Data Representation
Lab 3: Trains

Lecturer: Andrew Beatty

Write a program that stores the data for all trains in Ireland in a csv file
Use the Irish rail API
http://api.irishrail.ie/realtime/realtime.asmx/getCurrentTrainsXML
to retrieve the data.

Then as an exercise only store trains that are south of Dublin:
For data sets of this size I would normally get all the data, and perform analysis (deletions) later.

Get the data:

1. Go to the URL and check that it works, have a quick look at the format of the XML.
2. Create a python program that reads the XML from the URL and prints it out, using beautifulSoup. Check it does retrieve the data.

```
import requests
import csv
from bs4 import BeautifulSoup

url = "http://api.irishrail.ie/realtime/realtime.asmx/get
CurrentTrainsXML"
page = requests.get(url)

soup = BeautifulSoup(page.content, 'xml')
print (soup.prettify())
```

3. Once you are happy this works, comment out the print statement.
4. Modify the program to print out each of the trains. I.e. find the listings and iterate through them to print each out. Check it works

```
listings = soup.findAll("objTrainPositions")

    for listing in listings:
        print(listing)
```

5. Comment out the print and modify the program so that it prints out the latitudes

```
      print(listing.TrainLatitude.string)
         # or
         # print(listing.find('TrainLatitude').string)
```

6. Ok. Let's now store this one property into a CSV:
    a. Before the for loop open the CSV file, I am using **with**, so make sure that you indent the for loop so that it is in the with block.

```
with  open('week03_train.csv', mode='w') as train_file:
    train_writer = csv.writer(train_file, delimiter='\t', quotechar='"', quoting=csv.QUOTE_MINIMAL)

    listings = soup.findAll("objTrainPositions")
    for listing in listings:
        #print(listing)
```

    b. In the for loop now create an array called entryList, append in the latitude and store that in the CSV.

```
entryList = []
entryList.append(listing.find('TrainLatitude').string)
train_writer.writerow(entryList)
```

7. Test this by running the program and seeing if the CSV file you made stores all the latitudes.
8. The problem asked for all the properties in the XML file, so we could just repeat the append line for each of the properties, this will work, but it makes the code long and I am lazy so:
    a. At the top of the program make an array called retrieveTags that will store all the names of the tags we want to retrieve.

```
retrieveTags=['TrainStatus',
              'TrainLatitude',
              'TrainLongitude',
              'TrainCode',
              'TrainDate',
              'PublicMessage',
              'Direction'
              ]
```

    b. Then change the **append** line to be a for loop that iterates through these tag names.

```
            entryList = []
            for retrieveTag in retrieveTags:
                  #print (listing.find(retrieveTag).string)
                  entryList.append(listing.find(retrieveTag).string)
            train_writer.writerow(entryList)
```

Reduce the dataset size

For a dataset this size I would normally retrieve it all and perform my analysis on it and reduce it later. But for an exercise I want to reduce the dataset while we are retrieving it, this is handy for very large datasets.
We are only going to store trains that are south of Dublin, IE have a latitude less then that of Dublin (approx. 53.4)

9. In the for loop of the listings before we make the entry list get the latitude of this train and store it as a float, then check if it is less then the latitude of Dublin (approx. 53.4)

```
for listing in listings:
        lat =float( listing.TrainLatitude.string)
        if (lat < 53.4):

            entryList = []
            for retrieveTag in retrieveTags:
                #print (listing.find(retrieveTag).string)
                entryList.append(listing.find(retrieveTag).string)
            train_writer.writerow(entryList)
```

That's it, take a break