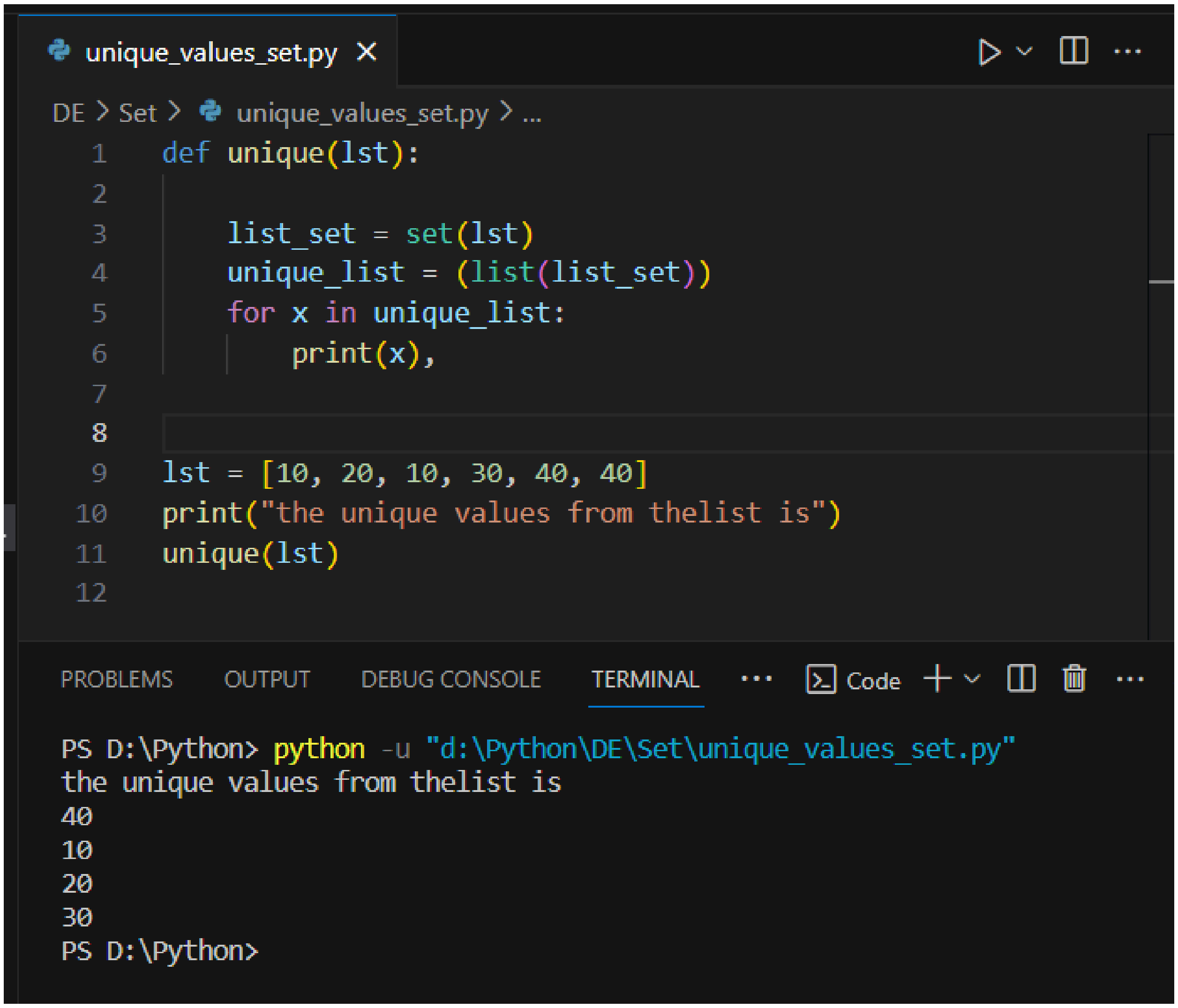
DAY-9 (Python Assignment - 3)
Mitushi Vishwakarma

- **Get unique values from list using map and set**
  - Insert the values of the list in a set.
  - Set only stores a value once even if it is inserted more than once.
  - At end, convert the set into list.

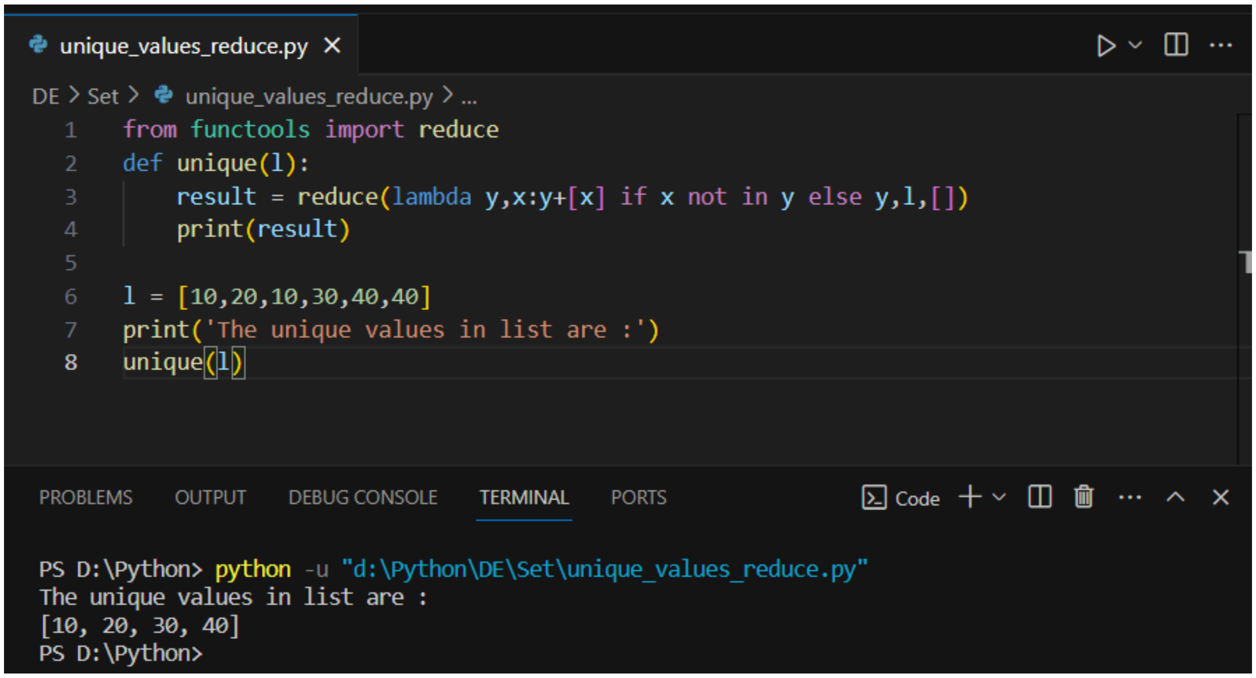


```python
def unique(lst):

    list_set = set(lst)
    unique_list = (list(list_set))
    for x in unique_list:
        print(x),


lst = [10, 20, 10, 30, 40, 40]
print("the unique values from thelist is")
unique(lst)
```

```
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_set.py"
the unique values from thelist is
40
10
20
30
PS D:\Python>
```

- ## Get Unique Values From a List in Python Using reduce() function

  Using reduce() from functools module to iterate over list to check for duplicates.

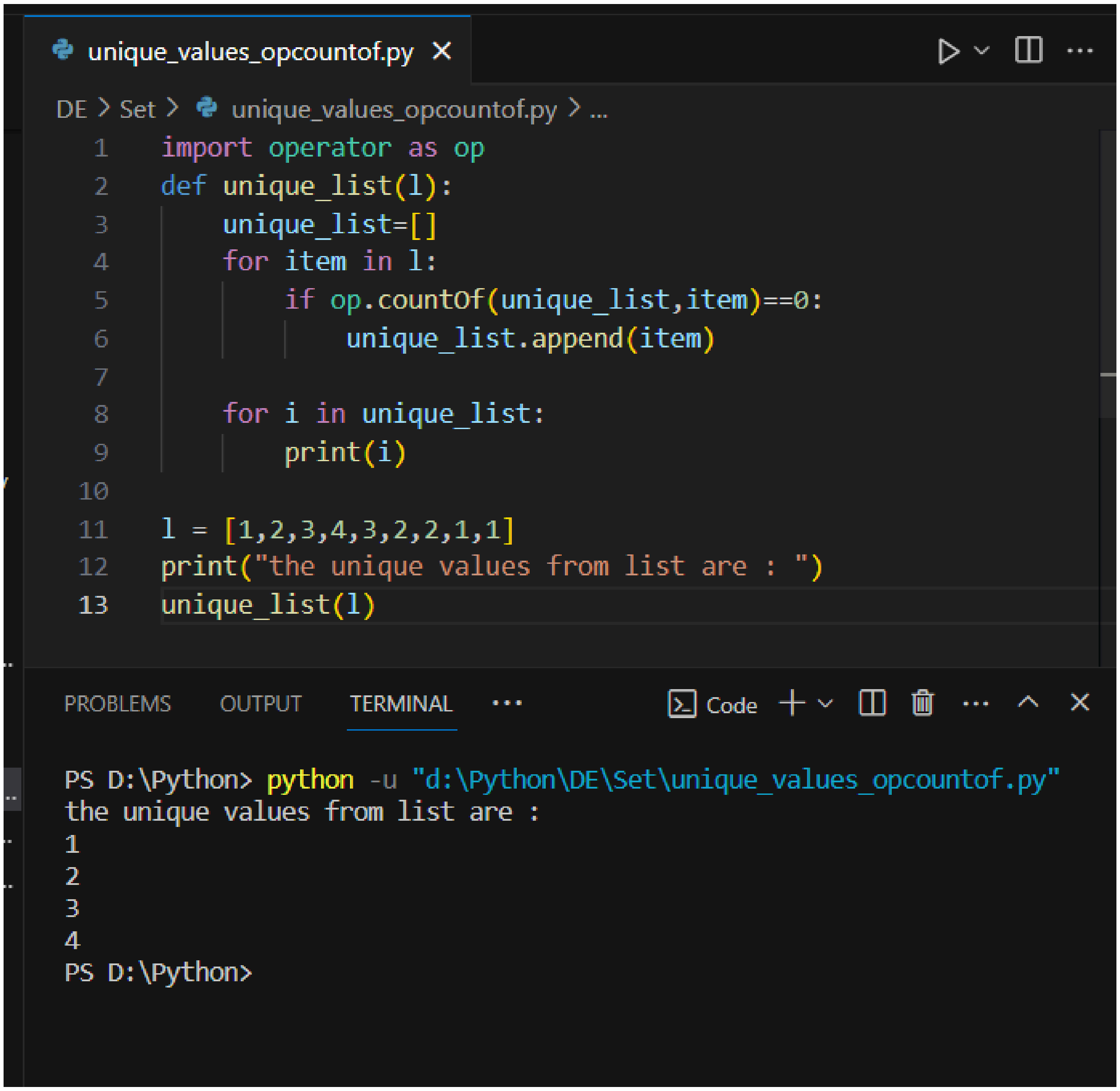


```python
from functools import reduce
def unique(l):
    result = reduce(lambda y,x:y+[x] if x not in y else y,l,[])
    print(result)

l = [10,20,10,30,40,40]
print('The unique values in list are :')
unique(l)
```

```
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_reduce.py"
The unique values in list are :
[10, 20, 30, 40]
PS D:\Python>
```

- ## Get Unique Values From a List in Python Using Operator.countOf() method

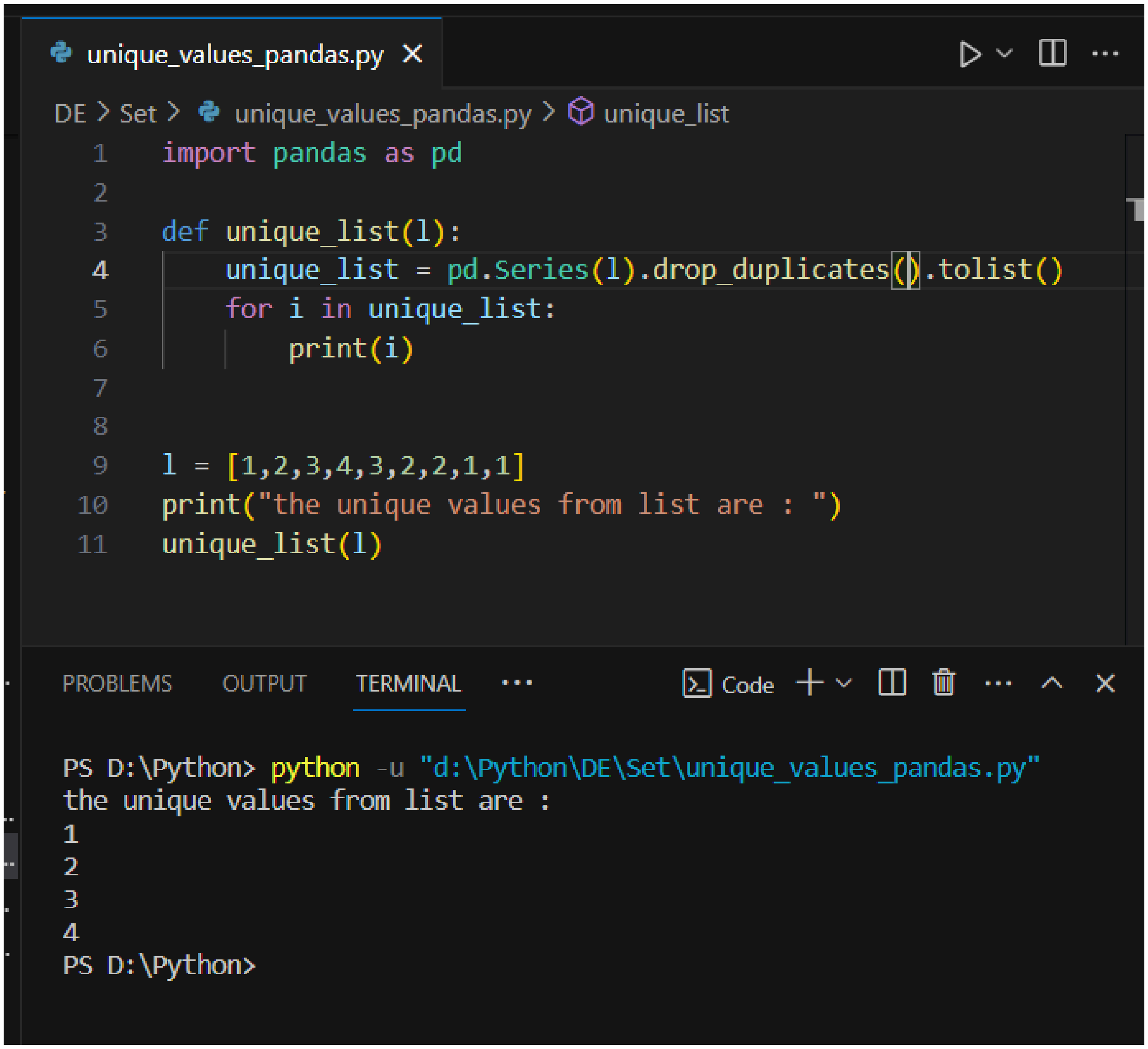
We can get unique list of values using countOf method from operator module which counts the occurrence of value passed into it.


unique_values_opcountof.py
DE > Set > unique_values_opcountof.py > ...
PROBLEMS
OUTPUT
TERMINAL
Code

```python
import operator as op
def unique_list(l):
    unique_list=[]
    for item in l:
        if op.countOf(unique_list,item)==0:
            unique_list.append(item)

    for i in unique_list:
        print(i)

l = [1,2,3,4,3,2,2,1,1]
print("the unique values from list are : ")
unique_list(l)
```

```
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_opcountof.py"
the unique values from list are :
1
2
3
4
PS D:\Python>
```

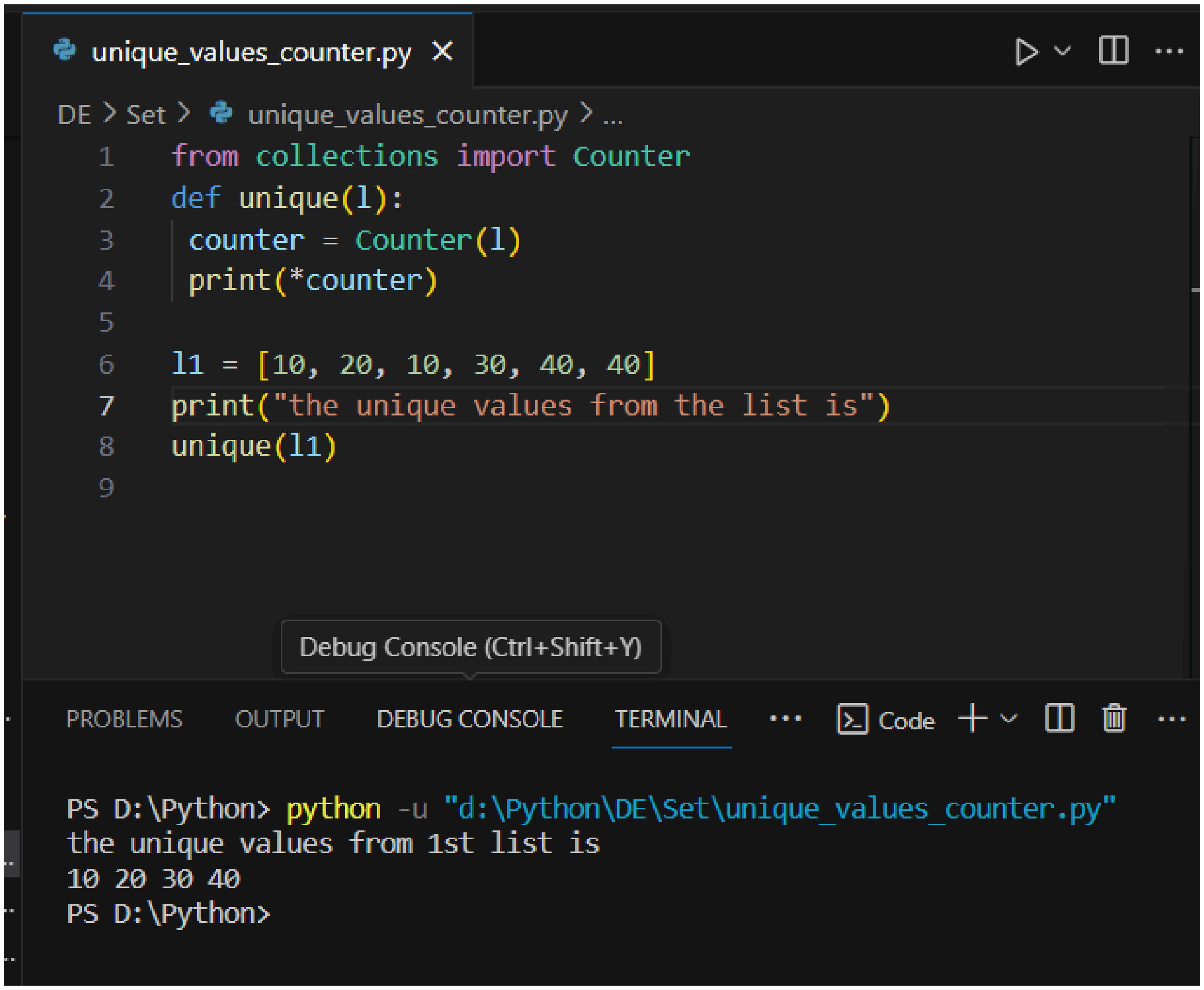
- **Get Unique Values From a List in Python Using pandas module :** Here we use pandas to create a series and passing the list in the series and then using drop_duplicates() to remove the duplicate occurrences.



```python
import pandas as pd

def unique_list(l):
    unique_list = pd.Series(l).drop_duplicates().tolist()
    for i in unique_list:
        print(i)


l = [1,2,3,4,3,2,2,1,1]
print("the unique values from list are : ")
unique_list(l)
```

```
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_pandas.py"
the unique values from list are :
1
2
3
4
PS D:\Python>
```

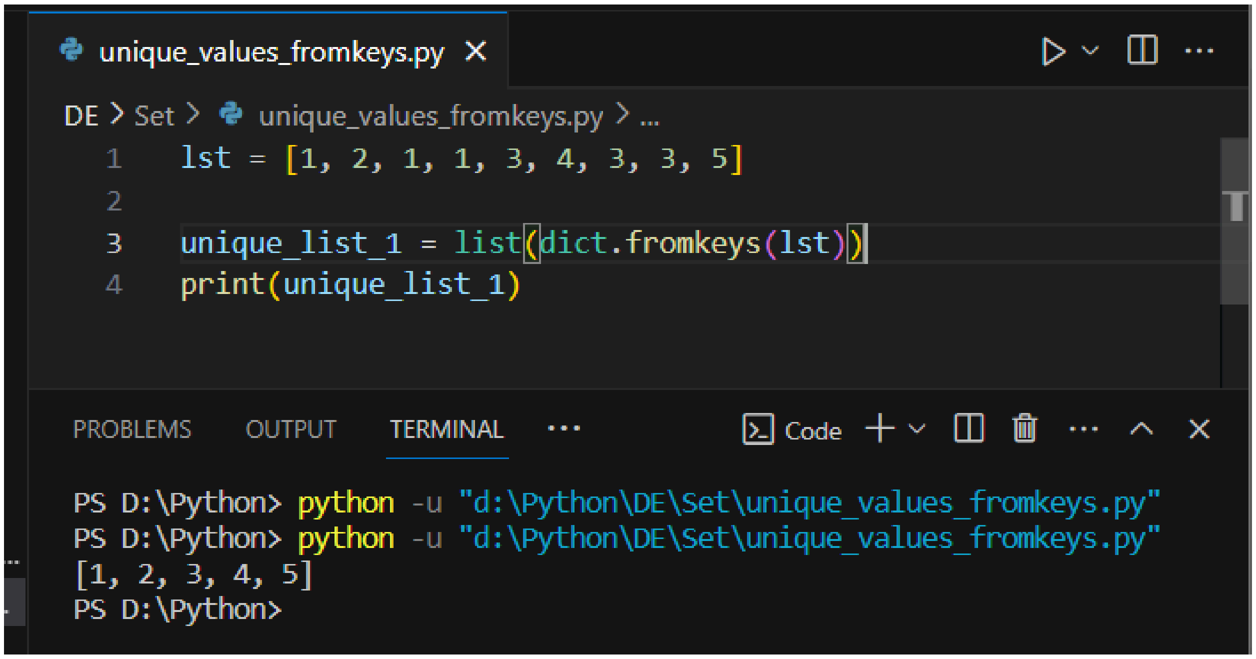
- **Get Unique Values From a List in Python Using collections.Counter() :** The Counter is a convenient and efficient way to count the occurrences of elements in a collection, typically a list. The * operator is used to retrieve unique values of list.


unique_values_counter.py
DE > Set > unique_values_counter.py > ...
Debug Console (Ctrl+Shift+Y)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code

```python
from collections import Counter
def unique(l):
 counter = Counter(l)
 print(*counter)

l1 = [10, 20, 10, 30, 40, 40]
print("the unique values from the list is")
unique(l1)
```

```
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_counter.py"
the unique values from 1st list is
10 20 30 40
PS D:\Python>
```

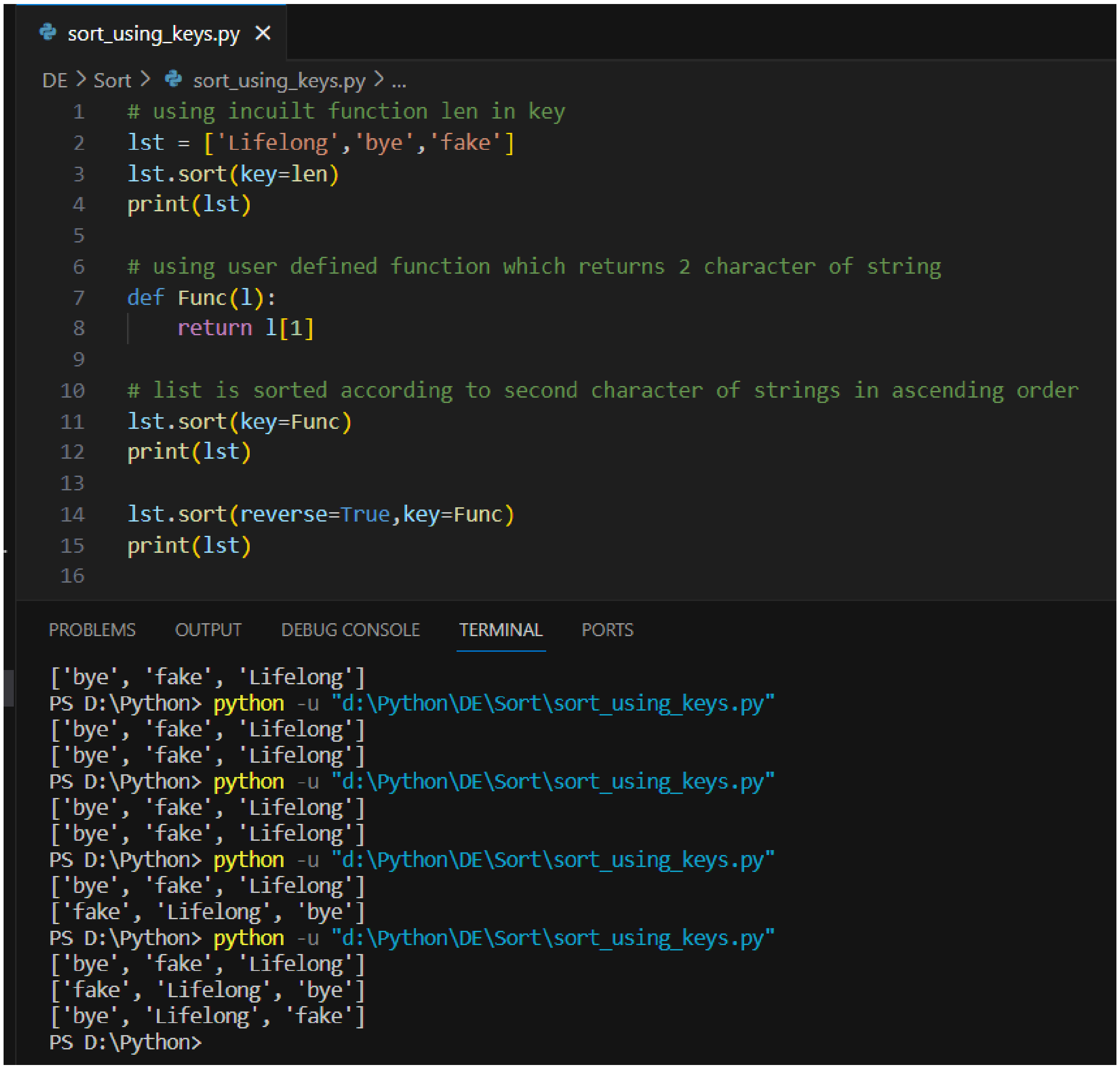
- **Get Unique Values From a List Using dict.fromkeys():**
  Here we are using dictionary properties of not having same keys by using the list as the argument in fromkeys() method which return a dictionary with unique keys and none value. Then again converting the dictionary into list gives unique list.


unique_values_fromkeys.py
DE > Set > unique_values_fromkeys.py > ...
PROBLEMS OUTPUT TERMINAL
Code

```python
lst = [1, 2, 1, 1, 3, 4, 3, 3, 5]

unique_list_1 = list(dict.fromkeys(lst))
print(unique_list_1)
```

```
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_fromkeys.py"
PS D:\Python> python -u "d:\Python\DE\Set\unique_values_fromkeys.py"
[1, 2, 3, 4, 5]
PS D:\Python>
```

- **Sort Python lists using key**
  The sort() method of lists has a attribute key which takes any specified function as argument to sort the list on the result of the specified function.


sort_using_keys.py
DE > Sort > sort_using_keys.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

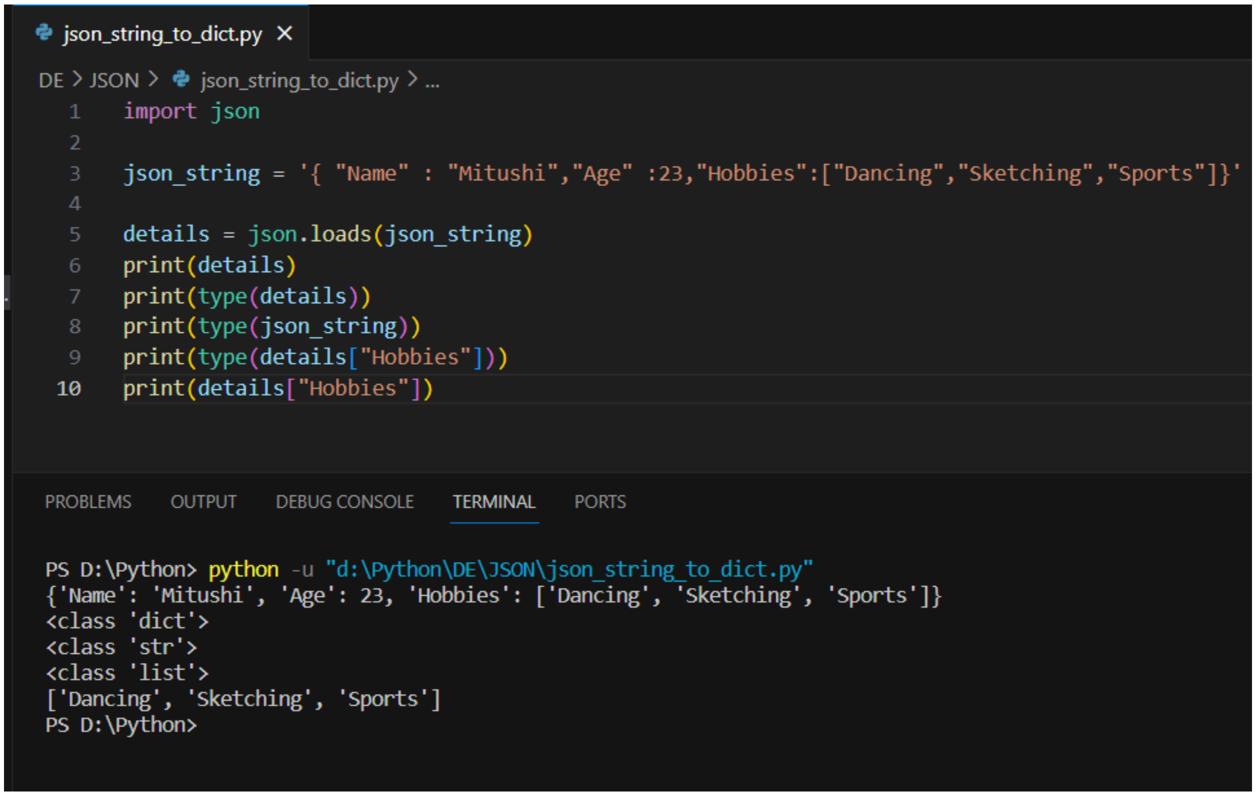
```python
# using incuilt function len in key
lst = ['Lifelong','bye','fake']
lst.sort(key=len)
print(lst)

# using user defined function which returns 2 character of string
def Func(l):
    return l[1]

# list is sorted according to second character of strings in ascending order
lst.sort(key=Func)
print(lst)

lst.sort(reverse=True,key=Func)
print(lst)

```

```
['bye', 'fake', 'Lifelong']
PS D:\Python> python -u "d:\Python\DE\Sort\sort_using_keys.py"
['bye', 'fake', 'Lifelong']
['bye', 'fake', 'Lifelong']
PS D:\Python> python -u "d:\Python\DE\Sort\sort_using_keys.py"
['bye', 'fake', 'Lifelong']
['bye', 'fake', 'Lifelong']
PS D:\Python> python -u "d:\Python\DE\Sort\sort_using_keys.py"
['bye', 'fake', 'Lifelong']
['fake', 'Lifelong', 'bye']
PS D:\Python> python -u "d:\Python\DE\Sort\sort_using_keys.py"
['bye', 'fake', 'Lifelong']
['fake', 'Lifelong', 'bye']
['bye', 'Lifelong', 'fake']
PS D:\Python>
```

Overview of JSON Strings and Files

- **Convert JSON String to Dictionary Python**

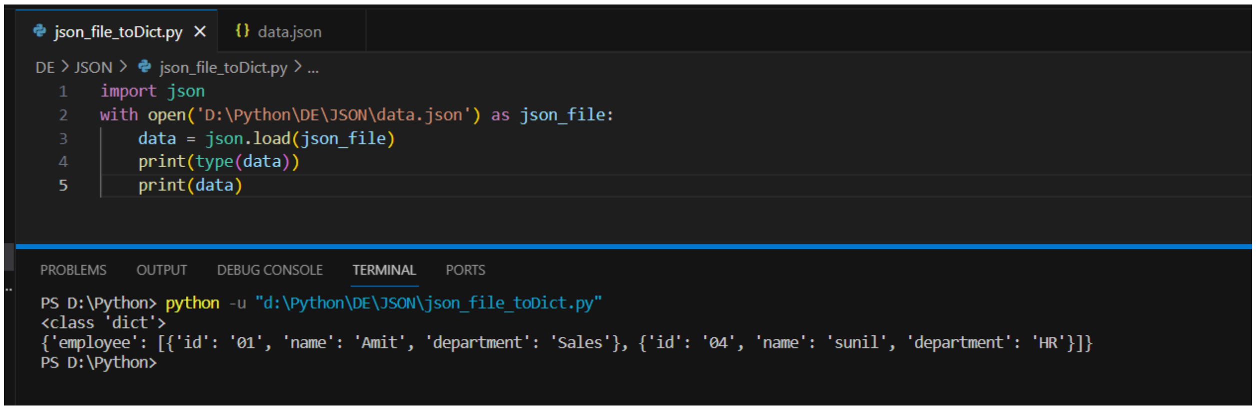
Using json inbuilt module in python and loads method to convert json string into dictionary.



```python
import json

json_string = '{ "Name" : "Mitushi","Age" :23,"Hobbies":["Dancing","Sketching","Sports"]}'

details = json.loads(json_string)
print(details)
print(type(details))
print(type(json_string))
print(type(details["Hobbies"]))
print(details["Hobbies"])
```

```
PS D:\Python> python -u "d:\Python\DE\JSON\json_string_to_dict.py"
{'Name': 'Mitushi', 'Age': 23, 'Hobbies': ['Dancing', 'Sketching', 'Sports']}
<class 'dict'>
<class 'str'>
<class 'list'>
['Dancing', 'Sketching', 'Sports']
PS D:\Python>
```

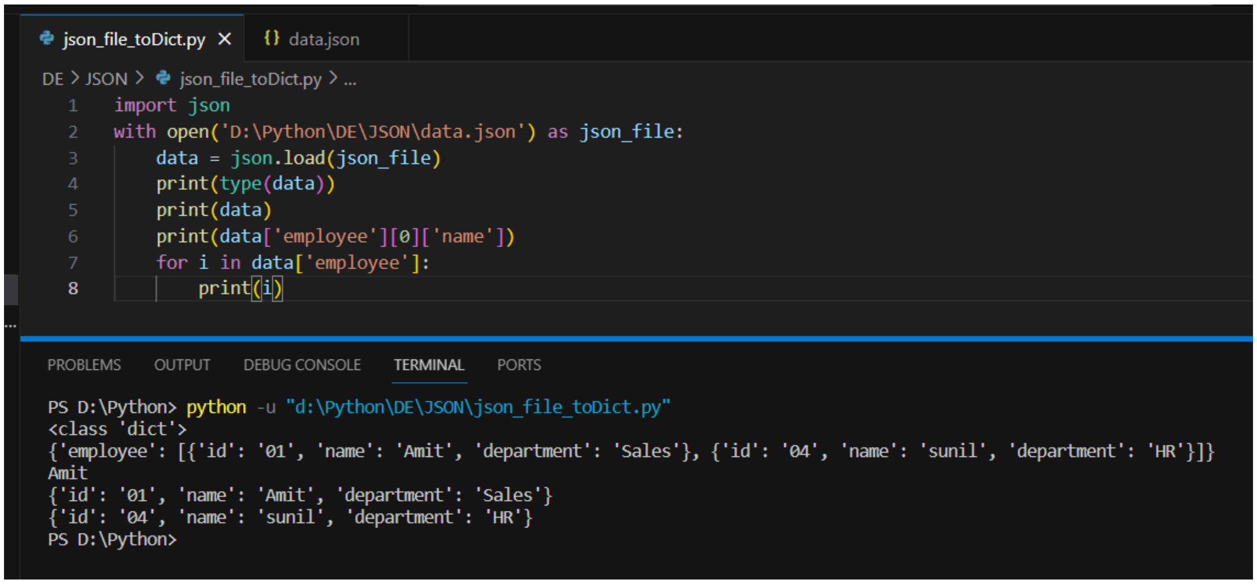
- **Convert JSON File to Python Object :** Json File data can be converted to python object i.e. dictionary using json.load() function passing the opened json file object in it.



```python
import json
with open('D:\Python\DE\JSON\data.json') as json_file:
    data = json.load(json_file)
    print(type(data))
    print(data)
```

```
PS D:\Python> python -u "d:\Python\DE\JSON\json_file_toDict.py"
<class 'dict'>
{'employee': [{'id': '01', 'name': 'Amit', 'department': 'Sales'}, {'id': '04', 'name': 'sunil', 'department': 'HR'}]}
PS D:\Python>
```

- ## Python read JSON file

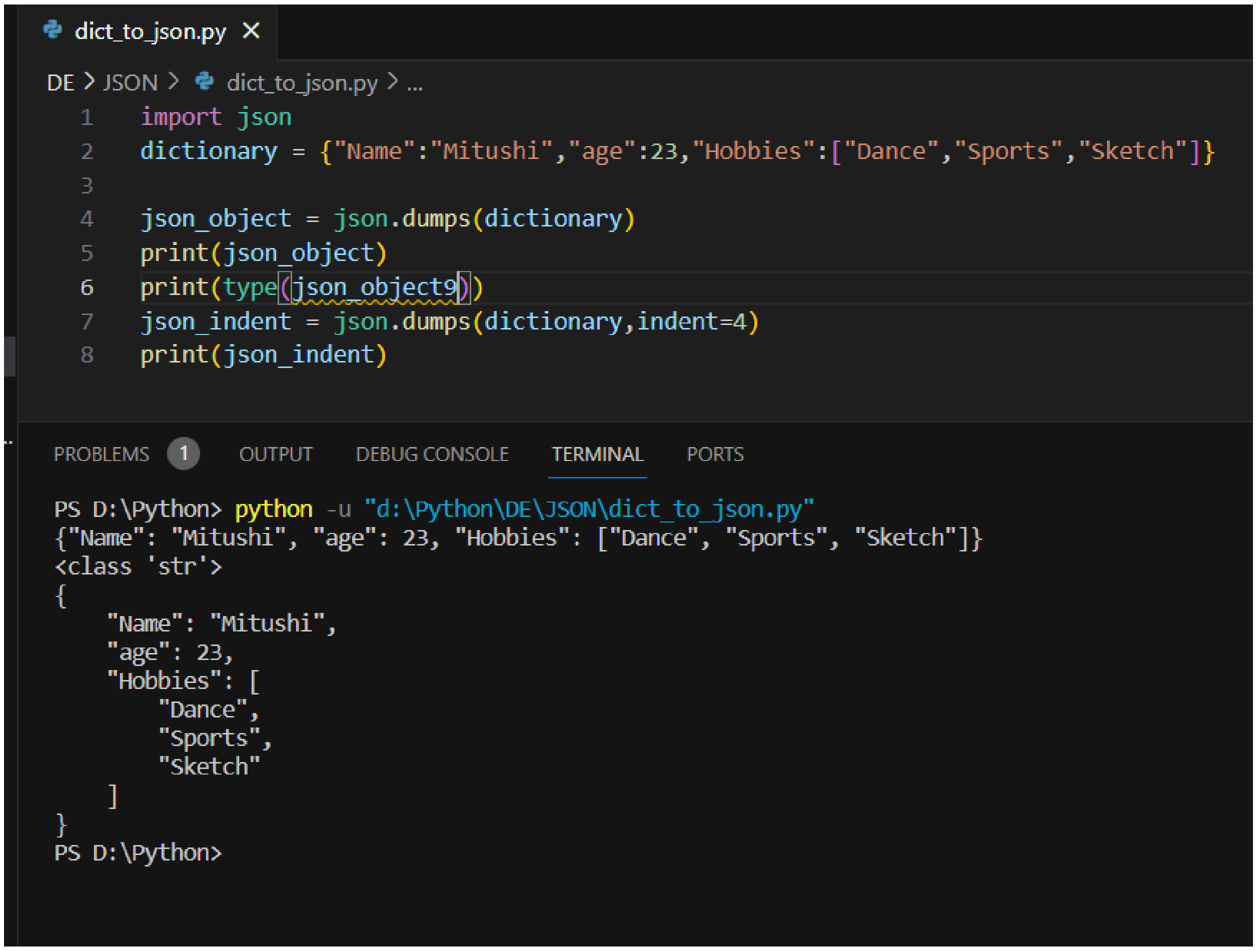


```python
import json
with open('D:\Python\DE\JSON\data.json') as json_file:
    data = json.load(json_file)
    print(type(data))
    print(data)
    print(data['employee'][0]['name'])
    for i in data['employee']:
        print(i)
```

```
PS D:\Python> python -u "d:\Python\DE\JSON\json_file_toDict.py"
<class 'dict'>
{'employee': [{'id': '01', 'name': 'Amit', 'department': 'Sales'}, {'id': '04', 'name': 'sunil', 'department': 'HR'}]}
Amit
{'id': '01', 'name': 'Amit', 'department': 'Sales'}
{'id': '04', 'name': 'sunil', 'department': 'HR'}
PS D:\Python>
```

- ## Convert Python Dict to JSON

  Converting python dictionary into json object using json.dumps(). The attribute indent is used to give required whitespaces before each data in output.



```python
import json
dictionary = {"Name":"Mitushi","age":23,"Hobbies":["Dance","Sports","Sketch"]}

json_object = json.dumps(dictionary)
print(json_object)
print(type(json_object9))
json_indent = json.dumps(dictionary,indent=4)
print(json_indent)
```

```
PS D:\Python> python -u "d:\Python\DE\JSON\dict_to_json.py"
{"Name": "Mitushi", "age": 23, "Hobbies": ["Dance", "Sports", "Sketch"]}
<class 'str'>
{
    "Name": "Mitushi",
    "age": 23,
    "Hobbies": [
        "Dance",
        "Sports",
        "Sketch"
    ]
}
PS D:\Python>
```