



০০১-০০৮ পাইথনের কিছু বেসিক কোড (পর্ব-২)

Statistics কে কিছুটা বিরতি রেখে আমরা এখন পাইথন ধরছি, পরে আবার আমরা *Statistics continue* করবো।

Python-এ ডিকশনারী (Dictionaries)

ডিকশনারী হল key-value pair ভিত্তিক ডাটা স্ট্রাকচার, যেখানে কী (key) হতে হবে অনন্য (unique) এবং অপরিবর্তনীয় (immutable), কিন্তু মান (value) যেকোনো ধরনের হতে পারে।

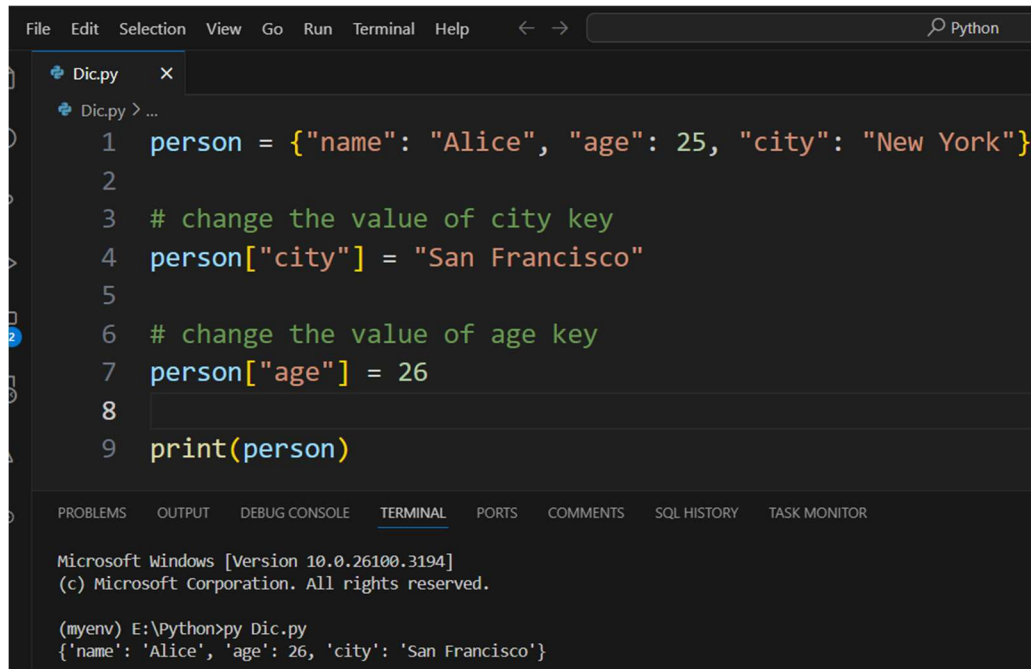
```
File Edit Selection View Go Run Terminal Help Python
Dic.py
1 person = {"name": "Alice", "age": 25, "city": "New York"}
2 print(person)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS SQL HISTORY TASK MONITOR
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Dic.py
{'name': 'Alice', 'age': 25, 'city': 'New York'}
```

এখানে name, age, city হলো keys। আর Alice, 25, New York হলো values।

১) ডিকশনারীর মান পরিবর্তন করাঃ



The screenshot shows a Python IDE with a file named 'Dic.py'. The code in the editor is as follows:

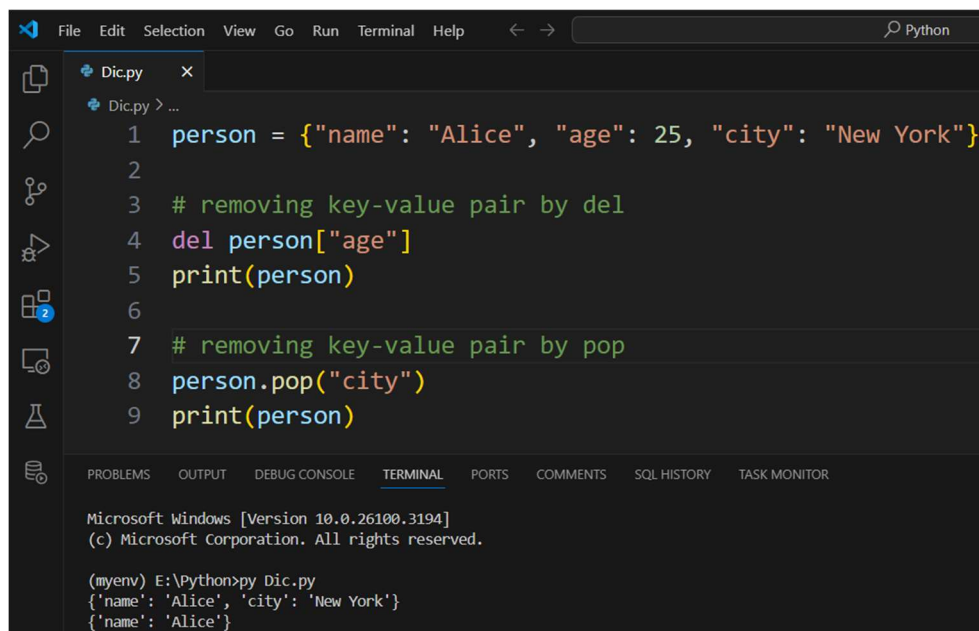
```
1 person = {"name": "Alice", "age": 25, "city": "New York"}
2
3 # change the value of city key
4 person["city"] = "San Francisco"
5
6 # change the value of age key
7 person["age"] = 26
8
9 print(person)
```

The terminal output at the bottom shows the execution of the script:

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Dic.py
{'name': 'Alice', 'age': 26, 'city': 'San Francisco'}
```

২) ডিকশনারী থেকে key-value pair সরানোঃ



The screenshot shows a Python IDE with a file named 'Dic.py'. The code in the editor is as follows:

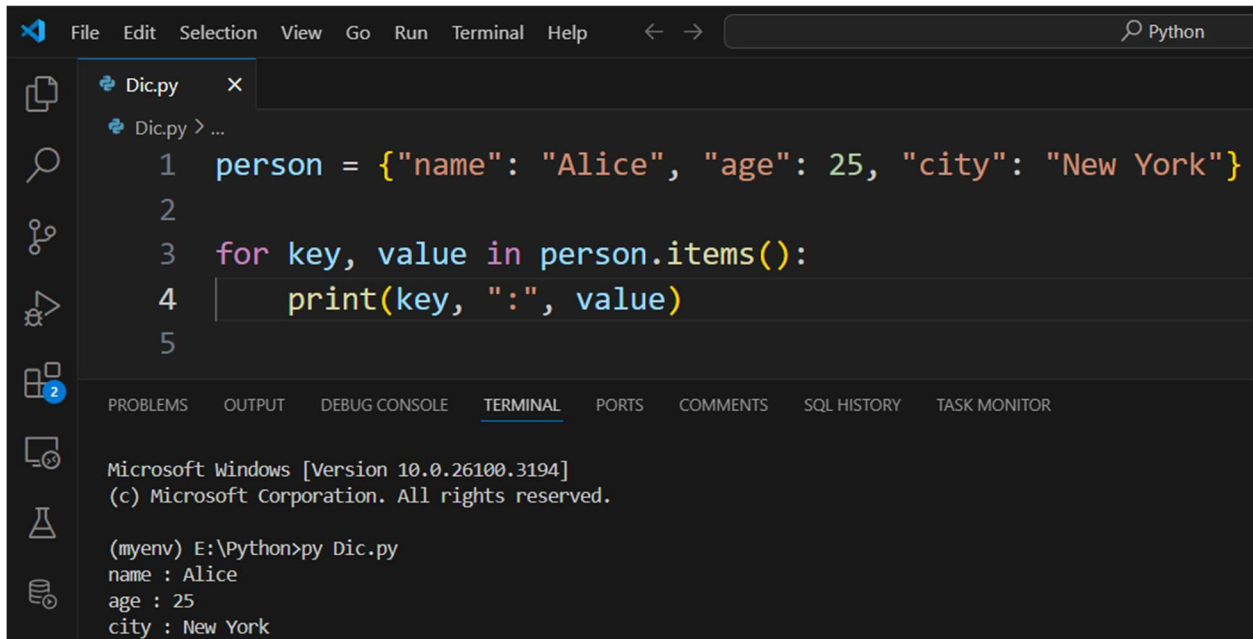
```
1 person = {"name": "Alice", "age": 25, "city": "New York"}
2
3 # removing key-value pair by del
4 del person["age"]
5 print(person)
6
7 # removing key-value pair by pop
8 person.pop("city")
9 print(person)
```

The terminal output at the bottom shows the execution of the script:

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Dic.py
{'name': 'Alice', 'city': 'New York'}
{'name': 'Alice'}
```

৩) ডিকশনারীর উপর লুপ চালানোঃ



The screenshot shows the Visual Studio Code editor with a file named `Dic.py`. The code defines a dictionary `person` with keys `name`, `age`, and `city`. A `for` loop iterates over `person.items()`, printing each key-value pair. The output in the terminal is:

```
person = {"name": "Alice", "age": 25, "city": "New York"}
for key, value in person.items():
    print(key, ":", value)
```

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Dic.py
name : Alice
age : 25
city : New York
```

৪) ডিকশনারী তে key বিদ্যমান আছে কি না চেক করাঃ



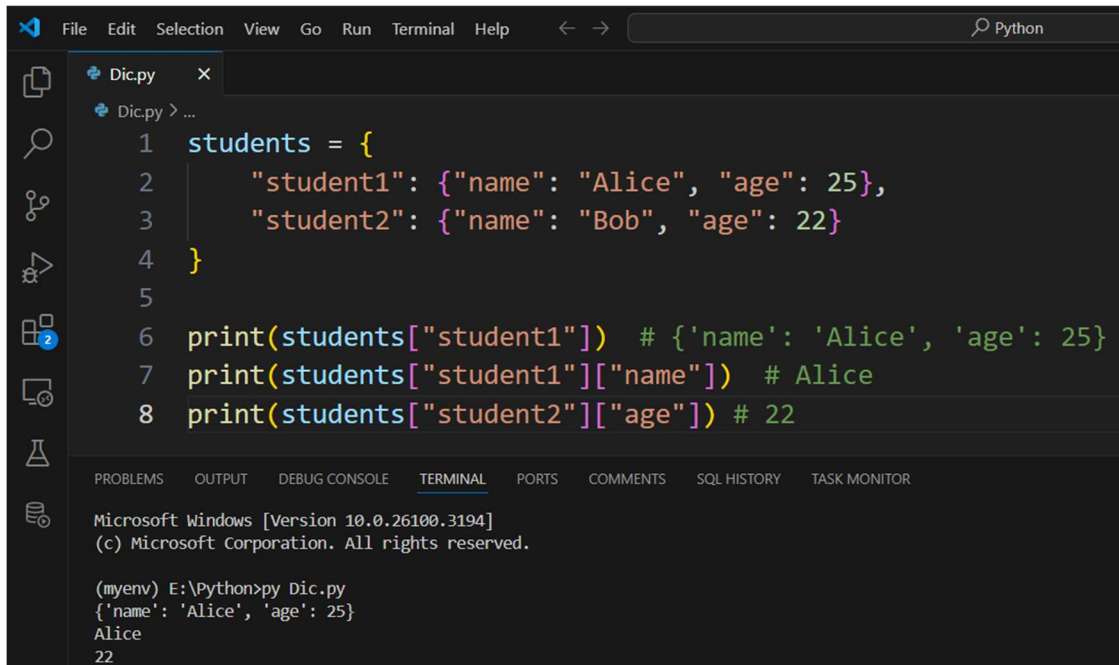
The screenshot shows the Visual Studio Code editor with a file named `Dic.py`. The code defines a dictionary `person` with keys `name`, `age`, and `city`. An `if` statement checks if the key `"name"` is present in the dictionary. If it is, it prints `"Name is present."`; otherwise, it prints `"Name is not present."`. The output in the terminal is:

```
person = {"name": "Alice", "age": 25, "city": "New York"}
if "name" in person:
    print("Name is present.")
else:
    print("Name is not present.")
```

```
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Dic.py
Name is present.
```

৫) নেস্টেড ডিকশনারী (Nested Dictionary)



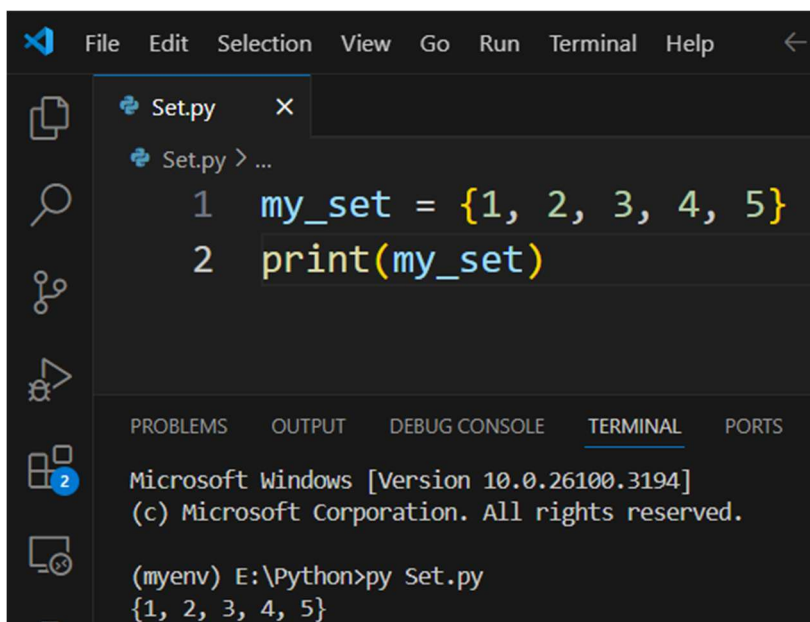
The screenshot shows a Python IDE with a file named 'Dic.py'. The code defines a dictionary 'students' with two keys: 'student1' and 'student2'. Each key maps to another dictionary containing 'name' and 'age' attributes. The code then prints the nested dictionaries and specific values using indexing.

```
1 students = {  
2     "student1": {"name": "Alice", "age": 25},  
3     "student2": {"name": "Bob", "age": 22}  
4 }  
5  
6 print(students["student1"]) # {'name': 'Alice', 'age': 25}  
7 print(students["student1"]["name"]) # Alice  
8 print(students["student2"]["age"]) # 22
```

The terminal output shows the execution of the script, displaying the nested dictionary for 'student1' and the individual values 'Alice' and '22'.

সেট (Set)

সেট হল অনন্য এবং অপরিবর্তনীয় elements এর একটি সংগ্রহ।



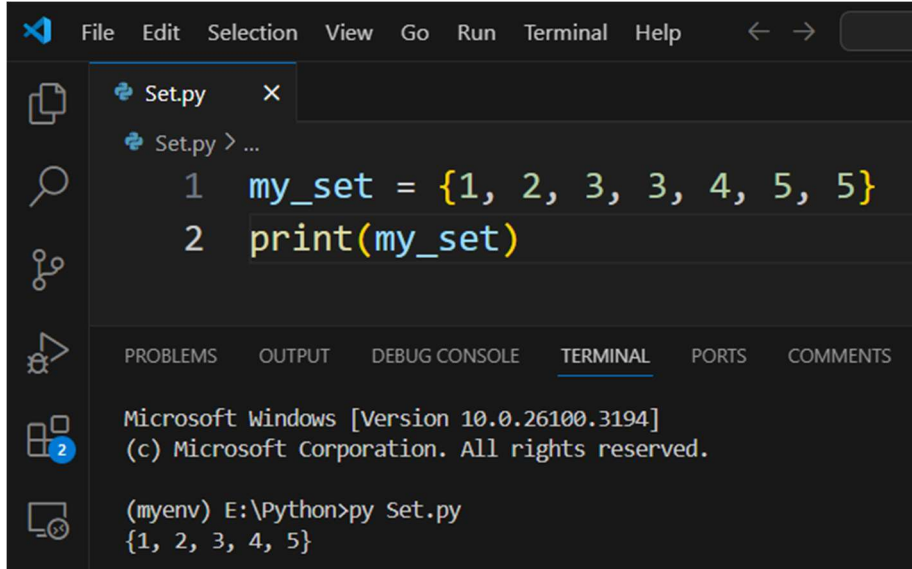
The screenshot shows a Python IDE with a file named 'Set.py'. The code creates a set 'my_set' containing the numbers 1, 2, 3, 4, and 5, and then prints it.

```
1 my_set = {1, 2, 3, 4, 5}  
2 print(my_set)
```

The terminal output shows the execution of the script, displaying the set {1, 2, 3, 4, 5}.

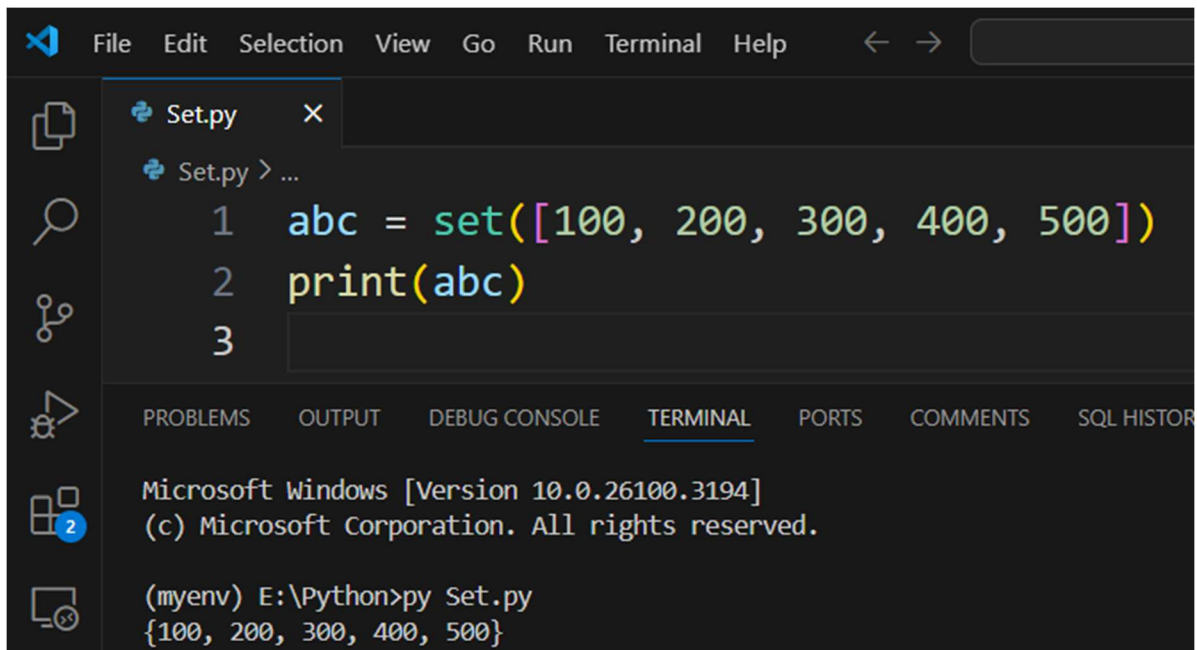
সেট এ কখনই duplicate element যোগ হয় না।

যেমনঃ



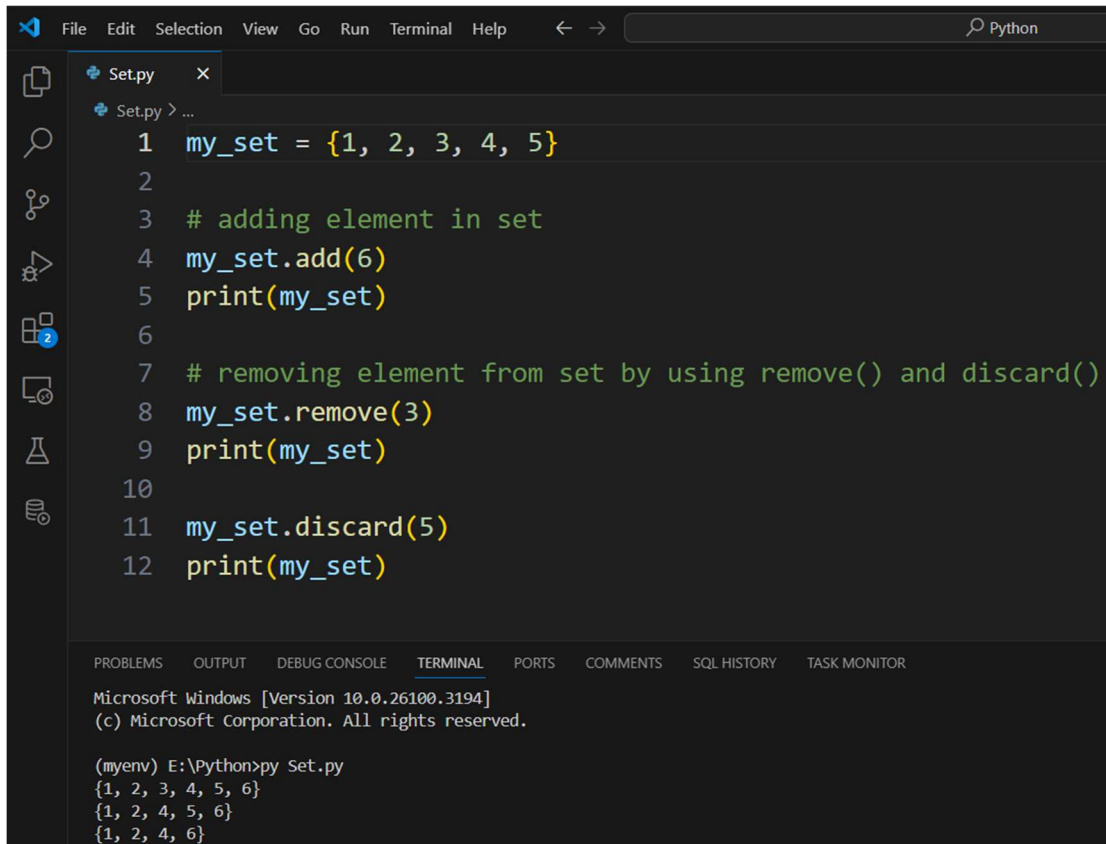
```
File Edit Selection View Go Run Terminal Help  
Set.py  
Set.py > ...  
1 my_set = {1, 2, 3, 3, 4, 5, 5}  
2 print(my_set)  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS  
Microsoft Windows [Version 10.0.26100.3194]  
(c) Microsoft Corporation. All rights reserved.  
(myenv) E:\Python>py Set.py  
{1, 2, 3, 4, 5}
```

s) set() ফাংশন ব্যবহার করে সেট তৈরি করাঃ



```
File Edit Selection View Go Run Terminal Help  
Set.py  
Set.py > ...  
1 abc = set([100, 200, 300, 400, 500])  
2 print(abc)  
3  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS SQL HISTOR  
Microsoft Windows [Version 10.0.26100.3194]  
(c) Microsoft Corporation. All rights reserved.  
(myenv) E:\Python>py Set.py  
{100, 200, 300, 400, 500}
```

২) সেটে উপাদান যোগ করা ও সরানোঃ



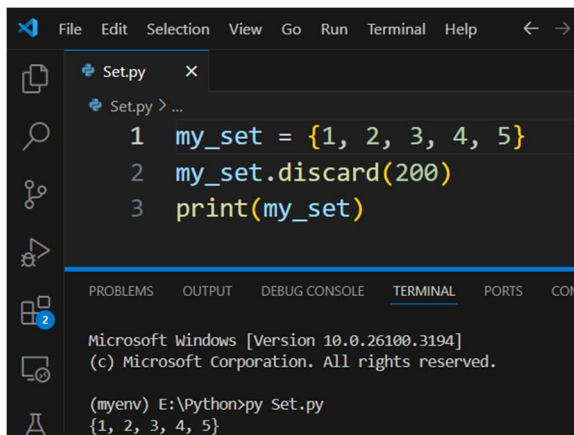
```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2
3 # adding element in set
4 my_set.add(6)
5 print(my_set)
6
7 # removing element from set by using remove() and discard()
8 my_set.remove(3)
9 print(my_set)
10
11 my_set.discard(5)
12 print(my_set)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COMMENTS SQL HISTORY TASK MONITOR

Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Set.py
{1, 2, 3, 4, 5, 6}
{1, 2, 4, 5, 6}
{1, 2, 4, 6}

কিন্তু যদি কোনো element সেই set এ না থাকে, তাহলে `remove()` আর `discard()` কিছুই করতে পারবে না। যেমনঃ



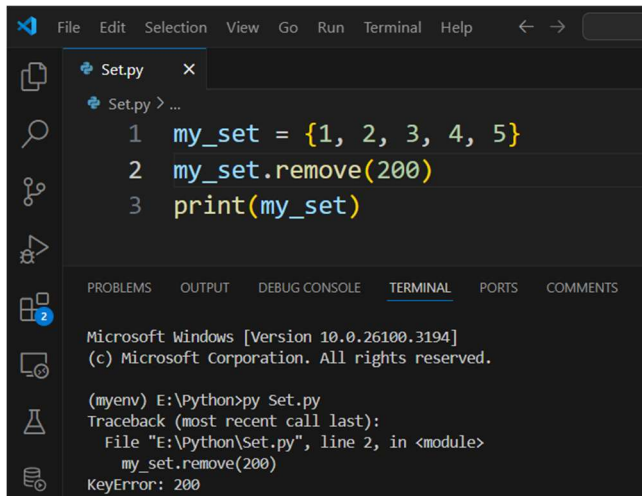
```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2 my_set.discard(200)
3 print(my_set)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS COM

Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Set.py
{1, 2, 3, 4, 5}

কিন্তু remove() এর ক্ষেত্রে error দেখাবে।



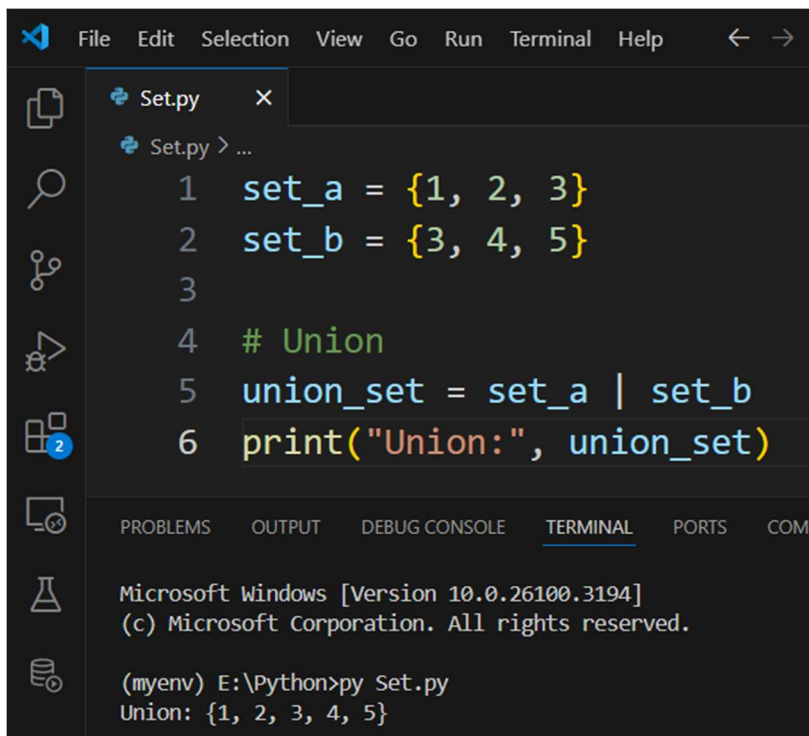
```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2 my_set.remove(200)
3 print(my_set)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Set.py
Traceback (most recent call last):
  File "E:\Python\Set.py", line 2, in <module>
    my_set.remove(200)
KeyError: 200
```

৩) সেট অপারেশন (Union, Intersection, Difference)

ইউনিয়ন

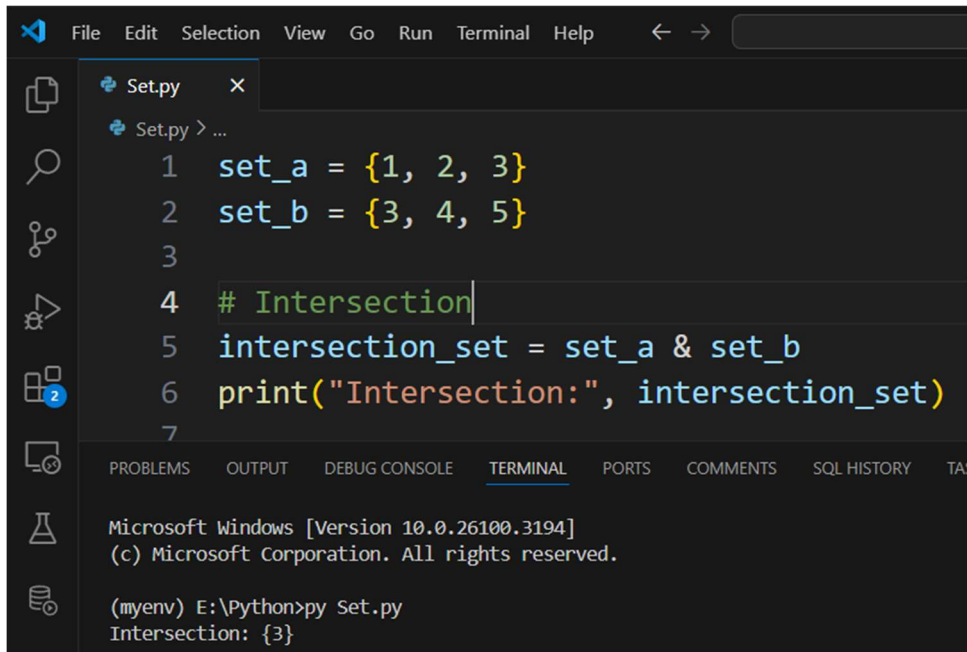


```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 set_a = {1, 2, 3}
2 set_b = {3, 4, 5}
3
4 # Union
5 union_set = set_a | set_b
6 print("Union:", union_set)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COM
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Set.py
Union: {1, 2, 3, 4, 5}
```


ইন্টারসেকশন



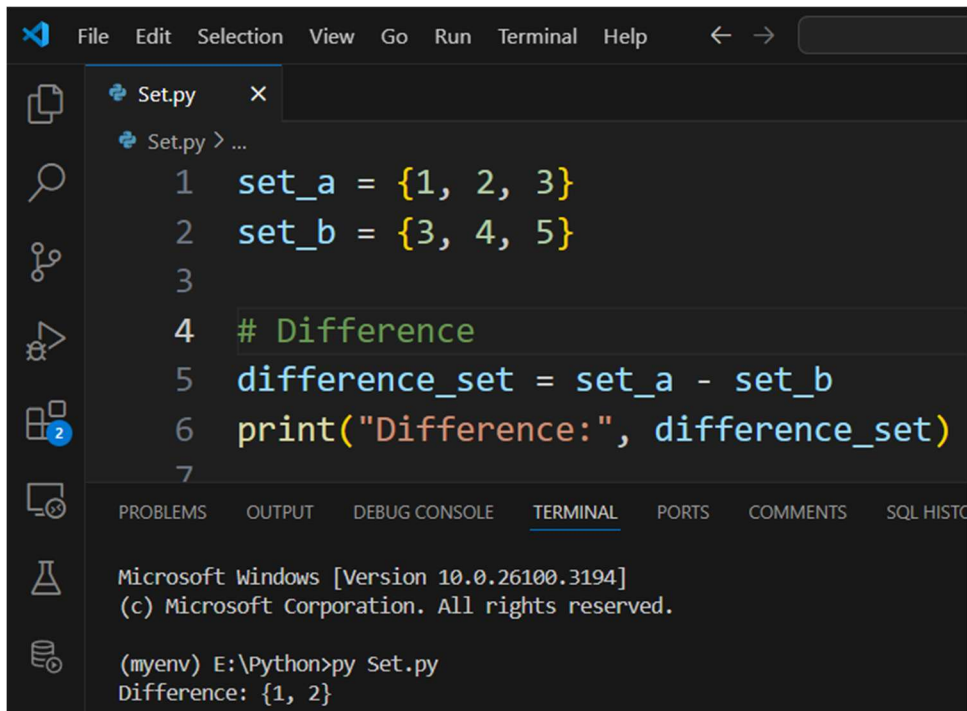
The screenshot shows the Visual Studio Code editor with a file named 'Set.py'. The code defines two sets, set_a and set_b, and calculates their intersection. The terminal output shows the result of the intersection.

```
File Edit Selection View Go Run Terminal Help
Set.py
Set.py > ...
1 set_a = {1, 2, 3}
2 set_b = {3, 4, 5}
3
4 # Intersection
5 intersection_set = set_a & set_b
6 print("Intersection:", intersection_set)
7
```

Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Set.py
Intersection: {3}

ডিফারেন্স



The screenshot shows the Visual Studio Code editor with a file named 'Set.py'. The code defines two sets, set_a and set_b, and calculates their difference. The terminal output shows the result of the difference.

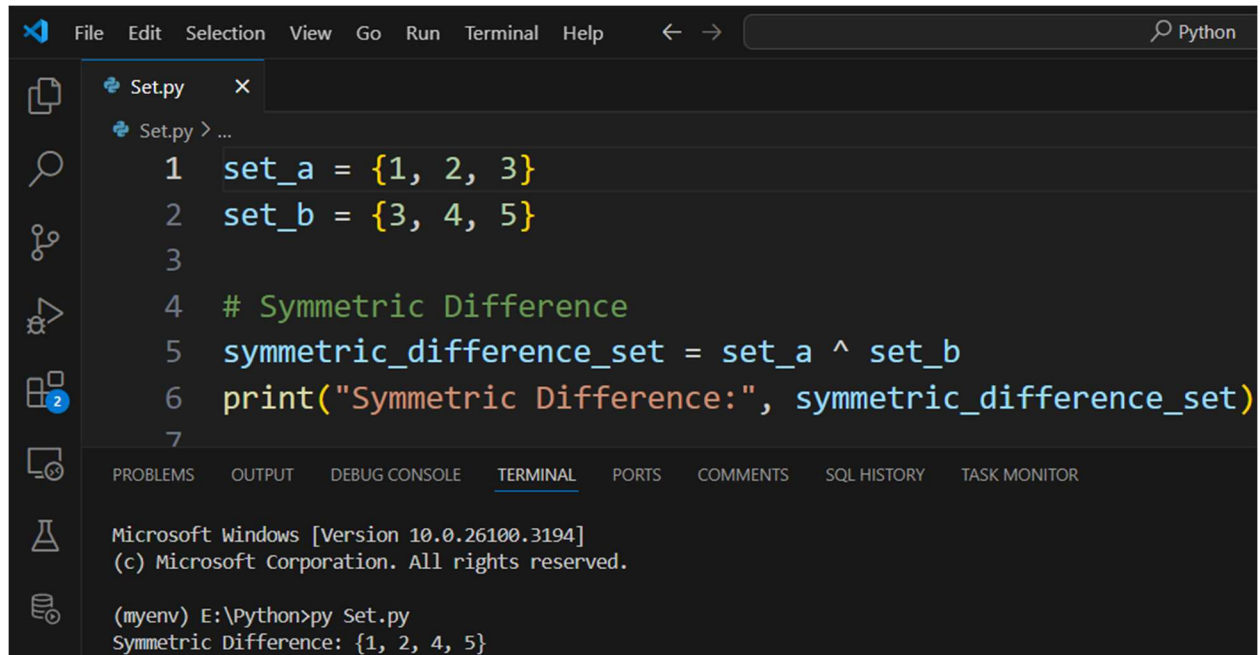
```
File Edit Selection View Go Run Terminal Help
Set.py
Set.py > ...
1 set_a = {1, 2, 3}
2 set_b = {3, 4, 5}
3
4 # Difference
5 difference_set = set_a - set_b
6 print("Difference:", difference_set)
7
```

Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Set.py
Difference: {1, 2}

সিমেট্রিক ডিফারেন্স এর ক্ষেত্রে দুটা সেটের common value রিমুভ করে ইউনিয়ন করে।

অর্থাৎ সিমেট্রিক ডিফারেন্স = ইউনিয়ন – ইন্টারসেকশন



The screenshot shows a Python IDE with a file named 'Set.py'. The code defines two sets, 'set_a' and 'set_b', and calculates their symmetric difference using the XOR operator (^). The result is printed as 'Symmetric Difference: {1, 2, 4, 5}'. The terminal output shows the command 'python Set.py' and the same result.

```
File Edit Selection View Go Run Terminal Help  
Set.py  
Set.py > ...  
1 set_a = {1, 2, 3}  
2 set_b = {3, 4, 5}  
3  
4 # Symmetric Difference  
5 symmetric_difference_set = set_a ^ set_b  
6 print("Symmetric Difference:", symmetric_difference_set)  
7  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS SQL HISTORY TASK MONITOR  
Microsoft Windows [Version 10.0.26100.3194]  
(c) Microsoft Corporation. All rights reserved.  
(myenv) E:\Python>py Set.py  
Symmetric Difference: {1, 2, 4, 5}
```