Data Representation
Lab 2: Trains

Lecturer: Andrew Beatty

Write a program that prints the data for all trains in Ireland to the console.
(You can modify this to store all the data 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 have a train code that starts with D:
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 minidom. Check it does retrieve the data.

```
import requests
import csv
from xml.dom.minidom import parseString

url = "http://api.irishrail.ie/realtime/realtime.asmx/getCurrentTrainsXML"
page = requests.get(url)
doc = parseString(page.content)
# check it works
print (doc.toprettyxml()) #output to console comment this out once you know it works

# if I want to store the xml in a file. You can comment this out later
with open("trainxml.xml","w") as xmlfp:
    doc.writexml(xmlfp)
```

3. Once you are happy this works, comment out the print statement.

4. Modify the program to print out each of the trainscodes. I.e. find the listings and iterate through them to print each traincode out. Check it works

```
objTrainPositionsNodes = doc.getElementsByTagName("objTrainPositions")
for objTrainPositionsNode in objTrainPositionsNodes:
    traincodenode = objTrainPositionsNode.getElementsByTagName("TrainCode").item(0)
    traincode = traincodenode.firstChild.nodeValue.strip()
    print (traincode)
```

5. Comment out the print and modify the program so that it prints out the latitudes
6. Ok. Let's now store this one property into a CSV:
    a. Before the for loop open the CSV file, using **with**, so make sure that you indent the for loop so that it is in the with block.

```
# I had an issue with blank lines in the file so found solution at
# https://stackoverflow.com/questions/3348460/csv-file-written-with-python-has-blank-lines-between-each-r
# adding the newline= '' parameter
with  open('week03_train.csv', mode='w', newline='') as train_file:
    train_writer = csv.writer(train_file, delimiter='\t', quotechar='"', quoting=csv.QUOTE_MINIMAL)


      objTrainPositionsNodes = doc.getElementsByTagName("objTrainPositions")
      for objTrainPositionsNode in objTrainPositionsNodes:
            traincodenode = objTrainPositionsNode.getElementsByTagName("TrainCode").item(0)
             traincode = traincodenode.firstChild.nodeValue.strip()
```

    b. In the for loop now create an array called dataList, append in the traincode and store that in the CSV. (ie you can replace the print(traincode)

```
dataList = []
dataList.append(traincode)
train_writer.writerow(dataList)
```

7. Test this by running the program and seeing if the CSV file you made stores all the traincodes.

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.

```
dataList = []
for retrieveTag in retrieveTags:
      datanode = objTrainPositionsNode.getElementsByTagName(retrieveTag).item(0)
      dataList.append(datanode.firstChild.nodeValue.strip())

train_writer.writerow(dataList)
```

9. As an exercise only store the trains whose traincode starts with a D

That's it, take a break