**Paweł Odwald**

**01/12/2020**

**Sprawozdanie nr 3**

**Temat: Web Scrapping w Pythonie**

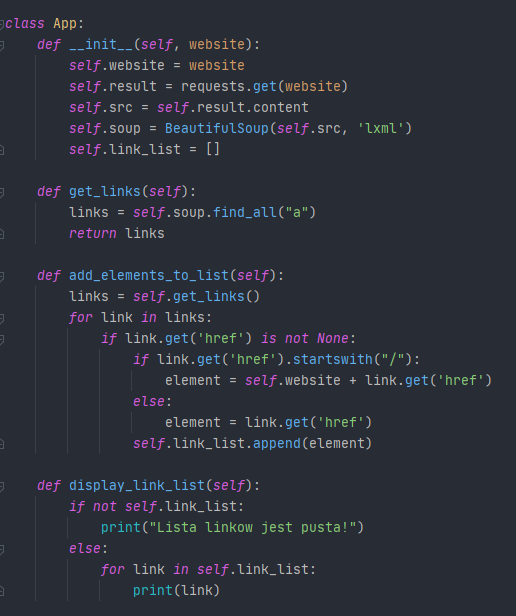
**Teoria:**

Główną zaletą korzystania z wirtualnego środowiska jest unikanie instalacji pakietów globalnie. Instalowanie pakietów globalnie może spowodować błędy w projektach, które korzystają z innych wersji pakietu oraz błędy w narzędziach systemowych. Korzystanie z wirtualnego środowiska jest uznawane jako dobra praktyka podczas korzystania z pakietów w naszych projektach.

Web scrapping pozwala pobrać dane ze stron internetowych i zapisać je w postaci plików lokalnie na komputerze.

**Przebieg zadania:**

**Definicja klasy**



**\_\_init\_\_:**konstruktor klasy który tworzy zmienne przechowujące:

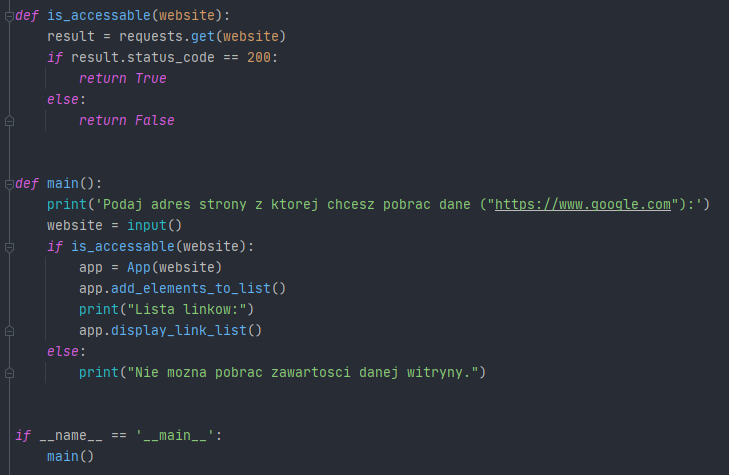
* Adres witryny
* Źródło strony
* Obiekt klasy BeautifulSoup
* Pustą listę linków, później uzupełnioną elementami po wywołaniu odpowiednich metod

**get\_links:** zwraca zawartość listy przechowującej elementy z tagiem *<a>*;

**add\_elements\_to\_list:** dodaje elementy do listy *link\_list* zawierające wszystkie adresy, do której linki podanej strony prowadzą; elementy mogły mieć wartość *None* lub zaczynały się od znaku */*, zamiast zacząć się od adresu strony a następnie znaku */* oznaczającego podstrony, także zadbano o korektę danych wartości dodając odpowiednie warunki *if*

**display\_link\_list:** wypisuje wszystkie elementy listy *link\_list* z pamięci

**Tworzenie obiektu klasy App oraz wywoływanie metod**



Jest to możliwe, że strona zwróci nam wartość statusu która odmówi nam pomyślnego pobrania danych. W danym przypadku, użytkownik dostanie komunikat o błędzie pobierania zawartości danej witryny i zakończy się praca programu. Jeżeli natomiast strona pozwoli nam pobierać z niej dane, to wywołujemy po kolei metody dodania linków do listy, a następnie wypisania ich przez program.