Page No.
Date

Experiment 1.2

Aim: To understand the basic data types, operators expressions and input output statements.

Theory:

A. Data types: A data type defines the type of data. The data types in python are divided in two categories:

1. Immutable data types - values cannot be changed, which are:- Numbers, string, Tuple.
2. Mutable data types - values can be changed, which are :- List, Dictionaries, sets

1. Number / Numeric Data Type

a. Integer - In Python, integers are zero, positive or negative whole numbers without a fractional part and having unlimited precision.

Ex:

```
Input
num=100
print(num)
Output
100
>
```

b. Long - A long is an integer type value that has unlimited length.

C. Float - values with decimal points are the float values, there is no need to specify the data type in python.

Ex.

```
Input                    output
fnum=34.45               34.45
print(fnum)              >
```

d. Complex number → Numbers with real and imaginary parts are known as complex numbers. Python is able to identify these complex numbers with values.

Ex.

Input:	Output
cnum = 3+4j	(3+4j)
print (cnum)	>

e. Binary, Octal and Hexadecimal

Binary: 0b (zero + 'b') and 0B (zero + 'B')

Ex.

Input	Output
num = 0b101	5
print (num)	

Octal: 0o (zero + 'o') and 0O (zero + 'O')

Ex:

Input	Output
num = 0o32	26
print (num)	

Hexa decimal: 0x (zero + 'x') and 0X (zero + 'X')

Ex.

Input	output
num = 0xFF	255
print (num)	

2. String data type: String is a sequence of characters in python. The data type of string in Python is called "str".

Ex.

Input	Output
s = "this is a string"	This is a string
print (s)	

3. Tuple data type - It is an ordered collection of elements enclosed in round brackets and separated by commas.

Ex. Input
```
t = (1,2,3,4,5)
print(t)
```
output
(1,2,3,4,5)

4. List data type - It is similar to tuple, but list is enclosed in square bracket and elements are separated by commas.

Ex.

Input
```
lis = [1,2,3,4,5]
print (lis)
```
output
[1,2,3,4,5]

5. Dictionary data type - Dictionary is a collection of key and value pairs. A dictionary does not allow duplicate keys but values can be duplicate.

Ex.

Input
```
dic = { 1: 'apple', 2: 'ball' }
print(dic)
```
output
{1: 'apple', 2: 'ball'}

6. Set data type - Sets are used to store multiple items in a single variable. A set is an unordered and unindexed collection of items.

Ex.

Input
```
set = { 2, "bye" }
print (set)
```
Output
{2, 'bye'}

B. Operators: operators are used to perform operations on variables and values. Python divides the operators in the following groups:

1. Arithmetic operators

Operator	Name	Example
+	Addition	x+y
-	Subtraction	x-y
*	multiplication	x*y
/	Division	x/y
%	modulus	x%y
**	Exponentation	x**y
//	Floor division	x//y

2. Assignment operators

Operator	Example
=	y= a+b
+=	m+=10
-=	m-=10
*=	m*=10
/=	m/=10
%=	m%=10
**=	m**=10
//=	m//=10

3. Logical operators:

Operator	Description	Example
and	Returns true if both statements are true	x<5 and x<10
or	Returns true if one of the statement is true	x<5 or x<4.
not	Reverse the result, returns False if true	not(x<5 and x<4).

4. member ship operators.

Operator	Description	Example
in	Returns true if a sequence with specified value is present	x in y
not in	Returns true if a sequence with specified value is not present	x not in y

Example: Input

x = ["apple", "banana"]

print ("banana" in x).

output : True

5. Bitwise operators:

Operator	Name
&	AND
|	OR
^	XOR
~	NOT
<<	zero fill left shift
>>	signed right shift

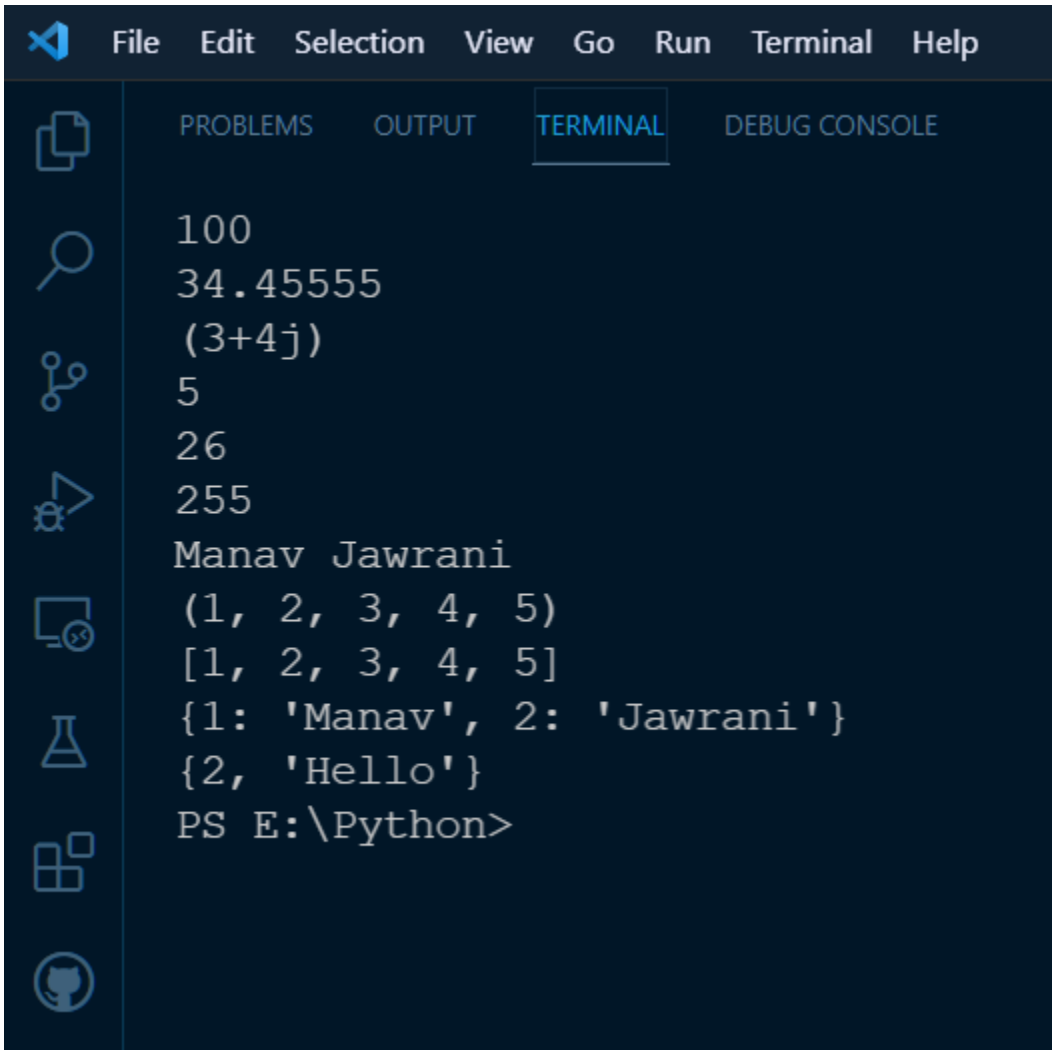
C. Expressions:
Expressions are representations of value. They are different from statements in the fact that statement do something while expressions are representation of value. Python expressions only contain identifiers, literals and operators.

D. Input and Output Functions:
Taking input from user. Sometimes a developer might want to take input from the user at some point of program. To do this python provides an input() function. To show the output in python. we can use print("") function to display the output on prompt.

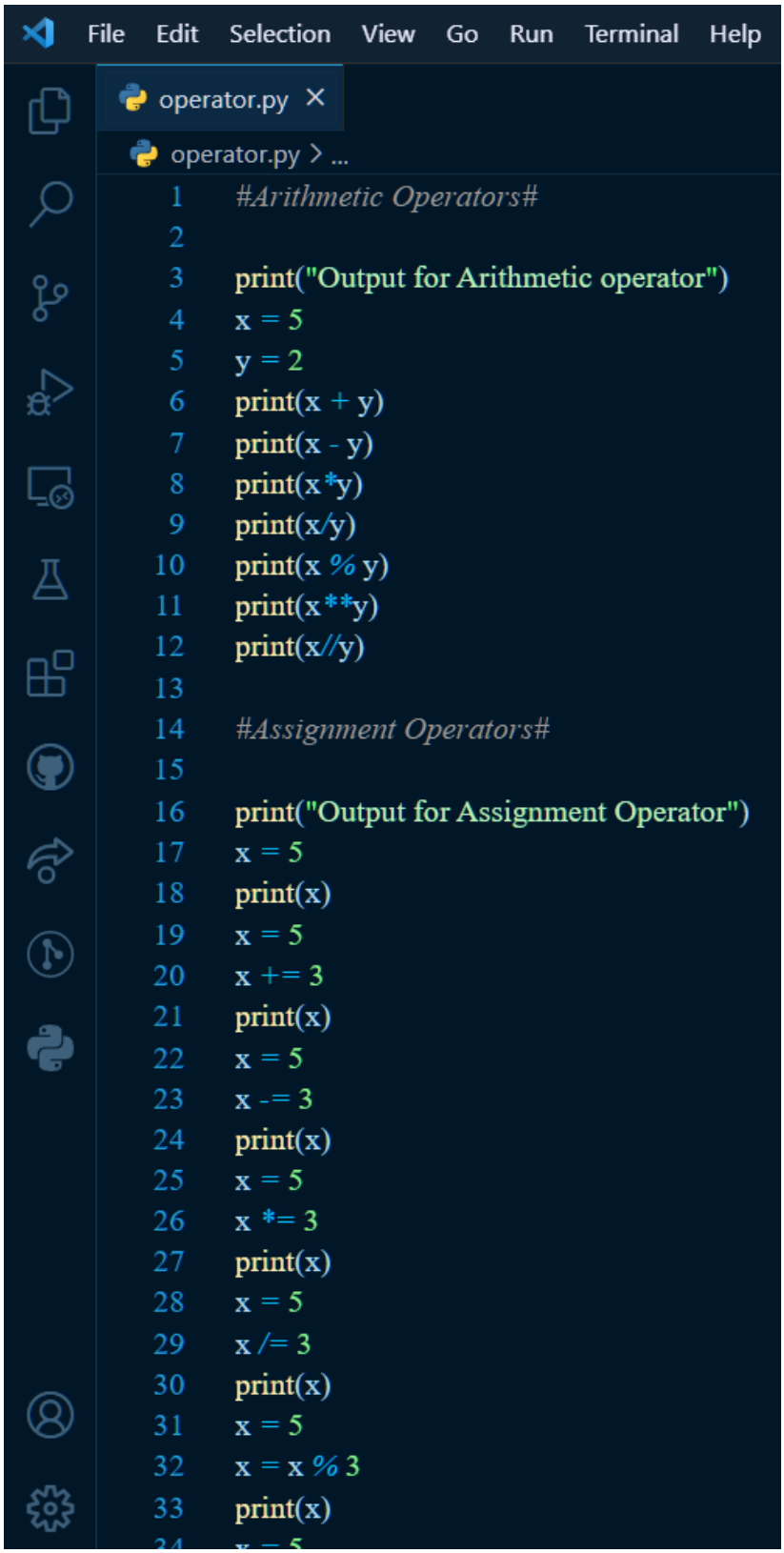
Programs:

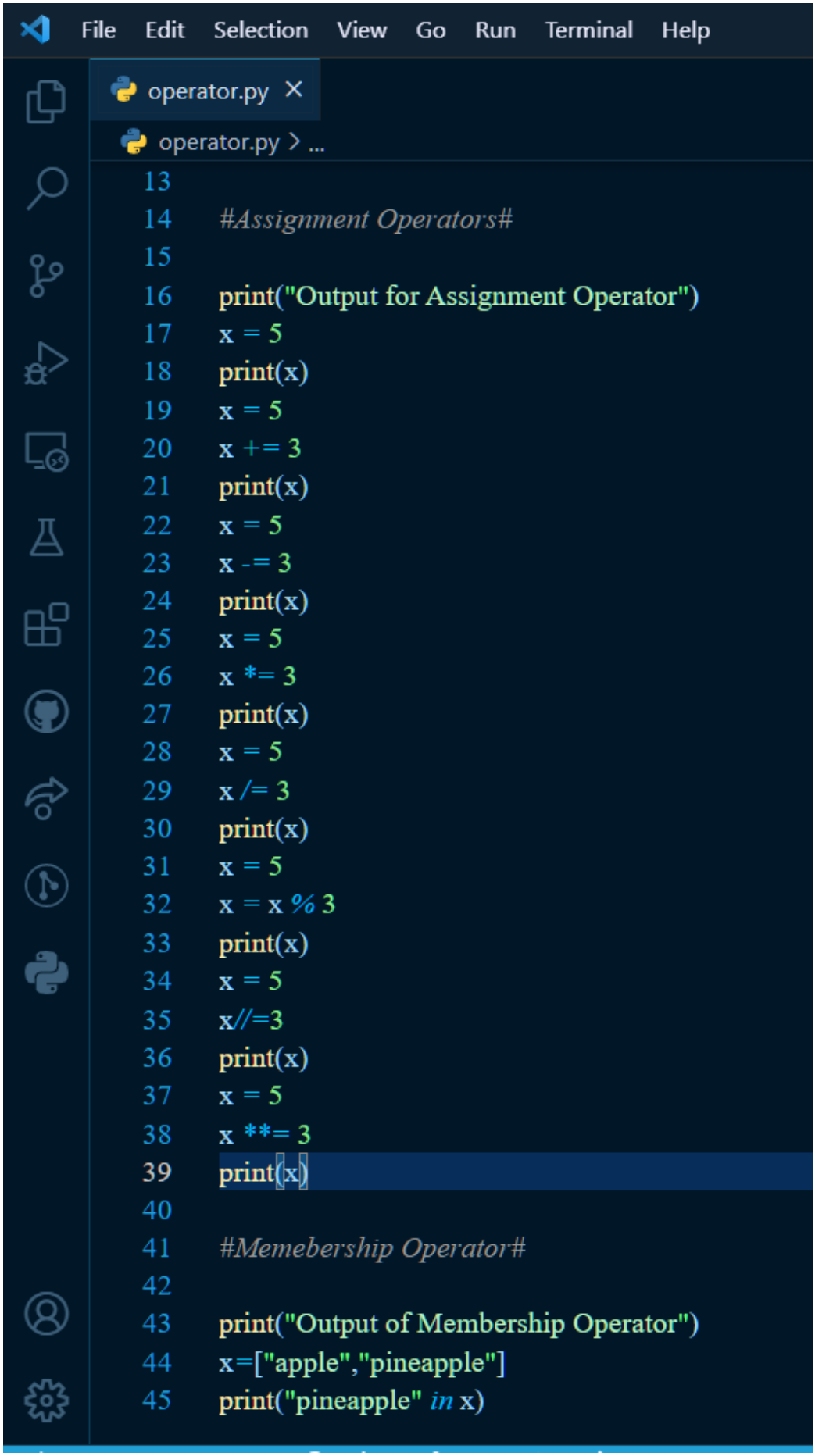
A. Data types

Output:

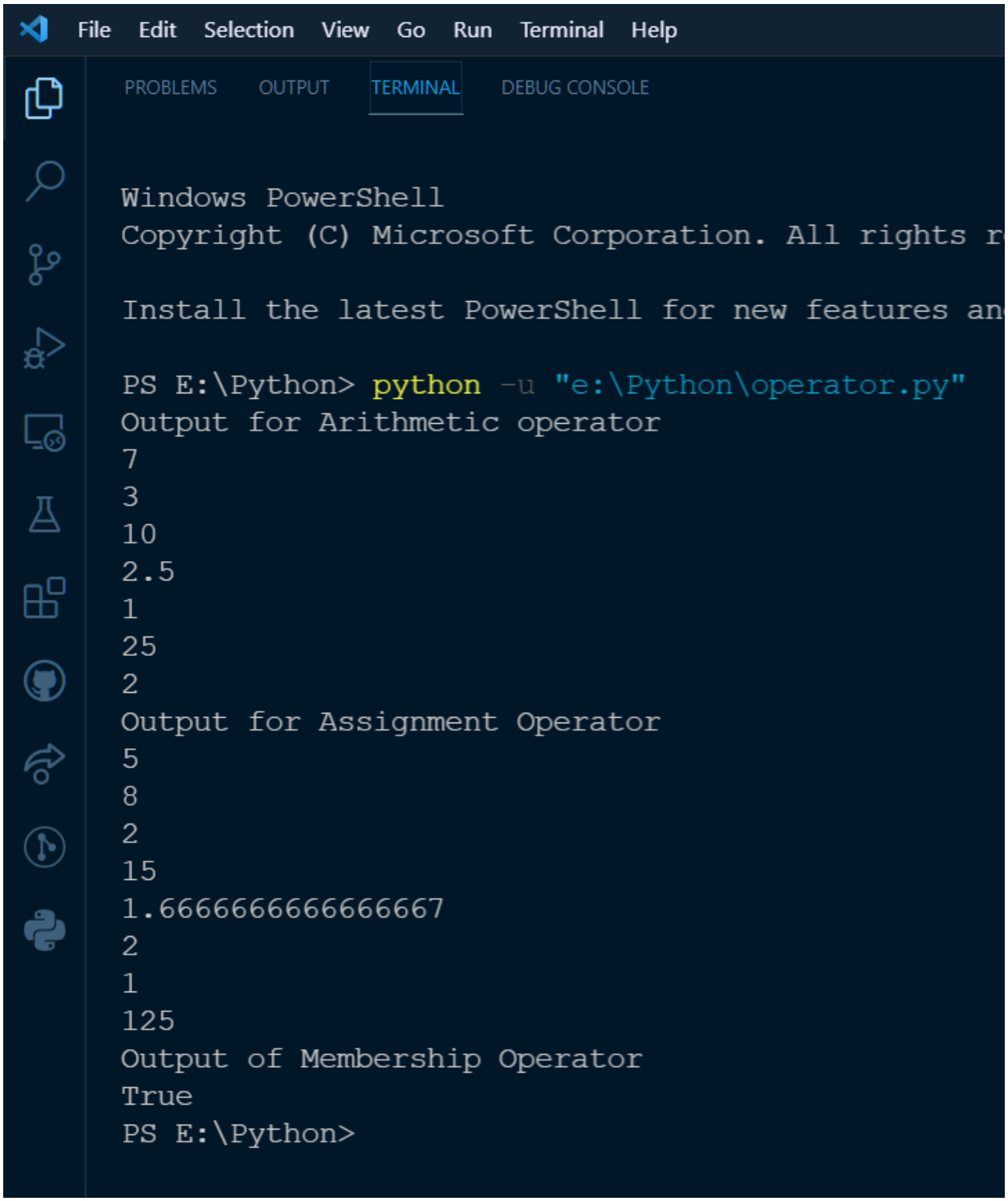

File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
100
34.45555
(3+4j)
5
26
255
Manav Jawrani
(1, 2, 3, 4, 5)
[1, 2, 3, 4, 5]
{1: 'Manav', 2: 'Jawrani'}
{2, 'Hello'}
PS E:\Python>

B. Operators


File Edit Selection View Go Run Terminal Help
operator.py
operator.py > ...
#Arithmetic Operators#
print("Output for Arithmetic operator")
x = 5
y = 2
print(x + y)
print(x - y)
print(x*y)
print(x/y)
print(x % y)
print(x**y)
print(x//y)
#Assignment Operators#
print("Output for Assignment Operator")
x = 5
print(x)
x = 5
x += 3
print(x)
x = 5
x -= 3
print(x)
x = 5
x *= 3
print(x)
x = 5
x /= 3
print(x)
x = 5
x = x % 3
print(x)


File Edit Selection View Go Run Terminal Help
operator.py
operator.py > ...
13
14 #Assignment Operators#
15
16 print("Output for Assignment Operator")
17 x = 5
18 print(x)
19 x = 5
20 x += 3
21 print(x)
22 x = 5
23 x -= 3
24 print(x)
25 x = 5
26 x *= 3
27 print(x)
28 x = 5
29 x /= 3
30 print(x)
31 x = 5
32 x = x % 3
33 print(x)
34 x = 5
35 x//=3
36 print(x)
37 x = 5
38 x **= 3
39 print(x)
40
41 #Memebership Operator#
42
43 print("Output of Membership Operator")
44 x=["apple","pineapple"]
45 print("pineapple" in x)

Output:



Conclusion: We have understood different data types , operators , expressions and input output statements.