We have three numeric data types – **integer**, **float** and **complex**. Let's see how we can use arithmetic operators on these data objects. Since you are already familiar with arithmetic operations in general, you can quickly learn about how the operators work in Python.

Addition

Operator for Addition, +

Adds the operands on either side of the operator.

Example 1	**Example 2**
>>> a=21	>>> a=-21
>>> b=10	>>> b=10
>>> c=a+b	>>> c=a+b
>>> c	>>> c
31	-11

Subtraction

Operator for Subtraction, -

Subtracts the operand on the right from the operand on the left.

Example 1 Example 2

```
>>> a=21    >>> a=-21

>>> b=10    >>> b=10

>>> c=a-b   >>> c=a-b

>>> c       >>> c

11          -31
```

Multiplication

Operator for Multiplication, *

Multiplies values on either side of the operator.

Example 1 Example 2

```
>>> a=21    >>> a= -21
>>> b=10    >>> b=10
>>> c=a*b   >>> c=a*b
>>> c       >>> c
210         -210
```

Division

Operator for Division, /

Divides left hand operand by right hand operand.

Example 1 Example 2

```
>>> a=21    >>> a= -21
```

>>> b=10 >>> b=10

>>> c=a/b >>> c=a/b

>>> c >>> c

2.1 -2.1

Modulus

Operator for Modulus, %

Returns the remainder of division of the operand on the left by the operand on the right of the operator.

Example 1

>>> a=21

>>> b=10

>>> c=a%b

Example 2

>>> a= -21

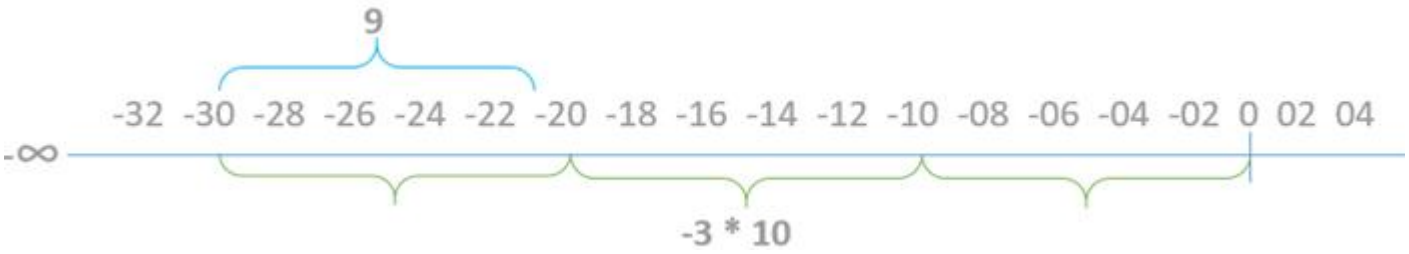
>>> b=10

>>> c=a%b

```
>>> c
1
```

```
>>> c
9
```

Note: In Python, the modulus is calculated towards negative infinity. Thus, 21%10 is 1 because the closest multiple of 10 towards negative infinity is 20. But -21%10 is 9 because the closest multiple of 10 towards negative infinity is -30 and -30+9 is -21.



Exponent

Operator for Exponent, **

Calculates the value of the operand on the left raised to operand on the right of the operator.

Example 1 **Example 2**

```
>>> a=4        >>> a=-4

>>> b=3        >>> b=3

>>> c=a**b     >>> c=a**b

>>> c          >>> c

64             -64
```

Floor division

Operator for Floor Division, //

Returns the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity).

Example 1 **Example 2**

```
>>> a=9
>>> b=5
>>> c=a//b
>>> c
1
```

Explanation: 9/5 = 1.8. So the floor division result removes the 8 and returns 1.

```
>>> a=-9
>>> b=5
>>> c=a//b
>>> c
-2
```

Explanation: -9/5 = -1.8. However, the floor division rounds this away from 0 and makes it -2.