Python - Numbers

Number data types store numeric values. They are immutable data types, means that changing the value of a number data type results in a newly allocated object.

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 −

int (signed integers) − They are often called just integers or ints, are positive or negative whole numbers with no decimal point.

long (long integers) − Also called longs, they are integers of unlimited size, written like integers and followed by an uppercase or lowercase L.

float (floating point real values) − Also called floats, they represent real numbers and are written with a decimal point dividing the integer and fractional parts. Floats may also be in scientific notation, with E or e indicating the power of 10 (2.5e2 = 2.5 x 10^2 = 250).

complex (complex numbers) − are of the form a + bJ, where a and b are floats and J (or j) represents the square root of -1 (which is an imaginary number). The real part of the number is a, and the imaginary part is b. Complex numbers are not used much in Python programming.

Examples

Here are some examples of numbers

int	long	float	complex
10	51924361L	0.0	3.14j
100	-0x19323L	15.20	45.j
-786	0122L	-21.9	9.322e-36j
080	0xDEFABCECBDAECBFBAEL	32.3+e18	.876j

-0490	535633629843L	-90.	-.6545+0J
-0x260	-052318172735L	-32.54e100	3e+26J
0x69	-4721885298529L	70.2-E12	4.53e-7j

Python allows you to use a lowercase L with long, but it is recommended that you use only an uppercase L to avoid confusion with the number 1. Python displays long integers with an uppercase L.

A complex number consists of an ordered pair of real floating point numbers denoted by a + bj, where a is the real part and b is the imaginary part of the complex number.

Number Type Conversion

Python converts numbers internally in an expression containing mixed types to a common type for evaluation. But sometimes, you need to coerce a number explicitly from one type to another to satisfy the requirements of an operator or function parameter.

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

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

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

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

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

Mathematical Functions

Python includes following functions that perform mathematical calculations.

Sr.No.	Function & Returns (description)
1	**abs(x)** The absolute value of x: the (positive) distance between x and zero.
2	**ceil(x)** The ceiling of x: the smallest integer not less than x
3	**cmp(x, y)** -1 if x < y, 0 if x == y, or 1 if x > y
4	**exp(x)** The exponential of x: e^x

5	**fabs(x)** The absolute value of x.
6	**floor(x)** The floor of x: the largest integer not greater than x
7	**log(x)** The natural logarithm of x, for x> 0
8	**log10(x)** The base-10 logarithm of x for x> 0.
9	**max(x1, x2,...)** The largest of its arguments: the value closest to positive infinity
10	**min(x1, x2,...)** The smallest of its arguments: the value closest to negative infinity
11	**modf(x)** The fractional and integer parts of x in a two-item tuple. Both parts have the same sign as x. The integer part is returned as a float.
12	**pow(x, y)** The value of x**y.
13	**round(x [,n])** **x** rounded to n digits from the decimal point. Python rounds away from zero as a tie-breaker: round(0.5) is 1.0 and round(-0.5) is -1.0.
14	**sqrt(x)** The square root of x for x > 0

Random Number Functions

Random numbers are used for games, simulations, testing, security, and privacy applications. Python includes following functions that are commonly used.

Sr.No.	Function & Description
1	**choice(seq)** A random item from a list, tuple, or string.

2	**randrange ([start,] stop [,step])** A randomly selected element from range(start, stop, step)
3	**random()** A random float r, such that 0 is less than or equal to r and r is less than 1
4	**seed([x])** Sets the integer starting value used in generating random numbers. Call this function before calling any other random module function. Returns None.
5	**shuffle(lst)** Randomizes the items of a list in place. Returns None.
6	**uniform(x, y)** A random float r, such that x is less than or equal to r and r is less than y

Trigonometric Functions

Python includes following functions that perform trigonometric calculations.

Sr.No.	Function & Description
1	**acos(x)** Return the arc cosine of x, in radians.
2	**asin(x)** Return the arc sine of x, in radians.
3	**atan(x)** Return the arc tangent of x, in radians.
4	**atan2(y, x)** Return atan(y / x), in radians.
5	**cos(x)** Return the cosine of x radians.
6	**hypot(x, y)** Return the Euclidean norm, sqrt(x*x + y*y).
7	**sin(x)** Return the sine of x radians.

8	**tan(x)** Return the tangent of x radians.
9	**degrees(x)** Converts angle x from radians to degrees.
10	**radians(x)** Converts angle x from degrees to radians.

Mathematical Constants

The module also defines two mathematical constants −

Sr.No.	Constants & Description
1	**pi** The mathematical constant pi.
2	**e** The mathematical constant e.