

Python!

Lekcja 6 – parametryzacja i moduły

- Moduł, importowanie modułów
- Metody importowania
- Przydatne moduły
- PYTHONPATH
- Parametryzacja programów
- Nasz pierwszy moduł



Moduł – plik z rozszerzeniem `‘.py’`, zawierający definicje i wyrażenia.

Moduły mogą być importowane, aby korzystać z zawartych w nich funkcjonalności. Można myśleć o nich jak o bibliotekach.

Wiele modułów może wspólnie dostarczających konkretnego rozwiązania mogą zostać zebrane w *paczkę*.

Paczki można pobrać i wgrać ręcznie lub użyć programu pip.

Paczki można przeglądać tutaj:
<https://pypi.python.org/pypi/>

Python posiada wiele wbudowanych modułów, wiele z nich wspomagających zarządzanie systemem. Krótka lista modułów przydatnych przy pisaniu skryptów systemowych:

- `re` – dla operacji z użyciem wyrażeń regularnych
- `shutil` – operacje na plikach
- `csv` – zestaw narzędzi do manipulowania plikami CSV
- `os` – obsługa interfejsu systemu operacyjnego
- `logging` – tworzenie logów
- `json` – operacje na danych w formacie JSON
- `sys` – funkcje typowe dla systemu

Python zezwala na wiele metod importowania modułów lub ich fragmentów

- `import <module>` - importuje cały moduł
- `from <module> import <something>` - importuje funkcję lub klasę z modułu
- `import <module> as <module_renamed>` - importuje moduł i zmienia sposób odwołania

```
>>> import sys
>>> sys.version
'3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)]'
>>> import sys as technikum
>>> technikum.version
'3.5.2 (v3.5.2:4def2a2901a5, Jun 25 2016, 22:01:18) [MSC v.1900 32 bit (Intel)]'
```

PYTHONPATH – zmienna środowiskowa definiująca, gdzie Python będzie szukać modułów do importowania (domyślnie będzie szukać również w katalogu skryptu)

```
>>> import sys
>>> print(sys.path)
['',
 'C:\\Users\\rszulist\\AppData\\Local\\Programs\\Python\\Python35-32\\Lib\\idlelib',
 'C:\\Users\\rszulist\\AppData\\Local\\Programs\\Python\\Python35-32\\python35.zip',
 'C:\\Users\\rszulist\\AppData\\Local\\Programs\\Python\\Python35-32\\DLLs',
 'C:\\Users\\rszulist\\AppData\\Local\\Programs\\Python\\Python35-32\\lib',
 'C:\\Users\\rszulist\\AppData\\Local\\Programs\\Python\\Python35-32',
 'C:\\Users\\rszulist\\AppData\\Local\\Programs\\Python\\Python35-32\\lib\\site-
packages']
>>>
```

Programy mogą być parametryzowane, podobnie jak funkcje.

Do parametryzacji można użyć jednego z dwóch modułów:

- `sys` – dla parametrów wiersza poleceń
\$ `python test.py arg1 arg2 arg3`
- `getopt` – dla ‘długich’ opcji w skrypcie. Parsuje parametry wiersza poleceń w poszukiwaniu opcji i wartości. Składnia zaczerpnięta z Unixowego `getopts`
\$ `python test.py -h`

Czas na ćwiczenia ;)

Dziękuję za uwagę!

