In [1]:
```
!conda install -c anaconda lxml --yes
!conda install -c anaconda bs4 --yes
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/conda/envs/python

  added / updated specs:
    - lxml


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2020.6.24  |                0         133 KB  anaconda
    certifi-2020.6.20          |           py36_0         160 KB  anaconda
    libxslt-1.1.33             |       h7d1a2b0_0         577 KB  anaconda
    lxml-4.5.1                 |   py36hefd8a0e_0         1.4 MB  anaconda
    openssl-1.1.1g             |       h7b6447c_0         3.8 MB  anaconda
    ------------------------------------------------------------
                                           Total:         6.0 MB

The following NEW packages will be INSTALLED:

  libxslt            anaconda/linux-64::libxslt-1.1.33-h7d1a2b0_0
  lxml               anaconda/linux-64::lxml-4.5.1-py36hefd8a0e_0

The following packages will be UPDATED:

  ca-certificates    conda-forge::ca-certificates-2020.6.2~ --> anaconda::ca-
certificates-2020.6.24-0

The following packages will be SUPERSEDED by a higher-priority channel:

  certifi            conda-forge::certifi-2020.6.20-py36h9~ --> anaconda::cer
tifi-2020.6.20-py36_0
  openssl            conda-forge::openssl-1.1.1g-h516909a_0 --> anaconda::ope
nssl-1.1.1g-h7b6447c_0



Downloading and Extracting Packages
lxml-4.5.1           | 1.4 MB    | ##################################### | 10
0%
openssl-1.1.1g       | 3.8 MB    | ##################################### | 10
0%
ca-certificates-2020 | 133 KB    | ##################################### | 10
0%
certifi-2020.6.20    | 160 KB    | ##################################### | 10
0%
libxslt-1.1.33       | 577 KB    | ##################################### | 10
0%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting package metadata (current_repodata.json): done
```

```
Solving environment: done

## Package Plan ##

  environment location: /home/jupyterlab/conda/envs/python

  added / updated specs:
    - bs4


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    beautifulsoup4-4.9.1       |           py36_0         168 KB  anaconda
    bs4-4.9.1                  |                0           4 KB  anaconda
    soupsieve-2.0.1            |             py_0          33 KB  anaconda
    ------------------------------------------------------------
                                           Total:         204 KB

The following NEW packages will be INSTALLED:

  beautifulsoup4     anaconda/linux-64::beautifulsoup4-4.9.1-py36_0
  bs4                anaconda/noarch::bs4-4.9.1-0
  soupsieve          anaconda/noarch::soupsieve-2.0.1-py_0



Downloading and Extracting Packages
soupsieve-2.0.1      | 33 KB     | ##################################### | 10
0%
beautifulsoup4-4.9.1 | 168 KB    | ##################################### | 10
0%
bs4-4.9.1            | 4 KB      | ##################################### | 10
0%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

In [ ]:

In [2]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
import numpy as np

# get html content
response = requests.get("https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M")
soup = BeautifulSoup(response.text, 'lxml')

#extract wikitable
data = []
columns = []
table = soup.find(class_='wikitable')
for index, tr in enumerate(table.find_all('tr')):
    section = []
    for td in tr.find_all(['th','td']):
        section.append(td.text.rstrip())

    #First row of data is the header
    if (index == 0):
        columns = section
    else:
        data.append(section)


#convert list into Pandas DataFrame
toronto_df = pd.DataFrame(data = data,columns = columns)
assigned =  toronto_df['Borough']!="Not assigned"
assigned_df = toronto_df[assigned]

assigned_df.head()
```

Out[2]:

	Postal Code	Borough	Neighborhood
2	M3A	North York	Parkwoods
3	M4A	North York	Victoria Village
4	M5A	Downtown Toronto	Regent Park, Harbourfront
5	M6A	North York	Lawrence Manor, Lawrence Heights
6	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government

In [13]:
```
#len(set(assigned_df['Postal Code']))
assigned_df.shape
```

Out[13]:
```
(103, 3)
```

In [1]:
```
!conda install -c conda-forge geocoder --yes
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

In [3]:
```
import pandas as pd
import io
import requests
url="https://raw.githubusercontent.com/cs109/2014_data/master/countries.csv"
url="https://cocl.us/Geospatial_data"
s=requests.get(url).content
location_df=pd.read_csv(io.StringIO(s.decode('utf-8')))
location_df.head()
location_df.shape
```

Out[3]: (103, 3)

In [4]:
```
df_loc = pd.merge(assigned_df, location_df, on='Postal Code')
df_loc.head()
```

Out[4]:

	Postal Code	Borough	Neighborhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.718518	-79.464763
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494

In [32]:
```
#Extract subset of Borough including word - Toronto
print("extract Borough including word - Toronto")
toronto_data = df_loc[df_loc['Borough'].str.contains('Toronto')].reset_index(d
rop=True)
toronto_data.head()
```

```
extract Borough including word - Toronto
```

Out[32]:

	Postal Code	Borough	Neighborhood	Latitude	Longitude
0	M5A	Downtown Toronto	Regent Park, Harbourfront	43.654260	-79.360636
1	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.662301	-79.389494
2	M5B	Downtown Toronto	Garden District, Ryerson	43.657162	-79.378937
3	M5C	Downtown Toronto	St. James Town	43.651494	-79.375418
4	M4E	East Toronto	The Beaches	43.676357	-79.293031

In [13]:
```
!conda install -c conda-forge geopy --yes
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

In [31]:
```
#get location latitude and longitude of Toronto
from  geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="dnipro")
location = geolocator.geocode("Toronto")
print(location.address)
print("get location latitude and longitude of toronto")
print((location.latitude, location.longitude))
latitude = location.latitude
longitude = location.longitude
```

```
Toronto, Golden Horseshoe, Ontario, M5H 2N2, Canada
get location latitude and longitude of toronto
(43.6534817, -79.3839347)
```

In [33]:
```
#Create map of Toronto
#visualize boroughs including word "Toronto" and how they cluster together
import folium
# create map of Toronto using latitude and longitude values
map_toronto = folium.Map(location=[latitude, longitude], zoom_start=11)

# add markers to map
for lat, lng, label in zip(toronto_data['Latitude'], toronto_data['Longitude'
], toronto_data['Neighborhood']):
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=5,
        popup=label,
        color='blue',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7,
        parse_html=False).add_to(map_toronto)

map_toronto
```

Out[33]: