"""Create map to show criminal incidents in San Francisco from 2003 to 2015

and allow user to focus on specific neighborhood or category of incident"""

#import SFPD incidents API

#open data

#ask for raw input and give choice of neighborhood to look at or crime category

#parse the data to show only incident category (choose certain options) and location

#cull between neighborhood or crime category

#gather the latitude and longitudes of crimes

#pass data into mapbox

#plot the data on map of SF using (mapbox) - longitude/latitude

#different crime categories will be represented with different color dots on map?

#user can see map as a whole or type (zoom in on?) specific neighborhood to see in detail? (depends on mapbox)

#user can view map by specific crime categories or choose specific neighborhood to focus on

from urllib2 import urlopen

import json

#load api data into a variable to pass as parameter when creating a dictionary of coordinates

def import\_api():

api\_url = 'https://data.sfgov.org/resource/gxxq-x39z.json'

response = urlopen(api\_url)

json\_obj = json.load(response)

return json\_obj

#create function to load api data into dictionary which will be used to load into html

def create\_crime\_dictionary(crimes):

# crimes = []

# for incident in json\_obj:

# d = {}

# d['type'] = 'Feature'

# latitude = float(incident['y'])

# longitude = float(incident['x'])

# d['geometry'] = {'type':'Point', 'coordinates': [longitude,latitude]}

# crimes.append(d)

#Find javascript/mapbox code to display coordinates in a webpage with html

html = """<!DOCTYPE html>

<html>

<head>

<meta charset=utf-8 />

<title></title>

<script src='https://api.mapbox.com/mapbox.js/v2.4.0/mapbox.js'></script>

<link href='https://api.mapbox.com/mapbox.js/v2.4.0/mapbox.css' rel='stylesheet' />

<style>

body { margin:0; padding:0; }

.map { position:absolute; top:0; bottom:0; width:100%; }

</style>

</head>

<body>

<div id='map\_geo' class='map'> </div>

<script>

var geojson = """ + json.dumps(crimes) + """;

// call map api using javascript code and load my python code into usable variable (geojson)

L.mapbox.accessToken = 'pk.eyJ1IjoiYmFuZXJqZWVkIiwiYSI6ImNpc3doNzRpYzA1NmQyemtocTh5ZzkxMzYifQ.-w2jfFGPwju-dMjY-3NjMw';

var mapGeo = L.mapbox.map('map\_geo', 'mapbox.light').setView([37.76, -122.4], 12);

var myLayer = L.mapbox.featureLayer().setGeoJSON(geojson).addTo(mapGeo);

//mapGeo.scrollWheelZoom.disable();

</script>

</body>

</html>

"""

#writing the markup to a file which creates the webpage

with open('maptest3.html', mode = 'w') as my\_file:

my\_file.write(html)

#my\_file.write(js)

def main():

#Introduce user to map

print "Map of Criminal Incidents in San Francisco from 2003 - 2015"

#Ask user to view map by crime category or not

user\_choice = (raw\_input("Do you want to view a specific category of incidents? (Y or N) ")).upper()

if user\_choice == 'Y':

#prompt user to choose which category of criminal incidents to view

l = ["ARRESTS", "ARSON", "ASSAULTS", "BURGLARIES", "ROBBERIES", "SHOOTINGS", "THEFT", "VANDALISM"]

for i in range(len(l)):

print "Enter ", i, "to view", l[i], "in San Francisco"

# for i in l:

print "Press 1 to view ARRESTS in San Francisco"

print "Press 2 to view ARSON in San Francisco"

print "Press 3 to view ASSAULTS in San Francisco"

print "Press 4 to view BURGLARIES in San Francisco"

print "Press 5 to view ROBBERIES in San Francisco"

print "Press 6 to view SHOOTINGS in San Francisco"

print "Press 7 to view THEFT in San Francisco"

print "Press 8 to view VANDALISM in San Francisco"

category = l[int(raw\_input())]

#display map based on user input by modifying geojson variable

#print static map of arrest points using mapbox/geojson - pull all arrests from API and plot points

crimes = []

for obj in import\_api():

if category == obj["category"]:

d = {}

d['type'] = 'Feature'

latitude = float(obj['y'])

longitude = float(obj['x'])

d['geometry'] = {'type':'Point', 'coordinates': [longitude,latitude]}

crimes.append(d)

create\_crime\_dictionary(crimes)

if \_\_name\_\_ == '\_\_main\_\_':

main()