```
1 pip install yt-dlp
```

```
Collecting yt-dlp
  Downloading yt_dlp-2025.2.19-py3-none-any.whl.metadata (171 kB)
  ──────────────────────────────────────── 171.9/171.9 kB 2.4 MB/s eta 0:00:00
  Downloading yt_dlp-2025.2.19-py3-none-any.whl (3.2 MB)
  ──────────────────────────────────────── 3.2/3.2 MB 29.4 MB/s eta 0:00:00
Installing collected packages: yt-dlp
Successfully installed yt-dlp-2025.2.19
```

⌄ Video to Audio generation

```
 1 import yt_dlp
 2
 3 def download_audio(video_url):
 4     ydl_opts = {
 5     'format': 'bestaudio/best',
 6     'outtmpl': '/content/audio.mp3',
 7     'cookiefile': '/content/cookies.txt',  # Update this path
 8     'postprocessors': [{
 9         'key': 'FFmpegExtractAudio',
10         'preferredcodec': 'mp3',
11         'preferredquality': '192',
12     }],
13   }
14
15     with yt_dlp.YoutubeDL(ydl_opts) as ydl:
16         ydl.download([video_url])
17
18     return "/content/audio.mp3"
19
20 video_url = input("Enter YouTube video URL: ")
21 file_path = download_audio(video_url)
22 print(f"Downloaded audio: {file_path}")
```

```
Enter YouTube video URL: https://youtu.be/sK8SILOM37I
[youtube] Extracting URL: https://youtu.be/sK8SILOM37I
[youtube] sK8SILOM37I: Downloading webpage
[youtube] sK8SILOM37I: Downloading tv client config
[youtube] sK8SILOM37I: Downloading player 82345d49
[youtube] sK8SILOM37I: Downloading tv player API JSON
[youtube] sK8SILOM37I: Downloading ios player API JSON
[youtube] sK8SILOM37I: Downloading m3u8 information
[info] sK8SILOM37I: Downloading 1 format(s): 251
[download] Destination: /content/audio.mp3
[download] 100% of   39.37MiB in 00:00:00 at 47.13MiB/s
[ExtractAudio] Destination: /content/audio.mp3.mp3
Deleting original file /content/audio.mp3 (pass -k to keep)
Downloaded audio: /content/audio.mp3
```

```
 1 import os
 2
 3 # Check the files in /content directory
 4 print("Files in content:", os.listdir("/content"))
 5
 6 # Fix the file name if needed
 7 old_path = "/content/audio.mp3.mp3"
 8 new_path = "/content/audio.mp3"
 9
10 if os.path.exists(old_path):
11     os.rename(old_path, new_path)
12     print(f"Renamed: {old_path} → {new_path}")
13 else:
14     print("Audio file not found!")
```

```
Files in content: ['.config', 'audio.mp3.mp3', 'cookies.txt', 'sample_data']
Renamed: /content/audio.mp3.mp3 → /content/audio.mp3
```

```
1 import IPython.display as ipd
2
3 # Set the correct file path
4 audio_path = "/content/audio.mp3"
5
6 # Play the audio with the correct sample rate (e.g., 44100 Hz)
7 ipd.Audio(audio_path, rate=44100)
```

> Audio to Transcript Generation

[ ] ↳ 2 cells hidden

> Segmenting Transcripts (Using Segbot model)

[ ] ↳ 2 cells hidden