

# 1. Một số thao tác cơ bản với Video

## 01. Tổng quan về MoviePy

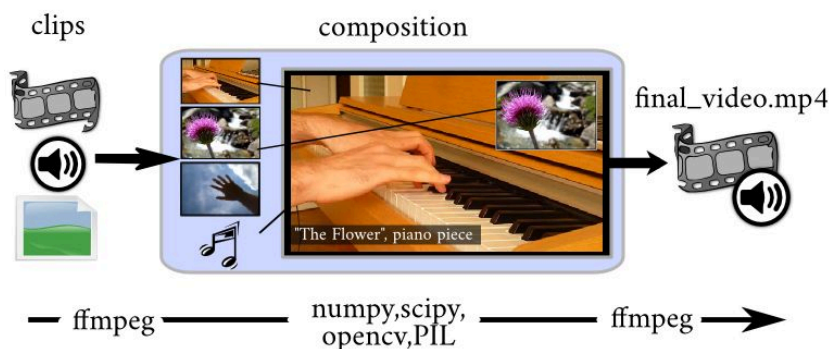
### Ưu điểm của MoviePy

- Đơn giản:** Các thao tác cơ bản có thể được thực hiện trên một dòng lệnh, mã lệnh dễ học và dễ hiểu đối với người mới bắt đầu.
- Linh hoạt:** Người dùng có toàn quyền kiểm soát các khung hình của video và âm thanh, đồng thời việc tạo hiệu ứng riêng cũng dễ dàng như Py.
- Portable:** Mã sử dụng phần mềm rất phổ biến như Numpy và FFMPEG. Và có thể chạy trên hầu hết mọi máy với hầu hết mọi phiên bản Python.

### Nhược điểm của MoviePy :

- MoviePy hiện chưa thể truyền phát video (đọc từ webcam hoặc hiển thị video trực tiếp trên máy tính ở xa)
- Nó không thực sự được thiết kế để xử lý video liên quan đến nhiều khung hình liên tiếp của một bộ phim (giống như ổn định video, bạn sẽ cần một phần mềm khác cho việc đó)
- Các vấn đề về bộ nhớ có thể phát sinh nếu người dùng sử dụng nhiều nguồn video, âm thanh và hình ảnh cùng một lúc (>100)

## 02. Cài đặt MoviePy và một số thư viện liên quan



### Sử dụng lệnh pip:

```
1 !pip install moviepy
2 !apt install imagemagick
3 !apt install libmagick++-dev
```

### Phương pháp thủ công:

Tải xuống các nguồn: trên [PyPI](#) hoặc [Github](#) → giải nén → gõ lệnh cmd:

```
python setup.py install
```

Tham khảo cài đặt: <https://zulko.github.io/moviepy/install.html>

## 03. Một số thao tác cơ bản

### 1. Đọc Video

**Video:** mytravel.mp4

**Import thư viện:**

```
1 from moviepy.editor import *
```

Để đọc video ta sử dụng lệnh:

```
1 clip = VideoFileClip("/content/mytravel.mp4")
```

### 2. Hiển thị Video

Để hiển thị video ta sử dụng lệnh: **ipython\_display** với **width** là kích thước video

```
1 clip.ipython_display(width = size)
```

### 3. Xuất Video

Để xuất video ta sử dụng lệnh **write\_videofile**

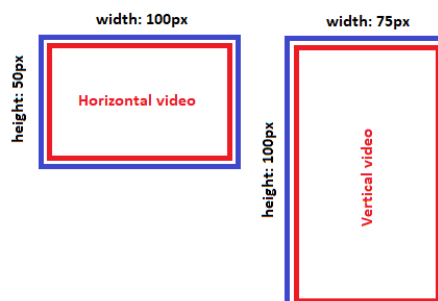
```
1 clip.write_videofile("Name you want".mp4,encoding)
```

### 4. Hiển thị độ dài video

Để hiển thị độ dài video ta sử dụng lệnh **duration**

```
1 clip= VideoFileClip("Video_1.mp4")  
2 print(str(clip.duration))
```

### 5. Trích xuất kích thước video



Một kích thước video được xác định bởi 2 yếu tố độ rộng (width - w) và độ cao (height - h)

Để thực hiện trích xuất kích thước video ta sử dụng lệnh size:

```
1 value = clip.size
```

Để thực hiện trích xuất độ rộng của một video bất kỳ ta sử dụng lệnh:

```
1 width = clip.w
```

## 6. Trích xuất tên file video

Để thực hiện trích xuất tên file video ta sử dụng lệnh filename

```
1 name = clip.filename
```

## 7. Trích xuất âm thanh video

Để thực hiện trích xuất âm thanh video ta sử dụng lệnh audio

```
1 audioclip = clip.audio
```

## 8. Trích xuất FPS của video

Tốc độ khung hình là tốc độ hiển thị những hình ảnh đó hoặc tốc độ bạn "lật" qua cuốn sách và thường được biểu thị là "khung hình trên giây" hoặc FPS. Mỗi hình ảnh đại diện cho một khung hình, vì vậy nếu video được quay và phát lại ở tốc độ 24fps, điều đó có nghĩa là mỗi giây video hiển thị 24 hình ảnh tĩnh riêng biệt.

Để thực hiện trích xuất tốc độ FPS video ta sử dụng lệnh fps

```
1 rate = clip.fps
```

## 9. Trích xuất video

Để trích xuất video ta sử dụng lệnh subclip:

```
1 clip = clip.subclip(Thời gian bắt đầu, Thời gian kết thúc)
```

## 10. Thay đổi kích thước video

Để trích xuất video ta sử dụng lệnh resize(n) với n là độ scale của video

```
1 clip = clip.resize(0.5)
```

## 11. Tạo bản sao video

Để tạo thêm một bản sao video ta sử dụng lệnh copy

```
copied_clip = clip.copy()
```

## 12. Xoay video

Để thực hiện thao tác xoay video ta sử dụng lệnh `rorate` với `degree` là góc xoay

```
1 clip = clip.rotate(degree)
```

## 13. Giảm âm lượng âm thanh

Để thực hiện thao tác giảm âm lượng ta sử dụng lệnh `volumex`

```
1 Reduce the audio volume (volume x 0.8)
2 clip = clip.volumex(0.8)
```

## 14. Thiết lập video phát tự động

Để video tự động phát khi trình chiếu ta sử dụng lệnh `ipython_display(autoplay = giá trị)`

```
1 clip.ipython_display(autoplay = 1)
```

Để video tự động lặp lại sau `n` lần ta hiệu chỉnh thuộc tính `loop` có trong `ipython_display`

```
1 clip.ipython_display(loop = n)
```

Để kiểm tra xem video có đang phát tại thời điểm nhất định ta sử dụng lệnh `is_playing(n)`, trong đó `n` là thời gian phát video

```
1 value = clip.is_playing(2)
```

## 15. Thêm nội dung vào Video

Để thực hiện thêm nội dung vào Video ta thực hiện các bước sau:

1. Tạo nội dung vào video thông qua lệnh **TextClip**

```
1 txt_clip = TextClip("Nội dung", fontsize = kích thước chữ, color = 'màu sắc')
```

2. Thiết lập vị trí và thời gian hiển thị sử dụng lệnh **set\_pos**

```
1 #setting position of text in the center and duration will be 10 seconds
2 txt_clip.set_pos('center').set_duration(10)
```

3. Thêm nội dung vào Video sử dụng lệnh **CompositeVideoClip**

```
video = CompositeVideoClip([clip, txt_clip])
```

## 16. Gán âm thanh vào Video

Để thực hiện gán âm thanh vào Video ta sử dụng lệnh `set_audio` và thực hiện:

### 1. Tải tệp âm thanh

```
1 # loading audio file
2 audioclip = AudioFileClip("allwell.mp3")
```

### 2. Gán âm thanh

```
1 # adding audio to the video clip
2 videoclip = clip.set_audio(audioclip)
```

## 17. Ghép các tập tin video/clips

Để thực hiện ghép các tập tin video:

### 1. Khởi tạo tập tin cần ghép

```
1 clip_1= moviepy.editor.VideoFileClip("Video_1.mp4")
2 clip_2= moviepy.editor.VideoFileClip("Video_2.mp4")
```

### 2. Sử dụng lệnh `concatenate_videoclips` để thực hiện ghép video

```
1 Mearged_video=moviepy.editor.concatenate_videoclips([clip_1,clip_2])
```

### 3. Lưu kết quả

```
1 # Saving File as output.mp4 in same folder
2 # libx264 is encoding lib for creating video stream(H.264)
3 Mearged_video.write_videofile("Output.mp4", codec='libx264')
4 print("Done")
```

## 18. Trích xuất độ dài video

Để trích xuất độ dài video ta sử dụng lệnh `duration` và `print`

```
1 print(str(vid.duration))
```

## 19. Thiết lập lề (margins)

Để thiết lập lề cho video ta sử dụng lệnh `margin`

```
clip = clip.margin(chỉ số)
```

## 20. Phản chiếu (Mirror) video

Để phản chiếu video theo trục x ta sử dụng lệnh `fx(vfx.mirror_x)`

```
1 clip = clip.fx(vfx.mirror_x)
```

## 21. Điều chỉnh tốc độ video

Để điều chỉnh tốc độ video ta sử dụng lệnh `speedx`

```
1 speed_clip = clip.speedx(speed)
```

## 22. Chèn logo vào Video

Để điều chỉnh tốc độ video ta sử dụng phương thức **ImageClip**

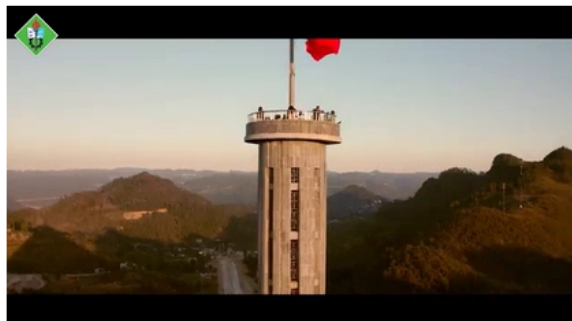
```
1 ImageClip('path to image').
2     .set_duration(clip.duration)
3     .resize(height=values) # if you need to resize...
4     .margin(right=values, top=values, opacity=values) # Thiết lập các thuộc tính vị trí, độ mờ
5     .set_pos(("align", "align"))
```

Ví dụ:

```
1 logo = (ImageClip("/content/spkt.png")
2         .set_duration(clip.duration)
3         .resize(height=30)
4         .margin(right=8, top=8, opacity=50)
5         .set_pos(("left", "top")))
```

Chèn logo vào video sử dụng phương thức **CompositeVideoClip**

```
1 final = CompositeVideoClip([clip, logo])
```



## 23. Hiển thị nhiều video trên cùng khung nhìn

Để hiển thị cùng lúc nhiều video trên cùng một khung nhìn ta sử dụng `clips_array(clips)`

```
1 clips = [[clip1, clip2],[clip3, clip4]]
2 final = clips_array(clips)
```



## 24. Cắt bỏ đoạn clip khỏi tệp Video

Để thực hiện cắt bỏ đoạn clip khỏi tệp video ta sử dụng lệnh `cutout(ts, te)`. Trong đó: `ts` là thời gian bắt đầu - `te` là thời gian kết thúc

```
1 clip = clip.cutout(3, 10)
```

## 04. BÀI TẬP

**Bài tập 1:** Đọc và hiển thị video [mycountry.mp4](#) và thực hiện các yêu cầu sau:

1. Hiển thị thời lượng video
2. Hiển thị kích thước video, kích thước từng thành phần (độ dài và độ rộng)
3. Trích xuất 10s đầu tiên của video và lưu lại với tên `video10.mp4`
4. Xoay video10 vừa trích xuất một góc 180 độ
5. Giảm âm lượng video10 còn 0.5
6. Giảm kích thước video10 còn 0.8
7. Gán âm thanh cho video10 với âm thanh gán vào là âm thanh từ giây thứ 5 của video gốc
8. Hiển thị video sau khi chỉnh sửa
9. Lưu video sau khi chỉnh sửa với tên `videoafter.mp4`

**Bài tập 2:** Cho video [sport.mp4](#) và thực hiện yêu cầu bên dưới:

1. Đọc và hiển thị video
2. Trích xuất 10s video biết rằng thời gian trích xuất bắt đầu từ giây thứ 5
3. Thêm lề cho video vừa trích xuất với `margin` bằng 40
4. Trích xuất tốc độ fps của video
5. Hiển thị video sau khi chỉnh sửa

**Bài tập 3:** Cho video [travel.mp4](#) và thực hiện yêu cầu bên dưới:

1. Đọc và hiển thị video
2. Hiển thị kích thước video, kích thước từng thành phần (độ dài và độ rộng)
3. Thực hiện `resize` video theo tỉ lệ 7/10
4. Hiển thị kích thước video, kích thước từng thành phần (độ dài và độ rộng) sau khi chỉnh
5. Trích xuất 20s đầu tiên của video và lưu lại với tên `video3.mp4`
6. Giảm âm lượng video3 còn 0.9
7. Thêm nội dung video với dòng chữ "My video"

8. Gán âm thanh cho video3 với âm thanh gán vào là âm thanh từ giây thứ 10 của video gốc
9. Hiển thị video sau khi chỉnh sửa
10. Lưu video sau khi chỉnh sửa với tên video3after.mp4

**Bài tập 4:** Cho video [secretpet1.mp4](#) và [secretpet2.mp4](#) thực hiện yêu cầu bên dưới:

1. Đọc và hiển thị 2 video
2. Tạo bản sao cho mỗi video
3. Trích xuất 10s đầu tiên cho mỗi video
4. Ghép 2 video vừa trích xuất ở câu 3
5. Thêm nội dung video với dòng chữ "Finish"
6. Gán âm thanh cho video vừa ghép với âm thanh bất kỳ
7. Hiển thị video sau khi chỉnh sửa
8. Lưu video sau khi chỉnh sửa với tên video4after.mp4

**Bài tập 5:** Sử dụng các thao tác cơ bản thực ghép nối các video để được video finish

**Clip:** [clip0.mp4](#) , [clip1.mp4](#) , [clip2.mp4](#) , [clip3.mp4](#) , [clip4.mp4](#) , [clip5.mp4](#) , [clip6.mp4](#)

**Video finish:** [videofinishintro.mp4](#)

**Audio:** [Audio-clip1.mp4](#) , [Audio-clip2.mp4](#)

**Thực hiện chèn logo** [spkt.png](#) vào video vừa ghép với thông số thuộc tính hình ảnh logo như sau: height: 50 - margin: left = 8 - top = 8 - opacity: 0 - position: right and top.

**File hoàn chỉnh:** Lưu lại với tên videoex5.mp4

**Bài tập 6:** Thực hiện thiết lập hiển thị cho các clip1 - clip2 - clip3 - clip4 ở bài tập 5 trên cùng một khung nhìn. Biết rằng mỗi clip được thay đổi tốc độ thành 0.9 và tự động phát khi hiển thị. File xử lý hoàn tất được lưu lại với tên videoex6.mp4

Clip1: Thực hiện trích xuất 5 giây đầu tiên

Clip2: Thực hiện trích xuất từ giây thứ 10 đến 15

Clip3: Thực hiện trích xuất từ giây thứ 2 đến 6

Clip4: Thực hiện trích xuất từ giây thứ 4 đến 11

**Bài tập 7:** Thực hiện chèn chữ vào videoex6.mp4 ở bài tập 6 với nội dung **"FIT'S CUP 2024"** với âm thanh được trích xuất từ Audio-clip2. Lưu lại với tên videoex7.mp4 sau khi hoàn tất

**Lưu lại video với tên: [MSSV\\_HoTen.ipynb](#) và nộp trên E-learning**