تجريف الويب من ويكيبيديا باستخدام بايثون

لمى اسبر علي، فيصل زنجرلي، محمد شريقي

ملخص: كشط الويب ، المعروف أيضًا باسم استخراج الويب أو الحصاد ، هو تقنية لاستخراج البيانات من شبكة الويب العالمية (WWW) وحفظها في نظام ملفات أو قاعدة بيانات لاسترادها أو تحليلها لاحقًا. في هذه المقالة، سنتعلم مفاهيم مختلفة لجرف الويب والهدف هو استخراج البيانات من صفحة Wikipedia الرئيسية وتحليلها من خلال تقنيات جرف الويب المختلفة. سوف نتعرف على تقنيات تجريف الويب المختلفة ، ووحدات Python لجرف الويب ، وعمليات استخراج البيانات ومعالجة البيانات. تجريف الويب هو عملية تلقائية لاستخراج المعلومات من الويب. ستمنحك هذه المقالة فكرة متعمقة عن تجريف الويب ولماذا يجب عليك اختيار تجريف الويب. سنقوم بانشاء ملف يحوي روابط المقالات وعناوينها الرئيسية.

الكلمات المفتاحية: تجريف الويب, ويكيبيديا

Web scraping from Wikipedia using Python

Lama, Faisal and Mohamed

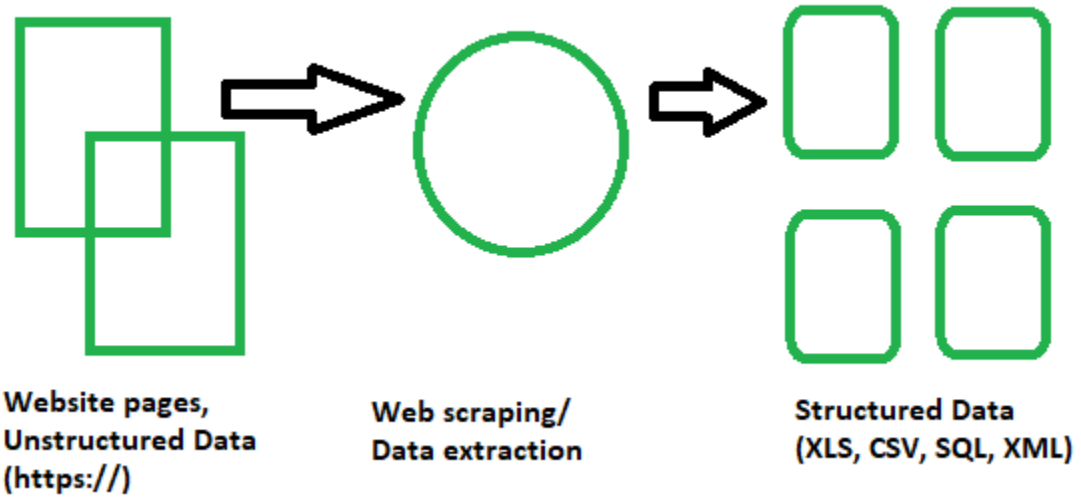
Abstract: Web scraping, also known as web extraction or harvest, is a technique for extracting data from the global web network (WWW) and preserving it in a file or database to recover or analyze it later. In this article, we will learn different concepts of the web cliff and the goal is to extract data from the main Wikipedia page and analyze it through various web cliff technologies. We will learn about different webing techniques, web cliffs, data extraction and data processing. Web scrape is an automatic process of extracting information from the web. This article will give you an in -depth idea of the web scraping and why you should choose the web scrolling. We will create a file that contains articles links and their main addresses.

Keywords: web scrapping, wikipedia

مقدمة

تجريف الويب في الأساس تقنية أو عملية يتم فيها تمرير كميات كبيرة من البيانات من عدد كبير من مواقع الويب عبر برنامج تجريف الويب المشفر بلغة برمجة ونتيجة لذلك ، يتم استخراج البيانات المنظمة التي يمكن حفظها محلياً في أجهزتنا بشكل منفصل في ملفات Excel أو JSON أو جداول البيانات. الآن ، لا يتعين علينا نسخ البيانات ولصقها يدوياً من مواقع الويب ولكن يمكن لأداة الجرف تنفيذ هذه المهمة لنا في بضع ثوانٍ.

يُعرف تجريف الويب أيضًا باسم تجريف الشاشة واستخراج بيانات الويب وحصاد الويب وما إلى ذلك.



الشكل 1 البنية العامة لتجريف الويب

إستخراج البيانات من صفحات الإنترنت من المواضيع الهامة حيث يستمر عدد مستخدمي شبكة الانترنت في التزايد عامًا بعد عام في ظل الانتشار الواسع لشبكة الإنترنت. يتزامن ذلك مع التقدم الهائل في التقنيات والعلوم، ويصاحب هذا التزايد تضخم كبير في المعلومات والبيانات، حيث تقول شركة CISCO (أشهر الشركات التي تقدم أجهزة البنية التحتية للشبكات) إن حجم مرور البيانات في الانترنت سيتجاوز واحد زيتا بايت خلال عام 2017 (1 زيتا بايت = ألف مليار جيجا بايت).

أحد أشكال هذا التضخم في البيانات يتمثل في الكمية الهائلة لصفحات الويب الموجودة والتي يتم إنشاؤها كل يوم، ومن المعروف أن هذه الصفحات يتم الوصول إليها عبر المتصفحات. أغلب صفحات الانترنت والمواقع لا تقدم خدمة حفظ نسخة من البيانات الموجودة فيها والخيار الوحيد هنا هو نسخ البيانات بشكل يدوي وحفظها في مكان وبالطريقة المناسبة، ولكن هذا الأمر سيكون

متعبًا للغاية وسيأخذ وقتا طويلًا في تنفيذه، وهنا يأتي دور أدوات وحلول إستخراج البيانات من صفحات الإنترنت (حيث يُمكن اطلاق مصطلح تجريف الويب على هذه العملية).

تُعرف Python بأنها أفضل لغة جرف للويب. يعد Scrapy and Beautiful Soup من بين الأطر المستخدمة على نطاق واسع استناداً إلى Python التي تجعل الجرف باستخدام هذه اللغة طريقاً سهلاً.

أهمية البحث وأهدافه:

يعد هذا البحث تطبيق عملي لتقنيات تجريف الويب وتمكن أهميته في إنشاء جدول بالمقالات الموجودة على ويكيبيديا وبالتالي متابعة أي مقالة جديدة دون الحاجة الى زيارة الموقع عبر المتصفح ويمكن تقديم هذه البيانات إلى الشركات المختصة بكتابة المحتوى لرؤية كل ما هو جديد على موقع ويكيبيديا.

أدوات وطرائق البحث:

1- مكتبة requests
2- مكتبة bs4
3- برنامج pycharm

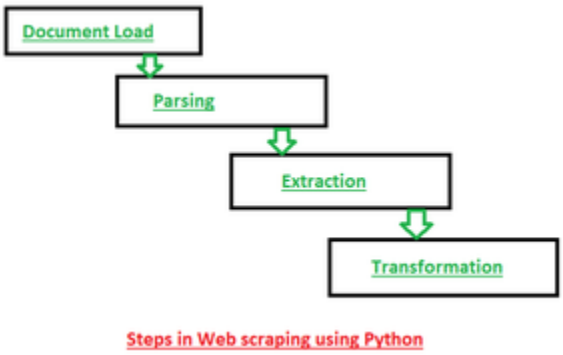
المنهجيات:

1- تجريف الويب:

كشط الويب ، المعروف أيضًا باسم استخراج الويب أو الحصاد ، هو تقنية لاستخراج البيانات من شبكة الويب العالمية (WWW) وحفظها في نظام ملفات أو قاعدة بيانات لاستردادها أو تحليلها لاحقًا. بشكل عام ، يتم إلغاء بيانات الويب باستخدام Hypertext Transfer Protocol (HTTP) أو من خلال متصفح الويب. يتم تحقيق ذلك إما يدويًا عن طريق المستخدم أو تلقائيًا عن طريق الروبوت أو زاحف الويب. نظرًا لحقيقة أن قدرًا هائلاً من البيانات غير المتجانسة يتم إنشاؤه باستمرار على WWW ،

فمن المسلم به على نطاق واسع أن تجريف الويب هو أسلوب فعال وقوي لجمع البيانات الضخمة. للتكيف مع مجموعة متنوعة من السيناريوهات ، أصبحت تقنيات تجريف الويب الحالية مخصصة من الإجراءات المخصصة الأصغر حجماً بمساعدة الإنسان إلى استخدام أنظمة مؤتمتة بالكامل قادرة على تحويل مواقع الويب بأكملها إلى مجموعة بيانات جيدة التنظيم. أدوات تجريف الويب الحديثة ليست قادرة فقط على تحليل لغات الترميز أو ملفات JSON ولكن أيضًا تتكامل مع التحليلات المرئية للكمبيوتر ومعالجة اللغة الطبيعية لمحاكاة كيفية تصفح المستخدمين البشريين لمحتوى الويب.

يمكن تقسيم عملية استخراج البيانات من الإنترنت إلى خطوتين متتاليتين ؛ الحصول على موارد الويب ثم استخراج المعلومات المطلوبة من البيانات التي تم الحصول عليها. على وجه التحديد ، يبدأ برنامج تجريف الويب عن طريق إنشاء طلب HTTP للحصول على موارد من موقع ويب مستهدف. يمكن تنسيق هذا الطلب إما في عنوان URL يحتوي على استعلام GET أو جزء من رسالة HTTP تحتوي على استعلام POST. بمجرد استلام الطلب ومعالجته بنجاح بواسطة موقع الويب المستهدف ، سيتم استرداد المورد المطلوب من موقع الويب ثم إرساله مرة أخرى إلى برنامج تجريف الويب العطاء. يمكن أن يكون المورد بتنسيقات متعددة ، مثل صفحات الويب التي تم إنشاؤها من HTML أو موجز البيانات بتنسيق XML أو JSON أو بيانات الوسائط المتعددة مثل الصور أو الصوت أو ملفات الفيديو. بعد تنزيل بيانات الويب ، تستمر عملية الاستخراج في تحليل البيانات وإعادة تنسيقها وتنظيمها بطريقة منظمة. هناك وحدتان أساسيتان لبرنامج تجريف الويب – وحدة لإنشاء طلب HTTP ، مثل Urllib2 أو السيلينيوم وأخرى لتحليل واستخراج المعلومات من كود HTML الخام ، مثل Beautiful Soup أو Pyquery. هنا ، تحدد وحدة Urllib2 مجموعة من الوظائف للتعامل مع طلبات HTTP ، مثل المصادقة وإعادة التوجيه وملفات تعريف الارتباط وما إلى ذلك ، بينما Selenium عبارة عن غلاف لمتصفح الويب يقوم بإنشاء متصفح ويب ، مثل Google Chrome أو Internet Explorer ، وتمكن المستخدمين من أتمتة عملية تصفح موقع الويب عن طريق البرمجة.



الشكل 2خطوات تجريف الويب

2- مكتبة beautifulsoap

فيما يتعلق باستخراج البيانات ، تم تصميم Beautiful Soup لكشط مستندات HTML ومستندات XML الأخرى. يوفر وظائف Pythonic مناسبة للتنقل والبحث وتعديل شجرة التحليل ؛ مجموعة أدوات لتحليل ملف HTML واستخراج المعلومات المطلوبة عبر lxml أو html5lib. يمكن أن يكتشف Beautiful Soup تلقائيًا ترميز التحليل تحت المعالجة وتحويله إلى تشفير يمكن للعميل قراءته. وبالمثل ، توفر Pyquery مجموعة من الوظائف المشابهة لـ Jquery لتحليل مستندات xml. ولكن بخلاف Beautiful Soup ، لا يدعم Pyquery سوى lxml لمعالجة XML السريعة.

يتم تنزيل هذه المكتبة من خلال التعليمة :

pip install beautifulsoup4

3- مكتبة requests:

هي مكتبة تعمل في طبقة التطبيقات مهمتها ارسال طلب الى سيرفر الـ http واستقبال الاستجابة على شكل صفحة html وهي ضرورية في تجريف الويب كوننا نقوم بتحميل صفحة الويب من أجل استخلاص المعلومات المطلوبة، ويجب تنزيل هذه المكتبة عن طريق التعليمة التالية:

pip install requests

4- لغة html:

لغة ترميز النص الفائق (HyperText Markup Language) اختصـار إتش تي إم إلHTML ، هي لغة ترميز تسـتخدم في إنشاء وتصميم صفحات ومواقع الويب، وتعتبر هذه اللّغة من أقدم اللّغات وأوسعها استخداما في تصميم صفحات الويب . HTMLهيكل صفحة الويب وتعطي متصفّح الإنترنت وصفا لكيفيّة عرضه لمحتوياتها، يمكن أن تساعده تقنيات مثل أوراق الأنماط المتتالية (CSS) ولغات البرمجة النصية مثل جافا سكريبت تستقبل متصفحات الويب مستندات HTML من خادم الويب أو من نظام الملفات وتعرضها، ووظيفة لغة HTML هي وصف بنية صفحات الويب هيكليًا.

العناصر في HTML هي اللبنة الأساسية لبناء مستنداتHTML ، إذ نستطيع عبرها إضافة الصور والكائنات التفاعلية مثل النماذج أو ملفات الفيديو والصوت؛ وتستطيع أيضًا إنشاء مستندات منظمة عبر استخدام وسوم للتصريح عن الفقرات والعناوين والروابط والاقتباسات والجداول وغيرها.

يمكن للغة HTML أن تُضمِّن برامجَ مكتوبةٍ بلغات مثل جافا سكريبت لتعديل سلوك ومحتوى صفحات الويب؛ وإضافة شيفرات أوراق الأنماط المتتالية CSS تؤدي إلى تعريف شكل وتخطيط المحتوى.

يبدأ المثال الآتي بالتصريح عن نوع المستند (DOCTYPE) الخاص بإصدار HTML5، ثم يُعرِّف العنصر الجذر <html> الذي يُشير إلى بدء مستند HTML. يحتوي العنصر <html> على عنصرين هما العنصر <head> و <body>؛ أما العنصر <head> فيحتوي على البيانات الوصفية التي تصف المستند مثل العنصر <title> الذي يضبط عنوان الصفحة والعنصر <meta> الذي ضبط هنا ترميز محارف المستند والعنصر <link> الذي أشار إلى مستند CSS والعنصر <script> الذي أشار إلى شيفرة JavaScript. وأما العنصر <body> فيمثِّل محتوى الصفحة نفسها، كالصور (العنصر <img>) والفقرات (العنصر <p>) وغير ذلك. لاحظ كيف ينتهي كل قسم من أقسام المستند بوسوم الإغلاق المناسبة.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Page Title</title>
    <link href="style.css" rel="stylesheet">
    <script src="javascript.js"></script>
  </head>
  <body>
    <img src="images/logo.png" alt="My Logo">
    <p>Hello World!</p>
  </body>
</html>
```

القسم العملي:

في البداية نستورد المكتبات اللازمة للعمل وهي مكتبة requests ومكتبة bs4 ومكتبة pandas من أجل إنشاء إطار بيانات خاص بالمعلومات المستخرجة من ويكيبيديا وحفظ هذا الاطار في ملف csv .

```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

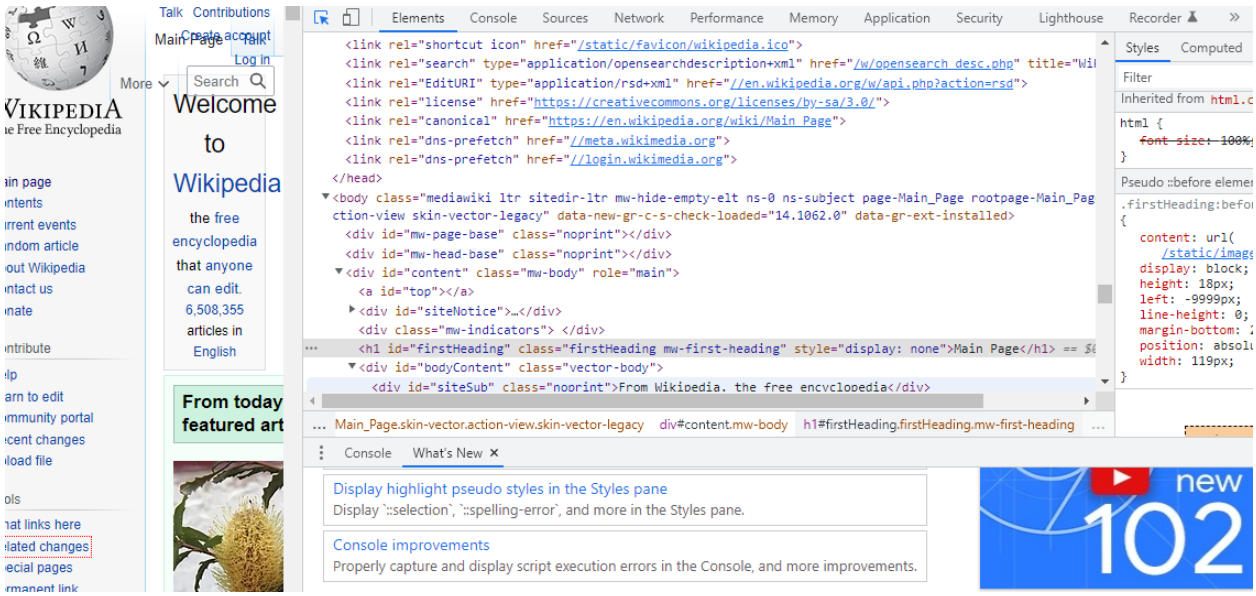
بعدها نقوم بطلب صفحة ويكيبيديا التالية

https://en.wikipedia.org/wiki/

وذلك باستخدام التابع get من المكتبة requests والذي يعيد استجابة تحوي صفحة الـ html المطلوبة. ثم نقوم بانشاء كائن من المكتبة beautifulsoup وذلك بتطبيق المحلل html.parser على صفحة الـ html هذه وبالتالي تحويل الصفحة بما تحويه من سمات html إلى مجموعة كائنات نستطيع الوصول الى محتواها بسهولة:

```
response = requests.get(url="https://en.wikipedia.org/wiki")
soup = BeautifulSoup(response.content, 'html.parser')
```

الخطوة التالية هي فحص الصفحة من أدوات المطور على متصفح كروم للتعرف على هيكلية الموقع وايجاد السمة الخاصة بالعنوان فنلاحظ كما هو موضح بالشكل 3 أن العنوان يكون موضوع ضمن الوسم id=firstHeading والروابط تكون ضمن الوسم id=bodyContent وضمن العلامة a من علامات html:



الشكل3 خيارات المطور من المتصفح

الخطوة التالية هي استخدام التابع find من اجل استخلاص العنوان وطباعة النص الخاص به وذلك باستخدام التابع string ضمن تعليمة الطباعة:

```
title = soup.find(id="firstHeading")
print(title.string)
```

وأيضاً نحتاج الى استخراج الروابط وذلك باستخدام التابعين find و find_all ونعرف متغير من نوع list فارغ من أجل حفظ الروابط بداخلها لاستخدامها لاحقاً مع إطار البيانات من pandas:

```
allLinks = soup.find(id="bodyContent").find_all("a")
links=[]
```

الآن نقوم بتعريف حلقة for من أجل تشكيل قائمة الروابط الخاصة بوكيبيديا وبالتالي هي الروابط التي تحوي /wiki/ وبعدها نقوم بطباعتها:

```
for link in allLinks:
   # Use this link to scrape
   linkToScrape = link["href"]
   if linkToScrape.startswith("/wiki/"):
      links.append("https://en.wikipedia.org"+linkToScrape)
print(links)
titles=[]
```

الخطوة التالية هي طلب كل صفحة باستخدام المكتبة requests وذلك من خلال الروابط الموجودة ضمن القائمة أعلاه ولتحقيق ضلك نحتاج لحلقة for نقوم من خلالها بطباعة كل عنوان وإضافته إلى القائمة titles.

```
for link in links:
   response = requests.get(url=link)
   soup = BeautifulSoup(response.content, 'html.parser')

   title = soup.find(id="firstHeading")
   print(title.string)
   titles.append(title.string)
```

الخطوة الأخيرة هي تشكيل متغير من نوع dictionary يحوي عنصرين أول عنصر المفتاح فيه هو الكلمة title والقيمة الخاصة بهذا المفتاح هي قائمة العناوين والعنصر الثاني المفتاح فيه هو link والقيمة الخاصة به هي قائمة الروابط وسنستخدم هذا المتغير لتشكيل إطار بيانات حسث عناوين الأعمدة هي المفاتيح والقيم الخاصة بالاعمدة هي القيم الخاصة بهذه المفاتيح وسنقوم بحفظ هذا الاطار كملف csv:

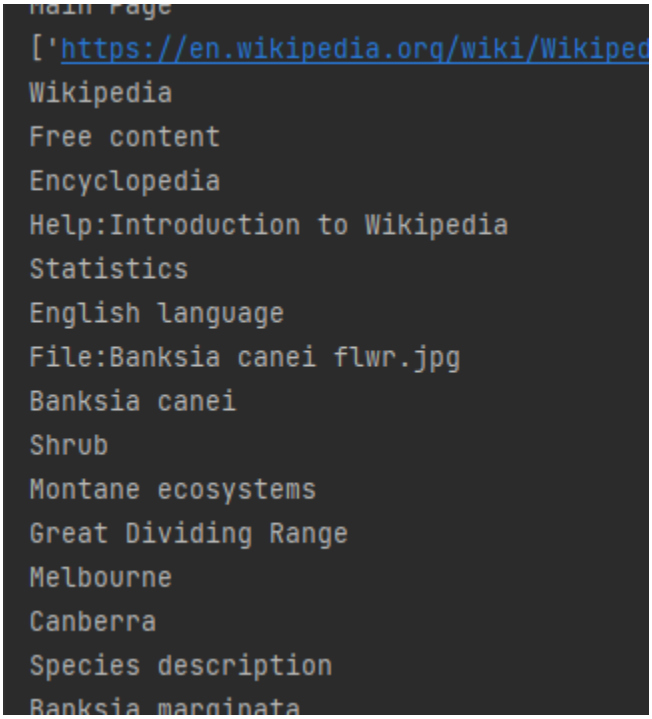
```
articles={
   "title":titles,"link":links
}
df=pd.DataFrame(articles)
df.to_csv('articles.csv')
```

النتائج:

عند تشغيل الكود يتم طباعة عنوان الصفحة الرئيسية في البداية وبعدها طباعة القائمة التي تحوي الروابط:

الشكل 4 خرج الكود

بعدها يتم فتح كل رابط واستخلاص العنوان من الصفحة وطباعته ووضعه ضمن قائمة:



الشكل 5 خرج الكود

محتويات ملف csv:

	title	links
0	Wikipedia	https://en.wikipedia.org/wiki/Wikipedia
1	Free content	https://en.wikipedia.org/wiki/Free_content
2	Encyclopedia	https://en.wikipedia.org/wiki/Encyclopedia
3	Help:Introduction to Wikipedia	https://en.wikipedia.org/wiki/Help:Introduction_to_Wikipedia
4	Statistics	https://en.wikipedia.org/wiki/Special:Statistics
5	English language	https://en.wikipedia.org/wiki/English_language
6	File:Banksia canei flwr.jpg	https://en.wikipedia.org/wiki/File:Banksia_canei_flwr.jpg
7	Banksia canei	https://en.wikipedia.org/wiki/Banksia_canei
8	Shrub	https://en.wikipedia.org/wiki/Shrub
9	Montane ecosystems	https://en.wikipedia.org/wiki/Montane_ecosystems#Subalpine_zon
10	Great Dividing Range	https://en.wikipedia.org/wiki/Great_Dividing_Range
11	Melbourne	https://en.wikipedia.org/wiki/Melbourne
12	Canberra	https://en.wikipedia.org/wiki/Canberra
13	Species description	https://en.wikipedia.org/wiki/Species_description
14	Banksia marginata	https://en.wikipedia.org/wiki/Banksia_marginata

articles

الشكل 6ملف النتائج

استنتاجات

لقد قمنا ببناء أداة جرف للويب في لغة بايثون تقوم بجرف صفحات ويكيبيديا. يتم استخلاص الروابط وعناوين المقالات ووضعها في ملف csv من ويكيبيديا. ويكيبيديا متساهلة جدًا عندما يتعلق الأمر بكشط الويب.

لقد تعلمنا مفاهيم مختلفة لكشط الويب والبيانات المقتبسة من صفحة Wikipedia الرئيسية وقمنا بتحليلها من خلال تقنيات كشط الويب المختلفة. ساعدتنا المقالة في الحصول على فكرة متعمقة عن تجريف الويب ، ومقارنتها بزحف الويب ، ولماذا يجب عليك اختيار تجريف الويب. تعلمنا أيضًا عن مكونات وعمل مكشطة الويب.

على الرغم من أن تجريف الويب يفتح العديد من الأبواب للأغراض الأخلاقية ، يمكن أن يكون هناك تجريف غير مقصود للبيانات من قبل ممارسين غير أخلاقيين مما يخلق خطرًا أخلاقيًا على العديد من الشركات والمؤسسات حيث يمكنهم استرداد البيانات بسهولة واستخدامها لوسائلهم الأنانية. يمكن أن يوفر جمع البيانات مع البيانات الضخمة معلومات عن السوق للشركة ومساعدتها على تحديد الاتجاهات والأنماط المهمة وتحديد أفضل الفرص والحلول. لذلك ، من الدقة توقع إمكانية ترقية استخراج البيانات إلى الأفضل قريبًا. لكن احذر من إساءة استخدام مواقع الويب ، واكتسح فقط البيانات التي يُسمح لك بكشطها.

المراجع:

1- Nair, V. G. (2014). *Getting started with beautiful soup*. Packt Publishing Ltd.
2- Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, 1-3.
3- https://ar.wikipedia.org/wiki/%D9%84%D8%BA%D8%A9_%D8%AA%D9%88%D8%B5%D9%8A%D9%81_%D8%A7%D9%84%D9%86%D8%B5_%D8%A7%D9%84%D9%81%D8%A7%D8%A6%D9%82
4- https://en.wikipedia.org/