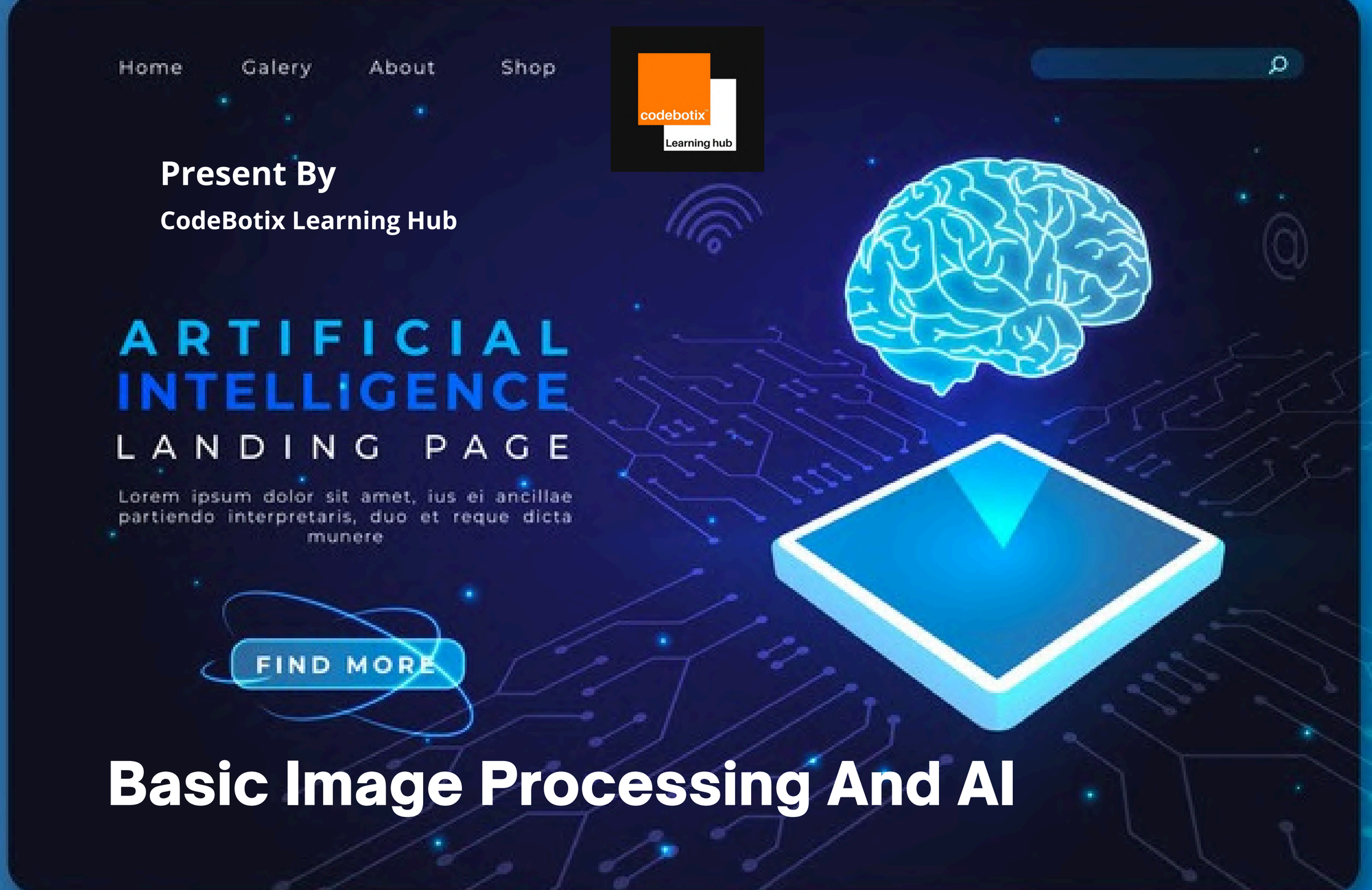
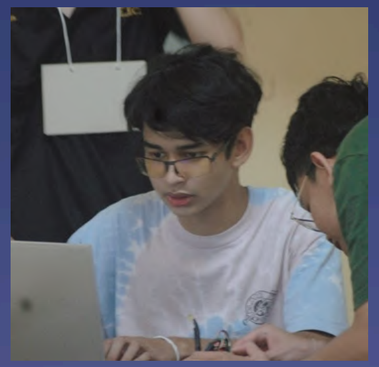


Basic Image Processing And AI

Let's get to know each other first



P’ BLUE

Bachelor Computer Engineering

King Mongkut's Institute Of Technology Ladkrabang
(KMITL)

Topic

which we could learning today?

1. OpenCV
2. Mediapipe - - - - - > Mediapipe Holistic
   (guide line)

" Wisdom comes with age, so just live and learn."

By William Landry

Open-cv

OpenCV
(Open Source Computer Vision Library)

- is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

https://opencv.org/get-started/

OpenCV

Face detection

Q1 Install Python

Q2

open your command prompt

```
pip install opencv-python
```

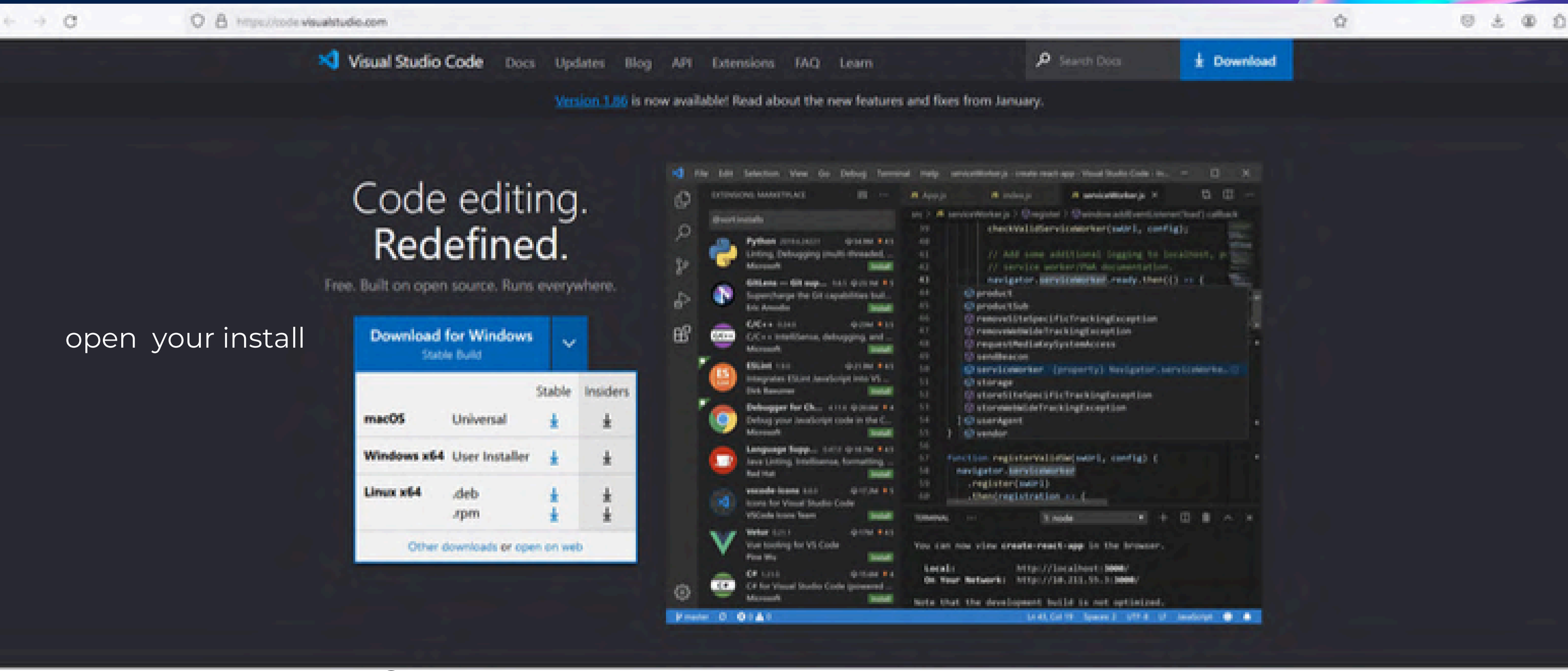
```
pip install numpy
```

We' ll use Visual Studio Code

- Creat new file your project (.py file)

Install Vscode

open your install



IntelliSense

Run and Debug

Built-in Git

Extensions

Q3 Coding

Model (face , eye , etc.)

https://github.com/opencv/opencv/tree/master/data

```python
import cv2
import numpy as np
import requests

xml_model_url = 'https://raw.githubusercontent.com/opencv/opencv/master/data/haarcascades/haarcascade_frontalface_default.xml'
xml_path = 'haarcascade_frontalface_default.xml'

def download_xml(url, path):
    response = requests.get(url)
    if response.status_code == 200:
        with open(path, 'wb') as file:
            file.write(response.content)
        print(f"Downloaded {path} successfully.")
    else:
        raise Exception(f"Failed to download {url}, status code: {response.status_code}")

# Download haarCascade XML file
download_xml(xml_model_url, xml_path)

cascade = cv2.CascadeClassifier(xml_path)

if cascade.empty():
    raise IOError('Unable to load the face cascade classifier xml file')
```

download model path

Initialize the model haarcascades

Link Video Basic AI

Q3

Coding

Model (face , eye , etc.)

https://github.com/opencv/opencv/tree/master/data

```python
# Video Capture Start 'Camera' ---> Source 0
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    objects = cascade.detectMultiScale(
        gray,
        scaleFactor=1.3,
        minNeighbors=5,
        minSize=(30, 30)
    )

    for (x, y, w, h) in objects:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

    cv2.imshow('Face Object Detection', frame)

    # Break the loop if 'q' key is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the capture and close all OpenCV windows
cap.release()
cv2.destroyAllWindows()
```

Initialize the webcam

Read a frame from the webcam

Convert the frame to grayscale

Perform object detection

When press key 'q' it will exist

Link Video Basic AI

Try on your self

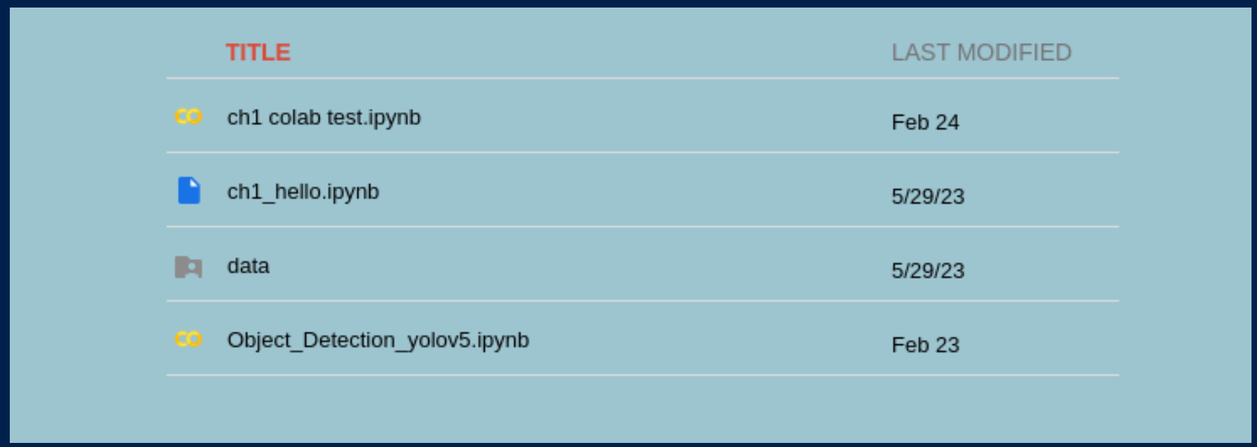
to run program

```
PS C:\Users\bluep\Desktop\AI Learning\Object Detecting_test> python face_detection_opencv.py
```

directory path file already in

python filename.py

อย่าลืม cd ตำแหน่งไฟล์ให้ถูกต้อง



TITLE	LAST MODIFIED
ch1 colab test.ipynb	Feb 24
ch1_hello.ipynb	5/29/23
data	5/29/23
Object_Detection_yolov5.ipynb	Feb 23

can try on others object example detection on this link : https://shorturl.at/pHIW6

DO YOU WANT CONTINUE

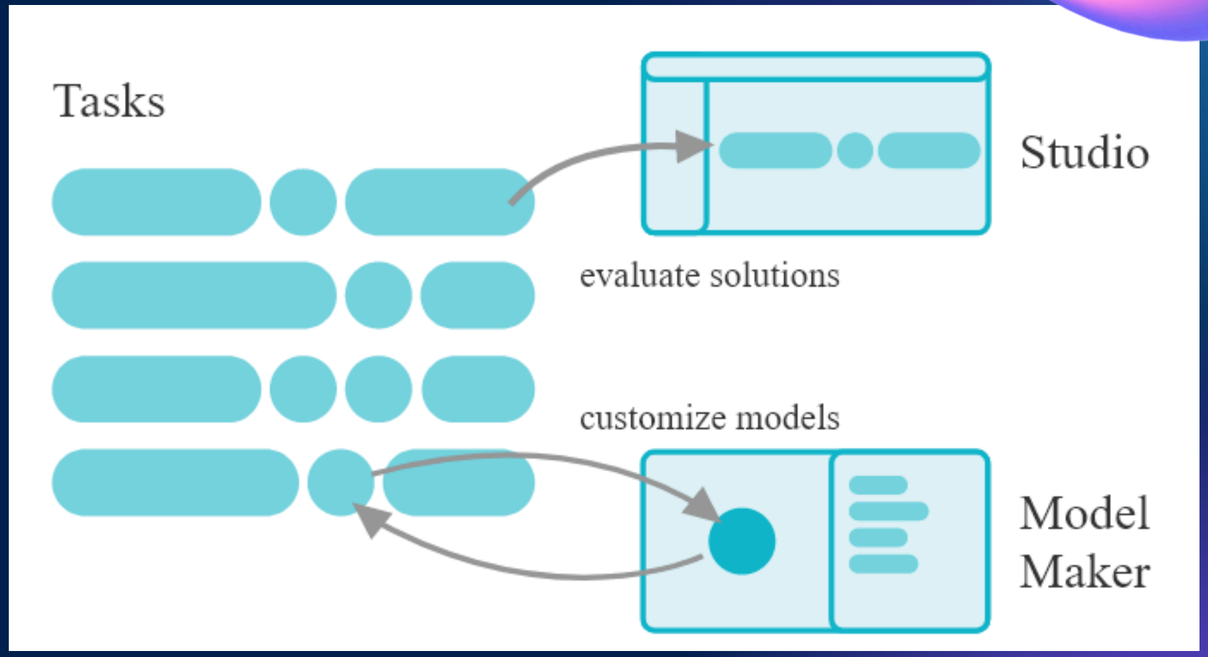
DETECTION ANALYZE TYPE >/?<

NEXT PAGE >>

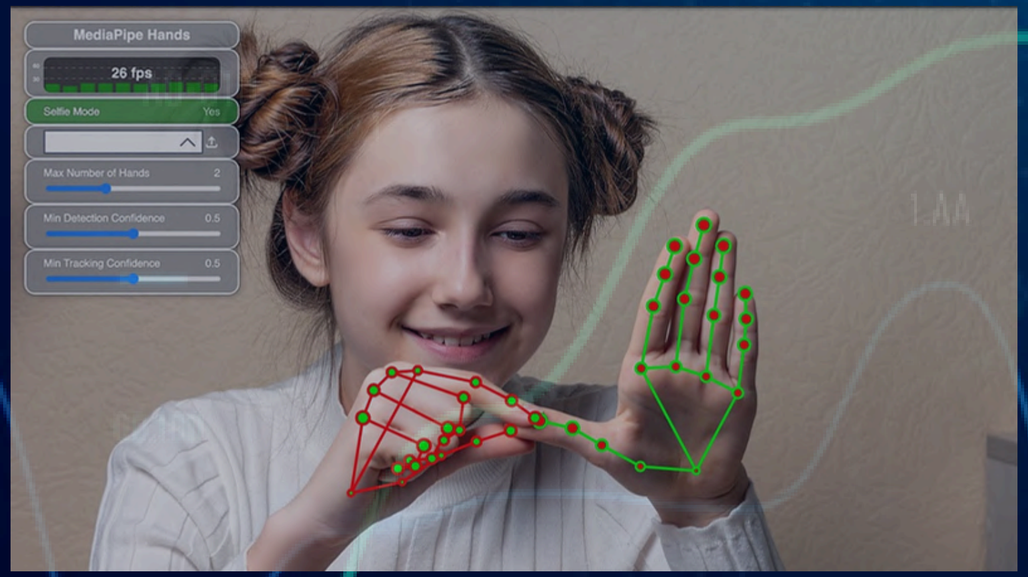
Mediapipe is?

- MediaPipe Solutions provides a suite of libraries and tools for you to quickly apply artificial intelligence (AI) and machine learning (ML) techniques in your applications.

- You can plug these solutions into your applications immediately, customize them to your needs, and use them across multiple development platforms. MediaPipe Solutions is part of the MediaPipe open source project, so you can further customize the solutions code to meet your application needs.

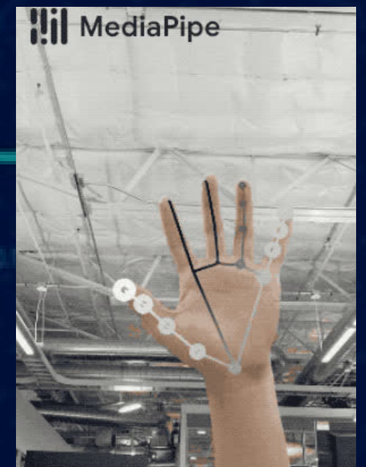


MediaPipe Hands

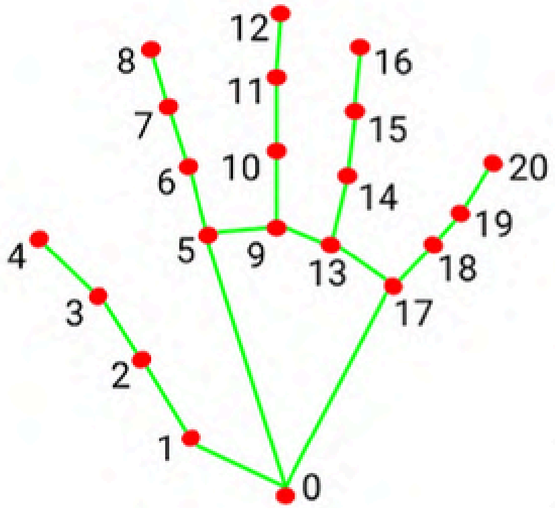


Today we will learn to use MediaPipe, which is a Machine Learning Solutions or a ready-made program from Google that can be used to do Hand Tracking accurately. And as fast as real-time tracking.

Because MediaPipe has a number of solutions available, ranging from gesture detection to facial recognition, today we're going to use MediaPipe Hands that detect hands.



21 point of finger Landmarks



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Hand Tracking starts by sensing the palm with the " Palm Detection Model " and then identifies 21 key hand locations.

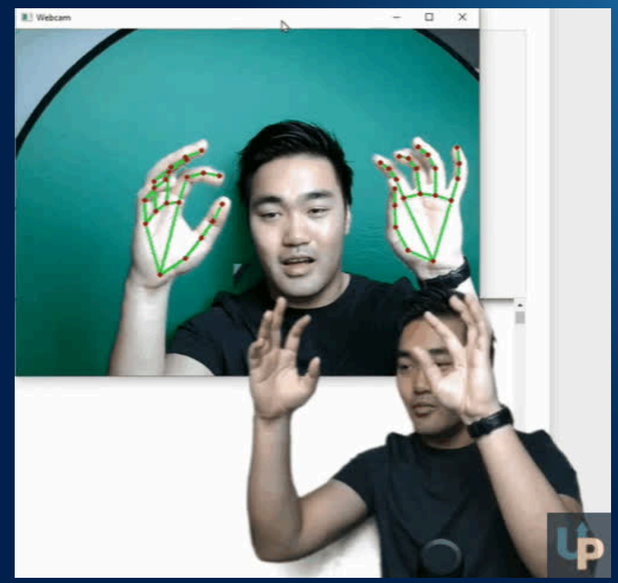
Hand Tracking

open your command prompt

Q1

```
pip install mediapipe
```

```
pip install opencv-python
```



Q3 Hand Track Coding

Our Vision

```python
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
mp_draw = mp.solutions.drawing_utils

webcam = cv2.VideoCapture(0)

hands = mp_hands.Hands()

while True:
    success, image = webcam.read()

    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    results = hands.process(image_rgb)

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_draw.draw_landmarks(image, hand_landmarks, mp_hands.HAND_CONNECTIONS)


    cv2.imshow("Webcam", image)
    cv2.waitKey(1)
```

Initialize the webcam

Initialize the Hand model

Read a frame from the webcam

Convert the frame from BGR to RGB

Process the frame to detect hand landmarks

Print the detected hand landmarks' positions

Try on your self

Hand Count Finger Send To NodeRed

open your command prompt

Q1

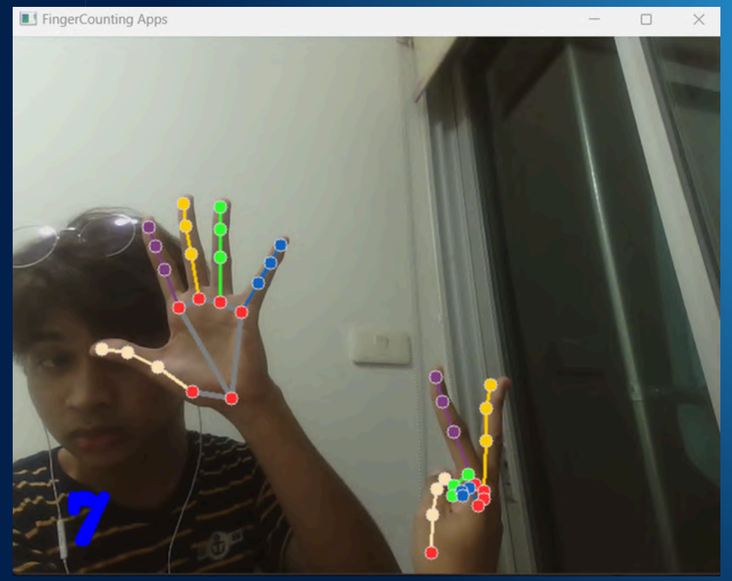
```
pip  install mediapipe
```

```
pip  install opencv-python
```

Q2

install NodeRed

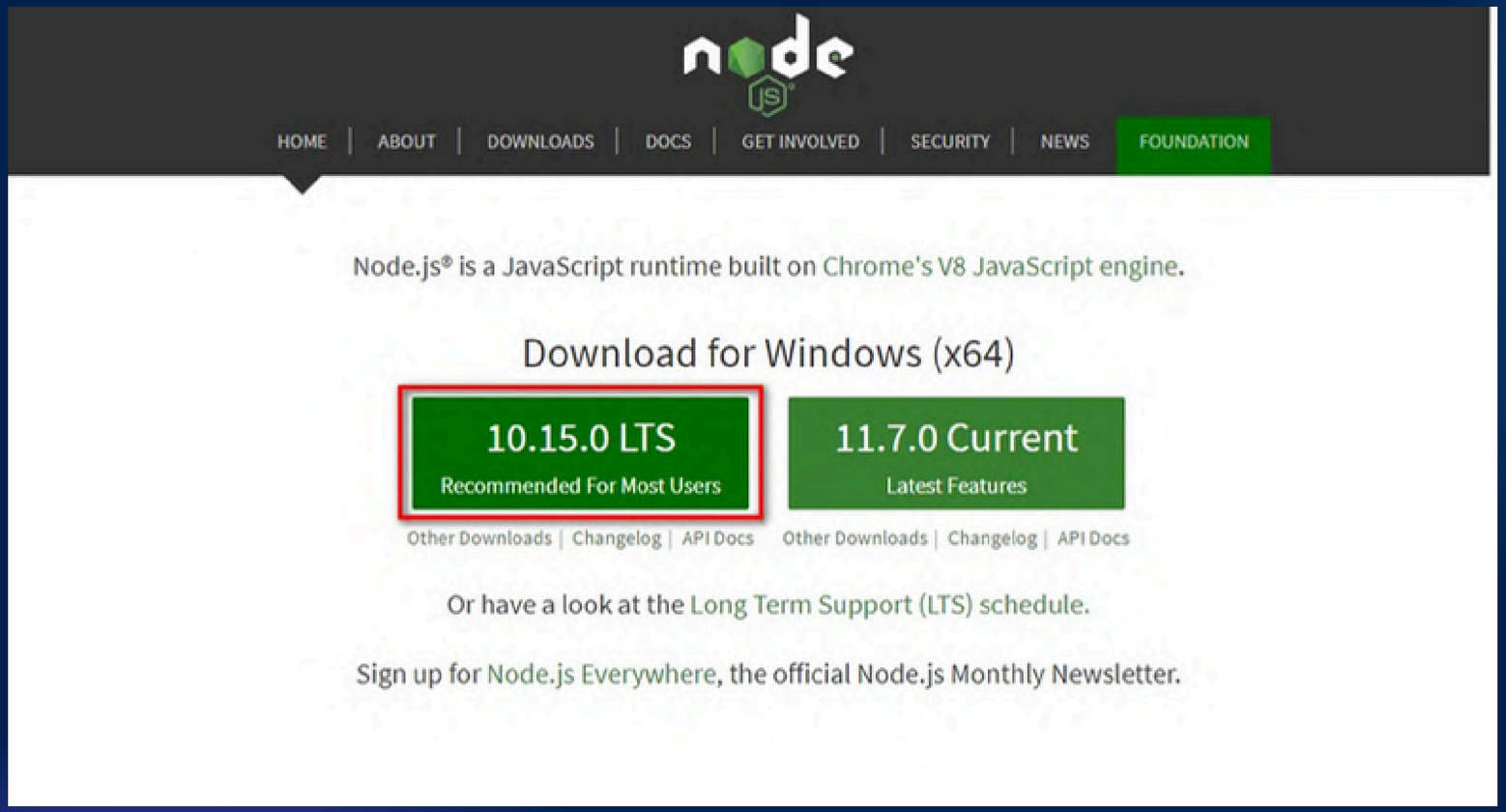
```
pip  install requests
```



Node-RED

Hand Count Finger Send To NodeRed

Q2 install NodeRed

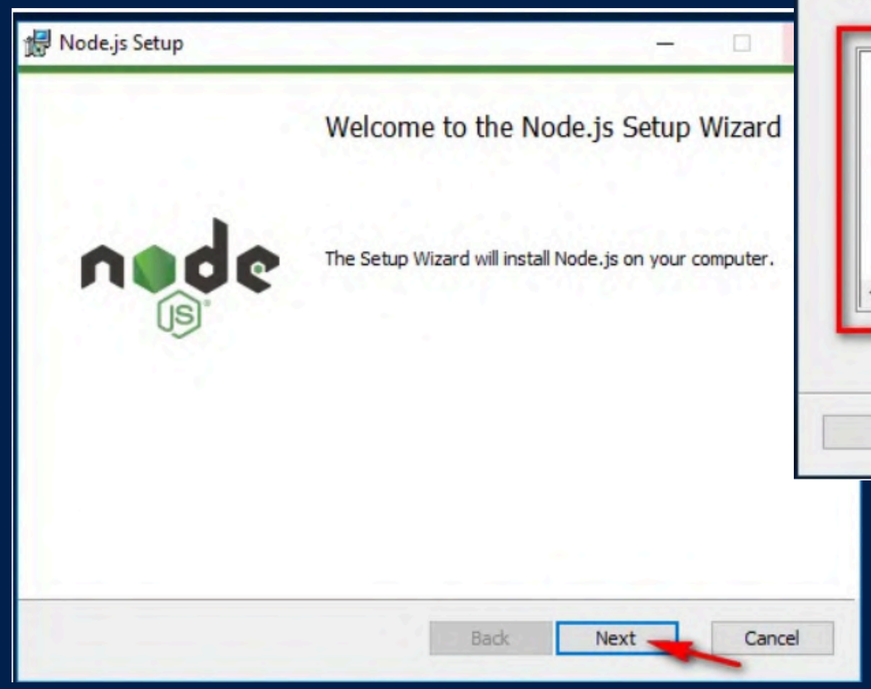


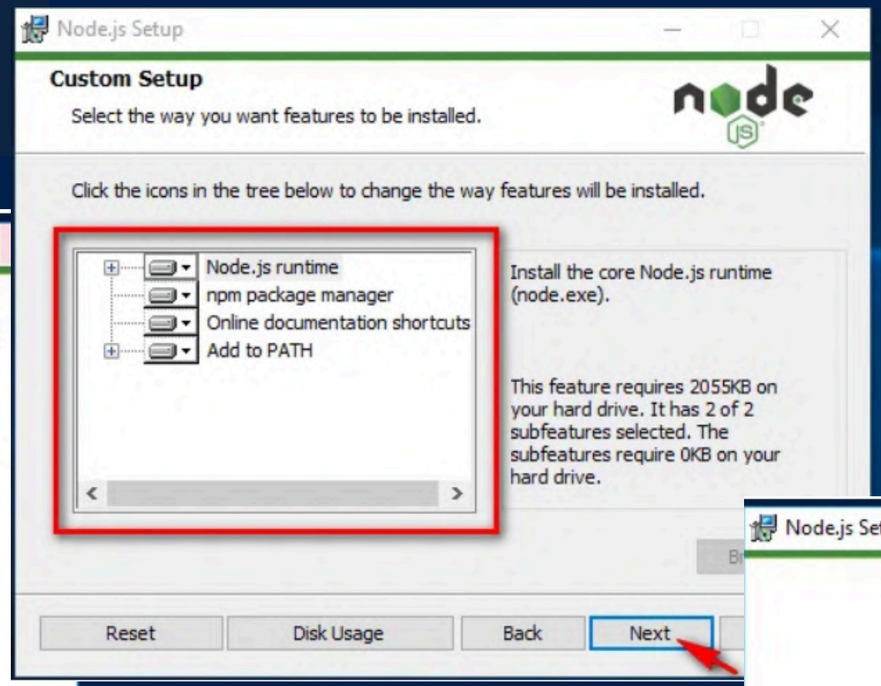
Node-RED

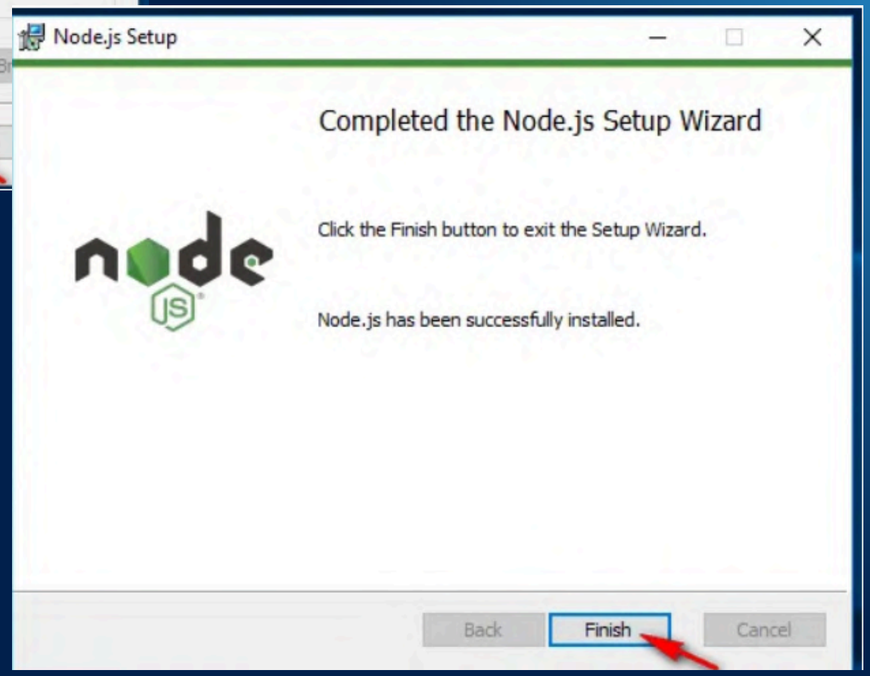
1. Install Node.JS (https://nodejs.org/en/)
Go to the nodejs.org website and choose to download the LTS version.

Hand Count Finger Send To NodeRed

Q2 install NodeRed





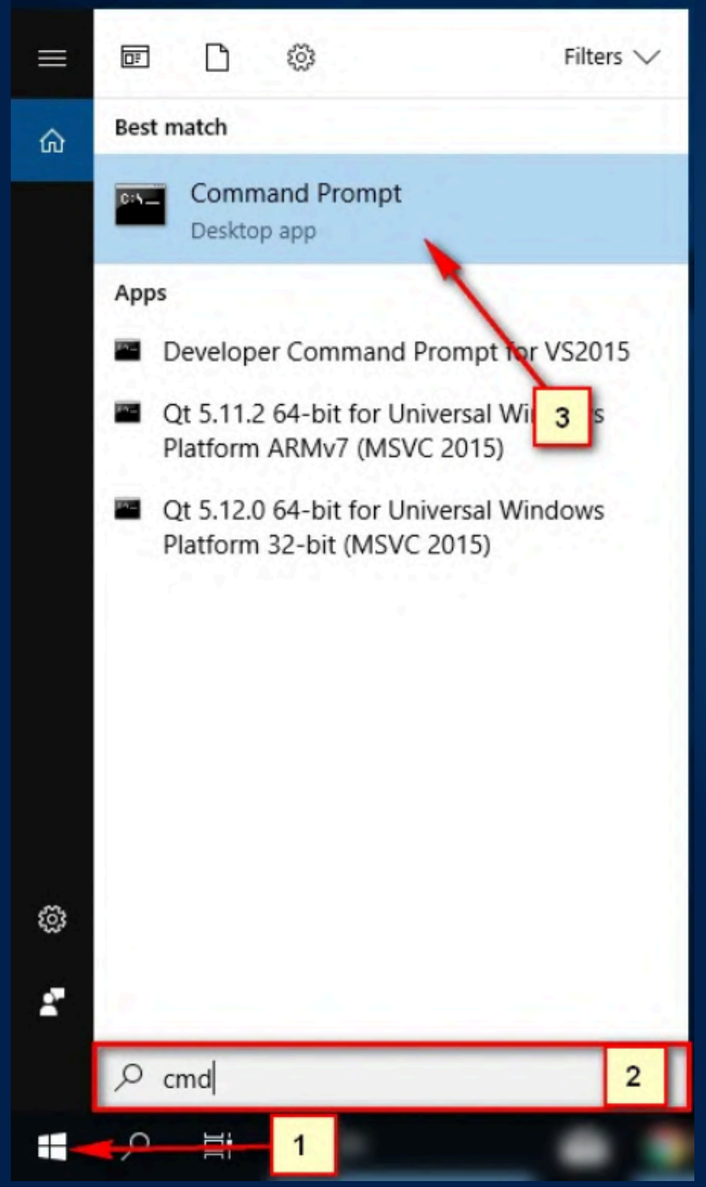


Hand Count Finger Send To NodeRed

Q2 install NodeRed

2. Check Version Node.JS with CMD

#1. Press the start menu button
#2.Type cmd to search
#3.Select Command Prompt



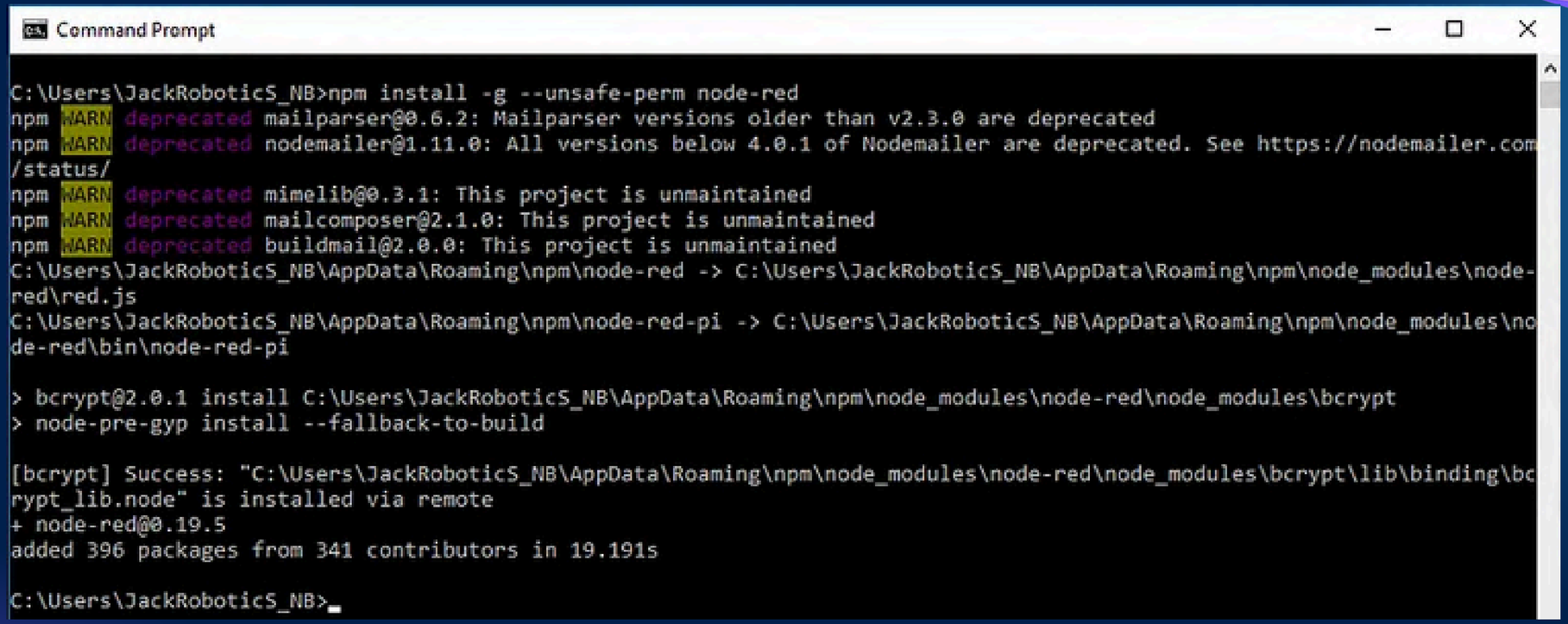
Node-RED

Hand Count Finger Send To NodeRed

Q2

```
npm install -g --unsafe-perm node-red
```

3. Install Node-Red with npm



When Node-Red is finished installing, it will show the screen as shown below. indicates that the installation is complete

Hand Count Finger Send To NodeRed

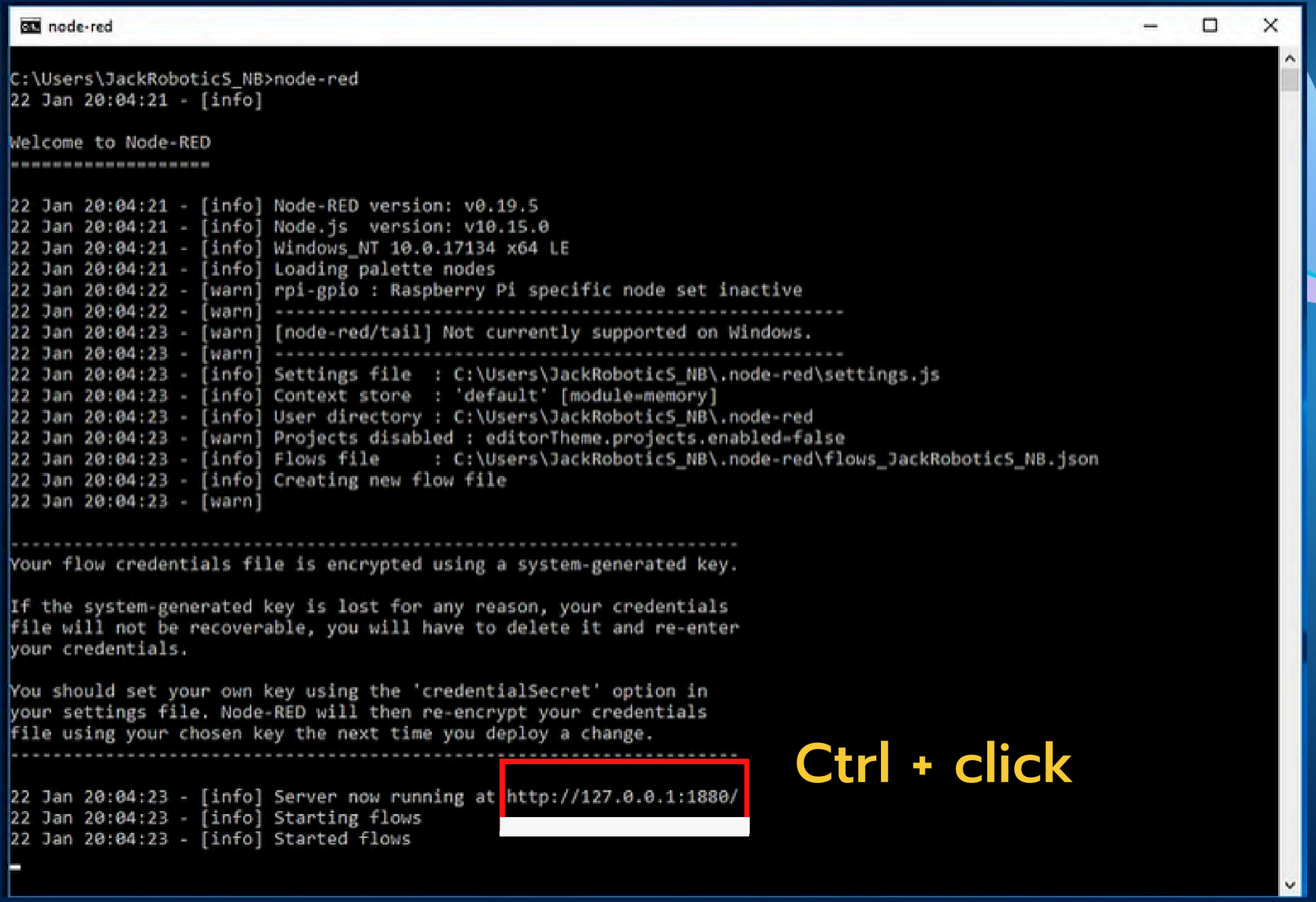
Q2 install NodeRed

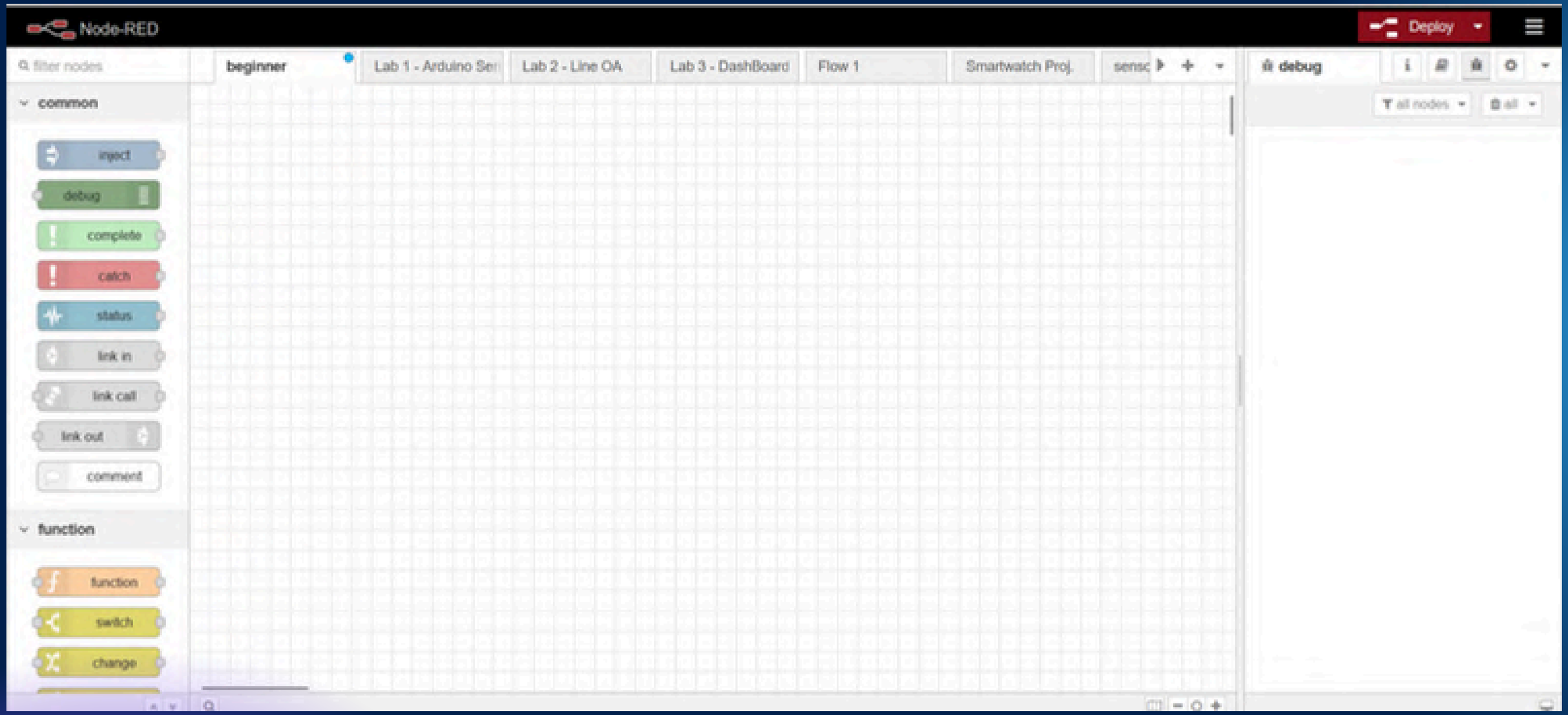
4. Run Node-Red

```
node-red
```

Type the command node-red

into cmd, it will look like the picture.

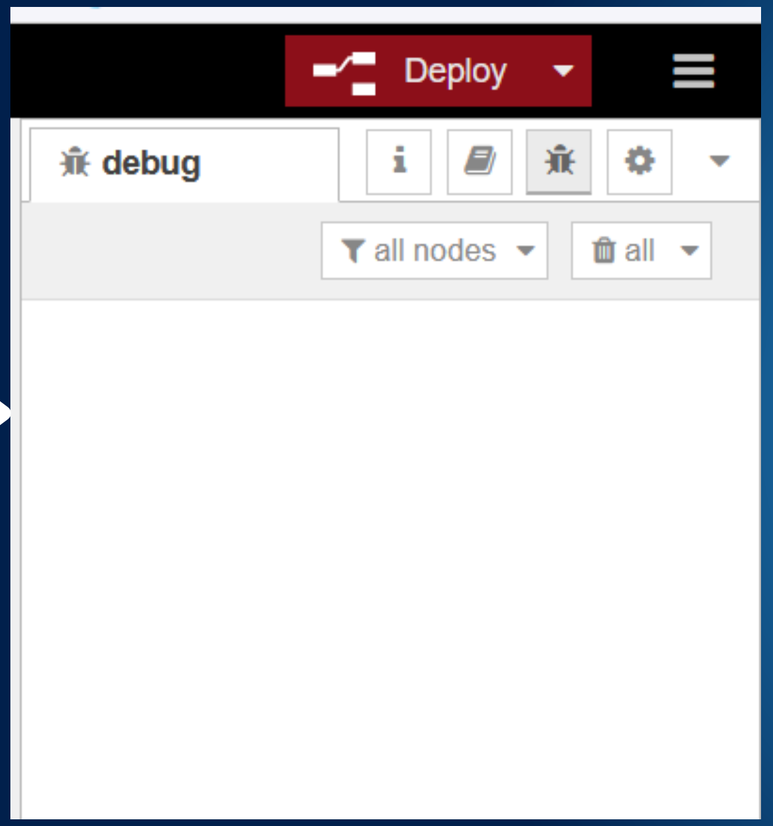




Workspace **Node-Red**

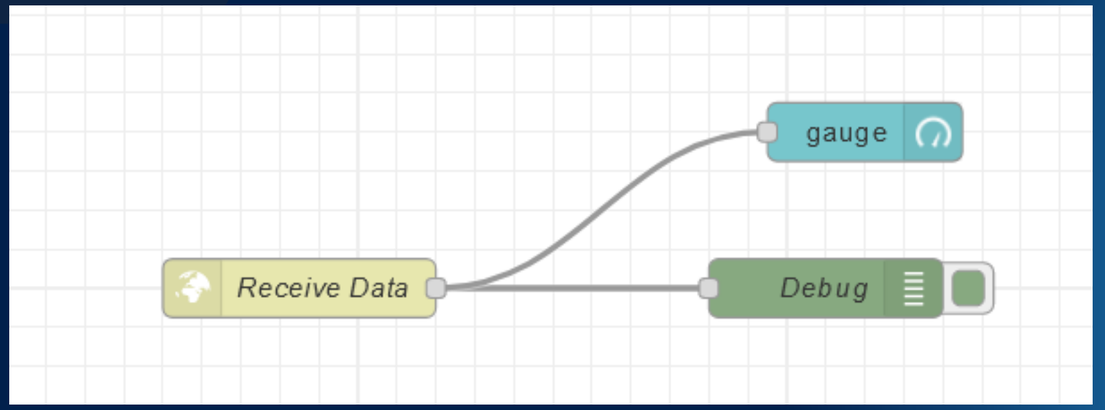
Deploy Button

- ปุ่มนี้ไว้กดทุกครั้งเวลา อัพเดท block code หลังแก้ไขเสร็จ

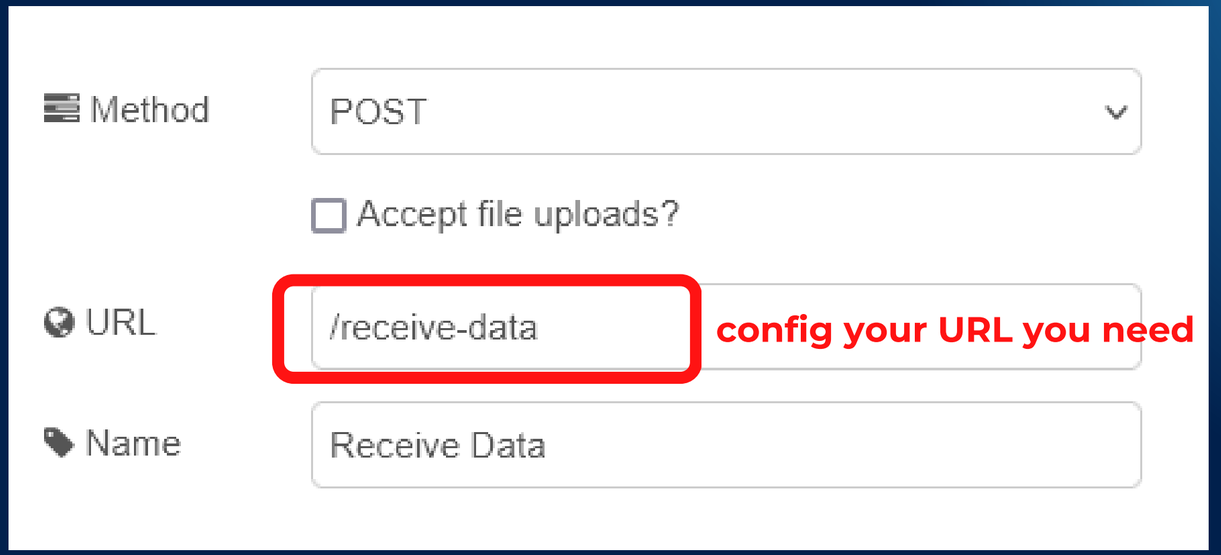


นำไว้ดู Debug เวลามีค่าเข้ามาจาก การส่งข้อมูล Data Value / Struct after processing

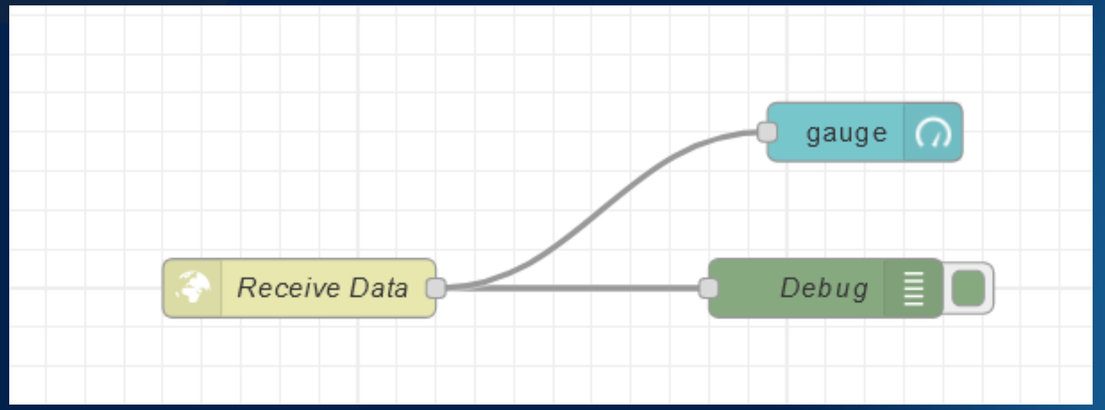
Q3 Configure padlet in node red



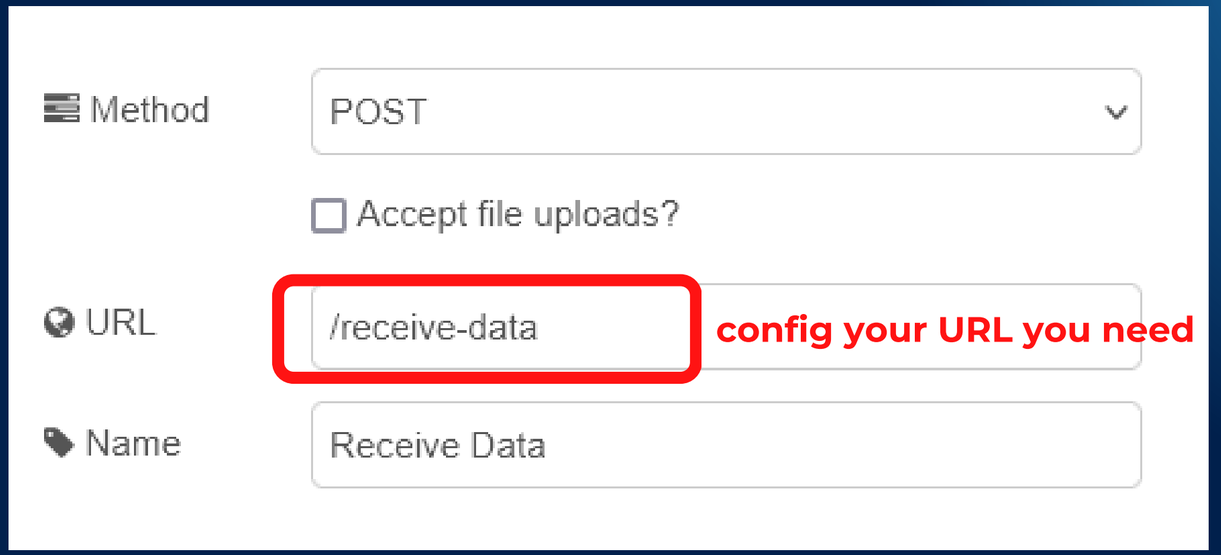
double-click in nodes Receive Data



Q3 Configure padlet in node red



double-click in nodes Receive Data



Q3 Hand Count Coding

Our Wisdom

URL

```python
import cv2
import mediapipe as mp
import requests
url = 'http://127.0.0.1:188(/receive-data'

mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
mp_hands = mp.solutions.hands

capture = cv2.VideoCapture(0)
with mp_hands.Hands(
  model_complexity=0,
  min_detection_confidence=0.5,
  min_tracking_confidence=0.5) as hands:
  while capture.isOpened():
    success, image = capture.read()
    if not success:
      print('Ignored empty webcam\'s frame')
      continue
    image.flags.writeable = False
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = hands.process(image)

    image.flags.writeable = True
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    fingerCount = 0
```

```python
    if results.multi_hand_landmarks:
      for hand_landmarks in results.multi_hand_landmarks:
        handIndex = results.multi_hand_landmarks.index(hand_landmarks)
        handLabel = results.multi_handedness[handIndex].classification[0].label

        handLandmarks = []

        for landmarks in hand_landmarks.landmark:
          handLandmarks.append([landmarks.x, landmarks.y])

        if handLabel == "Left" and handLandmarks[4][0] > handLandmarks[3][0]:
          fingerCount = fingerCount + 1
        elif handLabel == "Right" and handLandmarks[4][0] < handLandmarks[3][0]:
          fingerCount = fingerCount + 1

        if handLandmarks[8][1] < handLandmarks[6][1]:
          fingerCount = fingerCount + 1
        if handLandmarks[12][1] < handLandmarks[10][1]:
          fingerCount = fingerCount + 1
        if handLandmarks[16][1] < handLandmarks[14][1]:
          fingerCount = fingerCount + 1
        if handLandmarks[20][1] < handLandmarks[18][1]:
          fingerCount = fingerCount + 1

        mp_drawing.draw_landmarks(
          image,
          hand_landmarks,
          mp_hands.HAND_CONNECTIONS,
          mp_drawing_styles.get_default_hand_landmarks_style(),
          mp_drawing_styles.get_default_hand_connections_style()
        )
```

```python
    cv2.putText(image, str(fingerCount), (50,450), cv2.FONT_HERSHEY_COMPLEX_SMALL, 3, (255,0,0), 10)
    cv2.imshow('FingerCounting Apps',image)
```

youtube : https://www.youtube.com/watch?app=desktop&v=1iq9FxLxBIY

Q4

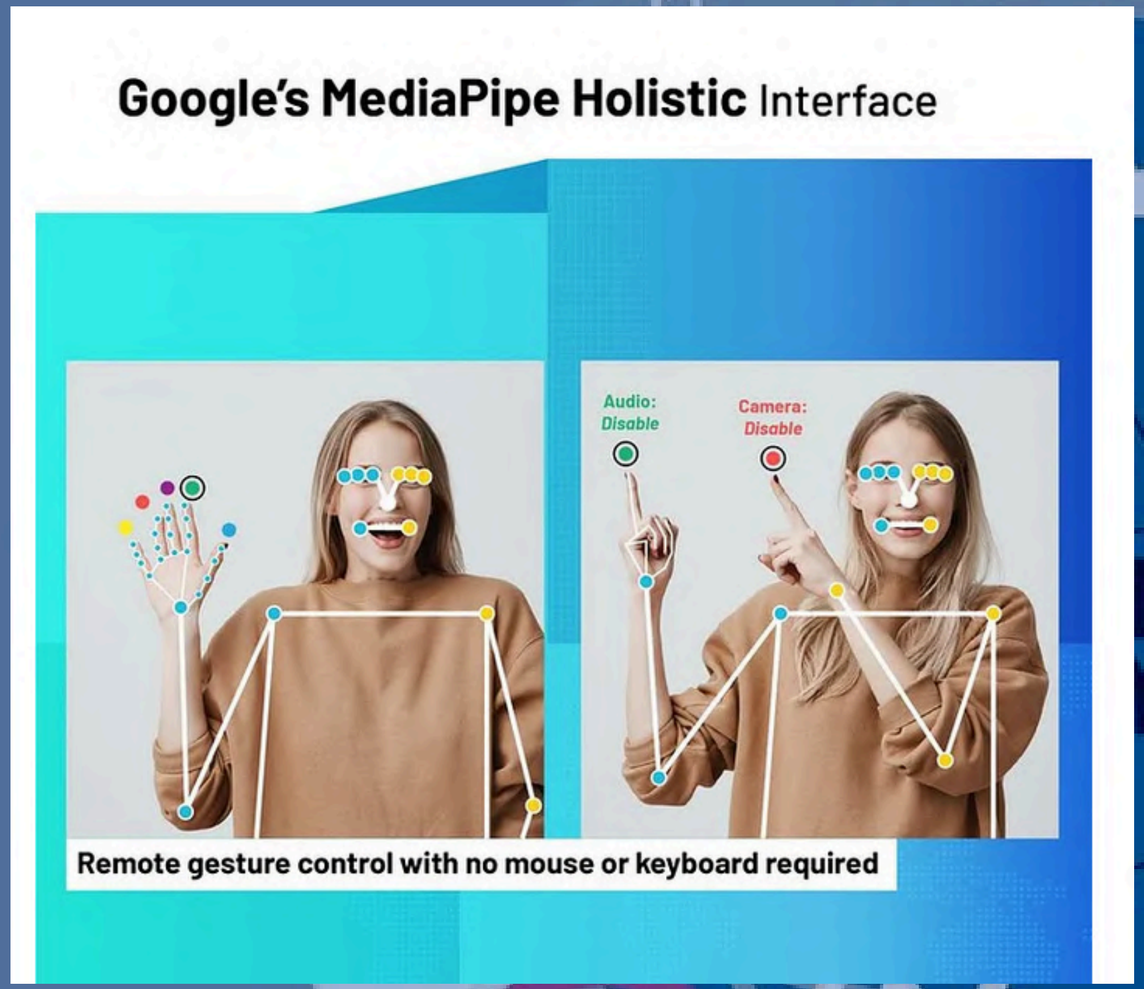
FingerCount Send To NodeRed Coding

youtube : https://www.youtube.com/watch?app=desktop&v=1iq9FxLxBIY

```
    if cv2.waitKey(1) == 27: # Check if the ASCII value of the pressed key is 27 (ESC key)
        break
    if cv2.waitKey(1) & 0xFF == ord('c'): # Check if the ASCII value of the pressed key is 99 (C key) // 32 is (backspace key)
      # send http buffer string to http in Node-red not recall.
      try:
          response = requests.post(url, data=str(fingerCount), timeout=5)  # Set timeout to 5 seconds
          if response.status_code == 200:
              print('Data sent successfully')
          else:
              print('Error sending data:', response.text)
      except requests.Timeout:
          print('Request timed out. Server did not respond in time.')
      except requests.RequestException as e:
          print('An error occurred:', e)
capture.release()
```

github code : https://github.com/ThedBoyZ/CodeBotix_Camp_Image

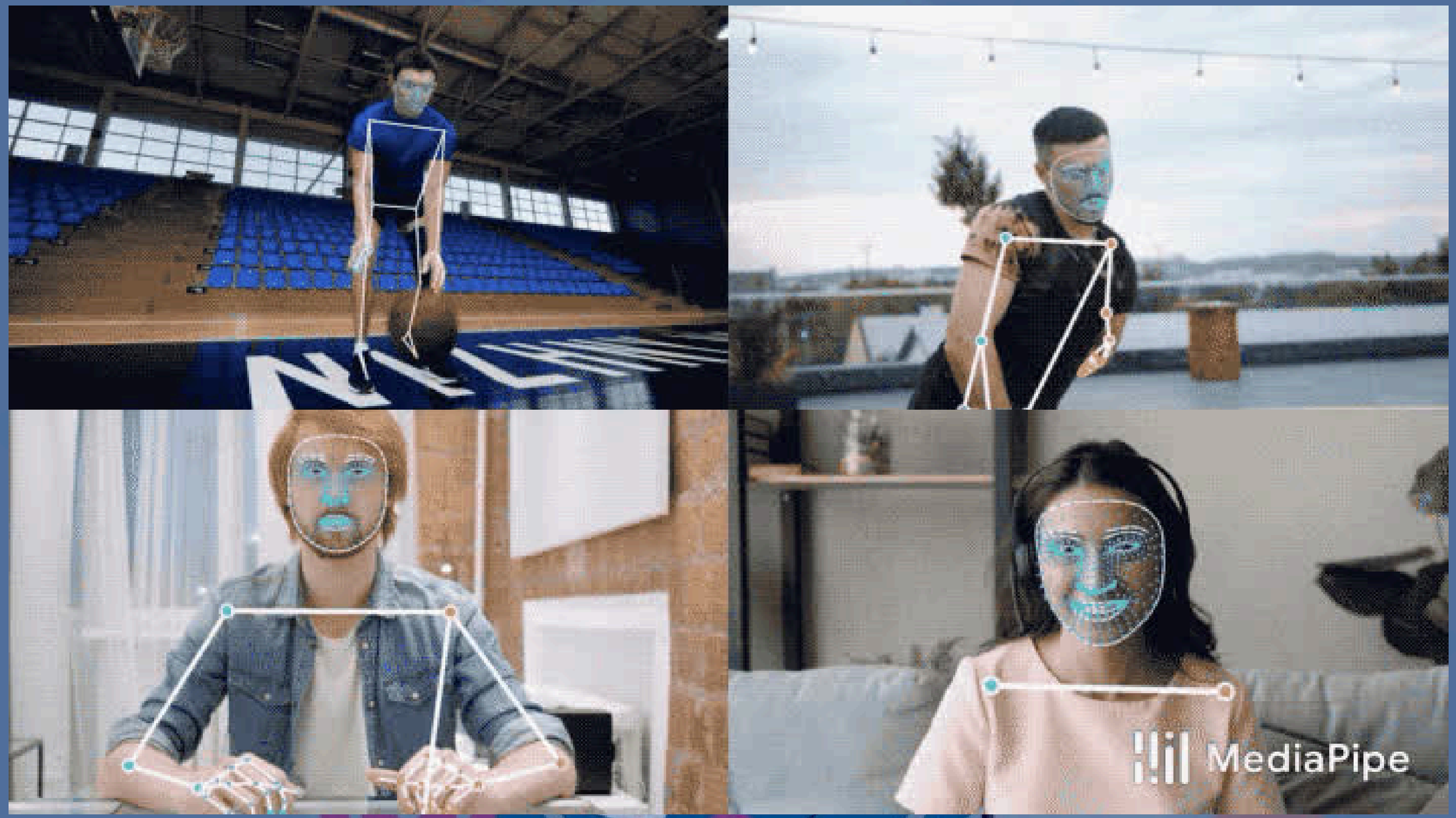
Intelligence



This gesture control and command technique will take us to the next level. And allow us to use new innovations that other devices cannot do

It can be tried on MediaPipe Holistic Interface from Google at:

https://mediapipe.dev/demo/holistic_remote/



https://mediapipe.dev/demo/holistic_remote/

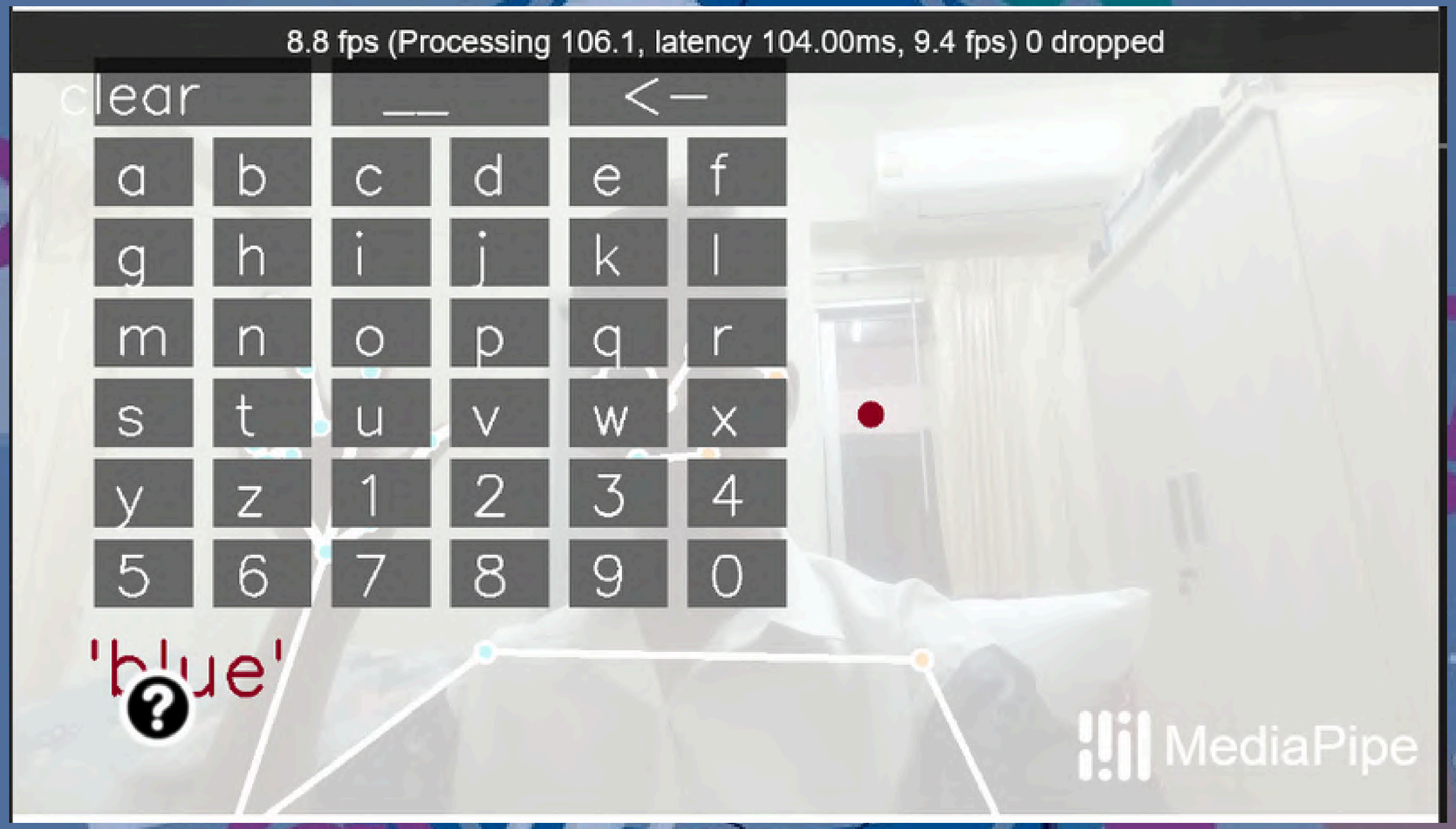
Holistic demo

1 1 นิ้ว มือซ้าย หรือ ขวา

2 3 นิ้ว มือซ้าย

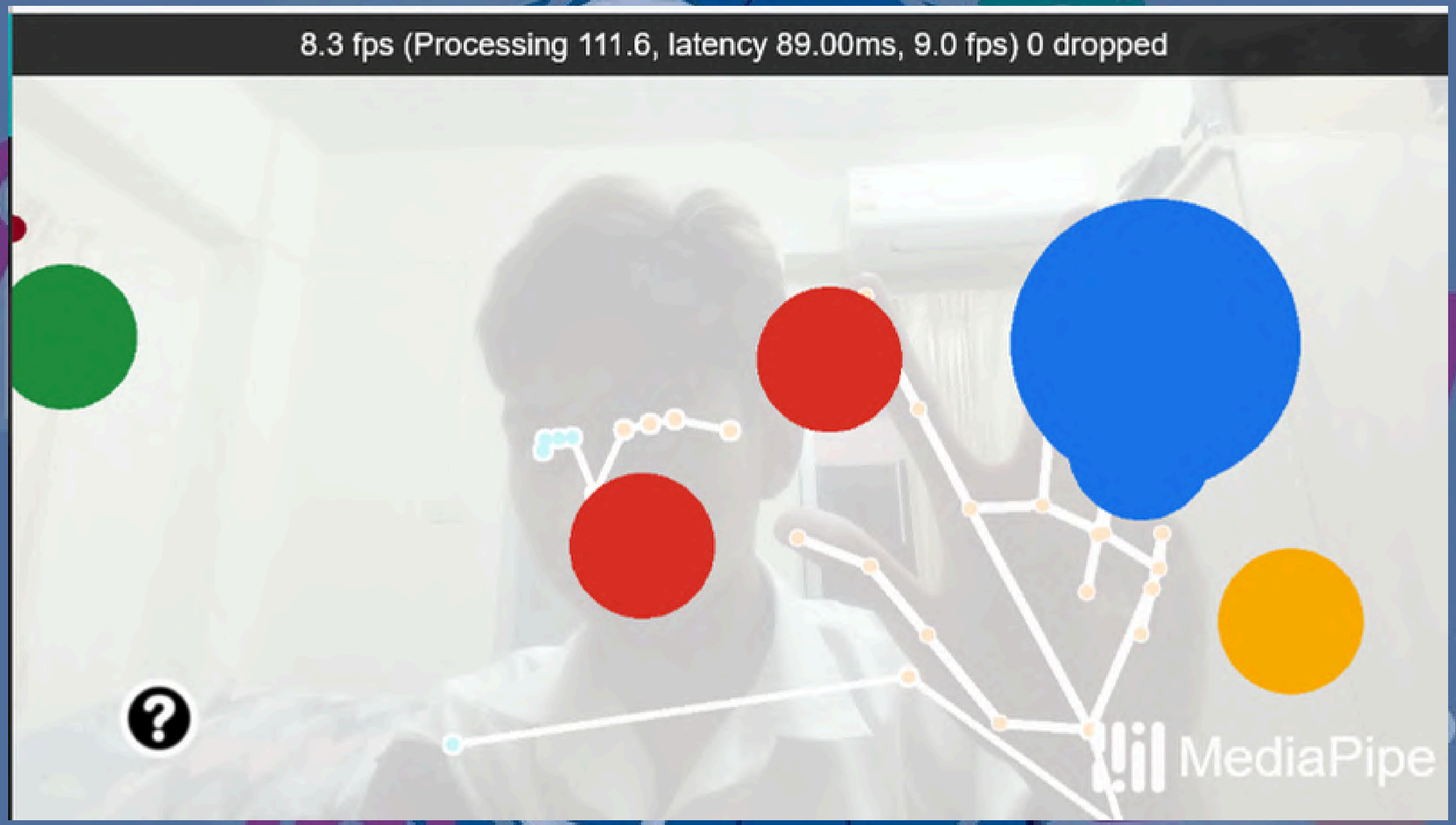
3 3 นิ้ว มือขวา

Holistic demo



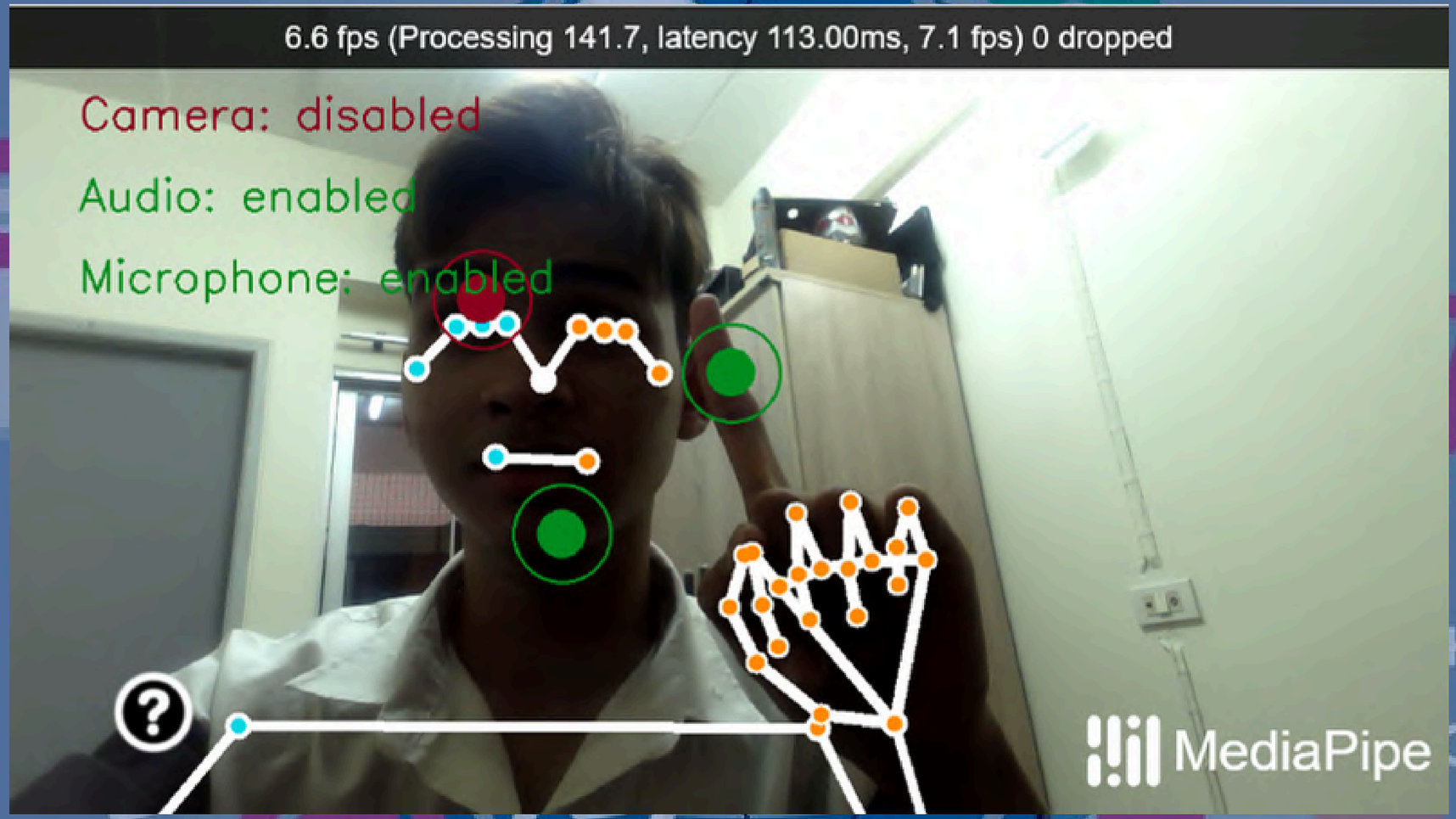
ลองเขียนชื่อตัวเอง เป็นภาษาอังกฤษ

Holistic demo



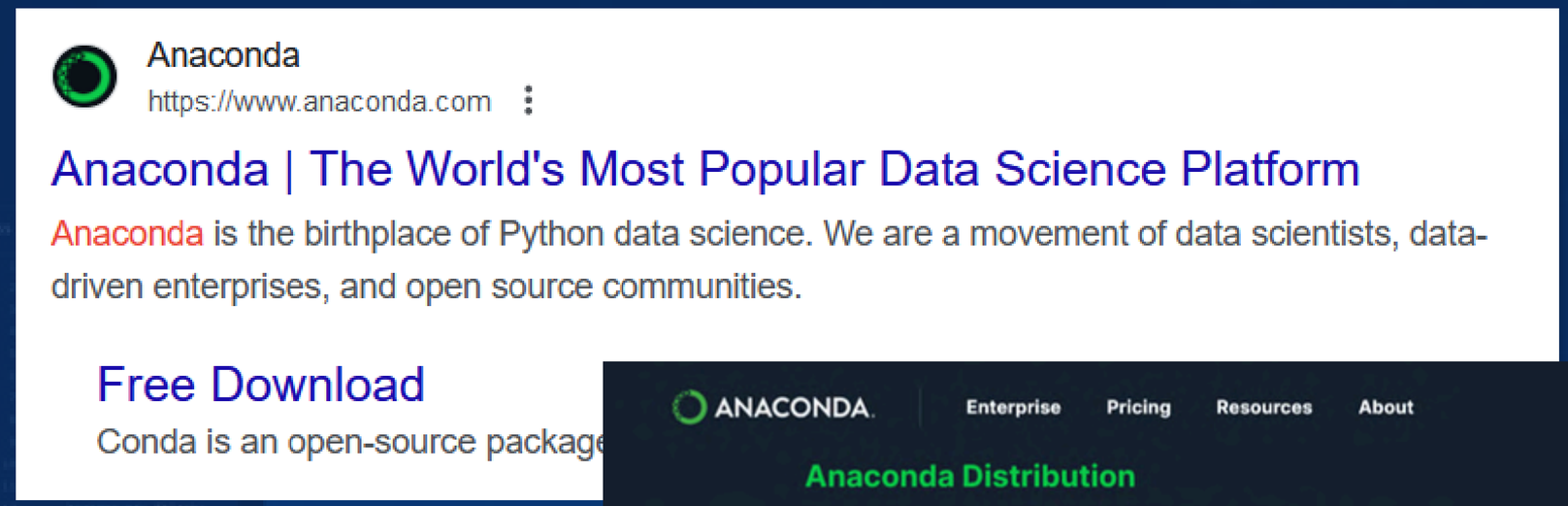
สามารถใช้มือแทน เมาส์ในการลากกรุ๊ปสีจัดหมวดหมู่ได้

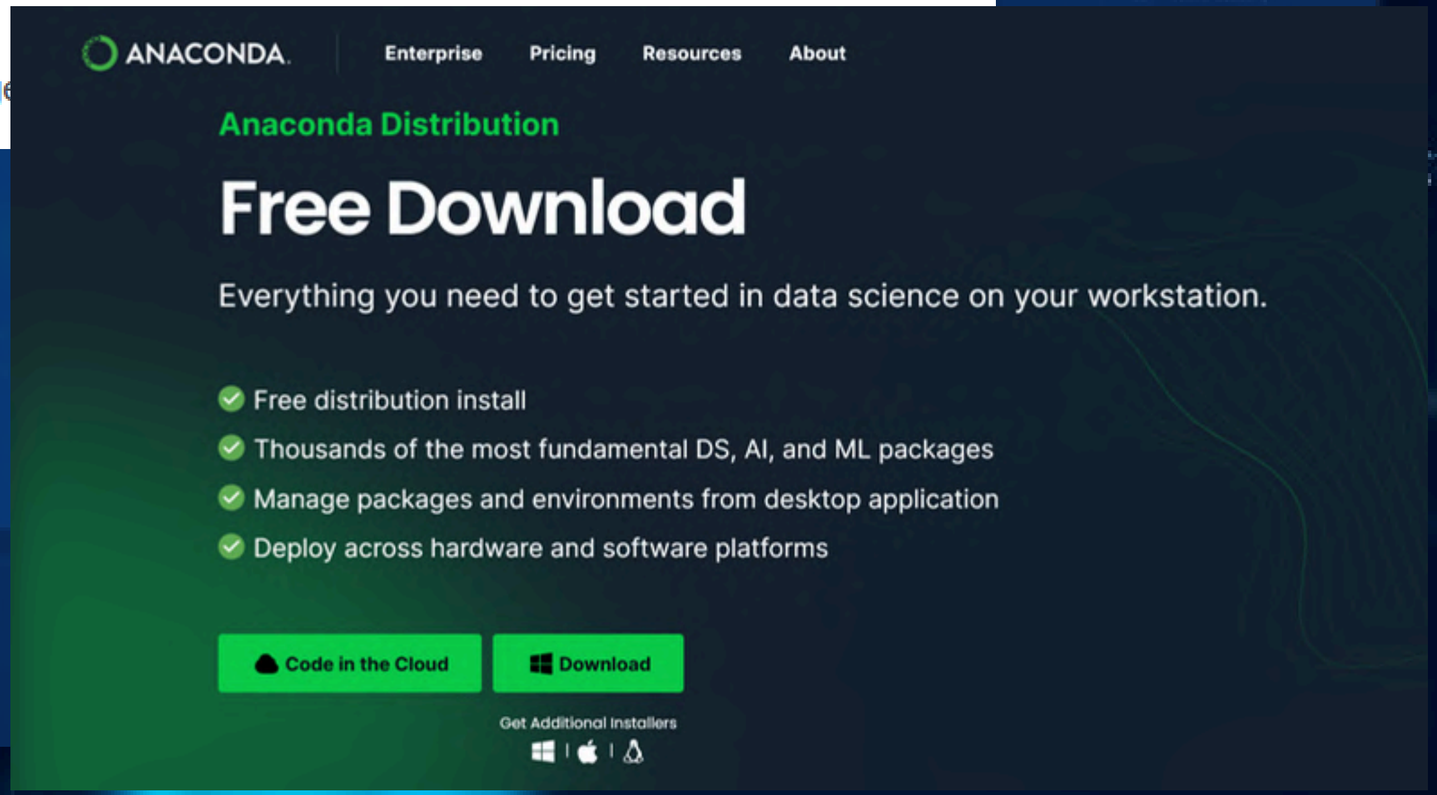
Holistic demo



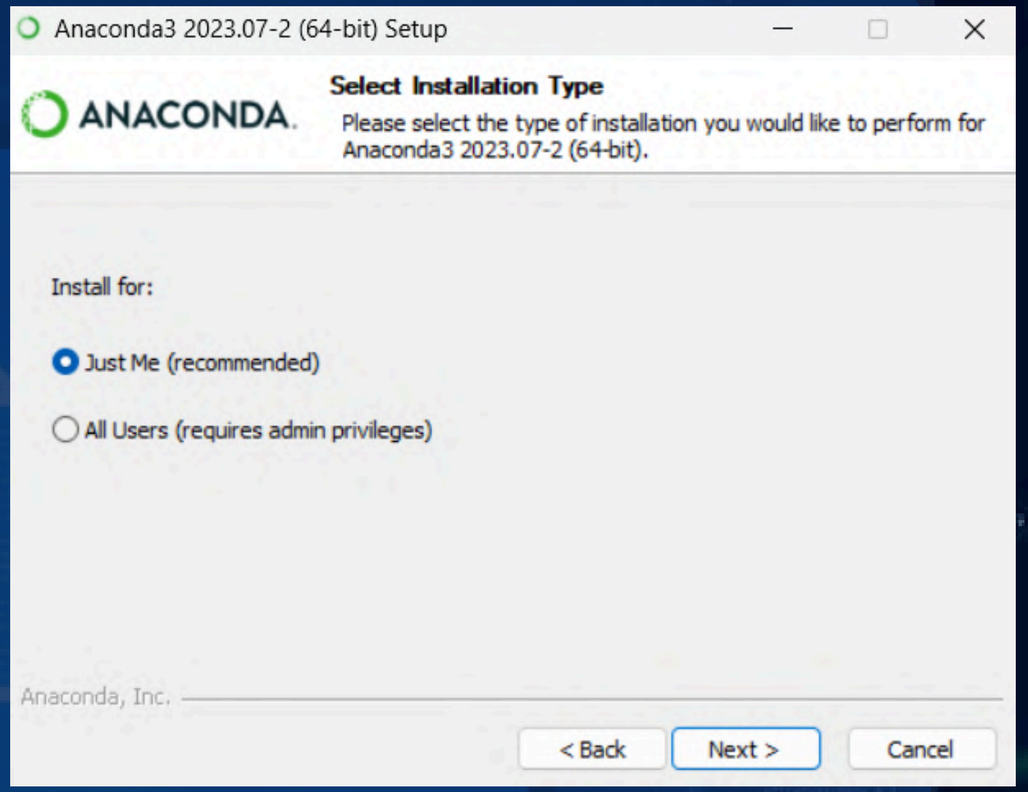
Disable / Enable mode

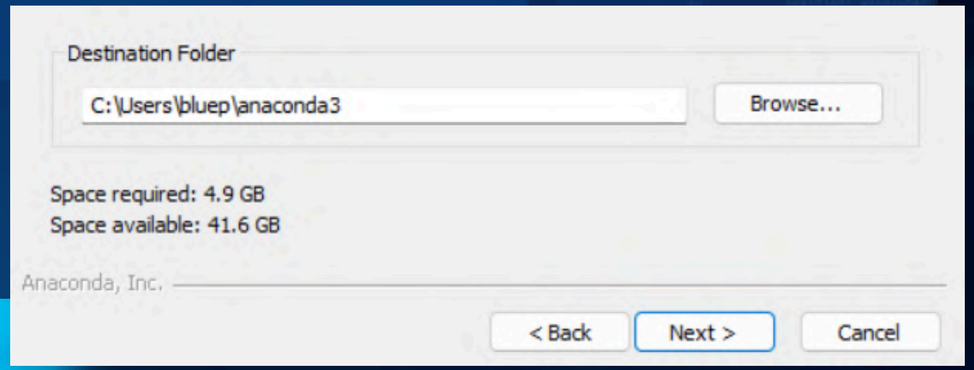
Anaconda Navigator



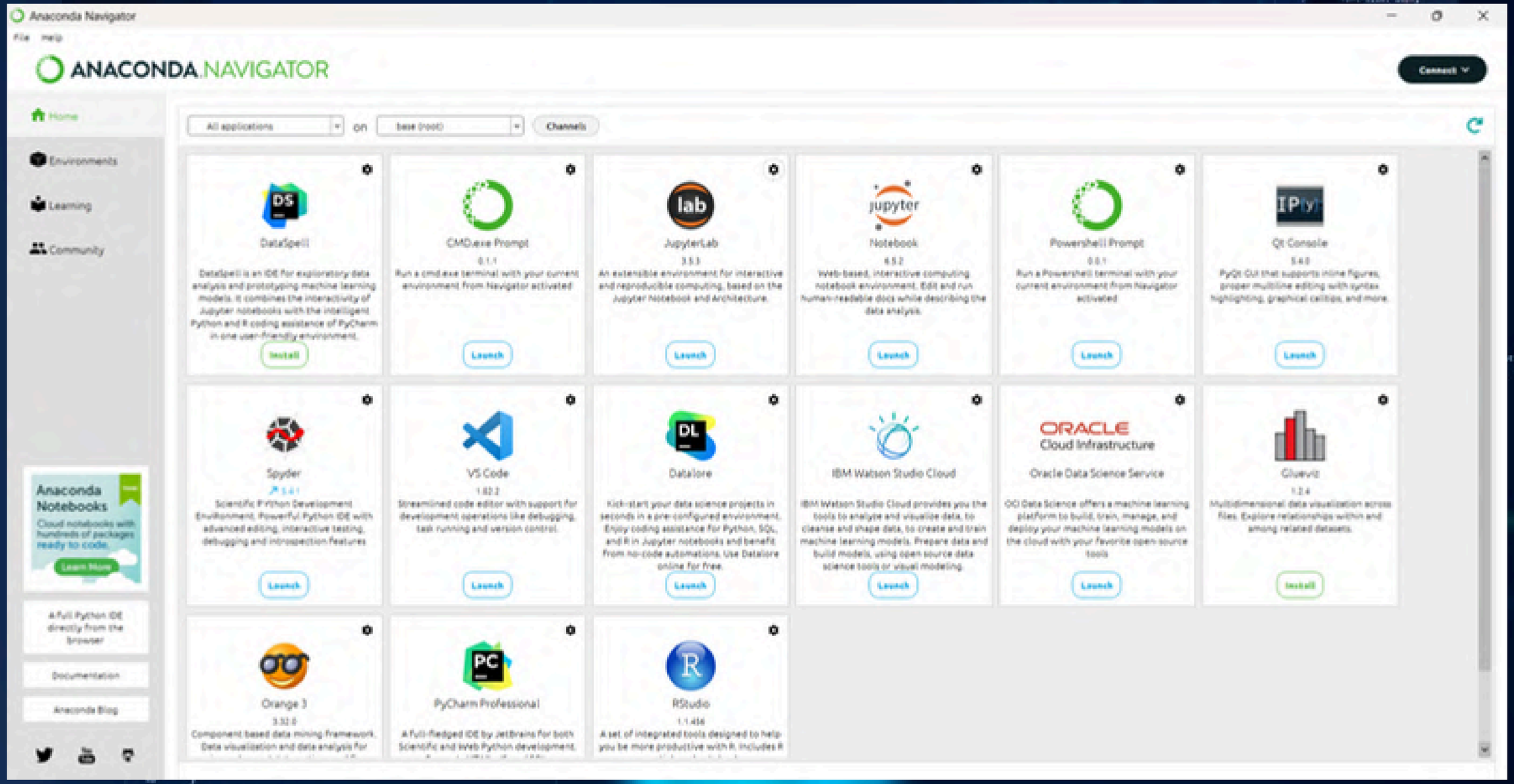


Anaconda Navigator

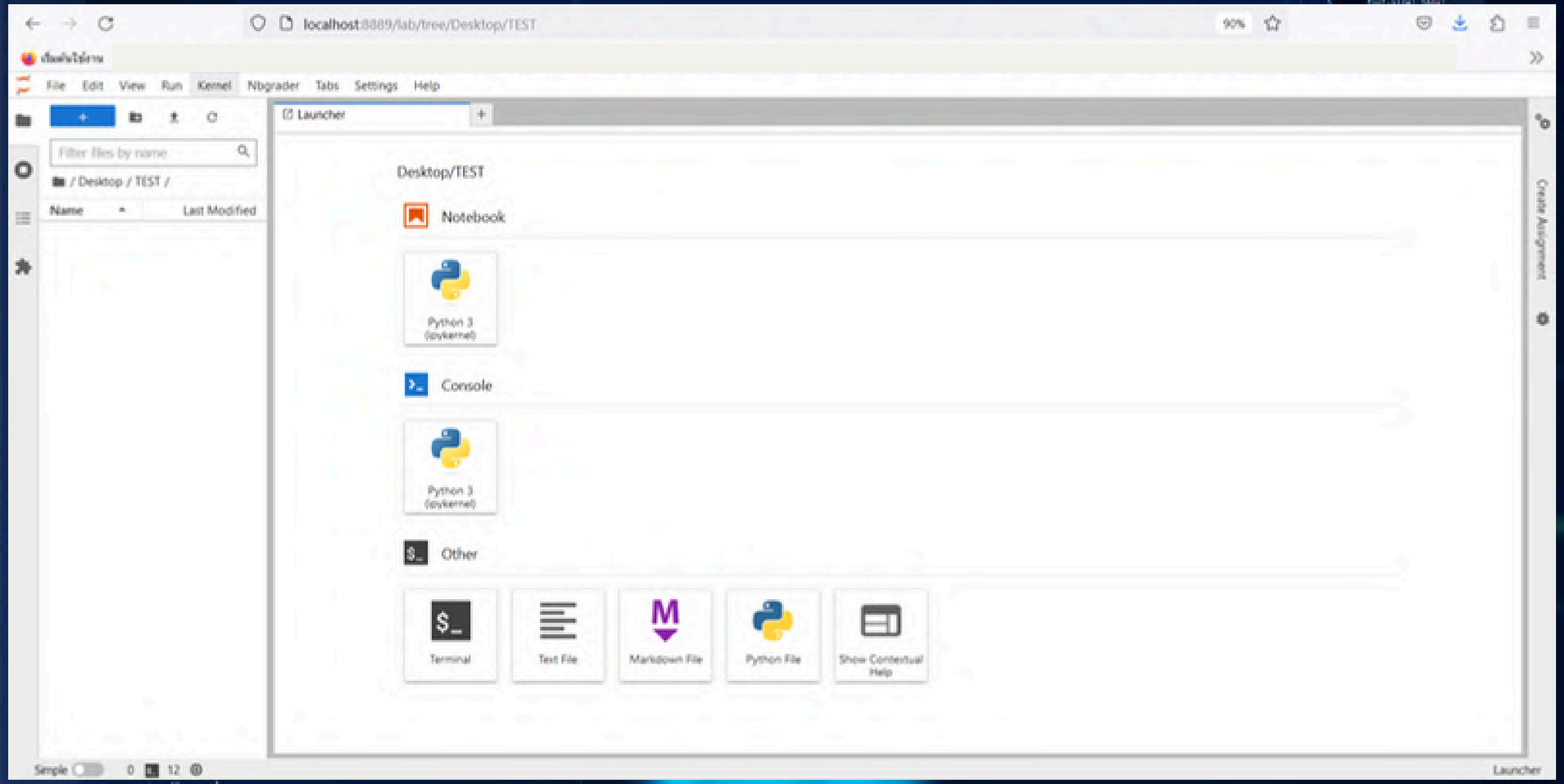




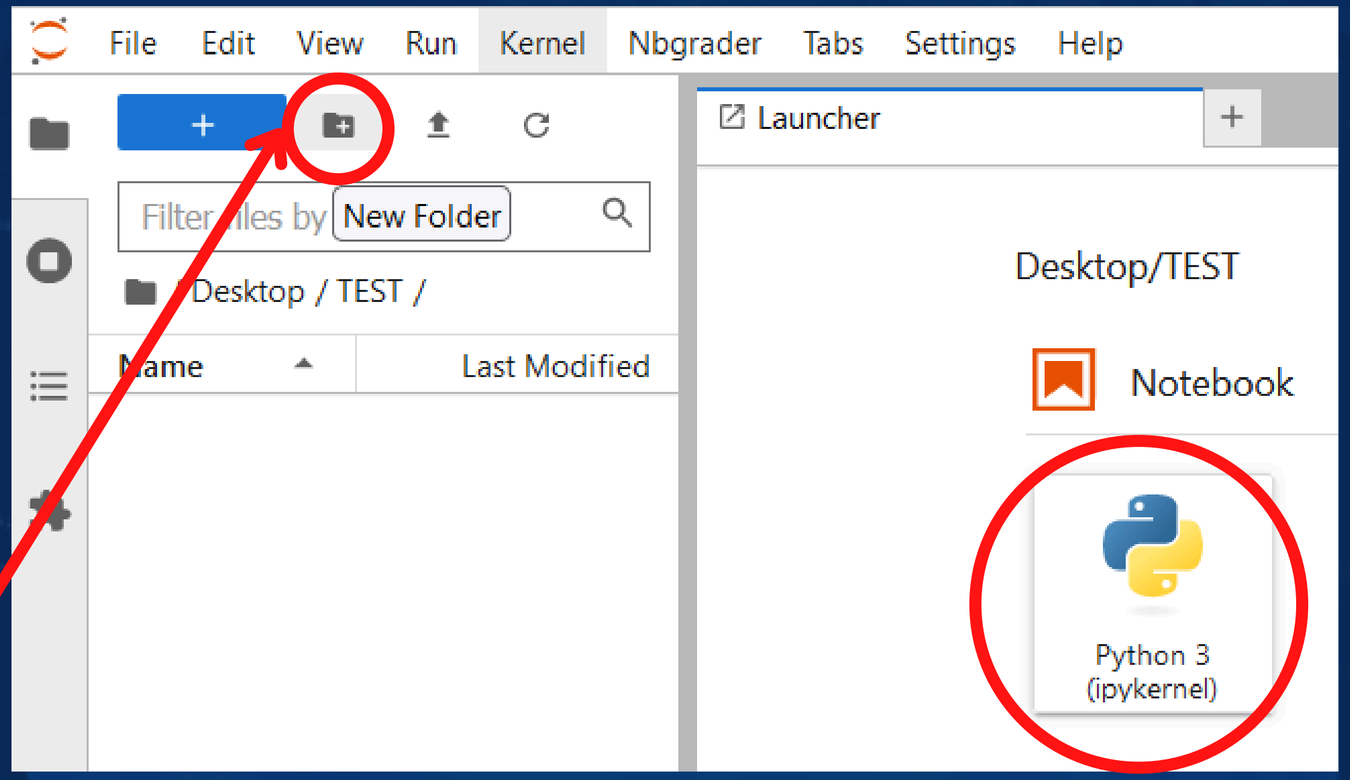
Anaconda Navigator



Anaconda Navigator



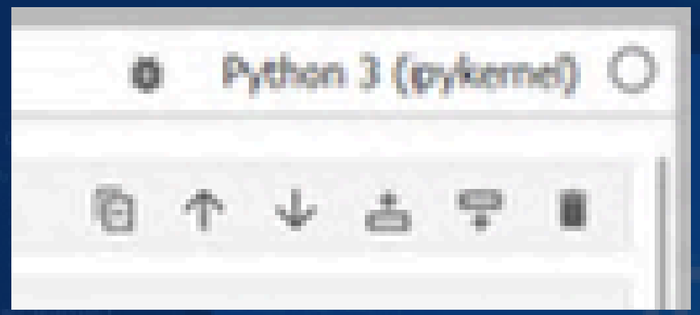
Anaconda Navigator



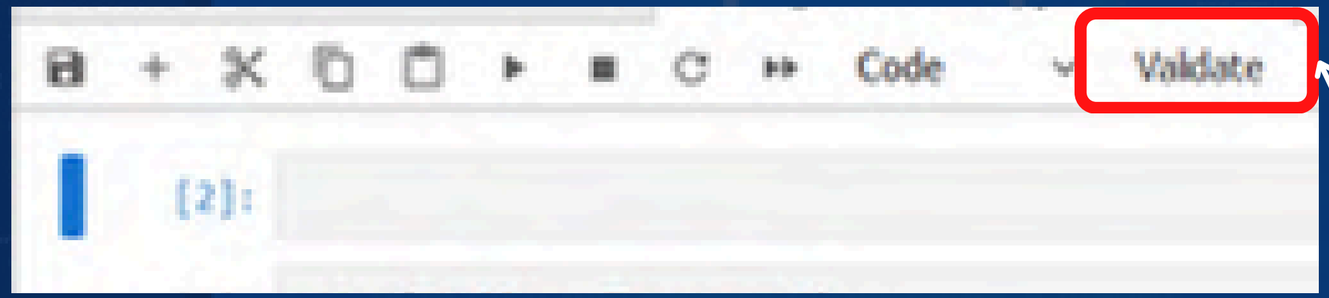
สร้าง Floder ใหม่

สร้าง สร้างไฟล์ .ipynb เพื่อเขียน
โปรแกรมใน jupyter notebook

Anaconda Navigator



python 3 (ipykernel) ในการรันโค้ด



ไว้สำหรับการทดสอบ save ไฟล์
ตรวจสอบความถูกต้องทั้งหมด

Jupyter Notebook

image processing

Jupyter Notebook

image processing

```
[5]: print("Hello , Theeramet")
```

```
Hello , Theeramet
```

ทดสอบ test jupyter Lab ¶

```
[6]: a = 4
```

```
[7]: b = 2
```

part 1

```
[9]: c = a // b
     c
```

```
[9]: 2
```

```
[11]: c = a * b
      c
```

```
[11]: 8
```

part2

```
[12]: print(c)
```

```
8
```

```
[13]: b = 10
```

```
[14]: b
```

```
[14]: 10
```

Jupyter Notebook

image processing

Jupyter Notebook

image processing

```
[1]: 2 ** 5
```

```
[1]: 32
```

comment

```
[2]: n = 20.5
     #n = 20.5
```

Data types

```
[4]: a = 3
     type(a)
```

```
[4]: int
```

```
[5]: b = 5.0
     type(b)
```

```
[5]: float
```

```
[7]: c = '4'
     type(c)
```

```
[7]: str
```

```
[13]: d = eval(c)
      type(d)
```

```
[13]: str
```

```
[16]: name = 'theeramet'
      sur  = ' chuaipayung'
      name + sur
```

```
[16]: 'theeramet chuaipayung'
```

Jupyter Notebook

image processing

print

[17]:
```
v = 2
sc = 3.673245
name = "Blue"
```

[18]:
```
print("Hi, %s a = %d b = %.2f" % (name, n, sc))
```

```
Hi, Blue a = 20 b = 3.67
```

[19]:
```
print('Hi {} {:.1f}' .format(name , sc))
```

```
Hi Blue 3.7
```

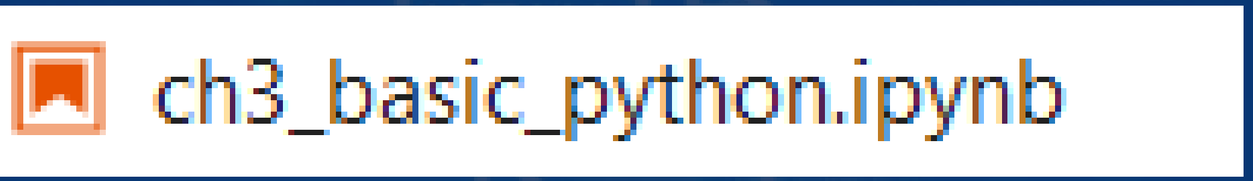
[21]:
```
b = 280903
print('data = {:,.2f}' .format(b))
```

```
data = 280,903.00
```

[ ]:

Jupyter Notebook

image processing



Jupyter Notebook

image processing

Python เบื้องต้น

[3]:
```
name = 'theeramet'
name = 'scimath'
print(name)
```

```
scimath
```

[ ]:
```
เครื่องหมายคำนวณ (Operators)
+ - * /
%   หารเอาเศษ
//  หารปัดเศษ
**  ยกกำลัง
```

[5]:
```
a = 10
```

[8]:
```
b = a / 2
b
```

[8]:
```
5.0
```

[10]:
```
5 // 2
```

[10]:
```
2
```

[11]:
```
5 % 2
```

[11]:
```
1
```

[12]:
```
2 ** 3
```

[12]:
```
8
```

Jupyter Notebook

image processing

Variable

[16]:
```
ราคา = 100
สุทธิ = ราคา / 4
ผลลัพธ์ = สุทธิ
print(สุทธิ)
```

```
25.0
```

[ ]:
```
2r   = 4
pass = 9
```

[23]:
```
r     = 3
pass_ = 50
```

Condition

[ ]:
```
>  <  >=    <=   !=
```

[25]:
```
a = 3
b = 7
c = 9
```

[26]:
```
a > b
```

[26]:
```
False
```

[27]:
```
a < b
```

[27]:
```
True
```

[28]:
```
a < b and a > c
```

[28]:
```
False
```

Jupyter Notebook

image processing

if

[31]:
```
img = 32
if img < 30 :
    print('Image off')
elif img >= 30 and img < 40:
    print('Image Process')
else:
    print('Turn on a Visualize Box')
```

```
Image Process
```

Iteration (Loop):

[35]:
```
for count in range(4,6):
    print("number = ",end="")
    print(count)
print()
print("done")

#print
print('data = ',a)
```

```
number = 4
number = 5

done
data =  3
```

Jupyter Notebook

image processing

while

```
[ ]: interest = 0
while interest < 4:
    print("hello naa")
    interest = interest + 1
print("end of sol")
```

```
hello naa
hello naa
hello naa
hello naa
end of sol
```

Function

```
[41]: def area(length):
    area = length * length
    return area

a = area(4)
print('Area = %d' % (a))
print('Area = ', a)
```

```
Area = 16
Area =  16
```

```
[42]: area(5)
```

```
[42]: 25
```

```
[43]: area(6)
```

```
[43]: 36
```

Jupyter Notebook

image processing

Container

List

```
[44]: personality = [20, 3, 19.5, 230, 145]
```

```
[45]: type(personality)
```

```
[45]: list
```

```
[47]: personality[0]
```

```
[47]: 20
```

```
[49]: personality[-4:-1]
```

```
[49]: [3, 19.5, 230]
```

```
[51]: for j in personality:
          print(j)
```

```
20
3
19.5
230
145
```

```
[52]: # enumerate
      for i,n in enumerate(personality):
          print(i,n)
```

```
0 20
1 3
2 19.5
3 230
4 145
```

Jupyter Notebook

image processing

```
[53]: data = [[20, 62, 22.4, 189, 123],
              [32, 47, 11.9, 168, 174],
              [24, 35, 51.43, 155, 144]]
```

```
[54]: data[0][1]
```

```
[54]: 62
```

```
[59]: data[2][3]
```

```
[59]: 155
```

Dictionary

```
[60]: book_lish = {'age':27,'ht':171,'bmi':18.4}
      type(book_lish)
```

```
[60]: dict
```

```
[62]: a = book_lish['ht']
      a
```

```
[62]: 171
```

```
[63]: y = book_lish['bmi']
      y
```

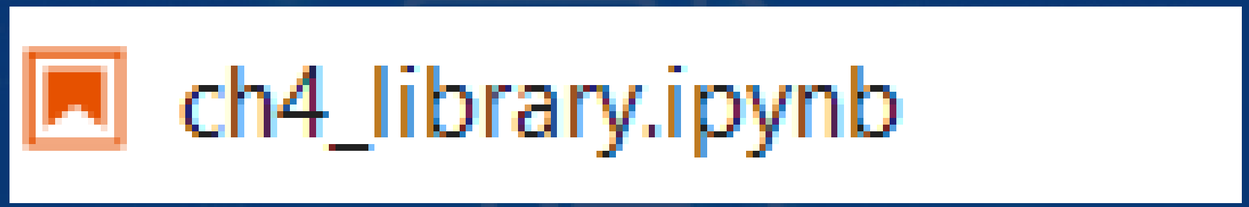
```
[63]: 18.4
```

```
[64]: for i in book_lish.values():
          print(i)
```

```
27
171
18.4
```

Jupyter Notebook

image processing



Jupyter Notebook

image processing

Library

การจัดการ Library

- Anaconda Navigator -pip (Package Installer for Python): https://pip.pypa.io/en/stable/
- conda: https://docs.conda.io/projects/conda/en/latest/commands.html

ตรวจสอบว่าติดตั้งแล้วหรือยัง

[2]:
```
!pip show pandas
```

```
Name: pandas
Version: 1.5.3
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: https://pandas.pydata.org
Author: The Pandas Development Team
Author-email: pandas-dev@python.org
License: BSD-3-Clause
Location: c:\users\bluep\anaconda3\lib\site-packages
Requires: numpy, python-dateutil, pytz
Required-by: datashader, holoviews, hvplot, seaborn, statsmodels, xarray
```

[3]:
```
!conda list pandas
```

```
# packages in environment at C:\Users\bluep\anaconda3:
#
# Name                    Version                   Build  Channel
pandas                    1.5.3           py310h4ed8f06_0
```

[4]:
```
!pip show matplotlib
```

```
Name: matplotlib
Version: 3.7.0
Summary: Python plotting package
Home-page: https://matplotlib.org
```

Jupyter Notebook

image processing

[5]:
```
!pip show mlxtend
```

```
WARNING: Package(s) not found: mlxtend
```

Pandas

read CSV

[3]:
```
import pandas as pd
```

[4]:
```
ds = pd.read_csv('data/advertise.csv')
```

[5]:
```
ds
```

[5]:

	Unnamed: 0	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9
...	...	...	...	...	...
195	196	38.2	3.7	13.8	7.6
196	197	94.2	4.9	8.1	9.7
197	198	177.0	9.3	6.4	12.8
198	199	283.6	42.0	66.2	25.5
199	200	232.1	8.6	8.7	13.4

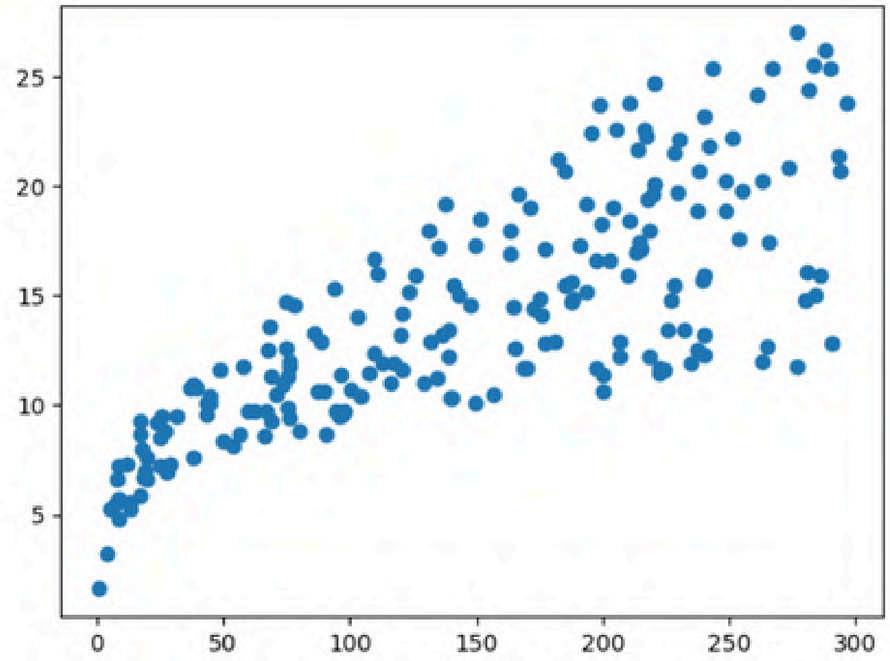
Jupyter Notebook

image processing

plot

[6]:
```
import matplotlib.pyplot as plt
```

[7]:
```
plt.scatter(ds.TV, ds.Sales)
plt.show()
```



[8]:
```
pip --version
```

```
pip 22.3.1 from C:\Users\bluep\anaconda3\lib\site-packages\pip (python 3.10)

Note: you may need to restart the kernel to use updated packages.
```

Jupyter Notebook

image processing

การจัดการ Library

[9]:
```
import pydotplus
```

pip (Package Installer for Python)

https://pip.pypa.io/en/stable/

[10]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\bluep\anaconda3\lib\site-packages (1.23.5)
Note: you may need to restart the kernel to use updated packages.
```

[11]:
```
!pip show mtcnn
```

```
WARNING: Package(s) not found: mtcnn
```

[12]:
```
!pip uninstall mtcnn
```

```
WARNING: Skipping mtcnn as it is not installed.
```

conda

[13]:
```
conda list pandas
```

```
# packages in environment at C:\Users\bluep\anaconda3:
#
# Name                    Version                   Build  Channel
pandas                    1.5.3           py310h4ed8f06_0

Note: you may need to restart the kernel to use updated packages.
```

[ ]:
```
!conda install numpy
```

[ ]:
```
# pip install mtcnn
```

Jupyter Notebook

image processing

```
[ ]: # pip install mtcnn
     !pip uninstall mtcnn
```

```
[ ]: !conda info

     # https://conda-forge.org/docs/user/introduction.html

     # https://conda.anaconda.org/conda-forge/
```

Numpy ¶

```
[ ]: import numpy as np
     data = [4, 7, 1, 2, 9, 3, 8, 16]
     np.mean(data)
```

```
[ ]: np.mean([4, 7, 1, 2, 9, 3, 8, 16])
```

```
[15]: np.std(data)
```

```
[15]: 4.575751304430781
```

```
[16]: np.min(data)
```

```
[16]: 1
```

```
[17]: np.max(data)
```

```
[17]: 16
```

```
[19]: np_a = np.array([data])
      type(np_a)
```

```
[19]: numpy.ndarray
```

Jupyter Notebook

image processing

```
[21]: U = [[6, 7, 4, 5, 1],
           [2, 8, 3, 6, 4],
           [1, 3, 2, 9, 6],
           [8, 9, 1, 7, 2]]

[30]: npu = np.array(U)
      npu

[30]: array([[6, 7, 4, 5, 1],
             [2, 8, 3, 6, 4],
             [1, 3, 2, 9, 6],
             [8, 9, 1, 7, 2]])

[31]: npu.shape

[31]: (4, 5)

[32]: npu[0,1]

[32]: 7

[33]: npu[2,4]

[33]: 6

[34]: npu[:,:]

[34]: array([[6, 7, 4, 5, 1],
             [2, 8, 3, 6, 4],
             [1, 3, 2, 9, 6],
             [8, 9, 1, 7, 2]])

[35]: npu.T

[35]: array([[6, 2, 1, 8],
             [7, 8, 3, 9],
             [4, 3, 2, 1],
```

Jupyter Notebook

image processing

```
[35]: array([[6, 2, 1, 8],
             [7, 8, 3, 9],
             [4, 3, 2, 1],
             [5, 6, 9, 7],
             [1, 4, 6, 2]])

[ ]: npu.reshape(1,-1)

[36]: array([[6, 7, 4, 5, 1, 2, 8, 3, 6, 4, 1, 3, 2, 9, 6, 8, 9, 1, 7, 2]])

[37]: npu.reshape(-1,1)

[37]: array([[6],
             [7],
             [4],
             [5],
             [1],
             [2],
             [8],
             [3],
             [6],
             [4],
             [1],
             [3],
             [2],
             [9],
             [6],
             [8],
             [9],
             [1],
             [7],
             [2]])
```

Jupyter Notebook

image processing

Random and unique

[38]:
```
np.random.rand(4)
```

[38]:
```
array([0.85592474, 0.00815198, 0.04118109, 0.78303578])
```

[40]:
```
mu, sigma = 3.2, 0.5
s = np.random.normal(mu, sigma, 1000)
```

[ ]:

Jupyter Notebook

image processing

Jupyter Notebook

image processing

Matplotlib

ใช้ร่วมกันกับ pandas

[2]:
```
!conda list matplotlib
```

```
# packages in environment at C:\Users\bluep\anaconda3:
#
# Name                    Version                   Build  Channel
matplotlib                3.7.0           py310haa95532_0
matplotlib-base           3.7.0           py310h4ed8f06_0
matplotlib-inline         0.1.6           py310haa95532_0
```

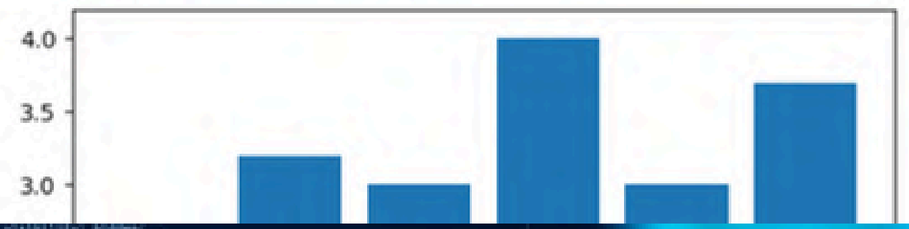
[5]:
```
import matplotlib.pyplot as plt
```

from matplotlib import pyplot as plt # เขียนแบบนี้ก็ได้

Bar

[6]:
```
x = [1,2,3,4,5,6]
y_data = [1,3.2,3.0,4,3,3.7]
```

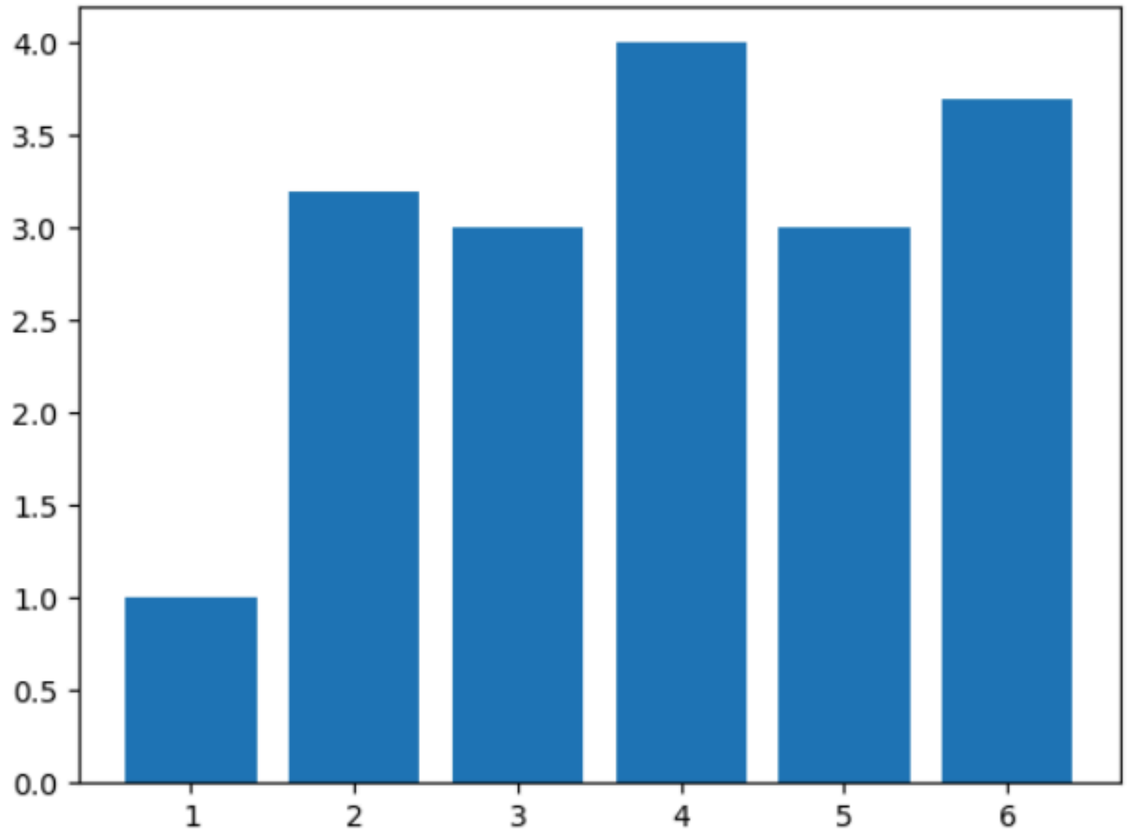
[7]:
```
plt.bar(x, y_data)
plt.show() # พล็อตกราฟแกน x แกน y แนวตั้ง
```



Jupyter Notebook

image processing

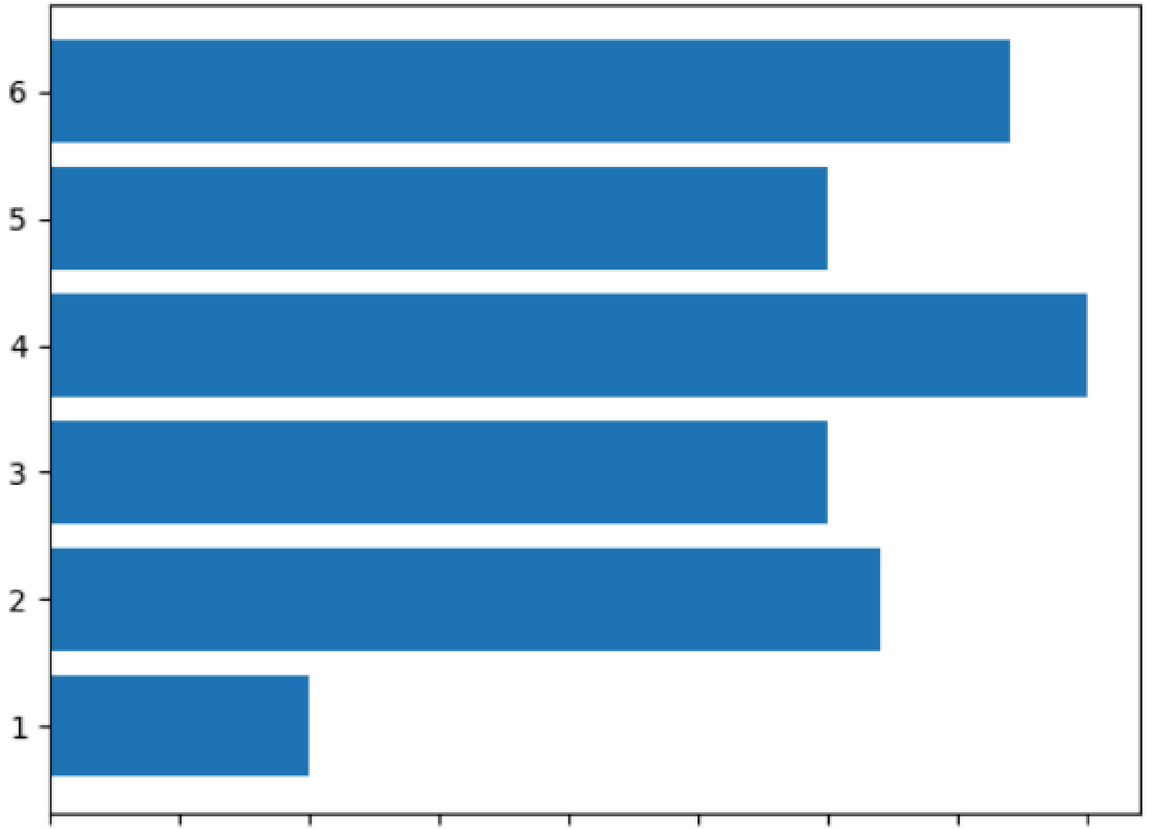
[7]:
```
plt.bar(x, y_data)
plt.show() # พล็อตกราฟแกน x แกน y แนวตั้ง
```



Jupyter Notebook

image processing

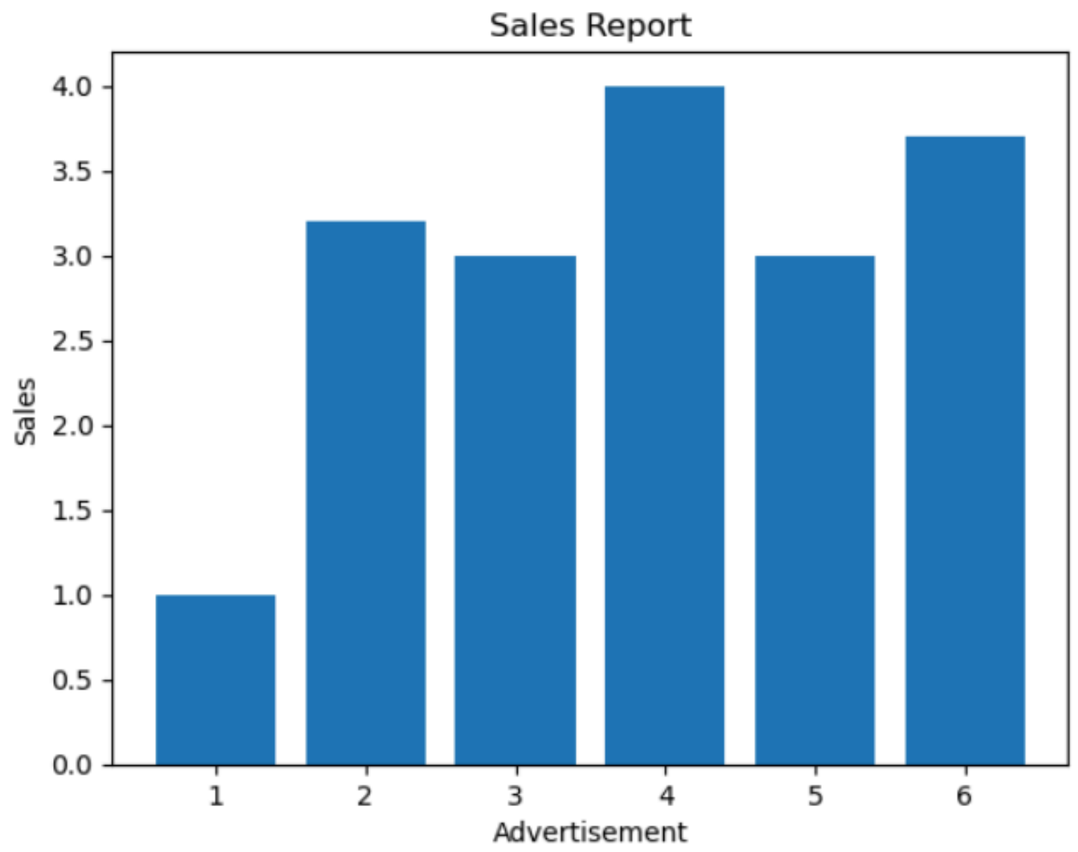
```
[8]: plt.barh(x, y_data) # พล็อตกราฟแท่ง แนวนอน
     plt.show()
```



Jupyter Notebook

image processing

```
[11]: plt.title('Sales Report')
      plt.bar(x, y_data)
      plt.xlabel('Advertisement')
      plt.ylabel('Sales')
      plt.show()
```

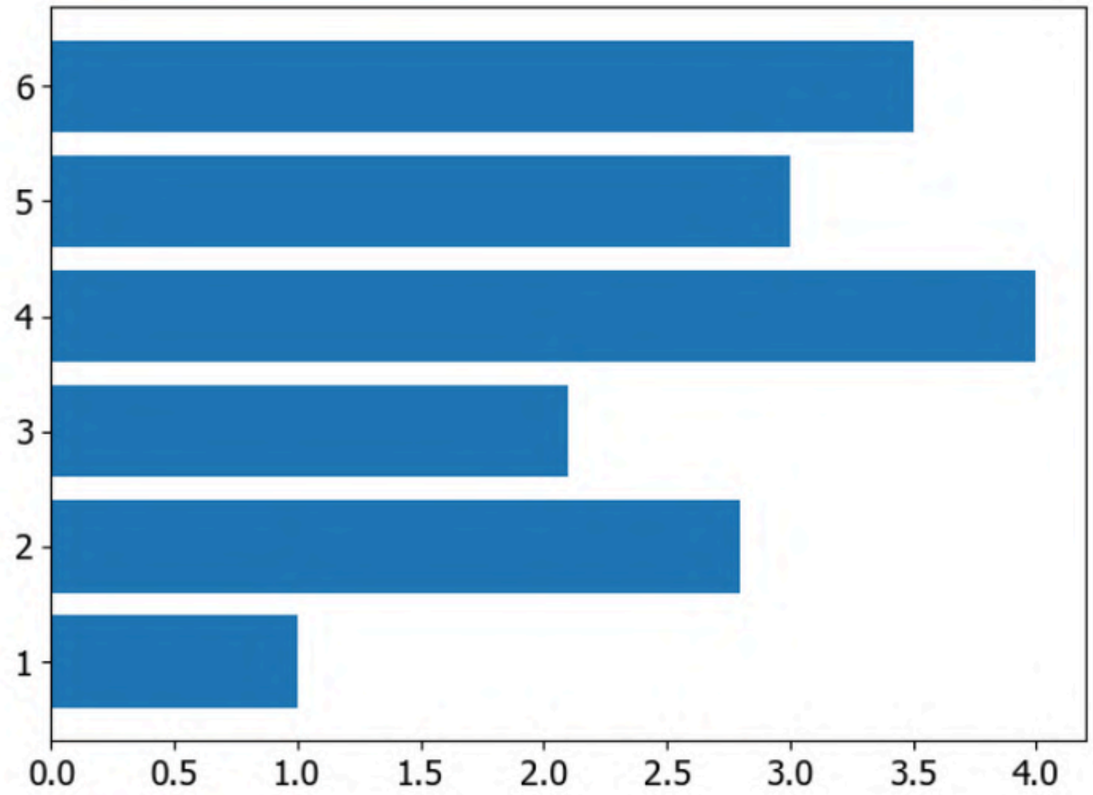


Jupyter Notebook

image processing

save

[19]:
```
plt.barh(x, ydata) # พล็อตกราฟแท่ง
plt.tight_layout()
plt.savefig('plot1.png', dpi=100)
plt.savefig('plot1.pdf', dpi=100)
plt.show()
```

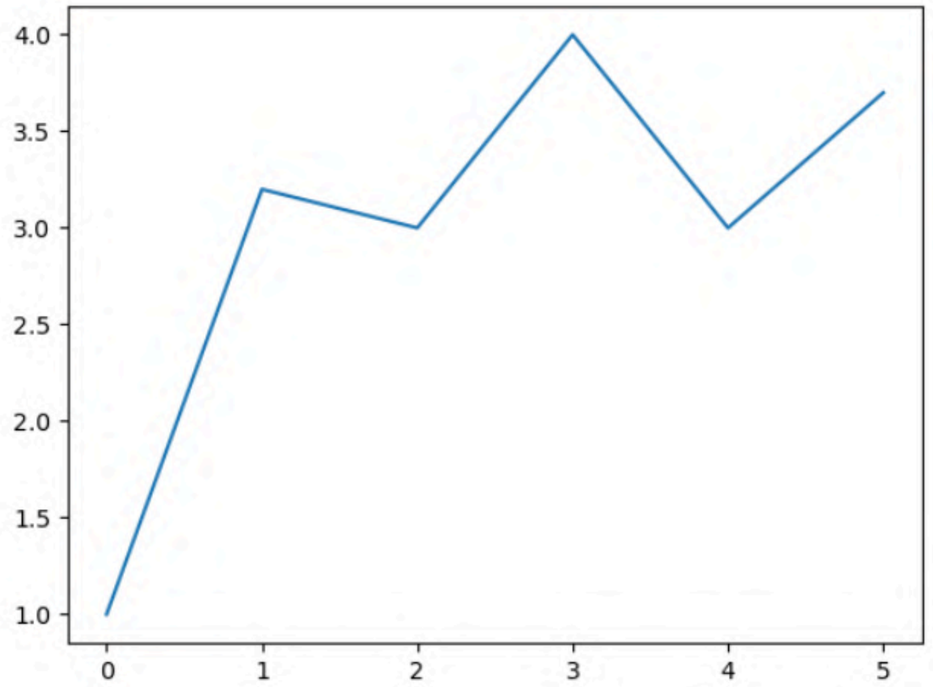


Jupyter Notebook

image processing

Line

[13]:
```
plt.plot(y_data)
plt.show()
```



Jupyter Notebook

image processing

Box plot

[15]:
```
age = [16,17,17,20,20,19,19,18,18,17,16,19,19,18,17,21]
plt.grid()
plt.boxplot(age)
plt.show()
```

[16]:
```
import pandas as pd
ds = pd.Series(age)
ds.describe().round(2)
```

Jupyter Notebook

image processing

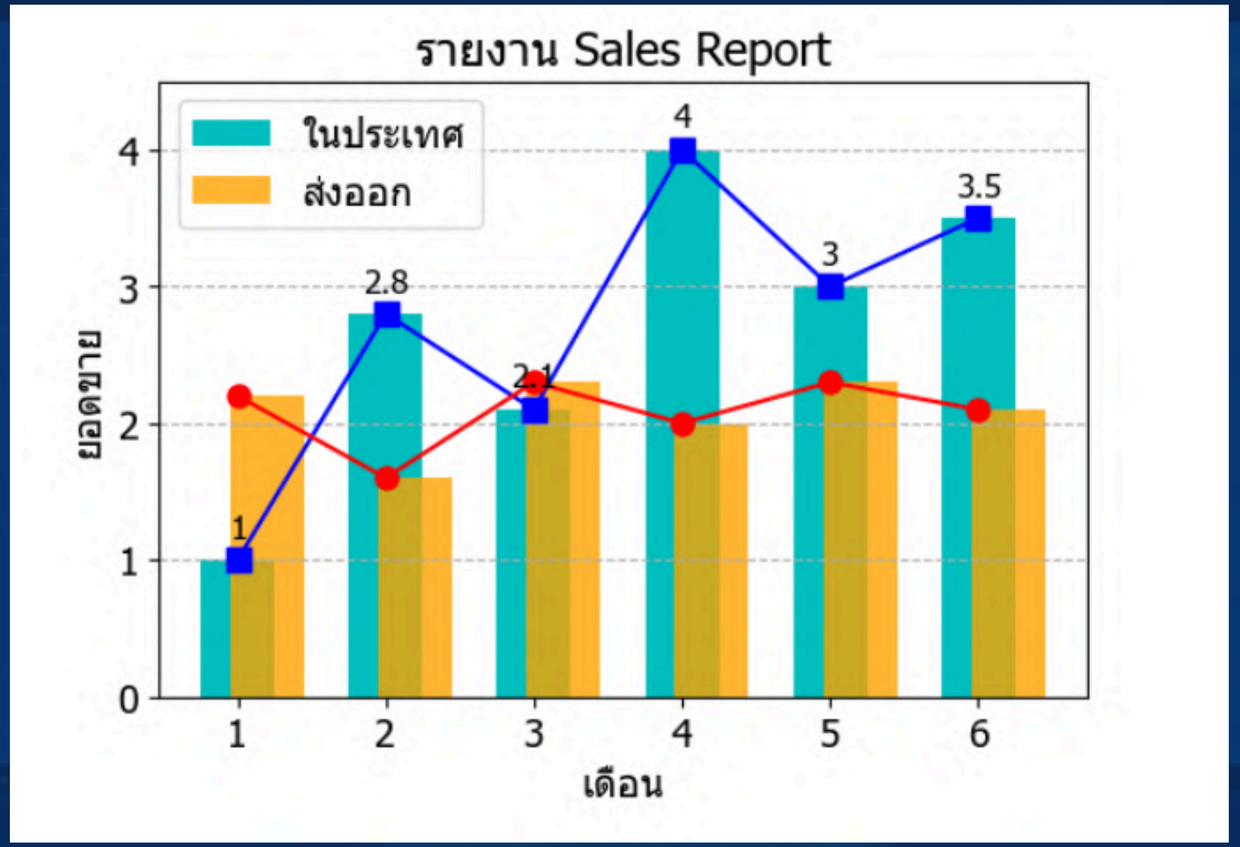
(Annotation) การเขียนป้ายรายละเอียด

[17]:

```
# เขียนรายละเอียดกำกับ (annotate)

xs = [1, 2, 3, 4, 5, 6]                 # x

ydata = [1,2.8,2.1,4,3,3.5]             # y1
ydata2 = [2.2, 1.6, 2.3, 2, 2.3, 2.1]   # y2

plt.rcParams.update({'font.size':14, 'font.family':'tahoma'})
plt.figure(figsize=(6, 4))  # ขนาดกราฟ

plt.grid(linestyle='--', axis='y')
plt.bar(x, ydata, color='c', width=0.5, label='ในประเทศ')
plt.bar([x + 0.2 for x in xs], # ขยับแท่งกราฟ
        ydata2, width=0.5, color='orange', alpha=0.8, label='ส่งออก')

plt.plot(x, ydata, color='b', marker='s', markersize=8) #plot กราฟเส้น
plt.plot(x, ydata2, color='r', marker='o', markersize=8)

plt.xlabel("เดือน")
plt.ylabel("ยอดขาย")
plt.title("รายงาน Sales Report")

for tx,ty in list(zip(x, ydata)):
    plt.annotate(ty, # ข้อความ label
                 (tx, ty), # point สำหรับ label
                 textcoords="offset points", # position ของข้อความ
                 xytext=(0, 8), # ระยะจาก points (x,y)
                 ha='center',
                 fontsize=12) # font

plt.legend()
plt.ylim(0, 4.5) #ขนาด limit ของแกน y
plt.show()
```

Jupyter Notebook

image processing



Jupyter Notebook

image processing

Jupyter Notebook

image processing

Seaborn

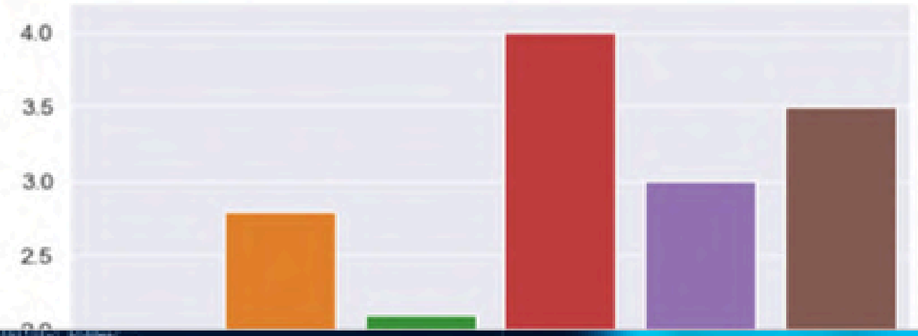
```
[ ]: !conda list seaborn
```

```
[11]: import seaborn as sns
      import matplotlib.pyplot as plt
```

```
[40]: xdata = [1,2,3,4,5,6]              # ข้อมูลสำหรับแกน x (ตัวอย่างนี้ใช้วิธีสมมุติค่าขึ้นมา)
      ydata = [1,2.8,2.1,4,3,3.5]     # ข้อมูลแกน y (x และ y ต้องมีจำนวนข้อมูลเท่ากัน)

      sns.barplot(xdata, ydata)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[40], line 4
      1 xdata = [1,2,3,4,5,6]              # ข้อมูลสำหรับแกน x (ตัวอย่างนี้ใช้วิธีสมมุติค่าขึ้นมา)
      2 ydata = [1,2.8,2.1,4,3,3.5]     # ข้อมูลแกน y (x และ y ต้องมีจำนวนข้อมูลเท่ากัน)
----> 4 sns.barplot(xdata, ydata)

TypeError: barplot() takes from 0 to 1 positional arguments but 2 were given
```

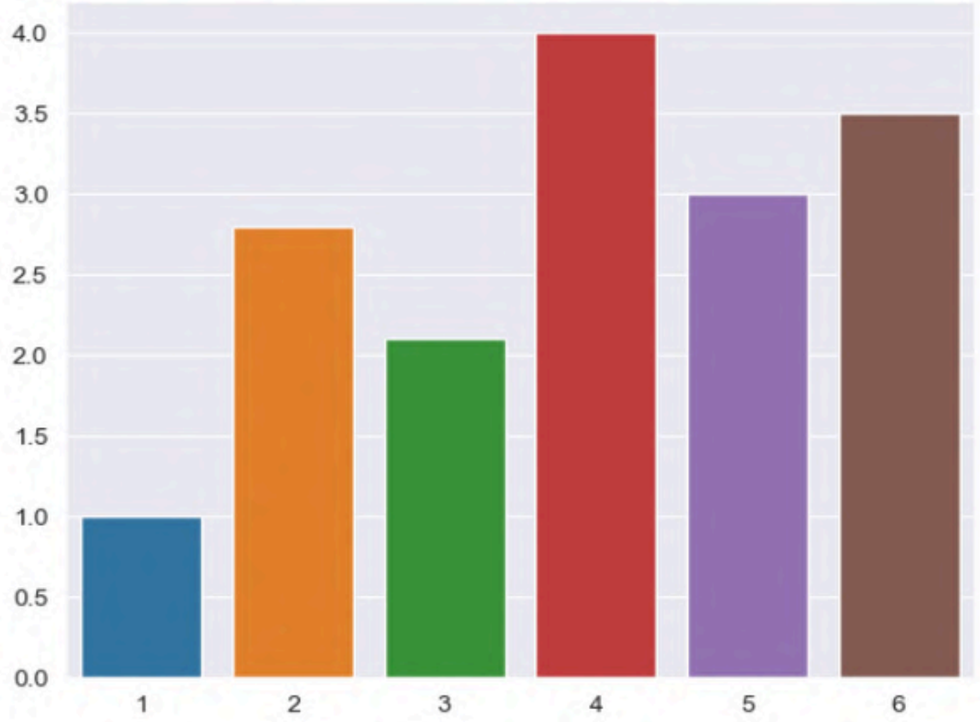
```
[22]: sns.barplot(x=xdata, y=ydata)
      plt.show()
```



Jupyter Notebook

image processing

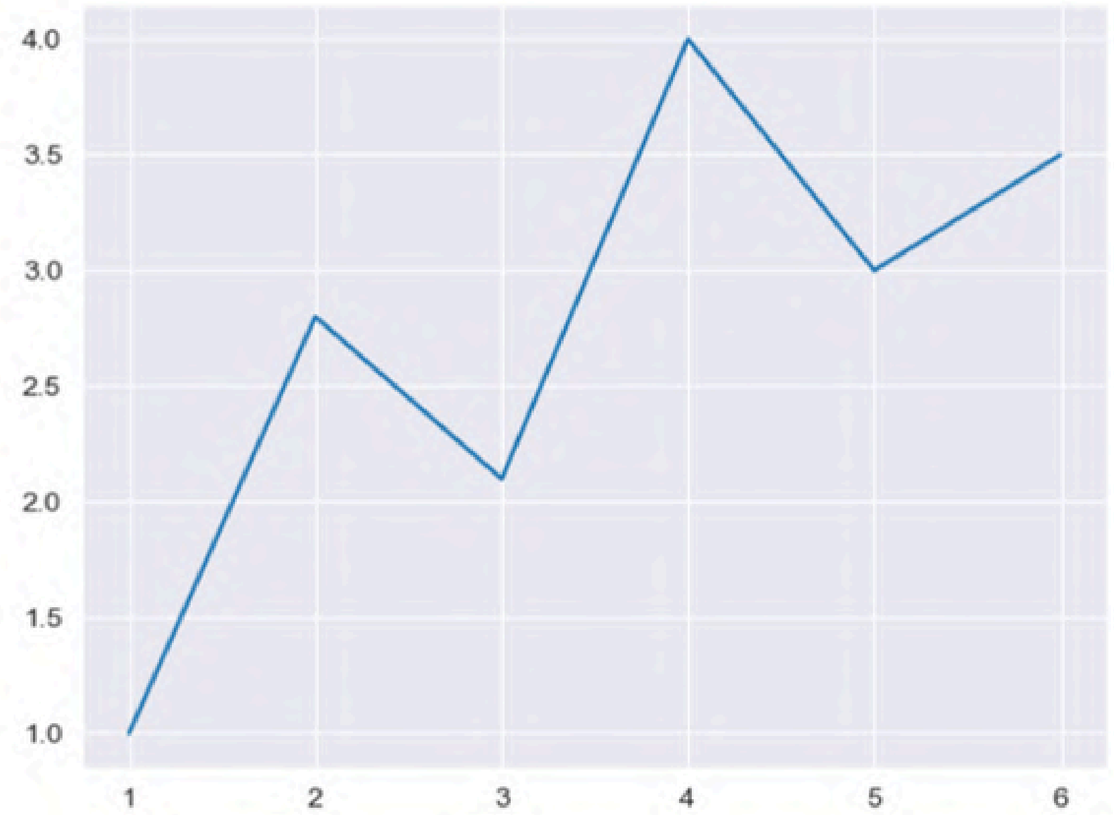
```
[22]: sns.barplot(x=xdata, y=ydata)
      plt.show()
```



Jupyter Notebook

image processing

```
[23]: sns.lineplot(x=xdata, y=ydata)
      plt.savefig('snsplot.png', dpi=100)
      # plt.show()
```

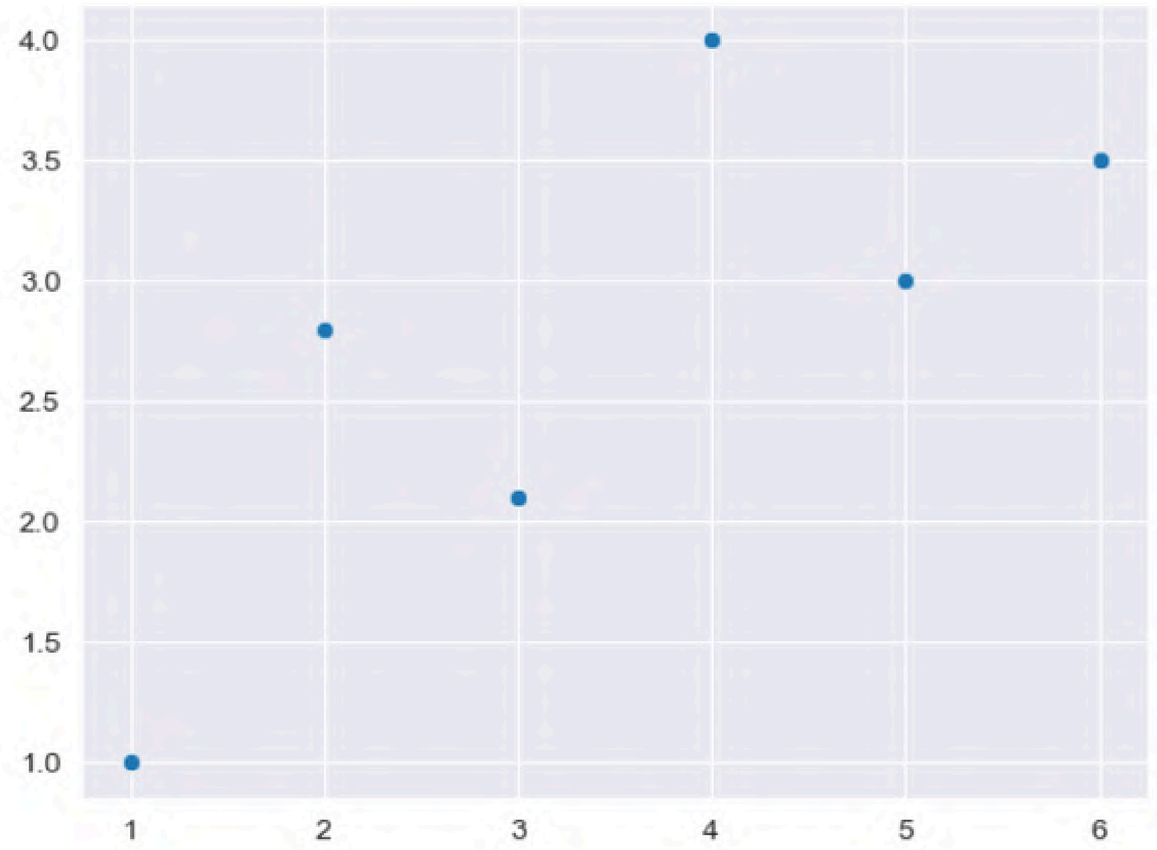


Jupyter Notebook

image processing

```
[24]: sns.scatterplot(x=xdata, y=ydata)
```

```
[24]: <Axes: >
```



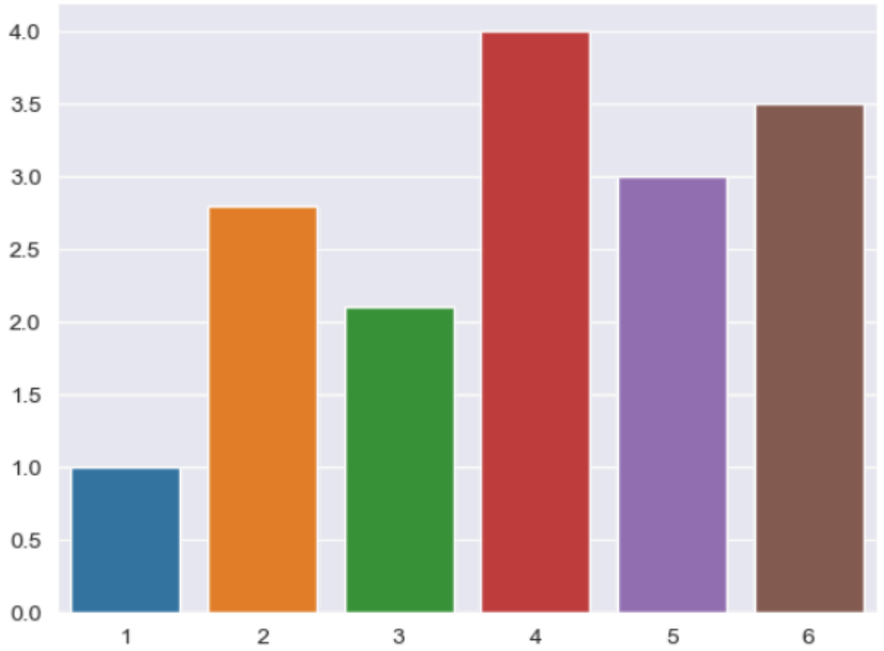
Jupyter Notebook

image processing

Style

[25]:
```
sns.set_style('darkgrid')
# sns.set_style('dark')
# sns.set_style('whitegrid')
# sns.set_style('white')
# sns.set_style('ticks')

sns.barplot(x=xdata, y=ydata)
```

[25]: <Axes: >

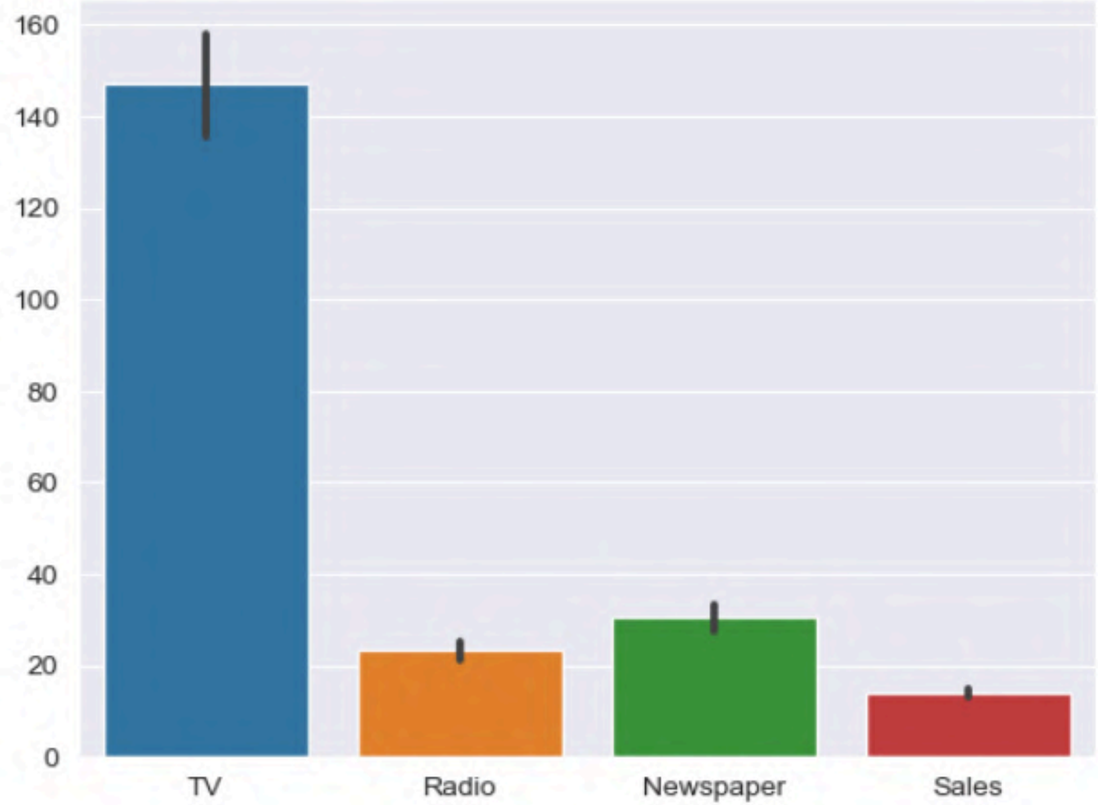


Jupyter Notebook

image processing

```
[30]: sns.barplot(data=df)
```

```
[30]: <Axes: >
```

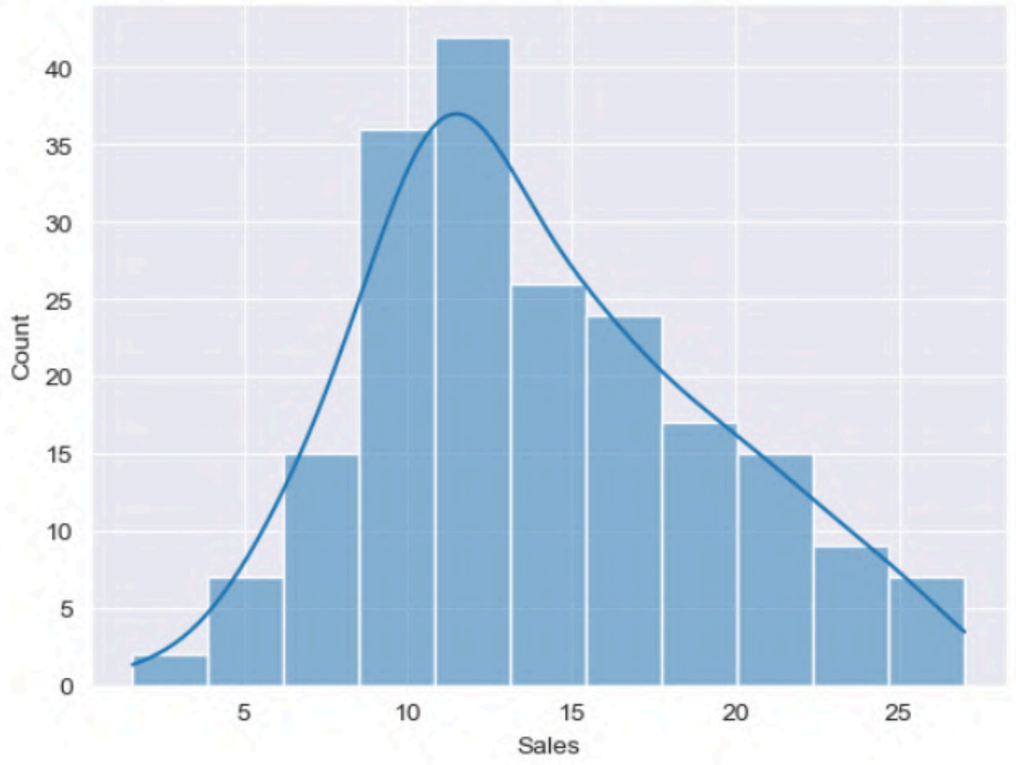


Jupyter Notebook

image processing

Histogram

[32]:
```
sns.set_style('darkgrid')

sns.histplot(df.Sales, kde=True)
# sns.distplot(df.Sales, kde=True)
plt.show()
```



Jupyter Notebook

image processing

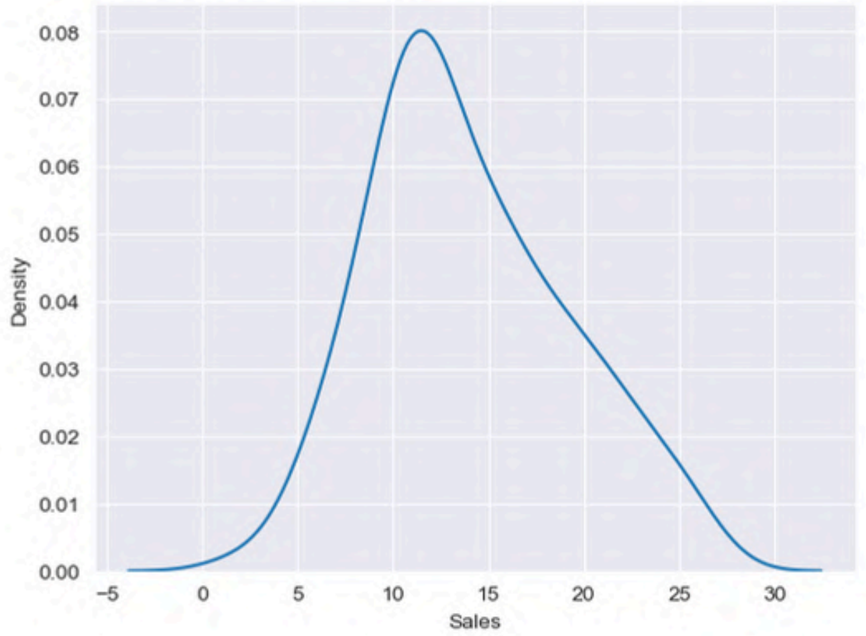
KDE

A kernel density estimate (KDE) plot (for visualizing the distribution of observations in a dataset)

```
[33]: sns.kdeplot(df.Sales, shade=False)
```

```
C:\Users\bluep\AppData\Local\Temp\ipykernel_14356\4156251864.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=False`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(df.Sales, shade=False)
```

```
[33]: <Axes: xlabel='Sales', ylabel='Density'>
```



Jupyter Notebook

image processing

Pairplot

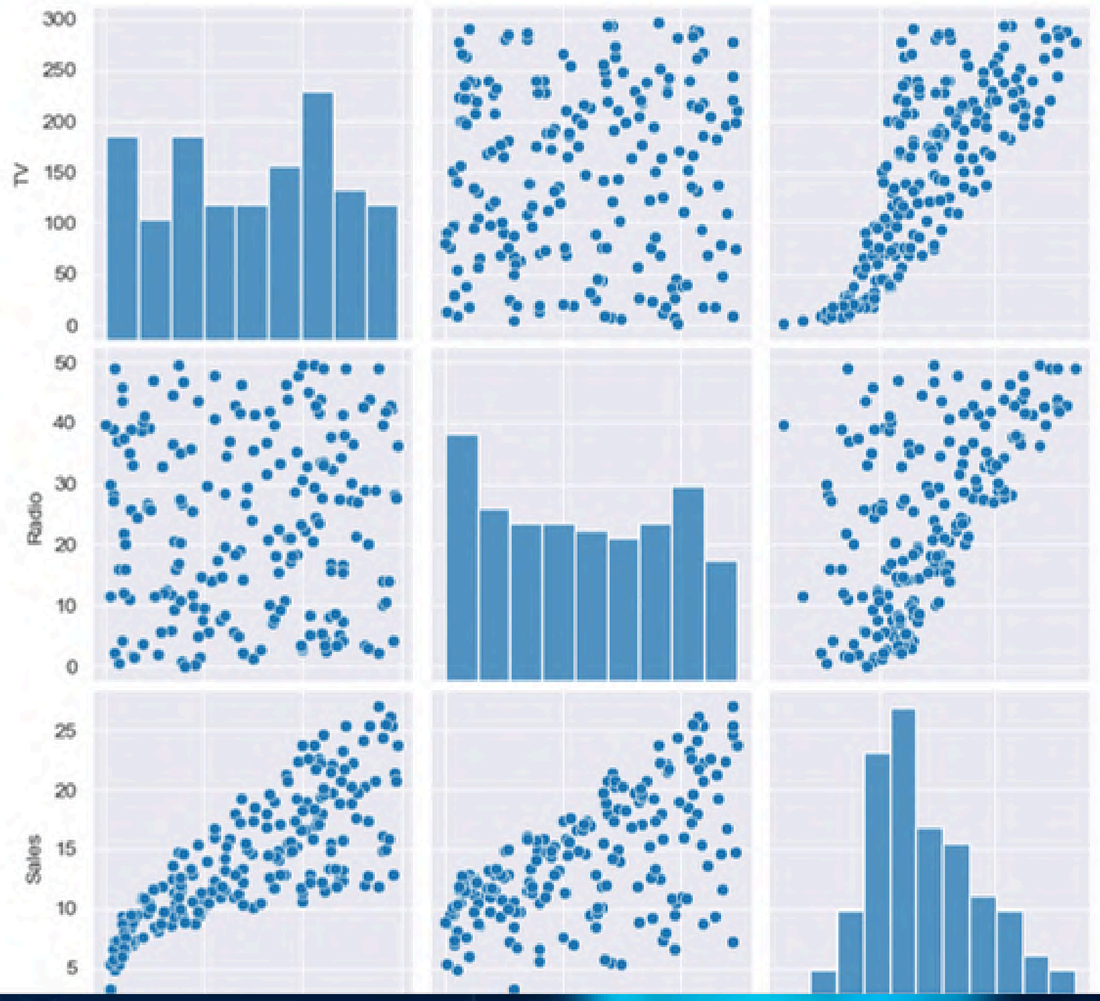
```
[34]: sns.pairplot(df)
      plt.show()
```

Jupyter Notebook

image processing

```
[35]: columns = ['TV', 'Radio', 'Sales']
      sns.pairplot(df[columns])
```

```
[35]: <seaborn.axisgrid.PairGrid at 0x2273672b220>
```



Jupyter Notebook

image processing

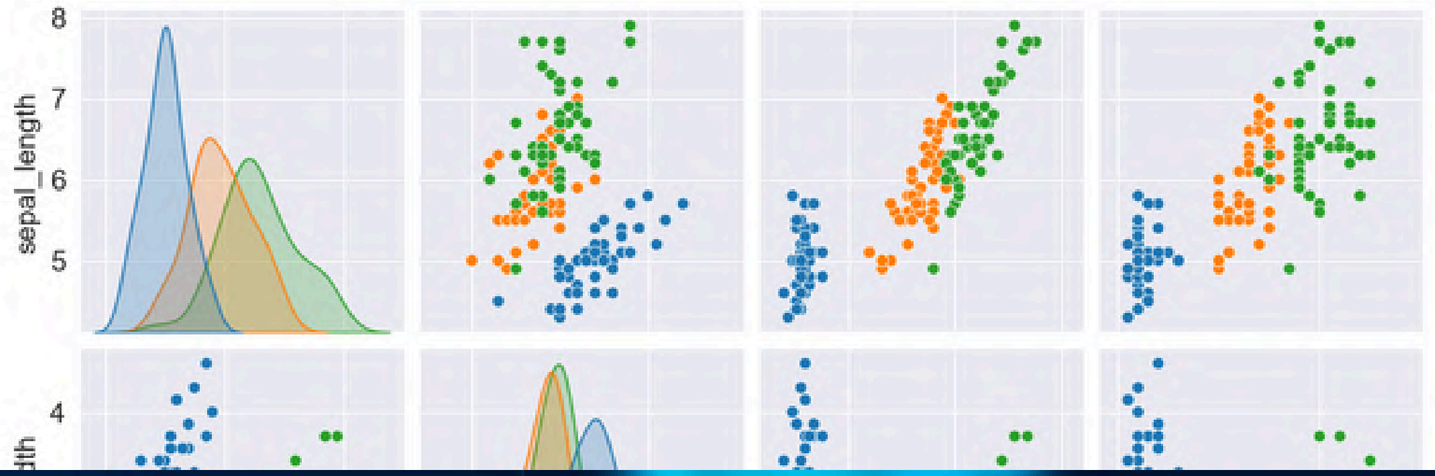
Iris dataset

```
[36]: df = sns.load_dataset('iris')
      df.head()
      df.tail()
```

[36]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

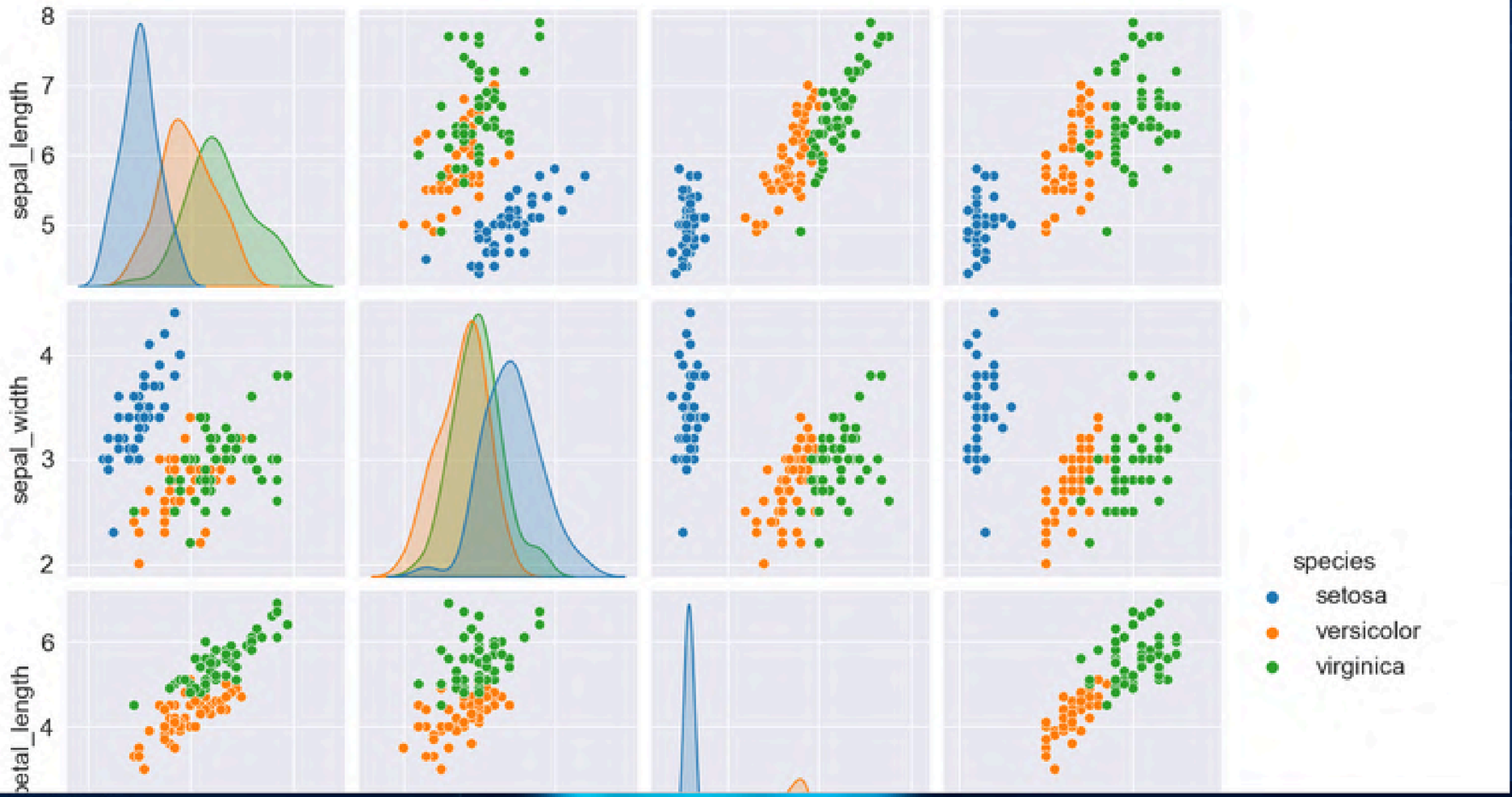
```
[37]: plt.rcParams.update({'font.size':14})
      # sns.pairplot(df)
      sns.pairplot(df, hue='species')
      plt.show()
```



Jupyter Notebook

image processing

```
[37]: plt.rcParams.update({'font.size':14})
      # sns.pairplot(df)
      sns.pairplot(df, hue='species')
      plt.show()
```



Jupyter Notebook

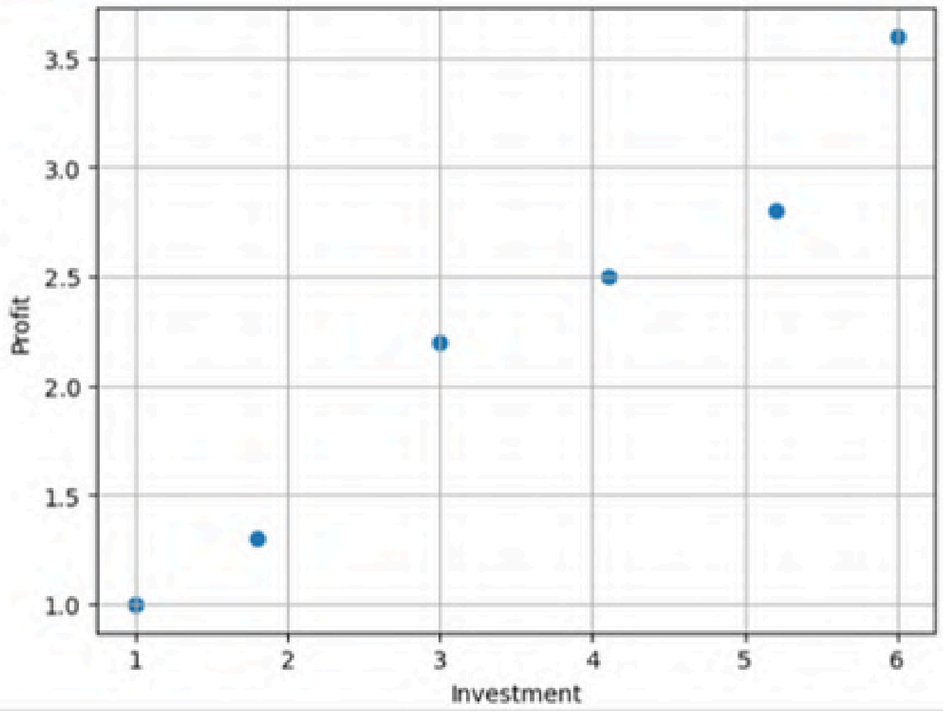
image processing

Jupyter Notebook

image processing

ข้อมูลปกติ Original Programming

[1]:
```
import matplotlib.pyplot as plt

x_data = [1.0, 1.8, 3.0, 4.1, 5.2, 6.0]
y_data = [1,   1.3, 2.2, 2.5, 2.8, 3.6]

plt.xlabel('Investment')
plt.ylabel('Profit')
plt.scatter(x_data, y_data)
plt.grid()
plt.show()
```



Jupyter Notebook

image processing

Machine Learning (ใช้ Scikit-learn)

[3]:
```
!conda list scikit-learn
```

```
# packages in environment at C:\Users\bluep\anaconda3:
#
# Name                    Version                   Build  Channel
scikit-learn              1.2.1           py310hd77b12b_0
scikit-learn-intelex      2023.0.2        py310haa95532_0
```

[4]:
```
import matplotlib.pyplot as plt
import numpy as np
```

Dataset

[6]:
```
x_data = [1.0, 1.8, 3.0, 4.1, 5.2, 6.0]
y_data = [1,   1.3, 2.2, 2.5, 2.8, 3.6]

x = np.array(x_data)        # Numpy Array
y = np.array(y_data)
```

[7]:
```
x
```

[7]:
```
array([1. , 1.8, 3. , 4.1, 5.2, 6. ])
```

[8]:
```
y
```

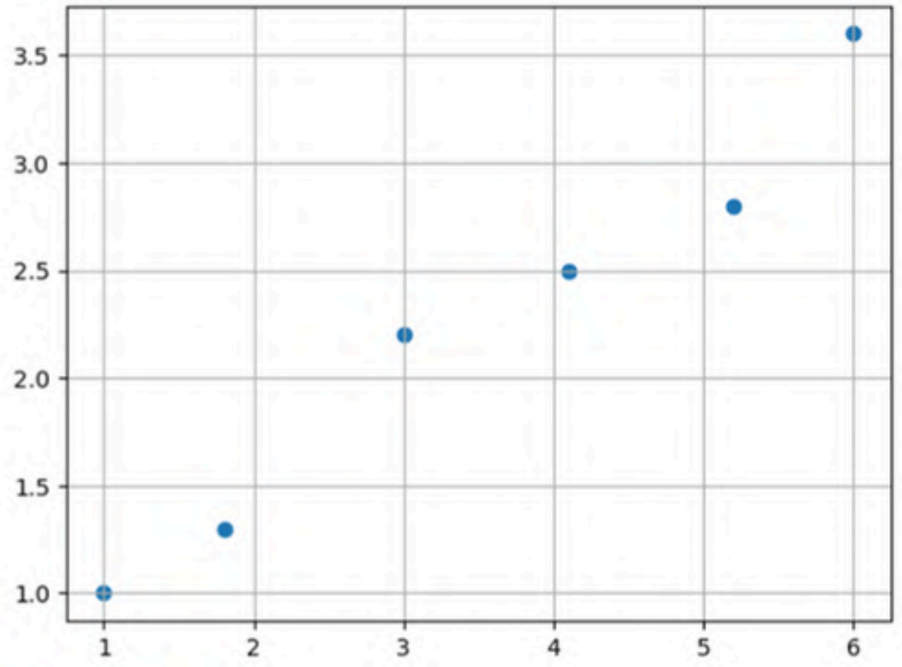
[8]:
```
array([1. , 1.3, 2.2, 2.5, 2.8, 3.6])
```

Jupyter Notebook

image processing

Data Visualization

[9]:
```
plt.scatter(x,y)
plt.grid()
plt.show()
```



[10]:
```
x = x.reshape(-1, 1)
x
```

[10]:
```
array([[1. ],
       [1.8],
       [3. ],
```

Jupyter Notebook

image processing

Model & Train

[11]:
```
from sklearn.linear_model import LinearRegression
```

[12]:
```
model = LinearRegression()
model.fit(x, y)
```

[12]:
```
▾ LinearRegression
LinearRegression()
```

Predict ทำนาย

[13]:
```
new_input = 2.5
model.predict([[new_input]])
```

[13]:
```
array([1.73670696])
```

[22]:
```
x_input = [[2.0],
           [2.5],
           [3],
           [7.0]] # ไม่จำเป็นต้องเรียงแถว

y_predict = model.predict(x_input)
y_predict
```

[22]:
```
array([1.49246448, 1.73670696, 1.98094944, 3.93488926])
```

[24]:
```
x_input = [2.0, 2.5, 3, 7.0]
x_input = np.array(x_input).reshape(-1, 1)

x_input
```

[24]:
```
array([[2. ],
       [2.5],
```

Jupyter Notebook

image processing

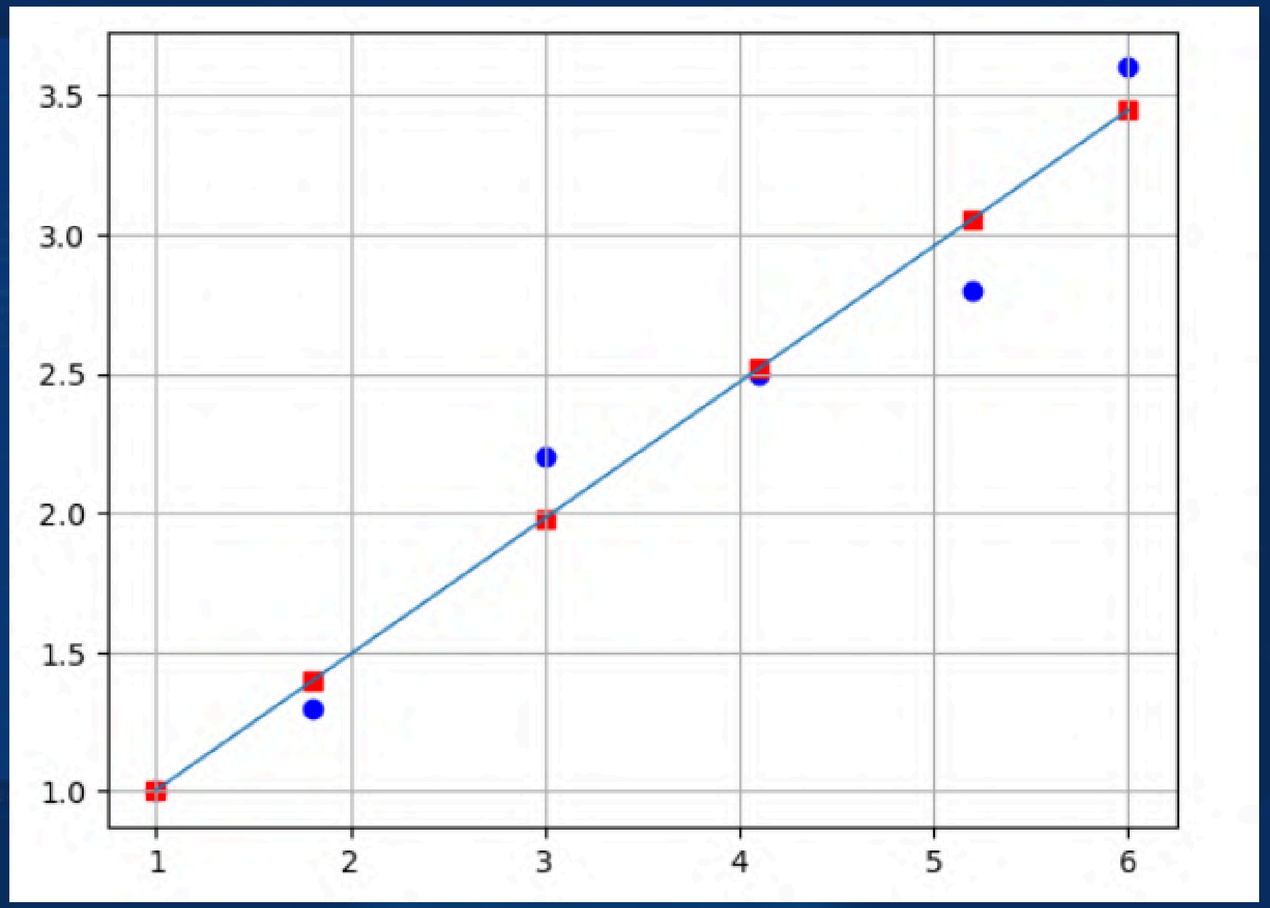
[28]:
```
# วนรอบแสดง input และ ผลลัพธ์
for i, y in enumerate(y_predict):
    print('x={} y={:.4f} '.format(x_input[i], y) )
```

```
x=[2.] y=1.4925
x=[2.5] y=1.7367
x=[3.] y=1.9809
x=[7.] y=3.9349
```

[27]:
```
predict = model.predict(x)

# plt.rcParams['figure.figsize'] = 4, 3  #

plt.grid()
plt.scatter(x_data, y_data, color='b')        # ข้อมูลจริง (real data points)
plt.plot(x_data, predict, linewidth='1')          # ทำนาย (prediction)
plt.scatter(x_data, predict, color='r', marker='s')
# plt.savefig('model1.png', dpi=100)
plt.show()
```

Jupyter Notebook

image processing



Jupyter Notebook

image processing

Jupyter Notebook

image processing

การใช้โมเดล Model to use

[2]:
```
import joblib
import numpy as np

model = joblib.load('model_math.pkl')
```

[3]:
```
model.coef_ , model.intercept_       # m and b
```

[3]:
```
(array([0.48848496]), 0.5154945733698053)
```

[4]:
```
x_input = [2.0, 2.5, 3, 5.0, 7.0]
x_input = np.array(x_input).reshape(-1, 1)

y_predict = model.predict(x_input)
y_predict
```

[4]:
```
array([1.49246448, 1.73670696, 1.98094944, 2.95791935, 3.93488926])
```

[5]:
```
# วนรอบแสดง input และ ผลลัพธ์
for i, y in enumerate(y_predict):
    print('x={} y={:.4f} '.format(x_input[i], y) )
```

```
x=[2.] y=1.4925
x=[2.5] y=1.7367
x=[3.] y=1.9809
x=[5.] y=2.9579
x=[7.] y=3.9349
```

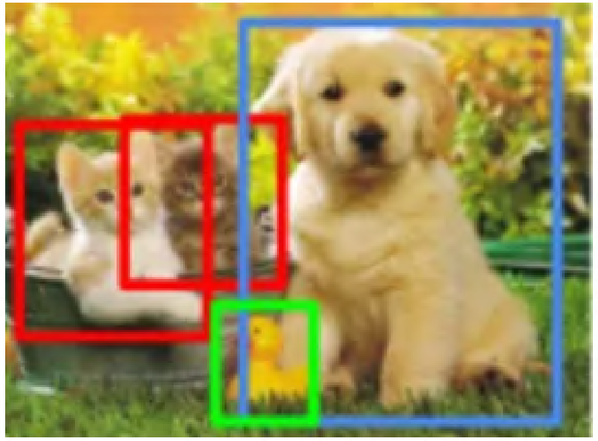
IMAGE CLASSIFICATION

AI

Image Classification

?

Image Detection

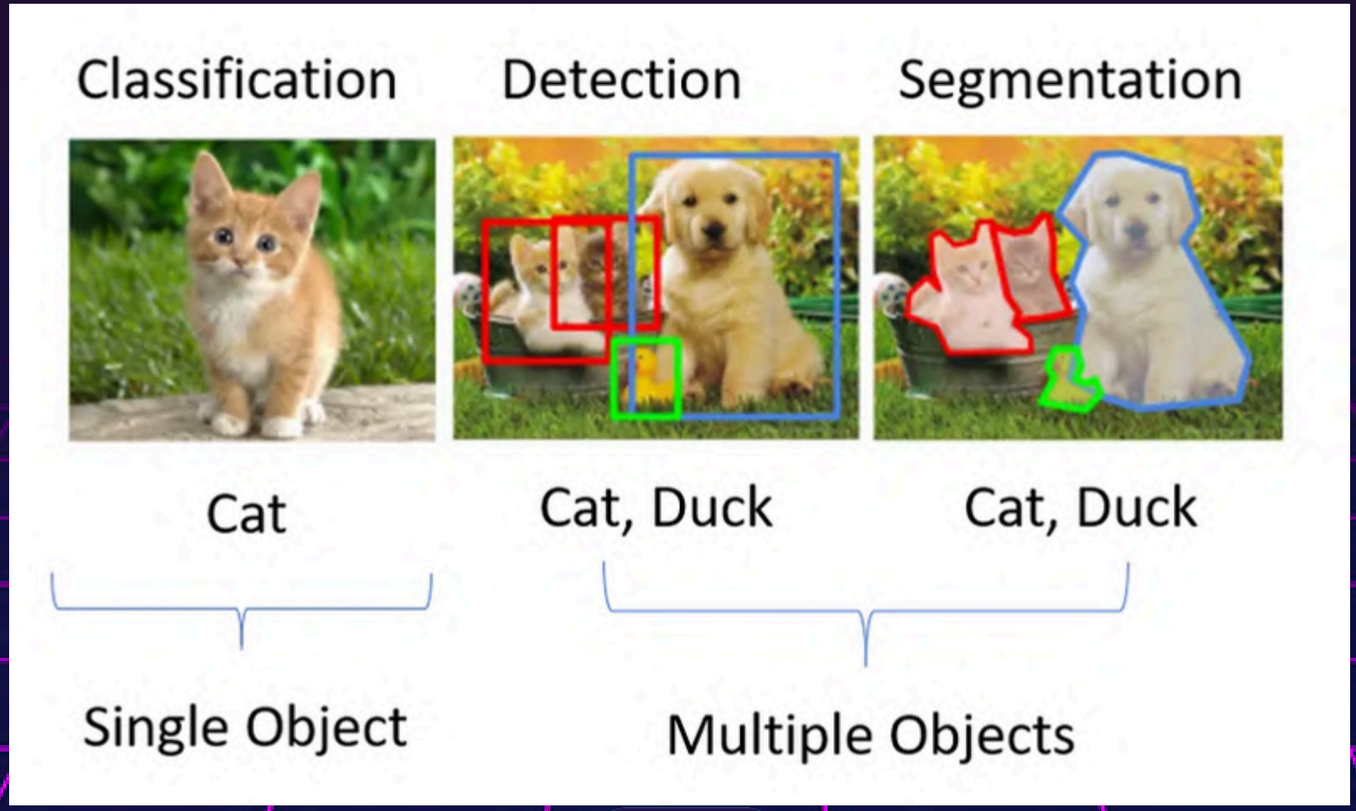


น้องแมวแอนด์เดอะแก๊งค์

image classification to AI :

EXIT

IMAGE CLASSIFICATION



EXIT

IMAGE CLASSIFICATION

Medical Sector

X-ray

AI Focus

Image

Segmentation

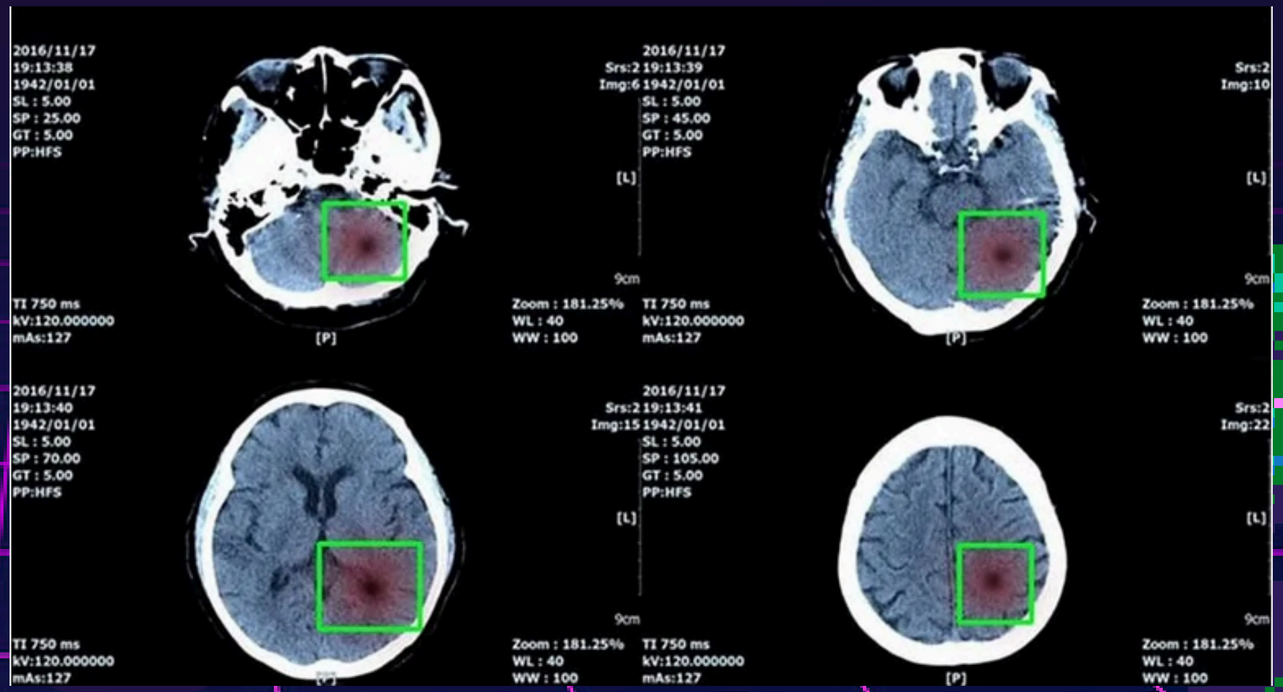
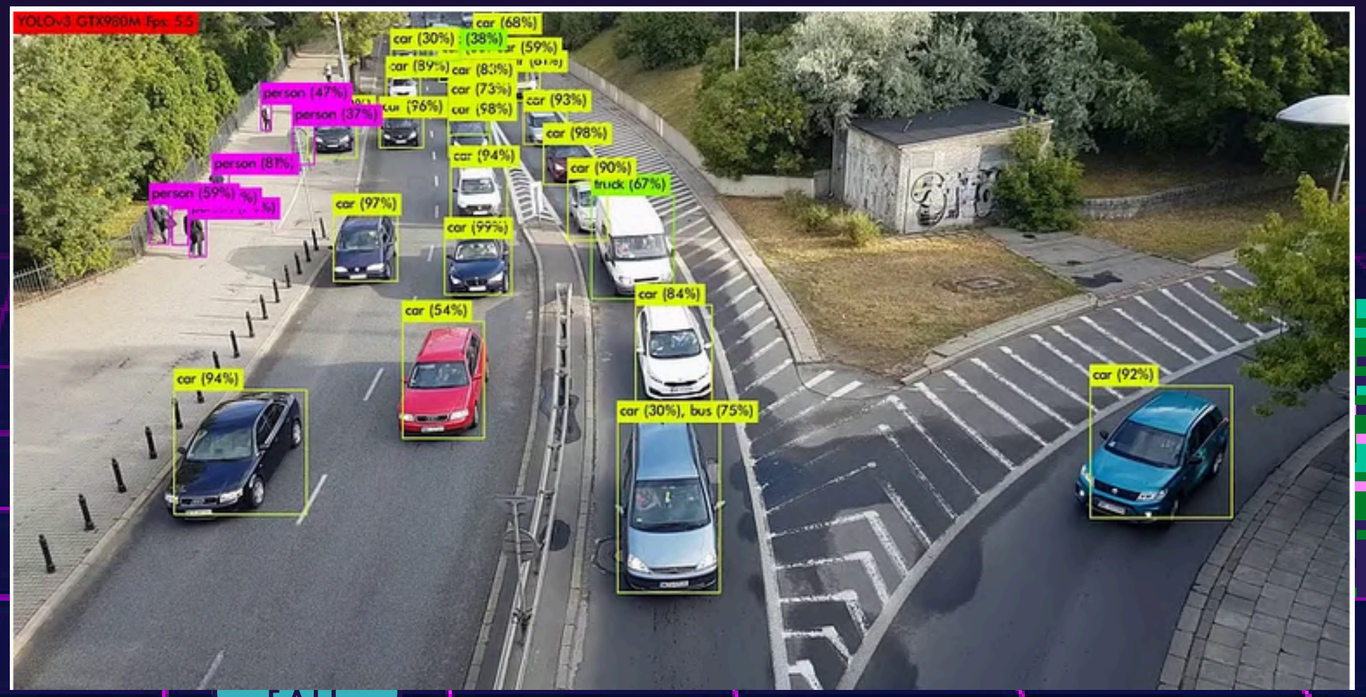


IMAGE CLASSIFICATION

Traffic Sector

AI

( )



YOLOV5 คืออะไร ?

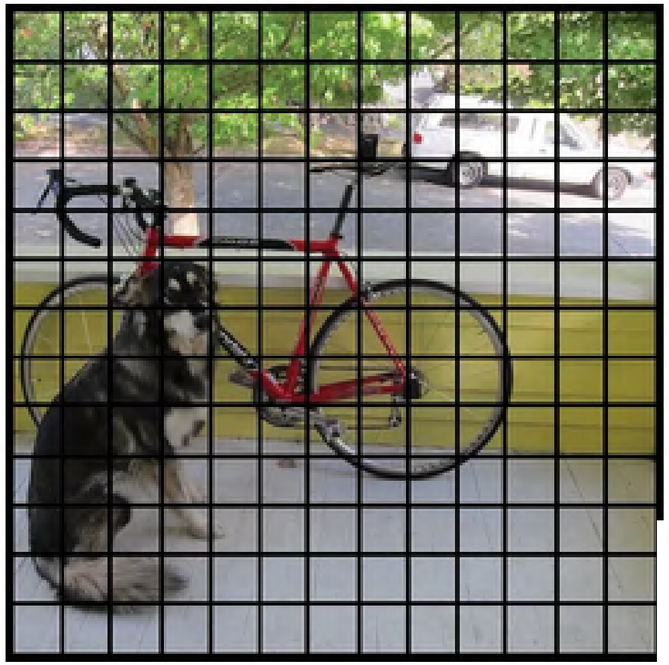
‘You only look once’
Version 10 YOLO
ultralytics
Image Detection

EXIT

YOLOV5 คืออะไร ?

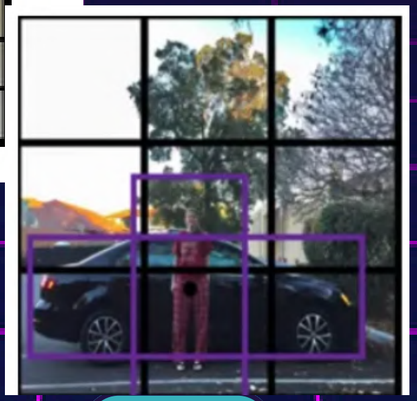
EXIT

YOLOV5 คืออะไร ?



grid

elements pixel



EXIT

layer predict

intersection

การติดตั้ง YOLOV5

5)

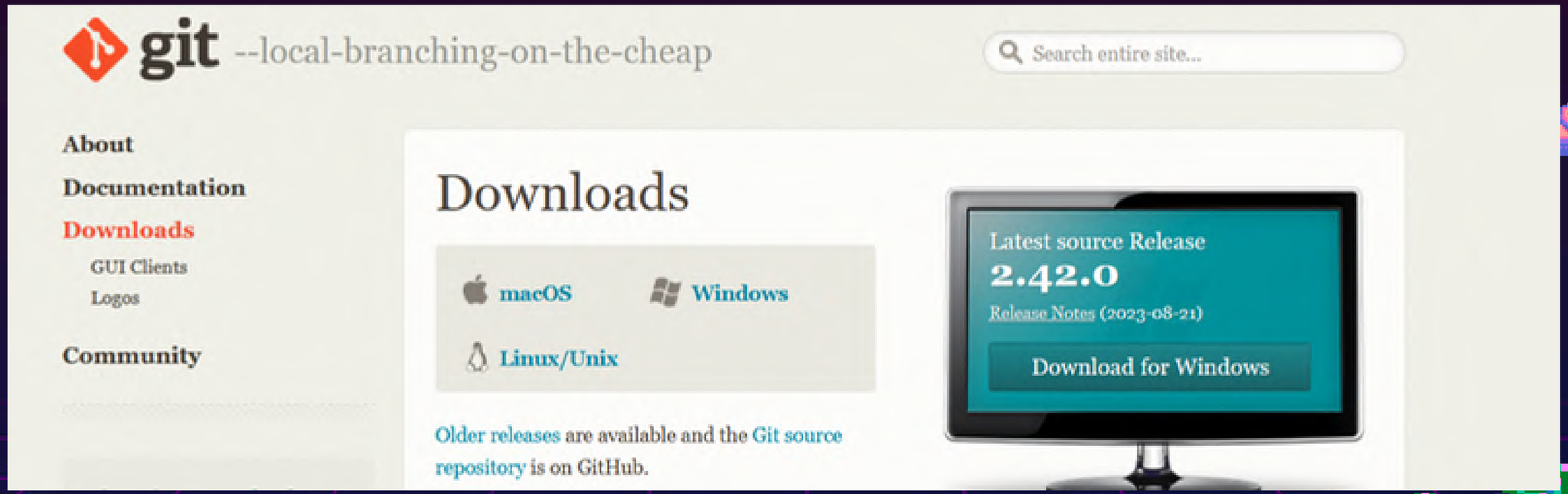
1) Image Detection
2) (Data Gathering and Collecting)
3) (Data Labeling) Image Augmentation)
4) (Modelling)
5) (Deployment)

medium yolov5 :

EXIT

การติดตั้ง YOLOV5

1.



EXIT

การติดตั้ง YOLOV5

2. YOLOv5 Library

```
## Clone repository
$ git clone https://github.com/ultralytics/yolov5

## Change directory to yolov5
$ cd yolov5

## install required library
$ pip install -r requirements.txt  # install
```

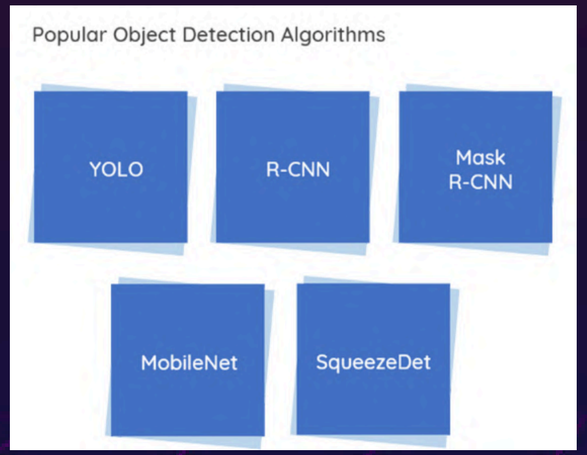
EXIT

LAB
การทดลองใช้ YOLOV5

EXIT

การใช้ YOLOV3

Classification Detect Person 0 - 1 (100%)



EXIT

การใช้ YOLOV3

0 - 1%

STEP 1 :

```
pip install opencv-python-headless requests
```

model , configuration coco name tag

STEP 2 :

YOLOv3 weights file:

- https://pjreddie.com/media/files/yolov3.weights

YOLOv3 configuration file:

- https://github.com/pjreddie/darknet/blob/master/cfg/yolov3.cfg

YOLOv3 class names file:

- https://github.com/pjreddie/darknet/blob/master/data/coco.names

EXIT

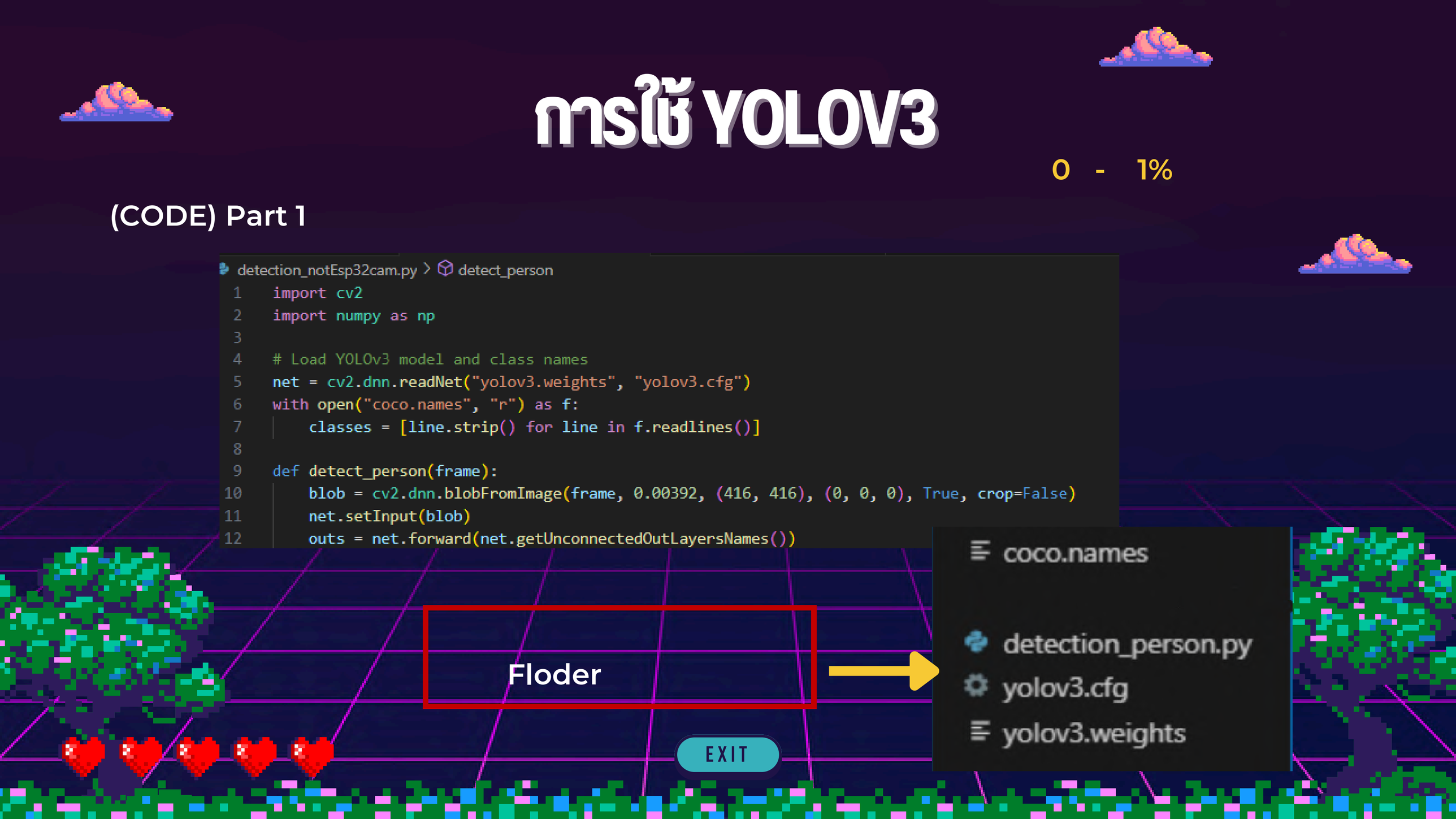
การใช้ YOLOV3

0 - 1%

(CODE) Part 1

detection_notEsp32cam.py > detect_person

```python
import cv2
import numpy as np

# Load YOLOv3 model and class names
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]

def detect_person(frame):
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(net.getUnconnectedOutLayersNames())
```



การใช้ YOLOV3

0 - 1%

(CODE) Part 2

```
13
14      class_ids = []
15      confidences = []
16      boxes = []
17      for out in outs:
18          for detection in out:
19              scores = detection[5:]
20              class_id = np.argmax(scores)
21              confidence = scores[class_id]
22              if confidence > 0.5 and classes[class_id] == 'person':
23                  center_x = int(detection[0] * frame.shape[1])
24                  center_y = int(detection[1] * frame.shape[0])
25                  w = int(detection[2] * frame.shape[1])
26                  h = int(detection[3] * frame.shape[0])
27                  x = int(center_x - w / 2)
28                  y = int(center_y - h / 2)
29                  boxes.append([x, y, w, h])
30                  confidences.append(float(confidence))
31                  class_ids.append(class_id)
32
33      indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
34
```

EXIT

การใช้ YOLOV3

0 - 1%

(CODE) Part 3

```
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            confidence = confidences[i]
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(frame, f'{label} {confidence:.2f}', (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    return frame

```

การใช้ YOLOV3

0 - 1%

(CODE) Part 4

```python
# Capture video from your webcam (change the argument to the video file if needed)
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    if frame is not None:
        frame = detect_person(frame)
        cv2.imshow("Person Detection", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

EXIT

การใช้ YOLOV3

(Result)

Thank You!

End of Session Image Processing