

## Day 14 Task:

### Python Data Types and Data Structures for DevOps

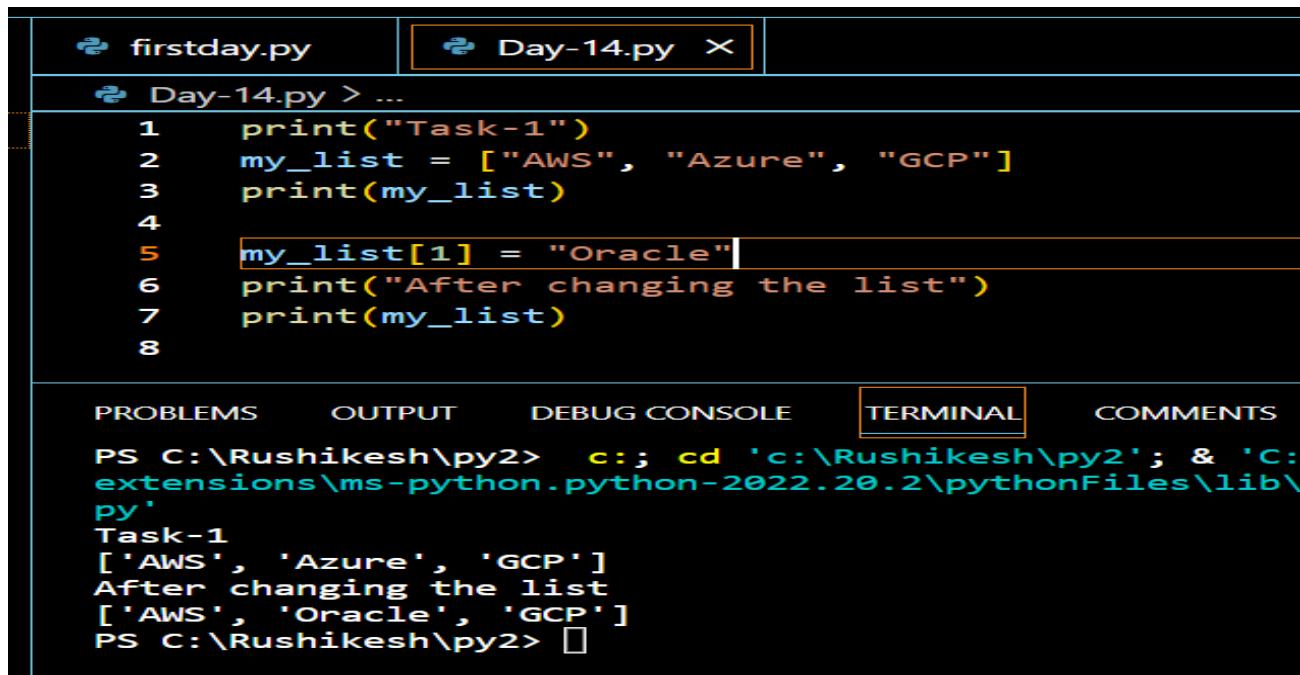
#### Tasks :

1. Give the Difference between List, Tuple and set. Do Handson and put screenshots as per your understanding.

➔ According to my understanding

- A **list** is a collection of items in a specific order, like a shopping list. You can add, remove, or change items in a list.
- A **tuple** is similar to a list, but once you create it, you can't change it. Think of it like a list of things that are fixed and cannot be changed.
- A **set** is a collection of unique items, like a set of unique numbers. If you try to add something that is already in the set, it will not be added again.

Examples :

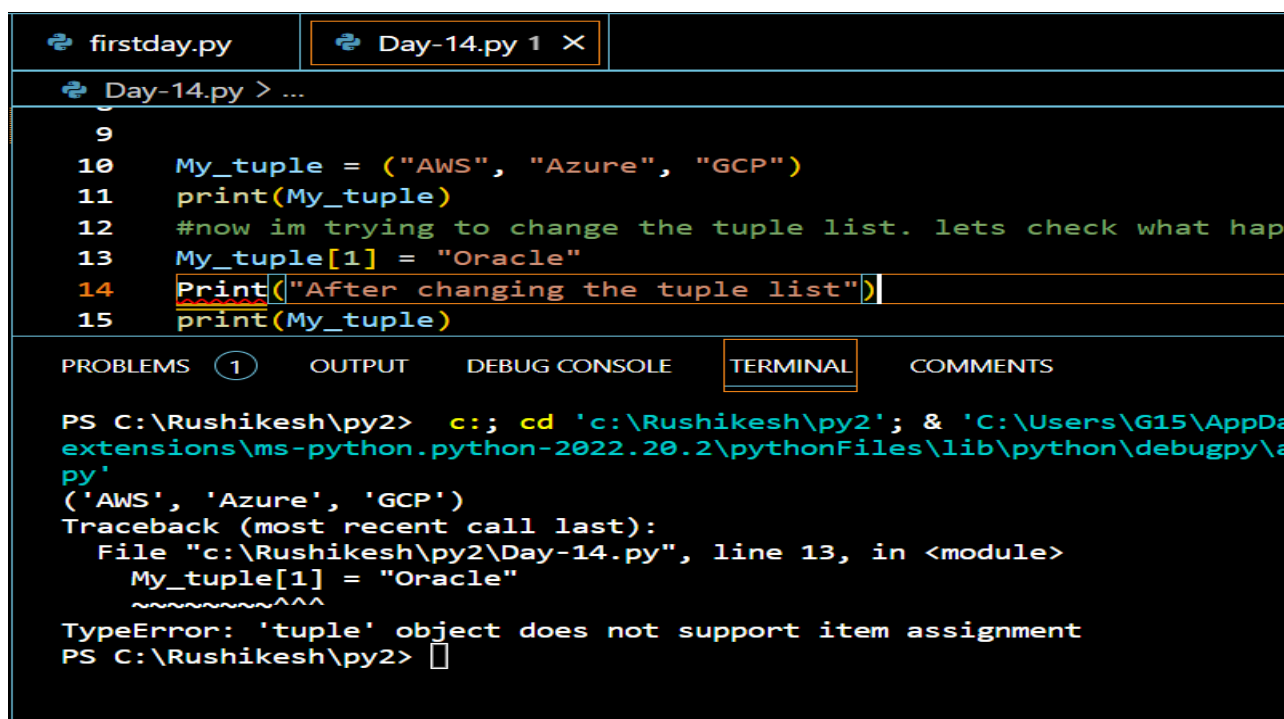


The screenshot shows a Python IDE with two tabs: 'firstday.py' and 'Day-14.py'. The 'Day-14.py' tab is active, showing a script that demonstrates list operations. The script is as follows:

```
1 print("Task-1")
2 my_list = ["AWS", "Azure", "GCP"]
3 print(my_list)
4
5 my_list[1] = "Oracle"
6 print("After changing the list")
7 print(my_list)
8
```

Below the code editor, there are five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'COMMENTS'. The 'TERMINAL' tab is active, showing the output of the script:

```
PS C:\Rushikesh\py2> c:; cd 'c:\Rushikesh\py2'; & 'C:\extensions\ms-python.python-2022.20.2\pythonFiles\lib\python\python.exe'
Task-1
['AWS', 'Azure', 'GCP']
After changing the list
['AWS', 'Oracle', 'GCP']
PS C:\Rushikesh\py2>
```



The screenshot shows a Python IDE with two tabs: 'firstday.py' and 'Day-14.py 1'. The 'Day-14.py 1' tab is active, showing a script that demonstrates tuple operations. The script is as follows:

```
9
10 My_tuple = ("AWS", "Azure", "GCP")
11 print(My_tuple)
12 #now im trying to change the tuple list. lets check what happens
13 My_tuple[1] = "Oracle"
14 Print("After changing the tuple list")
15 print(My_tuple)
```

Below the code editor, there are five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'COMMENTS'. The 'TERMINAL' tab is active, showing the output of the script and a traceback error:

```
PS C:\Rushikesh\py2> c:; cd 'c:\Rushikesh\py2'; & 'C:\Users\G15\AppData\Local\Programs\Python\Python311\python.exe'
('AWS', 'Azure', 'GCP')
Traceback (most recent call last):
  File "c:\Rushikesh\py2\Day-14.py", line 13, in <module>
    My_tuple[1] = "Oracle"
    ~~~~~^
TypeError: 'tuple' object does not support item assignment
PS C:\Rushikesh\py2>
```

```
firstday.py Day-14.py X
Day-14.py > ...
14 # Print("After changing the tuple list")
15 # print(My_tuple)
16
17 my_set = {"AWS", "Azure", "GCP"}
18 print(my_set)
19 my_set.add("AWS")
20 print("After Adding duplicate value to the set")
21 print(my_set)
22 my_set.add("Oracle")
23 print((variable) my_set: set[str] to the set")
24 print(my_set)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS C:\Rushikesh\py2> c:; cd 'c:\Rushikesh\py2'; & 'C:\Users\G15\AppData\Local\Programs\Python\Python311\python.exe' 'C:\Users\G15\AppData\Local\Programs\Python\Python311\pythonFiles\lib\python\debugpy\adapter\..\..\python\python.exe'
{'GCP', 'AWS', 'Azure'}
After Adding duplicate value to the set
{'GCP', 'AWS', 'Azure'}
After Adding Another value to the set
{'Oracle', 'GCP', 'AWS', 'Azure'}
PS C:\Rushikesh\py2>
```

2. Create below Dictionary and use Dictionary methods to print your favourite tool just by using the keys of the Dictionary.



```
firstday.py Day-14-1.py Day-14-2.py X
Day-14-2.py > ...
1 fav_tools = {
2     1:"Linux",
3     2:"Git",
4     3:"Docker",
5     4:"Kubernetes",
6     5:"Terraform",
7     6:"Ansible",
8     7:"Chef"
9 }
10
11 fav_tool = 5
12 print("My favorite tool is:", fav_tools[fav_tool])
13 print("after running another keys of the Dictionary")
14 Secind_fav_tool = 1
15 print("My Second favorite tool is:", fav_tools[Secind_fav_tool])
16
17

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
PS C:\Rushikesh\py2> c:; cd 'c:\Rushikesh\py2'; & 'C:\Users\G15\AppData\Local\Programs\Python\Python311\python.exe' 'C:\Users\G15\AppData\Local\Programs\Python\Python311\pythonFiles\lib\python\debugpy\adapter\..\..\python\python.exe'
My favorite tool is: Terraform
after running another keys of the Dictionary
My Second favorite tool is: Linux
PS C:\Rushikesh\py2>
```

3. Create a List of cloud service providers eg. `cloud_providers = ["AWS","GCP","Azure"]`

Write a program to add Digital Ocean to the list of `cloud_providers` and sort the list in alphabetical order.



firstday.py	Day-14-1.py	Day-14-3.py X	Day-14-2.py
-------------	-------------	---------------	-------------

```
Day-14-3.py > ...
1  cloud_providers = ["AWS","GCP","Azure"]
2  print("List of cloud providers:", cloud_providers)
3  print(" ")
4
5  cloud_providers.sort()
6  cloud_providers.append("Digital Ocean")
7  cloud_providers.sort()
8  print("after sorting & adding cloud providers")
9  print("List of cloud providers:", cloud_providers)
10 print(" ")
11
12 cloud_providers.extend(["Oracle"])
13 cloud_providers.sort()
14 print("after sorting & adding one more cloud providers")
15 print("List of cloud providers:", cloud_providers)
16
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	COMMENTS
----------	--------	---------------	----------	----------

```
PS C:\Rushikesh\py2> c:; cd 'c:\Rushikesh\py2'; & 'C:\Users\G15\AppData\Local\Programs\Python\Python39\python.exe' 'C:\Users\G15\AppData\Local\Programs\Python\Python39\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher'
List of cloud providers: ['AWS', 'GCP', 'Azure']

after sorting & adding cloud providers
List of cloud providers: ['AWS', 'Azure', 'Digital Ocean', 'GCP']

after sorting & adding one more cloud providers
List of cloud providers: ['AWS', 'Azure', 'Digital Ocean', 'GCP', 'Oracle']
PS C:\Rushikesh\py2>
```

In the above code, I first created a list `cloud_providers` containing **AWS**, **GCP**, **Azure** then I used the **append** method to add **Digital Ocean** to the list, and then I used the **sort** method to sort the list in alphabetical order. Finally, I printed the list to check the result.

Alternatively, the **.extend()** method can also be used to add elements to a list by adding the elements of an iterable like a list or tuple to the list it is called on.