This exercise will require you to pull some data from the Qunadl API. Qaundl is currently the most widely used aggregator of financial market data.

As a first step, you will need to register a free account on the http://www.quandl.com (http://www.quandl.com) website.

After you register, you will be provided with a unique API key, that you should store:

In [28]:

```
# Store the API key as a string - according to PEP8, constants are always named in all upper case
API_KEY = 'iusgGJ_BJosyK3VsC9uD'
```

Qaundl has a large number of data sources, but, unfortunately, most of them require a Premium subscription. Still, there are also a good number of free datasets.

For this mini project, we will focus on equities data from the Frankfurt Stock Exhange (FSE), which is available for free. We'll try and analyze the stock prices of a company called Carl Zeiss Meditec, which manufactures tools for eye examinations, as well as medical lasers for laser eye surgery: https://www.zeiss.com/meditec/int/home.html (https://www.zeiss.com/meditec/int/home.html). The company is listed under the stock ticker AFX_X.

You can find the detailed Quandl API instructions here: https://docs.quandl.com/docs/time-series (https://docs.quandl.com/docs/time-series)

While there is a dedicated Python package for connecting to the Quandl API, we would prefer that you use the *requests* package, which can be easily downloaded using *pip* or *conda*. You can find the documentation for the package here: http://docs.python-requests.org/en/master/ (http://docs.python-requests.org/en/master/)

Finally, apart from the *requests* package, you are encouraged to not use any third party Python packages, such as *pandas*, and instead focus on what's available in the Python Standard Library (the *collections* module might come in handy: https://pymotw.com/3/collections/ (https://pymotw.com/3/collections/)). Also, since you won't have access to DataFrames, you are encouraged to us Python's native data structures - preferably dictionaries, though some questions can also be answered using lists. You can read more on these data structures here: https://docs.python.org/3/tutorial/datastructures.html (https://docs.python.org/3/tutorial/datastructures.html)

Keep in mind that the JSON responses you will be getting from the API map almost one-to-one to Python's dictionaries. Unfortunately, they can be very nested, so make sure you read up on indexing dictionaries in the documentation provided above.

In [29]:

```python
# First, import the relevant modules

import json
import requests
import subprocess
import os
from pprint import pprint
```

In [30]:

```python
# Now, call the Quandl API and pull out a small sample of the data (only one day) to get a
glimpse
# into the JSON structure that will be ret
url = 'https://www.quandl.com/api/v3/datasets/FSE/AFX_X/data.json'
r = requests.get(url)
```

In [31]:

```python
r.json()['dataset_data'].keys()
```

Out[31]:

```
dict_keys(['limit', 'transform', 'column_index', 'column_names', 'start_dat
e', 'end_date', 'frequency', 'data', 'collapse', 'order'])
```

In [32]:

```python
r.json()['dataset_data']['column_names']
```

Out[32]:

```
['Date',
 'Open',
 'High',
 'Low',
 'Close',
 'Change',
 'Traded Volume',
 'Turnover',
 'Last Price of the Day',
 'Daily Traded Units',
 'Daily Turnover']
```

These are your tasks for this mini project:

1. Collect data from the Franfurt Stock Exchange, for the ticker AFX_X, for the whole year 2017 (keep in mind that the date format is YYYY-MM-DD).
2. Convert the returned JSON object into a Python dictionary.
3. Calculate what the highest and lowest opening prices were for the stock in this period.
4. What was the largest change in any one day (based on High and Low price)?
5. What was the largest change between any two days (based on Closing Price)?
6. What was the average daily trading volume during this year?
7. (Optional) What was the median trading volume during this year. (Note: you may need to implement your own function for calculating the median.)

1) Collect data from the Franfurt Stock Exchange, for the ticker AFX_X, for the whole year 2017 (keep in mind that the date format is YYYY-MM-DD).

In [33]:

```
url = 'https://www.quandl.com/api/v3/datasets/FSE/AFX_X/data.json?start_date=2017-01-01&end_date=2017-12-31'
r = requests.get(url)
```

In [34]:

```
print('start_date is: ', r.json()['dataset_data']['start_date'])
```

```
start_date is:  2017-01-01
```

2) Convert the returned JSON object into a Python dictionary.

In [35]:

```
json_data = r.json()
```

In [36]:

```
print(type(json_data))
```

```
<class 'dict'>
```

3) Calculate what the highest and lowest opening prices were for the stock in this period.

In [37]:

```python
open_prices = [daily[1] for daily in json_data['dataset_data']['data'] if daily[1] != None
]
max_open = str(max(open_prices))
min_open = str(min(open_prices))
print('The highest opening price was $' + max_open + ' and the lowest opening price was $'
+ min_open + ' in 2017.')
```

```
The highest opening price was $53.11 and the lowest opening price was $34.0 i
n 2017.
```

4) What was the largest change in any one day (based on High and Low price)?

In [38]:

```python
daily_change = [(x[2]-x[3]) for x in json_data['dataset_data']['data']]
print('largest change in one day is:', round(max(daily_change),2), 'in 2017')
```

```
largest change in one day is: 2.81 in 2017
```

5) What was the largest change between any two days (based on Closing Price)?

In [39]:

```python
close_prices = [daily[4] for daily in json_data['dataset_data']['data'] if daily[4] != Non
e]
change = [x - close_prices[i-1] for i,x in enumerate(close_prices)][1:]
pos_change = [abs(num) for num in change]
largest_close_change = str(round(max(pos_change), 2))
print('The largest change between any two consecutive days based on Closing Price was $' +
str(largest_close_change) + ' in 2017.')
```

```
The largest change between any two consecutive days based on Closing Price wa
s $2.56 in 2017.
```

6) What was the average daily trading volume during this year?

In [41]:

```python
avg_vol = sum(x[6] for x in json_data['dataset_data']['data'])/len(json_data['dataset_dat
a']['data'])
print('average daily trading volume during this year is: ', avg_vol)
```

```
average daily trading volume during this year is:  89124.33725490196
```

7) What was the median trading volume during this year. (Note: you may need to implement your own function for calculating the median.)

In [27]:

```python
med_vol = sorted(x[6] for x in json_data['dataset_data']['data'])[len(json_data['dataset_d
ata']['data']) - round(len(json_data['dataset_data']['data'])*0.5)]
print('The median trading volume during this year: ',  med_vol)
```

The median trading volume during this year: 76286.0