OPEN STREET MAP CASE STUDY PROJECT

MAP AREA San Francisco, CA, USA

https://mapzen.com/data/metro-extracts/your-extracts/99d97e82a282 (https://mapzen.com/data/metro-extracts/your-extracts/99d97e82a282) I chose this area with regards to the recent events, I am proud of how California is at the frontier to right what is wrong. Also, the idea of contributing to the community is pretty cool.

In [1]:

```
import xml.etree.cElementTree as ET
import pprint
import re
import codecs
import json
```

In [2]:

```
import xml.etree.ElementTree as ET  # Use cElementTree or lxml if too slow

OSM_FILE = "san-francisco_california.osm"
SF = "sanfran.osm.json"

k = 1

def get_element(osm_file, tags=('node', 'way', 'relation')):
    """Yield element if it is the right type of tag

    Reference:
    http://stackoverflow.com/questions/3095434/inserting-newlines-in-xml-file-
generated-via-xml-etree-elementtree-in-python
    """
    context = iter(ET.iterparse(osm_file, events=('start', 'end')))
    _, root = next(context)
    for event, elem in context:
        if event == 'end' and elem.tag in tags:
            yield elem
            root.clear()


with open(SF, 'wb') as output:
    output.write('<?xml version="1.0" encoding="UTF-8"?>\n')
    output.write('<osm>\n  ')

    # Write every kth top level element
    for i, element in enumerate(get_element(OSM_FILE)):
        if i % k == 0:
            output.write(ET.tostring(element, encoding='utf-8'))

    output.write('</osm>')
```

In [3]:

```
from collections import defaultdict

def count_tags(filename):

    counts = defaultdict(int)
    for line in ET.iterparse(filename):
        current = line[1].tag
        counts[current] += 1
    return counts


def test():

    tags = count_tags(SF)
    pprint.pprint(tags)



if __name__ == "__main__":
    test()
```

```
defaultdict(<type 'int'>, {'node': 1509564, 'nd': 1879946, 'member': 21618,
 'tag': 462518, 'relation': 2414, 'way': 156556, 'osm': 1})
```

In [4]:

```
lower = re.compile(r'^([a-z]|_)*$')
lower_colon = re.compile(r'^([a-z]|_)*:([a-z]|_)*$')
problemchars = re.compile(r'[=\+/&<>;\'"\?%#$@\,\. \t\r\n]')

def key_type(element, keys):
    if element.tag == "tag":

        if lower.search(element.attrib['k']):
            keys['lower'] += 1
        elif lower_colon.search(element.attrib['k']):
            keys['lower_colon'] += 1
        elif problemchars.search(element.attrib['k']):
            keys['problemchars'] += 1
        else:
            keys['other'] += 1

    return keys



def process_map(filename):
    keys = {"lower": 0, "lower_colon": 0, "problemchars": 0, "other": 0}
    for _, element in ET.iterparse(filename):
        keys = key_type(element, keys)

    return keys



def test_data():
    # You can use another testfile 'map.osm' to look at your solution
    # Note that the assertions will be incorrect then.
    keys = process_map(SF)
    pprint.pprint(keys)



if __name__ == "__main__":
    test_data()
```

```
{'lower': 300585, 'lower_colon': 156901, 'other': 4906, 'problemchars': 126}
```

In [5]:

```
def count_street(filename):
    streets = {}
    for event, elem in ET.iterparse(filename, events=('start', 'end')):
        if event == 'end':
            key = elem.attrib.get('k')
            if key == 'addr:street':
                street = elem.attrib.get('v')
                if street not in streets:
                    streets[street] = 1
                else:
                    streets[street] += 1
    return streets


postcodes = count_street(OSM_FILE)
sorted_by_occurrence = [(k, v) for (v, k) in sorted([(value, key) for (key, va
lue) in postcodes.items()], reverse=True)]

print 'street values and occurrence in San-Francisco_california.osm:\n'
pprint.pprint(sorted_by_occurrence)
```

```
street values and occurrence in San-Francisco_california.osm:

[('Irving Street', 593),
 ('9th Avenue', 542),
 ('14th Avenue', 432),
 ('Page Street', 408),
 ('10th Avenue', 395),
 ('12th Avenue', 392),
 ('Funston Avenue', 386),
 ('11th Avenue', 369),
 ('Taraval Street', 341),
 ('8th Avenue', 340),
 ('Church Street', 338),
 ('Sanchez Street', 314),
 ('Divisadero Street', 306),
 ('Dolores Street', 288),
 ('24th Street', 273),
 ('Lincoln Way', 253),
 ('Carl Street', 243),
 ('Market Street', 222),
 ('Judah Street', 216),
 ('7th Avenue', 216),
 ('Mission Street', 205),
 ('Pacheco Street', 204),
 ('Frederick Street', 192),
 ('Vallejo Street', 190),
 ('6th Avenue', 182),
 ('Castro Street', 176),
 ('Polk Street', 171),
 ('23rd Street', 171),
 ('Noriega Street', 170),
 ('Stanyan Street', 169),
 ('Folsom Street', 164),
 ('Green Street', 160),
 ('25th Street', 159),
 ('Waller Street', 156),
 ('Geary Boulevard', 156),
 ('Oak Street', 155),
 ('Valencia Street', 150),
 ('5th Avenue', 150),
 ('West Portal Avenue', 144),
 ('Wawona Street', 143),
 ('26th Street', 143),
 ('17th Street', 141),
 ('Haight Street', 137),
 ('Kirkham Street', 135),
 ('Guerrero Street', 132),
 ('Pacific Avenue', 128),
 ('Balboa Street', 128),
 ('Magellan Avenue', 126),
 ('Cesar Chavez Street', 124),
 ('Van Ness Avenue', 123),
 ('Parnassus Avenue', 120),
 ('Lawton Street', 118),
 ('Ortega Street', 117),
 ('Quintara Street', 116),
 ('Montgomery Street', 113),
('Howard Street', 111),
('Castenada Avenue', 109),
('14th Street', 103),
('Mason Street', 102),
('4th Avenue', 102),
('Fair Oaks Street', 101),
('27th Street', 100),
('Moraga Street', 99),
('22nd Street', 99),
('Hugo Street', 97),
('28th Street', 97),
('Willard Street', 96),
('Stockton Street', 94),
('Chestnut Street', 94),
('19th Avenue', 94),
('Powell Street', 93),
('29th Street', 93),
('3rd Avenue', 90),
('Clipper Street', 87),
('Jersey Street', 86),
('Duncan Street', 86),
('Cole Street', 86),
('Valley Street', 84),
('30th Street', 83),
('California Street', 80),
('15th Street', 79),
('15th Avenue', 79),
('Sutter Street', 78),
('Union Street', 76),
('Jackson Street', 74),
('Geary Street', 74),
('2nd Street', 74),
('Day Street', 73),
('Woodland Avenue', 72),
('Ocean Avenue', 72),
('Warren Drive', 69),
('Ulloa Street', 69),
('Kearny Street', 68),
('Grant Avenue', 67),
('Bush Street', 67),
('3rd Street', 67),
('Columbus Avenue', 66),
('9th Street', 66),
('16th Street', 66),
('Henry Street', 65),
('Forest Side Avenue', 64),
('16th Avenue', 60),
('Post Street', 58),
('Larkin Street', 58),
('Chattanooga Street', 56),
('San Jose Avenue', 55),
('Madrone Avenue', 55),
('Fillmore Street', 55),
('Dorantes Avenue', 55),
('McAllister Street', 54),
('Broadway Street', 53),
('Broadway', 53),
('Edgewood Avenue', 52),
('Shrader Street', 51),
('Natoma Street', 51),
('Cragmont Avenue', 51),
('2nd Avenue', 51),
('Lombard Street', 50),
('Langton Street', 50),
('Greenwich Street', 48),
('Vicente Street', 47),
('Santiago Street', 47),
('Noe Street', 47),
('Franklin Street', 46),
('Harrison Street', 45),
('8th Street', 45),
('Fulton Street', 43),
('40th Avenue', 43),
('18th Street', 42),
('Mendosa Avenue', 40),
('Hayes Street', 40),
('San Marcos Avenue', 38),
('Mars Street', 38),
('Dewey Boulevard', 38),
('Taylor Street', 37),
('Rivera Street', 36),
('Filbert Street', 36),
('Ellis Street', 36),
('Brannan Street', 35),
('Arguello Boulevard', 35),
('20th Avenue', 35),
('Beaver Street', 34),
('Vicksburg Street', 33),
('Sansome Street', 33),
('Randall Street', 33),
('Spear Street', 32),
('Bryant Street', 32),
('17th Avenue', 32),
('Pine Street', 31),
('Golden Gate Avenue', 31),
('4th Street', 31),
('Sotelo Avenue', 30),
('Portola Drive', 30),
('Corbett Avenue', 30),
('Lenox Way', 29),
('Grove Street', 29),
('Elizabeth Street', 29),
('Clementina Street', 29),
('Clement Street', 29),
('7th Street', 29),
('Steiner Street', 28),
('Jones Street', 28),
('Downey Street', 28),
('18th Avenue', 28),
('Fell Street', 27),
('Eddy Street', 27),
('Rausch Street', 26),
('Santa Rita Avenue', 25),
('Alton Avenue', 25),
('36th Avenue', 25),
('27th Avenue', 25),
('20th Street', 25),
('Washington Street', 24),
("O'Farrell Street", 24),
('Marcela Avenue', 23),
('Locksley Avenue', 23),
('Lopez Avenue', 22),
('Ventura Avenue', 21),
('New Montgomery Street', 21),
('Dore Street', 21),
('39th Avenue', 21),
('34th Avenue', 21),
('32nd Avenue', 21),
('26th Avenue', 21),
('21st Street', 21),
('Sacramento Street', 20),
('Prescott Court', 20),
('John Street', 20),
('Hyde Street', 20),
('Cortland Avenue', 20),
('Aloha Avenue', 20),
('6th Street', 20),
('31st Avenue', 20),
('Broderick Street', 19),
('37th Avenue', 19),
('10th Street', 19),
('Townsend Street', 18),
('Stevenson Street', 18),
('Clayton Street', 18),
('35th Avenue', 18),
('23rd Avenue', 18),
('Lurline Street', 17),
('Gough Street', 17),
('Fort Mason', 17),
('Ashbury Street', 17),
('Sloat Boulevard', 16),
('Bayshore Boulevard', 16),
('Battery Street', 16),
('38th Avenue', 16),
('28th Avenue', 16),
('24th Avenue', 16),
('1st Street', 16),
('The Embarcadero', 15),
('Sumner Street', 15),
('South Van Ness Avenue', 15),
('Octavia Street', 15),
('Montalvo Avenue', 15),
('King Street', 15),
('30th Avenue', 15),
('21st Avenue', 15),
('Whitney Street', 14),
('Rodgers Street', 14),
('Pierce Street', 14),
('Minna Street', 14),
('Masonic Avenue', 14),
('Jefferson Street', 14),
('Bonita Street', 14),
('Berry Street', 14),
('25th Avenue', 14),
('White Street', 13),
('Wayne Place', 13),
('Rhode Island Street', 13),
('Linares Avenue', 13),
('Laguna Honda Boulevard', 13),
('Duboce Avenue', 13),
('Auburn Street', 13),
('33rd Avenue', 13),
('29th Avenue', 13),
('22nd Avenue', 13),
('Rockridge Drive', 12),
('Bernard Street', 12),
('Belvedere Street', 12),
('11th Street', 12),
('Scott Street', 11),
('Monterey Boulevard', 11),
('Hodges Alley', 11),
('Hill Point Avenue', 11),
('Webster Street', 10),
('Steuart Street', 10),
('Owens Street', 10),
('Junipero Serra Boulevard', 10),
('Diamond Street', 10),
('Brown Street', 10),
('5th Street', 10),
('Rivoli Street', 9),
('Mariposa Street', 9),
('Lyon Street', 9),
('Francisco Street', 9),
('Clay Street', 9),
('McCoppin Street', 8),
('Leavenworth Street', 8),
('Lakeshore Plaza', 8),
('Laguna Street', 8),
('Jasper Place', 8),
('Holloway Avenue', 8),
('Gateview Court', 8),
('Florida Street', 8),
('Eucalyptus Drive', 8),
('Diamond Heights Boulevard', 8),
('Devonshire Way', 8),
('Claremont Boulevard', 8),
('South Park', 7),
('Sola Avenue', 7),
('Sheldon Terrace', 7),
('North Point Street', 7),
('Main Street', 7),
('Front Street', 7),
('Fremont Street', 7),
('De Haro Street', 7),
('Central Avenue', 7),
('Buena Vista Avenue West', 7),
('Belmont Avenue', 7),
('Beale Street', 7),
('Bannam Place', 7),
('Jessie Street', 6),
('Hillway Avenue', 6),
('Embarcadero Center', 6),
('Chenery Street', 6),
('California Avenue', 6),
('Whiting Way', 5),
('Vallejo', 5),
('Turk Street', 5),
('Trenton Street', 5),
('Russ Street', 5),
('Mason', 5),
('Marina Boulevard', 5),
('Kansas Street', 5),
('Heron Street', 5),
('Hattie Street', 5),
('Gold Street', 5),
('Fresno Street', 5),
('Drumm Street', 5),
('Davis Street', 5),
('Beachmont Drive', 5),
('Yorba Street', 4),
('Yerba Buena Lane', 4),
('Woodacre Drive', 4),
('Waverly Place', 4),
('Treat Avenue', 4),
('States Street', 4),
('San Leandro Way', 4),
('Ross Alley', 4),
('Ritch Street', 4),
('Ringold Street', 4),
('Rayburn Street', 4),
('Radio Terrace', 4),
('Potrero Avenue', 4),
('Pier 39', 4),
('Missouri St', 4),
('Lomita Avenue', 4),
('Lily Street', 4),
('Liberty Street', 4),
('Lagunitas Drive', 4),
('Johnstone Drive', 4),
('Gates Street', 4),
('Division Street', 4),
('Cortes Avenue', 4),
('Cecilia Avenue', 4),
('Buchanan Street', 4),
('Bluxome Street', 4),
('Beckett Street', 4),
('Bayside Village Place', 4),
('Bay Street', 4),
('Baker Street', 4),
('Winston Drive', 3),
('Walter Street', 3),
('Uranus Terrace', 3),
('Upper Terrace', 3),
('Salmon Street', 3),
('Precita Avenue', 3),
('Osgood Place', 3),
('Oakdale Avenue', 3),
('Mission Bay Boulevard North', 3),
('Mint Plaza', 3),
('Mesa Avenue', 3),
('Loomis Street', 3),
('Jamestown Avenue', 3),
('Hillcrest Road', 3),
('Hill Street', 3),
('Fanning Way', 3),
('Connecticut Street', 3),
('Connecticut St', 3),
('Carolina Street', 3),
('Cargo Way', 3),
('Boynton Court', 3),
('Belden Place', 3),
('Beach Street', 3),
('Avenue of the Palms', 3),
('August Alley', 3),
('Alemany Boulevard', 3),
('Alabama Street', 3),
('Aerial Way', 3),
('19th Street', 3),
('Wisconsin Street', 2),
('Willie Mays Plaza', 2),
('Vermont Street', 2),
('Trinity Place', 2),
('Tompkins Avenue', 2),
('Tiffany Avenue', 2),
('Tennessee Street', 2),
('South Park Avenue', 2),
('San Bruno Avenue', 2),
('Russian Hill Place', 2),
('Romolo Place', 2),
('Redfield Alley', 2),
('Presidio Avenue', 2),
('Pollard Place', 2),
('Otis Street', 2),
('Omar Way', 2),
('Nottingham Place', 2),
('New Montgomery St', 2),
('Mission St', 2),
('Minnesota Street', 2),
('Merchant Street', 2),
('Merced Avenue', 2),
('Marin Street', 2),
('Long Bridge Street', 2),
('Lombard street', 2),
('Kissling Street', 2),
('Juri Street', 2),
('Jennings Street', 2),
('Ivy Street', 2),
('Illinois Street', 2),
('Hays Street', 2),
('Hawthorne Street', 2),
('Harriet Street', 2),
('Grenard Terrace', 2),
 ('Green St', 2),
 ('Garden Way', 2),
 ('Fowler Avenue', 2),
 ('Font Boulevard', 2),
 ('Florence Street', 2),
 ('Fishermans Wharf', 2),
 ('Ferry Building', 2),
 ('Farnsworth Lane', 2),
 ('Escolta Way', 2),
 ('Deming Street', 2),
 ('Delmar Street', 2),
 ('Decatur Street', 2),
 ('Cyril Magnin Street', 2),
 ('Columbia Square Street', 2),
 ('Claude Lane', 2),
 ('Cervantes Boulevard', 2),
 ('Cascade Walk', 2),
 ('Capp Street', 2),
 ('California', 2),
 ('Belles Street', 2),
 ('Bartlett Street', 2),
 ('Anza Street', 2),
 ('Ames Street', 2),
 ('Alameda Street', 2),
 ('13th Street', 2),
 ('townsend street', 1),
 ('Yosemite Avenue', 1),
 ('York Street', 1),
 ('York St', 1),
 ('Yacht Road', 1),
 ('Williams St', 1),
 ('William Saroyan Place', 1),
 ('Wilder Street', 1),
 ('Wentworth Place', 1),
 ('Wedemeyer', 1),
 ('Webster Street #502', 1),
 ('Wayland Street', 1),
 ('Walter U Lum Place', 1),
 ('Virginia Avenue', 1),
 ('Vine Terrace', 1),
 ('Via Ferlinghetti', 1),
 ('Van Ness Ave', 1),
 ('Van Ness', 1),
 ('Van Dyke Avenue', 1),
 ('Vallejo street', 1),
 ('Vallejo Street Stairway', 1),
 ('Valencia St', 1),
 ('United Nations Plaza', 1),
 ('Turk Boulevard', 1),
 ('Tillman Place', 1),
 ('Third Street', 1),
 ('Terry Francois Street', 1),
 ('Terry A Francois Boulevard', 1),
 ('Terry A Francois Blvd', 1),
 ('Tehama Street', 1),
 ('Stark Street', 1),
 ('Staples Avenue', 1),
 ('South Gate Road', 1),
 ('Sixth Street', 1),
 ('Shotwell Street', 1),
 ('Sheridan Avenue', 1),
 ('Sharon Street', 1),
 ('Shafter Avenue', 1),
 ('Sergeant John V Young Lane', 1),
 ('Second Street', 1),
 ('Saramento Street', 1),
 ('Sansome Street Suite 730', 1),
 ('Sansome Street Ste 730', 1),
 ('Sansome St #3500', 1),
 ('San Francisco Bicycle Route 2', 1),
 ('San Fernando Way', 1),
 ('Saint Francis Place', 1),
 ('SF 80 PM 4.5', 1),
 ('Russia Avenue', 1),
 ('Roosevelt Way', 1),
 ('Post', 1),
 ('Pollard', 1),
 ('Plaza Street', 1),
 ('Pier 50 B', 1),
 ('Pier 50 A', 1),
 ('Phelps Street', 1),
 ('Phelan Avenue', 1),
 ('Phelan Ave', 1),
 ('Pershing Drive', 1),
 ('Pennsylvania Avenue', 1),
 ('Pennsylvania Ave', 1),
 ('Paul Avenue', 1),
 ('Panorama Drive', 1),
 ('Pacific Avenue Mall', 1),
 ('Oriole Way', 1),
 ('One Letterman Drive', 1),
 ('Octavia Boulevard', 1),
 ('Norwich Street', 1),
 ('Noe St', 1),
 ('Niagara Avenue', 1),
 ('New Montgomery', 1),
 ('Nellie Street', 1),
 ('Napoleon Street', 1),
 ('Music Concourse Drive', 1),
 ('Museum Way', 1),
 ('Murray Street and Justin Drive', 1),
 ('Multi Use Building', 1),
 ('Moultrie Street', 1),
 ('Morena Ave', 1),
 ('Mississippi Street', 1),
 ('Mission Rock Street', 1),
 ('Mission Rock', 1),
 ('Mint Street', 1),
 ('Milton Street', 1),
 ('Middlefield Drive', 1),
 ('Mersey Street', 1),
 ('Mendell St', 1),
 ('Market Street Suite 3658', 1),
 ('Market St.', 1),
 ('Market St', 1),
 ('Marine Drive', 1),
 ('Marina Boulevard Building D', 1),
 ('Marina Blvd', 1),
 ('Maiden Lane', 1),
 ('MIssion Street', 1),
 ('Lyell Street', 1),
 ('Lusk Street', 1),
 ('Linden Street', 1),
 ('Lincoln Boulevard', 1),
 ('Lilac Alley', 1),
 ('Lexington Street', 1),
 ('Letterman Drive', 1),
 ('Leidesdorff Street', 1),
 ('Leavenworth St', 1),
 ('Lansing Street', 1),
 ('Lake Street', 1),
 ('Koret Way', 1),
 ('King', 1),
 ('Kearny st', 1),
 ('Kearny St', 1),
 ('Jose Sarria Court', 1),
 ('John F Kennedy Drive', 1),
 ('Jerrold Avenue', 1),
 ('Jerrold Ave', 1),
 ('Isadora Duncan Lane', 1),
 ('Industrial Street', 1),
 ('Hyde', 1),
 ('Hudson Street', 1),
 ('Hooper Street', 1),
 ('Holly Park Circle', 1),
 ('Hoff Street', 1),
 ('Hermann Street', 1),
 ('Henry Adams Street', 1),
 ('Hayes St', 1),
 ('Harding Road', 1),
 ('Hardie Place', 1),
 ('Hampshire Street', 1),
 ('Halleck St.', 1),
 ('Hallam Street', 1),
 ('Hagiwara Tea Garden Drive', 1),
 ('Grant Ave', 1),
 ('Gough', 1),
 ('Gilbert Street', 1),
 ('Geneva Avenue', 1),
 ('Funston Road', 1),
 ('Freelon Street', 1),
 ('Francis Street', 1),
 ('Fort Mason Center', 1),
 ('Flood Avenue', 1),
 ('Ferry Plaza', 1),
 ('Federal Street', 1),
 ('Eureka Street', 1),
 ('Erie Street', 1),
 ('Embarcadero', 1),
 ('Ellsworth Street', 1),
 ('Elim Street', 1),
 ('Dr Carlton P Goodlett Place', 1),
 ('Delancey Street', 1),
 ('Delancey St', 1),
 ('De Montfort Avenue', 1),
 ('De Boom Street', 1),
 ('Darien Way', 1),
 ('Danvers Street', 1),
 ('Daniel Burnham Court', 1),
 ('Cosmo Place', 1),
 ('Compton Road', 1),
 ('Commercial Street', 1),
 ('Clarence Place', 1),
 ('Clara Street', 1),
 ('Channel Street', 1),
 ('Cesar Chavez St St', 1),
 ('Castro St', 1),
 ('Campton Place', 1),
 ('California St.', 1),
 ('Butte Place', 1),
 ('Buena Vista Avenue East', 1),
 ('Buckingham Way', 1),
 ('Brosnan Street', 1),
 ('Broadway Street; Mason Street', 1),
 ('Brazil Avenue', 1),
 ('Brannan St #151', 1),
 ('Bosworth Street', 1),
 ('Beulah Street', 1),
 ('Belcher Street', 1),
 ('Behr Avenue', 1),
 ('Bay and Powell', 1),
 ('Bartol Street', 1),
 ('Bartlett Street #203', 1),
 ('Bacon Street', 1),
 ('Army Road', 1),
 ('Arlington Street', 1),
 ('Aptos Avenue', 1),
 ('Amber Drive', 1),
 ('Amador Street', 1),
 ('9th St', 1),
 ('434 Main Street', 1),
 ('42nd Avenue', 1),
 ('41st Avenue', 1),
 ('3301 Lyon Street', 1),
 ('303 Second Street', 1),
 ('30 Rickard Street', 1),
 ('2640 mason st', 1),
 ('24th St', 1),
 ('19th St', 1),
 ('19th & Linda San Francisco', 1),
 ('16th St #404', 1),
 ('15th', 1),
 ('14th St, San Francisco ', 1),
 ('14th St', 1),
 ('12th Street', 1)]
```

In [6]:

```
from collections import defaultdict
street_type_re = re.compile(r'\b\S+\.?$', re.IGNORECASE)
```

```
expected = ["Street", "Avenue", "Boulevard", "Drive", "Court", "Place", "Squar
e", "Lane", "Road", 
            "Trail", "Parkway", "Commons",'Terrace',"Plaza","Way","Center","Br
oadway","Post"]
# UPDATE THIS VARIABLE
mapping = { "St": "Street",
            "St.": "Street",
           'st':'Street',
            "Ave": "Avenue",
           'ave':"Avenue",
            "Rd.": "Road",
            "W.": "West",
            "N.": "North",
            "S.": "South",
            "E": "East",
           "Ln":'"Lane',
           'way':"Way"

          }


def audit_street_type(street_types, street_name):
    m = street_type_re.search(street_name)
    if m:
        street_type = m.group()
        if street_type not in expected:
            street_types[street_type].add(street_name)


def is_street_name(elem):
    return (elem.attrib['k'] == "addr:street")


def audit(osmfile):
    osm_file = open(osmfile, "r")
    street_types = defaultdict(set)
    for event, elem in ET.iterparse(osm_file, events=("start",)):
        if elem.tag == "node" or elem.tag == "way":
            for tag in elem.iter("tag"):
                if is_street_name(tag):
                    audit_street_type(street_types, tag.attrib['v'])

    return street_types

def update_name(name, mapping):
    after = []
    # Split name string to test each part of the name;
    # Replacements may come anywhere in the name.
    for part in name.split(" "):
        # Check each part of the name against the keys in the correction dict
        if part in mapping.keys():
            # If exists in dict, overwrite that part of the name with the dict
 value for it.
            part = mapping[part]
        # Assemble each corrected piece of the name back together.
        after.append(part)
    # Return all pieces of the name as a string joined by a space.
    return " ".join(after)

    return name
def map_test():
    st_types = audit(SF)
    pprint.pprint(dict(st_types))

    for st_type, ways in st_types.items():
        for name in ways:
            better_name = update_name(name, mapping)
            print name, "=>", better_name


if __name__ == '__main__':
    map_test()
```

```
{'151': set(['Brannan St #151']),
 '15th': set(['15th']),
 '2': set(['San Francisco Bicycle Route 2']),
 '203': set(['Bartlett Street #203']),
 '3500': set(['Sansome St #3500']),
 '3658': set(['Market Street Suite 3658']),
 '39': set(['Pier 39']),
 '4.5': set(['SF 80 PM 4.5']),
 '404': set(['16th St #404']),
 '502': set(['Webster Street #502']),
 '730': set(['Sansome Street Ste 730', 'Sansome Street Suite 730']),
 'A': set(['Pier 50 A']),
 'Alley': set(['August Alley',
               'Hodges Alley',
               'Lilac Alley',
               'Redfield Alley',
               'Ross Alley']),
 'Ave': set(['Grant Ave',
             'Jerrold Ave',
             'Morena Ave',
             'Pennsylvania Ave',
             'Phelan Ave',
             'Van Ness Ave']),
 'B': set(['Pier 50 B']),
 'Blvd': set(['Marina Blvd', 'Terry A Francois Blvd']),
 'Building': set(['Ferry Building', 'Multi Use Building']),
 'California': set(['California']),
 'Circle': set(['Holly Park Circle']),
 'D': set(['Marina Boulevard Building D']),
 'East': set(['Buena Vista Avenue East']),
 'Embarcadero': set(['Embarcadero', 'The Embarcadero']),
 'Ferlinghetti': set(['Via Ferlinghetti']),
 'Francisco': set(['19th & Linda San Francisco']),
 'Gough': set(['Gough']),
 'Hyde': set(['Hyde']),
 'King': set(['King']),
 'Mall': set(['Pacific Avenue Mall']),
 'Mason': set(['Fort Mason', 'Mason']),
 'Montgomery': set(['New Montgomery']),
 'Ness': set(['Van Ness']),
 'North': set(['Mission Bay Boulevard North']),
 'Palms': set(['Avenue of the Palms']),
 'Park': set(['South Park']),
 'Pollard': set(['Pollard']),
 'Powell': set(['Bay and Powell']),
 'Rock': set(['Mission Rock']),
 'St': set(['14th St',
            '19th St',
            '24th St',
            '9th St',
            'Castro St',
            'Cesar Chavez St St',
            'Connecticut St',
            'Delancey St',
            'Green St',
            'Hayes St',
            'Kearny St',
            'Leavenworth St',
            'Market St',
            'Mendell St',
            'Mission St',
            'Missouri St',
            'New Montgomery St',
            'Noe St',
            'Valencia St',
            'Williams St',
            'York St']),
 'St.': set(['California St.', 'Halleck St.', 'Market St.']),
 'Stairway': set(['Vallejo Street Stairway']),
 'Vallejo': set(['Vallejo']),
 'Walk': set(['Cascade Walk']),
 'Wedemeyer': set(['Wedemeyer']),
 'West': set(['Buena Vista Avenue West']),
 'Wharf': set(['Fishermans Wharf']),
 'st': set(['2640 mason st', 'Kearny st']),
 'street': set(['Lombard street', 'Vallejo street', 'townsend street'])}
Brannan St #151 => Brannan Street #151
Multi Use Building => Multi Use Building
Ferry Building => Ferry Building
King => King
Buena Vista Avenue West => Buena Vista Avenue West
California St. => California Street
Halleck St. => Halleck Street
Market St. => Market Street
Vallejo => Vallejo
Mason => Mason
Fort Mason => Fort Mason
townsend street => townsend street
Vallejo street => Vallejo street
Lombard street => Lombard street
Market Street Suite 3658 => Market Street Suite 3658
Holly Park Circle => Holly Park Circle
2640 mason st => 2640 mason Street
Kearny st => Kearny Street
Bay and Powell => Bay and Powell
SF 80 PM 4.5 => SF 80 PM 4.5
Buena Vista Avenue East => Buena Vista Avenue East
19th & Linda San Francisco => 19th & Linda San Francisco
Mission Bay Boulevard North => Mission Bay Boulevard North
Avenue of the Palms => Avenue of the Palms
South Park => South Park
Van Ness => Van Ness
Embarcadero => Embarcadero
The Embarcadero => The Embarcadero
Sansome Street Ste 730 => Sansome Street Ste 730
Sansome Street Suite 730 => Sansome Street Suite 730
16th St #404 => 16th Street #404
California => California
Wedemeyer => Wedemeyer
Via Ferlinghetti => Via Ferlinghetti
Webster Street #502 => Webster Street #502
15th => 15th
Pier 50 A => Pier 50 A
Vallejo Street Stairway => Vallejo Street Stairway
```

```
Bartlett Street #203 => Bartlett Street #203
Pier 50 B => Pier 50 B
Marina Boulevard Building D => Marina Boulevard Building D
Williams St => Williams Street
Mendell St => Mendell Street
Green St => Green Street
Cesar Chavez St St => Cesar Chavez Street Street
24th St => 24th Street
Mission St => Mission Street
Kearny St => Kearny Street
Valencia St => Valencia Street
Noe St => Noe Street
Hayes St => Hayes Street
19th St => 19th Street
York St => York Street
Connecticut St => Connecticut Street
Missouri St => Missouri Street
Castro St => Castro Street
14th St => 14th Street
9th St => 9th Street
New Montgomery St => New Montgomery Street
Delancey St => Delancey Street
Market St => Market Street
Leavenworth St => Leavenworth Street
Pollard => Pollard
Pacific Avenue Mall => Pacific Avenue Mall
Mission Rock => Mission Rock
New Montgomery => New Montgomery
San Francisco Bicycle Route 2 => San Francisco Bicycle Route 2
Gough => Gough
Pier 39 => Pier 39
Fishermans Wharf => Fishermans Wharf
Sansome St #3500 => Sansome Street #3500
Ross Alley => Ross Alley
Lilac Alley => Lilac Alley
August Alley => August Alley
Redfield Alley => Redfield Alley
Hodges Alley => Hodges Alley
Cascade Walk => Cascade Walk
Hyde => Hyde
Terry A Francois Blvd => Terry A Francois Blvd
Marina Blvd => Marina Blvd
Grant Ave => Grant Avenue
Morena Ave => Morena Avenue
Phelan Ave => Phelan Avenue
Van Ness Ave => Van Ness Avenue
Jerrold Ave => Jerrold Avenue
Pennsylvania Ave => Pennsylvania Avenue
```

In [7]:

```
def count_keys(filename):
    keys = {}
    for event, elem in ET.iterparse(filename, events=('start', 'end')):
        if event == 'end':
            key = elem.attrib.get('k')
            if key:
                if key not in keys:
                    keys[key] = 1
                else:
                    keys[key] += 1
    return keys

keys = count_keys(SF)
sorted_by_occurrence = [(k, v) for (v, k) in sorted([(value, key) for (key, va
lue) in keys.items()], reverse=True)]

print 'Keys and occurrence in San-Francisco_california.osm:\n'
pprint.pprint(sorted_by_occurrence)
```

```
Keys and occurrence in San-Francisco_california.osm:

[('building', 134961),
 ('height', 41037),
 ('addr:street', 20925),
 ('addr:housenumber', 20072),
 ('name', 20042),
 ('highway', 19989),
 ('addr:city', 16776),
 ('addr:postcode', 13362),
 ('addr:state', 12109),
 ('addr:country', 8759),
 ('tiger:county', 8739),
 ('tiger:cfcc', 8520),
 ('tiger:name_type', 8002),
 ('tiger:name_base', 7877),
 ('amenity', 6889),
 ('tiger:zip_left', 6321),
 ('tiger:zip_right', 5534),
 ('source', 5530),
 ('oneway', 4457),
 ('natural', 3590),
 ('lanes', 3491),
 ('taxon', 2982),
 ('species:en', 2837),
 ('leisure', 2605),
 ('type', 2503),
 ('shop', 2460),
 ('maxspeed', 2414),
 ('tiger:reviewed', 2093),
 ('lcn_ref', 2077),
 ('operator', 2076),
 ('railway', 1798),
 ('leaf_cycle', 1759),
 ('website', 1666),
 ('leaf_type', 1468),
 ('sidewalk', 1438),
 ('ref', 1411),
 ('tiger:tlid', 1366),
 ('tiger:source', 1364),
 ('tiger:separated', 1343),
 ('foot', 1284),
 ('surface', 1261),
 ('cuisine', 1261),
 ('phone', 1210),
 ('service', 1166),
 ('building:levels', 1135),
 ('bicycle', 1080),
 ('trolley_wire', 1048),
 ('restriction', 1030),
 ('crossing', 989),
 ('landuse', 983),
 ('cycleway:right', 946),
 ('ele', 883),
 ('cycleway', 839),
 ('created_by', 827),
 ('gnis:feature_id', 781),
 ('shelter', 668),
 ('OBJNAME', 668),
 ('route_ref', 648),
 ('tourism', 634),
 ('landcover', 617),
 ('lanes:backward', 607),
 ('turn:lanes', 606),
 ('footway', 603),
 ('lanes:forward', 601),
 ('wheelchair', 597),
 ('gnis:created', 582),
 ('gnis:county_id', 576),
 ('gnis:state_id', 575),
 ('ticker', 567),
 ('man_made', 556),
 ('access', 555),
 ('layer', 551),
 ('barrier', 544),
 ('opening_hours', 531),
 ('tiger:name_base_1', 517),
 ('area', 497),
 ('note', 438),
 ('BLDGID', 419),
 ('AREA_M2', 395),
 ('sport', 391),
 ('capacity', 376),
 ('bridge', 357),
 ('building:material', 351),
 ('route', 316),
 ('religion', 294),
 ('office', 294),
 ('alt_name', 282),
 ('hgv', 280),
 ('seamark:type', 263),
 ('turn_restrictions', 262),
 ('gauge', 260),
 ('bulb', 259),
 ('tiger:name_type_1', 249),
 ('fee', 238),
 ('electrified', 235),
 ('tiger:upload_uuid', 230),
 ('network', 226),
 ('tiger:zip_left_1', 220),
 ('parking', 216),
 ('NHS', 211),
 ('to', 203),
 ('from', 202),
 ('turn:lanes:forward', 200),
 ('denomination', 198),
 ('entrance', 190),
 ('turn:lanes:backward', 183),
 ('quantity', 178),
 ('source:hgv:state_network', 174),
 ('hgv:state_network', 174),
 ('traffic_signals:sound', 171),
 ('level', 164),
 ('gnis:county_name', 161),
 ('wikipedia', 156),
 ('smoking', 156),
 ('restriction:conditional', 155),
 ('old_ref', 154),
 ('lit', 149),
 ('source:pkey', 145),
 ('emergency', 138),
 ('addr:housename', 138),
 ('description', 129),
 ('cycleway:left', 129),
 ('backrest', 129),
 ('name:en', 123),
 ('maxweight', 122),
 ('gnis:import_uuid', 122),
 ('Trunk_Diam', 120),
 ('voltage', 119),
 ('frequency', 119),
 ('common', 117),
 ('Subclass', 117),
 ('atm', 115),
 ('name:zh', 114),
 ('colour', 114),
 ('old_name', 108),
 ('gnis:reviewed', 106),
 ('internet_access', 105),
 ('name_1', 102),
 ('tunnel', 101),
 ('nextbus:agency', 100),
 ('takeaway', 99),
 ('outdoor_seating', 99),
 ('covered', 99),
 ('sfgov:OBJNAME', 96),
 ('route_master', 95),
 ('outside', 95),
 ('nextbus:route', 93),
 ('motor_vehicle', 93),
 ('horse', 90),
 ('addr:interpolation', 89),
 ('payment:bitcoin', 88),
 ('FIXME', 88),
 ('public_transport', 87),
 ('historic', 87),
 ('hour_on', 83),
 ('email', 83),
 ('day_on', 82),
 ('survey:date', 81),
 ('usage', 77),
 ('building:part', 75),
 ('hour_off', 73),
 ('railway:traffic_mode', 71),
 ('placement:backward', 69),
 ('day_off', 69),
 ('bench', 69),
 ('wikidata', 64),
 ('cycle_network', 64),
 ('craft', 63),
 ('width', 62),
 ('incline', 61),
 ('source:hgv:national_network', 59),
 ('hgv:national_network', 59),
 ('place', 58),
 ('url', 57),
 ('traffic_calming', 56),
 ('brand', 56),
 ('tiger:zip_right_1', 54),
 ('cable_car', 54),
 ('fax', 52),
 ('construction', 51),
 ('payment:litecoin', 49),
 ('noaa:taghash', 49),
 ('noaa:lnam', 49),
 ('noaa:geohash', 49),
 ('drive_through', 49),
 ('toilets', 48),
 ('material', 48),
 ('gnis:edited', 46),
 ('addr.source:housenumber', 45),
 ('seamark:status', 44),
 ('roof:shape', 44),
 ('owner', 44),
 ('except', 44),
 ('wifi', 43),
 ('bicycle_parking', 43),
 ('waterway', 42),
 ('outside_atm', 42),
 ('noexit', 42),
 ('tactile_paving', 41),
 ('start_date', 41),
 ('location', 40),
 ('junction', 40),
 ('is_in', 38),
 ('information', 38),
 ('contact:phone', 36),
 ('subway', 35),
 ('power', 35),
 ('gnis:id', 35),
 ('exit_to', 35),
 ('sfgov.org:OFFICE_TYP', 34),
 ('sfgov.org:OBJECTID', 34),
 ('fire_hydrant:type', 34),
 ('import_uuid', 33),
 ('gnis:ST_num', 33),
 ('gnis:ST_alpha', 33),
 ('gnis:County_num', 33),
 ('gnis:County', 33),
 ('gnis:Class', 33),
 ('designation', 33),
 ('toilets:disposal', 32),
 ('tiger:name_direction_prefix', 31),
 ('roof:colour', 31),
 ('gosm:sig:8CBDE645', 31),
 ('toll', 30),
 ('name:ko', 30),
 ('direction', 30),
 ('delivery', 30),
 ('microbrewery', 29),
 ('dispensing', 29),
 ('traffic_signals', 28),
 ('parking:condition:right:2:time_interval', 28),
 ('parking:condition:right:2:reason', 28),
 ('parking:condition:right:2', 28),
 ('parking:condition:left:2:time_interval', 28),
 ('parking:condition:left:2:reason', 28),
 ('parking:condition:left:2', 28),
 ('depository', 28),
 ('bus', 28),
 ('psv', 27),
 ('operator:type', 27),
 ('maxlength', 27),
 ('ref:blklot', 26),
 ('Id', 26),
 ('tiger:zip_left_2', 25),
 ('turn:lanes:both_ways', 24),
 ('supervised', 24),
 ('lanes:both_ways', 24),
 ('seamark:name', 23),
 ('note:address', 23),
 ('min_height', 23),
 ('local_ref', 23),
 ('history', 23),
 ('grade', 23),
 ('addr:unit', 23),
 ('seamark:beacon_special_purpose:shape', 21),
 ('healthcare', 21),
 ('fixme', 21),
 ('destination', 21),
 ('artwork_type', 21),
 ('station_name', 20),
 ('collection_times', 20),
 ('roof:height', 19),
 ('drinking_water', 19),
 ('cutting', 19),
 ('control', 19),
 ('address', 19),
 ('wetap:status', 18),
 ('tram', 18),
 ('seamark:light:colour', 18),
 ('seamark:light:character', 18),
 ('seamark:beacon_special_purpose:colour', 18),
 ('passenger', 18),
 ('maxheight', 18),
 ('grades', 18),
 ('golf', 18),
 ('dry_clean', 18),
 ('parking:condition:both', 17),
 ('indoor', 17),
 ('bottle', 17),
 ('baby_hatch', 17),
 ('verified', 16),
 ('vending', 16),
 ('source:name', 16),
 ('seamark:light:height', 16),
 ('seamark:daymark:shape', 16),
 ('seamark:daymark:colour_pattern', 16),
 ('seamark:daymark:colour', 16),
 ('oneway:bicycle', 16),
 ('via', 15),
 ('toilets:wheelchair', 15),
 ('short_name', 15),
 ('public', 15),
 ('architect', 15),
 ('social_facility', 14),
 ('seamark:light:exhibition', 14),
 ('playground', 14),
 ('parking:lane:both', 14),
 ('parking:condition:both:time_interval', 14),
 ('parking:condition:both:residents', 14),
 ('parking:condition:both:maxstay', 14),
 ('name_2', 14),
 ('contact:website', 14),
 ('alterations', 14),
 ('seamark:light:sequence', 13),
 ('seamark:light:period', 13),
 ('public_transport:version', 13),
 ('motorcar', 13),
 ('island', 13),
 ('destination:street', 13),
 ('courts', 13),
 ('coin_op', 13),
 ('social_facility:for', 12),
 ('park_ride', 12),
 ('note:lanes', 12),
 ('maxspeed:trailer', 12),
 ('maxspeed:hgv', 12),
 ('disused', 12),
 ('water', 11),
 ('symbol', 11),
 ('street', 11),
 ('service:bicycle:repair', 11),
 ('segregated', 11),
 ('seamark:fog_signal:category', 11),
 ('outside_atm_operator', 11),
 ('is_in:state', 11),
 ('internet_access:fee', 11),
 ('inner', 11),
 ('gnis:feature_type', 11),
 ('finished_laundry', 11),
 ('crossing_ref', 11),
 ('building:levels:underground', 11),
 ('building:colour', 11),
 ('bearing', 11),
 ('vestibule_atm', 10),
 ('service:bicycle:retail', 10),
 ('seamark:daymark:construction', 10),
 ('seamark:beacon_lateral:shape', 10),
 ('seamark:beacon_lateral:colour', 10),
 ('seamark:beacon_lateral:category', 10),
 ('outside_atm_capacity', 10),
 ('noname', 10),
 ('healthcare:speciality', 10),
 ('capacity:disabled', 10),
 ('artist_name', 10),
 ('addr:full', 10),
 ('terminal', 9),
 ('stop_id', 9),
 ('source_ref', 9),
 ('ship:type', 9),
 ('name:de', 9),
 ('lanes:bus', 9),
 ('inside_atm', 9),
 ('dog', 9),
 ('wpt_description', 8),
 ('train', 8),
 ('tracktype', 8),
 ('stop', 8),
 ('service:bicycle:rental', 8),
 ('second_hand', 8),
 ('seamark:fog_signal:period', 8),
 ('seamark:fog_signal:group', 8),
 ('seamark:beacon_safe_water:shape', 8),
 ('samtrans_route_ref', 8),
 ('overtaking', 8),
 ('organic', 8),
 ('old_tourism', 8),
 ('name:es', 8),
 ('hoops', 8),
 ('handrail', 8),
 ('food', 8),
 ('field_marking', 8),
 ('disused:amenity', 8),
 ('contact:email', 8),
 ('wheelchair:description', 7),
 ('wetap:statusnote', 7),
 ('vestibule_atm_capacity', 7),
 ('traction', 7),
 ('tiger:zip_left_3', 7),
 ('taqueria', 7),
 ('seamark:light:range', 7),
 ('seamark:fog_signal:generation', 7),
 ('lcn', 7),
 ('ggt_route_ref', 7),
 ('fire_hydrant:position', 7),
 ('ethnicity', 7),
 ('country', 7),
 ('building:height', 7),
 ('area:highway', 7),
 ('NRHP', 7),
 ('wetap:photo', 6),
 ('vestibule_atm_operator', 6),
 ('unisex', 6),
 ('sloped_curb', 6),
 ('seamark:information', 6),
 ('seamark:fog_signal:sequence', 6),
 ('seamark:beacon_special_purpose:category', 6),
 ('ramp', 6),
 ('placement:forward', 6),
 ('placement', 6),
 ('payment:none', 6),
 ('parking:lane:left', 6),
 ('origin', 6),
 ('nextbus:stopid', 6),
 ('name:ru', 6),
 ('name:ca', 6),
 ('motorcycle', 6),
 ('military', 6),
 ('male', 6),
 ('kerb', 6),
 ('historic:name', 6),
 ('emergency:note', 6),
 ('bus:lanes:backward', 6),
 ('bridge:name', 6),
 ('attribution', 6),
 ('training', 5),
 ('trail_visibility', 5),
 ('tiger:zip_right_2', 5),
 ('tiger:name_direction_suffix', 5),
 ('taxi', 5),
 ('step_count', 5),
 ('source:addr:housenumber', 5),
 ('service:bicycle:pump', 5),
 ('self_service', 5),
 ('seamark:buoy_special_purpose:shape', 5),
 ('seamark:buoy_special_purpose:colour', 5),
 ('seamark:buoy_lateral:shape', 5),
 ('seamark:buoy_lateral:colour', 5),
 ('seamark:buoy_lateral:category', 5),
 ('parking:lane:right', 5),
 ('official_name', 5),
 ('nhd:reach_code', 5),
 ('name:fr', 5),
 ('localwiki', 5),
 ('laundry_service', 5),
 ('is_in:country_code', 5),
 ('female', 5),
 ('books', 5),
 ('attraction', 5),
 ('area:railway', 5),
 ('steps', 4),
 ('site', 4),
 ('seamark:buoy_lateral:colour_pattern', 4),
 ('sanitary_dump_station:pump-out', 4),
 ('roof:slope:direction', 4),
 ('ref:right', 4),
 ('ref:left', 4),
 ('proposed', 4),
 ('postal_code', 4),
 ('note:name', 4),
 ('note:lcn', 4),
 ('name:pt', 4),
 ('name:fa', 4),
 ('motor_vehicle:conditional', 4),
 ('is_in:country', 4),
 ('image', 4),
 ('hazard:bicycle', 4),
 ('exit_to:left', 4),
 ('drive_in', 4),
 ('disused:highway', 4),
 ('demolished', 4),
 ('change:lanes:forward', 4),
 ('change:backward', 4),
 ('building:min_level', 4),
 ('boundary', 4),
 ('artist', 4),
 ('admin_level', 4),
 ('access:backward', 4),
 ('vehicle', 3),
 ('traffic_sign', 3),
 ('telecom', 3),
 ('service:press', 3),
 ('ref:BART', 3),
 ('ramp:wheelchair', 3),
 ('platforms', 3),
 ('payment:dogecoin', 3),
 ('parking:condition:right', 3),
 ('outside_atm_op', 3),
 ('operator:wikipedia', 3),
 ('old_amenity', 3),
 ('num_row', 3),
 ('nudism', 3),
 ('name_alt', 3),
 ('name:ja', 3),
 ('name:he', 3),
 ('maxspeed:advisory', 3),
 ('map_size', 3),
 ('is_in:state_code', 3),
 ('inside', 3),
 ('health_facility:type', 3),
 ('has_vestibule_atm', 3),
 ('generator:source', 3),
 ('fuel', 3),
 ('former_name', 3),
 ('fence_type', 3),
 ('disused:shop', 3),
 ('diet:vegetarian', 3),
 ('destination:ref', 3),
 ('dance:teaching', 3),
 ('currency:USD', 3),
 ('cost:coffee', 3),
 ('contact:fax', 3),
 ('change:lanes:backward', 3),
 ('change:forward', 3),
 ('car_wash', 3),
 ('caltrans:type', 3),
 ('caltrans:dynsegpm', 3),
 ('caltrans:district', 3),
 ('brand:wikidata', 3),
 ('boxes', 3),
 ('board_type', 3),
 ('animal_shelter', 3),
 ('alt_name_1', 3),
 ('aeroway', 3),
 ('Street Ont', 3),
 ('Street Fro', 3),
 ('Sign Legen', 3),
 ('Direction', 3),
 ('unsigned_ref', 2),
 ('tower:type', 2),
 ('theatre:genre', 2),
 ('surveillance', 2),
 ('station_id', 2),
 ('station', 2),
 ('stateofrepair', 2),
 ('stars', 2),
 ('source:geometry', 2),
 ('smoothness', 2),
 ('seamark:beacon_special_purpose:reflectivity', 2),
 ('seamark:beacon_lateral:reflectivity', 2),
 ('roof:material', 2),
 ('reviewed', 2),
 ('repair', 2),
 ('rental', 2),
 ('ref:nrhp', 2),
 ('psv:lanes:forward', 2),
 ('psv:lanes:backward', 2),
 ('population', 2),
 ('payment:credit_cards', 2),
 ('parking:condition:right:time_interval', 2),
 ('parking:condition:right:residents', 2),
 ('parking:condition:right:maxstay', 2),
 ('parking:condition:left:time_interval', 2),
 ('parking:condition:left:residents', 2),
 ('parking:condition:left:maxstay', 2),
 ('parking:condition:left', 2),
 ('park', 2),
 ('old_religion', 2),
 ('old_denomination', 2),
 ('notes', 2),
 ('name:ta', 2),
 ('name:kn', 2),
 ('name:hi', 2),
 ('mown', 2),
 ('map', 2),
 ('light_rail', 2),
 ('levels', 2),
 ('leisure_1', 2),
 ('landuse_1', 2),
 ('inside_atm_op', 2),
 ('hov', 2),
 ('has_outside_atm', 2),
 ('handrail:right', 2),
 ('handrail:left', 2),
 ('half_court', 2),
 ('ggt_drop_off_only', 2),
 ('furniture', 2),
 ('exit_to:right', 2),
 ('end_date', 2),
 ('drop_off_only', 2),
 ('drive_through_atm', 2),
 ('diplomatic', 2),
 ('diet:vegan', 2),
 ('diet:gluten_free', 2),
 ('cooperative', 2),
 ('contact:facebook', 2),
 ('community:gender', 2),
 ('community', 2),
 ('comment', 2),
 ('clothes', 2),
 ('bus:lanes:forward', 2),
 ('bridge:structure', 2),
 ('bridge:old_name', 2),
 ('bridge:movable', 2),
 ('bridge:alt_name', 2),
 ('basketball', 2),
 ('atm_inside', 2),
 ('animal_shelter:adoption', 2),
 ('addr:street:source', 2),
 ('addr:place', 2),
 ('addr:housenumber:source', 2),
 ('addr:floor', 2),
 ('abandoned:shop', 2),
 ('_OBJNAME_', 2),
 ('Stevenson Place', 2),
 ('zoo', 1),
 ('z_order', 1),
 ('yelp', 1),
 ('wood', 1),
 ('wine', 1),
 ('wholesale', 1),
 ('wetland', 1),
 ('wetap:quality', 1),
 ('wetap:bottle', 1),
 ('visibility', 1),
 ('vestibule_depository', 1),
 ('vestibule_atm_op', 1),
 ('vestibule_atm_fee', 1),
 ('vendor_model', 1),
 ('valet', 1),
 ('unmarked', 1),
 ('undefined', 1),
 ('turn:forward', 1),
 ('trade', 1),
 ('toilets:position', 1),
 ('tiger:zip_right_3', 1),
 ('tiger:zip_left_4', 1),
 ('tiger:name_base_2', 1),
 ('tiger:STATEFP', 1),
 ('tiger:PLCIDFP', 1),
 ('tiger:PLACENS', 1),
 ('tiger:PLACEFP', 1),
 ('tiger:PCINECTA', 1),
 ('tiger:PCICBSA', 1),
 ('tiger:NAMELSAD', 1),
 ('tiger:NAME', 1),
 ('tiger:MTFCC', 1),
 ('tiger:LSAD', 1),
 ('tiger:FUNCSTAT', 1),
 ('tiger:CPI', 1),
 ('tiger:CLASSFP', 1),
 ('theatre', 1),
 ('terrace', 1),
 ('surveillance:type', 1),
 ('support', 1),
 ('studio', 1),
 ('structure', 1),
 ('store_number', 1),
 ('sports', 1),
 ('speech_output:en', 1),
 ('species', 1),
 ('source:incline', 1),
 ('slipway:type', 1),
 ('shop:historic', 1),
 ('shelter_type', 1),
 ('service_times', 1),
 ('service:bicycle:second_hand', 1),
 ('service:bicycle:parts', 1),
 ('service:bicycle:diy', 1),
 ('service:bicycle:dealer', 1),
 ('service:bicycle:chain_tool', 1),
 ('service area', 1),
 ('seamark:source', 1),
 ('seamark:light:group', 1),
 ('samtrans_drop_off_only', 1),
 ('sac_scale', 1),
 ('rooms', 1),
 ('roof:orientation', 1),
 ('restriction:truck', 1),
 ('relation', 1),
 ('ref:clipper:terminal', 1),
 ('ref:clipper:machine', 1),
 ('recycling_type', 1),
 ('recycling:glass_bottles', 1),
 ('recycling:cans', 1),
 ('rank', 1),
 ('railroad', 1),
 ('pupuseria', 1),
 ('proposed:height', 1),
 ('proposed:building:floors', 1),
 ('post_office:type', 1),
 ('population:date', 1),
 ('plant:output:electricity', 1),
 ('place_name', 1),
 ('pizza', 1),
 ('pedestrians', 1),
 ('pedestrian', 1),
 ('payment:visa', 1),
 ('payment:vertcoin', 1),
 ('payment:peercoin', 1),
 ('payment:mastercard', 1),
 ('payment:discover_card', 1),
 ('payment:coins', 1),
 ('payment:cash', 1),
 ('parking_space', 1),
 ('park:type', 1),
 ('parapet', 1),
 ('outerspatial:id', 1),
 ('outdoor', 1),
 ('opening_hours:url', 1),
 ('oneway:psv', 1),
 ('old_name_1', 1),
 ('old_name:vi', 1),
 ('note_3', 1),
 ('note_2', 1),
 ('nist:state_fips', 1),
 ('nist:fips_code', 1),
 ('nextbus:dir', 1),
 ('nextbus', 1),
 ('newaesthetic', 1),
 ('name_type', 1),
 ('name_base', 1),
 ('name:vi', 1),
 ('name:uk', 1),
 ('name:tt', 1),
 ('name:sv', 1),
 ('name:pa', 1),
 ('name:lt', 1),
 ('name:it', 1),
 ('name:eu', 1),
 ('name:da', 1),
 ('name:cs', 1),
 ('name:ar', 1),
 ('muni_route_ref', 1),
 ('mooring', 1),
 ('monitoring:bicycle', 1),
 ('memorial', 1),
 ('media:commons', 1),
 ('maxwidth', 1),
 ('maxstay', 1),
 ('map_type', 1),
 ('manager', 1),
 ('link', 1),
 ('language:en', 1),
 ('landuse_3', 1),
 ('landuse_2', 1),
 ('lamp_type', 1),
 ('is_in:iso_3166_2', 1),
 ('is_in:continent', 1),
 ('is_in:city', 1),
 ('inside_atm_capacity', 1),
 ('inside_at_operator', 1),
 ('hist_name', 1),
 ('highway:disused', 1),
 ('heritage:operator', 1),
 ('heritage', 1),
 ('health_specialty:speech_therapy', 1),
 ('health_specialty:physiotherapy', 1),
 ('hazard', 1),
 ('handrail:center', 1),
 ('guidepost', 1),
 ('golf:par', 1),
 ('golf:course', 1),
 ('gluten_free', 1),
 ('garage', 1),
 ('fuel:unleaded', 1),
 ('fuel:octane_91', 1),
 ('fuel:kerosene', 1),
 ('fuel:diesel', 1),
 ('fuel:biodiesel', 1),
 ('ford', 1),
 ('floating', 1),
 ('fenced', 1),
 ('event', 1),
 ('established', 1),
 ('emergency:notes', 1),
 ('elevator', 1),
 ('elevation', 1),
 ('drive_through_atm_capacity', 1),
 ('drink:juice', 1),
 ('disused:railway', 1),
 ('disused:name', 1),
 ('display', 1),
 ('diet:lactose_free', 1),
 ('diet:halal', 1),
 ('diaper', 1),
 ('designer', 1),
 ('description:zh', 1),
 ('description:en', 1),
 ('depth', 1),
 ('date', 1),
 ('csp:unitcode', 1),
 ('csp:globalid', 1),
 ('county:name', 1),
 ('county:ansi', 1),
 ('county:abbrev', 1),
 ('conveying', 1),
 ('content', 1),
 ('contact:yelp', 1),
 ('contact:rss', 1),
 ('contact:myspace', 1),
 ('contact:instagram', 1),
 ('contact:google_plus', 1),
 ('contact:atom', 1),
 ('community:en', 1),
 ('comedy', 1),
 ('coffee', 1),
 ('census:population', 1),
 ('category', 1),
 ('cash_in', 1),
 ('car', 1),
 ('camera:type', 1),
 ('camera:mount', 1),
 ('cafe', 1),
 ('cables', 1),
 ('bus:lanes', 1),
 ('building:use', 1),
 ('building:shape', 1),
 ('building.source:levels', 1),
 ('bubbler', 1),
 ('bridge:support', 1),
 ('border_type', 1),
 ('boat', 1),
 ('blind:description:en', 1),
 ('baby', 1),
 ('automated', 1),
 ('atm:operator', 1),
 ('atm:network', 1),
 ('amtrak_drop_off_only', 1),
 ('amenity_1', 1),
 ('amenity:disused', 1),
 ('alt_url', 1),
 ('alt_name:vi', 1),
 ('alt_name2', 1),
 ('advertising', 1),
 ('addr:suite', 1),
 ('addr:source:housenumber', 1),
 ('addr:pier', 1),
 ('addr:flats', 1),
 ('addr:door', 1),
 ('access:conditional', 1),
 ('abandoned:name', 1),
 ('abandoned:amenity', 1),
 ('Tiger:MTFCC', 1),
 ('OBJECTID', 1),
 ('MTFCC', 1),
 ('LAND_NAME', 1),
 ('LAND_ID', 1),
 ('DEPT', 1),
 ('CITY_OWNED', 1),
 ('CATEGORY', 1),
 ('AWATER', 1),
 ('AREAID', 1),
 ('ALAND', 1),
 ('99addr:city', 1)]
```

In [8]:

```
def count_postcodes(filename):
    postcodes = {}
    for event, elem in ET.iterparse(filename, events=('start', 'end')):
        if event == 'end':
            key = elem.attrib.get('k')
            if key == 'addr:postcode':
                postcode = elem.attrib.get('v')
                if postcode not in postcodes:
                    postcodes[postcode] = 1
                else:
                    postcodes[postcode] += 1
    return postcodes

postcodes = count_postcodes(OSM_FILE)
sorted_by_occurrence = [(k, v) for (v, k) in sorted([(value, key) for (key, va
lue) in postcodes.items()], reverse=True)]

print 'Postcode values and occurrence in San-francisco_california.osm:\n'
pprint.pprint(sorted_by_occurrence)
```

```
Postcode values and occurrence in San-francisco_california.osm:

[('94122', 4761),
 ('94116', 2118),
 ('94117', 1222),
 ('94133', 1111),
 ('94103', 800),
 ('94127', 707),
 ('94109', 453),
 ('94114', 350),
 ('94121', 265),
 ('94102', 209),
 ('94110', 179),
 ('94123', 140),
 ('94108', 132),
 ('94131', 130),
 ('94107', 130),
 ('94105', 128),
 ('94113', 80),
 ('94118', 75),
 ('94111', 65),
 ('94104', 64),
 ('94115', 54),
 ('94124', 42),
 ('94132', 35),
 ('94112', 31),
 ('94158', 25),
 ('94129', 8),
 ('94143', 7),
 ('94134', 6),
 ('90214', 6),
 ('94122-1515', 3),
 ('CA', 2),
 ('CA:94103', 1),
 ('CA94107', 1),
 ('CA 94133', 1),
 ('CA 94111', 1),
 ('CA 94108', 1),
 ('95115', 1),
 ('94188', 1),
 ('94166', 1),
 ('94164', 1),
 ('94130', 1),
 ('941234', 1),
 ('94121-3131', 1),
 ('9412', 1),
 ('94118-4504', 1),
 ('94118-1316', 1),
 ('94117-9991', 1),
 ('94103-3124', 1),
 ('94087', 1),
 ('94017', 1),
 ('94015', 1),
 ('94013', 1),
 ('41907', 1),
 ('14123', 1),
 ('1087', 1)]
```

In [11]:

```
def update_postcode(postcode):
    print postcode
    if re.match(r'^\d{5}$', postcode):
        return postcode
    try:
        return re.findall(r'^(\d{5})-\d{4}$', postcode)[0]
    except:
        pass
```

In [ ]:

```
import xml.etree.cElementTree as ET
import pprint
import re
import codecs
import json

OSM_FILE = "san-francisco_california.osm"
SF = "sanfran.osm.json"

lower = re.compile(r'^([a-z]|_)*$')
lower_colon = re.compile(r'^([a-z]|_)*:([a-z]|_)*$')
problemchars = re.compile(r'[=\+/&<>;\'"\?%#$@\,\. \t\r\n]')

CREATED = [ "version", "changeset", "timestamp", "user", "uid"]

def is_address(elem):
    if elem.attrib['k'][:5] == "addr:":
        return True

def is_postcode(elem):
    if elem.attrib['k'] == 'addr:postcode':
        return True


def shape_element(element):
    node = {}
    if element.tag == "node" or element.tag == "way" :

        address_info={}
        nd_info=[]
        node['type']=element.tag
        node['id']=element.attrib['id']
        if 'visible' in element.attrib.keys():
            node['visible']=element.attrib['visible']
        if 'lat' in element.attrib.keys():
            node['pos']=[float(element.attrib['lat']), float(element.attrib['l
on'])]
        node['created']={'version':element.attrib['version'],
                         'changeset':element.attrib['changeset'],
                         'timestamp':element.attrib['timestamp'],
                         'user':element.attrib['user'],
                         'uid':element.attrib['uid']}
        for tag in element.iter('tag'):
            p = problemchars.search(tag.attrib['k'])

            if p:
                continue
            elif is_address(tag): #addr is present
                if ':' in tag.attrib['k'][5:]: #checking for second :
                    continue
                else: #first colon processing
                    after_colon = tag.attrib['k'][5:]
                    if after_colon == 'postcode':
                        address_info[tag.attrib['k'][5:]] = update_postcode(ta
g.attrib['v'])
                    else :
                        address_info[tag.attrib['k'][5:]]=update_name(tag.attr
ib['v'], mapping)

            # if addr is present -- then you get string after the : using [5:]
            # take the string after : in a variable and check if that's postco
de
            # if yes then update_postcode
            # else update update_name

            else:
                node[tag.attrib['k']] = tag.attrib['v']
        if address_info != {}:
            node['address'] = address_info
        for tag2 in element.iter('nd'):
            nd_info.append(tag2.attrib['ref'])

        if nd_info != []:
            node['node_refs'] = nd_info
        return node
    else:
        return None


def process_map(file_in, pretty = False):

    file_out = "{0}.json".format(file_in)
    data = []
    with codecs.open(file_out, "w") as fo:
        for _, element in ET.iterparse(file_in):
            el = shape_element(element)
            if el:
                data.append(el)
                if pretty:
                    fo.write(json.dumps(el, indent=2)+"\n")
                else:
                    fo.write(json.dumps(el) + "\n")
    return data

def test():
    data = process_map(SF, True)
    pprint.pprint(data)



if __name__ == "__main__":
    test()
```

94117
94117
94117
94117
94102
94111
94115
94103
94110
94110
94107
94103
94103
94103-3124
94102
94158
94107
94103
94115
94115
94118
94133
94104
94107
94110
94132
94103
94117
94102
94102
94115
94121
94124
94110
94108
94112
94121
94123
94134
94114
94133
94105
94115
94102
14123
94117
94133
94112
94104
94133
94109
94109
94103
94104
94118
94118
94102

94102
94121
94121
94109
94109
94109
94109
94109
94103
94123
94133
94103
94121
94115
94110
94109
94109
94103
94110
94118
94118
94118
94114
94105
94122
94103
94114
94114
94117
94110
94107
94107
94111
94114
94114
94114
94114
94114
94114
94122
94107
94133
94103
94109
94127
94127
94127
94127
94127
94127
94127
94127
94127
94127
94127
94102
94127

94127
94127
94127
94111
94117
94117
94117
94102
94117
94114
94131
94158
94116
94116
94116
94116
94116
94116
94102
94102
94102
94109
94102
94102
94109
94109
94109
94109
94109
94109
94109
94109
94133
94114
94109
94132
94103
94103
94103
94103
94103
94103
94103
94103
94108
94108
94108
94102
94102
94102
94109
94109
94109
94109
94109
94109
94109

94117
94109
94109
94109
94109
94129
94103
94116
94107
94107
94111
94111
94117
94117
94131
94102
94102
94103
94117
94102
94102
94110
94117
94110
94133
94110
94121
94111
94104
94122
94103
94127
94108
94110
94103
94134
94117
94127
94127
94114
94103
94110
94103
94107
94103
94105
94105
94105
94104
94107
94115
94109
94133
94117
94117
94110
94109

94103
94107
94105
94117
94109
94108
94108
94109
94108
94102
94105
94114
94116
94116
94111
94103
94103
94114
94107
94133
94110
94102
94107
94107
94103
94105
94105
94105
94105
94105
94133
94121
94102
94105
94111
94103
94110
94110
94115
94115
94109
94122
94115
94109
94102
94109
94103
94103
94103
94108
94117
94110
94114
94114
94114
94114
94114

94114
94114
94114
94114
94103
94102
94102
94118
94110
94118
94118
94118
94121
94117
94117
94103
94103
94103
94132
94132
94132
94132
94132
94132
94132
94132
94103
94118
94121
94121
94103
94102
94121
94116
94105
94115
94107
94107
94107
94102
94105
94107
94121
94110
94103
94103
94103
94103
94103
94102
94115
94114
94103
94114
94109
94102
94112

94108
94108
94108
94117
94117
94117
94103
94102
94133
94117
94108
94117
94103
94114
94114
94114
94117
94115
94117
94117
94110
94110
94110
94122
94107
94118
94118
1087
94087
94114
94105
94107
94110
94103
94110
94109
94105
94105
94105
94105
94107
94112
94112
94112
94105
94103
94117
94133
94123
94121
94121
94121
94121
94103
94107
94107
94114

94103
94103
94114
94110
94133
94108
94109
94103
94115
94117
94117
94110
94109
94109
94109
94103
94107
94107
94107
94102
94110
94124
94110
94110
94118
94102
94108
94102
94110
94117
94102
94105
94105
94107
CA
94107
94117
94107
94111
94133
94118
94127
94107
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114

94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114

94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94123
94123
94114
94102
94110
94103
94123
94110
94143
94103
94117
94133
94133
94123
94123
94123
94123
94133
94107
94107
94107
94107
94110
94110
94105
94158
94143
94102
94107
94105
94110
94111
94102
94118
94103
94124
94102
94122
94132
94123
94109
94129
94108
94188
94124
94114
94122

94122
94122
94122
94102
94110
94107
94117
94122
94110
94110
94110
94103
94108
94117
94110
94122
94158
94127
41907
94131
94131
94122
94110
94104
94102
94102
94114
94102
94102
94110
94110
94115
94110
94103
94109
94118
94109
94123
94103
94109
94123
94109
94109
94118
94102
94107
94109
94111
CA 94133
94103
94105
94103
94103
94103
94103
94103
94103

94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94103
94111
94103
94103
94109
94110
94109
94109
94107
94110
94108
94122
94102

94105
94133
94114
94114
94114
94114
94114
94114
94114
94114
94114
94114
94124
94114
94114
94114
94110
94127
94127
94110
94110
94110
94110
94110
94110
94110
94110
94108
94103
94114
94133
94103
94124
94124
94124
94111
94110
94117
94117
94117
94122
94121
94121-3131
94121
94121
94121
94121
94121
94121
94103
94107
94111
94121
94109
94122
94103
94103