Comparing Scrapping Libraries in Python

A speed and accuracy comparison

Tapasweni Pathak
Shaifali Agarwal
PyCon India 2015

- Software Developer @SAP Labs, Bangalore.
- GSOC’15 Mentor.
- GHC India organizing committee member.
- <3es to code.
- Scrapped 100+ websites for a startup.

Tapasweni Pathak

- Student.
- GSOC Developer for Systers Org.
- Openstack Contributor.

Shaifali Agarwal

Scrape?

- Extracting the information you want from any web page.

Don’t Want to Scrape?

- So many APIs.
- Some are even open source.
- Selenium, best to fetch after the initial page loads right?

Why Scrpae?

- Not everyone has a API.
- Not everyone expose there API.
- Not all APIs are understandable.
- You have the control.
- Fetch what you want.

What This Talk will cover?

- LXML vs. BeautifulSoup (with numerous pages)
- Scrapy: Why is it so easy to use? How fast can we go?
- What if page is broken?
- Who utilizes xpath and css select for identifying elements and why is it good if X does that?
- LXML with XPath
- LXML with CSS
- Beautiful Soup

What This talk will not cover?

- How to write a scrapper?

BeautifulSoup AND Lxml

- Top libraries for scraping.
- As long as page is not broken, both are very accurate.

LXML Over beautifulsoup?

- LXML uses xpath to identify elements.
- LXML uses cssselect to identify elements.
- The power.

Beautifulsoup over lxml

- Beautiful Soup supports the HTML parser included in Python’s standard library, but it also supports a number of third-party Python parsers.
- BS4 supports LXML Parser!
- So it comes up with advanced features and interface of BeautifulSoup without most of the performance hit.

From BeautifulSoup Creator

- That's it! Have fun! I wrote Beautiful Soup to save everybody time. Once you get used to it, you should be able to wrangle data out of poorly-designed websites in just a few minutes. Send me email if you have any comments, run into problems, or want me to know about your project that uses Beautiful Soup.

  --Leonard
- Reading between the lines :

  BS is a brilliant one-person project designed to save time to extract data out of poorly-designed websites. The goal is to save time right now, to get the job done, not necessarily to save you time in the long term, and definitely not to optimize the performance of your software.

- The C libraries libxml2 and libxslt have huge benefits:... Standards-compliant... Full-featured... fast. fast! FAST! ... lxml is a new Python binding for libxml2 and libxslt...
- The Crux!

From Why LXML

Methodology

- Wrote accurate* scrapers.
- Used pstats and cProfile to determine the time and function calls.
- Took number of trails to find the average.

*Please make them more accurate, if you can, they are live on my GitHub.

CASE Study 1

code

Who understands Pictures.

Library Used	Average Function Calls
LXML with XPATH	
LXML with CSS	
BeautifulSoup	

Who understands numbers.

Accuracy?

- As said, EQUAL!

CASe Study 2

- Website name?

- Graph.

Who understands Pictures

Who understands numbers

- Table as for the previous case study.

Accuracy

- No there is no 100% guaranty.
- They differ.

CASe Study 3

Who Understands Pictures

Who understands Numbers

Winner is...

- LXML with XPath!

  Not by much, you see.

SCRaPy....

- Uses LXML with XPath to find elements.
- Uses Twisted for asynchronous crawling.
- Best to crawl or spidering the web.
- What about speed?

CASE study 1

Fucntion calls

conclusion

- LXML with XPath.
- Xpath is confusing? I know. The use cssselect. Very close in speed.

Questions?

- Ask now?

  Raise your hand.
- Ask Later?

  Tweet to TapasweniPathak.
  Mail me tapaswenipathak@gmail.com

- Thanks!