

Scopo del programma: il programma è programmato per rispondere a 3 domande

1: Qual è la data di oggi?

2: Che ore sono?

3: Come ti chiami?

Il programma continuerà a chiedere “Cosa vuoi sapere?” Fino a quando l’utente non scriverà “**esci**”

Se la richiesta del utente non rispecchia nessuna di queste 3 domande e non richiede di uscire il programma stamperà “**Non ho capito la tua domanda.**”

```
import datetime
```

```
def assistente_virtuale(comando):
```

```
    if comando == "Qual è la data di oggi?": ***
```

```
        oggi = datetime.datetime.today()
```

Errore 1 :

oggi = datetime.datetime.today() -> oggi = datetime.date.today() è richiesto il punto tra date e today per la data odierna.

```
    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")
```

Errore 2: lo \ è un errore ci vanno solo i due punti

* Per rendere più pulito e facile da leggere per l’utente conviene scrivere “L’ora attuale è: “ mettendo i due punti con lo spazio alla fine

```
    elif comando == "Come ti chiami?": ***
```

```
    risposta = "Mi chiamo Assistente Virtuale"
```

```
    else:        risposta = "Non ho capito la tua domanda."
```

```
    return risposta
```

```
while True
```

Errore 3: Dopo il While True sono richiesti i 2 punti

```
comando_utente = input("Cosa vuoi sapere? ")
```

CONSIGLI:

* Sarebbe meglio aggiungere alla fine del input un .lower() per far sì che il programma legga sempre in minuscolo evitando errori se l’utente scrive con la prima lettera maiuscola o minuscola.

* Facendo così vanno cambiate le risposte nei vari if e elif mettendo tutti in minuscolo. (Ho segnalato i punti in questione con “***”)

```
if comando_utente.lower() == "esci":    * Il “.lower” fa sì che la risposta del utente sia tutta in  
minuscolo per evitare errori
```

```
print("Arrivederci!")
```

```
break
```

```
else:
```

```
print(assistente_virtuale(comando_utente))
```