Webscraping with Python

Carleton University data boot camp - June 24, 2016
Glen McGregor
sushiboy21@gmail.com

In this workshop, we'll learn how to use the Python programming language to "web scrape" or screen scrape data. Web scraping is the process of using small computer programs to robotically download large amounts of data. Python is an easy-to-use language and has very clear syntax. It's also open source and -- this is important in our industry -- completely free to use.

If you have a Mac, you already have Python installed. You can start using Python simply by typing "Python" in the Terminal program. You can also edit Python scripts in any text editor, save them with a .py extender, and run them from the Terminal by typing "python filename.py"

On a Windows computer, you will need to download and install from python.org. You can run your scripts using the included IDE (integrated development environment).

Today, we're going to use a web-based version of Python built into the Mac operating system. We can edit the scripts in any text editor, but the free TextWrangler is the best option. Save the files on the Desktop.

We'll run the files using a program called Terminal, which is built into the Mac and can be launched by clicking APPLICATIONS>UTILITIES>TERMINAL.

In the Terminal window, change directories to the Destop by typing:

```
cd ~/Desktop
```

To launch a Python file you've written, in Terminal type:

```
python myfile.py
```

Before we start scraping, we need to learn some basic Python functions. Go to ScraperWiki, choose CREATE A SCRAPER and then select PYTHON and try these commands.

*** Create a numerical variable:**

```
x = 1
print x
# do some math with the variable
x = x + 1
print x
```

*** Create a string variable**

```
my_name = "Bob"
print my_name
# concatenate the variable
my_name = my_name + " " + "Smith"
print my_name
```

*** Create a while loop**

```
x = 0
while x < 10:
      print x
      x = x + 1
```

*** Create a for loop**

```
my_name  = "Bob Smith"
for every_letter in my_name:
      print every_letter
```

*** Define a function**

```
def my_function(my_name):
      my_sentence = my_name + " is an enemy of the state"
      print my_sentence

my_function("Dave")
my_function("Glen")
my_function("Fred")
```

*** Use a Regular Expression search to parse a string**

```
import re
full_name = "Glen Edward McGregor"
middle_name = re.search("Glen(.+?)McGregor", full_name)
print middle_name
middle_name = middle_name.group(1)
print middle_name
```

*** Use the built in Python module called urllib2 to download a web page:**

```
import urllib2
our_url = "http://www.ottawacitizen.com"
the_page = urllib2.urlopen(our_url).read()
print the_page
```

* **Save the scraped data to the a data file on your hard disk**

```
import urllib2
our_url = "http://www.ottawacitizen.com"
the_page = urllib2.urlopen(our_url).read()
print the_page

our_file = open("citizen.html","a")
our_file.write(the_page)
```

Scraping Orders of Canada

If we combine these concepts we have learned so far, we can create a little program that will scrape a website. We'll scrape the database of people who received Orders of Canada appointments from the Governor General's website. You can use this data to build a list of everyone in your community who received one, or look at regional differences in the way the Orders are awarded to see if your area is under or over-represented.

First, we need to import the modules we need.

```
import urllib2
import re
```

Then we'll create a function to scrape a page we send it

```
def our_scraper(our_url):
    the_page = scraperwiki.scrape(our_url)
    print the_page
```

Go to gg.ca and navigate through the black bar at the top of the site to find the database by selecting Honours>Find A Recipient. Then select the Orders of Canada radio button and click search.

The page will generate results using this URL:

http://gg.ca/honours.aspx?ln=&fn=&t=12&p=&c=&pg=1&types=12

The website is giving us paginated results of 50 names at a time. If we change the URL to set the page

number to 2, we'll get the next 50 results.

http://gg.ca/honours.aspx?ln=&fn=&t=12&p=&c=&pg=2&types=12

The website also gives us the option of limiting the date range of the search. If we set it to the past three years, it will look like this:

http://gg.ca/honours.aspx?ln=&fn=&t=12&p=&c=&pg=1&types=12&advoocaf=2012-06-24&advoocat=2016-06-24

This should give us a ist of about 390 names, or six pages of data.

Now we'll create a loop in our Python script to scrape that URL, but increase the page number each time it calls the scrape function

```
x = 1
while x < 10:
      url = "http://gg.ca/honours.aspx?ln=&fn=&t=12&p=&c=&pg=" + str(x) +
      "&types=12&advoocaf=2012-06-24&advoocat=2016-06-24"
      our_scraper(url)
      x = x + 1
```

When the script runs, we should see hundreds of lines of HTML code. Yes, it's an ugly mess, because we are viewing the raw data without it being formatted by a web broswer. That's okay. The data we want in somewhere in that soup of HTML.

Now we want to create another function that will extract the data we're looking for in the HTML.

We can use Regular Expressions to extract the data we want from the HTML code, and store this in a function and save the data to disc:

```
def parse_text(the_page):
      the_page = re.sub("\r","", the_page)
      the_page = re.sub("\n","", the_page)
      the_page = re.sub(" +"," ", the_page)

      for listing in re.finditer('<a href="/honour.aspx\?id=.+?">(.+?)</a> </td>
<td>(.+?)</td> <td>(.+?)</td> </tr>', the_page):
            name = listing.group(1)
            city = listing.group(2)
            award = listing.group(3)
            print name, city, award
            line = name + "\t" + city + "\t" + award + "\n"
            f = open("orders.txt","a")
            f.write(line)
```

The entire script should look like this:

```
import urllib2
import re
import time

def parse_text(the_page):
      the_page = re.sub("\r","", the_page)
      the_page = re.sub("\n","", the_page)
      the_page = re.sub(" +"," ", the_page)
      for listing in re.finditer('<a href="/honour.aspx\?id=.+?">(.+?)</a> </td>
<td>(.+?)</td> <td>(.+?)</td> </tr>', the_page):
            name = listing.group(1)
            city = listing.group(2)
            award = listing.group(3)

            print name, city, award
            line = name + "\t" + city + "\t" + award + "\n"
            f = open("orders.txt","a")
            f.write(line)

def our_scraper(url):
      req = urllib2.Request(url)
      req.add_header('User-Agent', 'Mozilla/4.0 (compatible; MSIE 7.0b; Windows NT
6.0) If webscrape causes problems, call Glen McGregor 613.235.6685')
      response = urllib2.urlopen(req)

      the_page = response.read()
      #print the_page
      parse_text(the_page)

x = 1
while x < 10:
      url = "http://gg.ca/honours.aspx?ln=&fn=&t=12&p=&c=&pg=" + str(x) +
"&types=12&advoocaf=2012-06-24&advoocat=2016-06-24"
      our_scraper(url)
      x = x + 1
      time.sleep(1)
```