**Numbers**

Number data types store numeric values. Number objects are created when you assign a value to them. For example:

var1 = 1

var2 = 10

You can also delete the reference to a number object by using the del statement. The syntax of the del statement is:

**del** var1[,var2[,var3[....,varN]]]]

You can delete a single object or multiple objects by using the del statement. For example:

del var

del var\_a, var\_b

Python supports four different numerical types:

1. Int (signed integers)
2. Long (long integers), they can also be represented in octal and hexadecimal)
3. Float (floating point real values)
4. Complex (complex numbers)

Python allows you to use a lowercase l with long.

But it is recommended to use only an uppercase L to avoid confusion with the number 1.

Python displays long integers with an uppercase L.

|  |  |  |  |
| --- | --- | --- | --- |
| int | Long | float | complex |
| 10 | 51924361L | 0 | 3.14j |
| 100 | -0x19323L | 15.2 | 45.j |
| -786 | 0122L | -21.9 | 9.322e-36j |
| 80 | 0xDEFABCECBDAECBFBAEl | 32.3+e18 | .876j |
| -490 | 535633629843L | -90 | -.6545+0J |
| -0x260 | -052318172735L | -3.25E+101 | 3e+26J |
| 0x69 | -4721885298529L | 70.2-E12 | 4.53e-7j |

# Number Type Conversion

Python converts numbers internally in an expression containing mixed types to a common type for evaluation.

Sometimes, you need to coerce a number explicitly from one type to another to satisfy the requirements of an operator or function parameter.

**int(x)** convert x to a plain integer

**long(x)** convert x to a long integer

**float(x)** convert x to a floating-point number

**complex(x)** convert x to a complex number with real part x and imaginary part zero

**complex(x, y)** convert x and y to a complex number with real part x and imaginary part y. x and y are numeric expressions

# Mathematical Constants

**pi** The mathematical constant pi

**e** The mathematical constant e

#!/usr/bin/python

x = 36

y = 036

z = 0x36

print x

print y

print z

c = 3+4j

print c

ln = 3.20 \*\* 2

print ln

Output:

36

30

54

(3+4j)

10.24

# Division in Python 2 & 3

## Integer Operations in Python 2

#!/usr/bin/python

print 8 + 5

print 8 - 5

print 8 \* 5

print 8 / 5

print 8.0 / 5

Output:

13

3

40

1 # In Python 3 it will be 1.6

1.6

## Integer Operations in Python 3

#!/usr/bin/python

print (8 + 5)

print (8 - 5)

print (8 \* 5)

print (8 / 5)

print (8.0 / 5)

Output

13

3

40

1.6 # In Python 2 It will be 1

1.6