PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 7

AGENDA
DAY 7

- instalowanie i importowanie modułów
- praca z plikami
- pliki csv (wartości oddzielana przecinkami)
- with
- pickle

1.
import

ktoś już wykonał za nas pracę

IMPORT

```
import moduł, moduł2
from moduł import funkcja1, funkcja2
from moduł import *
```

```
string, datetime, copy, math, decimal,
random, os, csv, antigravity
```

FOLDER ROBOCZY

```
import sys
import os

print("ścieżki wyszukiwania Python:", sys.path)
print("aktualny folder roboczy:", os.getcwd())
```

Pamietać – PyCharm tworzy własne środowisko uruchomieniowe – dodaje do folderów wyszukiwania (sys.path) folder główny projektu, dlatego wskazujemy relatywną do gł. folderu ścieżkę (day6.fun7).

ścieżki wyszukiwania (sys.path) będą inne jeśli plik z pow. kodem uruchomimy:

a) w PyCharm, oraz
b) bezpośrednio w konsoli

2. PyPI & pip

Menadżer pakietów Python

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

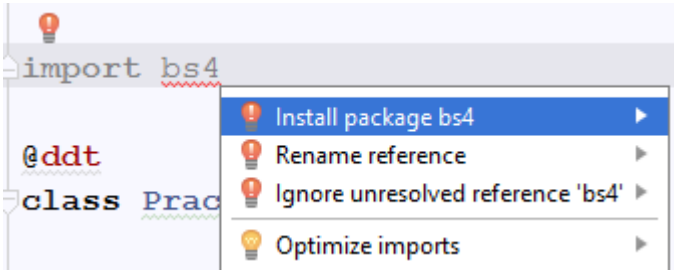
pip install *pakiet* – instalowanie modułu
pip uninstall *pakiet* - odinstalowanie

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

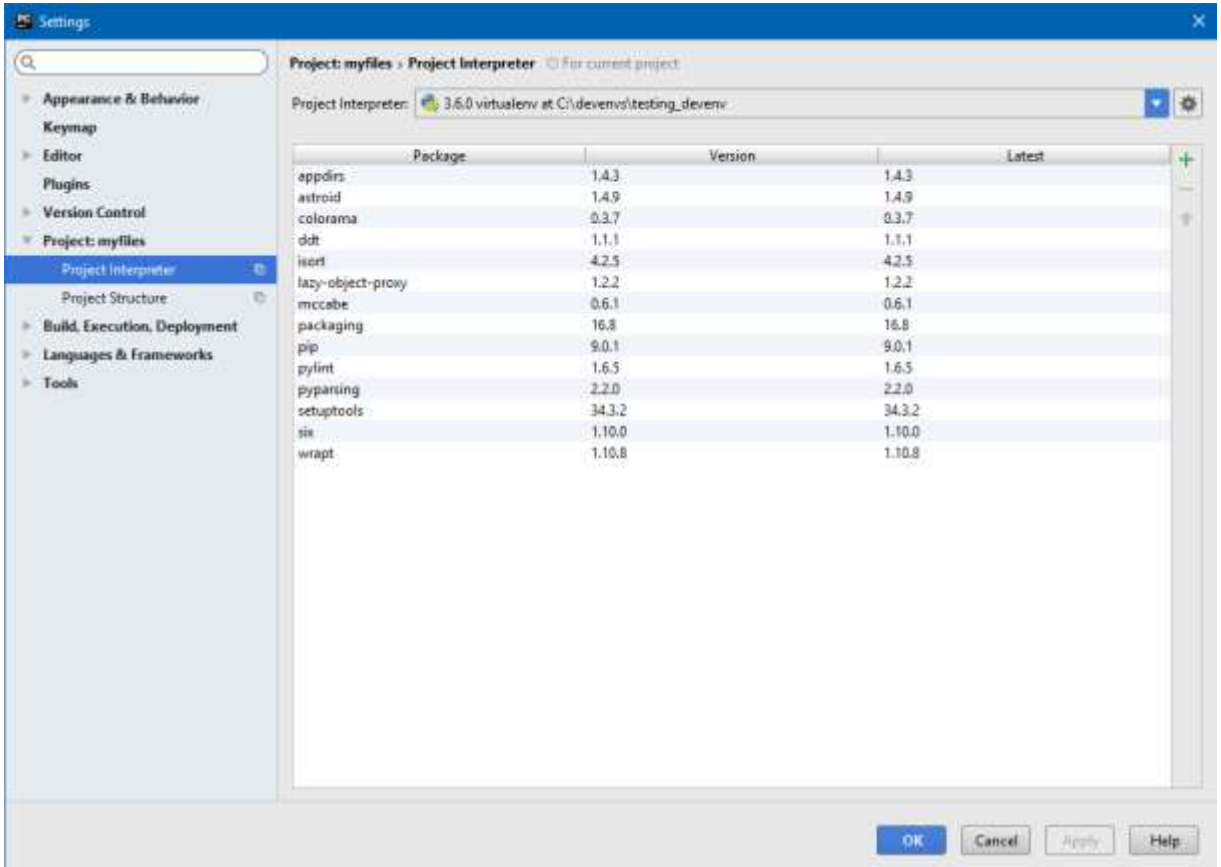
INSTALOWANIE PAKIETÓW W PYCHARM

- podpowiedzi przy pisaniu kodu – alt + enter



INSTALOWANIE PAKIETÓW W PYCHARM

file ->
settings ->
project ->
project interpreter



2.
Praca z plikami

pliki tekstowe

otwieramy plik

```
plik = open("scieżka_do_pliku", tryb)
```

tryby:
r – tylko do odczytu
w – zapisywanie pliku (stary plik o tej samej nazwie będzie usunięty)
r+ - do odczytu i zapisu
a – dopisywanie do pliku (dane są dopisane do końca istniejącego pliku)

pliki tekstowe

plik.read() – odczytanie całego pliku, zwracany jest string zawierający cały tekst pliku (włącznie ze znakami \n) – opc. argument – **int** określająca ilość bajtów do wczytania

plik.readline() – odczytanie jednej linii z pliku, zwracany jest string z linijką testu, włącznie ze znakiem \n

plik.readlines() – odczytuje cały tekst – zwraca listę stringów - linijek

```
for line in plik:
    print(line, end='')
```

with

pliki należy zamykac po użyciu:

```
plik = open(„plik.txt”)
    # kod
plik.close()
```

otwarcie pliku za pomocą with pozwala na automatyczne zamykanie pliku przez Pythona

```
with open(„plik.txt”) as  plik:
    print(plik.readline())
```

pliki tekstowe

plik.write(string) – zapisuje string do pliku w obecnej pozycji kursora, zwraca liczbę zapisanych znaków – należy pamiętać o znaku \n

plik.writelines(iterable) – zapisuje elementy z kolekcji jako poszczególne linie w pliku

Plik musi być otworzony w trybie do zapisu aby móc go zmieniać!

CSV

Pliki CSV – comma separated values – dane oddzielane przecinkami

```
Imie,Nazwisko,Adres,Telefon
Joanna,Kowalska,Gdansk Przytulna,64 654-65-45
Adam,Nowak,Gdynia Swietojanska,0700325487
```

Do obsługi plików CSV można użyć biblioteki csv

pickle

pickle to moduł służący do zapisywania obiektów do plików.

Zapisać (i odczytać) możemy każdy obiekt Python'a (listy z danymi, słowniki, klasy, instancje klas (żyjące obiekty) itd..

PICKLE UŻYCIE **(TRYB BINARNY!)**

```
import pickle

dane = ["Bartosz", "Mojo", 33]

with open("ogorek.pickle", "wb") as plik:
    pickle.dump(dane, plik)

# odczytanie
with open("ogorek.pickle", "rb") as plik:
    dane_wczytane = pickle.load(plik)

print(dane_wczytane)
```

Thanks!!