ASCII Art Animation

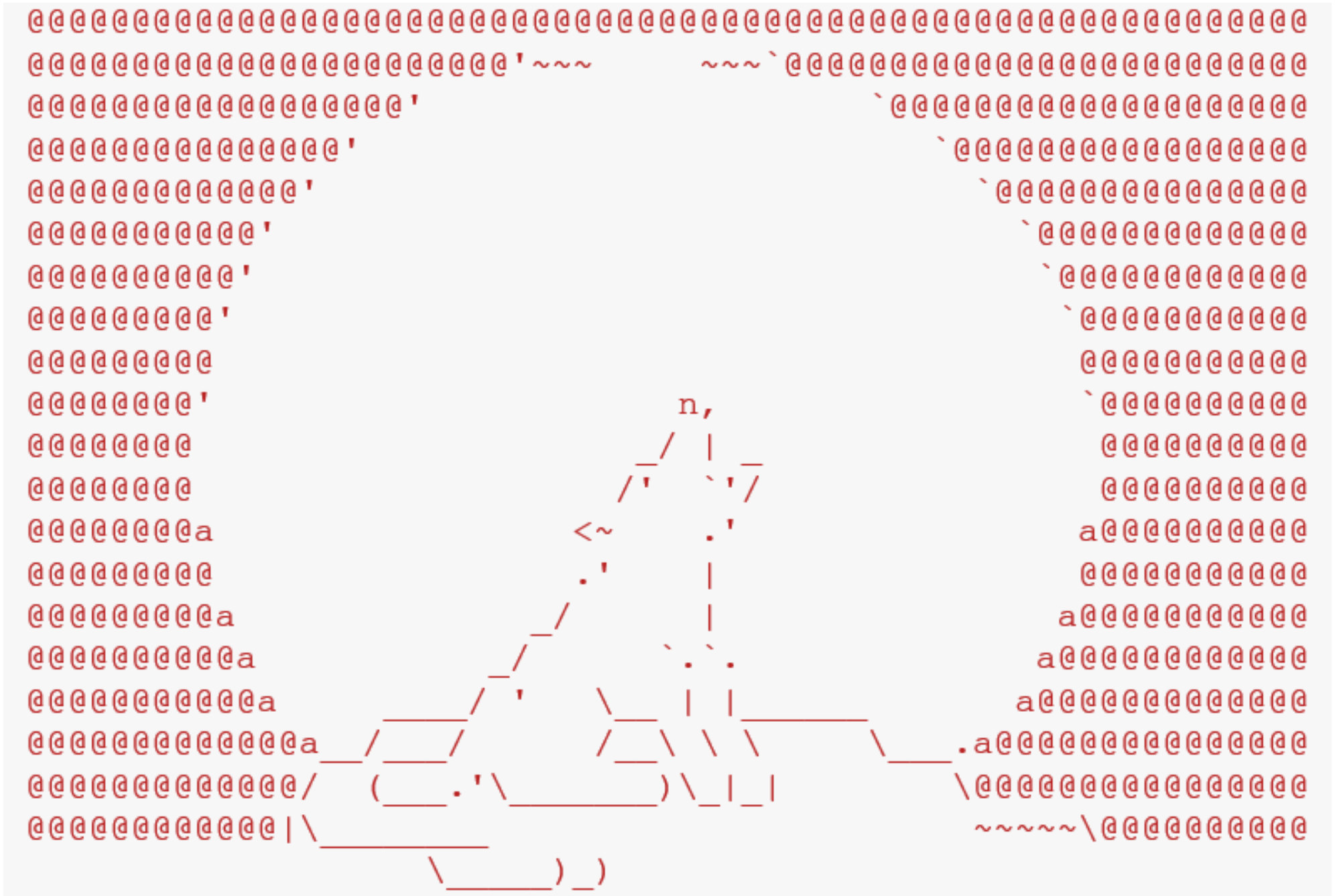
Team:#awfullyPYTHONIC Members:**金淑媛、王倩淑、杨思佳、尹悦舟**

导师：鲍杨

2018-6-15

01. Introduction

Why did we choose this topic and
how did we make decisions



ASCII ART ?

Our Allocation

- 金淑媛 and 尹悦舟 mainly focused on making PPT.
- 王倩淑 mainly focused on writing and revising the code.
- 杨思佳 mainly focused on writing and revising the report.

And we present together! ☺

General Steps

FIRST

Get some modules.

SECOND

Decide the main route to take.

THIRD

Solve some problems.

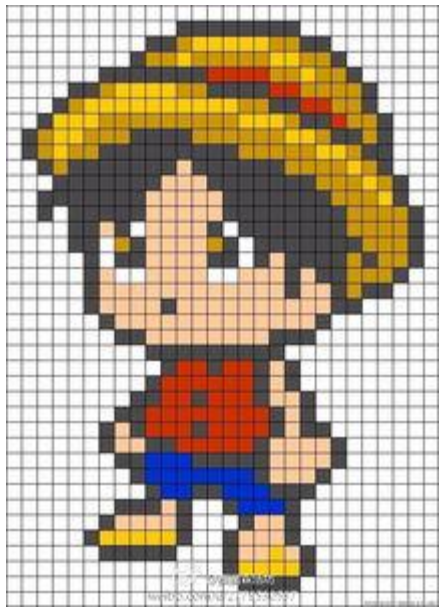
02. Thinking

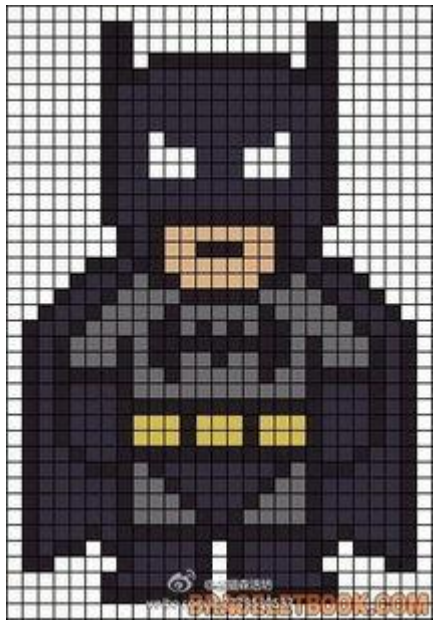
Convert pictures into ASCII character pictures

Pictures are made of pixels.

Every pixel has its gray value(light and shade degree).

The core of the algorithm is to match gray value to the corresponding character.





Convert pictures into ASCII character pictures

Gray value is vary from 0(black) to 255(white).
Color picture is made of three colors: Red, Green, Blue(RGB).
We can use the following formula to convert RGB values to the corresponding gray value:

$$Gray = 0.2126 * r + 0.7152 * g + 0.0722 * b$$

Or we can use the existing function in cv2 cv2.cvtColor() **to do this.**

The order of characters corresponding to gray value from large to small is:

$@B%8&WM#*oahkbdpqwmZO0QLCJUYXzcvunxrjft/\|()1{}[]?-_+~<>i!ll;:,\"^`'.

Convert pictures into ASCII character pictures

On the basis of the above contents, we use the following ideas for conversion:

1. correspond each character to the gray value of a section.

2. divide the picture into appropriate pixel blocks, calculate the gray value of each block line by line, and correspond to the characters.

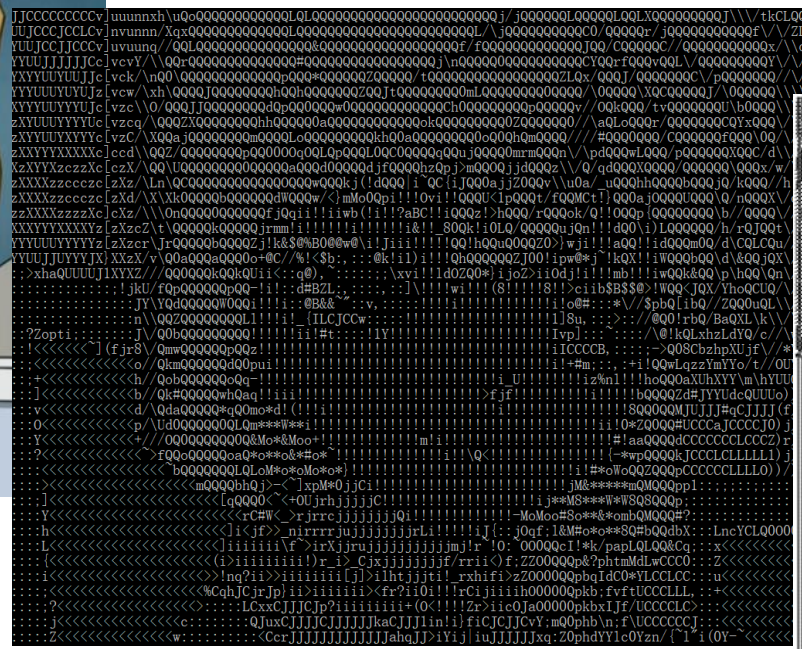
3. write characters line by line to a .txt file and generate pictures.

This is the original version.

Convert pictures into ASCII character pictures

The following is the result:

Result during the test and final result



Convert video into ASCII character video

Video is made of many frame images.

Read every frame in the video in a loop and then convert it into a character picture.

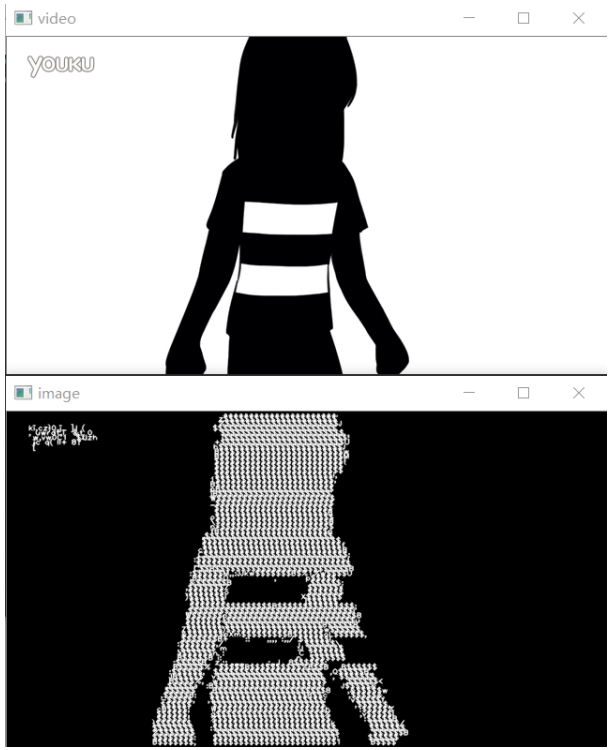
To make the video more complete, we add music to it afterwards.

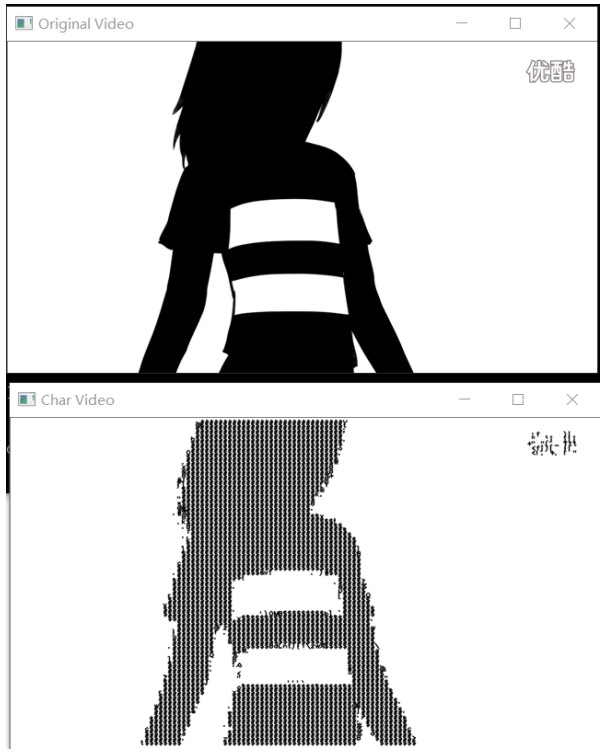
Click here to watch the character video

Problems we met and solved

There is a distortion of the edge on the right side.

Read line by line VS Read one by one





Brainstorm to add functions and refine codes

How about real-time transferring people?

Brainstorm to add functions and refine codes

Refine our codes

Combine

Add other functions

03. Modules

What modules we used in our code
and what are their functions

Modules

01 OpenCV

OpenCV was built to provide a common infrastructure for computer vision applications.

02 numpy

The numpy system has a powerful N dimensional array object. This tool can be used to store and process large matrices.

03 os

The os module provides a portable way of using operating system dependent functionality.

04 argparse

The argparse module makes it easy to write user-friendly command line

05 subprocess

The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

openCV

1. **cv2.imread（‘name’）------for pictures**
   cv2.videocapture(‘name’)-----for videos
2. **cv2.resize(img,(wide,height))**
3. **Gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRY).**

   # get the grayscale

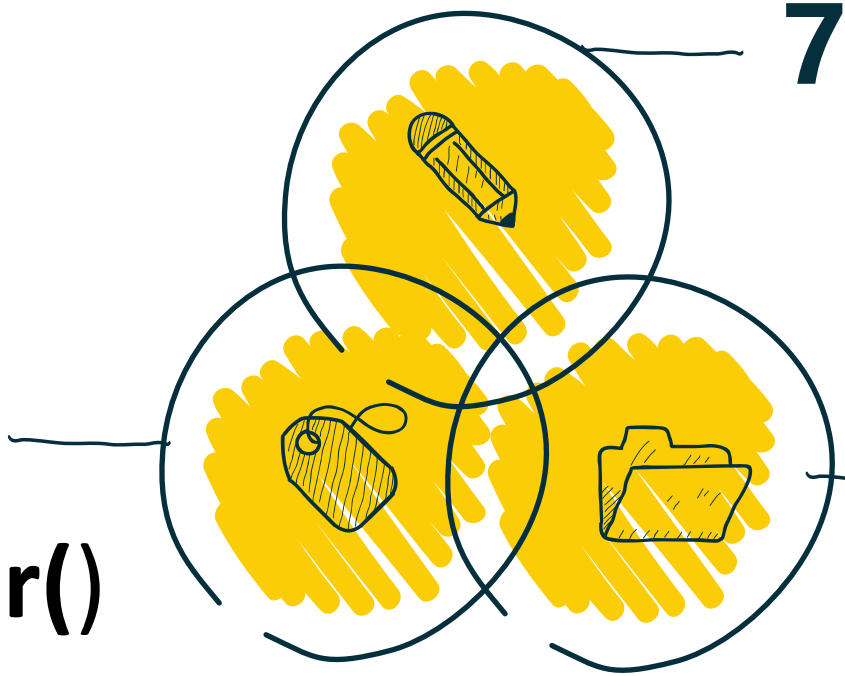
OpenCV

4. **cv2.putText**(img,’ words’ , (50,150),cv2.FONT_HERSHEY_COMPLEX, 6,(0,0,255),25)

5. **cv2.waitkey**(delay)
   # wait for key

6. **cv2.namedWindow**(‘Char Image’)
   #create a window

OpenCV



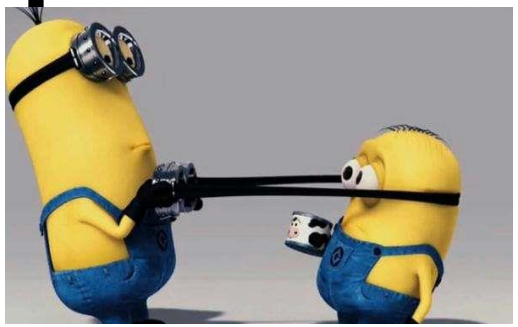
7. cv2.imshow(winame,mat)

8. cv2.imwrite
cv2.VideoWriter()

9. cv2.destroyAll Windows()

numpy

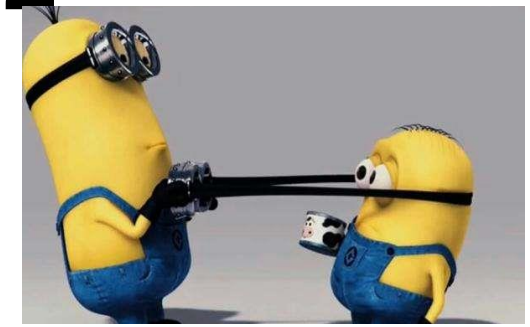
01



Numpy.zeros(6)

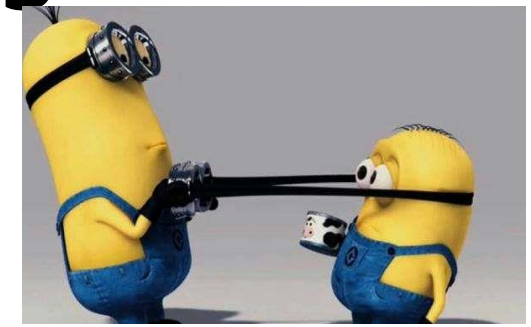
Numpy.zeros((2,3))

02



Numpy.ones((2,3))

03



Numpy.empty()

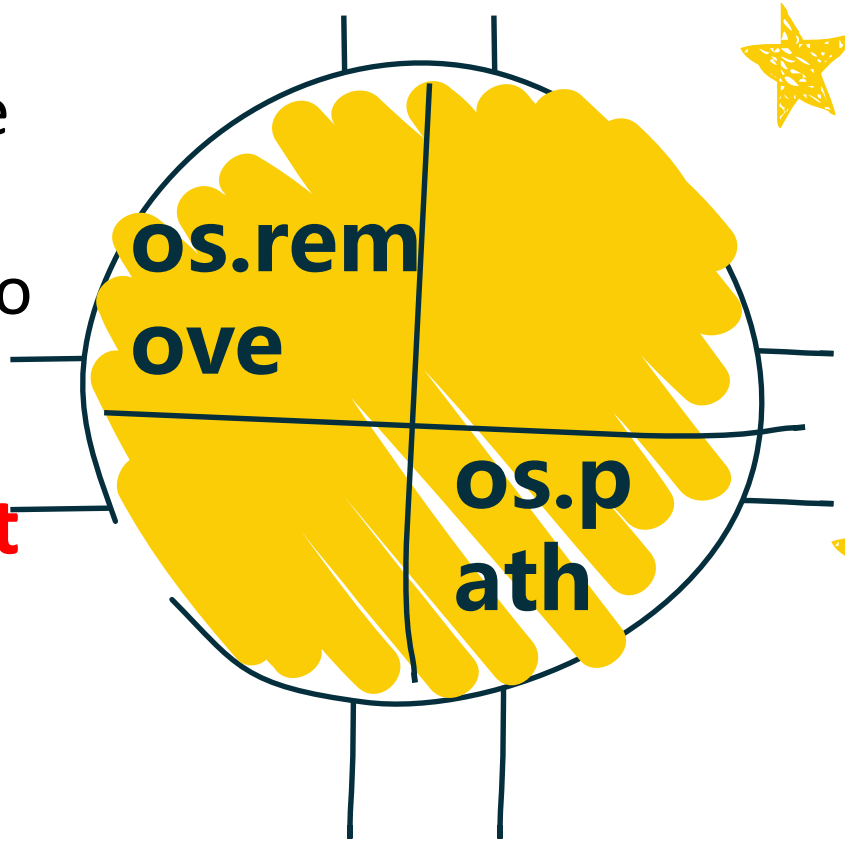
Several types of the data

create an empty canvas

```
canvas = np.zeros(sp, np.uint8)
```

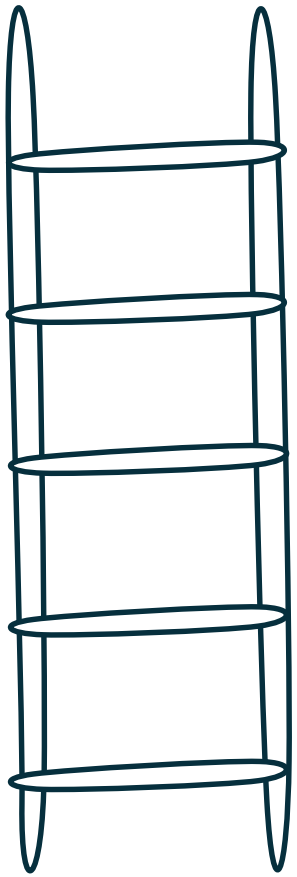
名称	描述
bool	用一个字节存储的布尔类型（True或False）
inti	由所在平台决定其大小的整数（一般为int32或int64）
int8	一个字节大小，-128 至 127
int16	整数，-32768 至 32767
int32	整数，-2 ** 31 至 2 ** 32 -1
int64	整数，-2 ** 63 至 2 ** 63 - 1
uint8	无符号整数，0 至 255
uint16	无符号整数，0 至 65535
uint32	无符号整数，0 至 2 ** 32 - 1
uint64	无符号整数，0 至 2 ** 64 - 1
float16	半精度浮点数：16位，正负号1位，指数5位，精度10位
float32	单精度浮点数：32位，正负号1位，指数8位，精度23位
float64或float	双精度浮点数：64位，正负号1位，指数11位，精度52位
complex64	复数，分别用两个32位浮点数表示实部和虚部
complex128或complex	复数，分别用两个64位浮点数表示实部和虚部

OS



- On Windows, attempting to remove a file that is in use causes an exception to be raised
- **os.remove(file.split(‘.’)[0] + ‘.mp3’)**
  os.remove(name +’ .avi’)
- We can use this function to split the pathname path into a pair (root, extension
- **os.path.splitext (file)[1]**

argparse



Argparse Is a comprehensive parameter processing libra

The first step in using argparse is to create a parser object and tell it what parameters it will have.

```
parser. add_argument()    #Add parameters
```

```
parser.add_argument('-r', '--ratio', type=int, nargs='?', default=10)
```

In this line, we add the parameter of compression ratio.

[subprocess]

```
158  # get audio file
159  def video2mp3(file):
160      out_file = file.split('.')[0] + '.mp3'
161      subprocess.call('ffmpeg -i ' + file + ' -f mp3 ' + out_file, shell=True)
```

subprocess.call()

Create a subprocess that runs at the same time with the main process

04. Conclusion

Let us present the result to you!

Thanks for your listening☺