'''
I want to extract from my Facebook photos the Location of the photos and put it in a dataframe. I just wanted to created a database with all the places I have been lately.

We can also use this to analyse events (I put an example here for yoga related events), groups, analyse your activity, marketing research, etc. on Facebook.

Practice skills: Facebook Graph API, python DataFrame
 First of all your photos must have a location specified if you want to get this feature. There are several articles on the web about how you can access your photo location, but Facebook strip out the EXIF data when photos are uploaded for privacy reasons. So if you want to access this information, a location must be assigned to your photos.

Create APP

Second, if you want to have access to public data on Facebook you need to have an account as a Facebook developper. So, go to developers.facebook.com and create an account with Facebook login at the bottom of the page.

Then acces the link developers.facebook.com/tools/explorer. Go to "My apps" in the top right corner and select "add a new app". Choose the name of your app and a category and then "Create App ID".

Get Access Token

Again get back to the same link developers.facebook.com/tools/explorer. Now, you will see "Graph API Explorer" below "My Apps" . From here select your app and then "Get Token". From this drop down, select "Get User Access Token". Select permissions from the menu that appears and then "Get Access Token." This token is granted only for 2 hours, so you need an extended token. On developers.facebook.com/tools/accesstoken select "Debug" corresponding to "User Token" and then select "Extend Token Access". You now have an extended access token.

How to connect to Facebook and get Data

You also need to specify which public data you want to access. You can consult the link:
 https://developers.facebook.com/docs/graph-api/reference/v2.7/

On https://developers.facebook.com/docs/facebook-login/permissions#permissions you need to select from a list what kind of data do you want to extract.

For the case here you will need user_photo, user_locations, user_tagged_places, user_events, but you can also consider something else if you are interrested in other type of data.

```python
'''
Let's now extract information about our photos. The programm is in Python and c
ontain comments for most of the commands.

The following libraries are mandatory. If they are not installed , you can simp
ly install them in the command prompt window using, for example for urllib3, th
e command "pip install urllib3".
The urllib3 module open arbitrary resources by URL
The facebook libary is designed to support the Facebook Graph API
The request package is needed for a higher - level HTTP interface----apparentl
y also the most downloaded package
The pandas and the numpy are the classical package for working with DataFrame a
nd some math in python.
'''
import urllib3
import facebook
import requests
import pandas as pd
import numpy as np
```

```python
#Inserting here the extended token you obtained from the Facebook developers pa
ge, User acces token

token='EAAEdjFcBVa8BAKUn9F03MR2dk5h8simPZB0i1w3EjZB85YoDOqNwT5JogIzzZBmDE6rtTDM
01ZA8ZCOtCjtUeG2tcfnvWqnJ97Twjjlj3eptVDEkXkIyg68WTZBXlCfs5dvpAxcStOCtnsfC4rSFKi
eMSO4uf2CdIZD'
```

```python
# The Graph API is made up of the objects or nodes in Facebook (e.g.,events, ph
otos) and the connections or edges between them (e.g., friends, photo tags, et
c).

graph = facebook.GraphAPI(access_token=token, version = 2.7)
```

```python
'''
So you can access events, photos, etc. I am only searching in my photos so, "/m
e/photos".
There are two type existing here: uploaded or tagged. I've choosed the first on
e.
The default value for the number of photos is 25. If you want to extract a larg
er number you have to set manually the limit
Please remember that you need to have a location and also a description for eac
h extracted photo
'''
photos= graph.request('me/photos/uploaded?limit=30')
```

```
#print(photos)
```

```
print(len(photosList))
```

```
30
```

```
photosid={}
n=len(photosList)
for i in range(0,n):
    photosid[i]=photosList[i]['id']
```

```
# Let's extract the created_time and place for our photos
Placephoto={}
for i in range(0,n):
    r1 = graph.get_object(id=photosid[i],  fields='created_time,id,place')
    Placephoto[i]=r1['place']
```

```
# red the raw data into parse data using json

admins_json={}
for i in range(0,n):
    admins = requests.get("https://graph.facebook.com/v2.7/"+photosid[i]+"/admins?access_token="+token)
    admins_json[i] = admins.json()
```

```
#print(Placephoto)
```

```python
pp={}
df1=[]
df=[]
df_cities=[]
df3={}
for i in range(0,n):
    pp[i]=pd.DataFrame(Placephoto[i])
    df1.append(pp[i])
    df2=df1[i]
    df3[i]=pd.DataFrame(df2['location'])
    df=pd.DataFrame(df.append(df3[i]), columns=['location'])

# not very conventional, but if you first want to check if df is of DataFrame type
isinstance(df,pd.DataFrame)
# if you want to print all information: city, country, latitude, longitude, zip
print(df)
# if you only want to see the cities
print(df.loc['city',])
# if you only want to see the countries
print(df.loc['country',])
```

```
                              location
city                         Stuttgart
country                        Germany
latitude                        48.7788
longitude                       9.18064
zip                               70173
city                         Stuttgart
country                        Germany
latitude                        48.7788
longitude                       9.18064
zip                               70173
city                         Stuttgart
country                        Germany
latitude                        48.7788
longitude                       9.18064
zip                               70173
city                         Stuttgart
country                        Germany
latitude                        48.7788
longitude                       9.18064
zip                               70173
city                         Stuttgart
country                        Germany
latitude                        48.7788
longitude                       9.18064
zip                               70173
city                         Stuttgart
country                        Germany
latitude                        48.7788
longitude                       9.18064
zip                               70173
...                                 ...
country                           Spain
latitude                        28.2439
longitude                      -16.8403
city                            Munich
country                        Germany
latitude                        48.1507
longitude                       11.5907
street           Englischer Garten
zip                               98617
city                         KÃ¶tzting
country                        Germany
latitude                        49.1728
longitude                       12.8466
zip                               93444
city                         KÃ¶tzting
country                        Germany
```

```
latitude                    49.1728
longitude                   12.8466
zip                           93444
city                        Kötzting
country                     Germany
latitude                    49.1728
longitude                   12.8466
zip                           93444
city                        Stuttgart
country                     Germany
latitude                    48.7793
longitude                    9.1847
street      Oberer Schlossgarten 6
zip                           70173

[133 rows x 1 columns]
              location
city        Stuttgart
city        Stuttgart
city        Stuttgart
city        Stuttgart
city        Stuttgart
city        Stuttgart
city     Los Gigantes
city         Tenerife
city         Tenerife
city         Tenerife
city            Adeje
city  Los Cristianos
city  Los Cristianos
city        El Médano
city         Tenerife
city         Tenerife
city         Tenerife
city         Tenerife
city     Los Gigantes
city     Los Gigantes
city     Los Gigantes
city     Los Gigantes
city           Munich
city         Kötzting
city         Kötzting
city         Kötzting
city        Stuttgart
         location
country  Germany
country  Germany
country  Germany
country  Germany
```

```
country    Germany
country    Germany
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country      Spain
country    Germany
country    Germany
country    Germany
country    Germany
country    Germany
```

```python
# You can also extract information about public events for a specific search te
rm, for example: yoga
events = graph.request('/search?q=yoga&type=event&limit=100')
```

```python
eventList = events['data']
```

```python
#Assume you want information about the second event returned
eventid = eventList[4]['id']
# if you want information about all events use a for loop as for the photos cas
e above
```

```python
event1 = graph.get_object(id=eventid,
fields='category,cover,description,end_time,guest_list_enabled,interested_coun
t,is_canceled,is_page_owned,is_viewer_admin,maybe_count,noreply_count,owner,par
ent_group,place,ticket_uri,timezone,type,updated_time')
Description=event1['description']
Time=event1['updated_time']
Place=event1['place']
```

```
#use this for the case of yoga-events analysis publicly promoted on facebook
admins = requests.get("https://graph.facebook.com/v2.7/"+eventid+"/admins?acces
s_token="+token)
admins_json = admins.json()
print(Description)
print(Time)
print(Place)
```

The Rockies are proud to announce the 5th Annual Yoga Day at Coors Field! This
unique event gives fans the opportunity to combine two components of an active
lifestyle - yoga and baseball!

You and your fellow "yogis" are invited to join us for a special pre-game yoga
session held on the field! The private, on-field yoga session will begin at 8:3
0am and last approximately 60 minutes.

Your Yoga Day at Coors Field ticket package includes the following:

A ticket to watch the Rockies take on the Brewers
Exclusive, pre-game field access for the hour-long yoga session
Special Rockies-themed yoga-promotional item!

Yoga: 8:30am to 9:30am

First Pitch: 6:10pm
2017-03-15T21:32:26+0000
{u'id': u'50021802933', u'name': u'Coors Field', u'location': {u'city': u'Denve
r', u'zip': u'80205', u'country': u'United States', u'longitude': -104.99408474
763, u'state': u'CO', u'street': u'2001 blake st', u'latitude': 39.75530259429
9}}