**Tadeusz Brzeski**

**28.11.2020**

**Sprawozdanie nr 3**

**Temat: Web Scrapping w Pythonie**

**Teoria:**

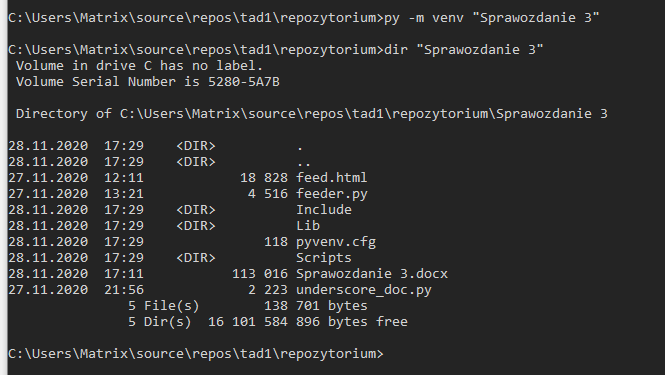
Wirtualne Środowisko pozwala na izolację instalowanych pakietów. Dzięki czemu można uniknąć konfliktu zależności (**dependency hell) w którym każdy program do działania potrzebuje innej wersji pakietu.**

Web Scrapping polega na „wyciąganiu” interesujących nas treści ze strony internetowej. Przykładem może być utworzony przeze mnie skrypt pobierający wszystkie zdjęcia z dowolnej strony internetowej.

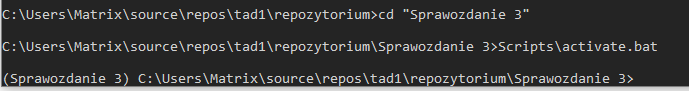
**Przebieg zadania:**

**Tworzenie wirtualnego środowiska:**

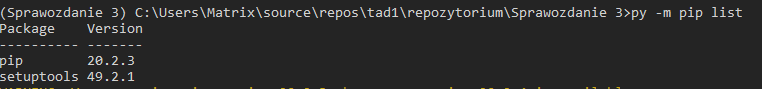
Aby stworzyć wirtualne środowisko wystarczyło wpisać komendę **venv** i podać ścieżkę projektu.

****

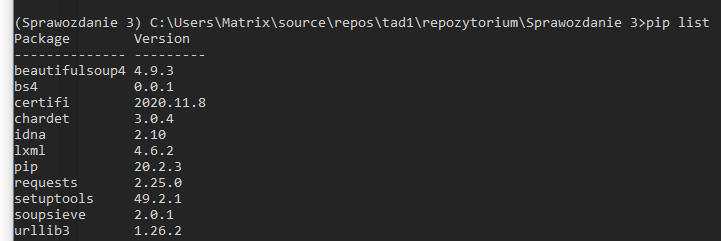
Następnie stworzone środowisko aktywowałem uruchamiając skrypt **activate**.

****

Lista pakietów w nowo utworzonym środowisku

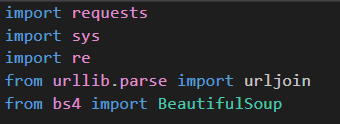
****

Po zainstalowaniu odpowiednich pakietów mogłem przejść do pisania skryptu

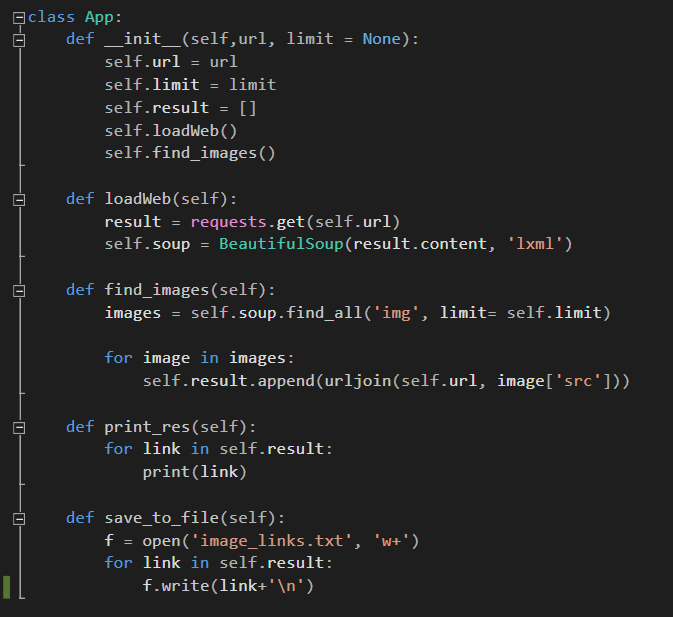
****

**Skrypt:**

Wykorzystywane biblioteki:



Klasa **App:**



Działanie funkcji w klasie App:

**\_\_init\_\_**: konstruktor pobierający url oraz ustalony limit który następnie uruchamia funkcje **loadWeb** i **find\_images**

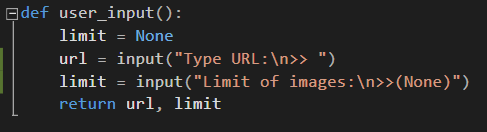
**loadWeb**: pobiera stronę internetową

**find\_images**: wyszukuje wszystkie zdjęcia zapisuje ich bezwzględną ścieżkę

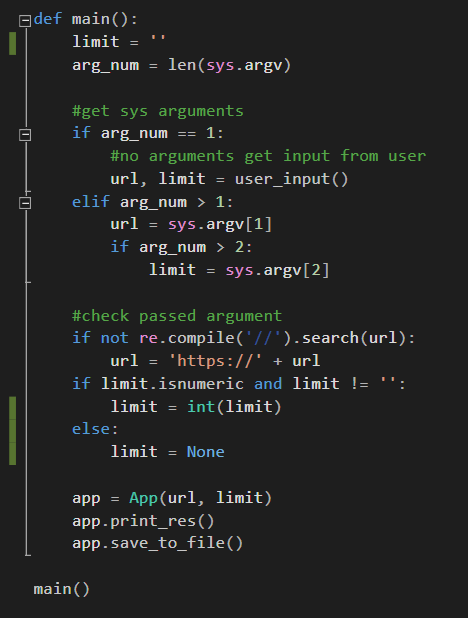
**print\_res**: wypisuje wyniki

**save\_to\_file**: zapisuje wyniki do pliku

Funcja **user\_input** pobierająca dane od użytkownika:



Funcja **main**: pobiera url oraz limit pobieranych obrazów, tworzy aplikację, następnie wyświetla oraz zapisuje linki.



**Rezultat:**

Po wywołaniu skryptu **imageScraper.py** z parametrem **"https://www.google.com/seachr?q=cat&hl=pl&tbm=isch"** wyświetla nam się lista linków do zdjęć.



Skrypt również utworzył plik **image\_links.txt**



Test pobranego linka przez skrypt:

