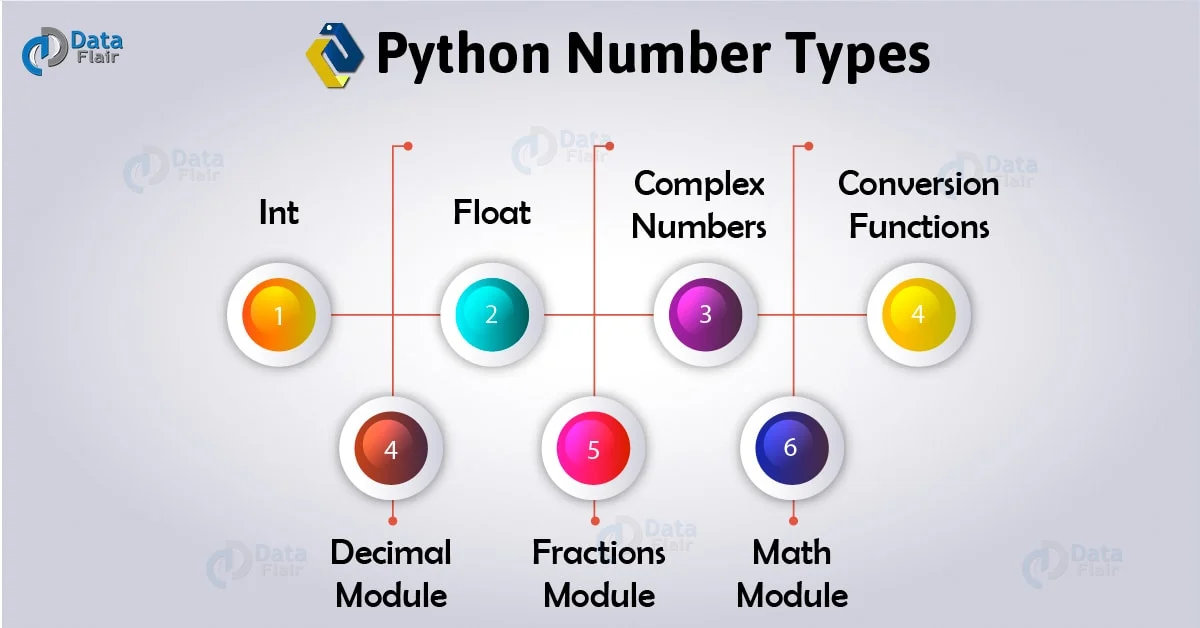
# **Python Numbers, Type Conversion and Mathematics**



The number [data types](https://www.programiz.com/python-programming/variables-datatypes) are used to store the numeric values.

Python supports integers, floating-point numbers and complex numbers. They are defined as int, float, and complex classes in Python.

* int - holds signed integers of non-limited length.
* float - holds floating decimal points and it's accurate up to 15 decimal places.
* complex - holds complex numbers.

## **Python Numeric Data Type**

Integers and floating points are separated by the presence or absence of a decimal point. For instance,

* 5 is an integer
* 5.42 is a floating-point number.

Complex numbers are written in the form, x + yj, where x is the real part and y is the imaginary part.

We can use the [type()](https://www.programiz.com/python-programming/methods/built-in/type) function to know which class a [variable](https://www.programiz.com/python-programming/variables-constants-literals) or a value belongs to.

Let's see an example,

num1 = 5

print(num1, 'is of type', type(num1))

num2 = 5.42

print(num2, 'is of type', type(num2))

num3 = 8+2j

print(num3, 'is of type', type(num3))

[Run Code](https://www.programiz.com/python-programming/online-compiler)

Output

5 is of type <class 'int'>

5.42 is of type <class 'float'>

(8+2j) is of type <class 'complex'>

In the above example, we have created three variables named num1, num2 and num3 with values 5, 5.42, and 8+2j respectively.

We have also used the type() function to know which class a certain variable belongs to. Since,

* 5 is an integer value, type() returns int as the class of num1 i.e <class 'int'>
* 5.42 is a floating value, type() returns float as the class of num2 i.e <class 'float'>
* 1 + 2j is a complex number, type() returns complex as the class of num3 i.e <class 'complex'>

## **Number Systems**

The numbers we deal with every day are of the decimal (base 10) number system.

But computer programmers need to work with binary (base 2), hexadecimal (base 16) and octal (base 8) number systems.

In Python, we can represent these numbers by appropriately placing a prefix before that number. The following table lists these prefixes.

| Number System | Prefix |
| --- | --- |
| Binary | 0b or 0B |
| Octal | 0o or 0O |
| Hexadecimal | 0x or 0X |

Here are some examples

print(0b1101011) # prints 107

print(0xFB + 0b10) # prints 253

print(0o15) # prints 13

[Run Code](https://www.programiz.com/python-programming/online-compiler)

## **Type Conversion in Python**

In programming, type conversion is the process of converting one type of number into another.

Operations like addition, subtraction convert integers to float implicitly (automatically), if one of the operands is float. For example,

print(1 + 2.0) # prints 3.0

[Run Code](https://www.programiz.com/python-programming/online-compiler)

Here, we can see above that 1 (integer) is converted into 1.0 (float) for addition and the result is also a floating point number.

#### Explicit Type Conversion

We can also use built-in functions like [int()](https://www.programiz.com/python-programming/methods/built-in/int), [float()](https://www.programiz.com/python-programming/methods/built-in/float) and [complex()](https://www.programiz.com/python-programming/methods/built-in/complex) to convert between types explicitly. These functions can even convert from [strings](https://www.programiz.com/python-programming/string).

num1 = int(2.3)

print(num1) # prints 2

num2 = int(-2.8)

print(num2) # prints -2

num3 = float(5)

print(num3) # prints 5.0

num4 = complex('3+5j')

print(num4) # prints (3 + 5j)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

Here, when converting from float to integer, the number gets truncated (decimal parts are removed).

Similarly when converting from integer to float, .0 is postfixed to the number.

To learn more about type conversion in Python, visit [Python Type Conversion](https://www.programiz.com/python-programming/type-conversion-and-casting).

## **Python Random Module**

Python offers the random module to generate random numbers or to pick a random item from an iterator.

First we need to import the random module. For example,

import random

print(random.randrange(10, 20))

list1 = ['a', 'b', 'c', 'd', 'e']

# get random item from list1

print(random.choice(list1))

# Shuffle list1

random.shuffle(list1)

# Print the shuffled list1

print(list1)

# Print random element

print(random.random())

[Run Code](https://www.programiz.com/python-programming/online-compiler)

Output

15

a

['d', 'b', 'c', 'e', 'a']

0.6716121217631744

To learn more about the random module, visit Python [Random Module](https://www.programiz.com/python-programming/modules/random).

## **Python Mathematics**

Python offers the math module to carry out different mathematics like trigonometry, logarithms, probability and statistics, etc. For example,

import math

print(math.pi)

print(math.cos(math.pi))

print(math.exp(10))

print(math.log10(1000))

print(math.sinh(1))

print(math.factorial(6))

[Run Code](https://www.programiz.com/python-programming/online-compiler)

Output

3.141592653589793

-1.0

22026.465794806718

3.0

1.1752011936438014

720

Here is the full list of functions and attributes available in the [Python math module](https://www.programiz.com/python-programming/modules/math).

# **Python Mathematical Functions**

## **What is math module in Python?**

The math [module](https://www.programiz.com/python-programming/modules) is a standard module in Python and is always available. To use mathematical functions under this module, you have to import the module using import math.

It gives access to the underlying C library functions. For example,

# Square root calculation

import math

math.sqrt(4)

[Run Code](https://www.programiz.com/python-programming/online-compiler)

This module does not support complex datatypes. The [cmath module](https://docs.python.org/3.0/library/cmath.html) is the complex counterpart.

## **Functions in Python Math Module**

Here is the list of all the functions and attributes defined in math module with a brief explanation of what they do.

| Function | Description |
| --- | --- |
| ceil(x) | Returns the smallest integer greater than or equal to x. |
| copysign(x, y) | Returns x with the sign of y |
| fabs(x) | Returns the absolute value of x |
| factorial(x) | Returns the factorial of x |
| floor(x) | Returns the largest integer less than or equal to x |
| fmod(x, y) | Returns the remainder when x is divided by y |
| frexp(x) | Returns the mantissa and exponent of x as the pair (m, e) |
| fsum(iterable) | Returns an accurate floating point sum of values in the iterable |
| isfinite(x) | Returns True if x is neither an infinity nor a NaN (Not a Number) |
| isinf(x) | Returns True if x is a positive or negative infinity |
| isnan(x) | Returns True if x is a NaN |
| ldexp(x, i) | Returns x \* (2\*\*i) |
| modf(x) | Returns the fractional and integer parts of x |
| trunc(x) | Returns the truncated integer value of x |
| exp(x) | Returns e\*\*x |
| expm1(x) | Returns e\*\*x - 1 |
| log(x[, b]) | Returns the logarithm of x to the base b (defaults to e) |
| log1p(x) | Returns the natural logarithm of 1+x |
| log2(x) | Returns the base-2 logarithm of x |
| log10(x) | Returns the base-10 logarithm of x |
| pow(x, y) | Returns x raised to the power y |
| sqrt(x) | Returns the square root of x |
| acos(x) | Returns the arc cosine of x |
| asin(x) | Returns the arc sine of x |
| atan(x) | Returns the arc tangent of x |
| atan2(y, x) | Returns atan(y / x) |
| cos(x) | Returns the cosine of x |
| hypot(x, y) | Returns the Euclidean norm, sqrt(x\*x + y\*y) |
| sin(x) | Returns the sine of x |
| tan(x) | Returns the tangent of x |
| degrees(x) | Converts angle x from radians to degrees |
| radians(x) | Converts angle x from degrees to radians |
| acosh(x) | Returns the inverse hyperbolic cosine of x |
| asinh(x) | Returns the inverse hyperbolic sine of x |
| atanh(x) | Returns the inverse hyperbolic tangent of x |
| cosh(x) | Returns the hyperbolic cosine of x |
| sinh(x) | Returns the hyperbolic cosine of x |
| tanh(x) | Returns the hyperbolic tangent of x |
| erf(x) | Returns the error function at x |
| erfc(x) | Returns the complementary error function at x |
| gamma(x) | Returns the Gamma function at x |
| lgamma(x) | Returns the natural logarithm of the absolute value of the Gamma function at x |
| pi | Mathematical constant, the ratio of circumference of a circle to it's diameter (3.14159...) |
| e | mathematical constant e (2.71828...) |

Here is the full list of functions and attributes available in the [Python math module](https://www.programiz.com/python-programming/modules/math).