**Модули**

Как уже говорилось, одним из преимуществ языка Python является множество разнообразных функций, которые уже реализованы и готовы к использованию. Такие функции упакованы в модули. В Python модулем называется библиотека функций, которую можно подключать к своим программам.

**Модуль math**

Модуль math – один из наиважнейших в Python. Этот модуль предоставляет обширный функционал для проведения вычислений с вещественными числами (числами с плавающей точкой).

Для использования этих функций в начале программы необходимо подключить модуль, что делается командой **import**:

import math

# программный код

После подключения модуля мы можем использовать его функции. Пусть мы хотим:

* вычислить \sqrt22​ (корень квадратный из двух);
* округлить число 3.83.8 до ближайшего целого числа вверх и вниз

Соответствующий программный код, решающий задачи выглядит так:

import math

num1 = math.sqrt(2) # вычисление корня квадратного из двух

num2 = math.ceil(3.8) # округление числа вверх

num3 = math.floor(3.8) # округление числа вниз

print(num1)

print(num2)

print(num3)

и выводит:

1.4142135623730951

4

3

**Особенности подключения модулей**

Как можно заметить из примера выше, для вызова функции мы вынуждены указывать название модуля и символ точки. С другой стороны, если функции используются достаточно часто, то такой вызов (постоянное указание названия модуля и символа точки) может усложнить программу и сделать её менее читабельной. Для того, чтобы не указывать название модуля и символ точки при вызове функций, мы пишем следующий код:

from math import \*

num1 = sqrt(2) # вычисление корня квадратного из двух

num2 = ceil(3.8) # округление числа вверх

num3 = floor(3.8) # округление числа вниз

print(num1)

print(num2)

print(num3)

Таким образом, подключение модуля следующим образом:

from math import \*

позволяет не писать название модуля и символ точки. При таком способе подключения, импортируются абсолютно все функции модуля math.

Если нужно использовать только некоторые функции модуля, то мы можем импортировать только их следующим образом:

from math import sqrt, ceil

Теперь мы можем вызывать функции sqrt() и ceil() без префикса math., однако мы не можем вызвать функцию floor(), так как она не подключена:

from math import sqrt, ceil

print(sqrt(25))

print(ceil(34.7))

print(floor(12.8)) # приведет к ошибке, так как функция floor не подключена

**Список функций модуля math**

Список наиболее часто используемых функций модуля math:

| **Функция** | **Описание** |
| --- | --- |
| **Округления** |  |
| int() | Округляет число в сторону нуля |
| round(x) | Округляет число x до ближайшего целого. Если дробная часть числа равна 0.5, то число округляется до ближайшего четного числа |
| round(x, n) | Округляет число x до n знаков после точки |
| floor(x) | Округляет число x вниз («пол») |
| ceil(x) | Округляет число x вверх («потолок») |
| abs(x) | Модуль числа x (абсолютная величина) |
| **Корни, логарифмы, степени и факториал** |  |
| sqrt(x) | Квадратный корень числа x |
| pow(x, n) | Возведение числа x в степень n |
| log(x) | Натуральный логарифм числа x. Основание натурального логарифма равно числу e |
| log10(x) | Десятичный логарифм числа x. Основание десятичного логарифма равно числу 10 |
| log(x, b) | Логарифм числа x по основанию b |
| factorial(n) | Факториал натурального числа n |
| **Тригонометрия** |  |
| degrees(x) | Преобразует угол x, заданный в радианах, в градусы |
| radians(x) | Преобразует угол x, заданный в градусах, в радианы |
| cos(x) | Косинус угла x, задаваемого в радианах |
| sin(x) | Синус угла x, задаваемого в радианах |
| tan(x) | Тангенс угла x, задаваемого в радианах |
| acos(x) | Возвращает угол в радианах от 00 до \pi*π*, cos которого равен x |
| asin(x) | Возвращает угол в радианах от - \frac{\pi}{2}−2*π*​ до \frac{\pi}{2}2*π*​, sin которого равен x |
| atan(x) | Возвращает угол в радианах от - \frac{\pi}{2}−2*π*​ до \frac{\pi}{2}2*π*​, tan которого равен x |
| atan2(y, x) | Полярный угол (в радианах) точки с координатами (x, y) |

    Для извлечения квадратного корня можно воспользоваться кодом n \*\* 0.5, вместо math.sqrt(n).

**Список констант модуля math**

Модуль math предоставляет ряд встроенных математических констант:

| **Константа** | **Описание** |
| --- | --- |
| pi | Число \pi = 3.141592653589793*π*=3.141592653589793 |
| e | Число e = 2.718281828459045*e*= 2.718281828459045 ([константа Эйлера](http://concepture.club/post/nauka/vse-chto-nuzhno-znat-o-konstante-e)) |

**Примечания**

**Примечание 1.** Все функции модуля math возвращают значение, которое можно вывести на экран, присвоить другой переменной или использовать в математическом выражении.

**Примечание 2.** Для использования функций int(), float(), abs(), min(), max(), round() подключать модуль math нет необходимости. Это так называемые встроенные функции.

**Примечание 3.** Вызов функций pow(x, n) можно заменить использованием оператора возведения в степень: x\*\*n.