We’re going to look at how to gather stock prices and user sentiment data using Python! Shouldn’t be too hard so let’s get started!

**Installing Python**

First thing we’re going to want to do is install Python so we can make API calls that will give us the data that we need. There’s several ways of going about this. I’m going to describe how to do so using Notepad++ instead of IDLE (the default editor for Python) because Notepad++ allows for easier readability and better clarification on indentation.

Firstly, click this link here to be taken to a python download webpage: <https://www.python.org/downloads/>

Be sure to download the latest version of Python! (3.5.2 at the time of this writing).

We’re also going to download Notepad++ so you can make edits to our program in case you want to get more sentiment or stock data from quandl.

Click this link to be taken to the Notepad++ download page: <https://notepad-plus-plus.org/download/v7.2.2.html>

Either download the 32-bit or 64-bit installer based off whatever system type your computer is (you can find this if you go to **My Computer->System properties** and look for **System type:** under the **System** window).

Once you have those both installed, we can move onto Quandl.

**Quandl API**

We’re going to be using a service called Quandl to get our stock prices and user sentiment data. We’re going to need an API key in order to get the data which will require us to create an account, which is quick and free.

First, head to this link here to create an account: <https://www.quandl.com/>

Go to **Sign In->Create One->(input a username, email, and password)** and that’s it! Quandl should now generate an API key for you to use in extracting data (don’t worry about writing it down/remembering it. As long as you’re signed into Quandl, it’ll automatically use your API key when you request data).

Now that we have our API key, let’s look up some stock prices for Microsoft. Click here for their stock data: <https://www.quandl.com/data/EOD/MSFT-Microsoft-Corporation-MSFT-Stock-Prices-Dividends-and-Splits>

At this point, we’re only interested in the data for the year of 2014, so in our dates textbox be sure to change the dates to show a range from **Jan 1, 2014 – Jan 1, 2015**

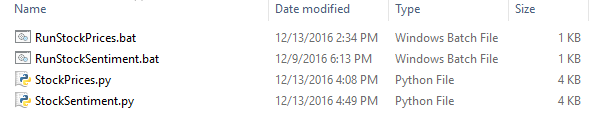
Now that this is done, by clicking the **Download** button in the upper-right corner, and clicking **JSON**. It should generate something that looks like this: 

Doesn’t mean a whole lot to us right now, but through Python we’re going to make something of it, so for now just keep a tab open with this url so we can reference it in a bit.

Do the same thing for Microsoft’s stock sentiment data with the same date range and keep the JSON url open as well for future referencing. The link can be found here: <https://www.quandl.com/data/AOS/MSFT-AlphaOne-Sentiment-Impact-Scores-for-Microsoft-Corporation-MSFT>

**Using Python**

Now that we have the stock and sentiment data in a JSON format, we’re going to use Python to extract the information and write it to a .csv so we can make sense of the data.

Open up the .zip file the instructor has sent out to the class. Inside are 4 files: 

Highlight only the documents with the **.py** extension and open them with Notepad++. The **.bat** documents exist to make running our Python code simpler.

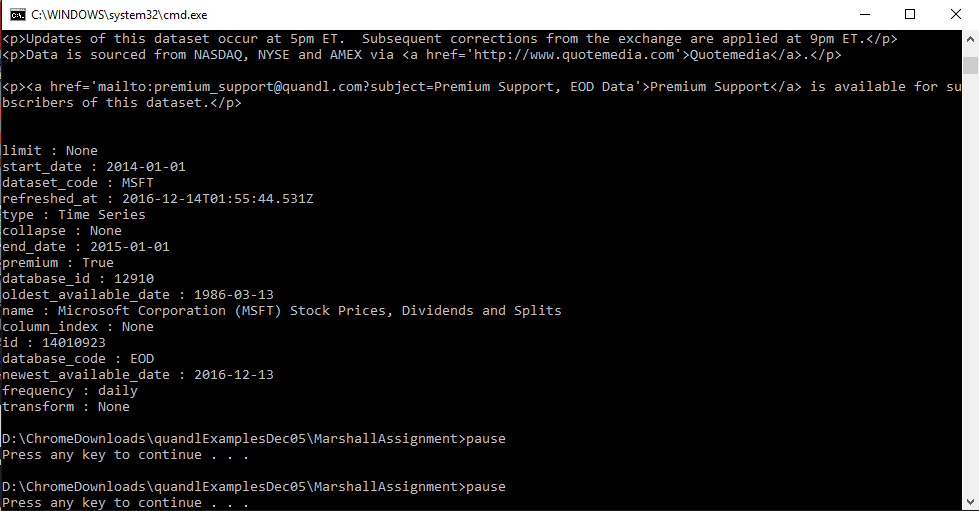
Navigate over to **StockPrices.py**

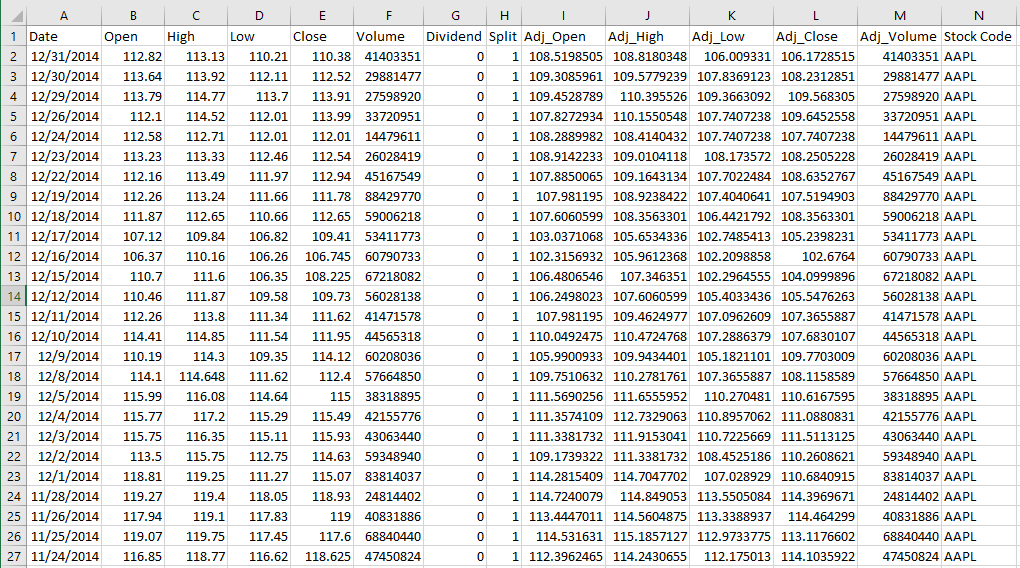
There are currently 5 variables that are storing JSON data urls. If you were to run the code as is, it would generate a .csv file with stock information for the 5 companies currently listed.

We’re now going to add Microsoft’s information to be parsed by this Python program. Create a new variable under **URL\_AmericanExpress** called **URL\_Microsoft** and set the value equal to the url we created earlier for Microsoft’s stock prices.

Once this is done, we now need to have the class **URLInput** call the url so we can get information from Quandl’s API. Scroll down to the end of the program and under **Cisco = URL()** add a line and enter **Microsoft = URL()**. Once this is done, under **AmericanExpress.URLInput(URL\_AmericanExpress)** add a line and enter **Microsoft.URLInput(URL\_Microsoft)**

Be sure to save the document at this point. Now close Notepad++ and double click on **RunStockPrices.bat** (this will run your Python program)

Once this finishes, you should see: 

Also, a new file appeared in your directory called **StockPrices.csv** which contains information that should be a little more readable to us: 

That’s it (for getting stock data anyways)! In order to get user sentiment data, perform the exact same steps, except use the url we generated for Microsoft sentiment data and put it into **StockSentiment.py** using the same method we used for our Stock Price Python program! This will generate a new document called **StockSentiment.csv** that will have our sentiment data in a readable format thanks to the power of Python!