

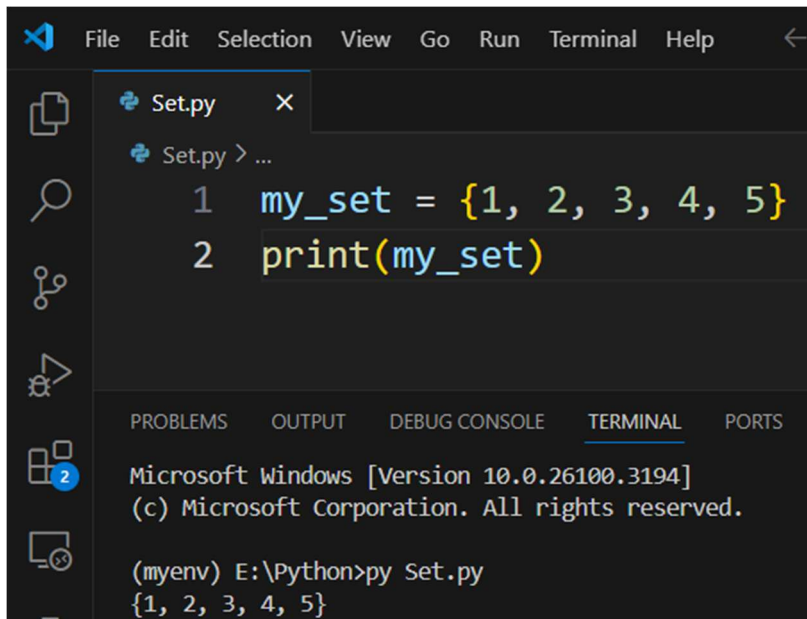


## ০০১-০০৬ পাইথনের কিছু বেসিক কোড (পর্ব-২)

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

### সেট (Set)

সেট হল অনন্য এবং অপরিবর্তনীয় elements এর একটি সংগ্রহ।



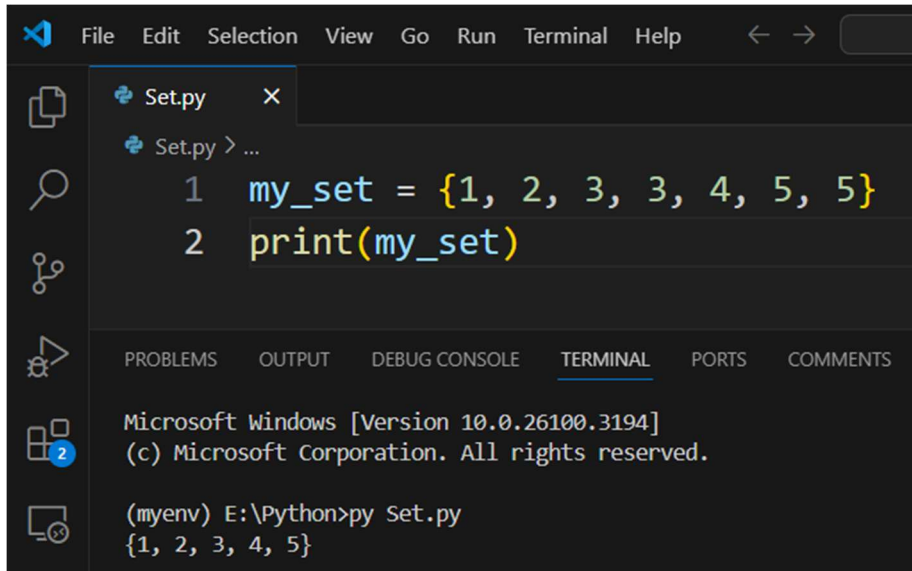
```
File Edit Selection View Go Run Terminal Help
Set.py
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2 print(my_set)

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

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

সেট এ কখনই duplicate element যোগ হয় না।

যেমনঃ {1, 2, 3, 3, 4, 5, 5} কে প্রিন্ট করলে দেখাবে {1, 2, 3, 4, 5}



The screenshot shows the Visual Studio Code editor with a file named `Set.py` open. The code in the editor is:

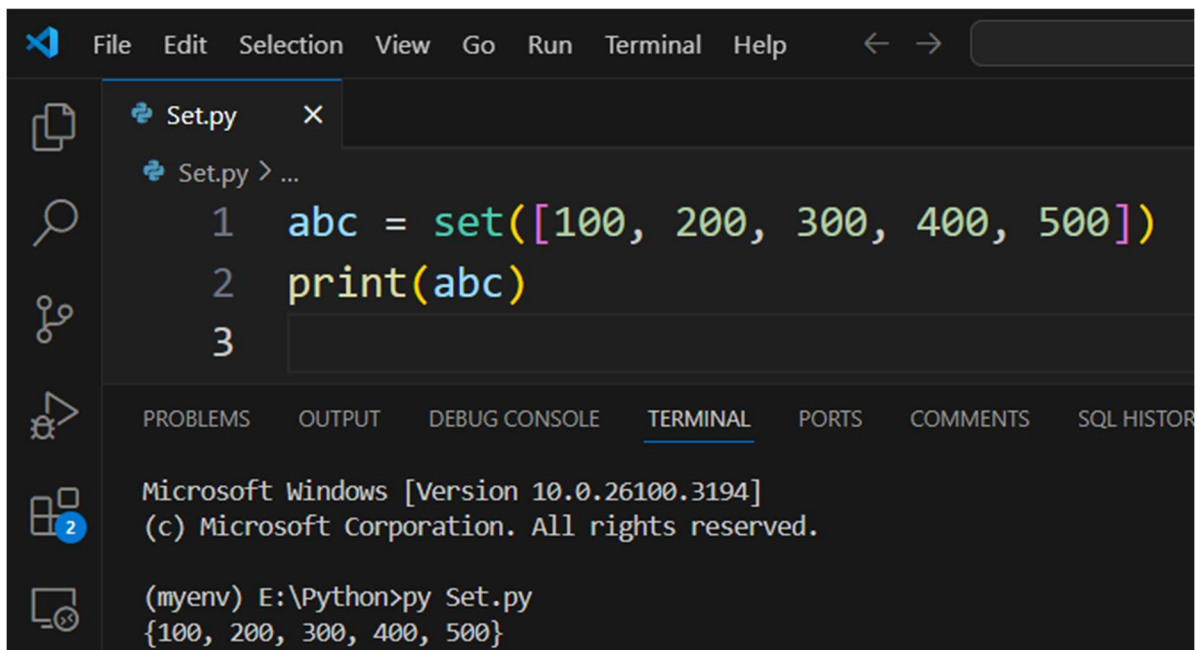
```
1 my_set = {1, 2, 3, 3, 4, 5, 5}
2 print(my_set)
```

Below the editor, the **TERMINAL** tab is active, showing the output of running the script:

```
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



The screenshot shows the Visual Studio Code editor with a file named `Set.py` open. The code in the editor is:

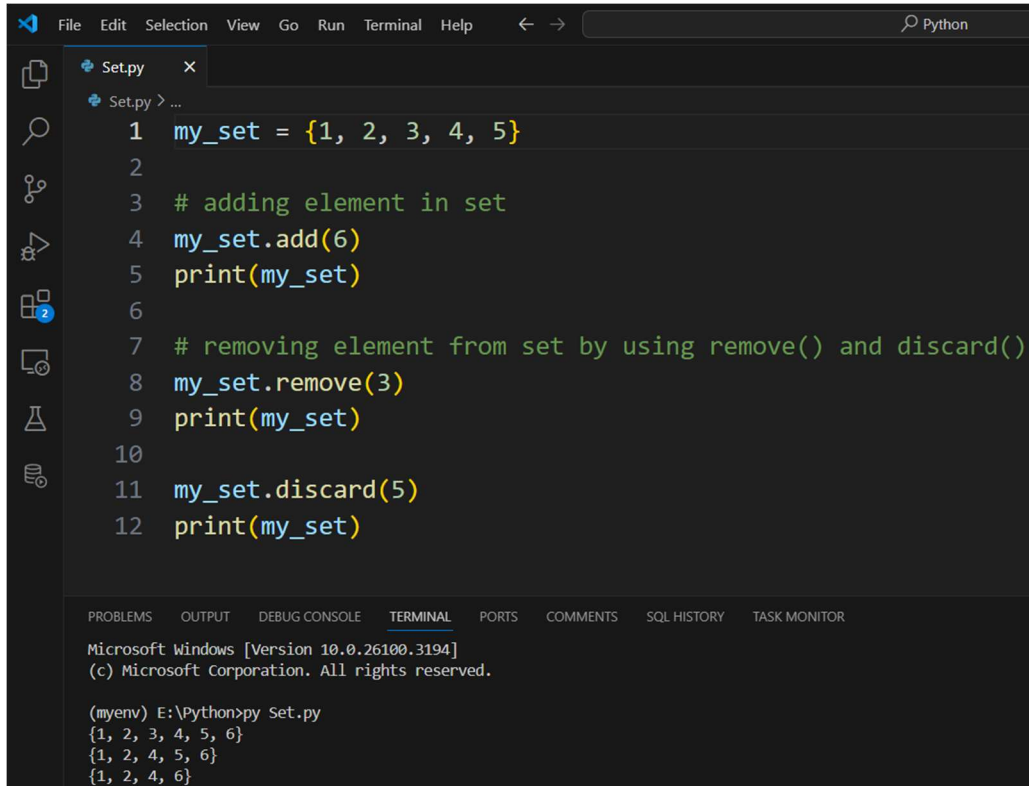
```
1 abc = set([100, 200, 300, 400, 500])
2 print(abc)
3
```

Below the editor, the **TERMINAL** tab is active, showing the output of running the script:

```
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 element যোগ করা ও সরানো: add() দিয়ে সেটে element যোগ করা হয়, আর remove() আর discard() দিয়ে element সরানো হয়।

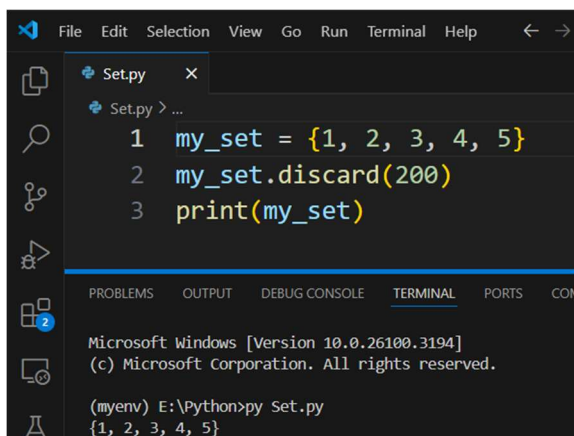


```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2
3 # adding element in set
4 my_set.add(6)
5 print(my_set)
6
7 # removing element from set by using remove() and discard()
8 my_set.remove(3)
9 print(my_set)
10
11 my_set.discard(5)
12 print(my_set)

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

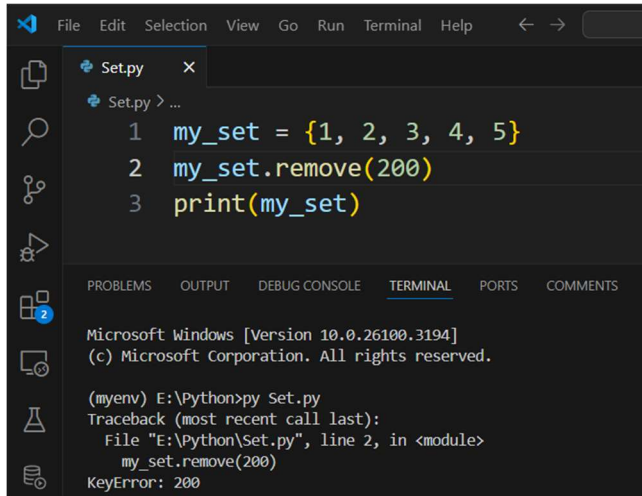


```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2 my_set.discard(200)
3 print(my_set)

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।



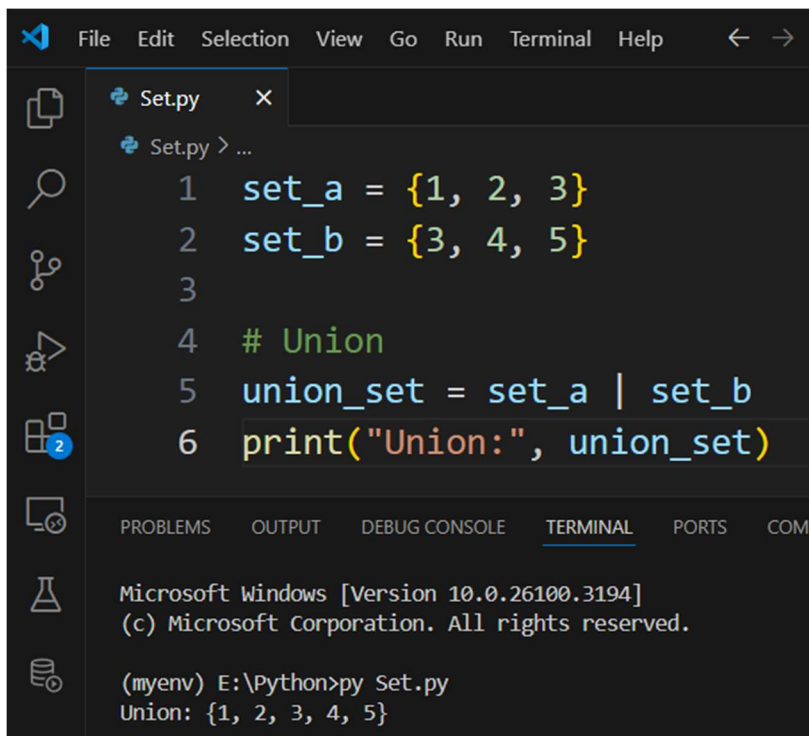
```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 my_set = {1, 2, 3, 4, 5}
2 my_set.remove(200)
3 print(my_set)

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

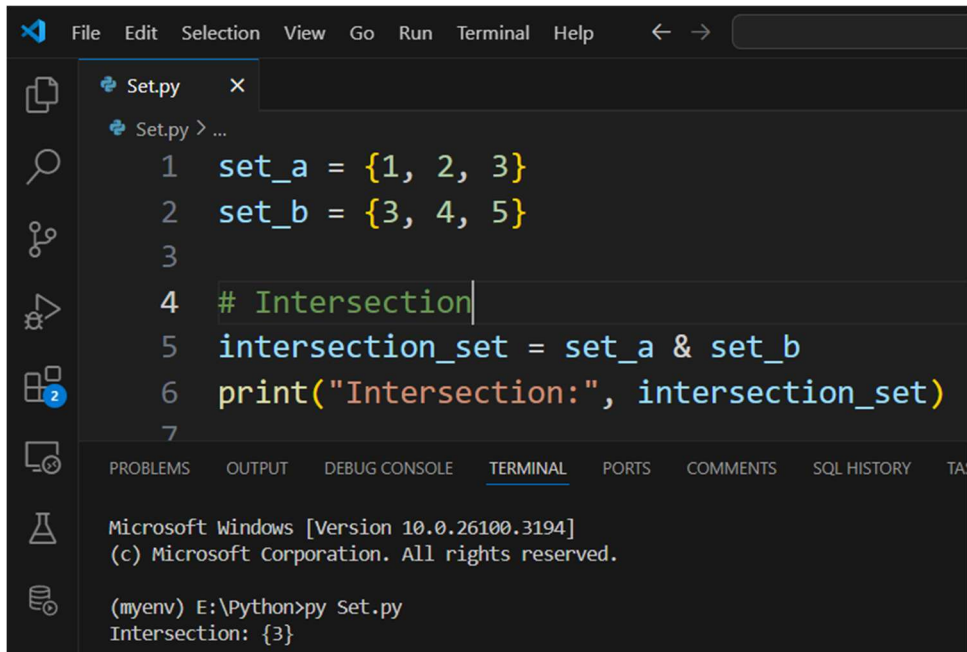


```
File Edit Selection View Go Run Terminal Help
Set.py x
Set.py > ...
1 set_a = {1, 2, 3}
2 set_b = {3, 4, 5}
3
4 # Union
5 union_set = set_a | set_b
6 print("Union:", union_set)

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



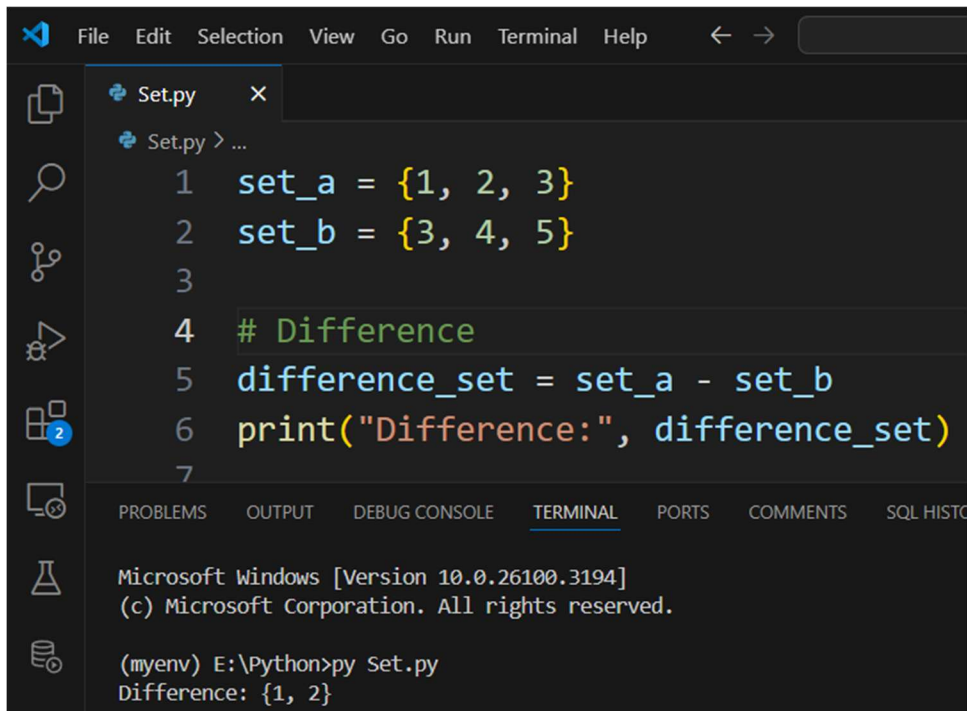
The screenshot shows the Visual Studio Code editor with a file named 'Set.py' open. The code defines two sets, 'set\_a' and 'set\_b', and calculates their intersection. The terminal output shows the result of the intersection.

```
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' open. The code defines two sets, 'set\_a' and 'set\_b', and calculates their difference. The terminal output shows the result of the difference.

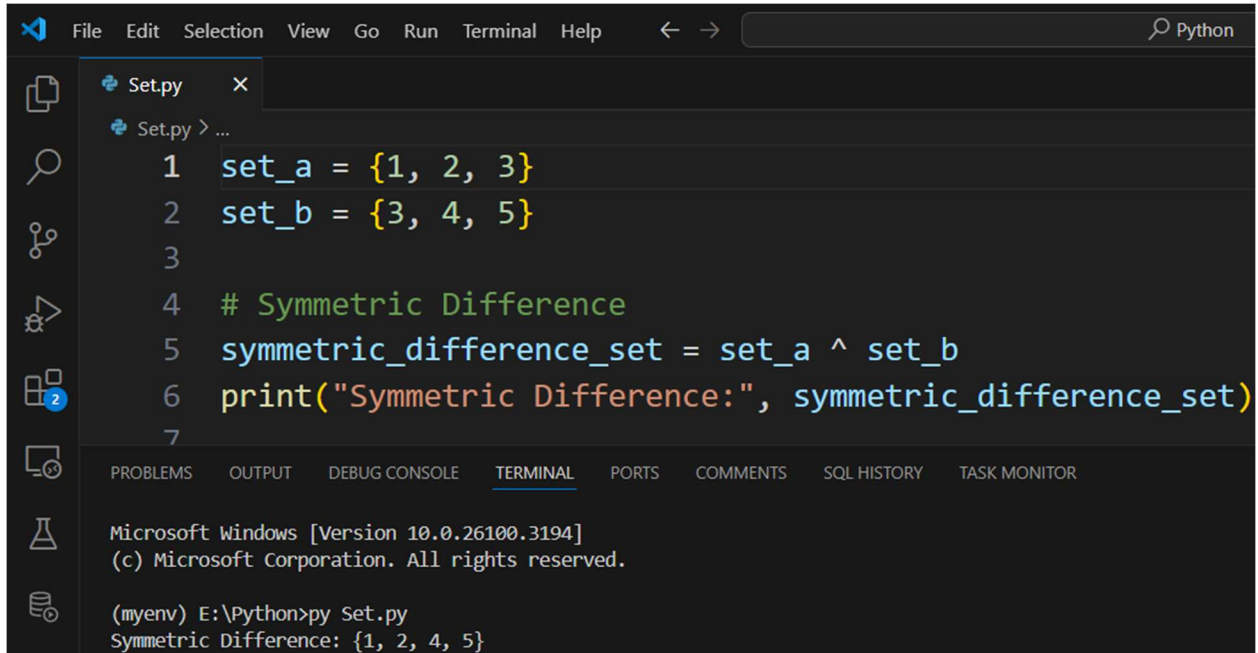
```
1 set_a = {1, 2, 3}
2 set_b = {3, 4, 5}
3
4 # Difference
5 difference_set = set_a - set_b
6 print("Difference:", difference_set)
7
```

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

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

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

অর্থাৎ সিমেন্ট্রিক 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 output is printed as 'Symmetric Difference: {1, 2, 4, 5}'. The terminal window at the bottom shows the command 'python Set.py' and the resulting output.

```
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
```

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}

## কন্ডিশনাল স্টেটমেন্টঃ

অন্যান্য প্রোগ্রামিং ভাষার মত পাইথনেরও কন্ডিশনাল স্টেটমেন্ট আছে। এখানে else if কে elif হিসেবে লেখা হয়।

উদাহরণ হিসেবে দেখানো হল, একটা ভেরিয়েবল num দেওয়া আছে, তাকে conditional statement এ রাখা হল। num যদি ধনাত্মক সংখ্যা হয়, তাহলে দেখাবে Positive number। আর যদি num শূন্য হয়, তাহলে দেখাবে Zero। আর num যদি ঋণাত্মক সংখ্যা হয়, তাহলে দেখাবে negative number।

নিচে ৩ টা কোড দেখানো হল যাদের num এর মান ধনাত্মক সংখ্যা, শূন্য আর ঋণাত্মক সংখ্যা।

```
File Edit Selection View Go Run Terminal Help
Condition.py X
Condition.py > ...
1 num = 10
2 if num > 0:
3     print('Positive number')
4 elif num == 0:
5     print('Zero')
6 else:
7     print('Negative number')
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMM
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Condition.py
Positive number
```

```
File Edit Selection View Go Run Terminal Help
Condition.py X
Condition.py > ...
1 num = 0
2 if num > 0:
3     print('Positive number')
4 elif num == 0:
5     print('Zero')
6 else:
7     print('Negative number')
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMM
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Condition.py
Zero
```

```
File Edit Selection View Go Run Terminal Help
Condition.py X
Condition.py > ...
1 num = -30
2 if num > 0:
3     print('Positive number')
4 elif num == 0:
5     print('Zero')
6 else:
7     print('Negative number')
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMM
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

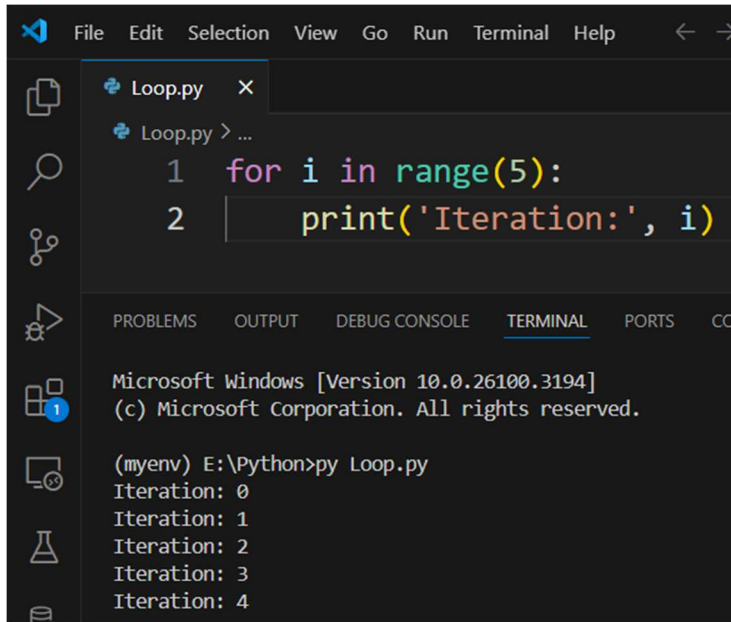
(myenv) E:\Python>py Condition.py
Negative number
```

## ফর লুপঃ

ফর লুপের ক্ষেত্রে রেঞ্জ ব্যবহার করতে হয় (যদি আমরা সংখ্যা নিয়ে কাজ করি)। `range(n)` এর ক্ষেত্রে শূন্য থেকে  $n-1$  পর্যন্ত লুপ চলে।

নিচে এর উদাহরণ দেওয়া হলো





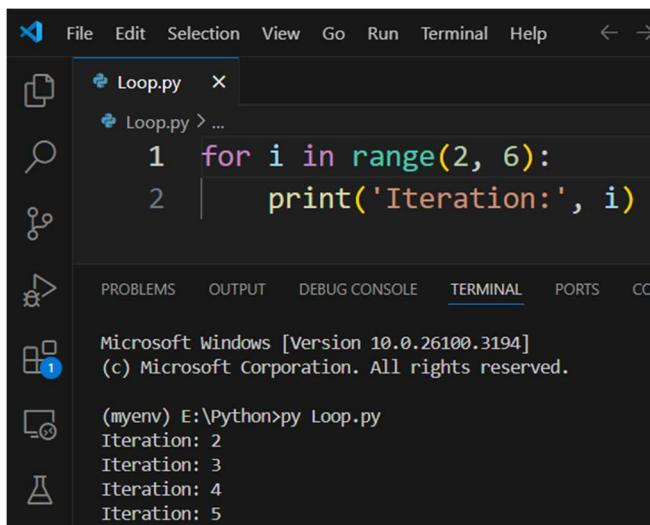
```
File Edit Selection View Go Run Terminal Help
Loop.py x
Loop.py > ...
1 for i in range(5):
2     print('Iteration:', i)

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

(myenv) E:\Python>py Loop.py
Iteration: 0
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
```

এখানে `range(5)` এর মানে 0 to 4। অর্থাৎ `range(n)` মানে 0 to  $n-1$ .

যদি আমরা 0 থেকে স্টার্ট না করতে চাই, তখন `range(s,n)` ব্যবহার করতে হবে। `range(s,n)` হবে  $s$  থেকে  $n-1$  পর্যন্ত। যেমনঃ



```
File Edit Selection View Go Run Terminal Help
Loop.py x
Loop.py > ...
1 for i in range(2, 6):
2     print('Iteration:', i)

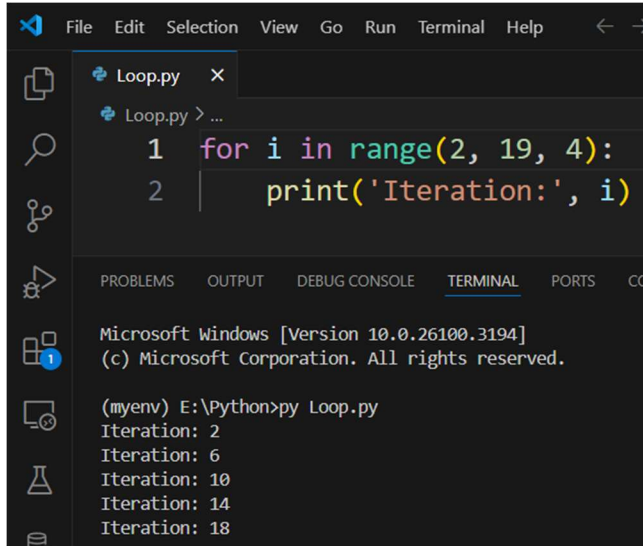
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CO
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Loop.py
Iteration: 2
Iteration: 3
Iteration: 4
Iteration: 5
```

এখানে `range(2,6)` হলো 2 থেকে 5। অর্থাৎ `range(s,n)` হবে  $s$  থেকে  $n-1$  পর্যন্ত।



যদি 1 বার ইনক্রিমেন্ট না করে 1 এর অধিক বার ইনক্রিমেন্ট করতে চাই তাহলে `range(s,n,m)` ব্যবহার করতে হবে। এক্ষেত্রে `range(s,n,m)` হবে s থেকে n-1 পর্যন্ত m বার ইনক্রিমেন্ট হয়ে। যেমনঃ



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

```
1 for i in range(2, 19, 4):  
2     print('Iteration:', i)
```

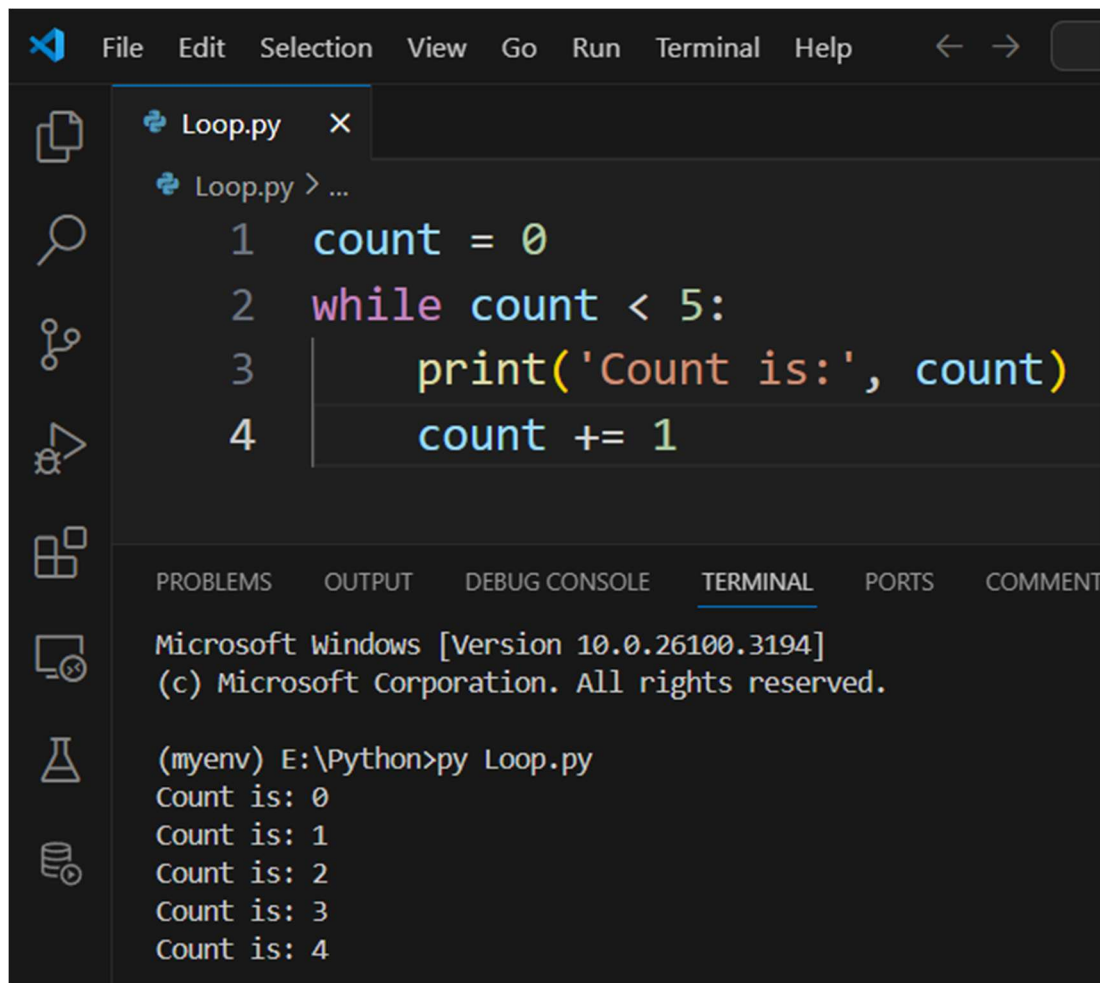
The terminal output shows the execution of the code:

```
Microsoft Windows [Version 10.0.26100.3194]  
(c) Microsoft Corporation. All rights reserved.  
  
(myenv) E:\Python>py Loop.py  
Iteration: 2  
Iteration: 6  
Iteration: 10  
Iteration: 14  
Iteration: 18
```

এখানে `range(2,19,4)` হলো 2 থেকে 18 তে যাবে 4 ইনক্রিমেন্ট হয়ে, যেমনঃ  $2+4=6$  আর  $6+4=10$  এভাবে।

## While লুপঃ

অন্যান্য প্রোগ্রামিং ভাষার মত পাইথনেও While loop আছে। নিচে এর চিত্র দেওয়া হলো।



The image shows a code editor window with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The editor has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Remote Explorer, and Testing. The main editor area shows a file named 'Loop.py' with the following Python code:

```
1 count = 0
2 while count < 5:
3     print('Count is:', count)
4     count += 1
```

Below the editor, the 'TERMINAL' tab is active, showing the output of running the script:

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

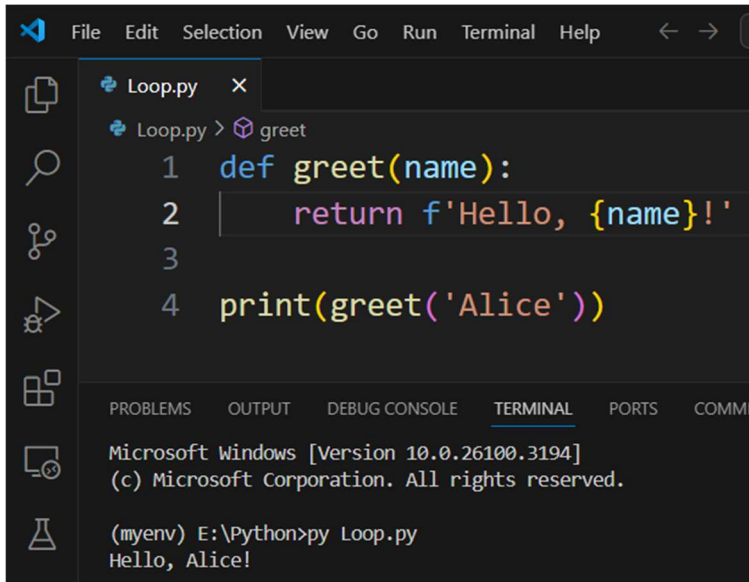
(myenv) E:\Python>py Loop.py
Count is: 0
Count is: 1
Count is: 2
Count is: 3
Count is: 4
```

## ফাংশন (Functions)

যদিও পাইথনে C এর মত `main()` ফাংশন নেই, তা সত্ত্বেও পাইথনে ফাংশন বানানো যায় `def` এর মাধ্যমে।

স্ট্রিং `return` এর ক্ষেত্রে `return` এর পর `f` লিখে স্ট্রিং দিতে হয়।

নিচে ফাংশনের উদাহরণ দেওয়া হলো-



```
File Edit Selection View Go Run Terminal Help
Loop.py x
Loop.py > greet
1 def greet(name):
2     return f'Hello, {name}!'
3
4 print(greet('Alice'))

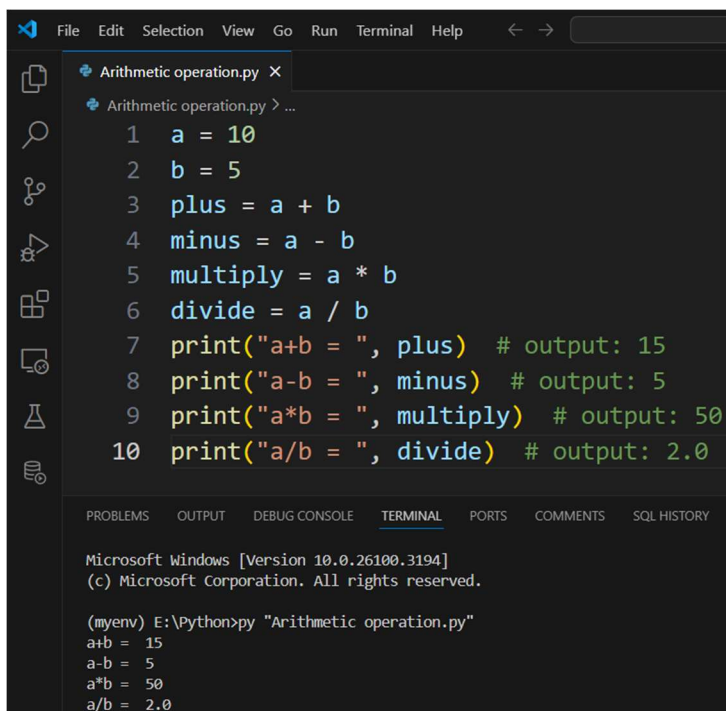
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Microsoft Windows [Version 10.0.26100.3194]
(c) Microsoft Corporation. All rights reserved.

(myenv) E:\Python>py Loop.py
Hello, Alice!
```

## অ্যারিদমেটিক অপারেশনঃ

পাইথনে অ্যারিদমেটিক অপারেশন নিচে দেখানো হল-

### ১) যোগ, বিয়োগ, গুন, ভাগঃ



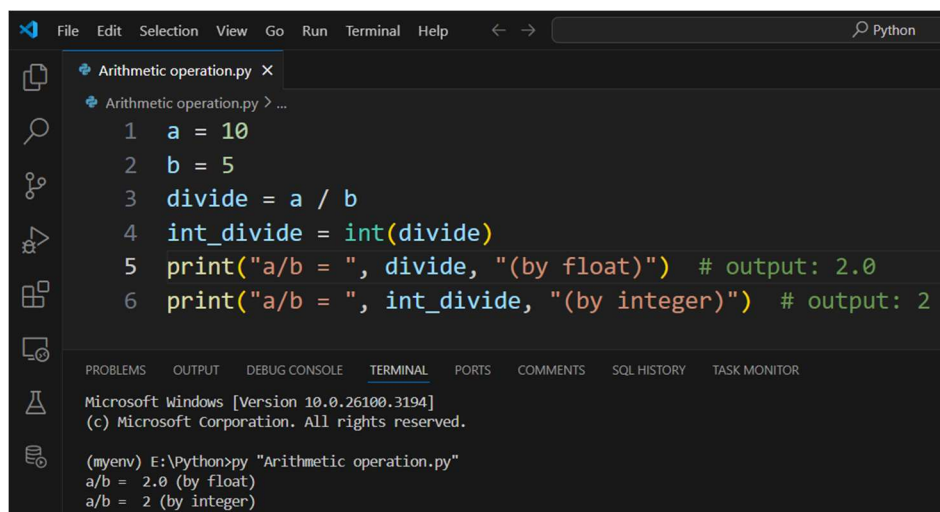
```
File Edit Selection View Go Run Terminal Help
Arithmetic operation.py x
Arithmetic operation.py > ...
1 a = 10
2 b = 5
3 plus = a + b
4 minus = a - b
5 multiply = a * b
6 divide = a / b
7 print("a+b = ", plus) # output: 15
8 print("a-b = ", minus) # output: 5
9 print("a*b = ", multiply) # output: 50
10 print("a/b = ", divide) # output: 2.0

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

(myenv) E:\Python>py "Arithmetic operation.py"
a+b = 15
a-b = 5
a*b = 50
a/b = 2.0
```

এখানে  $10/5 = 2.0$  দেখিয়েছে, শুধু 2 দেখায়নি। অর্থাৎ দুটি integer ভাগ হলে float হবে।

এখন  $10/5 = 2$  (integer) এ আনতে হলে `int()` ফাংশন ব্যবহার করতে হবে।



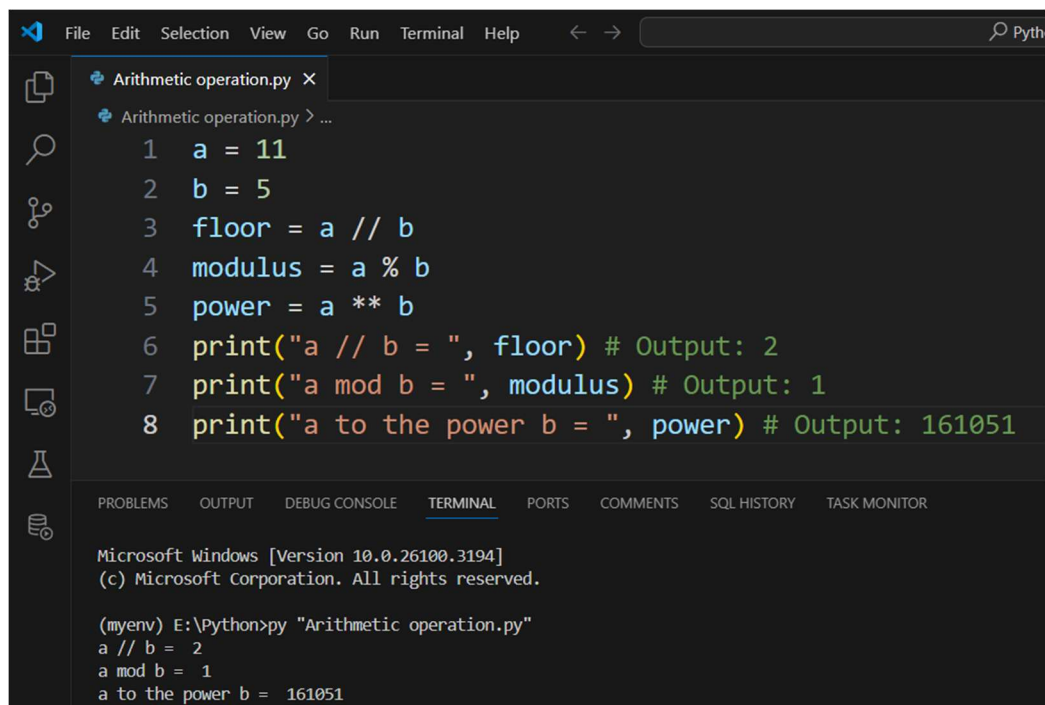
```
File Edit Selection View Go Run Terminal Help Python
Arithmetic operation.py X
Arithmetic operation.py > ...
1 a = 10
2 b = 5
3 divide = a / b
4 int_divide = int(divide)
5 print("a/b = ", divide, "(by float)") # output: 2.0
6 print("a/b = ", int_divide, "(by integer)") # output: 2

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 "Arithmetic operation.py"
a/b = 2.0 (by float)
a/b = 2 (by integer)
```

২) পূর্ণসংখ্যা ভাগ ( Floor Division //), ভাগশেষ (Modulus %), ঘাত (Power \*\*)



```
File Edit Selection View Go Run Terminal Help Python
Arithmetic operation.py X
Arithmetic operation.py > ...
1 a = 11
2 b = 5
3 floor = a // b
4 modulus = a % b
5 power = a ** b
6 print("a // b = ", floor) # Output: 2
7 print("a mod b = ", modulus) # Output: 1
8 print("a to the power b = ", power) # Output: 161051

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 "Arithmetic operation.py"
a // b = 2
a mod b = 1
a to the power b = 161051
```

এখানে,  $a**b$  হচ্ছে  $a^b$ ।  $a//b$  এর ক্ষেত্রে  $a = 11$  আর  $b = 5$  হলে  $11/5 = 2.2$  এর ক্ষেত্রে পূর্ণসংখ্যা 2 আসবে, ভগ্নাংশ বাদ যাবে। Modulus এর ক্ষেত্রে  $11\%2$  হলো  $11/2$  এর ভাগশেষ 1।

### ৩) অগ্রাধিকার (Operator Precedence)

Python-এ অপারেটরগুলোর অগ্রাধিকার (precedence) আছে। নিম্নলিখিত ক্রমে তারা কার্যকর হয়:

<div> <div>সর্বোচ্চ অগ্রাধিকার</div> <div>↓</div> <div>সর্বনিম্ন অগ্রাধিকার</div> </div>	( )	বন্ধনী
	**	ঘাত/ সূচক/ Power
	*, /, //, %	গুণ, ভাগ, পূর্ণসংখ্যা ভাগ, ভাগশেষ
	+, -	যোগ, বিয়োগ

### উদাহরণঃ

```

Arithmetic operation.py
1 a = 10
2 b = 5
3 c = 2
4
5 result1 = a + b * c
6 print(result1) # Output: 20 (First 5*2 = 10, then 10+10 = 20)
7
8 result2 = (a + b) * c
9 print(result2) # Output: 30 (First 10+5 = 15, then 15*2 = 30)
10
11 # Here the parenthesis has higher precedence than the multiplication operator,
12 # so the (a+b) has performed first.

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

(myenv) E:\Python>py "Arithmetic operation.py"
20
30

```

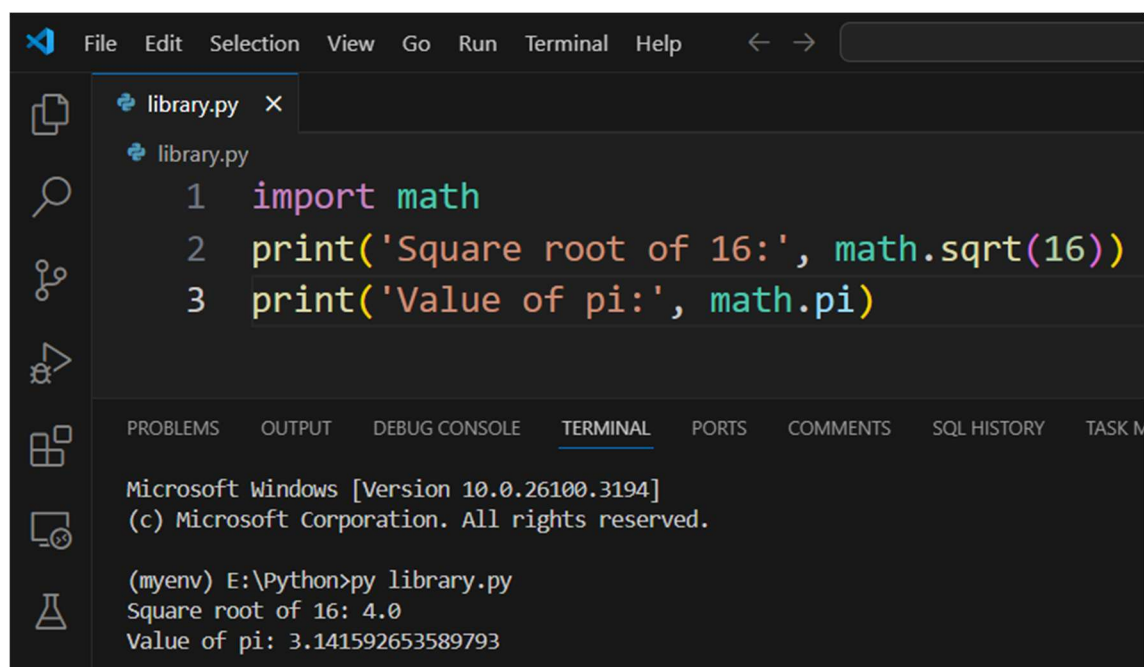
এখানে result1 এর ক্ষেত্রে  $10+5*2$ , সেক্ষেত্রে প্রথমে  $5*2=10$ , এরপর  $10+10=20$ , যেহেতু যোগ এর চেয়ে গুনের অগ্রাধিকার বেশি।

আবার result2 এর ক্ষেত্রে  $(10+5)*2$ , সেক্ষেত্রে প্রথমে  $10+5=15$ , এরপর  $15*2=30$ , কারণ গুনের চেয়ে বন্ধনীর অগ্রাধিকার বেশি, আর  $10+5$  বন্ধনীর মধ্যে আবদ্ধ আছে।

## Python লাইব্রেরি বা টুল ব্যবহার করাঃ

লাইব্রেরি বা টুল ব্যবহার এর ক্ষেত্রে আমরা import ব্যবহার করবো।

যেমনঃ math library ব্যবহার করে স্কয়ার রুট (বর্গমূল) আর পাই ব্যবহার করা।



```
File Edit Selection View Go Run Terminal Help
library.py x
library.py
1 import math
2 print('Square root of 16:', math.sqrt(16))
3 print('Value of pi:', math.pi)

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

(myenv) E:\Python>py library.py
Square root of 16: 4.0
Value of pi: 3.141592653589793
```

## উপসংহার (Conclusion)

এই নির্দেশিকায় Python-এর মৌলিক ধারণাগুলি কভার করা হয়েছে। এখন আপনি ভেরিয়েবল, ডাটা টাইপ, কন্ট্রোল স্ট্রাকচার, ফাংশন, এবং লাইব্রেরি নিয়ে কাজ করতে পারবেন। ধন্যবাদ।