

Organizacja i import własnych modułów w Pythonie

1. Co to jest moduł?

Moduł w Pythonie to plik z rozszerzeniem .py zawierający definicje zmiennych, funkcji lub klas. Pozwala na logiczne organizowanie kodu i ponowne jego wykorzystanie w różnych projektach.

2. Gdzie umieszczać własne moduły?

- W tym samym katalogu co plik główny skryptu
- W podkatalogu (np. „moduly/”), który jest częścią projektu
- W lokalnej paczce (folderze z plikiem __init__.py)
- W katalogach dodanych do zmiennej środowiskowej PYTHONPATH

3. Jak importować moduły?

Plik funkcje.py w tym samym katalogu:

```
import funkcje
funkcje.nazwa_funkcji()
```

Import konkretnej funkcji:

```
from funkcje import nazwa_funkcji
nazwa_funkcji()
```

Import z aliasem:

```
import funkcje as f
f.nazwa_funkcji()
```

4. Przykładowa struktura projektu

```
projekt/
|
├─ main.py
├─ funkcje.py
└─ narzedzia/
    ├── __init__.py
    └─ pomoc.py
```

5. Import z podfolderu (pakietu)

W pliku main.py:

```
from narzedzia import pomoc
pomoc.funkcja()
```

Lub:

```
from narzedzia.pomoc import funkcja
funkcja()
```

6. Rozszerzanie ścieżki importu – PYTHONPATH i sys.path

```
# Tymczasowe dodanie ścieżki w kodzie:  
import sys  
sys.path.append("C:/moja_sciezka/do_modulow")  
import moj_modul
```

7. Dobre praktyki organizowania kodu modułowego

- Unikaj nadpisywania nazw wbudowanych (np. math.py, os.py)
- Stosuj nazwy opisowe dla modułów
- Każdy moduł niech odpowiada za konkretną funkcjonalność
- Dodawaj plik __init__.py do folderów z modułami - nawet pusty
- Utrzymuj spójność nazewnictwa i struktury