Chapter 9

Code Samples

Code from Your First Multi-Page Scrape

```
#!/usr/bin/env python
import urllib2
from bs4 import BeautifulSoup
import unicodecsv
import time
output = open('fullPathtoFile.txt', 'w')
writer = unicodecsv.writer(output, delimiter='\t',
encoding='utf-8')
writer.writerow([’date’,’vendor’,’description’,’value’])
years=[200405,200506,200607,200708,200809,200910,201011,
201112,201213,201314,201415,201516,201617]
quarters=[1,2,3,4]
for fiscalYear in years:
    for quarter in quarters:
        url = 'http://www.tpsgc-pwgsc.gc.ca/cgi-
        bin/proactive/cl.pl?lang=eng;SCR=L;Sort=0;PF=CL' +
        str(fiscalYear) + 'Q' + str(quarter) +'.txt'
        response = urllib2.urlopen(url)
        time.sleep(5)
        HTML = response.read()
        soup = BeautifulSoup(HTML, "html.parser")
        for eachrow in soup.findAll('tr'):
            data =[]
            cells = eachrow.findAll('td')
            for eachdataitem in cells:
                data.append(eachdataitem.text)
            writer.writerow(data)
            print data
```

Code from Scraping Sites that Use JavaScript and AJAX will be added to this file later.

Code for Scraping Sites that Don't Want to be Scraped/Scraping Sites that Use Search Forms:

```
import time, unicodecsv

from bs4 import BeautifulSoup

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.support.ui import Select

from selenium.webdriver.common.desired_capabilities import
DesiredCapabilities

caps = DesiredCapabilities.FIREFOX

caps["marionette"] = True

outfile = open("c:/WriteFiles/AirOperators.csv",'wb')

writer = unicodecsv.writer(outfile,dialect='excel')

writer.writerow(['Name','Trade Name(s)','Address'])

driver=webdriver.Firefox(capabilities=caps)

driver.implicitly_wait(30)

driver.get('http://wwwapps.tc.gc.ca/saf-sec-sur/2/CAS-
SAC/olsrlel.aspx?lang=eng')

driver.find_element_by_id("txtName").clear()

driver.find_element_by_id("txtName").send_keys("Air")

Select(driver.find_element_by_id("ddlCar")).select_by_visible_te
xt("car 705 - Airline Operations")

driver.find_element_by_id("btnSearch").click()

driver.find_element_by_id("btnAll").click()

time.sleep(10)

mainPage = driver.page_source

soup = BeautifulSoup(mainPage,'html.parser')

theTable = soup.find('table')
```

```
theRows = theTable.findAll('tr')

for row in theRows:

    outList = []

    theCells = row.findAll('td')

    for cell in theCells:

        outList.append(cell.text.strip())

    writer.writerow(outList)
```

Code for Scraping Using Regular Expressions

```
import urllib2, re, time
def name_extractor(the_page):
      the_page = re.sub("\r", "", the_page)
      the_page = re.sub("\n", "", the_page)
      the_page = re.sub(" +", " ", the_page)
      for each in re.finditer('<a
href="/honour\.aspx\?id=.+?>(.+?)</a>\s?</td>\s?<td>\s?(.+?)\s?<
/td>', the_page):
      recipient = each.group(1)
      city_prov = each.group(2)
      print recipient, city_prov

counter =1
while counter < 6:
      url = "http://www.gg.ca/honours.aspx?t=12&types=12&pg=" +
str(counter)
      the_page = urllib2.urlopen(url).read()
      name_extractor(the_page)
      counter = counter + 1
      time.sleep(1)
```

Code from Building a Twitter Bot:

```
import tweepy
import urllib3.contrib.pyopenssl
urllib3.contrib.pyopenssl.inject_into_urllib3()
TWEEPY_CONSUMER_KEY = #'YOUR_API_KEY'
TWEEPY_CONSUMER_SECRET = #'YOUR_SECRET_API_KEY'
TWEEPY_ACCESS_TOKEN = #'YOUR_ACCESS_TOKEN'
TWEEPY_ACCESS_TOKEN_SECRET = #'YOUR_SECRET_ACCESS_TOKEN'
auth1 = tweepy.auth.OAuthHandler(TWEEPY_CONSUMER_KEY,
TWEEPY_CONSUMER_SECRET)
auth1.set_access_token(TWEEPY_ACCESS_TOKEN,TWEEPY_ACCESS_TOKEN_SECRET)
```

```
api = tweepy.API(auth1)
def tweetit(statusUpdate):
    api.update_status(status=statusupdate)
    print statusupdate
statusupdate = "This is a test Tweet from our new Python bot!"
tweetit(statusUpdate)
```

Code from Building your First News App

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>First News App</title>
</head>
<body>
    <h3>Minimum down payment calculator</h3>
    <p>Enter the purchase price of the home you have in mind, then click "submit".</p>
    <label for="purchase-price">Purchase price</label>
    <input type="text" name="purchase-price" id="dollars" placeholder="500000">
    <input type="submit" value="Submit" onclick="calculate()">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
    <script>
        function calculate() {
            var price = $('input#dollars').val();
            var downpayment = price * 0.2;
            $('p.results').remove();
            $('body').append('<p class="results">Your minimum down payment on this house is: $' + downpayment + '</p>');
        }
    </script>

</body>
</html>
```