**Maurycy Szmuc**

**16.05.2021**

**Sprawozdanie nr 3**

**Temat: Web Scrapping w Pythonie**

**Teoria:**

**Wirtualne środowiska**

Wirtualne środowiska - odzielone od siebie środowiska programistyczne, w którym instalujemy jedynie biblioteki i pakiety potrzebne dla projektu. Pozwala nam to na oddzielenie od siebie programów używających np. dwóch różnych wersji pakietu/biblioteki, co zapobiega konfliktu zależności.

**W jakich sytuacjach powinno się stworzyć wirtualne środowisko:**

* Tworzysz wiele projektów zależnych od różnych wersji tych samych pakietów lub masz projekt, który musi być odizolowany od niektórych pakietów z powodu konfliktu zależności
* Pracujesz w środowisku Python, w którym nie możesz modyfikować katalogu pakietów witryn. Może to być spowodowane tym, że pracujesz w ściśle kontrolowanym środowisku, takim jak hosting zarządzany, lub na serwerze, na którym nie można zmienić wyboru interpretera (lub używanych w nim pakietów) ze względu na wymagania produkcyjne.
* Chcesz poeksperymentować z określoną kombinacją pakietów w ściśle kontrolowanych okolicznościach, na przykład w celu przetestowania zgodności krzyżowej lub kompatybilności wstecznej.
* Chcesz uruchomić „podstawową” wersję interpretera języka Python w systemie bez pakietów innych firm i instalować tylko te pakiety dla każdego projektu w razie potrzeby.

**Web scrapping**

Web Scraping - wydobywanie danych wyjściowych innego programu, automatyzując powtarzalne wpisywanie lub kopiowanie i wklejanie. Pozyskane dane często są przechowywane w ustukturyzowanym formacie. Skrobanie jest używane do:

* Pobierania informacji odnośnie produktów konkurencji przez firmy
* Pobierania informacji o potencjalnych klientach
* Porównywania cen w celu znalezienia najtańszej
* "Kompilacji" najnowszych danych z różnych stron i wstawianie tych informacji na jedną stronę (np.portale z ofertami pracy)

Możemy skrobać stronę wysyłając żadanie do serwera, na którym się znajduje wskazana przez nas strona, po czym filtruje ją w poszukiwaniu elementów HTML, które określiliśmy, i wyodrębnia zawartość, którą poleciliśmy wyodrębnić.

**Przebieg zadania:**

Część praktyczna (kod, screeny, opis działania funkcji, opis przebiegu zadania)

Użytkownik podaje programowi dwa różne artykuły z Wikipedii,jeden początkowy, a drugi docelowy,a program ma za zadanie używając jedynie hiperłączy, odnaleźć stronę docelową.



W powyższym screenie tworze wirtualne środowisko o nazwie venv, po czym instaluje na nim requests oraz bs4(Beautiful Soup).



Lista wszystkich zainstalowanych pakietów.



Importowanie pakietów do skryptów i deklaracja zmiennych globalnych.



Funkcja finishWiki wypisuje tablicę, jest wykonywana albo po wykonania 50 hopek, albo po znalezieniu celowego url.

Funkcja saveToTable zapisuje dane strony(nazwe artykułu i link) do tablicy, razem z nr hopki w której skrypt się na nią dostał.

Funkcja responseGet wysyła żądanie do serwera o kod źródłowy strony.



Funkcja scrapeWiki na samym początku wzywa funkcje responseGet, aby otrzymać kood źródłowy, po czym zapisuje nazwę artykułu i zapisuje ją do tablicy. Sprawdza, czy hopka jest równa 51, jeśli jest to wzywa finishWiki, jeśli nie to szuka celowego adresu użytkownika na stronie. Jeśli znajdzie, to wzywa ponownie responseGet aby dostać kod źródłowy docelowej strony, po czym zapisuje nazwę artykułu oraz url do tablicy i wzywa resultWiki. Jeśli nie znajdzie, to wybiera losowy link ze strony("który nie jest indeksem, ani linkiem do zdjęcia") i wzywa samego siebie, tym razem z losowym linkiem ze strony.