```
# Mount the google drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
files_path = '/content/drive/MyDrive/OSM_Tags'
```

```
file = files_path + 'export.geojson'
```

```
import json
```

```
# Opening the geojson file
with open('/content/drive/MyDrive/OSM_Tags/export.geojson', 'r') as f:
  data=json.load(f)
```

```
print(data)
```

```
{'type': 'FeatureCollection', 'generator': 'overpass-ide', 'copyright': 'The data included in this document is from www.openstreetmap.or
```

```
! pip install geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting geopandas
  Downloading geopandas-0.12.2-py3-none-any.whl (1.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 15.7 MB/s eta 0:00:00
Requirement already satisfied: shapely>=1.7 in /usr/local/lib/python3.9/dist-packages (from geopandas) (2.0.1)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.9/dist-packages (from geopandas) (1.4.4)
Collecting pyproj>=2.6.1.post1
  Downloading pyproj-3.5.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (7.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.8/7.8 MB 54.2 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from geopandas) (23.0)
Collecting fiona>=1.8
  Downloading Fiona-1.9.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 16.1/16.1 MB 48.5 MB/s eta 0:00:00
Requirement already satisfied: certifi in /usr/local/lib/python3.9/dist-packages (from fiona>=1.8->geopandas) (2022.12.7)
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.9/dist-packages (from fiona>=1.8->geopandas) (8.1.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.9/dist-packages (from fiona>=1.8->geopandas) (6.1.0)
Collecting munch>=2.3.2
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.9/dist-packages (from fiona>=1.8->geopandas) (22.2.0)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.9/dist-packages (from pandas>=1.0.0->geopandas) (1.22.4)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.9/dist-packages (from pandas>=1.0.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.9/dist-packages (from pandas>=1.0.0->geopandas) (2022.7.1)
Requirement already satisfied: six in /usr/local/lib/python3.9/dist-packages (from munch>=2.3.2->fiona>=1.8->geopandas) (1.16.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata->fiona>=1.8->geopandas) (3.1
Installing collected packages: pyproj, munch, cligj, click-plugins, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.9.2 geopandas-0.12.2 munch-2.5.0 pyproj-3.5.0
```

```
# Read the Geojson file using Geopandas package
import geopandas as gpd
dataframe1 = gpd.read_file('/content/drive/MyDrive/OSM_Tags/export.geojson')
```

```
dataframe1
```

	id	@id	access	addr:city	addr:district	addr:floor	addr:full	addr:housename	addr:house
0	relation/3718263	relation/3718263	NaN	NaN	NaN	NaN	NaN	NaN	
1	relation/4281628	relation/4281628	NaN	NaN	NaN	NaN	NaN	NaN	
2	relation/5364282	relation/5364282	NaN	NaN	NaN	NaN	NaN	NaN	
3	relation/10567786	relation/10567786	NaN	NaN	NaN	NaN	NaN	NaN	
4	relation/12276871	relation/12276871	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
550	node/10739481125	node/10739481125	NaN	NaN	NaN	G	NaN	NaN	
551	node/10739580130	node/10739580130	NaN	NaN	NaN	NaN	NaN	NaN	
552	node/10740580613	node/10740580613	NaN	NaN	NaN	G	NaN	NaN	

```
# Slicing the dataframe to have a look at the relevant columns
cols = [1, 88, 89, 93, 150]

dataframe_new = dataframe1[dataframe1.columns[cols]]

dataframe_new
```

	@id	name	name:en	name:mr	geometry
0	relation/3718263	अभियांत्रिकी महाविद्यालय, पुणे	College of Engineering, Pune	अभियांत्रिकी महाविद्यालय, पुणे	MULTIPOLYGON (((73.85501 18.52926, 73.85546 18...
1	relation/4281628	माळीवाडा(भोस्करवाडा)	NaN	NaN	POLYGON ((76.64206 19.24254, 76.64206 19.24221...
2	relation/5364282	सुयोग संकुल	Suyog Sankul	सुयोग संकुल	POLYGON ((73.78290 18.58923, 73.78307 18.58922...
3	relation/10567786	लोहगड	Lohgad	लोहगड	MULTIPOLYGON (((73.47772 18.70728, 73.47771 18...
4	relation/12276871	मोरया गोसावी मंदीर	Moraya Gosavi Mandir	मोरया गोसावी मंदीर	MULTIPOLYGON (((73.77860 18.62635, 73.77860 18...
...	...	...	...	...	...
550	node/10739481125	हरी ॐ ज्यूस बार	Hari Om juice bar	हरी ॐ ज्यूस बार	POINT (73.77348 18.59096)
551	node/10739580130	बालाजी सुपरमार्केट	Balaji supermarket	बालाजी सुपरमार्केट	POINT (73.77712 18.61082)
552	node/10740580613	तांबडा पांढरा	Tambda Pandhra	तांबडा पांढरा	POINT (73.78357 18.61174)
553	node/10742654055	साई दर्शिनी कॅफे	Sai Darshini Cafe	साई दर्शिनी कॅफे	POINT (73.76400 18.59628)

```
dataframe1.columns
```

```
Index(['id', '@id', 'access', 'addr:city', 'addr:district', 'addr:floor',
       'addr:full', 'addr:housename', 'addr:housenumber', 'addr:neighbourhood',
       ...
       'tower:type', 'tunnel', 'type', 'water', 'waterway', 'website',
```

```
       'wheelchair', 'wikidata', 'wikipedia', 'geometry'],
      dtype='object', length=151)
```

```
import pandas as pd
```

```
import regex as re
```

```
# Define the Unicode for Marathi Language
marathi_pattern = re.compile('[\u0900-\u097F]+')
```

```
type(marathi_pattern)
```

```
_regex.Pattern
```

```
# Define a function to check if the rows of name column has any marathi charcaters
def extract_marathi(row):
  name = row['name']
  matches = marathi_pattern.findall(name)
  if matches:
    row['name:mr'] = ''.join(matches)
    row['name'] = marathi_pattern.sub('', name)
    return row
```

```
# Apply the changes
dataframe1 = dataframe1.apply(extract_marathi, axis=1)
```

```
dataframe1
```

	id	@id	access	addr:city	addr:district	addr:floor	addr:full	addr:housename	addr:house
0	relation/3718263	relation/3718263	NaN	NaN	NaN	NaN	NaN	NaN	
1	relation/4281628	relation/4281628	NaN	NaN	NaN	NaN	NaN	NaN	

```
# Slicing the dataframe to have a look at the relevant columns after making changes
cols1 = [1, 88, 89, 93, 150]
C1_dataframe = dataframe1[dataframe1.columns[cols1]]


C1_dataframe
```

	@id	name	name:en	name:mr	geometry
0	relation/3718263	,	College of Engineering, Pune	अभियांत्रिकीमहाविद्यालयपुणे	MULTIPOLYGON (((73.85501 18.52926, 73.85546 18...
1	relation/4281628	()	NaN	माळीवाडाभोस्करवाडा	POLYGON ((76.64206 19.24254, 76.64206 19.24221...
2	relation/5364282		Suyog Sankul	सुयोगसंकुल	POLYGON ((73.78290 18.58923, 73.78307 18.58922...
3	relation/10567786		Lohgad	लोहगड	MULTIPOLYGON (((73.47772 18.70728, 73.47771 18...
4	relation/12276871		Moraya Gosavi Mandir	मोरयागोसावीमंदीर	MULTIPOLYGON (((73.77860 18.62635, 73.77860 18...
...	...	...	...	...	...
550	node/10739481125		Hari Om juice bar	हरीॐज्यूसबार	POINT (73.77348 18.59096)
551	node/10739580130		Balaji supermarket	बालाजीसुपरमार्केट	POINT (73.77712 18.61082)
552	node/10740580613		Tambda Pandhra	तांबडापांढरा	POINT (73.78357 18.61174)
553	node/10742654055		Sai Darshini Cafe	साईदर्शिनीकॅफे	POINT (73.76400 18.59628)
554	node/10743428464		Anyaa Spa	अन्यास्पा	POINT (73.78709 18.59541)

555 rows × 5 columns

```
dataframe1.to_csv('/content/drive/MyDrive/OSM_Tags/Output')
```

553	node/10742654055	node/10742654055	NaN	NaN	NaN	G	NaN	NaN	

```
#Transfering marathi values from name:en to name:mr
# Renaming the name:en tag to name_en
C1_dataframe.rename(columns = {'name:en':'name_en'}, inplace = True)
```

```
<ipython-input-240-c06294364849>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  C1_dataframe.rename(columns = {'name:en':'name_en'}, inplace = True)
```

```
# Define a function to check if the rows of name:en column has any marathi charcaters
def extract_marathin(row):
  name_en = row['name_en']
  matches = marathi_pattern.findall(name_en)
  if matches:
    row['name:mr'] = ''.join(matches)
    row['name_en'] = marathi_pattern.sub('', name_en)
    return row


# Apply the changes
C1_dataframe = C1_dataframe.apply(extract_marathin, axis=1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-243-4d40ad3da1df> in <module>
      1 # Apply the changes
----> 2 C1_dataframe = C1_dataframe.apply(extract_marathin, axis=1)
```

```
import string
```

```
# Define a function to check the rows of name:en column if there are any characters other than english and if there are transfer them to new
def is_english(s):
  return all(c in string.printable for c in s)
```

```
dataframe1['Non_english'] = ''
```

```
TypeError: expected string or buffer
```

```
for index, row in dataframe1.iterrows():
  if not is_english(str(row['name:en'])):
    non_english = ''.join([c for c in str(row['name:en']) if not is_english(c)])
    dataframe1.at[index, 'Non_english'] = non_english
```

```
dataframe1
```

	id	@id	access	addr:city	addr:district	addr:floor	addr:full	addr:housename	addr:house
0	relation/3718263	relation/3718263	NaN	NaN	NaN	NaN	NaN	NaN	
1	relation/4281628	relation/4281628	NaN	NaN	NaN	NaN	NaN	NaN	
2	relation/5364282	relation/5364282	NaN	NaN	NaN	NaN	NaN	NaN	
3	relation/10567786	relation/10567786	NaN	NaN	NaN	NaN	NaN	NaN	
4	relation/12276871	relation/12276871	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	
550	node/10739481125	node/10739481125	NaN	NaN	NaN	G	NaN	NaN	
551	node/10739580130	node/10739580130	NaN	NaN	NaN	NaN	NaN	NaN	
552	node/10740580613	node/10740580613	NaN	NaN	NaN	G	NaN	NaN	
553	node/10742654055	node/10742654055	NaN	NaN	NaN	G	NaN	NaN	
554	node/10743428464	node/10743428464	NaN	NaN	NaN	G	NaN	NaN	

555 rows × 152 columns

```
cols_new = [1, 88, 89, 93, 150, 151]
```

```
Newdataframe = dataframe1[dataframe1.columns[cols_new]]
```

Newdataframe

	@id	name	name:en	name:mr	geometry	Non_english
0	relation/3718263	,	College of Engineering, Pune	अभियांत्रिकीमहाविद्यालयपुणे	MULTIPOLYGON (((73.85501 18.52926, 73.85546 18...	
1	relation/4281628	()	NaN	माळीवाडाभोस्करवाडा	POLYGON ((76.64206 19.24254, 76.64206 19.24221...	
2	relation/5364282		Suyog Sankul	सुयोगसंकुल	POLYGON ((73.78290 18.58923, 73.78307 18.58922...	
3	relation/10567786		Lohgad	लोहगड	MULTIPOLYGON (((73.47772 18.70728, 73.47771 18...	
4	relation/12276871		Moraya Gosavi Mandir	मोरयागोसावीमंदीर	MULTIPOLYGON (((73.77860 18.62635, 73.77860 18...	
...	...	...	...	...	...	...
550	node/10739481125		Hari Om juice bar	हरीॐज्यूसबार	POINT (73.77348 18.59096)	
551	node/10739580130		Balaji supermarket	बालाजीसुपरमार्केट	POINT (73.77712 18.61082)	
552	node/10740580613		Tambda Pandhra	तांबडापांढरा	POINT (73.78357 18.61174)	
553	node/10742654055		Sai Darshini Cafe	साईदर्शिनीकॅफे	POINT (73.76400 18.59628)	

```
# Since all the rows of new column are empty which means there are only non english characters in name:en column
print(Newdataframe['Non_english'].isnull)
```

```
<bound method Series.isnull of 0
1
2
3
4
    ..
550
551
552
553
554
Name: Non_english, Length: 555, dtype: object>
```

Newdataframe

	@id	name	name:en	name:mr	geometry	Non_engli
0	relation/3718263	,	College of Engineering, Pune	अभियांत्रिकीमहाविद्यालयपुणे	MULTIPOLYGON (((73.85501 18.52926, 73.85546 18...	
1	relation/4281628	()	NaN	माळीवाडाभोस्करवाडा	POLYGON ((76.64206 19.24254, 76.64206 19.24221...	
2	relation/5364282		Suyog Sankul	सुयोगसंकुल	POLYGON ((73.78290 18.58923, 73.78307 18.58922...	
3	relation/10567786		Lohgad	लोहगड	MULTIPOLYGON (((73.47772 18.70728, 73.47771 18...	

```
def extract_marathi_new(row):
  Non_english = row['Non_english']
  matches = marathi_pattern.findall(Non_english)
  if matches:
    row['name:mr'] = ''.join(matches)
    row['Non_english'] = marathi_pattern.sub('', Non_english)
    return row
```

+ Code + Text

```
Newdataframe = Newdataframe.apply(extract_marathi_new, axis=1)
```

```
Newdataframe
```

```
0      None
1      None
2      None
3      None
4      None
       ...
550    None
551    None
552    None
553    None
554    None
Length: 555, dtype: object
```

```
dataframe1.to_csv('/content/drive/MyDrive/OSM_Tags/Final_output.csv')
```

```
geojson = dataframe1.to_json()
```

```
geojson
```


```
'{"type": "FeatureCollection", "features": [{"id": "0", "type": "Feature", "properties": {"id": "relation/3718263", "@id":
"relation/3718263", "access": null, "addr:city": null, "addr:district": null, "addr:floor": null, "addr:full": null, "add
r:housename": null, "addr:housenumber": null, "addr:neighbourhood": null, "addr:place": null, "addr:postcode": null, "add
r:state": null, "addr:street": null, "addr:subdistrict": null, "addr:suburb": null, "addr:unit": null, "air_conditioning":
null, "alt_name": null, "alt_name:en": "COEP", "alt_name:mr": null, "alt_name_1": null, "amenity": "college", "animal_shel
ter": null, "area": null, "artwork_type": null, "atm": null, "bench": null, "bicycle": null, "bin": null, "branch": null,
"brand": null, "brand:wikidata": null, "brand:wikipedia": null, "bridge": null, "building": null, "building:levels": null,
"building:material": null, "bus": null, "capacity": null, "charge": null, "check_date": null, "clothes": null, "contact:fa
```


✓ 0s completed at 4:19 PM