In [1]:
```
import twint
import tweepy as tw
import pandas as pd
import credentials
import re
import nest_asyncio
```

In [69]:
```
class Twitter_Authentication:
    #Function used to connect with the Twitter API
    def app_authenticate(self):
        auth =tw.OAuthHandler(credentials.API_KEY, credentials.KEY_SECRET)
        auth.set_access_token(credentials.ACCESS_TOKEN,credentials.ACCESS_TOKEN_SECRET)
        return auth
class Data_preprocessing:
    #Function use for dealing with missing data
    def handle_missing_data(self,df,column,fill_with):
        df[column]=df[column].replace(np.nan,fill_with)

    # Function used to remove and clean tweets from special chracters
    def clean_tweets_content(self, tweet):
        return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)|(RT)", " ", tweet).split())

    #This function is used to get the percentage of dataset column
    def get_col_percentage(self,col,df):

        total=df[col].value_counts()
        percentage=round(df[col].value_counts(dropna=False,normalize=True)*100,3)
        # or percentage=round((df[col]/df[col].sum())*100,2)
        res=pd.concat([total,percentage],axis=1,keys=["Total No.","Percentage"])
        #res['Percentage'] = res['Percentage'].astype(str) + '%'
        return res


class Tweets_Scraping_user:
    #Impelementation of the constructor
    def __init__(self,scraper="twint"):
        self.scraper=scraper
        if scraper.lower()=="twint":
            self.conf=twint.Config()
        else:
            au=Twitter_Authentication()
            self.auth =au.app_authenticate()
            self.api = tw.API(self.auth, wait_on_rate_limit=True)

    def get_tweets_from_user_tweepy(self,user):
        tweets=tw.Cursor(self.api.user_timeline,screen_name=user).items(50)
        collected_tweets=[i for i in tweets]
        return collected_tweets

    def tweets_to_DataFrame(self,tweets):
        pre=Data_preprocessing()
        tweets_data=[{
            "tweet":pre.clean_tweets_content(tweet.text),
            "tweet_len":len(tweet.text),
            "date":tweet.created_at,
            "source":tweet.source,
            "likes":tweet.favorite_count,
            "retweets":tweet.retweet_count,
            "No._followers":tweet.user.followers_count,
            "lang":tweet.lang
        } for tweet in tweets]

        df=pd.DataFrame(tweets_data)
        return df

    def scrape_user(self,user):
        if self.scraper.lower()=="twint":
            nest_asyncio.apply()
            self.conf.Username =user
            self.conf.Limit=50
            self.conf.Lang = "en"
            self.conf.Hide_output=True
            self.conf.Pandas=True
            twint.run.Search(self.conf)
            df = twint.storage.panda.Tweets_df
            return df
        else:
            tweets=self.get_tweets_from_user_tweepy(user)
            df=self.tweets_to_DataFrame(tweets)
            return df

    def scrape_twt_user_info(self,user):
        if self.scraper.lower()=="twint":
            nest_asyncio.apply()
            self.conf.Username = "google"
            self.conf.Hide_output=True
            self.conf.Store_object = True
```

```
            twint.run.Lookup(self.conf)
            u = twint.output.users_list[0]
            df_info=pd.DataFrame({"id":u.id,
                                  "name":u.name,
                                  "bio":u.bio,
                                  "followers_num":u.followers,
                                  "tweets_num":u.tweets,
                                  "likes_num":u.likes},index=[0])
            return df_info

        else:
            u=self.api.get_user(screen_name=user)
            df_info=pd.DataFrame({"id":u.id,
                                  "name":u.screen_name,
                                  "bio":u.description,
                                  "followers_num":u.followers_count,
                                  "tweets_num":u.statuses_count,
                                  "likes_num":u.favourites_count},index=[0])
            return df_info
```

In [70]:
```
t=Tweets_Scraping_user()
```

In [71]:
```
u=t.scrape_twt_user_info("google")
```

In [72]:
```
u
```

Out[72]:

	id	name	bio	followers_num	tweets_num	likes_num
0	20536157	Google	#HeyGoogle	26005239	162477	3223