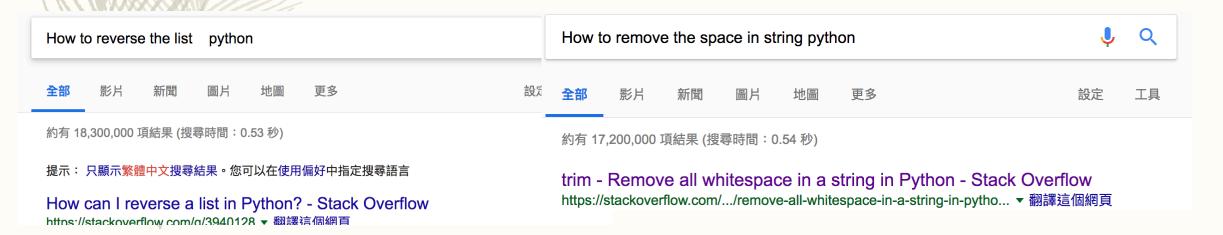


# Python3

- 建議 3.5 +
- Array 處理
  - 安装numpy → pip3 install numpy
- 影像處理
  - 1. 安裝opencv → pip3 install opencv-python
  - 2. 安裝Pillow → pip3 install Pillow
  - ...

# Python語法

- Suppose 大家都會了
- https://github.com/mediaic/Python\_OpenCV\_Lab/blob/master/python\_review.ipynb
- You can google it for any question
- 練習思考搜尋Google的關鍵字



# Numpy

 Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

numpy

Your image is a numpy array!

```
import numpy as np
import cv2
from PIL import Image

img = cv2.imread('Lenna.jpg')

img = Image.open('Lenna.jpg')
img = np.array(img)
```

numpy

– python& numpy tutorial :

http://cs231n.github.io/python-numpy-tutorial/



# Use Python to do CV OpenCV & PIL

 Opencv: opencv for C++ was popular and useful, the opencvpython is also useful for computer vision application.

http://opencv-pythontutroals.readthedocs.io/en/latest/index.html

– PIL Image package: Open source Python Image Library, and it is more and more popular because of Pytorch?

https://pillow.readthedocs.io/en/5.2.x/

### How to Choose them?

- Opency can almost do all CV applications.
- When doing Deep Learning, image augmentation is popular. And Pytorch (toolkit) use PIL package to do image preprocessing. (pytorch幫你包好)
- So you can use torchvision.transforms.Resize(xxx)(img)



Enlarge your Dataset

# Opency, Sklearn, Numpy Doc.

Opencv python API: <a href="https://docs.opencv.org/3.0-beta/modules/refman.html">https://docs.opencv.org/3.0-beta/modules/refman.html</a>

- sklearn : http://scikit-learn.org/stable/

- numpy: directly google: eg. numpy min

## Read, Show, Save Image

```
import cv2
   import numpy as np
   ## Load image
   img = cv2.imread('Lenna.jpg') ## BGR
 6 ## Grayscale
   img = cv2.imread('Lenna.jpg',0)
   ## Show image
   cv2.imshow('image',img)
   cv2.waitKey(0)
   cv2.destroyAllWindows()
13
   ## Save image
   cv2.imwrite('test.jpg',img)
```

```
## Show image
import matplotlib.pyplot as plt
plt.imshow(img[:,:,::-1]) ## To RGB
plt.show()

plt.imshow(img,cmap='gray')
plt.show()
```



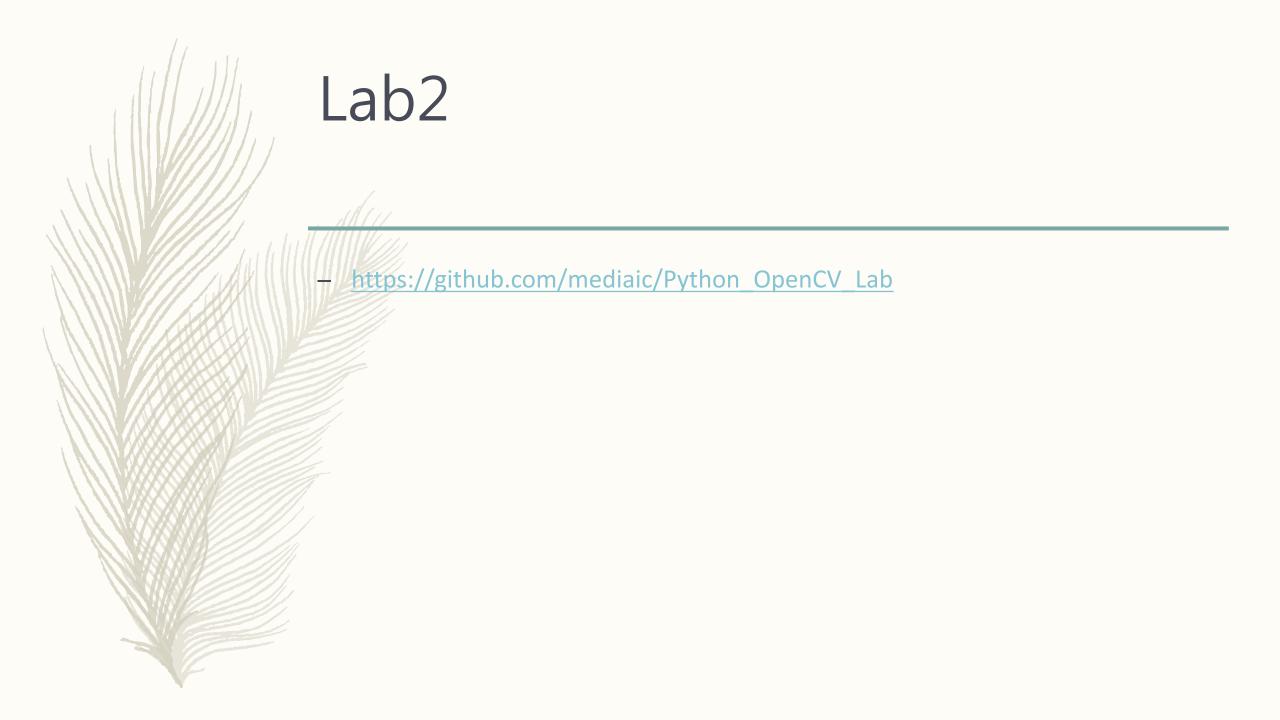
# Lab1 Related Opencv

```
## Resize
cv2.resize()
## Color space
cv2.cvtColor()
## smoothing
cv2.blur(),cv2.GaussianBlur(),cv2.mediaBlur()
## PCA Compute
mean,eigenvectors = cv2.PCACompute(matrix,mean=None)
```

```
Lab1 Related numpy
       ## min, max
       np.min(), np.max()
       ## dot
       np.dot()
       ## flatten
       np.flatten()
```

### Lab1 Related sklearn

```
## KNN
from sklearn.neighbors import KNeighborsClassifier
#宣告
KNN = KNeighborsClassifier(args)
# Feed Training Data
KNN.fit(X_train,Y_train)
# Predict Class
KNN.predict(X_test)
# 直接告訴你acc
KNN.score(X_test,Y_test)
```



```
Lab2 Related Opencv
        ## Opencv
        # ColorMap
        cv2.applyColorMap()
        # Padding
        cv2.copyMakeBorder() #use cv2.BORDER_REFLECT
        # SURF
        cv2.xfeatures2d.SURF_create()
        cv2.drawKeyPoints()
```

# Lab2 Related Numpy, Scipy

```
## Numpy scipy
# Save & load
np.save(),np.load()
# concatenate
np.concatenate()
# mean
np.mean()
# reciprocal
np.reciprocal()
# load .mat file
import scipy.io as sio
sio.loadmat()
# Euclidean distance
from scipy.spatial import distance
distance.euclidean()
```

```
Lab2 Related Sklearn
          ## Sklearn
          # Kmeans
          from sklearn.cluster import KMeans
          Kmeans = KMeans(args)
          Kmeans.fit_predict(X)
```

# Lab2 Related Matplotlib

```
plt.figure()
plt.bar(x,y)
plt.title()
plt.xlabel()
plt.ylabel()
plt.savefig()
```

```
import matplotlib.pyplot as plt
```