

低スペック環境でも使える  
エンコーダー

**FFmpeg**

# 自己紹介 88IO

- 普通科3年（4月から名工生）
- 高校1年のときからPythonを学習
- ThinkPad信者



@88IOBF



# 自分の元環境

## IBM X40/X41 (2005年発売)

CPU -> Pentium M 1.5GHz  
RAM -> DDR2 2.0GB  
SSD -> 1.8inch IDE 32GB  
OS -> Ubuntu14.04 LTS



テキストエディタ : Vim8.0  
動画再生 : VLC  
画像編集 : Gimp  
音声編集 : Audacity  
動画編集 : FFmpeg



# FFmpeg とは

- 動画と音声を記録・変換・再生するためのフリーソフトウェア（ビデオコンバーター）
- クロスプラットフォーム
- 様々なコーデックに対応しており、多くのソフトウェアの内部で使われている

# FFmpeg とは

## メリット

- 軽量、高速である(並列処理に対応)
- 拡張子変更は大抵簡単にできる
- 簡易的な動画編集機能を備えている

## デメリット

- CUIツールなので、コマンドラインでの操作が必須
- コマンドがやたら長くなる
- 学習コストが意外と高い

# FFmpeg のインストール

## Windows

```
choco install ffmpeg (Chocolateyを使用)
```

## Mac

```
brew install ffmpeg (Homebrewを使用)
```

## Ubuntu

```
sudo apt install ffmpeg
```

# FFmpeg の使い方

## 基本コマンド

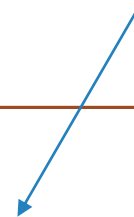
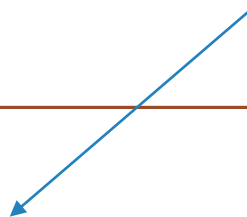
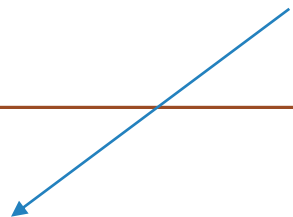
ffmpeg ` 入力オプション ` -i 入力ファイル ` 出力オプション ` 出力ファイル

初期設定

入力

加工

出力



# FFmpeg の使い方

ファイル形式変更 (avi -> mp4)

```
ffmpeg -i input.avi output.mp4
```

フレームレート変更 (30fps -> 60fps)

```
ffmpeg -r 30 -i input.mp4 -r 60 output.mp4
```

リサイズ ( -> 1280x720)

```
ffmpeg -i input.avi -vf "scale=1280:720" output.mp4
```



# FFmpeg の使い方

カット (10.0s ~ 30.0s)

```
ffmpeg -i input.mp4 -ss 10.0 -t 20.0 output.mp4
```

クロップ ((20, 30)から640x480)

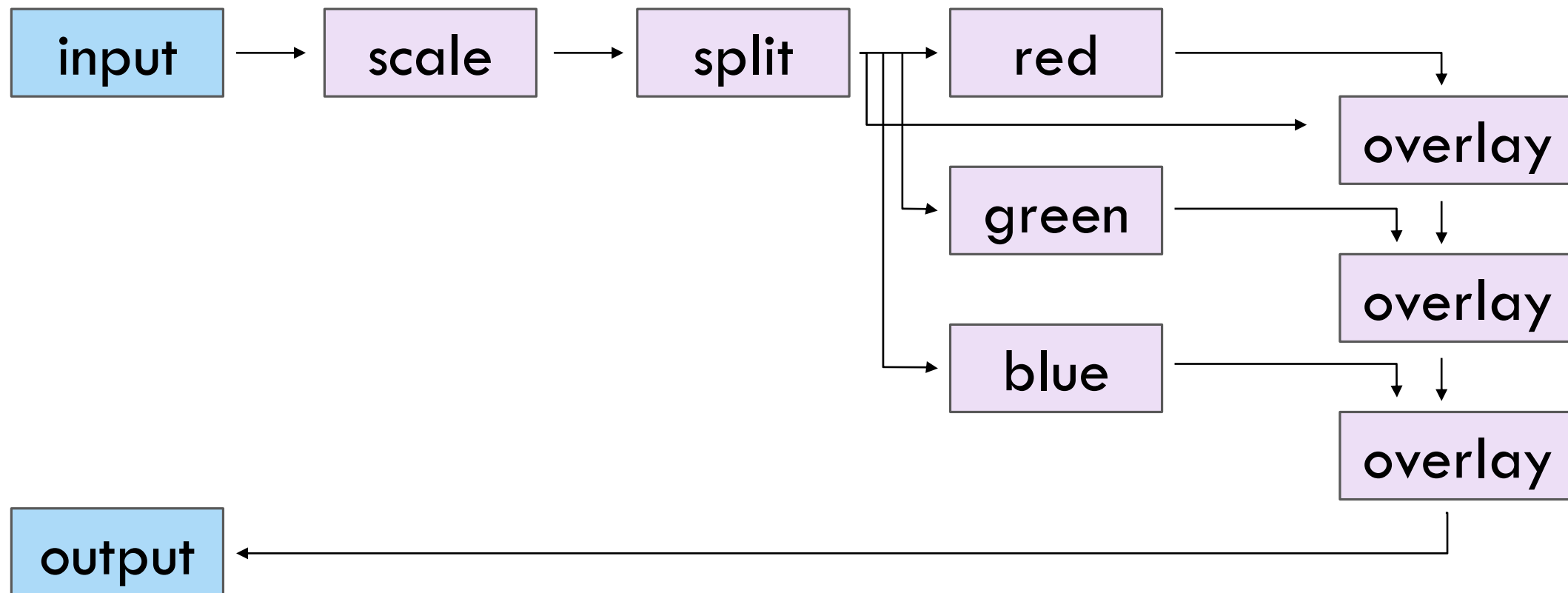
```
ffmpeg -i input.mp4 -vf "crop=640:480:20:30" output.mp4
```

静止画を出力 ( 0001.jpg~)

```
ffmpeg -i input.mp4 -f image2 %04d.jpg
```

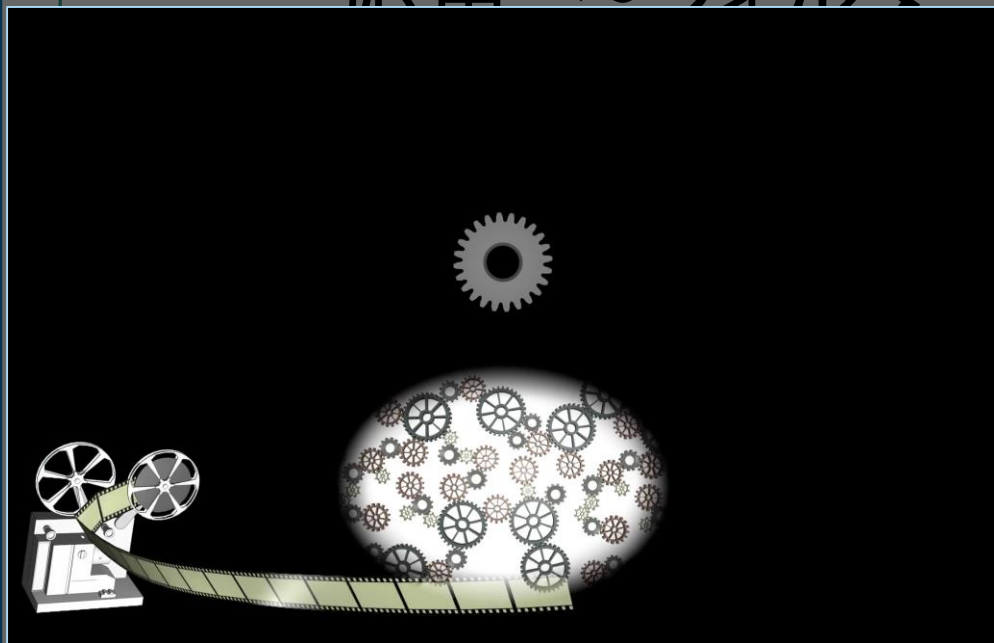
# FFmpeg の使い方

## 応用 ~フィルター~の概念~

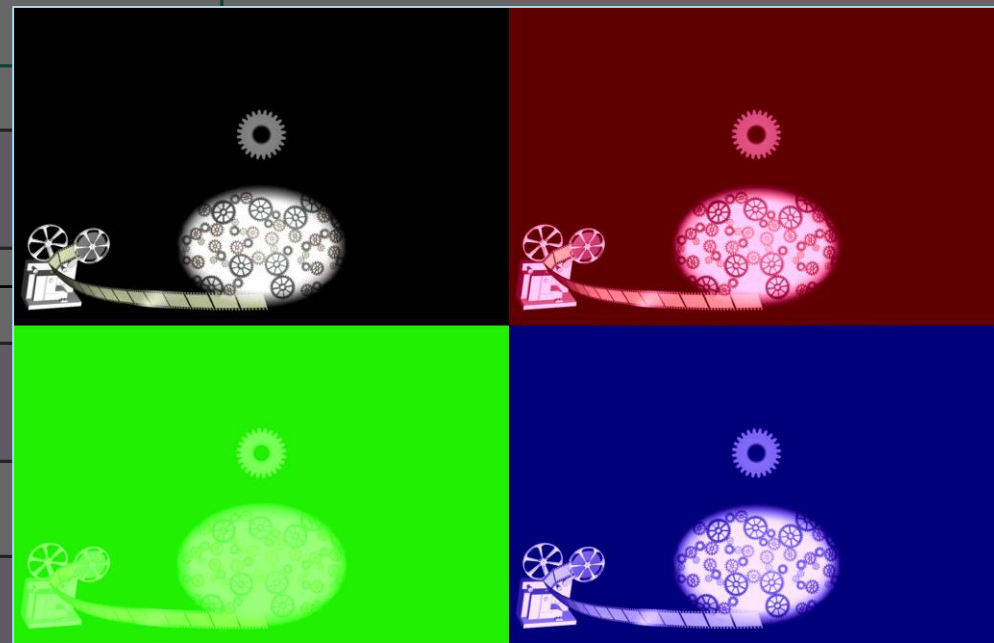
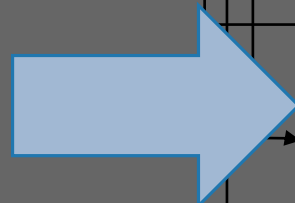


# FFmpeg の使い方

応用 ～フィルター～の概念～



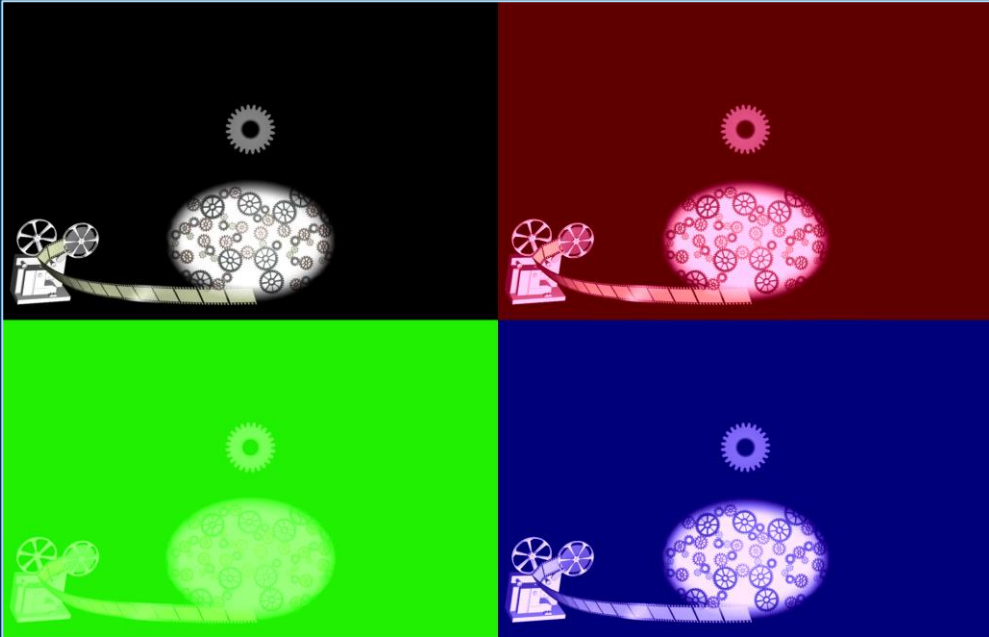
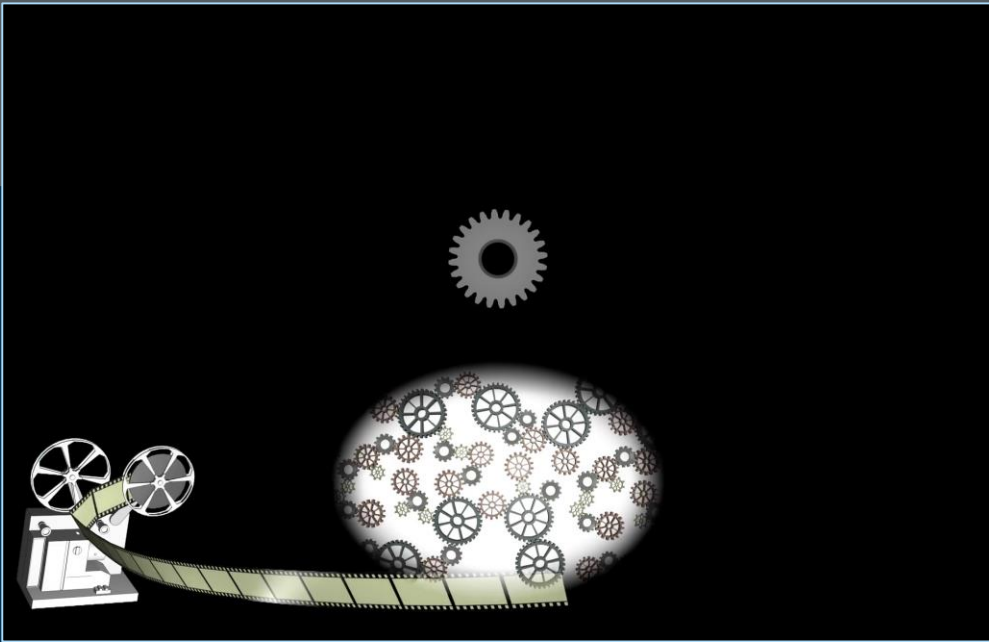
split



overlay

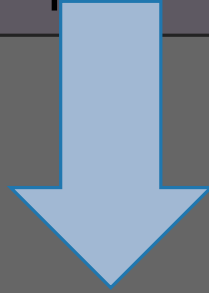
output





の概念

split



```
ffmpeg -i input.mp4
-filter_complex "
[0]scale=iw/2:-1,
split=4[red][green][blue][origin];
[red]eq=gamma_r=5.0[red];
[green]eq=gamma_g=5.0[green];
[blue]eq=gamma_b=5.0[blue];
[0][origin]overlay=0[x];
[x][red]overlay=w[y];
[y][green]overlay=0:h[z];
[z][blue]overlay=w:h[out]"
-map "[out]" output.mp4
```

# FFmpeg 現在の利用方法

- 動画の圧縮、リサイズ、クリッピング
- 字幕付与、合成等の簡易編集
- コーデック変更 (H.265非対応の場合等)
- AviUtlとの併用 (avi出力->VLC->FFmpeg->)

# 展望

**まだ何かできるはず！**

**みんなに使ってもらいたい！**

**でも、コマンド入力の敷居が高い！**

# 展望

**そうだ、ノードエディタ化しよう！**

# 展望

大学入学後にBlenderのノードエディタライクなGUIアプリを制作し、FFmpegの利用難易度を下げる！

