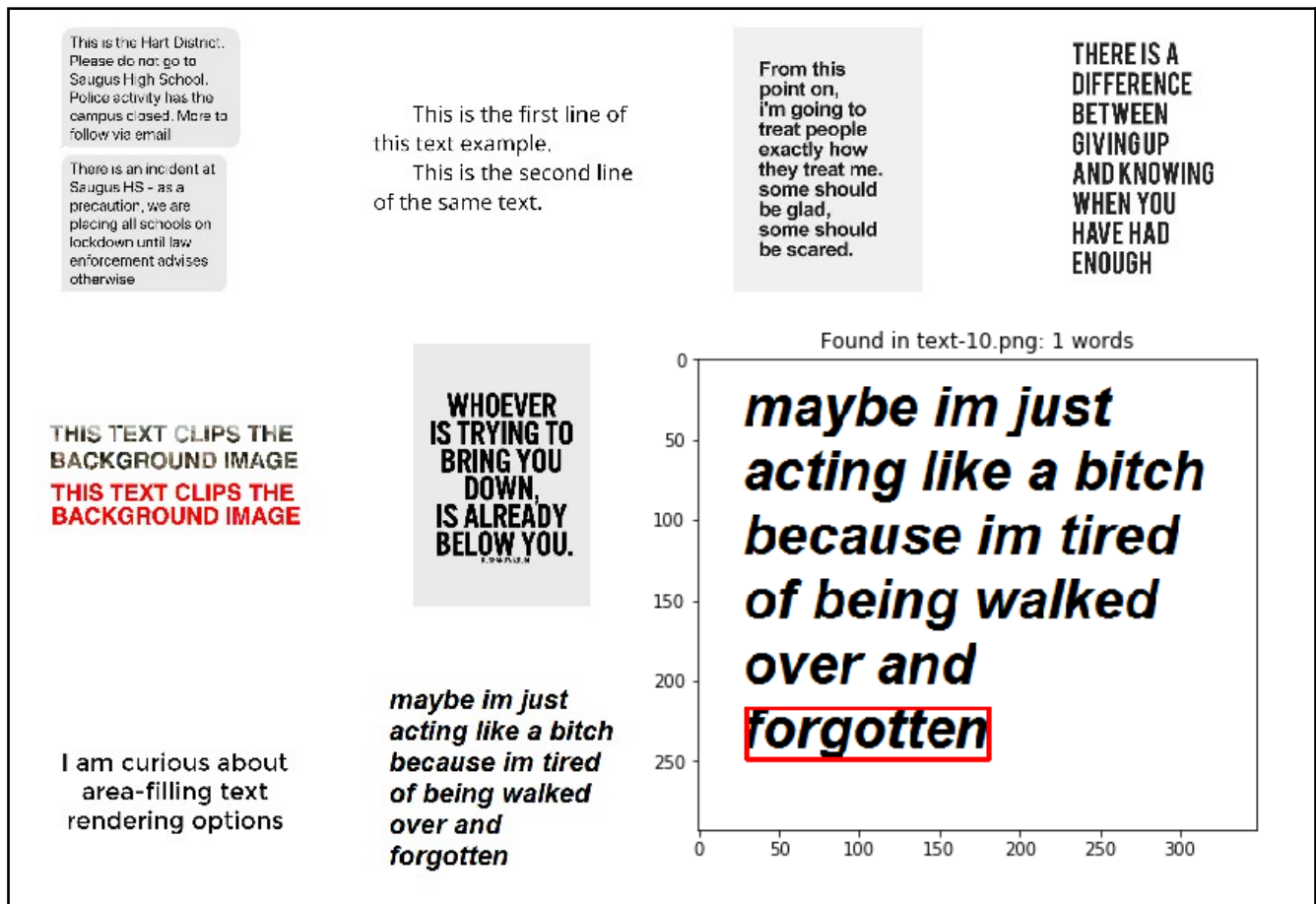


การเขียนโปรแกรมภาษาไพธอน และการประยุกต์ใช้กับงานทางด้านการประมวลผลภาพ



โดย
ผศ.ดร.โอฬาริก สุรินตะ
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

Table of Contents

ตัวแปรประเภท List.....	4
กำหนดตัวแปรประเภท List.....	4
ลบตัวแปรประเภท List.....	4
การคัดลอกตัวแปรประเภท List.....	4
ตรวจสอบประเภทของตัวแปร และจำนวนสมาชิกใน List.....	5
นับจำนวนของข้อมูลที่อยู่ใน List.....	5
ตรวจสอบข้อมูลที่อยู่ใน List.....	5
ตรวจสอบข้อมูลที่อยู่ใน List และแสดงค่า Index ของข้อมูลที่ตรวจสอบ.....	6
เข้าถึงข้อมูลที่อยู่ใน List โดยการให้ Index.....	6
แสดงข้อมูลลำดับแรกใน List.....	7
แสดงข้อมูลลำดับสุดท้ายใน List.....	8
แสดงข้อมูลบางส่วน (Slice).....	8
แสดงข้อมูลใน List โดยใช้ for-loop.....	9
แสดงข้อมูลใน List โดยใช้ While-loop.....	10
การเพิ่มข้อมูลลงไป ใน List.....	11
การเพิ่มข้อมูลด้วยคำสั่ง append.....	11
การเพิ่มข้อมูลด้วยคำสั่ง extend.....	11
การแทรกข้อมูลด้วยคำสั่ง insert.....	11
การเชื่อม List ด้วย concatenate.....	12
การแสดงข้อมูลใน List แบบ Reverse.....	12
ค้นหาข้อมูลใน List.....	12
การลบข้อมูลใน List.....	13
การเคลียร์ (ข้อมูล) ทั้งหมดภายใน List.....	13
การลบข้อมูลใน List ด้วยคำสั่ง pop.....	13
การลบข้อมูลใน List ด้วยคำสั่ง remove.....	13
จุดเด่นของ List.....	14
จัดการข้อมูลด้วย Pickle.....	16
ใช้ Pickle สำหรับบันทึกข้อมูล.....	17
แสดงไฟล์ .pkl ในโฟลเดอร์.....	17
ใช้ Pickle เพื่อโหลดข้อมูล.....	17
แสดงข้อมูลที่โหลดมาจากไฟล์ std.pkl.....	17
การแสดงผลข้อมูลจากไดเรกทอรี.....	19
แสดงไดเรกทอรีปัจจุบัน.....	19
แสดงผลข้อมูลจากไดเรกทอรี โดยใช้ os.....	19
แสดงผลข้อมูลจากไดเรกทอรี โดยใช้ glob.....	19
ตัวอย่างการแสดงผลรูปภาพ.....	20
สร้างฟังก์ชันอ่านข้อมูลจากไดเรกทอรี.....	21
สร้างฟังก์ชันเพื่ออ่านรูปภาพ.....	22

ตัวอย่างการเรียกใช้ฟังก์ชันเพื่อแสดงรูปภาพ.....	22
โปรแกรมค้นหาข้อความในรูปภาพ (Search Text in Image).....	25
เรียกใช้ไฟล์จาก Google Drive.....	25
ติดตั้งโปรแกรม Tesseract.....	26
สร้างฟังก์ชันที่เกี่ยวข้อง.....	26
แสดงรูปภาพตัวอักษร.....	27
ค้นหาบริเวณตัวอักษร (Text Detection) จากรูปภาพ.....	28
ค้นหา (Search Text) ที่ต้องการโดยใช้ Regular Expression.....	30

ตัวแปรประเภท List

กำหนดตัวแปรประเภท List

```
name = list()

# or
name = []
```

ลบตัวแปรประเภท List

```
del name
```

```
# ไม่พบตัวแปร name
name
```

```
-----
NameError                                 Traceback (most recent call last)
<ipython-input-15-9bc0cb2ed6de> in <module>()
----> 1 name
```

```
NameError: name 'name' is not defined
```

การคัดลอกตัวแปรประเภท List

```
name = ['olarik', 'kaveepoj']
```

```
import copy

cp_name = copy.deepcopy(name)

print(cp_name)
```

```
['olarik', 'kaveepoj']
```

ตรวจสอบประเภทของตัวแปร และจำนวนสมาชิกใน List

```
name = ['olarik', 'suwich', 'kaveepoj', 'suwichai']  
  
print('Type of variable: ', type(name))  
print('Number of Element in list: ', len(name))
```

```
Type of variable: <class 'list'>  
Number of Element in list: 4
```

นับจำนวนของข้อมูลที่อยู่ใน List

```
name = ['olarik', 'suwich', 'kaveepoj', 'suwich']  
  
print('Occurrence of a given item:', name.count('suwich'))
```

```
Occurrence of a given item: 2
```

ตรวจสอบข้อมูลที่อยู่ใน List

```
# case sensitive  
'olarik' in name
```

```
False
```

```
'Olarik' in name
```

```
True
```

ตรวจสอบข้อมูลที่อยู่ใน List และแสดงค่า Index ของข้อมูลที่ตรวจสอบ

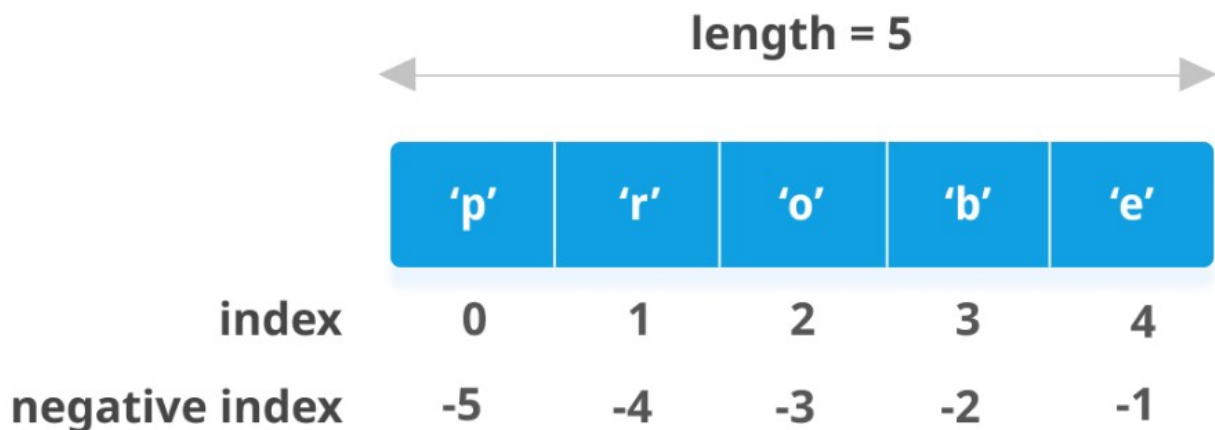
```
name = ['olarik', 'suwich', 'kaveepoj', 'suwich']
search = 'suwich'
idx = name.index(search)
print('The index of', search, ': ', idx)
```

The index of suwich : 1

```
search = 'kaveepoj'
idx = name.index(search)
print('The index of', search, ': ', idx)
```

The index of kaveepoj : 2

เข้าถึงข้อมูลที่อยู่ใน List โดยการใช้ Index



List = [0, 1, 2, 3, 4, 5]

0	1	2	3	4	5
---	---	---	---	---	---

List[0] = 0

List[0:] = [0,1,2,3,4,5]

List[1] = 1

List[:] = [0,1,2,3,4,5]

List[2] = 2

List[2:4] = [2, 3]

List[3] = 3

List[1:3] = [1, 2]

List[4] = 4

List[:4] = [0, 1, 2, 3]

List[5] = 5

แสดงข้อมูลลำดับแรกใน List

```
: print('index 0:', name[0])
   print('index 1:', name[0])
```

```
index 0: Olarik
index 1: Olarik
```

```
name  
['olarik', 'suwich', 'kaveepoj', 'suwich']
```

```
# Index ของ Python เริ่มที่ 0  
name[0]  
  
'olarik'
```

แสดงข้อมูลลำดับสุดท้ายใน List

```
name[-1]  
  
'suwich'
```

แสดงข้อมูลบางส่วน (Slice)

```
# แสดงข้อมูลตั้งแต่ลำดับแรกจำนวน 3 ชุด  
name[:3]
```

```
['olarik', 'suwich', 'kaveepoj']
```

```
# แสดงข้อมูลตั้งแต่ลำดับที่ 2 ถึงลำดับสุดท้าย  
name[1:]
```

```
['suwich', 'kaveepoj', 'suwich']
```

```
# โดยทั่วไป name[2:4] หมายถึง แสดงข้อมูลจากตัวแปร List จากลำดับที่ 2 ถึงลำดับที่ 4 ซึ่งหมายถึง 2,3 และ 4  
# แต่ใน Python จะแสดงข้อมูลของตำแหน่งที่ 2 และ 3  
name[2:4]
```

```
['kaveepoj', 'suwich']
```


แสดงข้อมูลใน List โดยใช้ for-loop

```
for n in name:  
    print(n)
```

```
Olarik  
suwich  
kaveepoj  
suwich
```

```
for i in range(0, len(name)):  
    print(name[i])
```

```
Olarik  
suwich  
kaveepoj  
suwich
```

```
for idx, n in enumerate(name):  
    print(['',idx,'], n)
```

```
[ 0 ] Olarik  
[ 1 ] suwich  
[ 2 ] kaveepoj  
[ 3 ] suwich
```

แสดงข้อมูลใน List โดยใช้ While-loop

```
i=0
while(len(name)>i):
    print(name[i])
    i += 1
```

```
Olarik
suwich
kaveepoj
suwich
```

```
i=0
while(i<len(name)):
    print('[',i,']',name[i])
    i += 1
```

```
[ 0 ] Olarik
[ 1 ] suwich
[ 2 ] kaveepoj
[ 3 ] suwich
```

การเพิ่มข้อมูลลงไปใน List

```
name = ['Olarik', 'suwich', 'kaveepoj', 'suwich']
```

การเพิ่มข้อมูลด้วยคำสั่ง append

สามารถเพิ่มข้อมูลได้ครั้งละ 1 ชุด

```
name.append('Pariwat')
```

```
print(name)
```

```
['Olarik', 'suwich', 'kaveepoj', 'suwich', 'Pariwat']
```

การเพิ่มข้อมูลด้วยคำสั่ง extend

โดยสามารถเพิ่มข้อมูลลงไปใน list ได้มากกว่า 1 ชุดต่อครั้ง

```
name.extend(['sarayut', 'sangdaow'])
```

```
print(name)
```

```
['Olarik', 'suwich', 'kaveepoj', 'suwich', 'Pariwat', 'sarayut', 'sangdaow']
```

การแทรกข้อมูลด้วยคำสั่ง insert

```
# แทรกข้อมูล 'blue' ณ ตำแหน่งก่อนสุดท้าย (index -1)
name.insert(-1, 'blue')
```

```
print(name)
```

```
['Olarik', 'suwich', 'kaveepoj', 'suwich', 'Pariwat', 'sarayut', 'blue', 'sangdaow']
```

```
# แทรกข้อมูล 'yellow' ณ ตำแหน่งแรก (index 0)
name.insert(0, 'yellow')
```

```
print(name)
```

```
['yellow', 'Olarik', 'suwich', 'kaveepoj', 'suwich', 'Pariwat', 'sarayut', 'blue', 'sangdaow']
```

```
# แทรกข้อมูล 'red' ณ ตำแหน่งที่ 1 (index 1)
name.insert(1, 'red')
```

```
print(name)
```

```
['yellow', 'red', 'Olarik', 'suwich', 'kaveepoj', 'suwich', 'Pariwat', 'sarayut', 'blue', 'sangdaow']
```

การเชื่อม List ด้วย concatenate

```
color = ['yellow', 'blue']
color1 = ['red', 'green']

new_color = color+color1
print(new_color)

['yellow', 'blue', 'red', 'green']
```

การแสดงข้อมูลใน List แบบ Reverse

```
new_color.reverse()

print(new_color)

['green', 'red', 'blue', 'yellow']
```

ค้นหาข้อมูลใน List

```
search = 'r'
for c in new_color:
    if(search in c):
        print('Found:', search, 'in [', c, ']')
    else:
        print('Not Found:', search, 'in [', c, ']')
```

```
Found: r in [ green ]
Found: r in [ red ]
Not Found: r in [ blue ]
Not Found: r in [ yellow ]
```

```
name = ['Olarik', 'suwich', 'kaveepoj', 'suwichai', 'Pariwat', 'sarayut', 'sangdaow']

search = 'suw'
for n in name:
    if(search in n):
        print('Found:', search, 'in [', n, ']')
```

```
Found: suw in [ suwich ]
Found: suw in [ suwichai ]
```

การลบข้อมูลใน List

```
name = ['Olarik', 'suwich', 'kaveepoj', 'suwichai', 'Pariwat', 'sarayut', 'sangdaow']
print(name)
```

```
['Olarik', 'suwich', 'kaveepoj', 'suwichai', 'Pariwat', 'sarayut', 'sangdaow']
```

การเคลียร์ (ข้อมูล) ทั้งหมดภายใน List

```
name.clear()
```

```
print(name)
```

```
[]
```

การลบข้อมูลใน List ด้วยคำสั่ง pop

คำสั่ง pop จะลบข้อมูลตัวล่าสุด (ตัวสุดท้าย หรือ Index ที่ -1) ออกจาก List

```
name = ['Olarik', 'suwich', 'kaveepoj', 'suwichai', 'Pariwat', 'sarayut', 'sangdaow']
```

```
name.pop()
print(name)
```

```
['Olarik', 'suwich', 'kaveepoj', 'suwichai', 'Pariwat', 'sarayut']
```

ในคำสั่ง pop สามารถกำหนด index ที่ต้องการลบได้

เช่น name.pop(0) หมายถึง การลบข้อมูลใน name โดยลบข้อมูลที่อยู่ใน index ที่ 0 ซึ่งก็คือ 'Olarik'

```
name.pop(0)
print(name)
```

```
['suwich', 'kaveepoj', 'suwichai', 'Pariwat', 'sarayut']
```

การลบข้อมูลใน List ด้วยคำสั่ง remove

```
name.remove('Pariwat')
print(name)
```

```
['suwich', 'kaveepoj', 'suwichai', 'sarayut']
```

การใช้คำสั่ง remove เพื่อลบข้อมูลออกจาก List จะต้องกำหนดข้อมูลที่จะลบให้ตรงกับข้อมูลที่อยู่ใน List เท่านั้น เนื่องจากเป็น case-sensitive

ดังนั้น 'Suwich' จึงมีค่าไม่เท่ากับ 'suwich'

```
name.remove('Suwich')
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-60-6b4fcd3681ea> in <module>()
----> 1 name.remove('Suwich')
```

```
ValueError: list.remove(x): x not in list
```

จุดเด่นของ List

สามารถเก็บข้อมูลได้หลากหลายประเภท

```
student = ['Wichulada', 'Information Technology', 160, 45.5]

for s in student:
    print('Data :', s)
    print('Type :', type(s))
    print()
```

Data : Wichulada

Type : <class 'str'>

Data : Information Technology

Type : <class 'str'>

Data : 160

Type : <class 'int'>

Data : 45.5

Type : <class 'float'>

```
student_data = [
    ['Wichulada', 'Information Technology', 160, 45.5],
    ['Surakiet', 'Geo-Informatics', 175, 55],
    ['Suwichai', 'New Media', 160, 57.2]
]

for student in student_data:
    print(student)
```

['Wichulada', 'Information Technology', 160, 45.5]

['Surakiet', 'Geo-Informatics', 175, 55]

['Suwichai', 'New Media', 160, 57.2]

```
detail = ['Name', 'Dep', 'Height', 'Weight']

print('Student Detail')
print('=====\n')
for student in student_data:
    for i in range(0, len(student)):
        print(detail[i],':', student[i])
    print()
```

Student Detail

=====

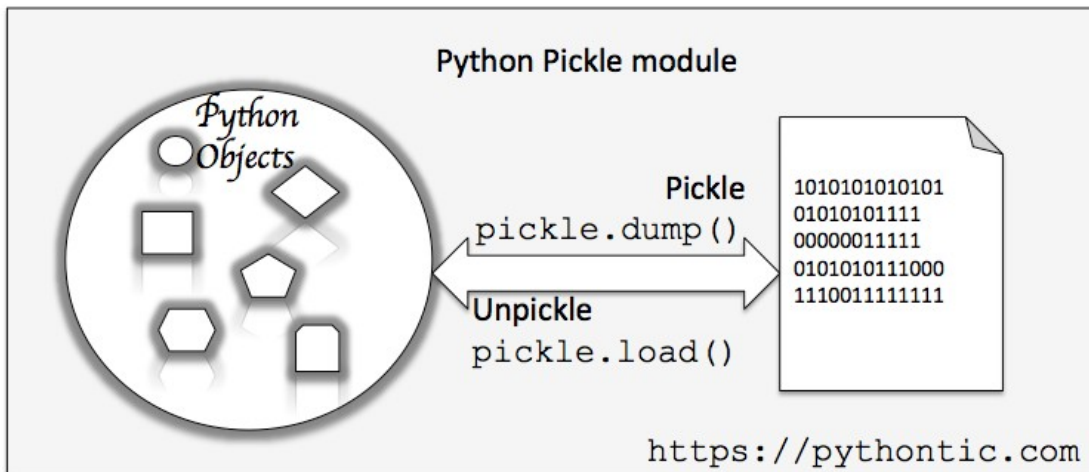
Name : Wichulada
Dep : Information Technology
Height : 160
Weight : 45.5

Name : Surakiet
Dep : Geo-Informatics
Height : 175
Weight : 55

Name : Suwichai
Dep : New Media
Height : 160
Weight : 57.2

จัดการข้อมูลด้วย Pickle

Pickle เป็น Module ในภาษา Python ที่ช่วยในการแปลง Object ให้อยู่ในรูปแบบของ character



```

import pickle
...
with open('mydata.pickle', 'wb') as mysavedata:
    pickle.dump([1, 2, 'three'], mysavedata)
...
with open('mydata.pickle', 'rb') as myrestoredata:
    a_list = pickle.load(myrestoredata)
print(a_list)

```

Always remember to import the "pickle" module.

To save your data, use "dump()".

Assign your restored data to an identifier.

Restore your data from your file using "load()".

The "b" tells Python to open your data files in BINARY mode.

Once your data is back in your program, you can treat it like any other data object.

ใช้ Pickle สำหรับบันทึกข้อมูล

```
import pickle

# บันทึกข้อมูลชื่อ std.pkl
filename = 'std.pkl'

with open(filename, 'wb') as f:
    pickle.dump([student_data, detail] , f)
```

แสดงไฟล์ .pkl ในโฟลเดอร์

```
!ls *.pkl
std.pkl
```

ใช้ Pickle เพื่อโหลดข้อมูล

```
import pickle

# โหลดข้อมูลจากไฟล์ std.pkl โดยเก็บตัวแปรที่โหลดมาเก็บไว้ที่ std_info, info
filename = 'std.pkl'
with open(filename, 'rb') as f:
    std_info, info = pickle.load(f)
```

แสดงข้อมูลที่โหลดมาจากไฟล์ std.pkl

```
std_info
```

```
[['Wichulada', 'Information Technology', 160, 45.5],
 ['Surakiet', 'Geo-Informatics', 175, 55],
 ['Suwichai', 'New Media', 160, 57.2]]
```

```
info
```

```
['Name', 'Dep', 'Height', 'Weight']
```

ตัวอย่างการบันทึกข้อมูลลง List

```
# ตัวอย่างข้อมูล
info = ['Name', 'Dep', 'Height', 'Weight']
std_info = [
    ['Wichulada', 'Information Technology', 160, 45.5],
    ['Surakiet', 'Geo-Informatics', 175, 55],
    ['Suwichai', 'New Media', 160, 57.2]
]
```

```
print('Please fill in your information')
print('=====')
std_name = input('Name: ')
std_dep = input('Department: ')
std_height = float(input('Height: '))
std_weight = float(input('Weight: '))

std_tmp = [std_name, std_dep, std_height, std_weight]
std_info.append(std_tmp)
```

```
Please fill in your information
=====
Name: Olarik
Department: Information Technology
Height: 165
Weight: 75
```

```
print('Student Detail')
print('=====\\n')
for student in std_info:
    for i in range(0, len(student)):
        print(info[i],':', student[i])
    print()
```

```
Student Detail
=====

Name : Wichulada
Dep : Information Technology
Height : 160
Weight : 45.5

Name : Surakiet
Dep : Geo-Informatics
Height : 175
Weight : 55

Name : Suwichai
Dep : New Media
Height : 160
Weight : 57.2

Name : Olarik
Dep : Information Technology
Height : 165.0
Weight : 75.0
```

การแสดงข้อมูลจากไดเรกทอรี

แสดงไดเรกทอรีปัจจุบัน

```
!pwd
```

```
/content
```

แสดงข้อมูลจากไดเรกทอรี โดยใช้ os

```
import os  
  
path = '/content'  
for entry in os.listdir(path):  
    print(entry)
```

```
.config  
dog-01.jpg  
dog-03.jpg  
dog-02.jpg  
dog-04.jpg  
sample_data
```

แสดงข้อมูลจากไดเรกทอรี โดยใช้ glob

```
import glob  
  
for file in glob.glob('/content/*.jpg*'):  
    print(file)
```

```
/content/dog-01.jpg  
/content/dog-03.jpg  
/content/dog-02.jpg  
/content/dog-04.jpg
```

```

import glob

path = '/content/'
all_files = glob.iglob(path + '.*[jp]*[g]', recursive=True)

path_img = [] #path_img = list() #create empty list
for file in all_files:
    path_img.append(file)

# show image path
for p in path_img:
    print(p)

```

```

/content/dog-01.jpg
/content/dog-03.jpg
/content/dog-02.jpg
/content/dog-04.jpg

```

ตัวอย่างการแสดงรูปภาพ

```

import cv2
import matplotlib.pyplot as plt

#path = 'C:\Desktop\workspace\dog-01.jpg'
def cv_read_image(path):
    img = cv2.imread(path)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    return img_rgb

r = 1
c = 2
#path = ['/content/dog-01.jpg', '/content/dog-02.jpg']
path = ['dog-01.jpg', 'dog-02.jpg']

cnt=0
plt.figure(figsize=(15,10))
for i in range(1, len(path)+1):
    try:
        plt.subplot(r, c, i)
        img = cv_read_image(path[cnt])
        plt.imshow(img)
        cnt = cnt+1
    except:
        pass

plt.show()

```

ผลลัพธ์



สร้างฟังก์ชันอ่านข้อมูลจากไดเรกทอรี

```
import glob
import cv2
import matplotlib.pyplot as plt

def path_image(path):
    all_files = glob.iglob(path + '.*[jp]*[g]', recursive=True)
    path_img = [] #path_img = list() #create empty list
    for file in all_files:
        path_img.append(file)

    return path_img
```

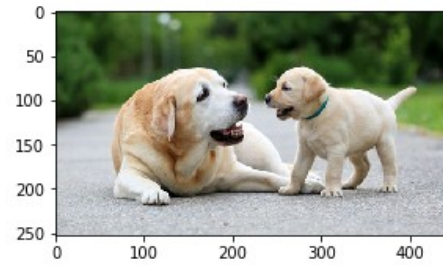
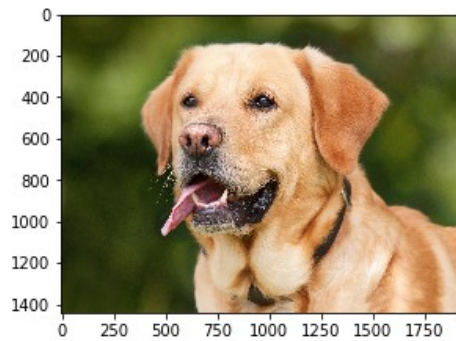
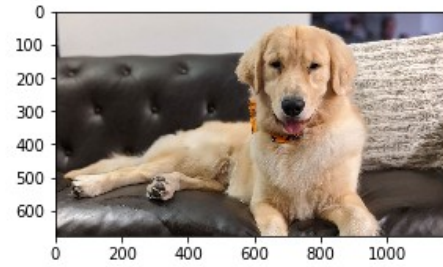
สร้างฟังก์ชันเพื่ออ่านรูปภาพ

```
def cv_read_image(path):  
    img = cv2.imread(path)  
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
  
    return img_rgb
```

ตัวอย่างการเรียกใช้ฟังก์ชันเพื่อแสดงรูปภาพ

```
: import math  
  
path = '/content/'  
path_img = path_image(path)  
  
r = math.floor(math.sqrt(len(path_img)))  
#path_img = ['/content/a.jpg', '/content/sub/b.jpg', '/content/sub/sub/abc.png']  
# len(path_img) = 5  
#math.sqrt(5) = 2.xxx  
#math.floor(2.xxxx) = 2  
cnt=0  
plt.figure(figsize=(10,10))  
for i in range(1, len(path)+1):  
    try:  
        plt.subplot(r, r, i)  
        img = cv_read_image(path_img[cnt])  
        plt.imshow(img)  
        cnt = cnt+1  
    except:  
        pass  
  
plt.show()
```

ผลลัพธ์



ตัวอย่างแสดงรูปภาพ

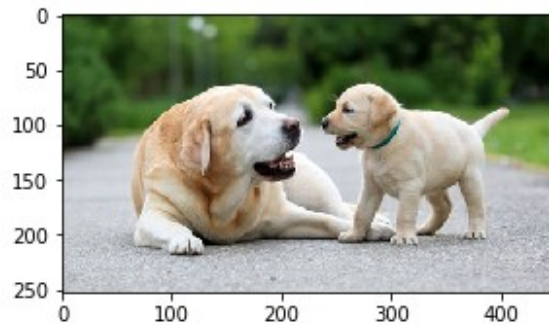
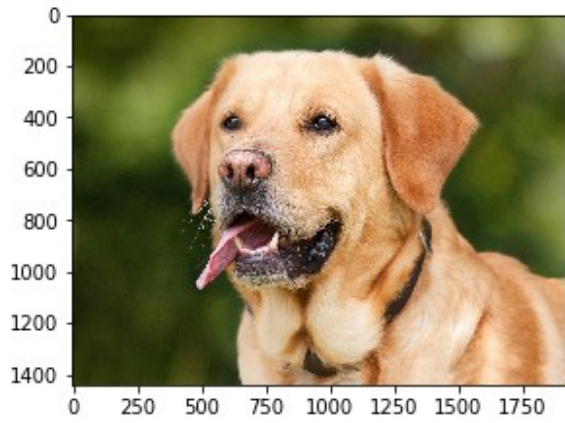
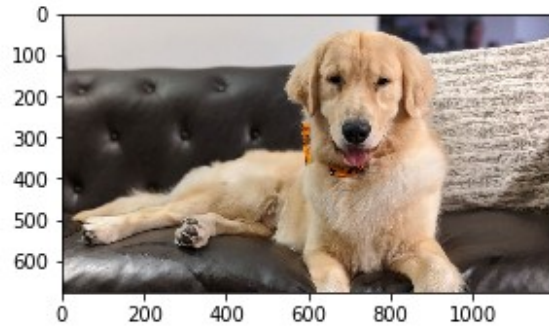
```
import math

path = '/content/'
path_img = path_image(path)

r = math.floor(math.sqrt(len(path_img))) + 1

plt.figure(figsize=(10,10))
for idx, p_img in enumerate(path_img):
    try:
        plt.subplot(r, r, idx+1)
        img = cv_read_image(p_img)
        plt.imshow(img)
    except:
        pass

plt.show()
```



โปรแกรมค้นหาข้อความในรูปภาพ (Search Text in Image)

เรียกใช้ไฟล์จาก Google Drive

ใช้ Library ของ google

```
from google.colab import drive
drive.mount('/gdrive')
#%cd /gdrive
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client=4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoc2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2f%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2f

Enter your authorization code:

.....

Mounted at /gdrive

เรียกดูข้อมูลจาก Google Drive

```
ls '/gdrive/My Drive/Colab Notebooks/2-2562-DIP/text-image'
```

```
text-01.jpg text-03.png text-05.png text-07.jpg text-09.jpg
text-02.png text-04.png text-06.jpg text-08.jpg text-10.png
```

ติดตั้งโปรแกรม Tesseract

ติดตั้งโดยใช้คำสั่ง pip และ apt-get

```

$ pip install pytesseract
$ apt-get install tesseract-ocr-tha

Requirement already satisfied: pytesseract in /usr/local/lib/python3.6/dist-packages (0.3.1)
Requirement already satisfied: Pillow in /usr/local/lib/python3.6/dist-packages (from pytesseract) (4.3.0)
Requirement already satisfied: olefile in /usr/local/lib/python3.6/dist-packages (from Pillow->pytesseract) (0.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnvidia-common-430
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd
The following NEW packages will be installed:
  tesseract-ocr tesseract-ocr-eng tesseract-ocr-osd tesseract-ocr-tha
0 upgraded, 4 newly installed, 0 to remove and 7 not upgraded.
Need to get 5,693 kB of archives.
After this operation, 16.9 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr-eng all 4.00-git24-0e00fe6-1.2 [1,58
Get:2 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr-osd all 4.00-git24-0e00fe6-1.2 [2,98
Get:3 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr amd64 4.00-git2288-10f4998a-2 [218 k
Get:4 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tesseract-ocr-tha all 4.00-git24-0e00fe6-1.2 [898
Fetched 5,693 kB in 2s (3,241 kB/s)
Selecting previously unselected package tesseract-ocr-eng.
(Reading database ... 135004 files and directories currently installed.)
Preparing to unpack .../tesseract-ocr-eng_4.00-git24-0e00fe6-1.2_all.deb ...
Unpacking tesseract-ocr-eng (4.00-git24-0e00fe6-1.2) ...
Selecting previously unselected package tesseract-ocr-osd.
Preparing to unpack .../tesseract-ocr-osd_4.00-git24-0e00fe6-1.2_all.deb ...
Unpacking tesseract-ocr-osd (4.00-git24-0e00fe6-1.2) ...
Selecting previously unselected package tesseract-ocr.

```

สร้างฟังก์ชันที่เกี่ยวข้อง

```

import glob
import cv2
import matplotlib.pyplot as plt
import pytesseract
from pytesseract import Output

def path_image(path):
    all_files = glob.iglob(path + '.*[jp]*[g]', recursive=True)
    path_img = [] #path_img = list() #create empty list
    for file in all_files:
        path_img.append(file)

    return path_img

def cv_read_image(path):
    img = cv2.imread(path)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    return img_rgb

def image_to_text(img, confidence=60, verbose=False):
    d = pytesseract.image_to_data(img, output_type=Output.DICT)
    n_boxes = len(d['text'])
    for i in range(n_boxes):
        if int(d['conf'][i]) > confidence:
            (x, y, w, h) = (d['left'][i], d['top'][i], d['width'][i], d['height'][i])
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

    if(verbose):
        plt.figure(figsize=(10,5))
        plt.imshow(img)
        plt.show()

    return d, img

```

แสดงรูปภาพตัวอักษร

```
import math

path = '/gdrive/My Drive/Colab Notebooks/2-2562-DIP/text-image/'
path_img = path_image(path)

r = math.floor(math.sqrt(len(path_img))) + 1
cnt=0
plt.figure(figsize=(15,15))
for i in range(1, len(path)+1):
    try:
        plt.axis('off')
        plt.subplot(r, r, i)
        img = cv_read_image(path_img[cnt])
        plt.imshow(img)
        cnt = cnt+1
    except:
        pass

plt.show()
```

ผลลัพธ์

This is the Hart District.
Please do not go to
Seagus High School.
Police activity has the
campus closed. More to
follow via email

There is an incident at
Seagus HS - as a
precaution, we are
placing all schools on
lockdown until law
enforcement advises
otherwise

This is the first line of
this text example.

This is the second line
of the same text.

From this
point on,
I'm going to
treat people
exactly how
they treat me.
some should
be glad,
some should
be scared.

THERE IS A
DIFFERENCE
BETWEEN
GIVING UP
AND KNOWING
WHEN YOU
HAVE HAD
ENOUGH

THIS TEXT CLIPS THE
BACKGROUND IMAGE
THIS TEXT CLIPS THE
BACKGROUND IMAGE

WHOEVER
IS TRYING TO
BRING YOU
DOWN,
IS ALREADY
BELOW YOU.

IF PEOPLE
ARE TRYING
TO BRING YOU
DOWN
IT ONLY MEANS
THAT YOU ARE
ABOVE THEM

The elusive
BIGTEXT
plugin exclusively
captured on film

I am curious about
area-filling text
rendering options

*maybe im just
acting like a bitch
because im tired
of being walked
over and
forgotten*

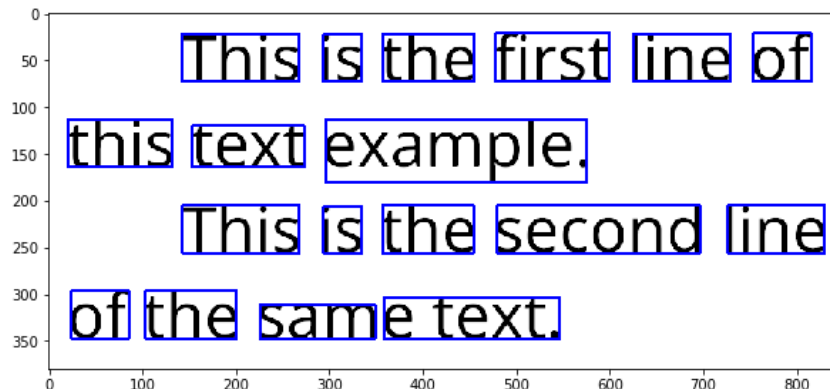
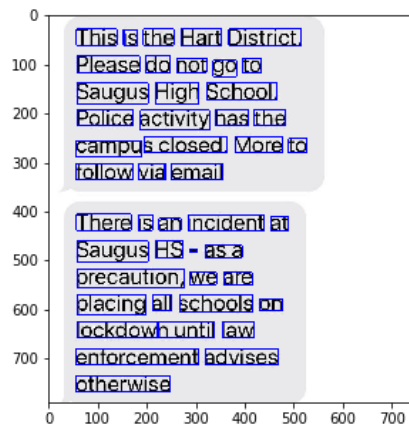
ค้นหาบริเวณตัวอักษร (Text Detection) จากรูปภาพ

```
import numpy as np

text_data = list()
img_texts = list()

for i in range(0, len(path_img)):
    img = cv_read_image(path_img[i])
    d, img_text = image_to_text(img, confidence=60, verbose=True)
    text_data.append(d)
    img_texts.append(img_text)
```

ผลลัพธ์



THERE IS A
DIFFERENCE
BETWEEN
GIVING UP
AND KNOWING
WHEN YOU
HAVE HAD
ENOUGH

THIS TEXT CLIPS THE
BACKGROUND IMAGE

THIS TEXT CLIPS THE
BACKGROUND IMAGE

ค้นหาคำ (Search Text) ที่ต้องการโดยใช้ Regular Expression

```

import re
import numpy as np

text_data = list()
img_texts = list()

# คำที่ต้องการค้นหา (เฉพาะภาษาอังกฤษ)
pattern = '[tT][he]'
#pattern = 'plugin'
print('Searching...\nPattern: {}'.format(pattern))

show_found_word = True

# ค้นหาคำในรูปภาพทั้งหมด
print('\n[Output]\n')
for i in range(0, len(path_img)):
    img = cv_read_image(path_img[i])
    search_img = np.array(img, copy=True)

    d, img_text = image_to_text(img, confidence=60, verbose=False)
    text_data.append(d)
    img_texts.append(img_text)

    n_boxes = len(text_data[i]['text'])

    chk = list()
    for ib in range(n_boxes):
        if int(text_data[i]['conf'][ib]) > 60:
            if re.search(pattern, text_data[i]['text'][ib]):
                (x, y, w, h) = (text_data[i]['left'][ib], text_data[i]['top'][ib],
                               text_data[i]['width'][ib], text_data[i]['height'][ib])
                cv2.rectangle(search_img, (x, y), (x + w, y + h), (255, 0, 0), 2)
            if show_found_word:
                print('Match:', text_data[i]['text'][ib])
            chk.append('Yes')

    if len(chk) > 0:
        print()
        plt.figure(figsize=(10,5))
        plt.title('Found in ' + path_img[i].split('/')[-1] + ': ' + str(len(chk)) + ' words')
        plt.imshow(search_img)
        plt.show()
        print()

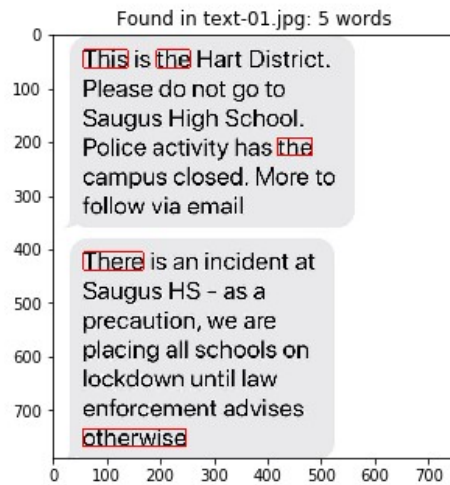
```

ผลลัพธ์

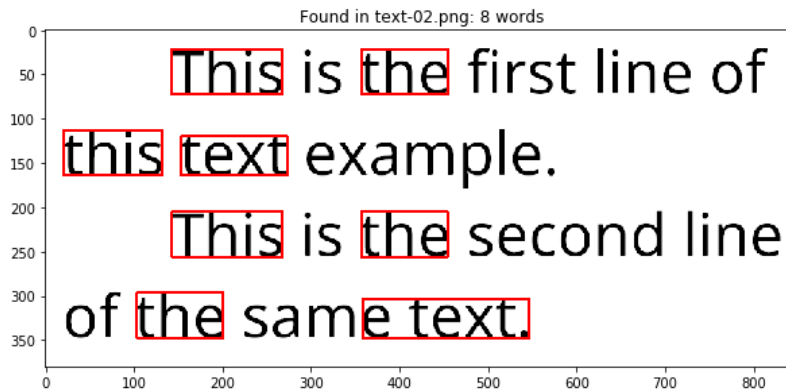
Searching...
 Pattern: [tT][he]

[Output]

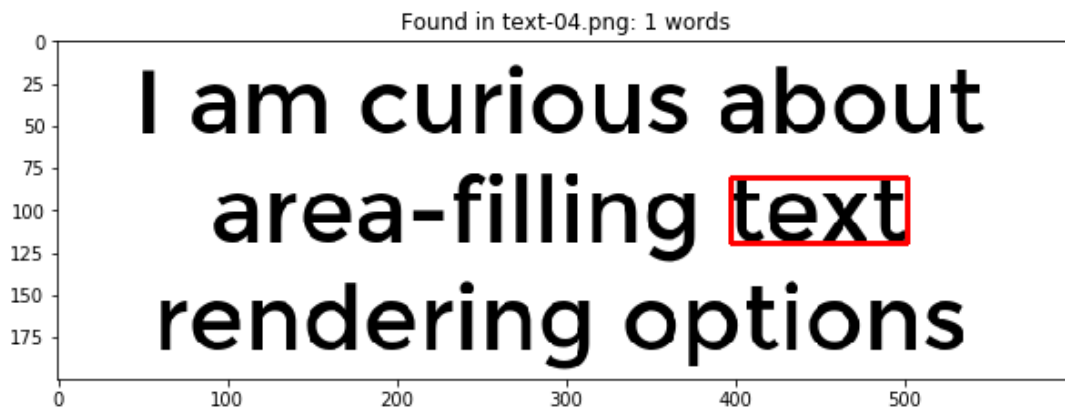
Match: This
 Match: the
 Match: the
 Match: There
 Match: otherwise



Match: This
 Match: the
 Match: this
 Match: text
 Match: This
 Match: the
 Match: the
 Match: text.



Match: text



Match: forgotten

