```
# INSTALL REQUIRED PACKAGES
# requests and pandas are python libraries, to install via pip, Python's package manager
# requests is for making HTTP requests (like fetching web pages)
# pandas is for working with structured data (tables, CSVs, etc)
# quiet makes the output cleaner
# each of the import lines imports a Python module or library

!pip install --quiet requests pandas

import requests
import re
import json
import pandas as pd

# RECEIVE VIDEO LINKS AND GIVE THEM NAMES
# add more as needed

videos = [
("Irish Trans Pride Parade", "https://www.youtube.com/watch?v=kf58rHXyWJk"),
("Malta Travel", "https://www.youtube.com/watch?v=Ys0V4wx0WsY"),
("Swedish Trans Experience", "https://www.youtube.com/watch?v=k6NYnGpBJbs"),
("Irish Young LGBs", "https://www.youtube.com/watch?v=gMRpVek5KD8"),
("MCC Brussels", "https://www.youtube.com/watch?v=t_6lYfTOJkQ"),
("Irish News Government", "https://www.youtube.com/watch?v=FLWyrbwLgnM"),
("Irish TV Safety", "https://www.youtube.com/watch?v=mNpxpVgwXuQ"),
("Euronews Report", "https://www.youtube.com/watch?v=1mNJaSO4694"),
("DW Report Berlin", "https://www.youtube.com/watch?v=xPPe3jrPGE8"),
("DW Report Pope", "https://www.youtube.com/watch?v=KdCJXf3ctnE"),
("DW Report UK", "https://www.youtube.com/watch?v=ta0H-ESCinw"),
]

# LABEL REQUEST TO SERVER WITH HEADERS
# this tells the web server what kind of device/browser is making the request
```

```python
# websites use this to customize responses or block bots

headers = {
   "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
}

# EXTRACT PUBLISH DATE AND VIEW COUNT
# this loops over our list of videos, making requests to each of their URLs
# it extracts JSON data from the HTML to get publish date and view count
# it then stores the results under headings we provide

results = []

for video_name, video_url in videos:
   print(f"Processing: {video_name} ({video_url})")
   # this lets us see what video the script is currently working on

   try:
       response = requests.get(video_url, headers=headers)
       # this makes HTTP GET requests to the video pages

       html = response.text
       # this gets the full HTML contents as text

       match = re.search(r'var ytInitialPlayerResponse = ({.*?});', html)
       if not match:
           match = re.search(r'ytInitialPlayerResponse\s*=\s*({.+?});', html)
       # ytInitialPlayerResponse is a JavaScript variable containing the video metadata

       if match:
           data_json = match.group(1)
           data = json.loads(data_json)
```

```
            video_details = data.get("videoDetails", {})
            publish_date = data.get("microformat", {}).get("playerMicroformatRenderer", {}).get("publishDate",
None)
            view_count = video_details.get("viewCount", None)
        # this extracts the data we want and converts it into a Python dictionary so it can be worked with

            # Then format view counts as integers
            try:
                view_count = int(view_count)
            except:
                view_count = None

            # Then store results
            results.append({
                "video_name": video_name,
                "video_url": video_url,
                "publish_date": publish_date or "Not found",
                "view_count": view_count if view_count is not None else "Not found"
            })
        else:
            print("Could not extract JSON data from this video.")
            results.append({
                "video_name": video_name,
                "video_url": video_url,
                "publish_date": "Error",
                "view_count": "Error"
            })

    except Exception as e:
        print(f"Error processing video: {e}")
        results.append({
            "video_name": video_name,
            "video_url": video_url,
```

```
            "publish_date": "Error",
            "view_count": "Error"
        })

# SAVE TO CSV
df = pd.DataFrame(results)
csv_filename = "video_metadata.csv"
df.to_csv(csv_filename, index=False)

print(f"\nSaved metadata for {len(df)} videos to {csv_filename}")

# DOWNLOAD CSV FROM GOOGLE COLAB
try:
    from google.colab import files
    files.download(csv_filename)
except ImportError:
    pass  # Not in Colab
```