

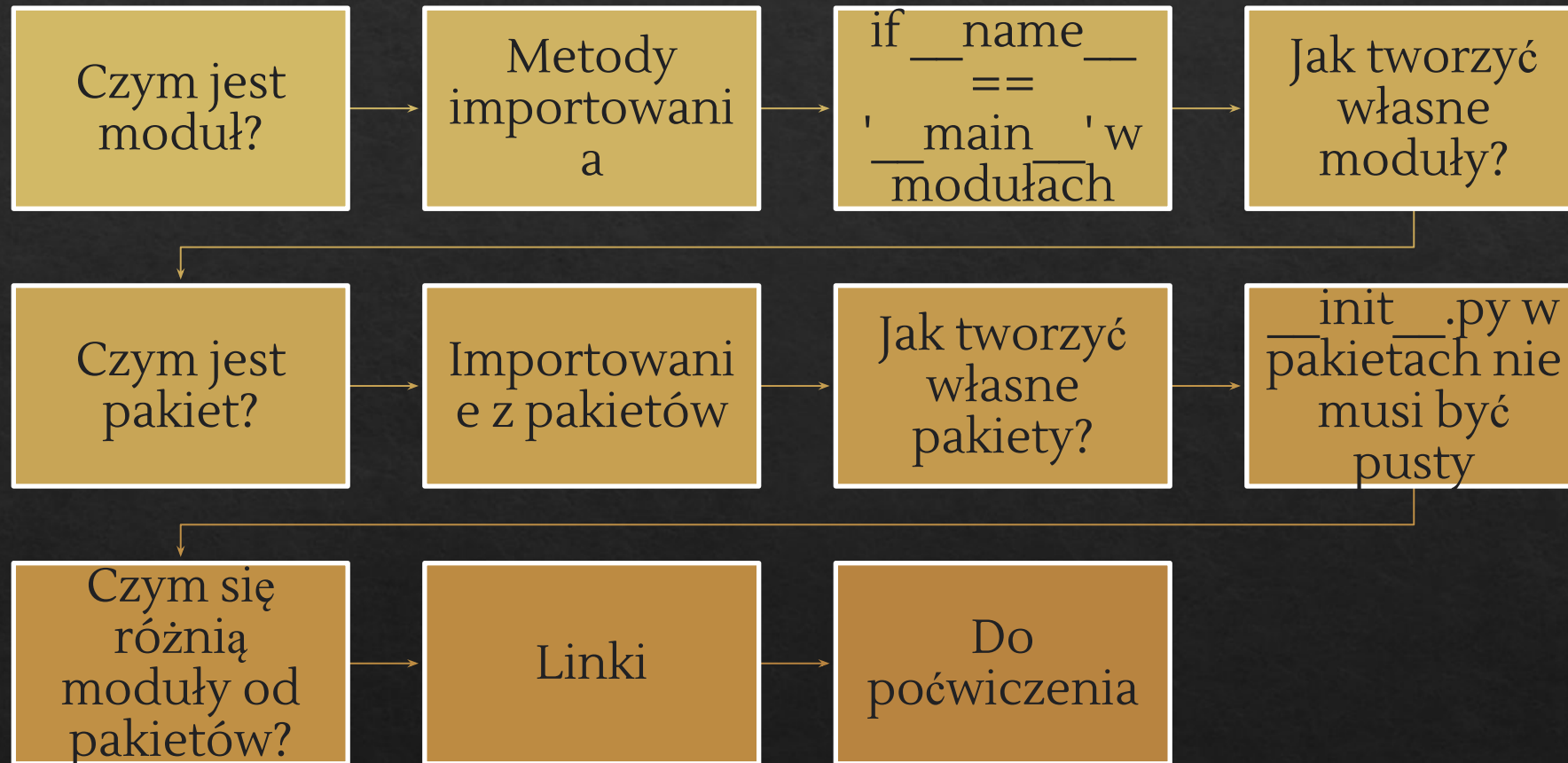
Python - moduły i pakiety

Marta Potocka

PyLight - Warszawa, 18.10.2018



Spis treści



Czym jest moduł?

- ♦ każdy plik z rozszerzeniem `.py`
- ♦ moduł po załadowaniu zachowuje się jak część pliku, do którego go zaimportowaliśmy
- ♦ importowany moduł może sam importować kolejne moduły

Metody importowania

1. `import module_name` importujemy moduł, aby użyć znajdujących się w nim funkcji używamy `module_name.function_name()`
importując moduł nie dodajemy rozszerzenia .py !
2. `from module_name import *` - importujemy wszystkie funkcje z danego modułu, możemy używać ich bezpośrednio
3. `from module_name import function_one, function_two` - importujemy wybrane funkcje z modułu, możemy używać tylko tych, które zaimportowaliśmy

if `__name__` == `'__main__'` w modułach

- ♦ `__name__` - specjalna zmienna, która wyświetla nazwę danego modułu
jeśli moduł został uruchomiony bezpośrednio, wtedy `__name__` == `'__main__'`
jeśli moduł został zaimportowany, wtedy pod `__name__` kryje się jego nazwa
- ♦ sprawdzenie `if __name__ == '__main__'` pozwala nam wydzielić blok kodu, który będzie się wykonywał tylko przy bezpośrednim uruchomieniu danego pliku
- ♦ jeśli taki plik zostanie zaimportowany to wykona się tylko jego część poza blokiem `if __name__ == '__main__'`

Jak tworzyć własne moduły?

tworzymy plik `mój_moduł.py`

jeśli chcemy, aby część kodu wykonywała się tylko przy bezpośrednim uruchomieniu pliku używamy `if __name__ == '__main__':`

gotowe!

Czym jest pakiet?

pakiet jest folderem, który zawiera w sobie jeden lub więcej modułów oraz plik `__init__.py`



plik `__init__.py` jest konieczny, aby poinformować Pythona, że dany folder jest pakietem

Importowanie z pakietów

```
import package_name.module_name
```

```
package_name.module_name.function_name()
```

```
import package_name.module_name  
as my_module
```

```
my_module.function_name()
```

```
from package_name import  
module_name
```

```
module_name.function_name()
```


Jak tworzyć własne pakiety?

- ◆ utwórz folder
- ◆ dodaj do niego przynajmniej jeden plik z rozszerzeniem `.py` - będący modułem
- ◆ dodaj do niego pusty plik `__init__.py`
- ◆ Gotowe!

`__init__` w pakietach nie musi być pusty

- ◆ `__init__.py` może być pusty i w tej roli służy jedynie do poinformowania Pythona, że dany folder jest pakietem
- ◆ `__init__.py` może również zawierać wewnętrzną konfigurację, np.:
 - ? `from .moduł import funkcja`
 - ? `from pakiet.moduł import funkcja`
 - ? (konieczna kropka lub nazwa pakietu, ponieważ Python będziesz szukał nazwy modułu w katalogu, z którego uruchamiamy skrypt!)
 - ? parametr `__all__ = ['moduł1', 'moduł2']` - pozwala podać listę modułów, które mają być automatycznie załadowane przy `from pakiet import *`

Czym się różnią moduły od pakietów?

- ♦ moduł jest pojedynczym plikiem z rozszerzeniem `.py`
- ♦ pakiet jest folderem, zawierającym jeden lub więcej modułów oraz plik `__init__.py`
- ♦ pakiet jest strukturą nadrzędną, pozwalającą logicznie zorganizować grupę modułów, np:
- ♦ matematyka - **pakiet**
 - ? algebra - **moduł**
 - ♦ odwróć_macierz - **funkcja**
 - ? analiza - **moduł**
 - ♦ policz_całkę - **funkcja**
 - ? geometria - **moduł**
 - ♦ sprawdź_relację_okręgów - **funkcja**

Linki

- ◊ <https://www.programiz.com/python-programming/modules> (eng)
- ◊ <https://www.programiz.com/python-programming/package> (eng)
- ◊ https://www.learnpython.org/pl/Moduly_i_pakiety (pl)

