In [1]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [2]:
```
url = "https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M"
urltext = requests.get(url).text
data = BeautifulSoup(urltext, 'lxml')

# creat a new Dataframe
columnlabel = ['Postalcode','Borough','Neighborhood']
toronto = pd.DataFrame(columns = columnlabel)

# loop through to find postcode, borough, neighborhood
content = data.find('div', class_='mw-parser-output')
table = content.table.tbody
postcode = 0
borough = 0
neighborhood = 0

for tr in table.find_all('tr'):
    i = 0
    for td in tr.find_all('td'):
        if i == 0:
            postcode = td.text
            i = i + 1
        elif i == 1:
            borough = td.text
            i = i + 1
        elif i == 2:
            neighborhood = td.text.strip('\n').replace(']','')
    toronto = toronto.append({'Postalcode': postcode,'Borough': borough
,'Neighborhood': neighborhood},ignore_index=True)
toronto = toronto[toronto.Borough!='Not assigned']
toronto = toronto[toronto.Borough!= 0]
toronto.reset_index(drop = True, inplace = True)
i = 0
```

```
for i in range(0,toronto.shape[0]):
    if toronto.iloc[i][2] == 'Not assigned':
        toronto.iloc[i][2] = toronto.iloc[i][1]
        i = i+1

df = toronto.groupby(['Postalcode','Borough'])['Neighborhood'].apply(',
 '.join).reset_index()
# drop burroughs with none assigned
df = df.dropna()
empty = 'Not assigned'
df = df[(df.Postalcode != empty ) & (df.Borough != empty) & (df.Neighbo
rhood != empty)]

# group neighborhoods with like burroughs
def neighborhood_list(grouped):
    return ', '.join(sorted(grouped['Neighborhood'].tolist()))

grp = df.groupby(['Postalcode', 'Borough'])
df2 = grp.apply(neighborhood_list).reset_index(name='Neighborhood')
df2.head()
```

Out[2]:

	Postalcode	Borough	Neighborhood
0	M1A\n	Not assigned\n	Not assigned\n
1	M1B\n	Scarborough\n	Malvern, Rouge
2	M1C\n	Scarborough\n	Rouge Hill, Port Union, Highland Creek
3	M1E\n	Scarborough\n	Guildwood, Morningside, West Hill
4	M1G\n	Scarborough\n	Woburn

In [3]:
```
print(df2.shape)
```

```
(180, 3)
```

In [ ]:
```

```