



Moduły w Pythonie

M@я3k Pүđ€£kØ

Programowanie w Pythonie

Spis treści

- Definicja modułu.
- Importowanie modułu
- Moduły wbudowane
- Najpopularniejsze moduły wbudowane
- Pakiet
- 1
- a
- a
- a

Moduł w pythonie

- Moduł to plik **.py**, który zawiera kod Pythona: funkcje, klasy, zmienne lub stałe.
- Polega na wydzieleniu fragmentu kodu do oddzielnego pliku.
 - Charakteryzuje się jasną i przejrzystą strukturą przy dużych możliwościach i elastyczności.
- Istnieją moduły wbudowane.
- Można też tworzyć własne.

```
import math
x = float (input ("Podaj liczbę: "))
p = math.sqrt (x)
print (f"pierwiastek z {x} wynosi {p}")
```

Importowanie modułu

- Korzystanie z wybranego modułu odbywa się po uprzednim zaimportowaniu go. Można to zrobić na kilka sposobów. Najprostsza instrukcja to **import nazwa_modulu**.

```
import math                                #import modułu
print(f'Pierwiastek z liczby 2 wynosi {math.sqrt(2)}')
```

- Wygodny jest zapis **import nazwa_modulu as skrot**, dzięki któremu można wywołać funkcje z tego modułu z użyciem skróconej formy:

```
import math as m                          #import modułu ze skrótem
print(f'Pierwiastek z liczby 2 wynosi {m.sqrt(2)}')
```

- Jeśli nie trzeba korzystać z całego modułu, a jedynie z wybranych jego funkcji to można użyć zapisu **from modul import funkcja1, funkcja2**:

```
from math import sqrt                     #import funkcji sqrt
print(f'Pierwiastek z liczby 2 wynosi {sqrt(2)}')
```

- Importuje się wszystko z modułu math. Do funkcji odwołuje się podając tylko jej nazwę.

```
from math import *                       #import wszystkiego z modułu math
print(f'Pierwiastek z liczby 2 wynosi {sqrt(2)}')
```

- Można zaimportować kilka modułów w jednej linii:

```
import math, os, time #import wielu modułów oddzielonych przecinkiem
print(f'Pierwiastek z liczby 2 wynosi {sqrt(2)}')
```

Moduły wbudowane

- Biblioteka standardowa Pythona jest bardzo obszerna i zawiera ponad 200 modułów, które są instalowane automatycznie razem z językiem Python.
- Pełną i aktualną listę wszystkich modułów można znaleźć w oficjalnej dokumentacji Pythona (Python Module Index).
- Pełna lista
<https://docs.python.org/pl/3/library/index.html>

Najpopularniejsze moduły wbudowane 1

Obsługa systemu i środowiska	
os	Interfejs do funkcji systemu operacyjnego (pliki, katalogi)
sys	Parametry i funkcje specyficzne dla interpretera (argumenty wiersza poleceń)
platform	Dostęp do danych platformy (system operacyjny, wersja Pythona)
shutil	Operacje wysokiego poziomu na plikach (kopiowanie, usuwanie)
pathlib	Zorientowana obiektowo obsługa ścieżek plików
Typy danych i algorytmy	
collections	Kontenery danych (Counter, defaultdict, namedtuple)
heapq	Algorytmy kolejek priorytetowych
enum	Obsługa typów wyliczeniowych
Kompresja i archiwizacja	
zipfile	Praca z plikami ZIP
gzip	Obsługa kompresji gzip
tarfile	Praca z archiwami tar

Najpopularniejsze moduły wbudowane 2

Matematyka i liczby losowe	
math	Funkcje matematyczne (sin, cos, log)
random	Generowanie liczb losowych
statistics	Funkcje statystyczne (średnia, mediana)
decimal	Precyzyjne obliczenia dziesiętne
secrets	Generowanie bezpiecznych losowych liczb (do kryptografii).
Sieć i internet	
socket	Niskopoziomowa obsługa sieci
http	Moduły do obsługi protokołu HTTP (klient/serwer)
urllib	Obsługa adresów URL (zapytania HTTP - request, parse, error)
smtplib	Wysyłanie wiadomości e-mail
uuid	Generowanie unikalnych identyfikatorów (RFC 9562)
Programowanie współbieżne i asynchroniczne	
asyncio	Asynchroniczne I/O, pętla zdarzeń, współprogramy
threading	Zarządzanie wątkami
multiprocessing	Przetwarzanie wieloprocessorowe

Najpopularniejsze moduły wbudowane 3

Czas i daty	
datetime	Manipulacje datą i czasem
time	Dostęp do funkcji czasu systemu
calendar	Funkcje związane z kalendarzem
Obsługa formatów plików i danych	
json	Kodowanie i dekodowanie formatu JSON
csv	Odczyt i zapis plików CSV
xml	Przetwarzanie plików XML
configparser	Obsługa plików konfiguracyjnych (INI)
sqlite3	Obsługa baz danych SQL
pickle	Serializacja i deserializacja obiektów Pythona
re	Wyrażenia regularne (Regular Expressions)
Inne moduły	
tkinter	Standardowa biblioteka GUI (interfejs graficzny)

Pakiet

- Pakiet to katalog zawierający wiele modułów. Z reguły są one tematycznie powiązane, co ułatwia organizację kodu i projektu.
- Pakiet to katalog, który zawiera wiele modułów (plików **.py**) i plik **__init__.py**. To właśnie ten plik sprawia, że Python rozpoznaje katalog jako pakiet. Plik **__init__.py** może być pusty – ale nie musi. Definiuje się w nim, które elementy pakietu mają być widoczne po imporcie.

- **Instalacja pakietów w Pythonie**

- instalacja popularnej biblioteki do tworzenia wykresów `matplotlib` wygląda następująco:

```
pip install matplotlib
```

- **Odinstalowanie pakietów w Pythonie**

- Odinstalowanie pakietu dokonujemy również przy użyciu polecenia `pip`:

```
pip uninstall matplotlib
```

Lista wszystkich bibliotek Pythona, zainstalowanych w systemie

- By zobaczyć listę wszystkich bibliotek Pythona, które zainstalowano ręcznie:
- Wiersz poleceń -> Python -> `help ('modules')`
- Lista

Tworzenie własnych modułów

- Moduł tworzy się jako oddzielny plik **.py**, w którym definiowane będą funkcje, klasy i zmienne.
- Poniższą funkcję **modul_liczby** można zapisać w pliku **modul.py**.

```
def modul_liczby(a):  
    if a < 0:  
        a = -a  
    else:  
        a = a  
    return a
```

```
a = float(input("Podaj liczbę: "))  
print(modul_liczby(a))
```

- Wtedy program główny będzie miał postać:

```
Import modul  
a = float(input("Podaj liczbę: "))  
print(modul.modul_liczby(a))
```

- Utworzony moduł możemy wykorzystać w innych plikach, po wcześniejszym zaimportowaniu go. Możemy zaimportować cały moduł korzystając z instrukcji **import modul**, bądź tylko wybrane funkcje **from modul import modul_liczby**.

```
from modul import modul_liczby  
a = float(input("Podaj liczbę: "))  
print(modul_liczby(a))
```

Ćwiczenie

1. Napisz program wczytujący liczbę n i wyliczający dla niej **silnię** $n!$ (mnożenie liczb od 1 do n) i **sumę** liczb od 1 do n . Każdą z nich zapisz w oddzielnej funkcji.
 - a) Powyższe funkcje zapisz w module **matma.py**.
 - b) W programie głównym wprowadź import tego modułu.
2. Napisz program wczytujący 10 liczb i obliczający w oddzielnych funkcjach różne informacje o tych liczbach.
 - a) Suma, średnia arytmetyczna, element maksymalny, element minimalny, różnicę między największym i najmniejszym elementem, liczbę liczb parzystych, liczbę liczb nieparzystych.
 - b) Powyższe funkcje zapisz w module **arytmetyka.py**.
 - c) W programie głównym wprowadź import tego modułu.
3. Napisz program wczytujący dwie liczby a i b , a potem realizujący na nich obliczeń. Każda z nich ma być realizowana w oddzielnej funkcji:
 - a) Suma, różnica, mnożenie, dzielenie, dzielenie modulo, potęgowanie a^b
 - b) Powyższe funkcje zapisz w module **matematyka.py**.
 - c) W programie głównym wprowadź import tego modułu.

Powtórzenie

- Jak podzielić kod programu na kilka plików
- Jak zaimportować moduł do programu?
- Omów często używane moduły wbudowane.
- Co to jest pakiet?
- Jak utworzyć własny moduł?