## Operators

### What is operator?

Operator is a special symbol, which is used to perform operations. Based on the operands on which it performs operations, the operators are classified into 3 categories

1. Binary Operators
2. Unary Operators
3. Ternary Operators

Binary Operator: An operator required 2 operands to perform operation is called binary operator

Unary Operator: an operator required 1 operand to perform operation is called unary operator

Ternary Operator: an operator required 3 operands to perform operation is called ternary operator.

### Types of operators

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Membership Operators
6. Identity Operators
7. Bitwise Operators
8. Conditional Operators
9. Walrus Operator

### Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations
Arithmetic operators are binary operators and this operator required 2 operands to perform operations.

| **Operators** | **Description** |
|---|---|
| + | Addition and Concatenation |
| - | Subtraction |
| * | Multiplication and Repeat |
| / | Float Division |
| // | Floor Division |
| % | Modular |
| ** | Exponent or Power of |

**+ Operator**

+ is binary operator and this operator is used to perform 2 operations

1. Adding numbers
2. Concatenating of sequences

If two operands are numeric type, **+** operator performs addition
If two operands are sequences, **+** operator performs concatenation

**Example:**

```
a=10
b=20
c=a+b
print(a,b,c)
f1=1.5
f2=1.4
```

```
f3=f1+f2
print(f1,f2,f3)
c1=1+2j
c2=2+3j
c3=c1+c2
print(c1,c2,c3)
b1=True
b2=False
b3=b1+b2
print(b1,b2,b3)
r1=10+1.5
print(r1)
r2=10+1.5+1+2j
print(r2)
```

**Output**

```
10 20 30
1.5 1.4 2.9
(1+2j) (2+3j) (3+5j)
True False 1
11.5
(12.5+2j)
```

Note: when performed arithmetic operations on 2 different data type, python always gives the result in broader type.

1. Complex
2. Float
3. Int

Int+int □ int
float+float□ float

complex+complex □ complex
int+float □ float
int+complex □ complex
float+complex □ complex
int+float+complex □ complex

```
>>> a=10
>>> b=1.5
>>> c=int(a+b)
>>> print(a,b,c)
10 1.5 11
```

**Example:**

```
# Write a program to input rollno,name
# 2 subject marks and calculate total

#Input
rollno=int(input("Rollno :"))
name=input("StudentName :")
sub1=int(input("Subject1Marks :"))
sub2=int(input("Subject2MArks :"))
#Process
total=sub1+sub2
#Output
print("Rollno",rollno)
print("StudentName",name)
print("Subject1Marks",sub1)
print("Subject2Marks",sub2)
print("TotalMarks ",total)
```

**Output**

```
Rollno :1
```

```
StudentName :naresh
Subject1Marks :98
Subject2MArks :78
Rollno 1
StudentName naresh
Subject1Marks 98
Subject2Marks 78
TotalMarks  176
```

**+** Operator also used for concatenation (combining)
If two operands are sequence type (list, tuple, string, range, bytes, bytearray). Both operands must be similar sequence type.

```
>>> x="python"+"language"
>>> print(x)
pythonlanguage
>>> y=[10,20]+[30,40]
>>> print(y)
[10, 20, 30, 40]
>>> z="python"+3.13
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    z="python"+3.13
TypeError: can only concatenate str (not "float") to str
>>> p="abc"+[1,2,3]
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    p="abc"+[1,2,3]
TypeError: can only concatenate str (not "list") to str
```

- Operator

Arithmetic subtraction operator
This operator is used to for subtracting numbers

**Example:**

```
a=10
b=5
c=a-b
print(a,b,c)
f1=1.9
f2=1.2
f3=f1-f2
print(f1,f2,f3)
b1=True
b2=False
b3=b1-b2
print(b1,b2,b3)
print(10+True)
print(1.5+True)
print(1.5+True-False)
print(1.5-True)
```

**Output**

```
10 5 5
1.9 1.2 0.7
True False 1
11
2.5
2.5
0.5
```

**Example:**

```
# Write a program to swap two numbers
```

```
n1=int(input("Enter First Number :"))
n2=int(input("Enter Second Number :"))
print("Before swaping ",n1,n2)
n3=n1
n1=n2
n2=n3
print("After swaping ",n1,n2)
n1=n1+n2
n2=n1-n2
n1=n1-n2
print("After swaping ",n1,n2)
n1,n2=n2,n1
print("After swaping ",n1,n2)
```

**Output**

```
Enter First Number :10
Enter Second Number :20
Before swaping  10 20
After swaping  20 10
After swaping  10 20
After swaping  20 10
```

***Operator**

This operator is used to perform two operations

1. Multiplication
2. Repeating of sequence

**Example:**

```
a=5
```

```
b=2
c=a*b
print(a,b,c)
f1=1.5
f2=1.5
f3=f1*f2
print(f1,f2,f3)
```

**Output**

```
5 2 10
1.5 1.5 2.25
```

**Example:**

```
# Write a program to find area of rectangle
# area=l*b

dim1=float(input("Input Dim1/L :"))
dim2=float(input("Input Dim2/B :"))
area=dim1*dim2
print("Area of Rectangle is ",area)
```

**Output**

```
Input Dim1/L :1.2
Input Dim2/B :1.2
Area of Rectangle is  1.44
```

**Example:**

```
>>> s1="a"*10
>>> print(s1)
aaaaaaaaaa
>>> A=[0]*20
>>> print(A)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
>>> s2="py"*5
>>> print(s2)
pypypypypy
>>> B=[5]*9
>>> print(B)
[5, 5, 5, 5, 5, 5, 5, 5, 5]
>>> C=[1,2]*3
>>> print(C)
[1, 2, 1, 2, 1, 2]
```

**/ Operator**

/ operator is called float division operator.
This operator divides two numbers and returns quotient
This operator returns quotient in float type

**Example:**

```
n1=4
n2=2
n3=n1/n2
print(n1,n2,n3)
print(type(n1),type(n2),type(n3))
a=5
b=2
c=a/b
print(a,b,c)
```

**Output**

```
4 2 2.0
<class 'int'> <class 'int'> <class 'float'>
```

5 2 2.5

**Example:**

```
# Write a program to find simple interest
# si=ptr/100

p=int(input("Amount :"))
t=int(input("Time :"))
r=float(input("Rate :"))
si=p*t*r/100
print("Simple Interest ",si)
```

**Output**

```
Amount :5000
Time :12
Rate :1.2
Simple Interest  720.0
```

**// Operator**

// operator is called floor division operator
This operator divides two numbers and returns quotient
It returns quotient in integer type. This integer value is less than quotient

```
>>> n1=5/2
>>> print(n1)
2.5
>>> n2=4/2
>>> print(n2)
2.0
>>> n3=5//2
```

```
>>> print(n3)
2
>>> n4=4//2
>>> print(n4)
2
>>> n5=-4//2
>>> print(n5)
-2
>>> n6=-5//2
>>> print(n6)
-3
>>> n7=4.0//2
>>> n7
2.0
>>> n8=5.0//2
>>> print(n8)
2.0
```

**% operator (modulo)**

This operator is used to find remainder

This operator divides two numbers and returns remainder

```
>>> 4%2
0
>>> 9%4
1
>>> 9%5
4
```

**** Exponent operator**

This operator is used to find power of a number

It is a binary operator and required 2 operands

```
>>> r1=5**2
>>> print(r1)
25
>>> r2=2**8
>>> print(r2)
256
>>> r3=4**3
>>> print(r3)
64
```

**Operator Precedence**

Operator precedence defines set of rules and regulations for execution of operators (OR) operator precedence tells order of execution of operators.

**Precedence of Arithmetic Operators**

| | |
|---|---|
| ** | Exponent |
| *,/,//,% | Multiply,float div,floor div, modulo |
| +,- | Addition,subtraction |

The operators within same box are given same precedence
The operators within same box are evaluated from left to right
But ** operator is evaluated from right to left