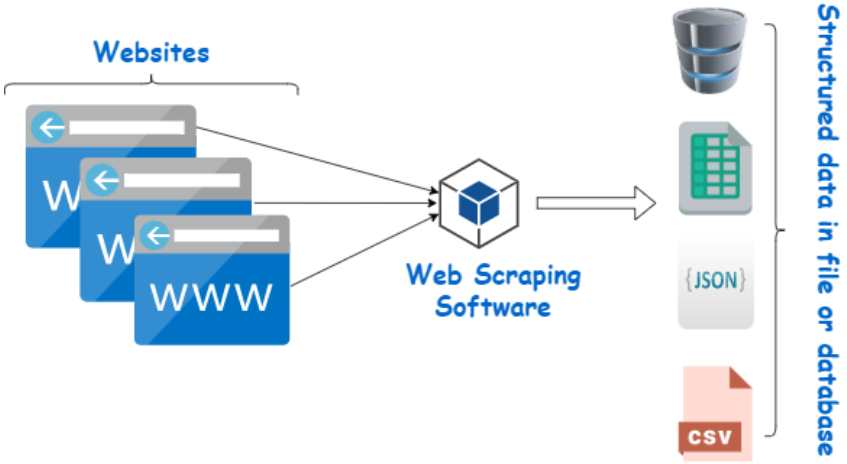
Język programowania wysokiego poziomu 3 - Przetwarzanie danych w języku Python

Celem ćwiczenia jest zapoznanie się z narzędziami do pozyskiwania danych ze stron internetowych i ich przetwarzania.



Źródło: webharvy

Abyśmy mieli na czym pracować konieczne jest wybranie strony docelowej a następnie połączenie się z nią (skorzystamy z przykładowej strony stworzonej do nauki scrapowania). Do tego celu przyda nam się biblioteka requests (należy ją wcześniej zaimportować).

Następnie należy wywołać funkcję get() z tego pakietu:

```
resp = requests.get(adres_url)
```

i otrzymamy prawdopodobnie:

<Response [200]>

W zmiennej resp otrzymaliśmy wszystkie informacje dotyczące tego, co zwrócił serwer. Jest to m.in. kod odpowiedzi.

404 oznacza, że nie znaleziono danej strony,

403 to brak praw dostępu,

200 oznacza, że żądanie obsłużono poprawnie.

Sam kod odpowiedzi znajduje się pod atrybutem status_code. Dla nas istotna będzie sama treść strony internetowej. Znajduje się ona w polu text. Powinien pokazać się kod strony - można porównać z tym wyświetlanym w przeglądarce po wciśnięciu F12.

Scrapowanie - proces pozyskiwania danych ze stron internetowych, najczęściej oparty na ekstrakcji i transformacji danych do pożądanego formatu.

Przy scrapowaniu warto użyć parsera HTML, który weźmie pod uwagę strukturę znaczników tworzącą stronę - np. pozwoli nam na wybór znacznika po jego nazwie, wartości atrybutów, relacji sąsiedztwa czy zawierania się. Użyjemy parsera BeautifulSoup. Załadujmy go:

```
from bs4 import BeautifulSoup
```

Teraz, tak jak zwykle, musimy zacząć od sparsowania kodu źródłowego:

```
soup = BeautifulSoup(html_doc, 'html.parser')
```

Zacznijmy od wyboru znacznika po jego nazwie. W tym prostym podejściu, gdy na stronie znajdują się dwa znaczniki o tej samej nazwie, wybierany jest pierwszy z nich:

`soup.title` - zwróci tytuł strony

Aby odwołać się do nazwy znacznika używamy pola name:

```
soup.title.name
```

Chcąc uzyskać wnętrze znacznika możemy skorzystać z kilku możliwości:

```
soup.title.string
```

`soup.title.text` - zwróci zawartość jako tekst

`soup.title.contents` - zwróci zawartość jako listę obiektów

Warto tu doprecyzować działanie contents. Zwrócona lista ta przechowuje rozbite wnętrze znacznika, co jest przydatne, gdy znajdują się w nim zagnieżdżone znaczniki. Przyjrzyjmy się, jak będą te pola działały dla bardziej złożonego znacznika <ul>:

Aby sprawdzić, czy dany znacznik w ogóle ma atrybut, używamy metody has_attr(), np:

```
soup.table.has_attr("border")
```

```
soup.table.has_attr("href")
```

W celu znalezienia interesującej nas sekcji możemy skorzystać z metody find() albo find_all(), które zwracają to, co pasuje do naszego kryterium. Podać możemy interesujący nas tag, ale istnieje możliwość korzystania z wyrażeń regularnych lub własnych funkcji. Find zwróci pierwszy napotkany fragment, find_all zwróci wszystkie, jakie zostaną znalezione.

Zadanie:

Wejdź na stronę https://www.scrapethissite.com/pages/forms/ i zapoznaj się z jej zawartością. Celem zadania jest ekstrakcja i przetworzenie danych z tabeli, zapis do pliku i (**jeśli wystarczy czasu, w przeciwnym razie przejdź do zadania z regexów**) późniejsza analiza (usunięcie ewentualnych pustych wierszy, usunięcie wybranej kolumny, dodanie

nowej, zastąpienie wybranej komórki inną wartością, obliczenie średniej dla dowolnej kolumny zawierającej wartości liczbowe, zrobienie dowolnego typu wykresu dla wybranej kolumny).

Wskazówki:

Najpierw należy zawęzić obszar poszukiwań. Możesz wykorzystać narzędzie podświetlające konkretny fragment kodu strony powiązany z obszarem, na który najedziesz myszką albo przeszukać kod strony pod kątem obecności jakiejś wartości obecnej w interesującym nas elemencie.

Następnie znajdź ten element (korzystając z metody find lub find_all) i sprawdź, czy otrzymano pożądany fragment.

Przeanalizuj znaczniki w tym fragmencie - czy coś się powtarza?

Mając tą wiedzę można iterować po kolejnych elementach naszego obiektu i wyciągać z niego interesujące nas części. Może być pomocna metoda strip(), która usunie niepotrzebne elementy z ciągu znaków.

Pożądaną treść możemy umieścić w DataFrame albo w innej strukturze, następnie przekształcić ją do df i zapisać jako csv. Przydatny będzie tutorial ze strony biblioteki Pandas, gdzie należy wyszukać interesujących nas treści (takie jak np. wybór kolumn, wierszy, komórek) https://pandas.pydata.org/docs/getting_started/intro_tutorials/index.html

oraz Matplotlib https://matplotlib.org/stable/index.html

Wyrażenia regularne

Zadanie:

Wejdź na stronę https://regexlearn.com/learn/regex101która umożliwia interaktywne ćwiczenie wyrażeń regularnych i przejdź przez zadania.

Polecam także zajrzeć na stronę https://regex101.com/ gdzie można wrzucić swój tekst i podać wyrażenie. Podświetlone zostaną trafione fragmenty.

Którkie podsumowanie znajduje się na pierwszej stronie instrukcji dra Wilkusa, choć omawiane wraz z Perlem, to istota pracy z regexami nie ulega zmianie https://home.agh.edu.pl/~mwilkus/os/p3.pdf