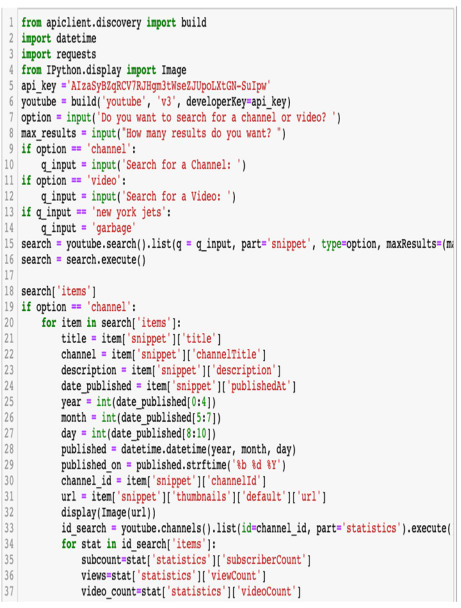
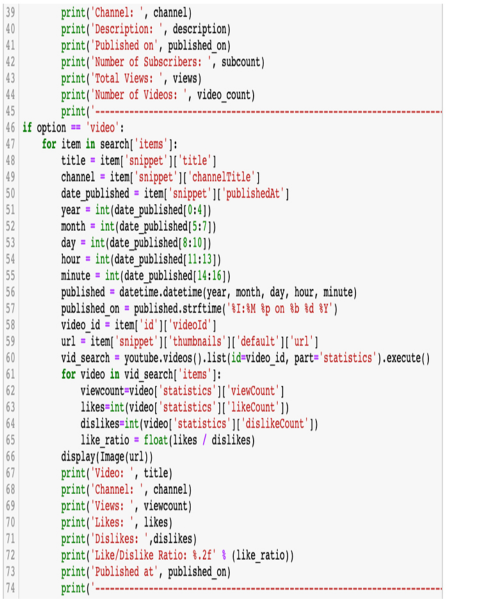


```python
from apiclient.discovery import build
import datetime
import requests
from IPython.display import Image
api_key ='AIzaSyBZqRCV7RJHgm3tWseZJUpoLXtGN-SuIpw'
youtube = build('youtube', 'v3', developerKey=api_key)
option = input('Do you want to search for a channel or video? ')
max_results = input("How many results do you want? ")
if option == 'channel':
    q_input = input('Search for a Channel: ')
if option == 'video':
    q_input = input('Search for a Video: ')
if q_input == 'new york jets':
    q_input = 'garbage'
search = youtube.search().list(q = q_input, part='snippet', type=option, maxResults=(ma
search = search.execute()

search['items']
if option == 'channel':
    for item in search['items']:
        title = item['snippet']['title']
        channel = item['snippet']['channelTitle']
        description = item['snippet']['description']
        date_published = item['snippet']['publishedAt']
        year = int(date_published[0:4])
        month = int(date_published[5:7])
        day = int(date_published[8:10])
        published = datetime.datetime(year, month, day)
        published_on = published.strftime('%b %d %Y')
        channel_id = item['snippet']['channelId']
        url = item['snippet']['thumbnails']['default']['url']
        display(Image(url))
        id_search = youtube.channels().list(id=channel_id, part='statistics').execute(
        for stat in id_search['items']:
            subcount=stat['statistics']['subscriberCount']
            views=stat['statistics']['viewCount']
            video_count=stat['statistics']['videoCount']
```

YOUTUBE DATA API



```python
        print('Channel: ', channel)
        print('Description: ', description)
        print('Published on', published_on)
        print('Number of Subscribers: ', subcount)
        print('Total Views: ', views)
        print('Number of Videos: ', video_count)
        print('------------------------------------------------------------------
if option == 'video':
    for item in search['items']:
        title = item['snippet']['title']
        channel = item['snippet']['channelTitle']
        date_published = item['snippet']['publishedAt']
        year = int(date_published[0:4])
        month = int(date_published[5:7])
        day = int(date_published[8:10])
        hour = int(date_published[11:13])
        minute = int(date_published[14:16])
        published = datetime.datetime(year, month, day, hour, minute)
        published_on = published.strftime('%I:%M %p on %b %d %Y')
        video_id = item['id']['videoId']
        url = item['snippet']['thumbnails']['default']['url']
        vid_search = youtube.videos().list(id=video_id, part='statistics').execute()
        for video in vid_search['items']:
            viewcount=video['statistics']['viewCount']
            likes=int(video['statistics']['likeCount'])
            dislikes=int(video['statistics']['dislikeCount'])
            like_ratio = float(likes / dislikes)
        display(Image(url))
        print('Video: ', title)
        print('Channel: ', channel)
        print('Views: ', viewcount)
        print('Likes: ', likes)
        print('Dislikes: ',dislikes)
        print('Like/Dislike Ratio: %.2f' % (like_ratio))
        print('Published at', published_on)
        print('------------------------------------------------------------------
```

WHAT OUR CODE DOES:

- Collects user input about what it is they want to search for
- Calls the API, which provides information and statistics for the video or channel that the user searched for
- Displays important information, as well as thumbnails, in a simple and easy to read output

HOW IT WORKS:

- Our program reads data from the YouTube Data API V3, and displays information about certain videos. From there, we dig through the data and select only the important parts, such as view count and like/dislike ratio for videos, or subscriber count and number of videos for channels. Our program streamlines the data and displays the most important elements all in one place.

OUTPUTS FOR "VIDEO":

1. THUMBNAIL
2. TITLE AND CHANNEL
3. DATE POSTED
4. VIEW COUNT
5. LIKES, DISLIKES, AND RATIO

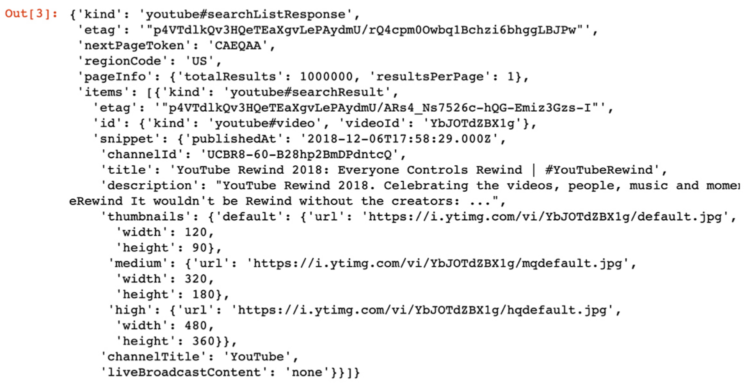
INPUTS:

1. CHOICE BETWEEN SEARCHING FOR A "CHANNEL" OR A "VIDEO"
2. NUMBER OF DESIRED RESULTS
3. THE USER'S SPECIFIC SEARCH QUERY

OUTPUTS FOR "CHANNEL":

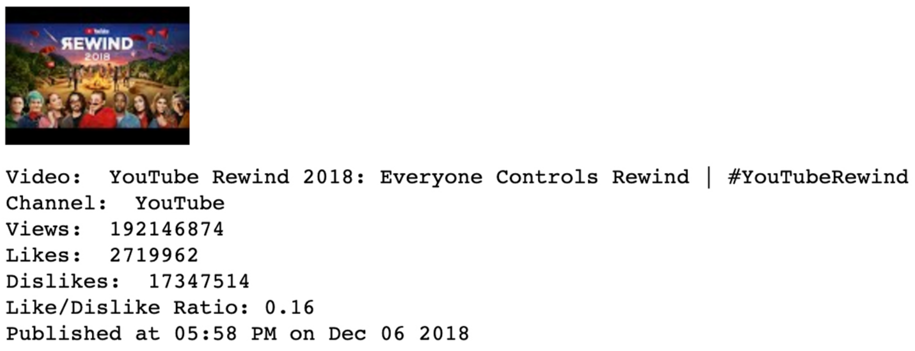
1. THUMBNAIL
2. TITLE AND DESCRPTION
3. DATE CREATED
4. SUBSCRIBER COUNT
5. TOTAL CHANNEL VIEWS

ORIGINAL OUTPUT:



```
Out[3]: {'kind': 'youtube#searchListResponse',
         'etag': '"p4VTdlkQv3HQeTEaXgvLePAydmU/rQ4cpm0Owbq1Bchzi6bhggLBJPw"',
         'nextPageToken': 'CAEQAA',
         'regionCode': 'US',
         'pageInfo': {'totalResults': 1000000, 'resultsPerPage': 1},
         'items': [{'kind': 'youtube#searchResult',
           'etag': '"p4VTdlkQv3HQeTEaXgvLePAydmU/ARs4_Ns7526c-hQG-Emiz3Gzs-I"',
           'id': {'kind': 'youtube#video', 'videoId': 'YbJOTdZBX1g'},
           'snippet': {'publishedAt': '2018-12-06T17:58:29.000Z',
            'channelId': 'UCBR8-60-B28hp2BmDPdntcQ',
            'title': 'YouTube Rewind 2018: Everyone Controls Rewind | #YouTubeRewind',
            'description': "YouTube Rewind 2018. Celebrating the videos, people, music and mome
        eRewind It wouldn't be Rewind without the creators: ...",
            'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/YbJOTdZBX1g/default.jpg',
              'width': 120,
              'height': 90},
             'medium': {'url': 'https://i.ytimg.com/vi/YbJOTdZBX1g/mqdefault.jpg',
              'width': 320,
              'height': 180},
             'high': {'url': 'https://i.ytimg.com/vi/YbJOTdZBX1g/hqdefault.jpg',
              'width': 480,
              'height': 360}},
            'channelTitle': 'YouTube',
            'liveBroadcastContent': 'none'}}]}
```

FINAL OUTPUT:



```
Video:  YouTube Rewind 2018: Everyone Controls Rewind | #YouTubeRewind
Channel:  YouTube
Views:  192146874
Likes:  2719962
Dislikes:  17347514
Like/Dislike Ratio: 0.16
Published at 05:58 PM on Dec 06 2018
```

JAY GRANT • JOHN FENTON • SANTIAGO HERNANDEZ