



<https://dfedorov.spb.ru>

Модуль – коллекция переменных и функций, которые объединены вместе в одном файле.

A terminal window with a grey title bar and three window control buttons (red, yellow, green) on the left. The text inside the terminal is `>>> import math`. The prompt `>>>` is in red, `import` is in orange, and `math` is in black.

```
>>> import math
```

Для получения доступа к переменным и функциям модуля его необходимо импортировать.

Импортирование модуля создает новую переменную с именем *math*.

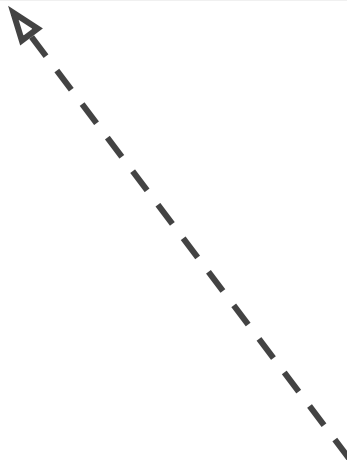
```
>>> help(math)
```

[Подробнее про модуль math](#)

```
>>> math.sqrt(9)
```



Указываем имя модуля,
который ранее
импортировали



Python находит *math*,
просматривает модуль, находит
функцию *sqrt* внутри этого
модуля и затем её выполняет

```
>>> from math import sqrt
>>> sqrt(9)
3.0
```

Импортировали
из модуля
только
функцию `sqrt`

```
>>> import math
>>> dir(math)
```

После импорта
отдельной
функции не
указываем имя
модуля.

Показывает
содержимое
модуля

math.factorial(50)

Возвращает факториал заданного числа x .

math.gcd(x)

Возвращает наибольший общий делитель (НОД) двух целых чисел a и b .

math.log(x[, base])

Если указано только число x , то возвращается натуральный логарифм данного числа. Если указано число x и основание $base$, то возвращается логарифм числа по указанному основанию.

math.exp(x)

Возвращает $e^{**}x$.

math.sin(x)

Возвращает синус угла x значение которого задано в радианах.

math.hypot(x, y)

Вычисляет значение гипотенузы по заданным значениям катетов x и y . Данная функция равносильна команде $\text{sqrt}(x*x + y*y)$.

[Подробнее про модуль math](#)

```
import math
```

```
# Гипотенуза прямоугольного треугольника с катетами a и b
```

```
a, b = 4, 3
```

```
print(math.hypot(a, b))          # 5.0
```

```
print(math.sqrt(a**2 + b**2))   # 5.0
```

```
print((a**2 + b**2)**0.5)       # 5.0
```

```
# Площадь круга
```

```
r = 1
```

```
print(math.pi * r**2)          # 3.141592653589793
```

```
# Вычисление площади треугольника по сторонам и углу между ними
```

```
a, b, alpha = 3, 4, 90
```

```
print(a * b * math.sin(math.radians(alpha))) # 5.0
```

```
# Округление чисел
```

```
print(round(-4/3, 2), round(4/3, 2))          # -1.33, 1.33
```

```
print(round(-4/3), round(4/3))              # -1, 1
```

```
print(int(-4/3), int(4/3))                  # -1, 1
```

```
print(math.ceil(-4/3), math.ceil(4/3))      # -1, 2
```

```
print(math.floor(-4/3), math.floor(4/3))    # -2, 1
```

```
print(math.trunc(-4/3), math.trunc(4/3))    # -1, 1
```

```
print(math.isclose(1/3, 0.333, abs_tol=0.001)) # True
```

```
print(math.isclose(1/3, 0.333, abs_tol=0.0001)) # False
```

```
# Степень и логарифмы
```

```
print(math.log(math.exp(2)))                # 2.0
```

```
print(math.factorial(5))                    # 120
```

```
>>> import math
>>> math.factorial(5)
```

$$\frac{n!}{k!(n - k)!}$$

```
n = 5
k = 3
```

```
import math
math.factorial(n) / (math.factorial(k) * math.factorial(n - k))
```

```
from math import factorial

factorial(n) / (factorial(k) * factorial(n - k))
```

```
from math import factorial as fact

fact(n) / (fact(k) * fact(n - k))
```


Упражнение

Напишите программу для вычисления функции:

$$f = \begin{cases} \sin x & \text{при } 0,2 \leq x \leq 0,9, \\ 1 & \text{в противном случае.} \end{cases}$$



**«Batteries included»
(батарейки в комплекте)**

Встроенные модули Python

<https://docs.python.org/3/py-modindex.html>

Например, [модуль os](#)

Философия Python

>>> **import this**

[Пояснения к философии](#)

Красивое лучше, чем уродливое.

Явное лучше, чем неявное.

Простое лучше, чем сложное.

Сложное лучше, чем запутанное.

Плоское лучше, чем вложенное.

Разреженное лучше, чем плотное.

Читаемость имеет значение.

Особые случаи не настолько особые, чтобы нарушать правила.

При этом практичность важнее безупречности.

Ошибки никогда не должны замалчиваться.

Если не замалчиваются явно.

Встретив двусмысленность, отбрось искушение угадать.

Должен существовать один — и, желательно, только один — очевидный способ сделать это.

Хотя он поначалу может быть и не очевиден, если вы не голландец [Шутливый намёк на разработчика языка].

Сейчас лучше, чем никогда.

Хотя никогда зачастую лучше, чем прямо сейчас.

Если реализацию сложно объяснить — идея плоха.

Если реализацию легко объяснить — идея, возможно, хороша.

Пространства имён — отличная штука! Будем делать их побольше!

Обзор сторонних модулей Python

<https://pypi.org/>

установите модуль **requests**

`!pip3 install <имя модуля>`

для проверки выполните:

`import requests`



<https://dfedorov.spb.ru>