Python Numbers with Examples

Python Built-in Datatypes include Numeric types, Boolean Type, Sequence Type, etc. Python Numbers are the ones with Numeric Types. Variables stores data of different types, and what operations can be performed on that data is represented by the datatype.

To store Number types, such as integer, floats, etc., Python Numbers includes datatypes, like the int, float, long and complex data types. Let us see some examples before going through each datatype:

```
#!/usr/bin/python

# int datatype
val1 = 1
val2 = 100
print("Value 1 (Integer): ",val1)
print("Value 2 (Integer): ",val2)

# float datatype
val3 = 6.7
val4 = 5E3
print("Value 3 (Float): ",val3)
print("Value 4 (Float): ",val4)

# complex datatype
val5 = 1j
val6 = 3.25+6.30j
print("Value 5 (Complex):",val5)
print("Value 6 (Complex):",val6)
```

The output is as follows:

```
Value 1 (Integer):  1
Value 2 (Integer):  100
Value 3 (Float):  6.7
Value 4 (Float):  5000.0
Value 5 (Complex): 1j
Value 6 (Complex): (3.25+6.3j)
```

Following are the types:

- Int
- Float
- Long
- Complex

Let us begin with int datatype:

Int Datatype

Int (Integer) is a positive or negative whole number, without a fractional part and of unlimited length. For example, 1, 5, 200, -10, 25, -550, etc. Also includes the octal and hexadecimal numbers. Let us now see some examples of the Int datatype in Python:

```
#!/usr/bin/python

# int datatype
val1 = 10
val2 = -100
print("Value 1 (Integer): ",val1)
print("Value 2 (Integer): ",val2)
```

The output as follows:

```
Value 1 (Integer): 10
Value 2 (Integer): -100
```

Let us see another example:

```
#!/usr/bin/python

## int datatype

# Hexadecimal number
val1 = 0x10;
print("Value 1: ",val1)

# Octal number
val2 = 0010;
print("Value 2: ",val2)
```

The output as follows:

```
Value 1: 16
Value 2: 8
```

Let us see another example to know the int type:

```
#!/usr/bin/python

## int datatype

val1 = 5
print("Value 1 type: ",type(val1))

val2 =  -10
print("Value 2 type: ",type(val2))

val3 =  0
print("Value 3 type: ",type(val3))
```

The output is as follows:

```
Value 1 type: <class 'int'>
Value 2 type: <class 'int'>
Value 3 type: <class 'int'>
```

Long Datatype

The Long datatype are int types with unlimited size. Always remember to suffix the number with L (lowercase or uppercase) for a long type. For example, 1234L, 4657688L, -5456859L, etc.

Note: Use Uppercase L, since lowercase l will lead to confusion in number, for example, 56586881l, difficulty in differentiation between 1 and lowercase l. Better approach to use uppercase L.

Note: In Python 2.7, int and long int are available, but in Python 3, only one type int works.

Let us see an example of long datatype in Python (Run in Python 2):

```
#!/usr/bin/python

# long datatype

val1 = 678676765L;
print("Value 1: ",val1)

val2 = 3545L;
print("Value 2: ",val2)
```

```
val3 = -8687687L
print("Value 3: ",val3)
```

The output is as follows (Run in Python 2):

```
Value 1:  678676765L
Value 2:  3545L
Value 3: -8687687L
```

Float datatype

Float is floating point positive or negative number, with decimal(s). The fractional part is also denoted as a notation i.e. E or e. This represents a shot form for the big float numbers.

For example, 325.6 is represented as:

```
3.256E2
```

For example, 2.615E8 is represented as:

```
261500000
```

Let us see an example for float datatype in Python:

```
#!/usr/bin/python

## float datatype

val1 = 10.30
print("Value 1: ",val1)

val2 = -300.10
print("Value 2: ",val2)

val3 = 3.256E2
print("Value 3: ",val3)

val4 = 2.615E8
print("Value 3: ",val4)
```

The output is as follows:

```
Value 1: 10.3
Value 2: -300.1
Value 3: 325.6
```

```
Value 3: 261500000.0
```

Let us see an example to know the float type:

```
#!/usr/bin/python

## float datatype

val1 = 5.2
print("Value 1 type: ",type(val1))

val2 =  3.256E2
print("Value 2 type: ",type(val2))
```

The output is as follows:

```
Value 1 type: <class 'float'>
Value 2 type: <class 'float'>
```

Complex Datatype in Python Numbers

Under complex datatype, numbers are a combination of real and imaginary parts. The imaginary component has a "j" as imaginary part. For example, 2 + 7j, 20j, 8-2j, -30j, etc.

Above, for 2 + 7j:

```
Real component is 2
Imaginary component is 6j
```

Let us see an example for complex datatype in Python:

```
#!/usr/bin/python

# complex datatype

val1 = 3 + 2j
print("Value 1: ",val1)

val2 = -200j
print("Value 2: ",val2)

val3 = 9 - 3j
print("Value 3: ",val3)

val4 = 10j
print("Value 3: ",val4)
```

The output is as follows:

```
Value 1: (3+2j)
Value 2: (-0-200j)
Value 3: (9-3j)
Value 3: 10j
```

Let us see another example to know the type:

```
#!/usr/bin/python

# complex datatype

val1 = 3 + 2j
print("Value 1 type: ",type(val1))

val2 = 5j
print("Value 2 type: ",type(val2))
```

The output is as follows:

```
Value 1 type: <class 'complex'>
Value 2 type: <class 'complex'>
```

Python Numbers include datatypes, such as int, float, long and complex.