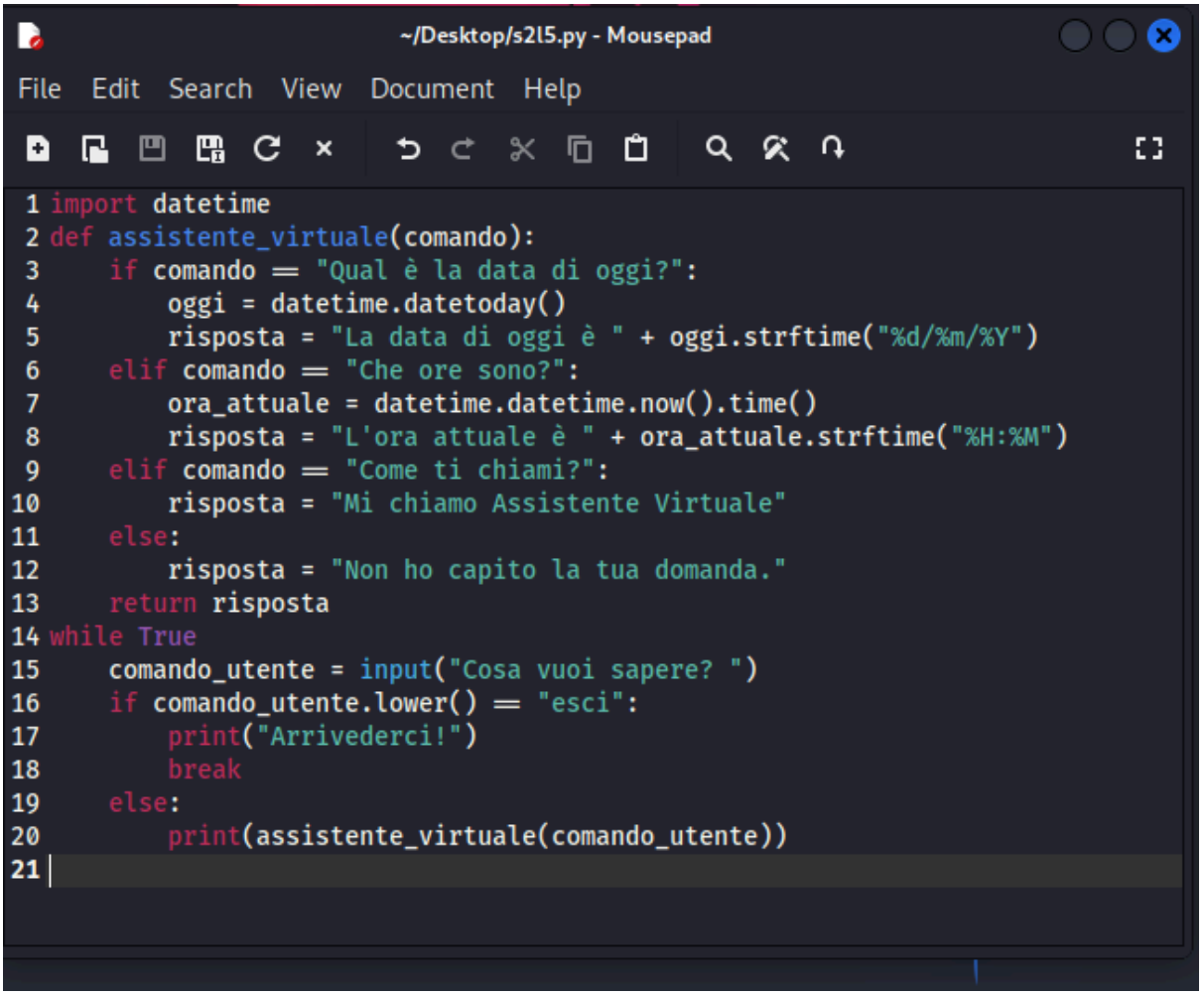
Project - S2L5 - Python Debugging Session

Today's project involved debugging a code aiming to provide the user with a personal assistant for current date and time, also adding a way to discover the program's name and a way to close the program. Hereunder you can find the original code provided:



Right away, we can see the import of the correct module in the first row: this gives Python the possibility of accessing all of the functionalities needed to provide the services aforementioned. The main functions of Python, opposed to C, are already included, so there's no need to import those as well. We then define the "assistente_virtuale" function to accept the parameter "comando".

The coder has now decided to proceed with the if/elif/else statement so that with varying conditions the user can have the three choices we listed before: this works well with the final objective of the program itself.

The first part of the if/elif/else statement, starting at row 3, tells us that if the user asks the question "Qual è la data di oggi?" the program will answer with today's date. We can find here the first bug as a syntax error: "datetime.datetoday" is not a valid command, as the correct syntax from the datetime module should be "datetime.date.today". By fixing this, we

can receive the expected answer, as the code is correctly indented as per Python's syntax under the "if" row, and all of the parameters are correctly assigned through "=".

The second part of the if/elif/else statement, starting with the elif at row 6, tells us that if the user asks the question "Che ore sono?", the program will answer with the current time. By testing it out, we can find that there's no issue with this part of the code: it works as intended and provides the user with the current time, as it's correctly indented and assigned.

The third part of the if/elif/else statement, starting with the elif at row 9, tells us that if the user asks the question "Come ti chiami?", the program will answer with its name. This works well as the program only needs to send back the string "Mi chiamo Assistente Virtuale", and there are no issues at this stage as it's correctly indented and assigned.

The fourth and final part of the if/elif/else statement, starting with the else at row 11, tells us that if there is any other answer except the ones expected, the program won't recognize the input and ask the user to correctly ask the question. We encounter here a second bug, which is a logical one: the user was never provided with a list of the available options, and this can therefore cause numerous user errors; although this part is later in the code it is worth noting now. We'll go over how to properly correct this when we reach row 15. Other than that, being very similar to the third part of the statement, it works as intended syntax-wise, as it's correctly indented and assigned.

We now reach the end of the first indentation, with a return command asking the program to go back to the start whenever it reaches the "risposta" parameter. This works properly as it's indented correctly and on the same levels as the others: if it were indented differently, it wouldn't work anymore as it wouldn't find the "risposta" parameter to return to.

At row 14, we go back to the first level of the code by erasing the indentation, and we ask the program through the "while" cycle to repeat the welcome message as long as the condition is true: considering that the following condition is a printed string expecting a user input, the condition will always be True and it should repeat endlessly until we give a command to end the cycle. By analyzing this row, we can discover the third bug as only "while True" is not the correct syntax: the issue lies in the fact that if we have to associate a condition to this statement, we have to correctly indent it but more importantly request it through the use of ":", as the colon itself signals the start of a new block of code in Python. We can now therefore add the ":" at the end of the statement and it will work as intended.

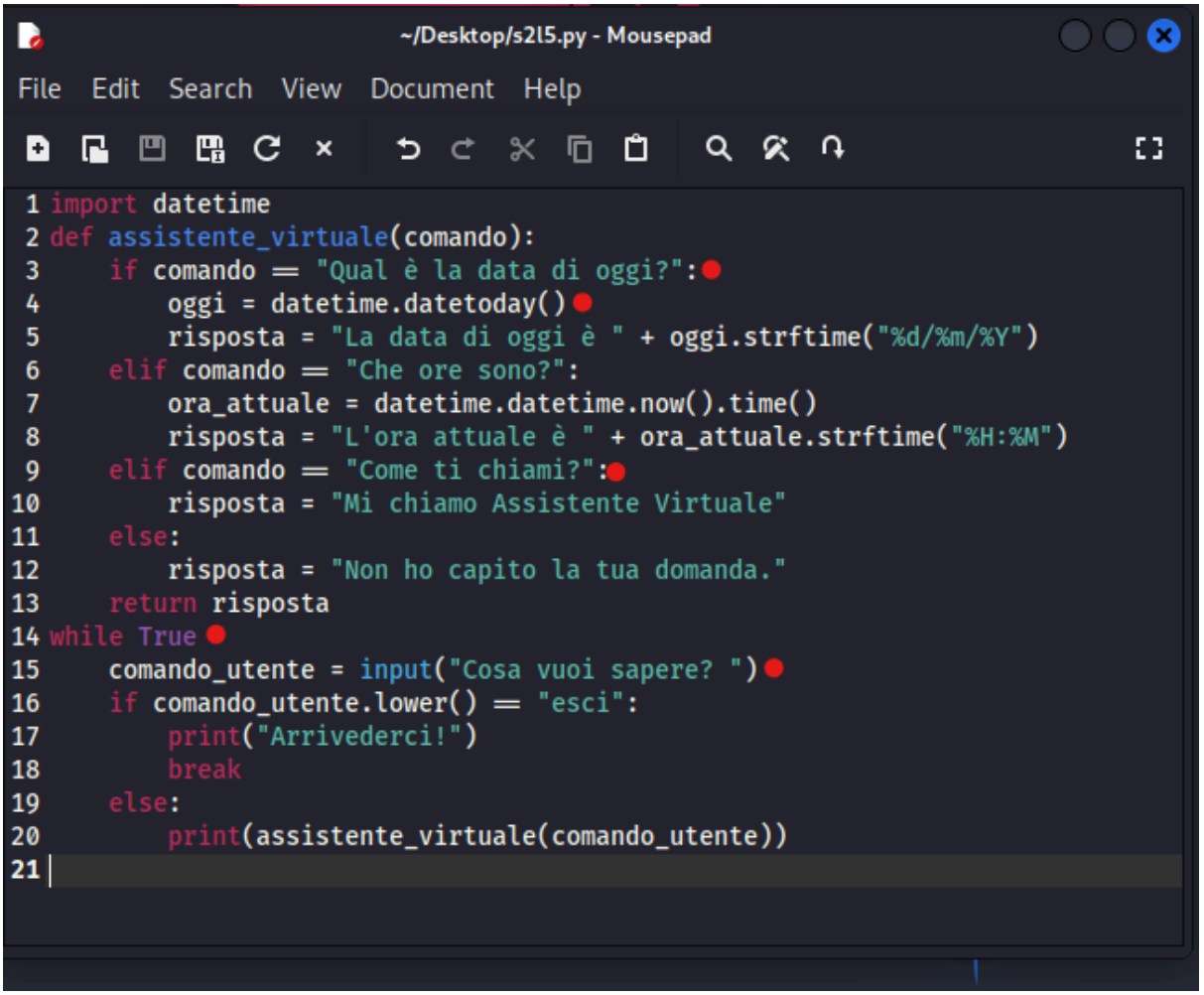
Now that we've reached row 15, we have the welcome message to the user. As the main function of a program is to be utilized by an user, it would be safer to provide a list of available options as a new user at this stage wouldn't know what the program is for, what the commands to make it work are, and not even what or who he's talking to. We have to assume that no instructions were provided beforehand and therefore the program wouldn't be able to be used at all. Moreover, it would be even safer to have the user choose between a list of numbers, and replacing the questions "Qual è la data di oggi?","Che ore sono?" and "Come ti chiami?" respectively with numbers each corresponding to a question at rows 3, 6 and 9 to minimize user error even more. I personally decided to replace this string with "Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per sapere come mi chiamo. Se vuoi uscire dal programma,

digita esci.”. We know correct rows 3, 6 and 9 with “if comando == “1/2/3”” to match the previous message. The string otherwise works as intended, as it’s expecting an input from the user as described in the previous rows.

Rows 16 through 18 are telling us that if the answer provided is “esci”, the program will break the while cycle which would otherwise run in an infinite loop, adding a goodbye string and the “.lower” parameter to allow for more flexibility in the user answer. It works correctly and with no issue.

The last part of the code at row 19 tells us that if we get any other answer from the user other than “esci”, it will print the first part of the code we defined. This works well and without issue.

In conclusion, you can find here a screenshot representing a list of the bugs with a red mark beside the corresponding row:



Hereunder instead, you can find a screenshot of the proposed correct version of the code, with the addition of user friendly features as previously specified:

main.py

Share Run

```
1  import datetime
2  def assistente_virtuale(comando):
3      if comando == "1":
4          oggi = datetime.date.today()
5          risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y") + "."
6      elif comando == "2":
7          ora_attuale = datetime.datetime.now().time()
8          risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M") + "."
9      elif comando == "3":
10         risposta = "Mi chiamo EpicHelp!"
11     else:
12         risposta = "Non ho capito la tua richiesta: per favore digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora
               attuale, digita 3 per sapere come mi chiamo. Se vuoi uscire dal programma, digita esci. "
13     return risposta
14 while True:
15     comando_utente = input("Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere
           l'ora attuale, digita 3 per sapere come mi chiamo. Se vuoi uscire dal programma, digita esci. " )
16     if comando_utente.lower() == "esci":
17         print("Arrivederci!")
18         break
19     else:
20         print(assistente_virtuale(comando_utente))
```

And a demonstration of how the corrected code works:

Output Clear

```
Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per sapere
    come mi chiamo. Se vuoi uscire dal programma, digita esci. 1
La data di oggi è 11/10/2024.
Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per sapere
    come mi chiamo. Se vuoi uscire dal programma, digita esci. 2
L'ora attuale è 10:13.
Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per sapere
    come mi chiamo. Se vuoi uscire dal programma, digita esci. 3
Mi chiamo EpicHelp!
Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per sapere
    come mi chiamo. Se vuoi uscire dal programma, digita esci. asdfghijkl
Non ho capito la tua richiesta: per favore digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per
    sapere come mi chiamo. Se vuoi uscire dal programma, digita esci.
Ciao! Sono il tuo assistente virtuale: digita 1 per conoscere la data di oggi, digita 2 per conoscere l'ora attuale, digita 3 per sapere
    come mi chiamo. Se vuoi uscire dal programma, digita esci. esci
Arrivederci!

=== Code Execution Successful ===
```

This way we achieved through simple debugging a clearer, more user friendly code that minimizes user error.