Working with Tuples: Indexing, Negative Indexing, and Immutability

Description

This project is a practical demonstration of Python tuple operations, including positive and negative indexing for data retrieval and the concept of immutability. Includes examples with plant names, showing how to access items, use reverse indexing, and delete entire tuples. The project highlights strong understanding of Python's built-in data structures.

How do you access an element from a tuple? Give an example (embed a screenshot of your code)

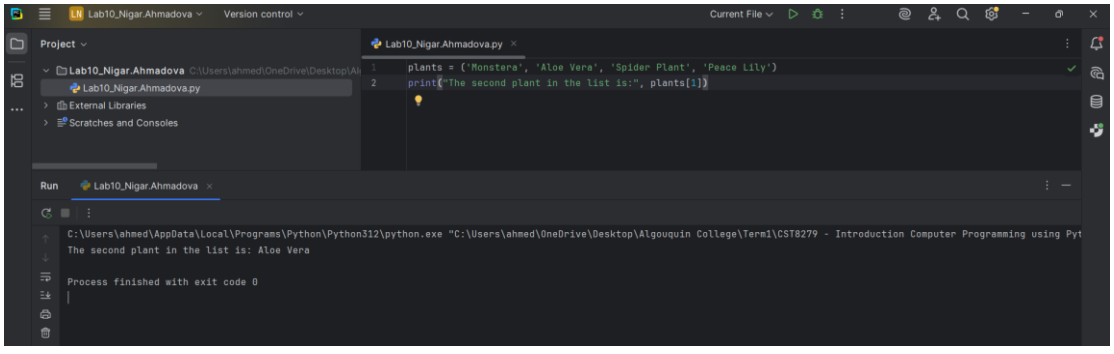
It is possible to access an element from a tuple using indexing. Tuple indexing starts with 0 for the first element, then 1 for the second element, and so on. To retrieve an item, we need to use square brackets with the index number.

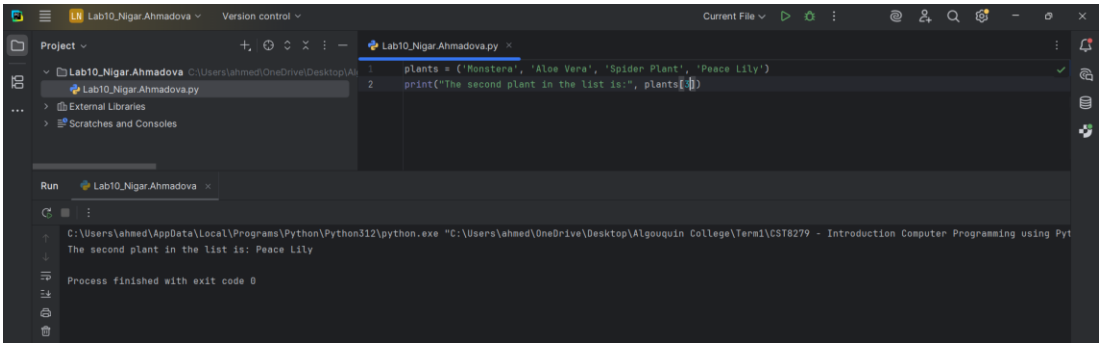
You can see in my example, where I created a tuple with 4 plants: Monstera, Aloe Vera, Spider Plant, and Peace Lily. If I want, for example, to get Aloe Vera (the second plant), I need to use plants[1]. Or if I want to retrieve the Peace Lily, I need to use plants[3].

Example: Accessing a plant name from a tuple

plants = ('Monstera', 'Aloe Vera', 'Spider Plant', 'Peace Lily')

print("The second plant in the list is:", plants[1])





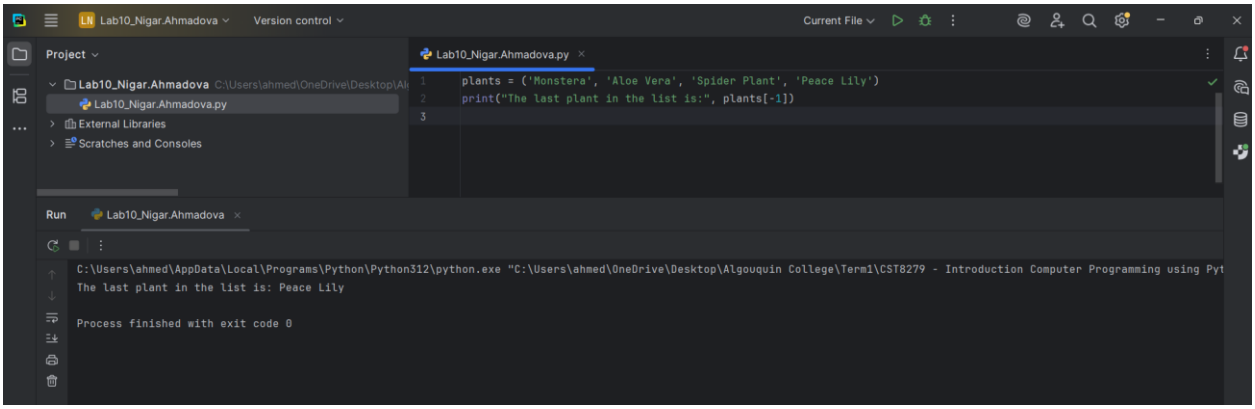
Does tuple allow negative indexing? How? Give an example (embed a screenshot of your code)

Yes, tuples let you use negative indexing. Negative indexes start from -1 for the last item, -2 for the second last, and so on. In my example, I used a tuple of plants again. When I write plants[-1], it gives me the last item in the tuple, which is “Peace Lily”.

Example: Negative indexing in a tuple

plants = ('Monstera', 'Aloe Vera', 'Spider Plant', 'Peace Lily')

print("The last plant in the list is:", plants[-1])



How does deleting work in a tuple? Give an example (embed a screenshot of your code)

Tuples are immutable, which means I can't change or delete individual items inside them. But I can delete the whole tuple itself using the del keyword. In my example, I created a plants tuple, printed it, then deleted it with *del plants*. After deleting, if I try to print the same tuple again, Python gives an error because it has been deleted.

Example: Deleting a tuple

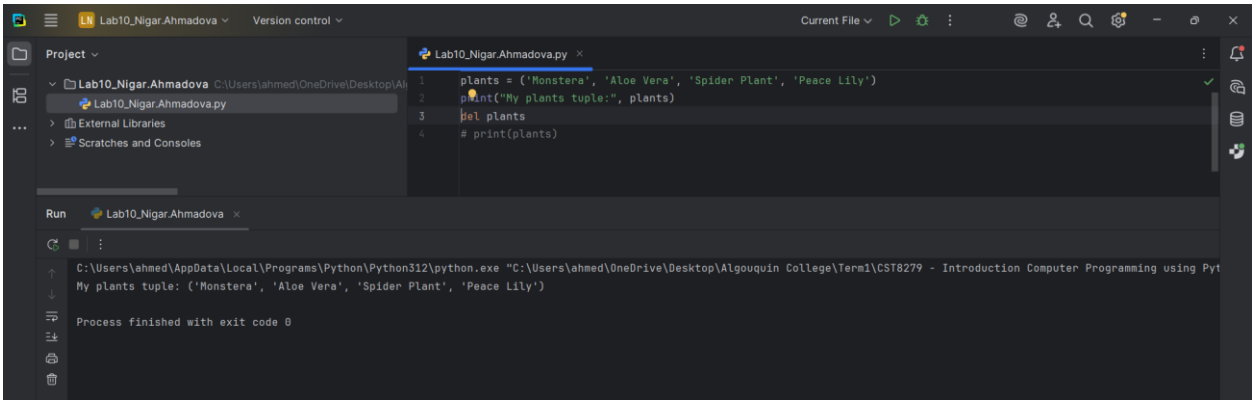
plants = ('Monstera', 'Aloe Vera', 'Spider Plant', 'Peace Lily')

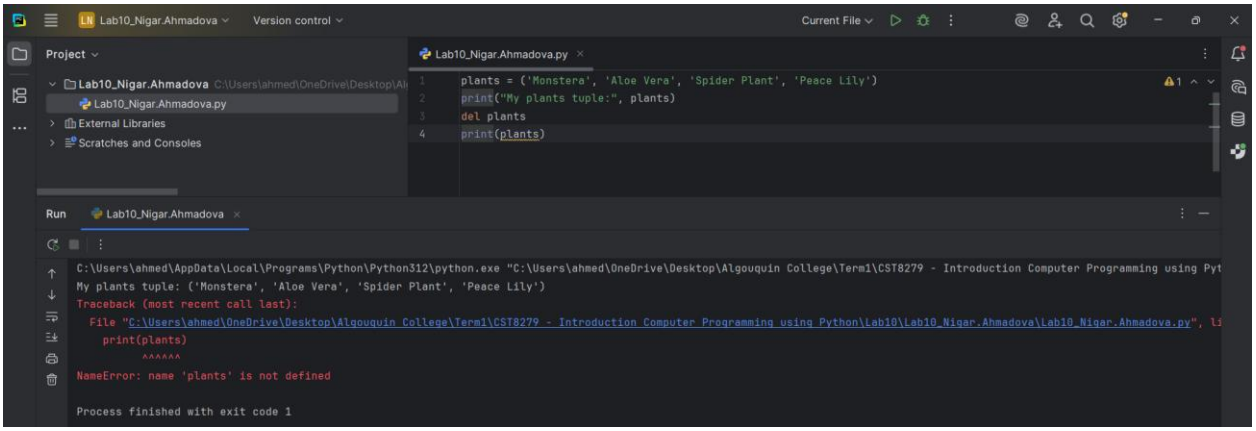
print("My plants tuple:", plants)

del plants

 # Uncommenting this line would cause an error because the plants tuple no longer exists

 # print(plants)





References

Python Software Foundation. (n.d.). *Built-in Types — Python 3.10.13 documentation*. Retrieved from https://docs.python.org/3/library/stdtypes.html#tuple

JetBrains. (n.d.). *PyCharm: The Python IDE for Professional Developers*. Retrieved from https://www.jetbrains.com/pycharm/

Purdue Online Writing Lab. (n.d.). *APA Formatting and Style Guide (7th Edition)*. Retrieved from https://owl.purdue.edu/owl/research_and_citation/apa_style/apa_formatting_and_style_guide/general_format.html