# # Presidentielle2017 Twitter Analysis

## ### First, set in the environment your Twitter credentials and MonkeyLearn token

In [ ]:

```
import os

TWITTER_CONSUMER_KEY = os.environ.get('TWITTER_CONSUMER_KEY')
TWITTER_CONSUMER_SECRET = os.environ.get('TWITTER_CONSUMER_SECRET')
TWITTER_ACCESS_KEY = os.environ.get('TWITTER_ACCESS_KEY')
TWITTER_ACCESS_SECRET = os.environ.get('TWITTER_ACCESS_SECRET')
```

## Then, let's just download them!

In *tweets_quantity*, set the amount of tweets you'll download

In [ ]:

```
hashtag = '#Presidentielle2017'
tweets_file_name = 'Presidentielle2017_tweets.csv'
tweets_quantity = 5000
```

In [ ]:

```
import tweepy
from time import clock, sleep
import csv
import sys

auth = tweepy.OAuthHandler(TWITTER_CONSUMER_KEY, TWITTER_CONSUMER_SECRET)
auth.set_access_token(TWITTER_ACCESS_KEY, TWITTER_ACCESS_SECRET)
api = tweepy.API(auth)

start = clock()
with open(tweets_file_name, 'w') as f:
    writer = csv.writer(f)

    class StreamListener(tweepy.StreamListener):

        collected_tweets = 0

        def on_status(self, status):
            try:
                tweet = status.text
                tweet = tweet.replace('\n', '\\n')
                timePass = clock() - start
                if timePass % 60 == 0:
                    print ("Pfffiou, finally!! I have been working for", timePass, "seconds
                if not ('RT @' in tweet):  # Exclude re-tweets
                    writer.writerow([tweet])
                    self.collected_tweets += 1
                    if self.collected_tweets % 1000 == 0:
                        print ("Look: I have collected for you ", self.collected_tweets, "t
                    if self.collected_tweets == tweets_quantity:
                        print ("Done! Finished! Finito! Feito! YAAAY ")
                        return False
                    pass

            except Exception as e:
                sys.stderr.write('Encountered Exception:' + str(e))
                pass

        def on_error(self, status_code):
            print('Error: ' + repr(status_code))
            return True  # False to stop

        def on_delete(self, status_id, user_id):
            """Called when a delete notice arrives for a status"""
            print("Delete notice for" + str(status_id) + '. ' + str(user_id))
            return

        def on_limit(self, track):
            """Called when a limitation notice arrives"""
            return

        def on_timeout(self):
            """Called when there is a timeout"""
            sys.stderr.write('Timeout...')
            sleep(10)
            return True

    streamingAPI = tweepy.streaming.Stream(auth, StreamListener())
    streamingAPI.filter(track=[hashtag])
```

## Let's make sentiment analysis on the tweets, only if they are in English

In [ ]:

```
from monkeylearn import MonkeyLearn

MONKEYLEARN_API_KEY = os.environ.get('MONKEYLEARN_API_KEY')
ml = MonkeyLearn(MONKEYLEARN_API_KEY)

module_id = 'pi_SyZF3Kje' # This is the id of the pipeline that we are using

tweets = []

chunk_size = min(500, limit)
chunk_count, count = 0, 0
chunk = []

with open(tweets_file_name, 'r') as f:
    for row in csv.reader(f):
        chunk.append(row)
        count += 1
        chunk_count += 1
        if chunk_count == chunk_size:
            data = {
                "texts": [{"text": sample[0]} for sample in chunk]
            }
            res = ml.pipelines.run(module_id, data)
            i = 0
            for d in res.result['results']:
                if d['lang'][0]["label"] == "English" and d['lang'][0]["probability"] > 0.6
                    tweets.append({"text": chunk[i][0], "sentiment": d["sentiment_tweet"][0
                i += 1
            chunk = []
            chunk_count = 0
print('Total tweets:', count)
print('Total tweets in English:', len(tweets))
positive_tweets = [tweet for tweet in tweets if tweet['sentiment']['label'] == 'positive']
print('Positive tweets:', len(positive_tweets))
negative_tweets = [tweet for tweet in tweets if tweet['sentiment']['label'] == 'negative']
print('Negative tweets:', len(negative_tweets))
neutral_tweets = [tweet for tweet in tweets if tweet['sentiment']['label'] == 'neutral']
print('Neutral tweets:', len(neutral_tweets))
```

## Now, let's extract the keywords for each of the three categories. You'll get the 10 most relevant for each.

We'll take the first *sample_size* tweets for each category, join them in one text and extract the keywords with MonkeyLearn. If *sample_size* is too big, the lenght of the text may fail because it may reach the lenght limit for your plan.

In [ ]:

```
sample_size = 5000

values = {
    "negative":{"found":0, "text":""},
    "neutral":{"found":0, "text":""},
    "positive":{"found":0, "text":""}
}

for tweet in tweets:
    sent = tweet["sentiment"]
    if sent["probability"] > 0.6:
        if values[sent["label"]]["found"] < sample_size:
            values[sent["label"]]["text"] += "\n" + tweet["text"]
            values[sent["label"]]["found"] += 1

    if values["negative"]["found"] >= sample_size and values["neutral"]["found"] >= sample_
        break

module_id = 'ex_y7BPYzNG' # This is the id of the keyword extractor

for sentName, sentDict in values.items():
    print(sentName keywords: )
    print()
    res = ml.extractors.extract(module_id, [sentDict["text"]])
    for d in res.result[0]:
        print(d["keyword"])
    sentDict["keywords"] = res.result
```

## Finally, let's get the sentiment values for the tweets where they mention some words.

### Fell free to play with the list!

In [ ]:

```
#these are the keywords, you can add more
counts = {
    "macron":{},
    "marine lepen":{},
    "ps":{},
    "fn":{},
    "en marche":{}
}

for itemName, itemDict in counts.items():
    itemDict["positive"] = 0
    itemDict["neutral"] = 0
    itemDict["negative"] = 0

for tweet in tweets:
    for keyName, keyDict in counts.items():
        if keyName in tweet["text"].lower():
            keyDict[tweet["sentiment"]["label"]] += 1

print(counts)
```