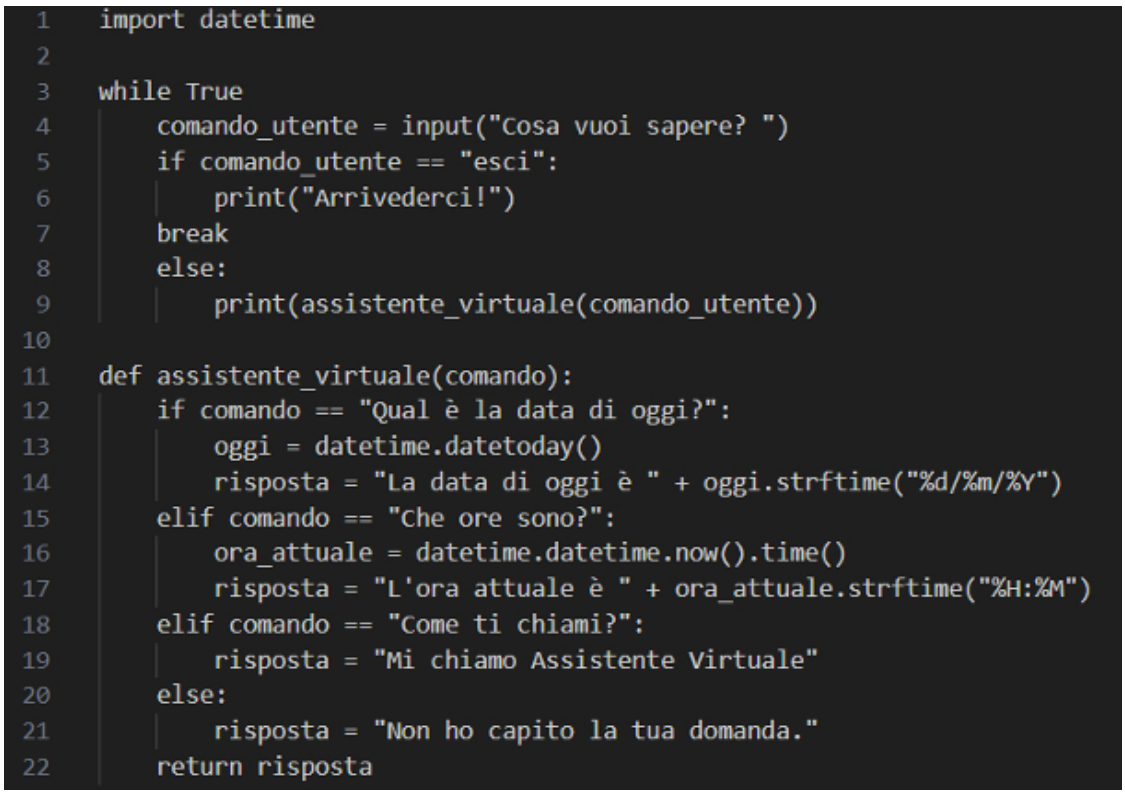
Analisi degli Errori e Suggerimenti UX dello Script Assistente Virtuale

Catalin Ardelean Pop

Obiettivo del documento

Elencare e descrivere gli errori individuati nello script Python dell'assistente virtuale, riportando il numero di riga, il problema riscontrato e la correzione suggerita. Inoltre vengono forniti suggerimenti per migliorare l'esperienza utente (UX) del programma.



```python
import datetime

while True
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente == "esci":
        print("Arrivederci!")
    break
    else:
        print(assistente_virtuale(comando_utente))

def assistente_virtuale(comando):
    if comando == "Qual è la data di oggi?":
        oggi = datetime.datetoday()
        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")
    elif comando == "Come ti chiami?":
        risposta = "Mi chiamo Assistente Virtuale"
    else:
        risposta = "Non ho capito la tua domanda."
    return risposta
```

Figura 1: Screenshot dello script Python dell'assistente virtuale.

Sommario

Il presente documento analizza gli errori principali presenti nello script Python dell'assistente virtuale e propone suggerimenti UX per rendere l'interazione più chiara. Per ogni errore vengono forniti:

- Il numero di riga del codice in cui si verifica l'errore,
- Una descrizione dettagliata del problema riscontrato,
- Suggerimenti concreti per correggere l'errore o migliorare la UX.

1 Descrizione Script

Lo script analizzato è un assistente virtuale in Python progettato per rispondere a domande dell'utente riguardo la data corrente, l'ora e il nome dell'assistente. Gestisce l'interazione con l'utente tramite un ciclo continuo fino a quando viene digitato il comando di uscita, fornendo risposte appropriate o un messaggio di errore per input non riconosciuti.

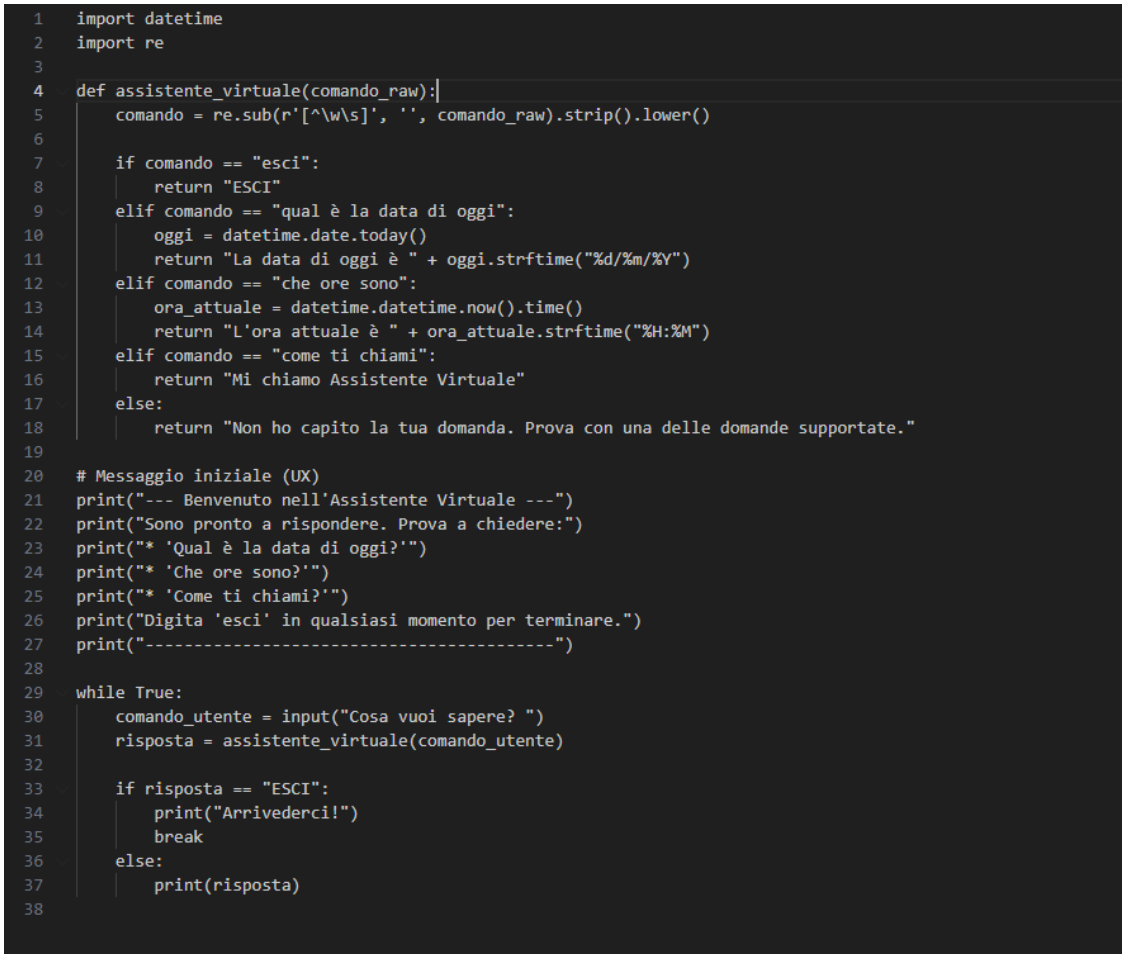
2 Lista degli errori

Riga	Errore	Correzione suggerita
9	La funzione `assistente_virtuale` è definita dopo il suo utilizzo nel ciclo `while`, il che causa errore nel momento dell'esecuzione	Spostare la definizione della funzione prima dell'inizio del ciclo `while`
7	Il `break` è posizionato al di fuori dell'istruzione `if`, interrompendo il ciclo immediatamente	Inserire il `break` all'interno del blocco `if comando_utente == "esci":`
2-20	Non viene gestita la formattazione case sensitive: input in minuscolo o maiuscolo non corrispondono alle domande predefinite	Normalizzare l'input dell'utente usando `.lower()` o simili per confronti case-insensitive
2-20	Spazi iniziali o finali nell'input possono impedire il riconoscimento corretto della domanda	Applicare `.strip()` all'input per rimuovere spazi bianchi iniziali e finali
2-20	La presenza di punteggiatura nell'input può causare errori di riconoscimento	Rimuovere punteggiatura o prevedere la gestione di simboli tramite il modulo `re`

3 Suggerimenti UX

Suggerimento UX	Implementazione proposta
Messaggio di benvenuto chiaro e descrittivo all'avvio	Stampare all'inizio un messaggio che indichi che l'assistente è pronto a rispondere, quali domande può gestire e come terminare, ad esempio: • 'Qual è la data di oggi?' • 'Che ore sono?' • 'Come ti chiami?' • 'Digita "esci" per terminare'
Gestione di input non validi	Stampare un messaggio come "Non ho capito la tua domanda. Prova con una delle domande supportate."
Suggerire miglioramenti futuri	Eventualmente aggiungere altre domande e funzionalità (ad es. data e ora in formato lungo, saluti, calcoli semplici) per rendere l'assistente più utile e interattivo

4 Soluzione aggiornata dello script



```python
import datetime
import re

def assistente_virtuale(comando_raw):
    comando = re.sub(r'[^\w\s]', '', comando_raw).strip().lower()

    if comando == "esci":
        return "ESCI"
    elif comando == "qual è la data di oggi":
        oggi = datetime.date.today()
        return "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando == "che ore sono":
        ora_attuale = datetime.datetime.now().time()
        return "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif comando == "come ti chiami":
        return "Mi chiamo Assistente Virtuale"
    else:
        return "Non ho capito la tua domanda. Prova con una delle domande supportate."

# Messaggio iniziale (UX)
print("--- Benvenuto nell'Assistente Virtuale ---")
print("Sono pronto a rispondere. Prova a chiedere:")
print("* 'Qual è la data di oggi?'")
print("* 'Che ore sono?'")
print("* 'Come ti chiami?'")
print("Digita 'esci' in qualsiasi momento per terminare.")
print("-----------------------------------------")

while True:
    comando_utente = input("Cosa vuoi sapere? ")
    risposta = assistente_virtuale(comando_utente)

    if risposta == "ESCI":
        print("Arrivederci!")
        break
    else:
        print(risposta)

```

Figura 2: Screenshot della versione aggiornata dello script.

Questa versione dello script presenta le seguenti migliorie:

- **Correzioni tecniche**: la funzione `assistente_virtuale` è definita prima del ciclo `while`, il `break` è collocato correttamente all'interno dell'istruzione `if`, e l'input dell'utente viene normalizzato per evitare errori di case, spazi o punteggiatura.
- **Miglioramenti UX**: messaggio di benvenuto chiaro che guida l'utente sulle domande disponibili, gestione di input non riconosciuti con messaggi esplicativi, e indicazioni su come terminare l'interazione.

Conclusione

Questo documento presenta un'analisi dettagliata degli errori dello script e suggerimenti pratici per migliorare l'esperienza utente. La combinazione di correzioni tecniche e indicazioni UX consente di ottenere un assistente virtuale più robusto, intuitivo e facile da usare.