Week 9 - Scraping Web Pages with BeautifulSoup

Overview

A very important skill to have is the ability to scrape data from the web. In many situations, you will need to pull information from some web site, and the data is not presented to you in a convenient way, or there is no official API that the site offers. To help hone these skills, for this week's assignment, we will be utilizing BeautifulSoup to gather data from various web pages.

Make sure to create a github repository for this assignment named **IS211_Assignment9**. All development should be done in this repository.

Useful Reminders

1. Read the assignment over a few times. At least twice. It always helps to have a clear picture of the overall assignment when understanding how to build a solution.
2. Think about the problem for a while, and even try writing or drawing a solution using pencil and paper or a whiteboard.
3. Before submitting the assignment, review the "Functional Requirements" section and make sure you hit all the points. This will not guarantee a perfect score, however.

Part I - CBS Football Stats

For this part of the assignment, you will scrape some football stats from the CBS webpage.

1. Go to the CBS NFL Stats webpage, located at http://www.cbssports.com/nfl/stats
2. For this example, lets look at the league leaders in touchdowns. Under "PLAYER STATISTICS", click on the Touchdowns links
3. Change the drop down to regular
4. Write a script called "football_stats.py" that will load this URL, parse it using BeautiulSoup, and output the list of top 20 players, including the player's position, team and total number of touchdowns

Part II - Stock Data

Another important task in the real world is to download stock data for future analysis. There are many services that you can get this kind of data from in a convenient format, but many of these services are for pay. We can do better! For this part of the assignment, we'll scrape stock data for Apple from Yahoo! finance.

1. To get data about Apple's stock, we'll use Yahoo! Finance, located at http://finance.yahoo.com
2. Type in Apples stock symbol, AAPL, into the search bar. This brings up AAPL's summary page. From here, click on the historical prices link in the sidebar, which should load up a URL like http://finance.yahoo.com/q/hp?s=AAPL+Historical+Prices
    a. Food for thought: if you wanted to scrape data for another stock, what would the URL look like?
3. Write a script called "apple_stock.py" that will load this URL, parse it using BeautiulSoup, and output the close price and date for all the dates shown on the page

Part III - Weather Data

As the last part of the assignment, we will scrape some weather data from the Weather Underground. For example, we will scrape the Monthly temperature (actual and forecasted) for January 2015, which is located here: http://www.wunderground.com/history/airport/KNYC/2015/4/1/MonthlyHistory.html. Write a script called "weather.py" that will load this URL, parse it using BeautiulSoup, and:

1. Display the actual temperatures for the days of the month that have passed
2. Display the forecasted temperatures for the days of the month that have not passed yet