Shedding Our Skins?

Reflections on a liaison librarian's attempt to learn how to scrape data from the web using Python

Jeremy Darrington
Politics Librarian, Princeton University Library
Presentation for the IASSIST 2014 Conference (Toronto)

The “what”

- “Web scraping”
  - Using software to extract info from websites
- “Python” python™
  - An open source programming language that is powerful and flexible, but also easy for beginners to learn
- “Liaison librarian”
  - Someone hired for his/her subject expertise—not technological prowess!—to support research in an academic department

The "why"

- Changing research environment for social scientists
  - New sources of data on the web
  - Many old sources now digitized or only published online
- Possibility (expectation?) of new roles and new skills for liaison librarians
  - Understanding researchers' environment and tools
  - Online data collection support/advice?
  - Web scraping as a new form of collection development?
- Personal interest
  - Programming is fun! (no, I'm not joking)
  - The tech support when tech support isn't available

The “catalyst”, part I

- “Web Scraping and Text Processing with Python” workshop in January 2014

The Program for Quantitative and Analytical Political Science (Q-APS)

Department of Politics, Princeton University
130 Corwin Hall, Princeton, New Jersey 08544-1012 (email)

Web Scraping and Text Processing with Python Workshop, January 27-31, 2014

Dates: January 27 – 31, 2014
Morning Session: 9:30–11:30am
Afternoon Session: 1:30–3:30pm
Location: TBA
Instructor: Radhika Saksena

Registration The registration for this workshop is now closed due to space limitations. If you are interested in being added to the waitlist, please fill out the waitlist form.
**Note: this workshop is open only to Princeton affiliates*

Over the last decade, both the variety and amount of data available to social scientists have expanded. These new data sources include administrative records (e.g., voter files, campaign finance and lobbying records), geo-referenced data (e.g., satellite maps, geocoded event data), and texts (e.g., speeches, court rulings, legislative bills). Many of these data sources can be accessed through the World Wide Web and as a consequence, techniques such as web scraping have become an essential part of social scientists' toolkit. The objective of this workshop is to introduce basic tools and techniques for automatic content extraction, parsing and other data-handling tasks that are commonly encountered in data-intensive research projects. The course will be taught in Python, and only a basic knowledge of general computing and programming (such as the R statistical programming taught at the Introductory Statistical Programming Camp) is assumed. We will cover techniques ranging from Python regular expressions and file manipulation, to the popular web scraping library ``Beautiful Soup" and PDF content extraction. The course ends with an introduction to the Twitter API for accessing Twitter content.

Latest Announcements

AskRC goes live!

Politics Statistical Programming Camp 2014

Web Scraping and Text Processing with Python Workshop

Advanced Statistical Programming Camp 2014

Advanced Statistical Programming Camp

Workshop: An introduction to git

more »

Recent Publications

The Supreme Court and Percolation in the Lower Courts: An Optimal Stopping Model

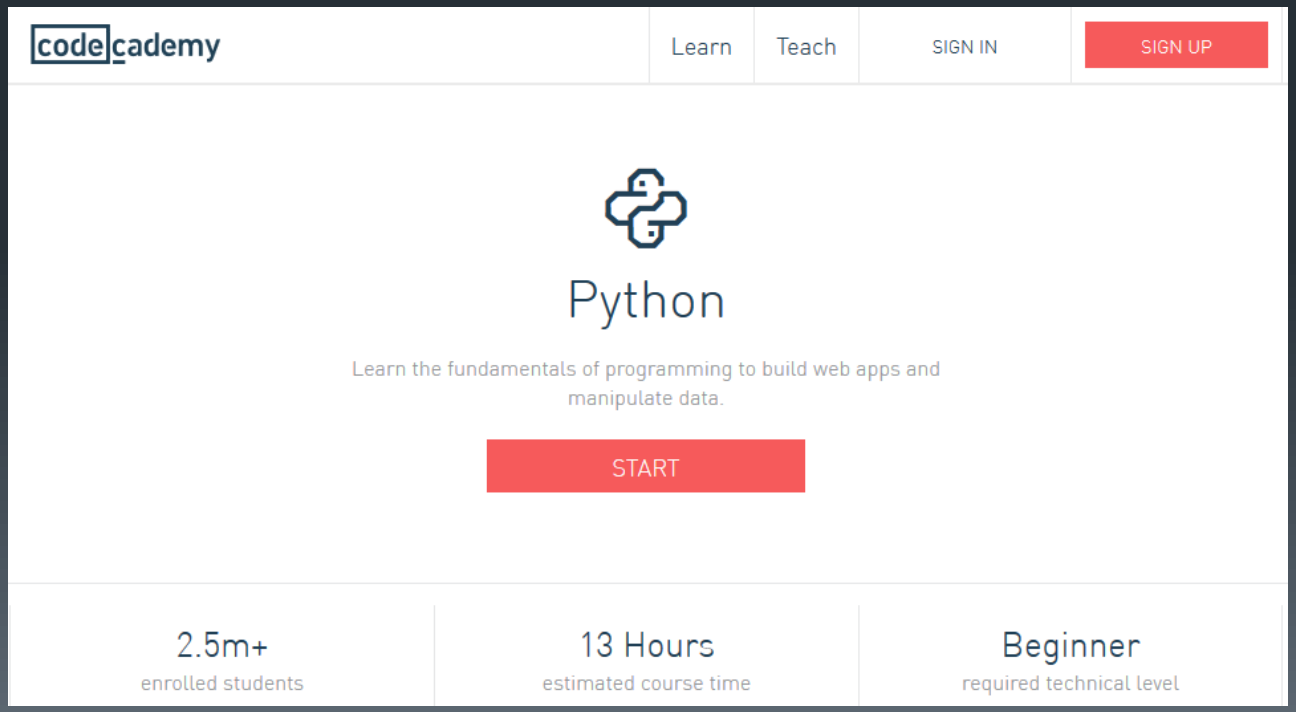
Poverty and Support for Militant Politics: Evidence from Pakistan

The “catalyst”, part II

- This presentation!
  - There’s nothing like a deadline and the risk of professional embarrassment to help you reach your goals!

The “how”, part I

- Codecademy course on Python
  - http://www.codecademy.com/tracks/python



Something of Value

For paperwork and accounting purposes, let's record the total value of your inventory. It's nice to know what we're worth!

Instructions

Let's determine how much money you would make if you sold all of your food.

01. Create a variable called `total` and set it to zero.

02. Loop through the `prices` dictionaries.

03. For each key in `prices`, multiply the number in `prices` by the number in `stock`. Print that value into the console and then add it to `total`.

04. Finally, outside your loop, `print total`.

Stuck? Get a hint!

script.py

```
prices = {
    "banana" : 4,
    "apple"  : 2,
    "orange" : 1.5,
    "pear"   : 3,
}
stock = {
    "banana" : 6,
    "apple"  : 0,
    "orange" : 32,
    "pear"   : 15,
}

total = 0

for key in prices:
    print key
    print "price: %s" % prices[key]
    print "stock: %s" % stock[key]
    print "total %s inventory value: $%d" % (key, prices[key]*stock[key])
    total += prices[key]*prices[key]
print total
```

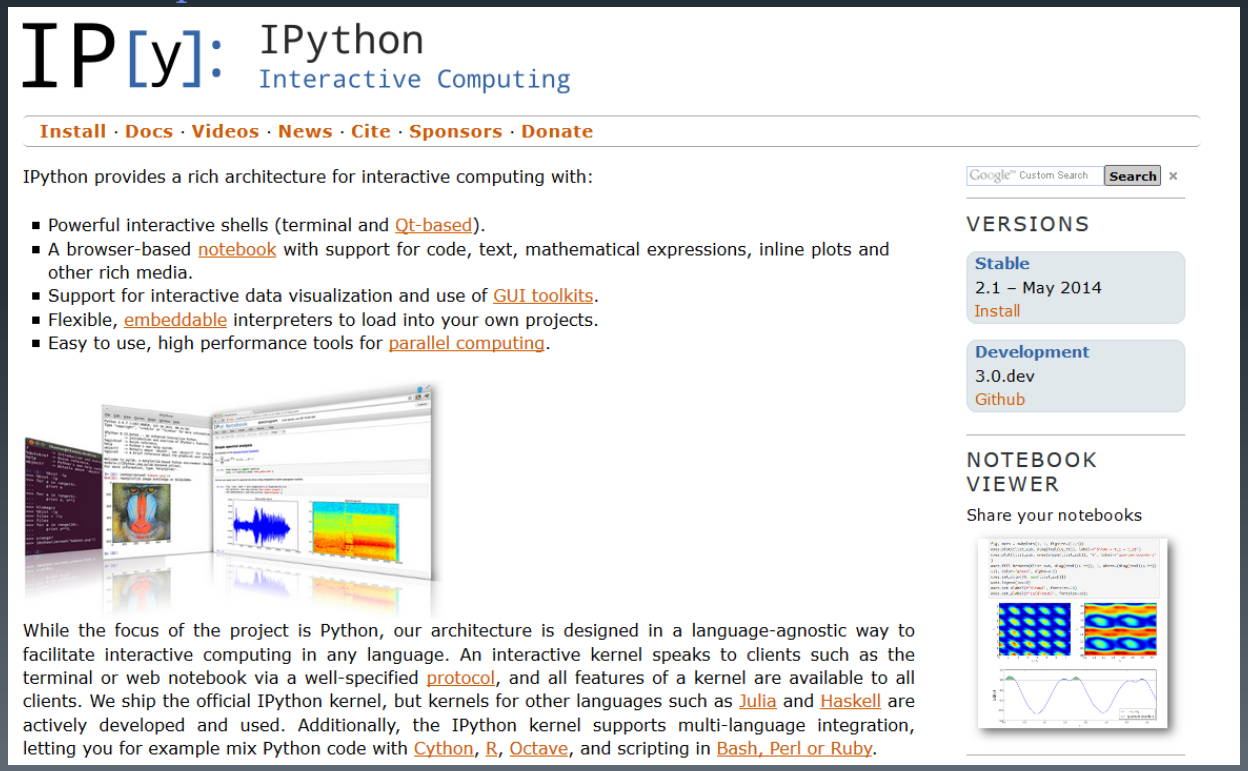
```
total pear inventory value: $45
banana
price: 4
stock: 6
total banana inventory value: $24
apple
price: 2
stock: 0
total apple inventory value: $0
31.25
None
```

Oops, try again. It looks like your code did not print the correct total.

Save & Submit Code | Reset Code

The “how”, part II

- Setting up Python on my own computer
  - IPython: http://ipython.org/; easy installation via Anaconda: http://continuum.io/downloads



IP[y]: IPython
Interactive Computing

Install · Docs · Videos · News · Cite · Sponsors · Donate

IPython provides a rich architecture for interactive computing with:

- Powerful interactive shells (terminal and Qt-based).
- A browser-based notebook with support for code, text, mathematical expressions, inline plots and other rich media.
- Support for interactive data visualization and use of GUI toolkits.
- Flexible, embeddable interpreters to load into your own projects.
- Easy to use, high performance tools for parallel computing.

While the focus of the project is Python, our architecture is designed in a language-agnostic way to facilitate interactive computing in any language. An interactive kernel speaks to clients such as the terminal or web notebook via a well-specified protocol, and all features of a kernel are available to all clients. We ship the official IPython kernel, but kernels for other languages such as Julia and Haskell are actively developed and used. Additionally, the IPython kernel supports multi-language integration, letting you for example mix Python code with Cython, R, Octave, and scripting in Bash, Perl or Ruby.

Google™ Custom Search Search

VERSIONS

Stable
2.1 – May 2014
Install

Development
3.0.dev
Github

NOTEBOOK VIEWER

Share your notebooks

IP[y]: Notebook Web_scraping_Day_2--regular_expressions Last Checkpoint: May 29 11:00 (autosaved) Logout

File Edit View Insert Cell Kernel Help

Code Cell Toolbar: None Env: np18py27-1.9

But this could clearly be rewritten more parsimoniously, which not only reduces lines of code but also extends it to work with additional urls and search strings. It took me a few tries, but the script below will now return results for additional URLs and search strings.

In [95]:

```
from bs4 import BeautifulSoup

urls = ["http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html", \
        "http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html", \
        "http://www.kantei.go.jp/jp/96_abe/statement/2014/0515kaiken.html"]
soups = []
for url in urls:
    response = urllib2.urlopen(url).read()
    soup = BeautifulSoup(response)
    soups.append(soup)
searchStrings = [u"はなく", u"景気回復", u"経済", "TPP", u"震災", u"安全保障"]
patterns = []
for searchString in searchStrings:
    pattern = re.compile(searchString)
    patterns.append(pattern)

for i, soup in enumerate(soups):
    total = 0
    for j, pattern in enumerate(patterns):
    #use the index from this loop to get each search string and the index from the parent loop to feed the proper url
        print '%s occurs on %s %d times.' % (searchStrings[j], urls[i], len(pattern.findall(soup.text)))
        total += len(pattern.findall(soup.text))
    print 'Total occurrences of all search terms is %d \n' % total
```

```
はなく occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html 2 times.
景気回復 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html 3 times.
経済 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html 8 times.
TPP occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html 5 times.
震災 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html 1 times.
安全保障 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0106kaiken.html 1 times.
Total occurrences of all search terms is 20

はなく occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html 3 times.
景気回復 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html 1 times.
経済 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html 5 times.
TPP occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html 1 times.
震災 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html 2 times.
安全保障 occurs on http://www.kantei.go.jp/jp/96_abe/statement/2014/0101nentou.html 3 times.
Total occurrences of all search terms is 15
```

The “how”, part II

- Viewing/sharing notebooks
  - http://nbviewer.ipython.org/
  - http://www.wakari.io/
  - My example notebooks: https://www.wakari.io/jdarring

Reflections

- Scraping is a valuable tool, but legal and ethical considerations abound
- Lots of great tools and tutorials, but …
- Learning any new skill takes time and repetition
  - The best learning happens when you tackle real problems
  - Group learning can be helpful, esp. in programming
  - Library administrators should be supportive and actively encourage exploration
- Programming skills are useful and transferable