

# Edição de vídeos com **MoviePy**

Live de Python # 212



## 1. Conceitos sobre edição

Um básico necessário

## 2. O moviepy

Arquivos, estruturas e forma de uso

## 3. Efeitos

Para áudio e vídeo

## 4. Hacks

Transcrição de áudio e geração de legendas / Cortes por silêncios



[picpay.me/dunossauro](https://picpay.me/dunossauro)



[apoia.se/livedepython](https://apoia.se/livedepython)



[pix.dunossauro@gmail.com](mailto:pix.dunossauro@gmail.com)



Ajude o projeto <3



Acássio Anjos, Ademar Peixoto, Adilson Herculano, Alexandre Harano, Alexandre Lima, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysso Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rafael, André Rocha, Aquiles Coutinho, Arnaldo Turque, Aurelio Costa, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, Carlos Alipio, Christiano Moraes, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Diogo Paschoal, Douglas Bastos, Douglas Braga, Douglas Zickuhr, Dutofanim Dutofanim, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Flávio Meira, Flavkaze Flavkaze, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Gustavo Dettenborn, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Hugo Cosme, Igor Taconi, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, João Lugão, João Paulo, João Rodrigues, Joelson Sartori, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, José Gomes, Joseíto Júnior, Jose Mazolini, Juan Gutierrez, Juliana Machado, Júlio Gazeta, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Miranda, Leonardo Mello, Leonardo Nazareth, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Carlos, Luiz Junior, Luiz Lima, Luiz Paula, Luiz Perciliano, Maicon Pantoja, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mateus Lisboa, Matheus Cortezi, Matheus Silva, Matheus Vian, Mírian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Otávio Barradas, Patricia Minamizawa, Paulo Braga, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, P Muniz, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Reinaldo Silva, Renato Veirich, Ricardo Silva, Riverfount Riverfount, Robson Maciel, Rodrigo Alves, Rodrigo Freire, Rodrigo Vaccari, Rodrigo Vieira, Rogério Sousa, Ronaldo Silva, Ronaldo Silveira, Rui Jr, Samanta Cicilia, Sara Selis, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago Oliveira, Thiago S, Thiago Souza, Tiago Minuzzi, Tony Dias, Valcilon Silva, Valdir Tegon, Victor Wildner, Vinícius Bastos, Vítor Gomes, Vitor Luz, Vladimir Lemos, Walter Reis, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Wilson Rocha, Xico Silvério, Yury Barros



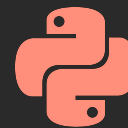
Obrigado você



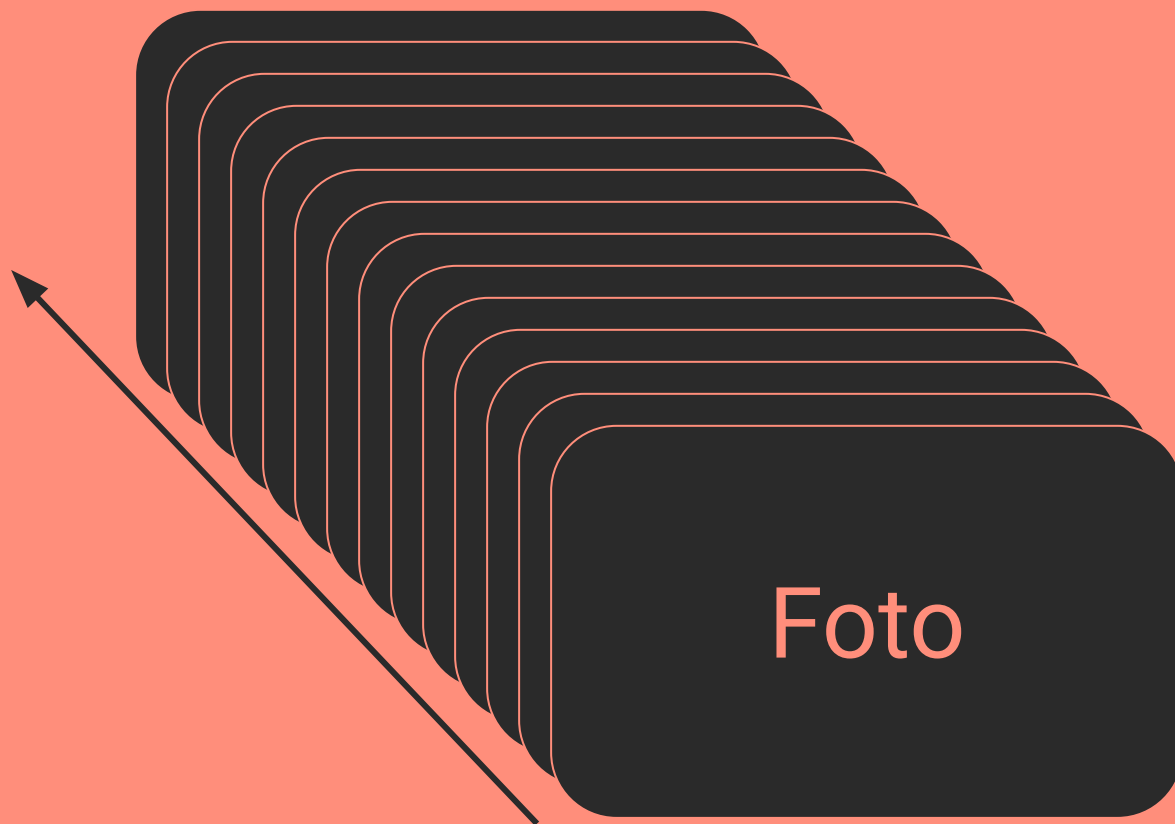
O básico necessário

Con  
ceitos

# Vídeo



Afinal, que raio é um vídeo

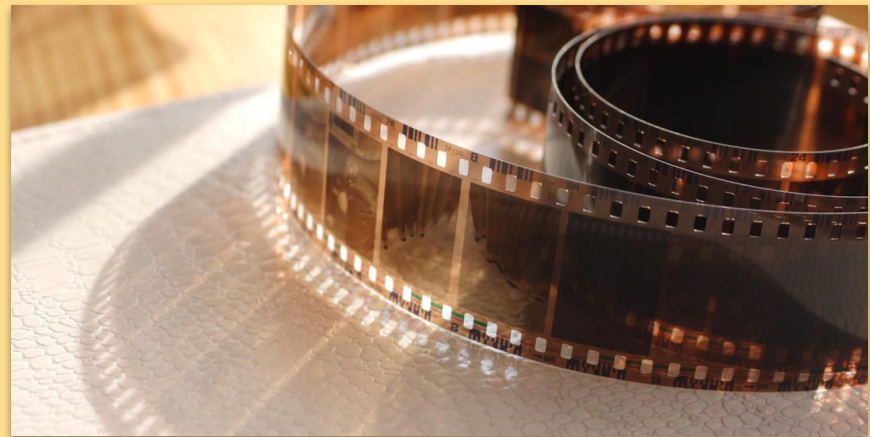
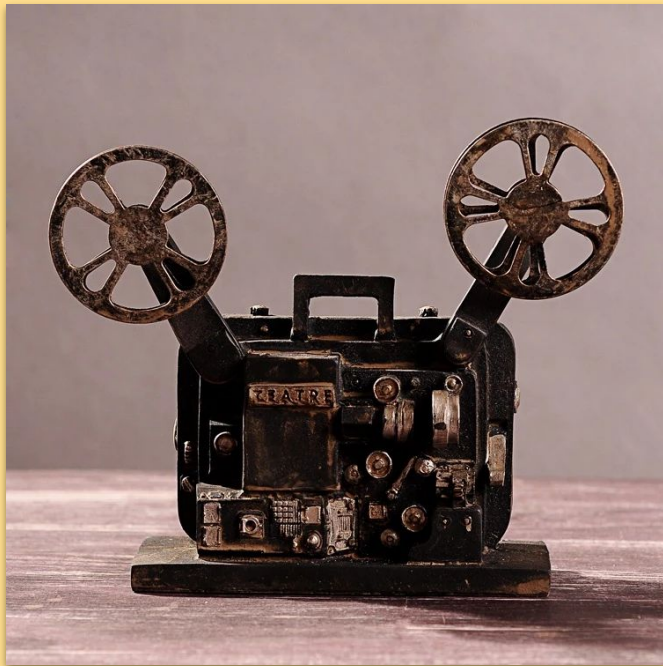


# Rolo de filme



<https://ijnet.org/pt-br/resource/9-dicas-para-ter-em-mente-ao-fazer-um-document%C3%A1rio>

# Rolo de filme



<https://ijnet.org/pt-br/resource/9-dicas-para-ter-em-mente-ao-fazer-um-document%C3%A1rio>  
<https://pt.aliexpress.com/item/4001310530525.html>



# Vídeo digital



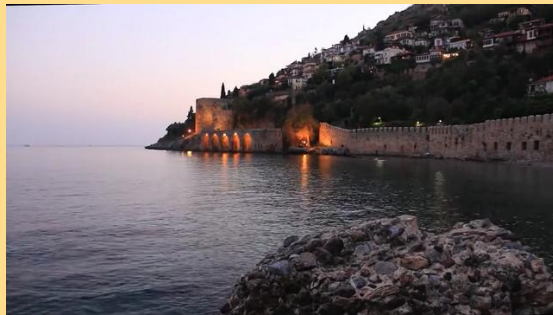
Um vídeo digital seque o mesmo conceito. Porém, ele pode ser **não linear**!



# Vídeo digital



Um vídeo digital seque o mesmo conceito. Porém, ele pode ser **não linear**!



Objetos de composição

# Movie py

Arquivos, estrutura  
e formas de uso

# 0 moviepy




Moviepy, é uma biblioteca para edição de vídeos, ela pode fazer:

- Operações básicas
  - Cortes
  - Concatenação
  - Inserção se textos
  - ...
- Composição de vídeo
- Processamento
- Efeitos
- Renderização



Teve seu início em 2014, criado por @Zulko sobre licença MIT.

11 Dec 2014

 Zulko

v0.2.2

3fef00d

Comparar

## Início do filmePy

Este é o primeiro lançamento de uma trilogia que incluirá "The Dark MoviePy" e "The Dark MoviePy Rises". Neste ponto, não tenho certeza para que servem os lançamentos.

Então, digamos que este lançamento celebra que o MoviePy é bastante estável em diferentes plataformas e Python 2/3, ou as pessoas pararam de relatar bugs.

Principais ideias para o próximo lançamento:

- Simplifique a instalação de dependências binárias do MoviePy. Isso pode ser feito usando o ImageIO para baixar o ffmpeg.
- Use back-ends melhores para criação de GIF, o ImageMagick é muito lento. Isso também pode ser feito com ImageIO.
- Faça um gerenciador de arquivos que mantenha fechadas as conexões com os arquivos que não são usados, pois as conexões de arquivos consomem muita memória RAM. Isso permitiria ter centenas de VideoClips ImageClips e AudioClips abertos, mas apenas alguns realmente fazendo uma conexão.

Vai ser divertido !

► Ativos 2

```
pip install moviepy
```



Instalação



```
pip install moviepy  
pip install pygame
```



Instalação



# Trabalhando com arquivos



O moviepy pode lidar com três tipos de arquivos:

- Vídeos
- Áudios
- Imagens

```
1  from moviepy.editor import AudioFileClip, ImageClip, VideoFileClip
2
3  video = VideoFileClip('assets/video_0.mp4')
4
5  audio = AudioFileClip('assets/musica_de_fundo.mp3')
6
7  imagem = ImageClip('image.png', duration=5)
```



# Atributos principais



O que podemos querer do vídeo:

```
1  video.size          # Dimensões do Clip
2  video.duration      # Duração do clip em segundos
3  video.fps           # Frames por segundo
4  # Um iterador frame a frame (matriz de imagem)
5  video.iter_frames
```

# Cortes e controle de tempo



```
1 video = (  
2     VideoFileClip('assets/video_0.mp4')  
3     .subclip(0, 5)  
4 )  
5  
6 audio = (  
7     AudioFileClip('assets/musica_de_fundo.mp3')  
8     .subclip(100, 105)  
9 )
```

# Compondo um vídeo



Para compor, precisamos de um objeto específico. Um objeto de composição.

```
12  composição = CompositeVideoClip(  
13      # Lista de coisas que serão compostas  
14      [video, video2]  
15  )  
16  
17  composição.audio = audio  
18  composição.write_videofile('teste.mp4')
```

# Elementos de composição

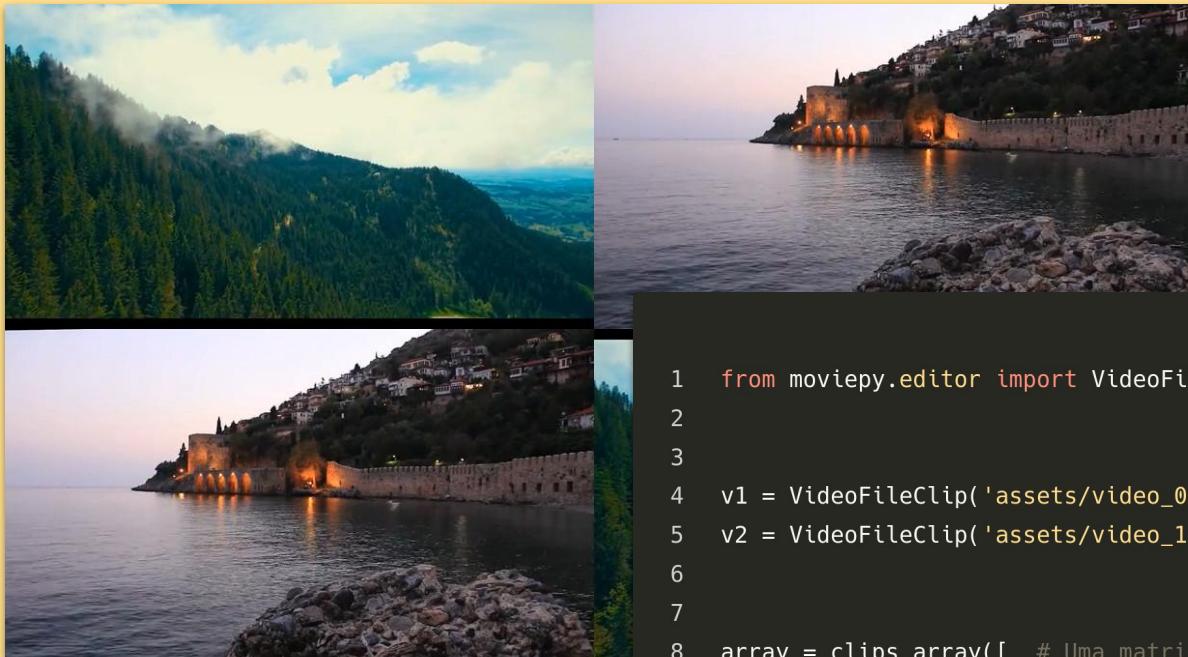


```
1 texto = TextClip(  
2     'Live de Python',  
3     color='white',  
4     fontsize=50  
5 ).set_duration(5)  
6  
7 cor = ColorClip(  
8     size=video.size,  
9     color=(255, 255, 0),  
10    duration=5  
11 )
```

# Composição lado a lado



# Composição lado a lado



```
1 from moviepy.editor import VideoFileClip, clips_array
2
3
4 v1 = VideoFileClip('assets/video_0.mp4').subclip(0, 5)
5 v2 = VideoFileClip('assets/video_1.mp4').subclip(0, 5)
6
7
8 array = clips_array([ # Uma matriz de vídeos
9     [v1, v2], # linha 1
10    [v2, v1], # linha 2
11 ])
12
13 array.preview()
```



# Juntando tudo (Linearmente)



```
1  from moviepy.editor import VideoFileClip, concatenate_videoclips
2
3  v1 = VideoFileClip('assets/video_0.mp4').subclip(0, 5)
4  v2 = VideoFileClip('assets/video_1.mp4').subclip(0, 5)
5
6  concatenados = concatenate_videoclips([v1, v2])
7  # V1 seguido de V2, produzindo um vídeo de 10 segundos
8  concatenados.preview()
```

## Possível bug



Caso os vídeos não sejam da mesma fonte, não tenham o mesmo fps ou não tenham a mesma dimensão pode originar em bugs. Para isso ser resolvido. Deve-se trocar o método de composição

```
1  concatenados = concatenate_videoclips(  
2      [v1, v2],  
3      method='compose',  
4  )
```



# Efeitos

Para áudio e vídeo

# Efeitos



Efeitos são caracterizados como qualquer mudança que pode ocorrer no vídeo ou no áudio. Mas eles podem ser coisas simples, como:

- Rotacionar
- Escalonar
- Colorir
- Controle de volume (normalização)

# Aplicação de efeitos



Alguns efeitos já estão implementados nos objetos de áudio e vídeo:

```
1  from moviepy.editor import VideoFileClip
2
3  video = (
4      VideoFileClip('assets/video_0.mp4')
5      .rotate(180) # rotaciona
6      .fadein(2)   # aparecimento
7      .fadeout(2)  # Finalização
8      .speedx(3)   # acelera e desacelera
9      .resize(2)   # Redimensiona
10 )
```



## .fx() (abrir a doc! não esquecer)

Porém uma outra gama de efeitos pode ser aplicada usando o método **.fx()**

```
1  from moviepy.editor import VideoFileClip
2  from moviepy.video.fx import all as vfx
3
4  video = (
5      VideoFileClip('assets/video_0.mp4')
6      .fx(vfx.accel_decel, 2) # Acelera o vídeo
7      .fx(vfx.blackwhite)
8      .fx(vfx.colorx, 1.2)    # Multiplica as cores por um fator (.4)
9      .fx(vfx.invert_colors)  # Inverte as cores
10     .fx(vfx.loop, 3)        # Clipe em loop por X vezes
11     .fx(vfx.make_loopable, 3) # Faz o fim misturar com o começo
12     .fx(vfx.margin, 10)     # Margem em torno do vídeo
13 )
```

<https://zulko.github.io/moviepy/ref/videofx.html>

# Efeitos de áudio



```
1  from moviepy.editor import AudioFileClip
2  from moviepy.audio.fx import all as afx
3
4  audio = (
5      AudioFileClip('assets/musica_de_fundo.mp3')
6      .subclip(60, 65)
7      .fx(afx.audio_fadein, 5)
8      .fx(afx.audio_fadeout, 5)
9      .fx(afx.volumex, 3)
10     .fx(afx.audio_normalize)
11     .fx(afx.audio_loop, 3)
12 )
```

# Efeitos de posição



```
1 from moviepy.editor import (  
2     ColorClip, VideoFileClip, CompositeVideoClip  
3 )  
4  
5 def pos(tempo):  
6     return (tempo*30, 10)  
7  
8 video = VideoFileClip('assets/video_0.mp4')  
9  
10 color = ColorClip(  
11     (200, 200),  
12     color=(0, 0, 0),  
13     duration=video.duration  
14 ).set_position(pos)  
15  
16  
17 comp = CompositeVideoClip([video, color])
```

# Hacks

Dicas que podem  
virar outras lives

# Moviepy + Pydub



Pydub é uma biblioteca para tratamento de áudio. Nela existe uma função para encontrar os silêncios de um áudio. Podemos:

- Oferecer melhor suporte a tratamento de audio
- Montar cortes automatizados
- Exemplo 98

**Gostariam de uma live sobre pydub?**



# Moviepy + Vosk



Vosk é uma biblioteca de reconhecimento de fala. Juntado as duas bibliotecas podemos:

- Gerar legendas automáticas
- Fazer cortes por voz
- Fazer um VideoGrep
  - <http://antiboredom.github.io/videogrep/>
- Exemplo 99

**Vocês gostariam de uma live de Vosk?**

# Coisas que não falamos, mas são possíveis



- Uso de máscaras
- Plots de gráficos em vídeo com matplotlib
- Motion design e vetores com Gizeh
- Fazer Gifs



[picpay.me/dunossauro](https://picpay.me/dunossauro)



[apoia.se/livedepython](https://apoia.se/livedepython)



[pix.dunossauro@gmail.com](mailto:pix.dunossauro@gmail.com)



Ajude o projeto <3

