Build Automatizálás Python-ban

4. hét házi feladat – Fehér Péter

# Feladat ismertetése

Munkámból adódóan, sokat foglalkozom weboldalakon lévő adatok manipulálásával, statisztikához és egyéb feladatokhoz, ezért csináltam egy Python kódot, ami lekérdezi a Cubix oldalán lévő összes aktív Python képzést, kiírja a nevüket, és a hozzá tartozó linket.

A feladathoz a requests és a beautifulsoup könyvtárakat használtam.

A requests könyvtár segítségével http kéréseket tudunk küldeni, ezzel ellenőrizve hogy elérhető-e, a cél, a beautifulsoup pedig lehetővé teszi, hogy weboldalak forráskódjából adatokat emeljünk ki.



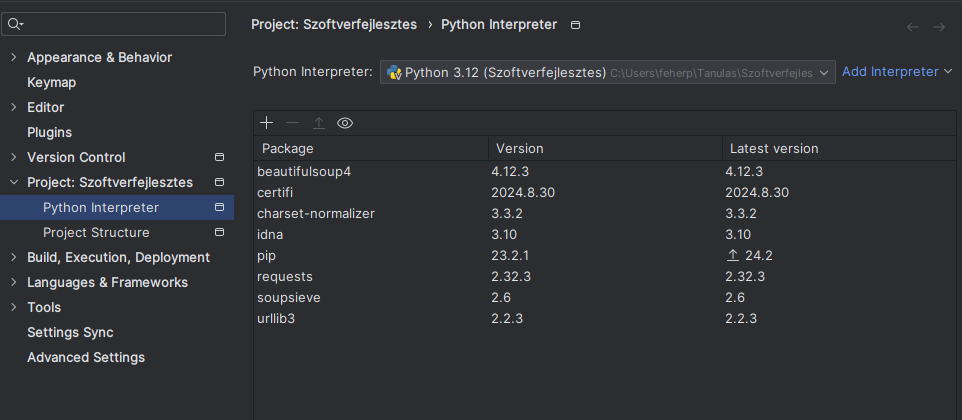
# Függőségek

## Kézi megvalósítás

A külső könyvtárak kézi telepítéséhez a terminálba írva

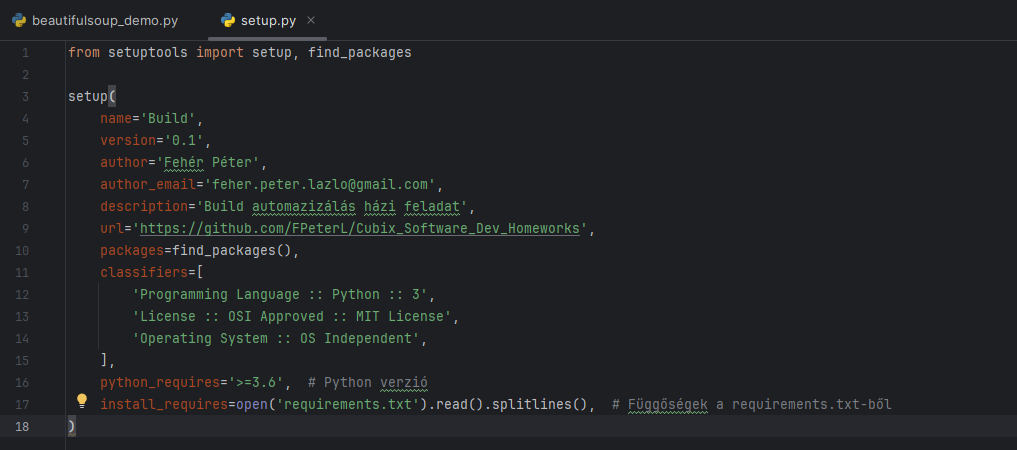
pip install requests  
pip install beautifulsoup4

parancsokat használtam.  
A File->Setting->Project: Projektnév -> Python Interpreter funkciót szoktam többnyire használni, itt látszik egyből a verzió, ha frissítés érhető el, illetve leírás, számomra kényelmesebb.



## Automatizált magvalósítás

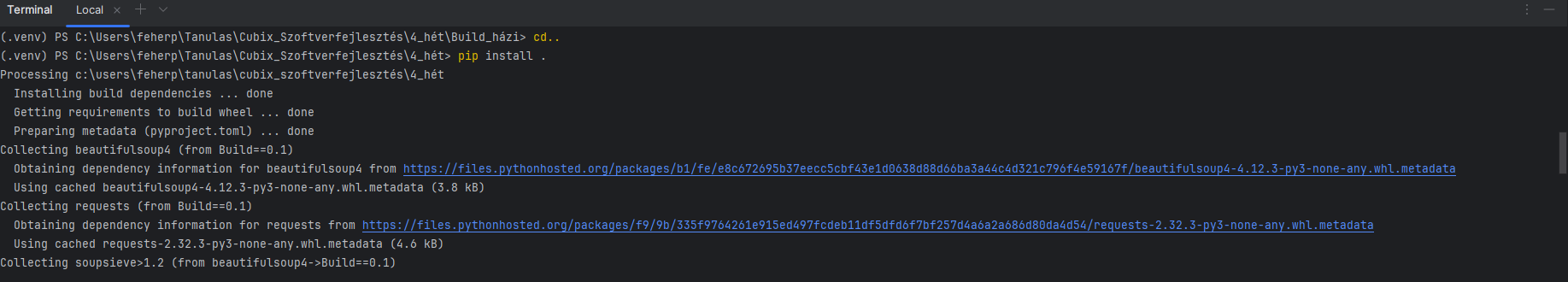
A külső könyvtárak automatizált kezeléséhez az interneten több megoldást láttam, például a **setup.py**-t:



Egy külön python script, illetve egy hozzá tartozó requriements.txt, melyből kiolvassa a telepítendő könyvtárakat. A konfiguráció metaadatokat tartalmaz a projektről, programról, fejlesztőről, illetve magára a pythonra vonatkozóan.

pip install .

paranccsal futtatható, mely kezeli a telepítéseket:



Alternatív megoldásként működhet egy sima szöveges file amibe felsoroljuk a telepítendő könyvtárakat, és a

pip install -r requirements.txt

telepíti őket. Ez a második megoldás, kicsi magán projektek esetén szerintem egyszerűbb, tekintve hogy a setup.py-nál leírások alapján figyelni jobban figyelni kell a mappa struktúrára.

# Elérhetőség

<https://github.com/FPeterL/Cubix_Software_Dev_Homeworks.git>

# Forráskód

## Setup.py

### Leírás

* **name**: A csomag neve, amely alatt a felhasználók telepíthetik.
* **version**: A csomag verziószáma, amely általában a szemantikus verziózás szabályai szerint van megadva.
* **author**: A csomag szerzőjének neve.
* **author\_email**: A szerző e-mail címe.
* **description**: A csomag rövid leírása, amely összefoglalja, hogy mi a csomag célja.
* **url**: A csomag weboldala vagy GitHub repository linkje.
* **packages**:A csomagolt modulok listája. A setuptools find\_packages() függvény az automatikusan megkeresi ezeket.
* **classifiers**: A csomaghoz tartozó kategóriák listája, amely segít másoknak megtalálni a csomagot. A Python Package Index (PyPI) számára releváns információkat tartalmaz.
* **python\_requires**: A Python verzióra vonatkozó követelmények. Meghatározza, hogy a csomag mely Python verzióval használható.
* **install\_requires**: A csomag függőségei, amelyek telepítésre kerülnek a pip által. Ezek a csomagok szükségesek a csomag működéséhez.

### Kód

from setuptools import setup, find\_packages  
  
setup(  
 name='Build',  
 version='0.1',  
 author='Fehér Péter',  
 author\_email='feher.peter.lazlo@gmail.com',  
 description='Build automazizálás házi feladat',  
 url='https://github.com/FPeterL/Cubix\_Software\_Dev\_Homeworks',  
 packages=find\_packages(),  
 classifiers=[  
 'Programming Language :: Python :: 3',  
 'License :: OSI Approved :: MIT License',  
 'Operating System :: OS Independent',  
 ],  
 python\_requires='>=3.6', # Python verzió  
 install\_requires=open('requirements.txt').read().splitlines(), # Függőségek a requirements.txt-ből  
)

## beautifulsoup.py

### Kód

#pip install -r requirements.txt --alternatív mód a könyvtárak telepítésére  
  
import requests  
from bs4 import BeautifulSoup  
  
#Cubix python képzések szűrés linkje  
url = "https://courses.cubixedu.com/kepzesek?\_gl=1%2a1ussfu4%2a\_ga%2aMTAyMjg2NjYxNS4xNzE5MzA5ODM4%2a\_ga\_KG4N6G61R4%2aMTcyODQ1NDA0MS4yMS4wLjE3Mjg0NTQwNTkuNDIuMC45NDYyNDc2NTc.%2a\_gcl\_au%2aOTgzMjU5OTY4LjE3MjgzOTM2MzA.&tags=python"  
  
# Az oldal alap URL-je a relatív hivatkozásokhoz  
base\_url = "https://courses.cubixedu.com"  
  
# Weboldal lekérése  
response = requests.get(url)  
  
# Oldal válaszának ellenőrzése  
if response.status\_code == 200:  
 print("Cubixon elérhető Python képzések listája:")  
 # HTML feldolgozása BeautifulSouppal  
 soup = BeautifulSoup(response.text, 'html.parser')  
  
 # Képzések keresése: 'course-box-v2' osztályú elemek, amelyek <a> tageket tartalmaznak  
 courses = soup.find\_all('div', class\_='course-box-v2')  
  
 # Képzések URL-jeinek és a belső címek lekérése  
 for course in courses:  
 link = course.find('a', href=True) # Keresünk <a> tagot href attribútummal  
 if link:  
 # Ellenőrizzük, hogy a link relatív-e, és ha igen, csak akkor fűzzük hozzá a base\_url-t  
 if link['href'].startswith('http'):  
 course\_url = link['href'] # Teljes URL, nincs szükség a base\_url-re  
 else:  
 course\_url = base\_url + link['href'] # Relatív URL, fűzzük hozzá a base\_url-t  
 # Az egyes képzések oldalának lekérése  
 course\_response = requests.get(course\_url)  
 if course\_response.status\_code == 200:  
 # Képzések oldala feldolgozása  
 course\_soup = BeautifulSoup(course\_response.text, 'html.parser')  
 # Az 'h1' elem keresése (a képzés címe)  
 h1\_tag = course\_soup.find('h1')  
 if h1\_tag:  
 print(f"Képzés címe: {h1\_tag.get\_text().strip()}")  
 print(f"Link a képzéshez: {course\_url}")  
 else:  
 print("Nem található 'h1' elem az oldalon.")  
 else:  
 print(f"Nem sikerült elérni a képzés oldalát. HTTP hiba: {course\_response.status\_code}")  
else:  
 print(f"Nem sikerült elérni az oldalt. HTTP hiba: {response.status\_code}")