

PROGETTO S2L5 - CONNEGGERE GLI ERRORI AL CODICE

```
1 import datetime
2
3 while True
4     comando_utente = input("Cosa vuoi sapere? ")
5     if comando_utente == "esci":
6         print("Arrivederci!")
7         break
8     else: X
9         ← print(assistente_virtuale(comando_utente))
10
11 def assistente_virtuale(comando):
12     if comando == "Qual è la data di oggi?":
13         oggi = datetime.datetime.today()
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()
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
```

AREA AZZURRA →

Se inserisco il ciclo While ancor prima di dichiarare la funzione, l'ultima print del ciclo while darà errore in quanto non riconosce la funzione assistente_virtuale.

AREA ROSSA →

Mancano i return, risposta da sola non stamperà nulla.

AREA ARANCIO →

- datetime.datetime.today() -> la libreria di Python ci dice che questa funzione deve essere scritta come date.today();
- datetime.datetime.now().time() -> questo stampa l'orario con i millisecondi. Ma noi i millisecondi non li vogliamo quindi facciamo come segue: ora_attuale = datetime.datetime.now().time() print(ora_attuale.strftime("%H:%M:%S"));
- ("%H:%M") -> qui stampiamo solo ore e minuti, ma può essere completo mettere anche i secondi, come mostrato prima.

AREA VERDE ACQUA →

Il brake va messo all'interno dell'if.

AREA GIALLA →

L'else in questo caso risulta essere superfluo, in quanto dopo un break il ciclo esce comunque.

Quindi sarebbe bene toglierlo e andare alla print successiva.

AREA ROSA →

Per il comando dell'uscita, è meglio confrontare in minuscolo, come segue: if comando_utente.lower() == "esci":