

1	import datetime
2	def assistente_virtuale(comando):
3	if comando == "Qual è la data di oggi?":
4	oggi = datetime.date.today()
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()
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
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))

Scopo del programma

Il codice dovrebbe implementare un programma in grado di rispondere a due semplici domande su ora e data ("La data di oggi è ..."; "L'ora attuale è ...") e a fornire un nome di sé stesso ("Mi chiamo Assistente Virtuale") quando richiesto dall'utente, che inserisce le sue domande sotto forma di stringa ("Qual è la data di oggi?"; "Che ore sono?"; "Come ti chiami") in seguito alla richiesta di input da parte del programma ("Cosa vuoi sapere?").

Nel caso in cui l'input inserito dall'utente sia diverso da quanto previsto dal codice, viene restituito il messaggio "Non ho capito la tua domanda" e l'utente viene nuovamente presentato con la richiesta di inserimento di un input.

Il programma fornisce infine la possibilità di terminarsi a seguito di un input "esci", stampando sul terminale il saluto "Arrivederci!" prima di arrestarsi.

Analisi del codice

Il codice si apre con l'importazione del modulo *datetime*, che include le funzioni che permettono di formattare orari e date, attraverso il comando *import* nella riga 1.

Nella riga 2 viene definita una funzione attraverso il comando *def* a cui viene dato il nome *assistente_virtuale* e il parametro *comando*. Successivamente, nelle righe 3-12, attraverso l'utilizzo di un ciclo "if-elif-else", vengono introdotte le istruzioni della funzione che devono essere eseguite da questa quando richiamata all'interno del programma.

Attraverso il ciclo "if-elif-else" è possibile controllare molteplici espressioni fino a trovare quella che restituisce un valore True di modo che il programma esegua la relativa istruzione o blocco di istruzioni. In questo caso:

- il blocco "if" (righe 3-5) definisce come valore True l'input "Qual è la data di oggi?", al quale assegna l'istruzione di ritornare la stringa "La data di oggi è " seguita dalla data corrente

utilizzando il metodo *.strftime*, che crea una stringa rappresentante il tempo secondo un formato specifico (qui, *giorno/mese/anno*);

- il primo blocco “elif” (6-8) definisce come valore True l’input “Che ore sono?” e vi assegna l’istruzione di ritornare la stringa “L’ora attuale è ” seguita dall’ora attuale secondo il format *ore:minuti*;
- il secondo blocco “elif” (9-10) definisce come valore True l’input “Come ti chiami?” e vi assegna l’istruzione di ritornare la stringa “Mi chiamo Assistente Virtuale”;
- infine, il blocco “else” (11-12) istruisce il programma di stampare in terminale la stringa “Non ho capito la tua domanda” nel caso in cui nessuno dei valori dei blocchi precedenti sia risultato True.

La riga 14, fuori dalla funzione, introduce un ciclo “while”. Tale ciclo viene utilizzato per eseguire ripetutamente un blocco di codice fino a quando una data condizione booleana viene valutata come False. Un ciclo “while True” viene pertanto eseguito infinitamente, fintanto che l’input dell’utente non corrisponde alla stringa “esci” (righe 16-18) a cui viene assegnata l’istruzione di terminare il programma (comando *break*).

L’input dell’utente viene richiesto alla linea 15, con il comando *input* che richiede all’utente di inserire una digitazione con prompt “Cosa vuoi sapere?”.

La riga 16 ha una doppia funzione:

1. istituisce la modalità di uscita dal programma attraverso la digitazione della stringa “esci” a cui viene fatto seguire il comando *break*;
2. attraverso l’aggiunta del metodo *.lower()*, restituisce una stringa con caratteri minuscoli qualunque sia il format della digitazione, di modo da uniformare lo stile di scrittura dell’input “esci”.

Il blocco “else” (righe 19-20) viene eseguito in tutti i casi in cui l’input alla riga 16 risultasse False (qualunque digitazione diversa da *esci*): viene qui richiamata la funzione che risponde all’input dell’utente in base a cosa questo digita, come già visto, eseguendo il ciclo “if-elif-else” che costituisce le istruzioni della funzione *assistente virtuale*.

Osservazioni

- Errore di sintassi: riga 14, manca “:” a chiusura della linea di codice.
- Errori logici: il parametro definito dalla funzione alla riga 2 (*comando*) non corrisponde al parametro definito dalla funzione chiamata alla riga 20 (*comando_utente*). Per risolvere si potrebbe uniformare la nomenclature delle variabili sostituendo *comando_utente* con *comando*.
- Casistiche non standard: l’input “esci” alla riga 16 è l’unico input che viene uniformato nello stile di grafia, mentre per tutti gli altri input si presuppone che l’utente digiti correttamente ogni stringa con solamente la prima lettera maiuscola. In caso contrario, qualunque input ritornerebbe la stringa “Non ho capito la tua domanda”. Per rimediare si potrebbe, similmente a quanto fatto nella riga 16 con il metodo *.lower()*, obbligare il programma a restituire sempre

una stringa con solo il primo carattere in maiuscolo di modo che input e stringhe dei comandi combacino, aggiungendo il metodo `.capitalize()` per ogni *comando*. Es:

3	<code>if comando.capitalize() == "Qual è la data di oggi?":</code>
---	--