

PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 13



AGENDA

DAY 13

- testowanie kodu, cd
- virtual env, pip
- praca z plikami
- modyfikowanie plików (zdjęcia)

| virtualenv

virtualenv

Wirtualne środowisko Python – kopia Pythona, ze specyficznymi ustawieniami, zainstalowanymi modułami itp.

Dzięki virtualenv możemy mieć środowiska z różnymi wersjami tych samych modułów.

pip install virtualenv

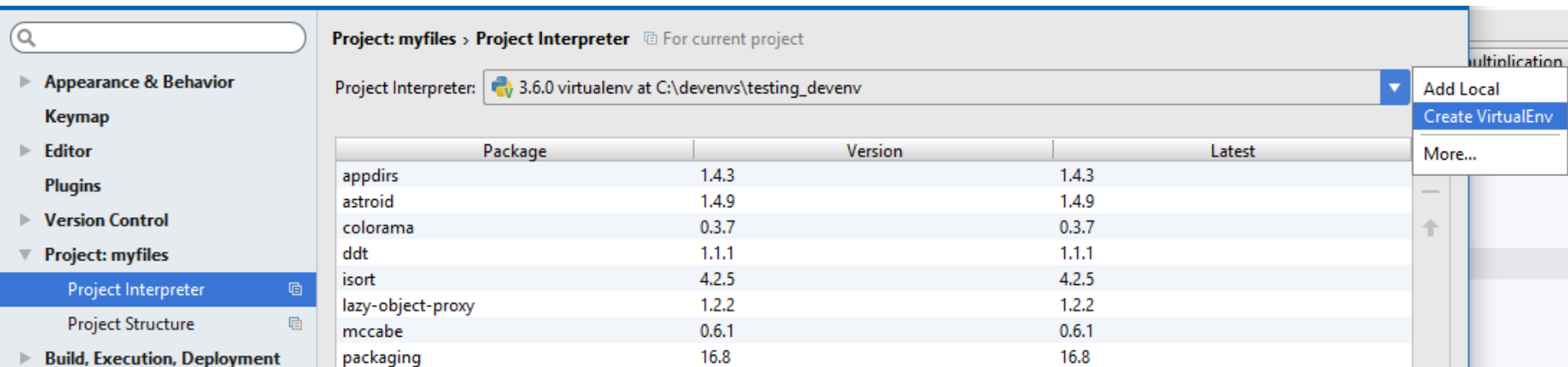
virtualenv mytestingenv – tworzymy virtualenv

mytestingenv/Scripts/activate – aktywujemy

(mytestingenv/**bin**/activate - Mac i Linux)

virtualenv w PyCharm

file -> settings -> project settings -> project interpreter



The screenshot shows the PyCharm interface with the 'Project Interpreter' settings window open. The left sidebar contains a search bar and a list of settings categories: Appearance & Behavior, Keymap, Editor, Plugins, Version Control, and Project: myfiles. The 'Project: myfiles' section is expanded, showing 'Project Interpreter' as the selected option. The main panel displays the 'Project Interpreter' settings for the current project. It shows the current interpreter as '3.6.0 virtualenv at C:\devenvs\testing_devenv'. Below this, a table lists installed packages and their versions. A context menu is open on the right side of the table, showing options: 'Add Local', 'Create VirtualEnv', and 'More...'. The table has three columns: 'Package', 'Version', and 'Latest'.

Package	Version	Latest
appdirs	1.4.3	1.4.3
astroid	1.4.9	1.4.9
colorama	0.3.7	0.3.7
ddt	1.1.1	1.1.1
isort	4.2.5	4.2.5
lazy-object-proxy	1.2.2	1.2.2
mccabe	0.6.1	0.6.1
packaging	16.8	16.8

| venv & pip

PyPI Python Package Index

lista dostępnych pakietów

pypi.python.org/pypi

pip

Menadżer pakietów instalowany razem z Python.
Komendy w wierszu poleceń:

pip help – ogólna pomoc

pip help install – pomoc dot. polecenia

pip list – lista zainstalowanych pakietów

pip search – szuka pakietów w repozytorium online

pip install *pakiet* – instalowanie modułu

pip uninstall *pakiet* - odinstalowanie

pip list -o -sprawdzenie nieaktualnych pakietów

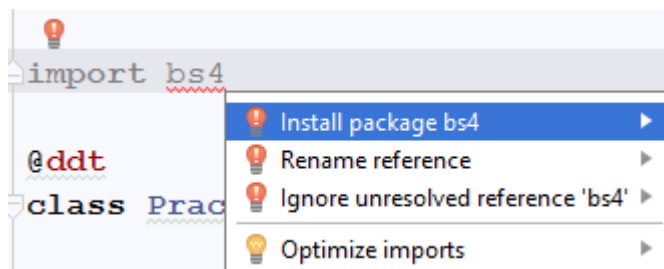
pip install -U *pakiet* - update pakietu

pip freeze > plik.txt – zapisanie informacji do pliku o pakietach

pip install -r plik.txt – zainstaluje wszystkie wymagane pakiety

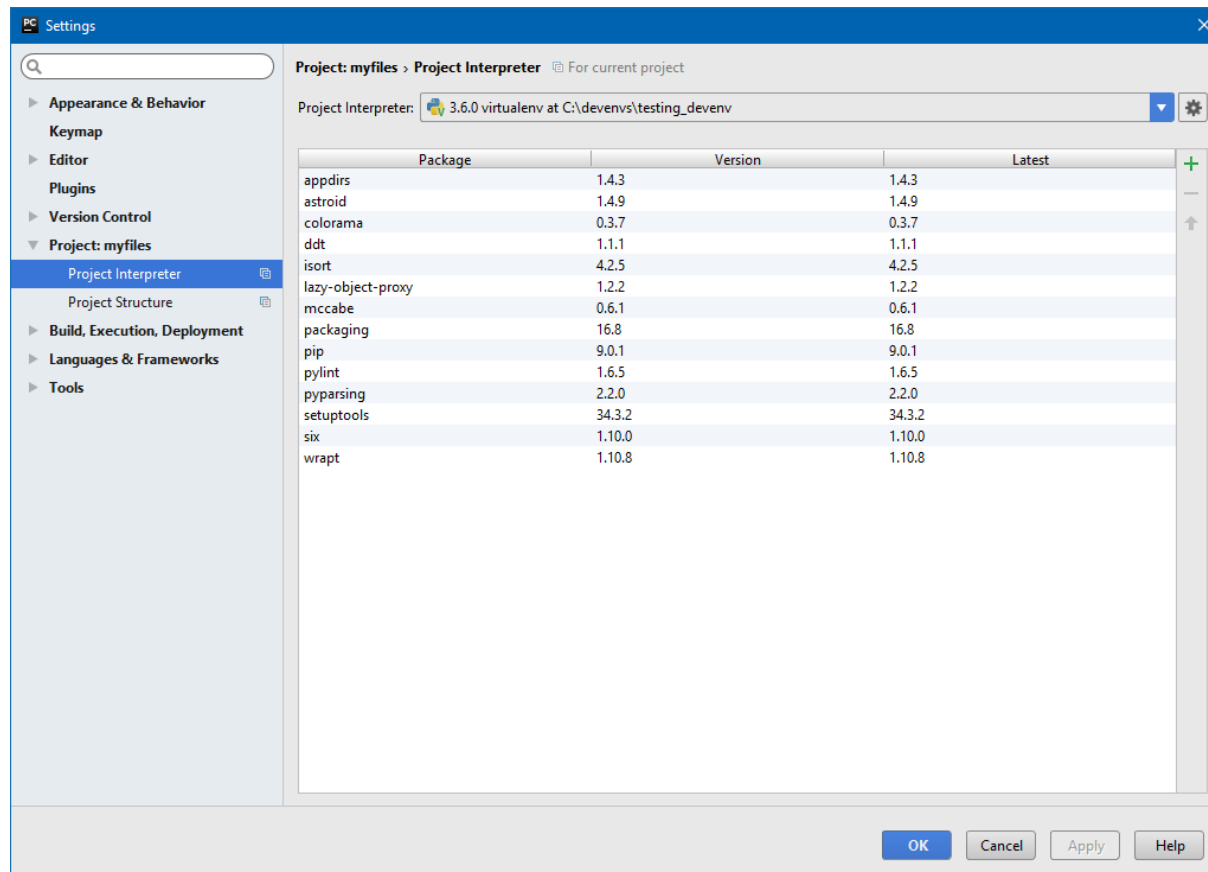
INSTALOWANIE PAKIETÓW W PYCHARM

- podpowiedzi przy pisaniu kodu – alt + enter



INSTALOWANIE PAKIETÓW W PYCHARM

file ->
settings ->
project ->
project interpreter



| praca z plikami

OS

- moduł os służy do pracy z plikami, ścieżkami, zmiennymi systemowymi

mkdir, chdir, getcwd, unlink, rmdir, listdir, walk

- os.path – działania na ścieżkach
split, join, abspath

<https://docs.python.org/3/library/os.html>

<https://docs.python.org/3/library/os.path.html>

shutil

wrapper dla poleceń systemowych, w pewnych sytuacjach ułatwia wykonanie poleceń

copytree, move, rmtree,

<https://docs.python.org/3/library/shutil.html>

send2trash

przenosi do kosza

```
send2trash.send2trash(plik)
```

argumenty programów

```
python plik.py
```

```
python plik.py opcja1 opcja2 opcja3
```

```
import sys
```

sys.argv – lista argumentów przekazanych do Python

```
['plik.py', 'opcja1', 'opcja2', 'opcja3']
```

sys.argv[1:] – lista samych opcji

PIL – pillow działania na zdjęciach

<https://pillow.readthedocs.io/en/4.0.x/>

<https://pillow.readthedocs.io/en/4.0.x/reference/Image.html>

```
pip install Pillow
```

```
from PIL import Image
```

```
foto = Image.open(plik)
```

```
copy, crop, filter, paste, resize, rotate, save
```




Thanks!!