Web Scraping with Python

by @sauravtom

Who uses Scrapers ?

Scrapers as backbone of Big Data

Importance in Industry level as well as side projects.

Data Scraping

Automated Process

Specify css or xml path (Chrome Web Inspector)

Grab the content

Store it in a database

Repeat (if necessary) via cron jobs

Scraping libraries in python

lxml

BS4

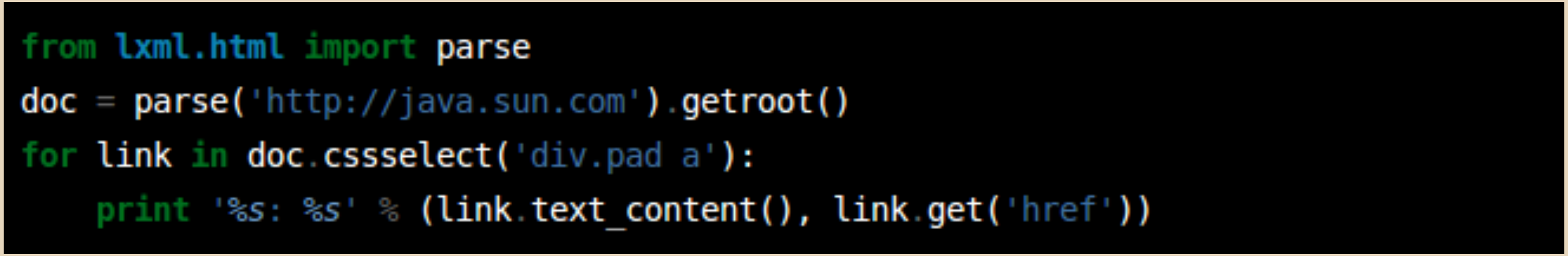
Scrapy

Mechanize

twill

lxml

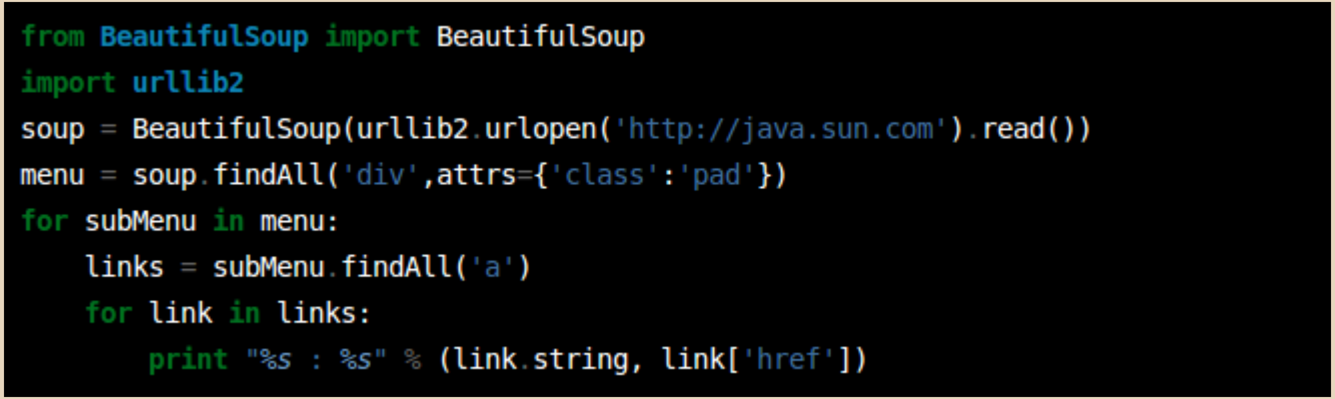
lightweight and faster



```
from lxml.html import parse
doc = parse('http://java.sun.com').getroot()
for link in doc.cssselect('div.pad a'):
    print '%s: %s' % (link.text_content(), link.get('href'))
```

bs4

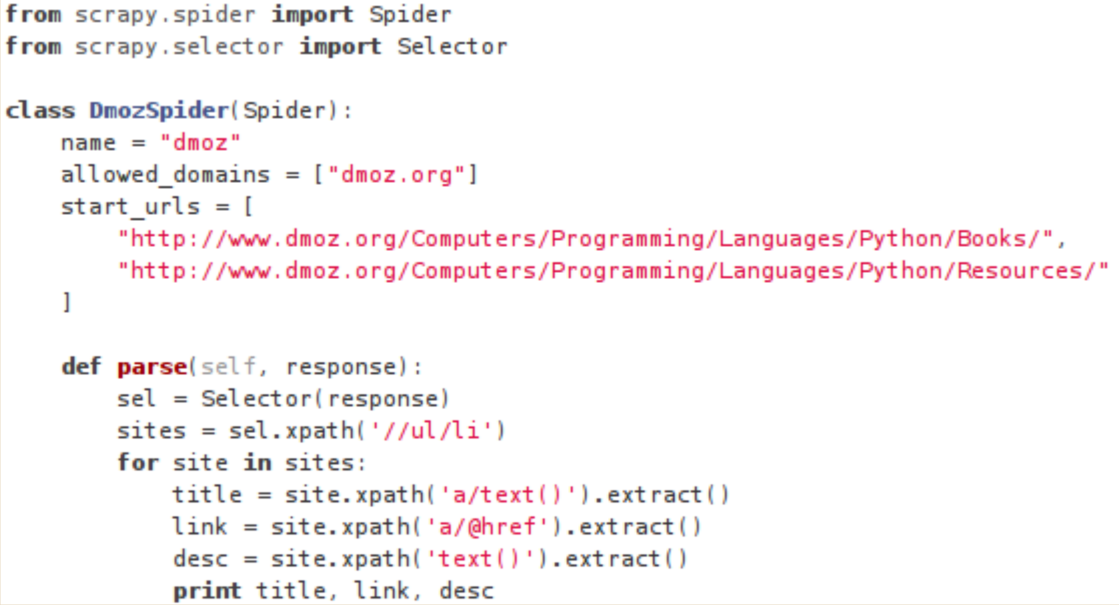
Good documentation and more popular



```
from BeautifulSoup import BeautifulSoup
import urllib2
soup = BeautifulSoup(urllib2.urlopen('http://java.sun.com').read())
menu = soup.findAll('div',attrs={'class':'pad'})
for subMenu in menu:
    links = subMenu.findAll('a')
    for link in links:
        print "%s : %s" % (link.string, link['href'])
```

scrapy

Complete web-spider framework



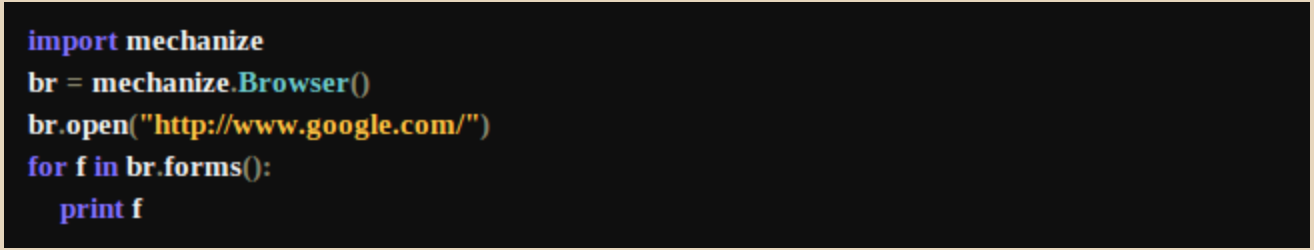
```python
from scrapy.spider import Spider
from scrapy.selector import Selector

class DmozSpider(Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
    ]

    def parse(self, response):
        sel = Selector(response)
        sites = sel.xpath('//ul/li')
        for site in sites:
            title = site.xpath('a/text()').extract()
            link = site.xpath('a/@href').extract()
            desc = site.xpath('text()').extract()
            print title, link, desc
```

mechanize

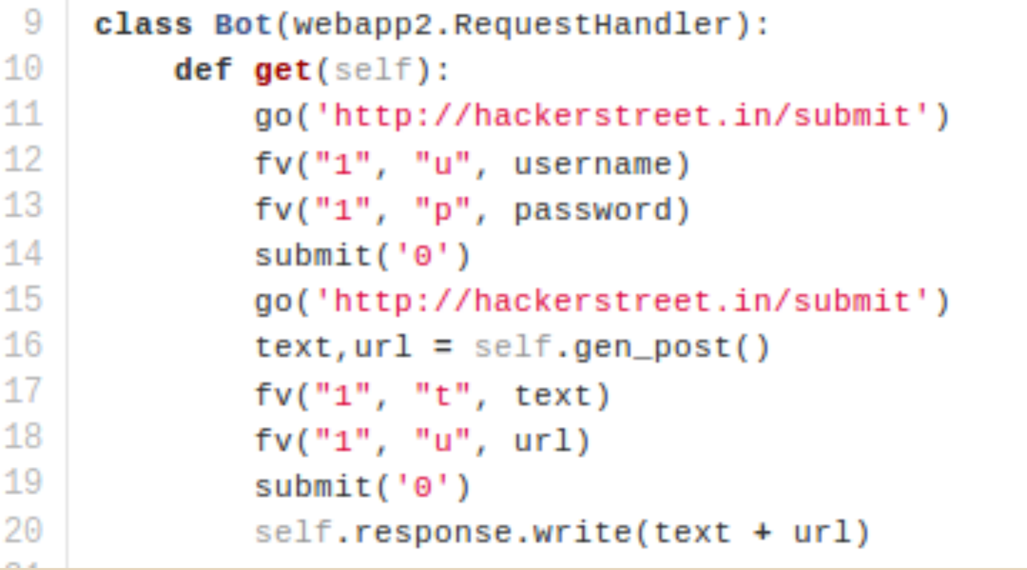
Scraper + browser automation



```
import mechanize
br = mechanize.Browser()
br.open("http://www.google.com/")
for f in br.forms():
   print f
```

twill

lightweight shell around mechanize



```python
class Bot(webapp2.RequestHandler):
    def get(self):
        go('http://hackerstreet.in/submit')
        fv("1", "u", username)
        fv("1", "p", password)
        submit('0')
        go('http://hackerstreet.in/submit')
        text,url = self.gen_post()
        fv("1", "t", text)
        fv("1", "u", url)
        submit('0')
        self.response.write(text + url)
```

Web is the API

What are APIs

Twitter example

(Dive into code ..)

Scraper Demonstration in bs4

Inspect the element

Find the node

Plug it in

(some code and pictures)

Making Scrapers faster

Thread and Queues

Use profiling to find bottlenecks

Use memcache to reduce file read-write times

Thats it !!

Scrape Wisely

Do not steal

Share your scrapers (github,scraperwiki)

(links to the code present in these slides)