WEB SCRAPING WITH PYTHON

10/2019

Applied Analytics Club

Set Up

- Google Chrome is needed to follow along with this tutorial.
- *Install the Selector Gadget Extension for Chrome as well.*
- If you haven't done already, download and install Anaconda Python 3 Version at:
- https://www.anaconda.com/distribution
- Next, use Terminal or Command Prompt to enter the following, one by one:
  - pip install bs4
  - pip install selenium
  - pip install requests
- Download all workshop materials @ ^
- *In case of errors, raise your hand and we will come around. For those who have successfully completed the install, please assist others.*

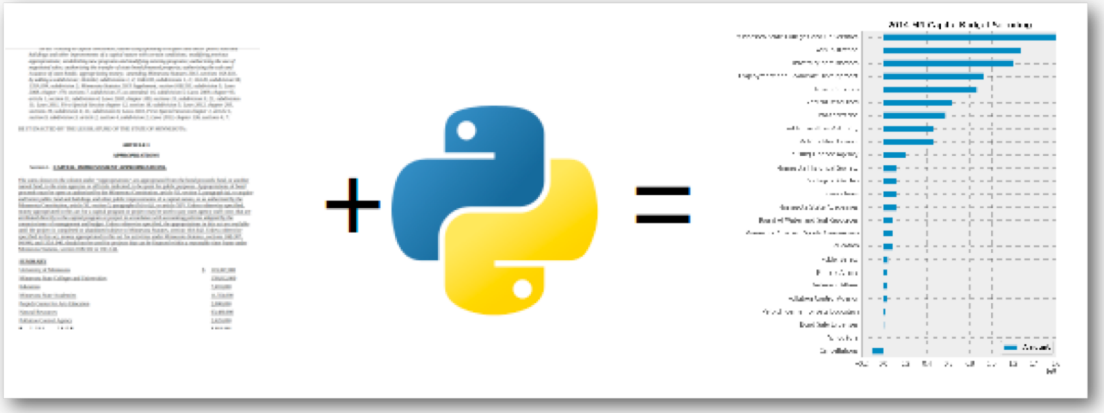
bit.ly/2Mmi6vH

Contents

- Define Scraping
- Python Basic Components (Data Types, Functions, Containers, For Loops)
- Applications:
  - *Beautiful Soup*
    - Demonstration with Follow Up
    - Practice Exercise
  - *Selenium*
    - Demonstration with Follow Up
    - Practice Exercise
- Things to keep in mind when scraping (robots.txt)
- Challenge Introduction
- Q & A

Web Scraping

- Used for extracting data from websites
- Automates the process of gathering data which is typically only accessible via a web browser
- Each website is naturally different, therefore each requires a slightly modified approach while scraping
- Not everything can be scrapped



Python Basics: Data Types

- Int e.g. 2,3,4
- Float e.g. 2.0,3.4, 4.3
- String e.g. “scraping ftw!”, ”John Doe”
- Boolean True, False
- Others (Complex, Unicode etc.)

Python Basics: Functions

- Functions start with “def” with the following format
  - *def function1(paramter1,parameter2):*

    answer = parameter1+paramter2

    return answer
- There are two ways to call functions:
  1. *Function1()*
     1. E.g. type(5) # int
  2. *Object.function1()*
     1. “python”.upper() # “PYTHON”
  - *Used under different circumstances (examples to come later)*

Python Basics: Lists

- Type of data container which is used to store multiple data at the same time
- Mutable (Can be changed)
- Comparable to R's vector
  - *E.g. list1 = [0,1,2,3,4]*
- Can contain items of varying data types
  - *E.g. list2 = [6,'harry', True, 1.0]*
- Indexing starts with 0
  - *E.g. list2[0] = 6*
- A list can be nested in another list
  - *E.g. [1 , [98,109], 6, 7]*
- Call the "append" function to add an item to a list
  - *E.g. list1.append(5)*

Python Basics: Dictionaries

- Collection of key-value pairs
- Very similar to JSON objects
- Mutable
- E.g. dict1 = {‘r’:4,’w’:9, ‘t’:5}
- Indexed with keys
  - *E.g. dict1[‘r’]*
- Keys are unique
- Values can be lists or other nested dictionaries
- A dictionary can also be nested into a list e.g. [{3:4,5:6}, 6,7]

Python Basics: For Loops

- Used for iterating over a sequence (a list, a tuple, a dictionary, a set, or a string)
- E.g.
  - *cities_list = [‘hong kong”, “new york”, “miami”]*
  - *for item in cities_list:*

```
        print(item)
# hong kong
# new york
# miami
```

Beautiful Soup

- Switch to Jupiter Notebook
  - *Open Anaconda*
  - *Launch Jupyter Notebook*
  - *Go to IMDB's 250 movies:*
    - https://www.imdb.com/search/title?genres=drama&groups=top_250&sort=user_rating,desc

Selenium

- Download the chrome web driver from
  - *http://chromedriver.chromium.org/downloads*
- Place the driver in your working directory
- Continue with Jupyter Notebook

Scraping Ethics

- Be respectful of websites' permissions
- View the website's robots.txt file to learn which areas of the site are allowed or disallowed from scraping
  - *You can access this file by replacing sitename.com in the following:* www.[sitename.com]/robots.txt
  - *E.g. imdb's robots txt can be found at https://www.imdb.com/robots.txt*
  - *You can also use https://canicrawl.com/ to check if a website allows scrapping*
- Don't overload website servers by sending too many requests. Use "time.sleep(xx)" function to delay requests.
  - *This will also prevent your IP address from being banned*

Interpreting the robots.txt file

- All pages of the website can be scrapped if you see the following:
  - *User-agent: **
  - *Disallow:*
- None of the pages of the website can be scrapped if you see the following:
  - *User-agent: **
  - *Disallow: /*
- Example from imdb →
  - *The sub-directories mentioned here are disallowed from being scrapped*

```
# robots.txt for https://www.imdb.com properties
User-agent: *
Disallow: /*/*/rg*/mediaviewer/rm*/tr
Disallow: /*/rg*/mediaviewer/rm*/tr
Disallow: /OnThisDay
Disallow: /ads/
Disallow: /ap/
Disallow: /find$
Disallow: /find/
Disallow: /gallery/rg*/mediaviewer/rm*/tr
Disallow: /list/ls*/_ajax
Disallow: /mymovies/
Disallow: /name/nm*/mediaviewer/rm*/tr
Disallow: /r/
Disallow: /register
Disallow: /registration/
Disallow: /search/name-text
Disallow: /search/title-text
Disallow: /title/tt*/mediaviewer/rm*/tr
Disallow: /tr/
Disallow: /tvschedule
Disallow: /updates
Disallow: /*/mediaviewer/*/tr
Disallow: /find
```

Take-home Challenge

- Scrape a fictional book store: http://books.toscrape.com/?
- Use what you have learned to create efficiently scrape the following data for Travel, Poetry, Art, Humor and Academic books:
  - *Book Title*
  - *Product Description*
  - *Price (excl. tax)*
  - *Number of Reviews*
- Store all of the data in a single Pandas DataFrame
- The most efficient scraper will be awarded with a prize
- Deadline for submissions are in a week from today, 4/18/2019 11:59pm

Resources

- https://github.com/devkosal/scraping_tutorial
  - *All code provided in this lecture can be found here*
- http://toscrape.com/
  - *Great sample websites to perform beginner to intermediate scrapping on*
- https://www.edx.org/course/introduction-to-computer-science-and-programming-using-python-0
  - *Introduction to Computer Science using Python*
  - *Highly recommended course on learning Python and CS form scratch*
- https://www.promptcloud.com/blog/how-to-read-and-respect-robots-file/
  - *Further reading on interpreting robots.txt*
- https://canicrawl.com/
  - *Check scraping permissions for any website*