virtuale(comando):
3     if comando= "Qual è la data di oggi?":
4         oggi = datetime.date.today() #1 Primo Errore
5         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
6     elif comando == "Che ore sono?":
7         ora_attuale = datetime.datetime.now().time() #2 Secondo Errore
8         risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
9     elif comando == "Come ti chiami?":
10        risposta = "Mi chiamo Assistente Virtuale"
11    else:
12        risposta = "Non ho capito la tua domanda." <<<<<
13    return risposta
14 while True: #3 Terzo errore
15     comando_utente = input("Cosa vuoi sapere? ")
16     if comando_utente.lower() == "esci":
17         print("Arrivederci!")
18         break
19     else:
20         print(assistente_virtuale(comando_utente))
21
22
23
24
```

25 Funzionante

26