

# Moduły

## 1. Wprowadzenie

Obsługa modułów składa się z dwóch różnych kwestii:

- Użycie już istniejącego modułu, napisanego przez kogoś innego lub stworzonego przez siebie (użytkownik modułu)
- Stworzenie modułu, bądź na własny użytek, bądź aby ułatwić życie innym programistom (dostawca modułu)

Moduł jest identyfikowany przez swą nazwę. Aby użyć dowolnego modułu, należy znać jego nazwę. Pewna liczba modułów dostarczana jest razem z samym językiem Python. Wszystkie te moduły, wraz z wbudowanymi funkcjami, tworzą bibliotekę standardową języka Python <https://docs.python.org/3/library/>

Każdy moduł składa się z pewnych. Elementy te mogą być funkcjami, zmiennymi, stałymi, klasami i obiektami.

## 2. Importowanie modułu

Metoda 1 – import modułu (osobna przestrzeń nazw)

```
import math
print(math.pi)
print(math.sin(math.pi/4))
```

Metoda 2 – import wskazanych elementów

```
from math import sin, pi
print(sin(pi/4))
```

Metoda 3 – import wszystkich elementów z modułu (metoda agresywna mogąca doprowadzić do konfliktu nazw)

```
from module math import *
print(math(pi/2))
```

### ALIASY

Można zamienić słowo `math`, i wprowadzić własną nazwę (czasami jest wygodnie zamienić nazwę długiego modułu na krótszą)

```
import math as m
print(m.sin(m.pi/2))

from math import pi as PI, sin as sine
print(sine(PI/2))
```

### 3. Standardowe moduły

#### Funkcja dir

Funkcja `dir()` potrafi wyświetlić wszystkie nazwy dostarczane przez konkretny moduł (moduł należy wcześniej zaimportować). Funkcja zwraca **posortowaną alfabetycznie listę**, która zawiera wszystkie nazwy elementów dostępnych w module. Nazwę modułu przekazujemy jako argument do funkcji `dir`

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

**Przykłady** wyświetlające zawartość modułu:

```
# Przykład 1
import math
for name in dir(math):
    print(name, end="\t")

# Przykład 2
import random
for ele in dir(random):
    if ele[0] != "_":
        print(ele)
```

#### Moduł math

Moduł `math` to zestaw funkcji trygonometrycznych, logarytmicznych oraz funkcji o funkcji ogólnego przeznaczenia jak np.

**Przykłady z modułu math**

```
from math import ceil, floor, trunc

x = 1.4
y = 2.6

print(floor(x), floor(y))
print(floor(-x), floor(-y))

print(ceil(x), ceil(y))
print(ceil(-x), ceil(-y))

print(trunc(x), trunc(y))
print(trunc(-x), trunc(-y))
```

#### Moduł random

Funkcja **`random()`** zwraca liczbę rzeczywistą `x` z zakresu `<0.0, 1.0)`

Następujące funkcje zwracają losowe wartości całkowite

```
randrange(start)
randrange(start, stop)
randrange(start, stop, krok)
randint(start, stop)
```

## Przykłady 1

```
from random import randrange, randint

lst=[]
for i in range(1000):
    lst.append(randrange(10))
print(set(lst))

lst=[]
for i in range(1000):
    lst.append(randrange(4,10))
print(set(lst))

lst=[]
for i in range(1000):
    lst.append(randint(0,10))
print(set(lst))
```

## Przykłady 2

```
from random import choice, sample

lst = [i for i in range(21)]

print(choice(lst))
print(sample(lst, 5))
print(sample(lst, 10))
```

## Moduł platform

```
from platform import platform, machine, processor, system, version
print(platform())
print(platform(1))
print(platform(0, 1))
print(machine())
print(processor())
print(system())
print(version())

from platform import python_implementation, python_version_tuple
print(python_implementation())
print(python_version_tuple())
```

## INNE

[Python Module Index – Python 3.11.1 documentation](#)