

Home

Galery

About

Shop



Present By

CodeBotix Learning Hub

ARTIFICIAL INTELLIGENCE

LANDING PAGE

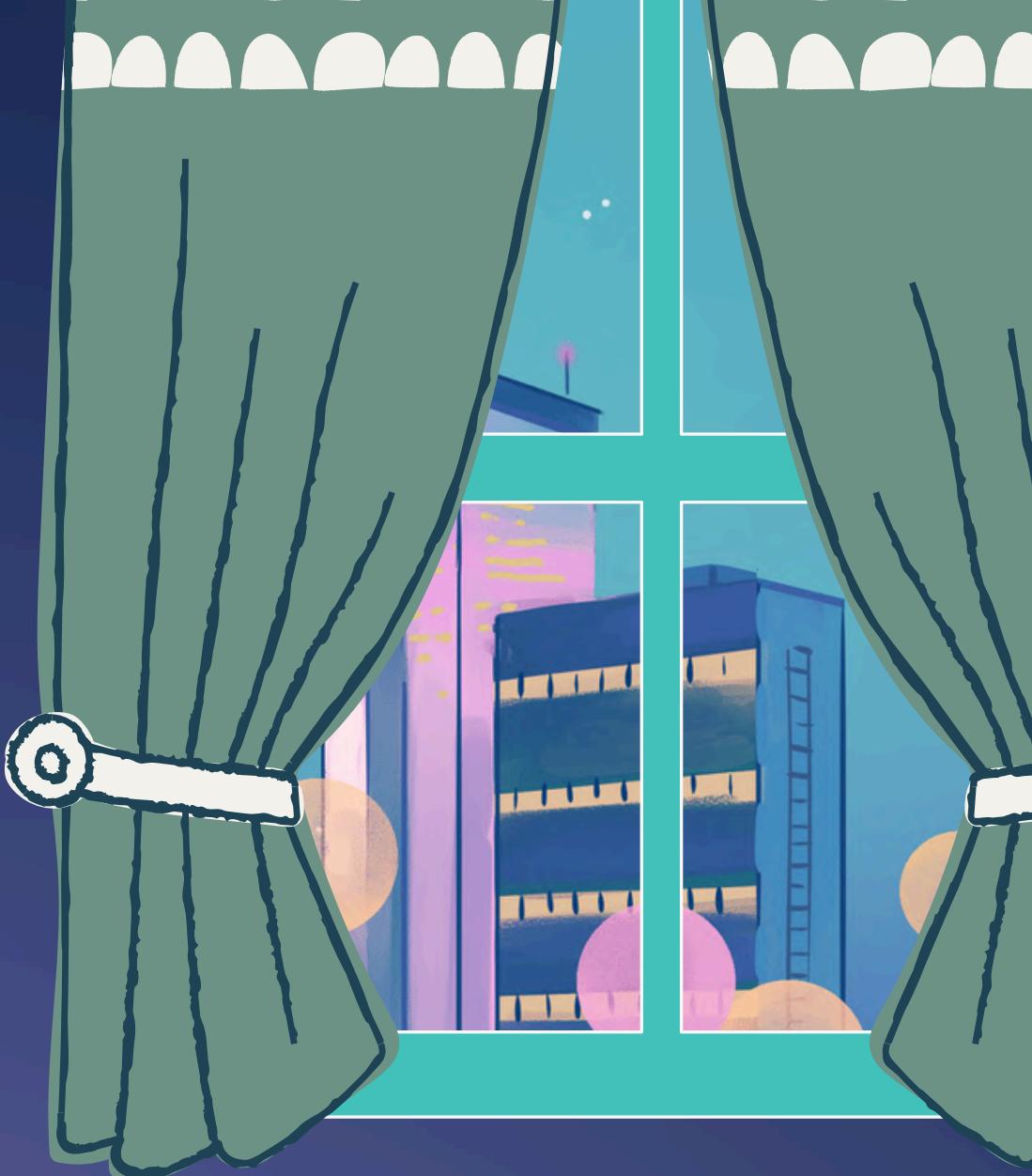
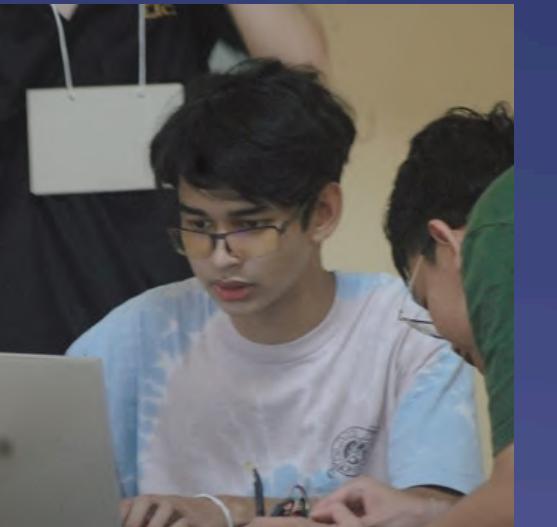
Lorem ipsum dolor sit amet, ius ei ancillae
partiendo interpretaris, duo et reque dicta
munere

FIND MORE

Basic Image Processing And AI



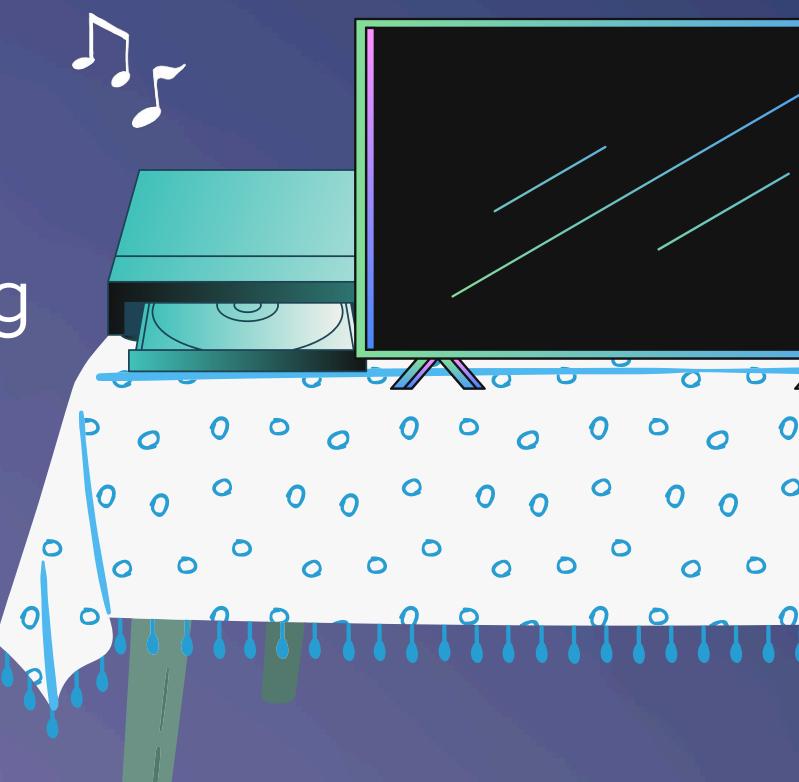
Let's get to know each other first



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/>

Face detection

Q1

Install Python

open your command prompt

Q2

```
pip install opencv-python
```

```
pip install numpy
```

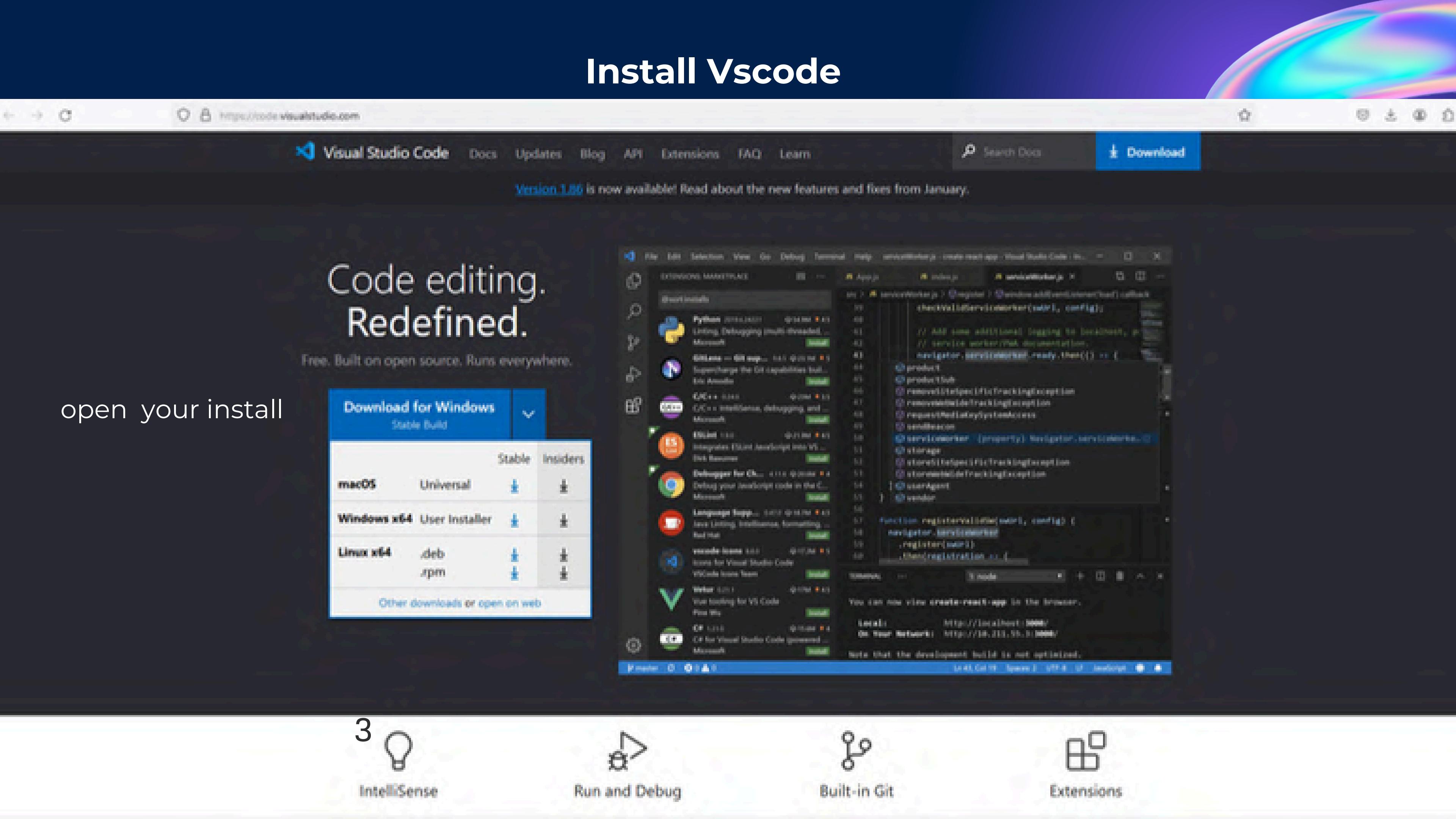


We'll use Visual Studio Code



- Create new file your project (.py file)

Install Vscode



Q3

Coding

Model (face , eye , etc.)

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

```
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)

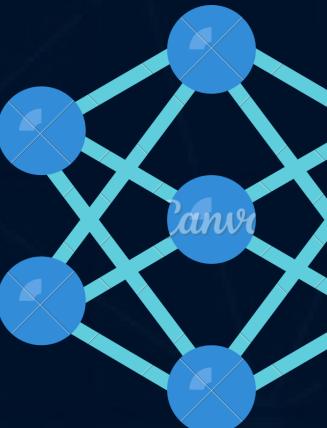
cascade = cv2.CascadeClassifier(xml_path)
if cascade.empty():
    raise IOError('Unable to load the face cascade classifier xml file')
```

download model path

Initialize the model haarcascades



Link Video Basic AI



Q3

Coding

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

Read a frame from the webcam

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

Convert the frame to grayscale

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

Perform object detection

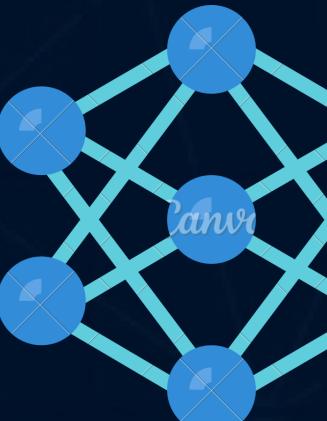
```
cv2.imshow('Face Object Detection', frame)
# Break the loop if 'q' key is pressed
if cv2.waitKey(1) & 0xFF == ord('q'):
```

When press key 'q' it will exist

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

Model (face , eye , etc.)

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



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 ตำแหน่งโฟล์เดอร์ที่ต้อง



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

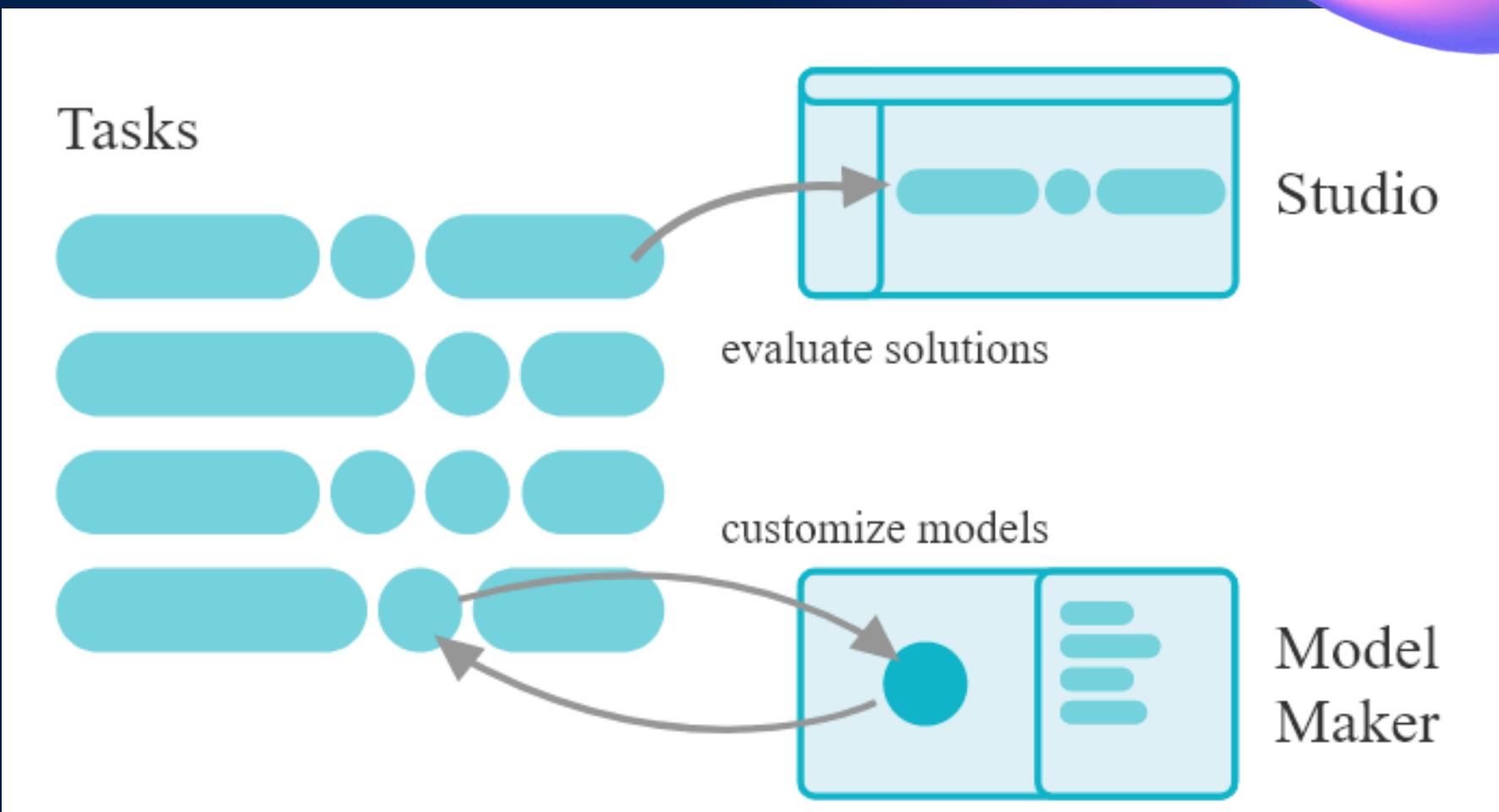
can try on others object example detection on this link : <https://shorturl.at/pHIW6>

**DO YOU WANT CONTINUE
DETECTION ANALYZE TYPE >/?<**

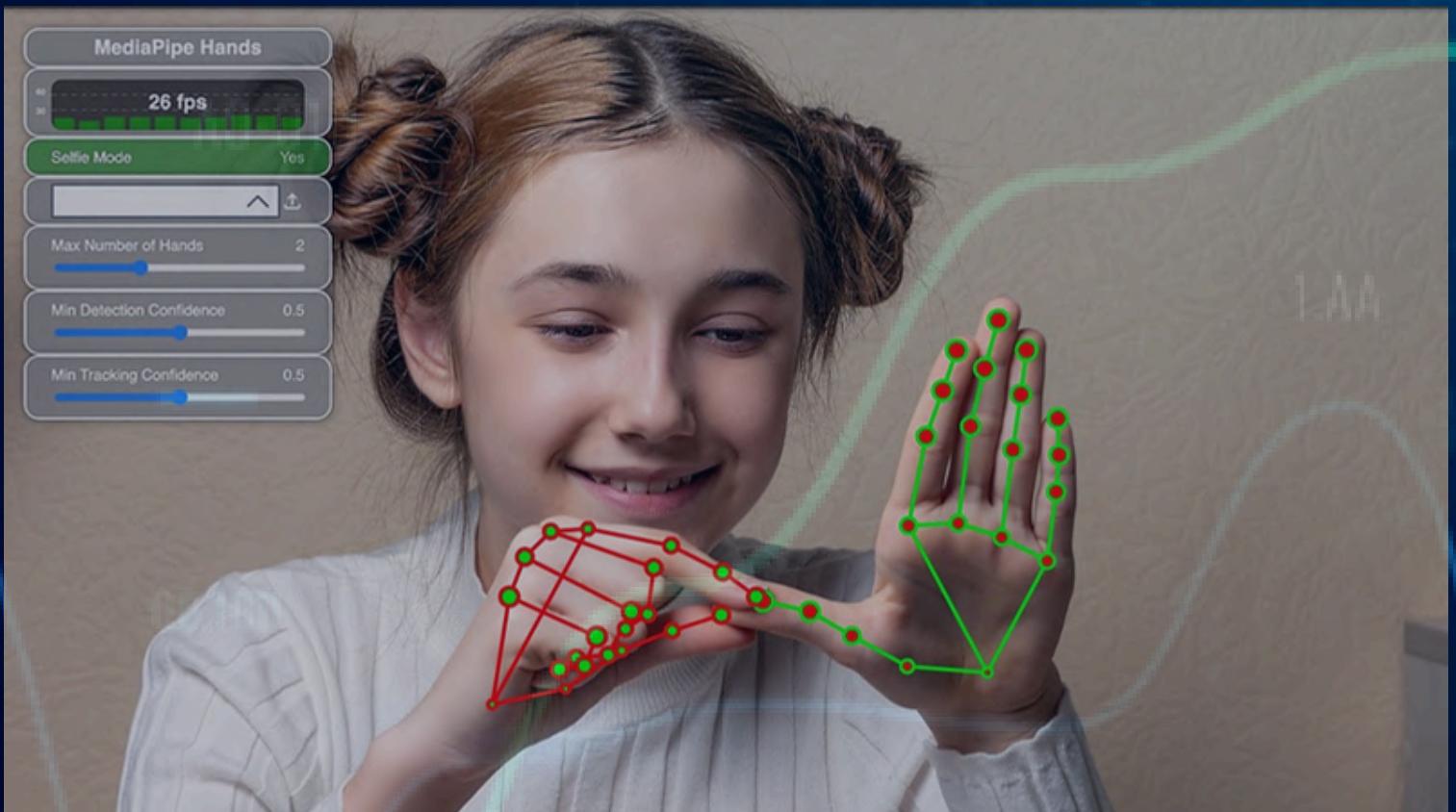
NEXT PAGE >>

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.

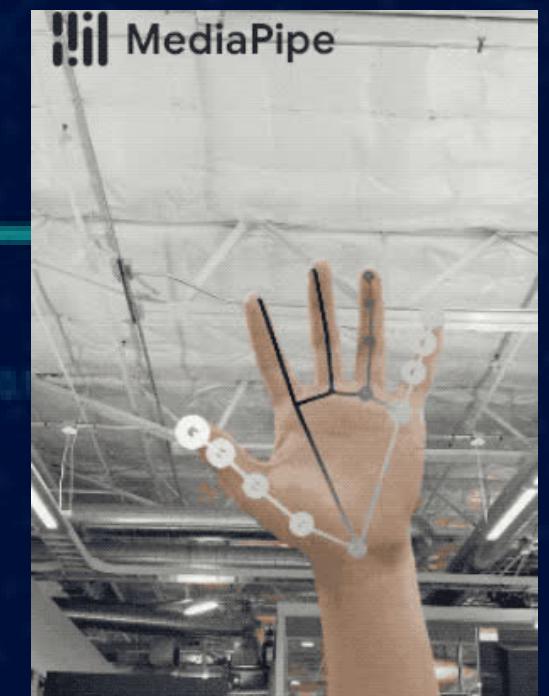


MediaPipe Hands

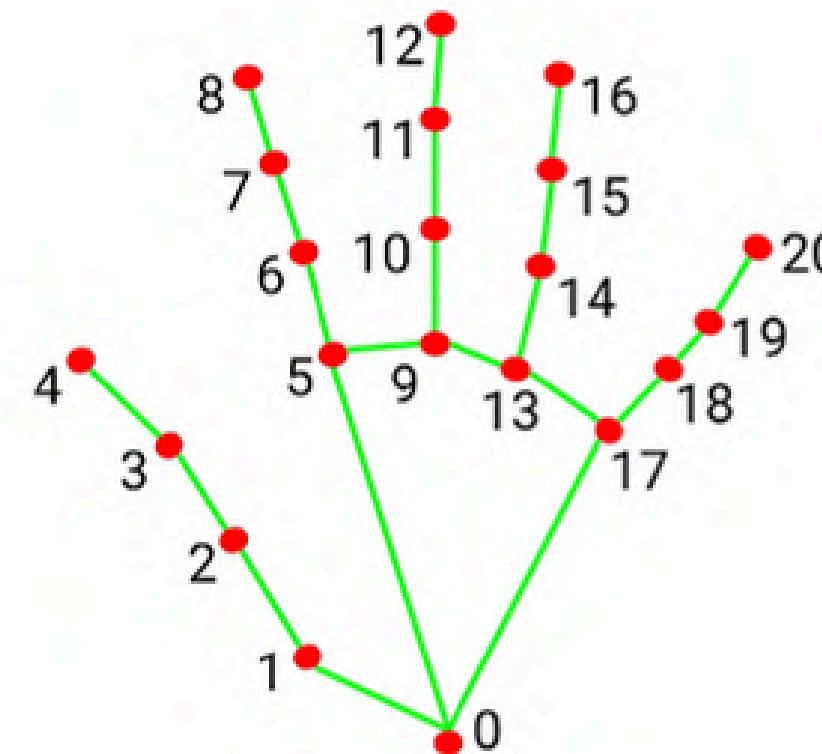


Today we will learn to use MediaPipe, which is a Machine Learning Solutions or a ready-made program from Google that can be used to do Hand Tracking accurately. And as fast as real-time tracking.

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 finger Landmarks



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

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

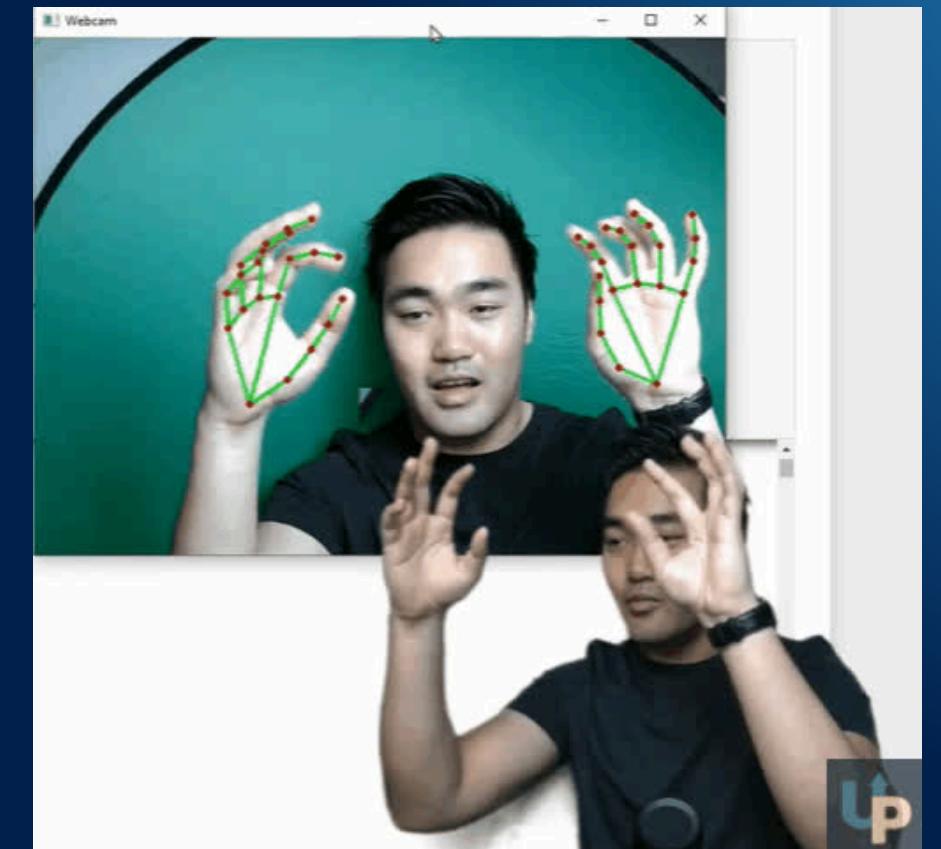
Hand Tracking

open your command prompt

Q1

```
pip install mediapipe
```

```
pip install opencv-python
```



Q3

Hand Track Coding

Our Vision

```
1 import cv2
2 import mediapipe as mp
3
4 mp_hands = mp.solutions.hands
5 mp_draw = mp.solutions.drawing_utils
6
7 webcam = cv2.VideoCapture(0) → Initialize the webcam
8
9 hands = mp_hands.Hands() → Initialize the Hand model
10
11 while True:
12     success, image = webcam.read() → Read a frame from the webcam
13
14     image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) → Convert the frame from BGR to RGB
15
16     results = hands.process(image_rgb) → Process the frame to detect
17         hand landmarks
18
19     if results.multi_hand_landmarks:
20         for hand_landmarks in results.multi_hand_landmarks:
21             mp_draw.draw_landmarks(image, hand_landmarks, mp_hands.HAND_CONNECTIONS)
22
23     cv2.imshow("Webcam", image)
24     cv2.waitKey(1)
```

Print the detected hand landmarks' positions



Try on your self

Hand Count Finger Send To NodeRed

open your command prompt

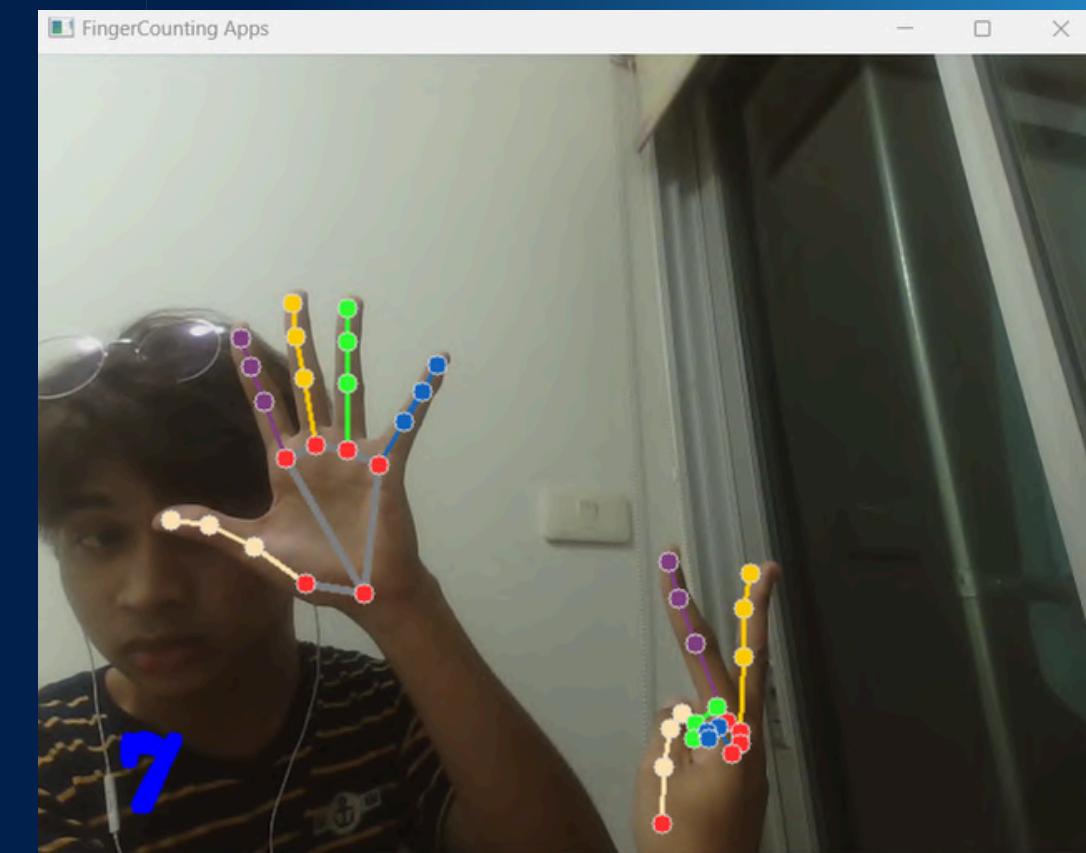
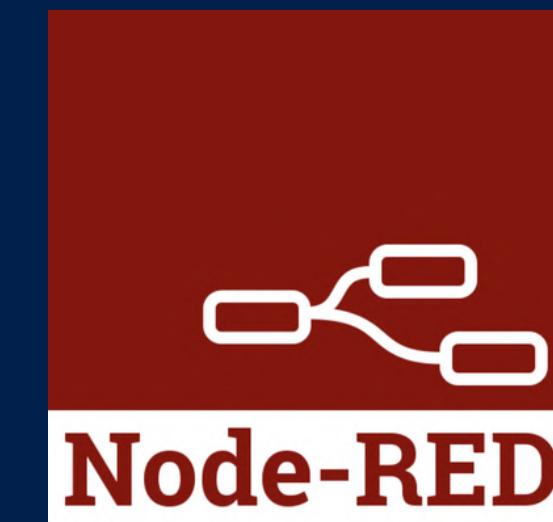
Q1

```
pip install mediapipe  
pip install opencv-python
```

Q2

install NodeRed

```
pip install requests
```

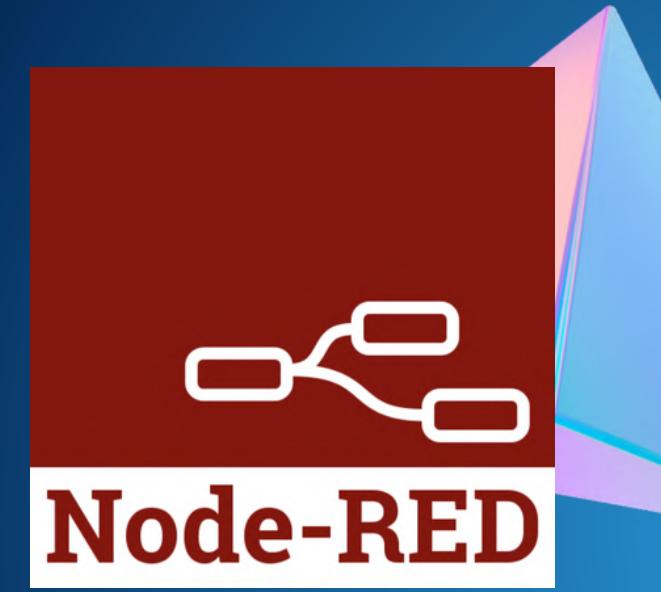


Hand Count Finger Send To NodeRed

Q2 |

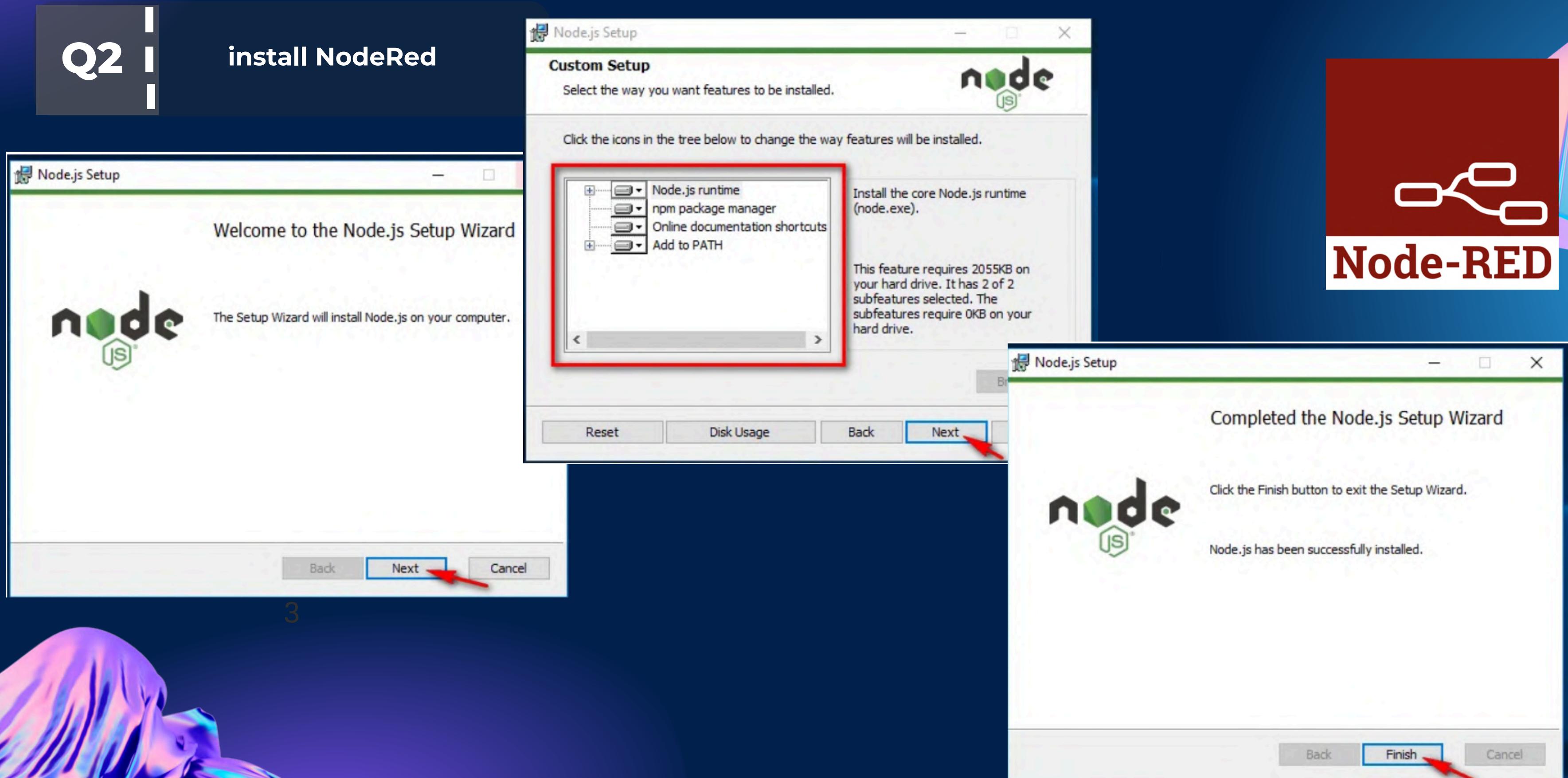
install NodeRed

The screenshot shows the official Node.js website. At the top, there's a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. The FOUNDATION link is highlighted in green. Below the navigation, the Node.js logo is displayed. A sub-header reads "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Underneath, there are two prominent download buttons: "Download for Windows (x64)" which has a red border around the "10.15.0 LTS" button, and another green button next to it labeled "11.7.0 Current". Both buttons have smaller text below them: "Recommended For Most Users" and "Latest Features" respectively. At the bottom of the download section, there are links for "Other Downloads", "Changelog", and "API Docs" for both versions. Below the download section, there's a link to the "Long Term Support (LTS) schedule" and a call to "Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter".



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



Hand Count Finger Send To NodeRed

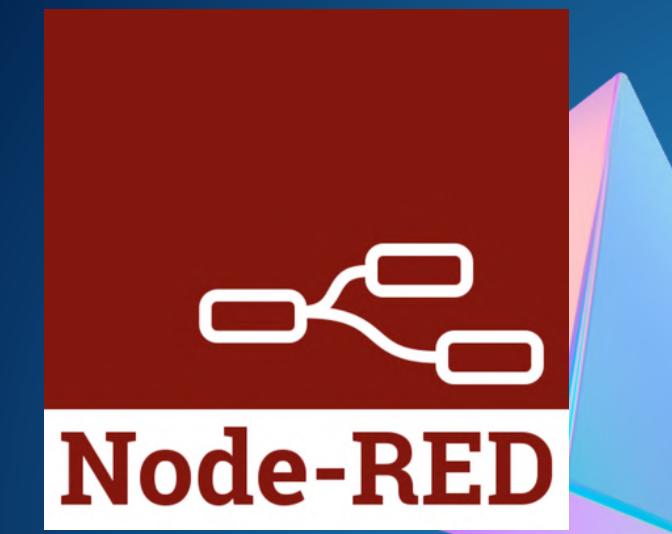
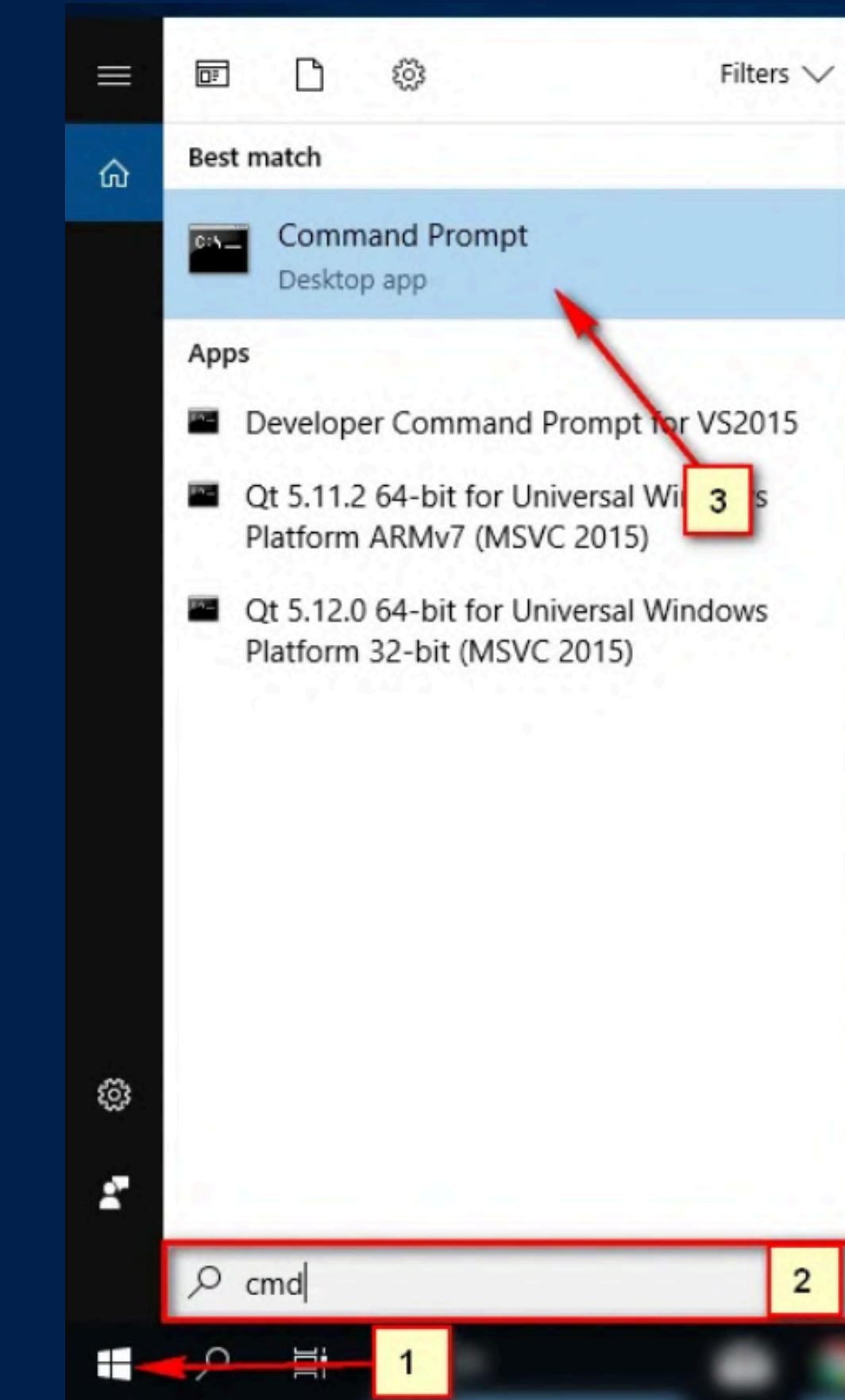


2. Check Version NodeJS with CMD

3

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

3

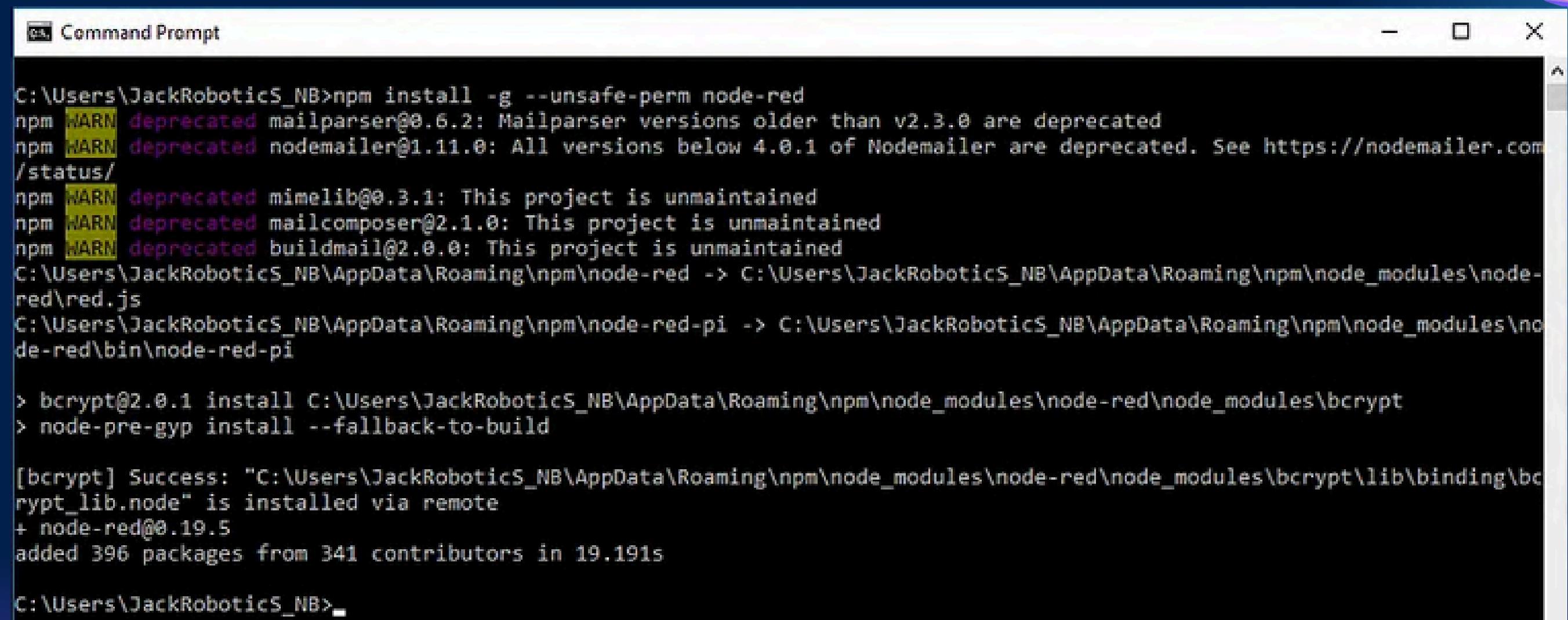


Hand Count Finger Send To NodeRed

Q2

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

3. Install Node-Red with npm



```
C:\Users\JackRobotics_NB>npm install -g --unsafe-perm node-red
npm [WARN] deprecated mailparser@0.6.2: Mailparser versions older than v2.3.0 are deprecated
npm [WARN] deprecated nodemailer@1.11.0: All versions below 4.0.1 of Nodemailer are deprecated. See https://nodemailer.com/about/
npm [WARN] deprecated mimelib@0.3.1: This project is unmaintained
npm [WARN] deprecated mailcomposer@2.1.0: This project is unmaintained
npm [WARN] deprecated buildmail@2.0.0: This project is unmaintained
C:\Users\JackRobotics_NB\AppData\Roaming\npm\node-red -> C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node-red.js
C:\Users\JackRobotics_NB\AppData\Roaming\npm\node-red-pi -> C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi

> bcrypt@2.0.1 install C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt
> node-pre-gyp install --fallback-to-build

[bcrypt] Success: "C:\Users\JackRobotics_NB\AppData\Roaming\npm\node_modules\node-red\node_modules\bcrypt\lib\binding\bcrypt_lib.node" is installed via remote
+ node-red@0.19.5
added 396 packages from 341 contributors in 19.191s

C:\Users\JackRobotics_NB>
```

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

4. Run Node-Red

node-red

Type the command node-red

into cmd, it will look like the picture.

3

```
node-red
C:\Users\JackRobotics_NB>node-red
22 Jan 20:04:21 - [info]

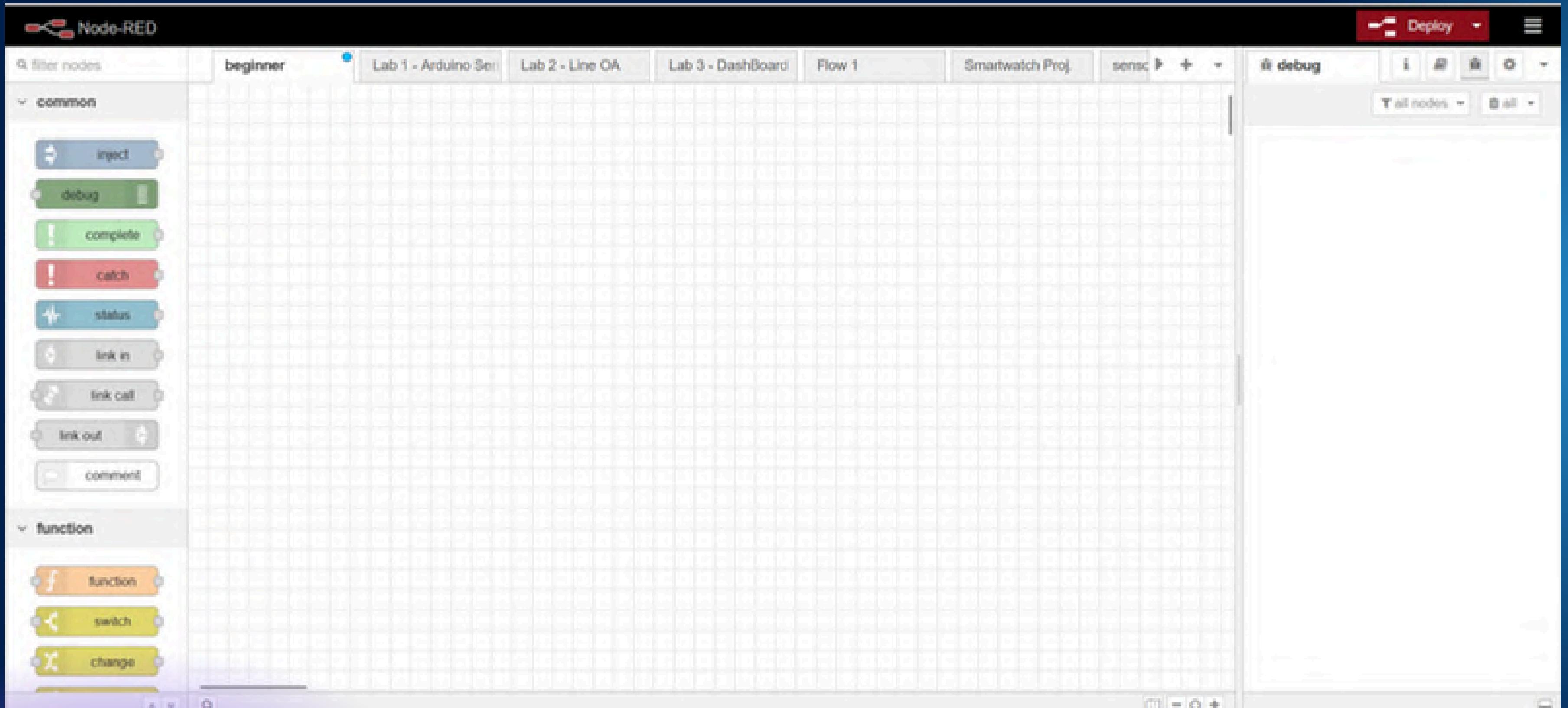
Welcome to Node-RED
-----
22 Jan 20:04:21 - [info] Node-RED version: v0.19.5
22 Jan 20:04:21 - [info] Node.js  version: v10.15.0
22 Jan 20:04:21 - [info] Windows_NT 10.0.17134 x64 LE
22 Jan 20:04:21 - [info] Loading palette nodes
22 Jan 20:04:22 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
22 Jan 20:04:22 - [warn] -----
22 Jan 20:04:23 - [warn] [node-red/tail] Not currently supported on Windows.
22 Jan 20:04:23 - [warn] -----
22 Jan 20:04:23 - [info] Settings file  : C:\Users\JackRobotics_NB\.node-red\settings.js
22 Jan 20:04:23 - [info] Context store  : 'default' [module=memory]
22 Jan 20:04:23 - [info] User directory : C:\Users\JackRobotics_NB\.node-red
22 Jan 20:04:23 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Jan 20:04:23 - [info] Flows file    : C:\Users\JackRobotics_NB\.node-red\flows_JackRobotics_NB.json
22 Jan 20:04:23 - [info] Creating new flow file
22 Jan 20:04:23 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

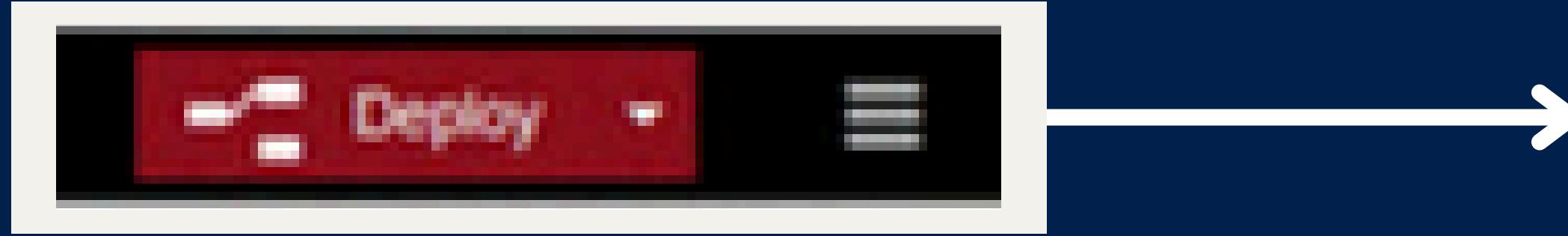
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
22 Jan 20:04:23 - [info] Server now running at http://127.0.0.1:1880/
22 Jan 20:04:23 - [info] Starting flows
22 Jan 20:04:23 - [info] Started flows
-----
```

Ctrl + click

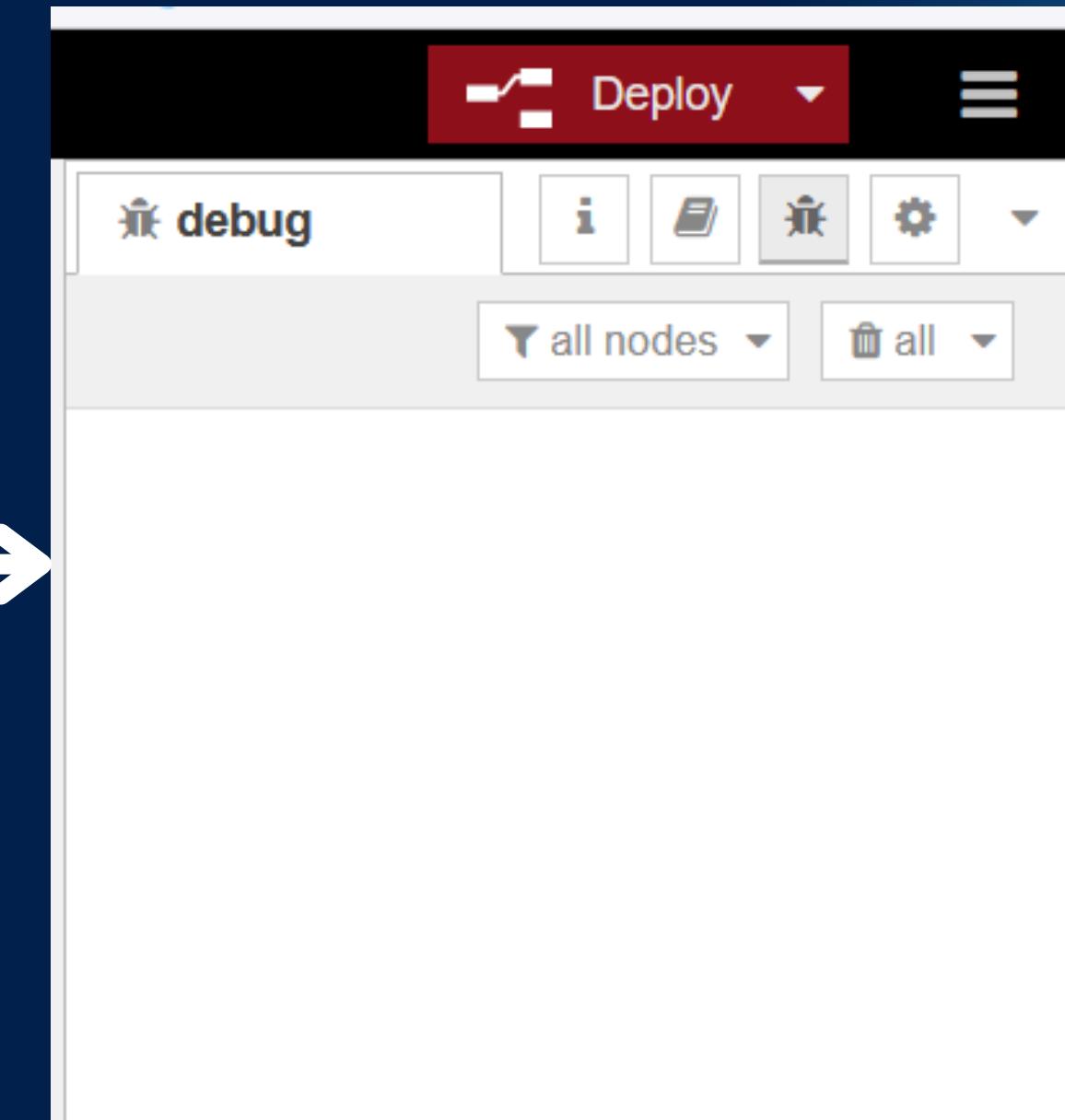


Workspace **Node-Red**

Deploy Button



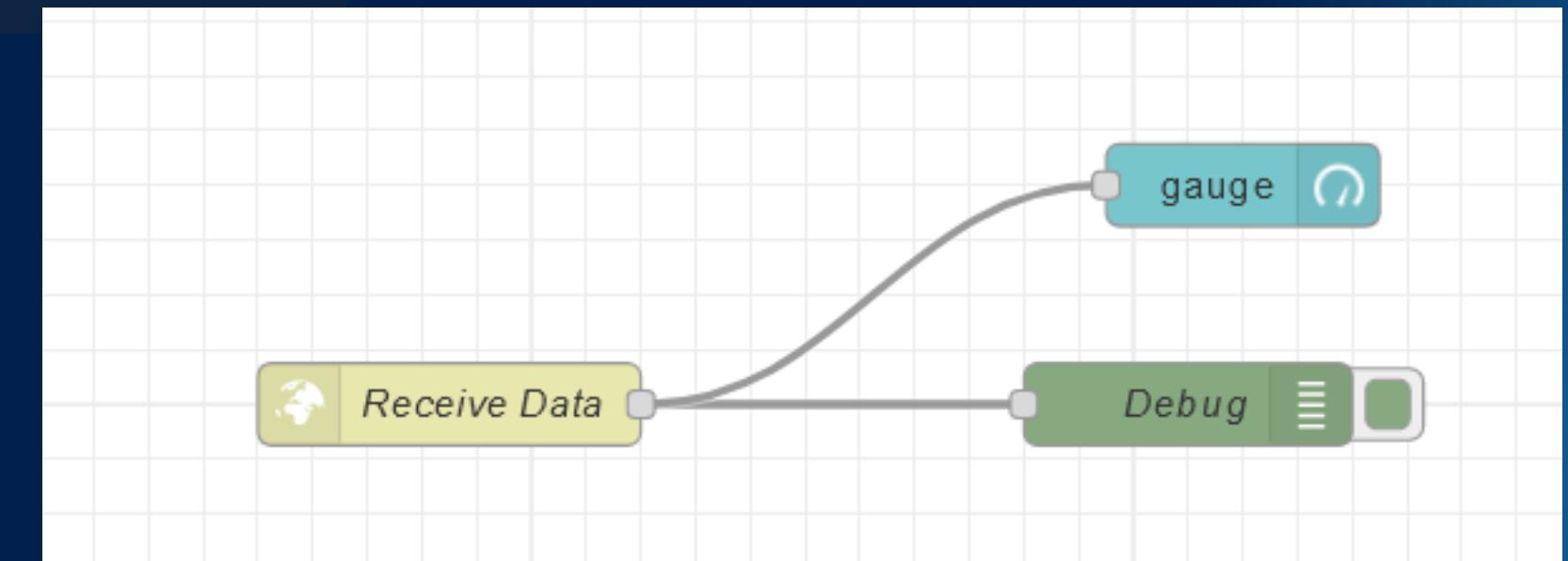
- บุ่มนี้ไว้กดทุกครั้งเวลา อัพเดต **block code**
หลังแก้ไขเสร็จ



นำไว้ดู Debug เวลาเมื่อค่าเข้ามาจากการส่งข้อมูล
Data Value / Struct after processing

Q3

Configure padlet in node red



double-click in nodes Receive Data

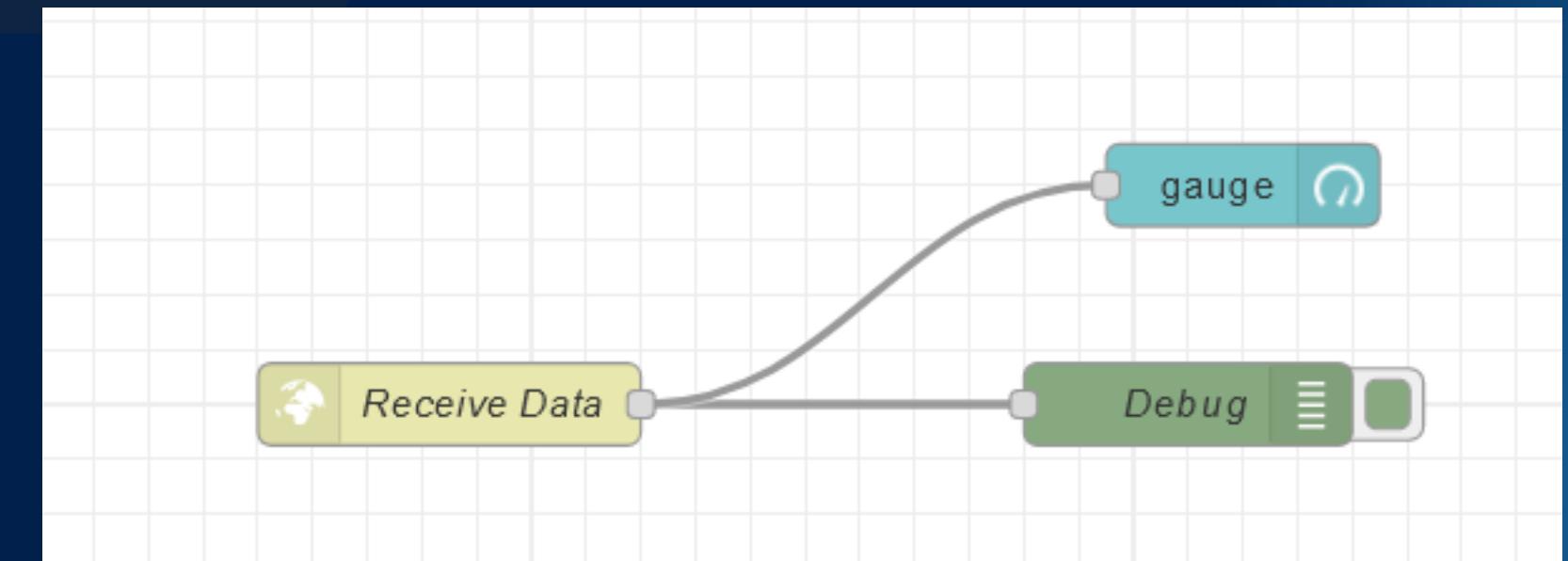
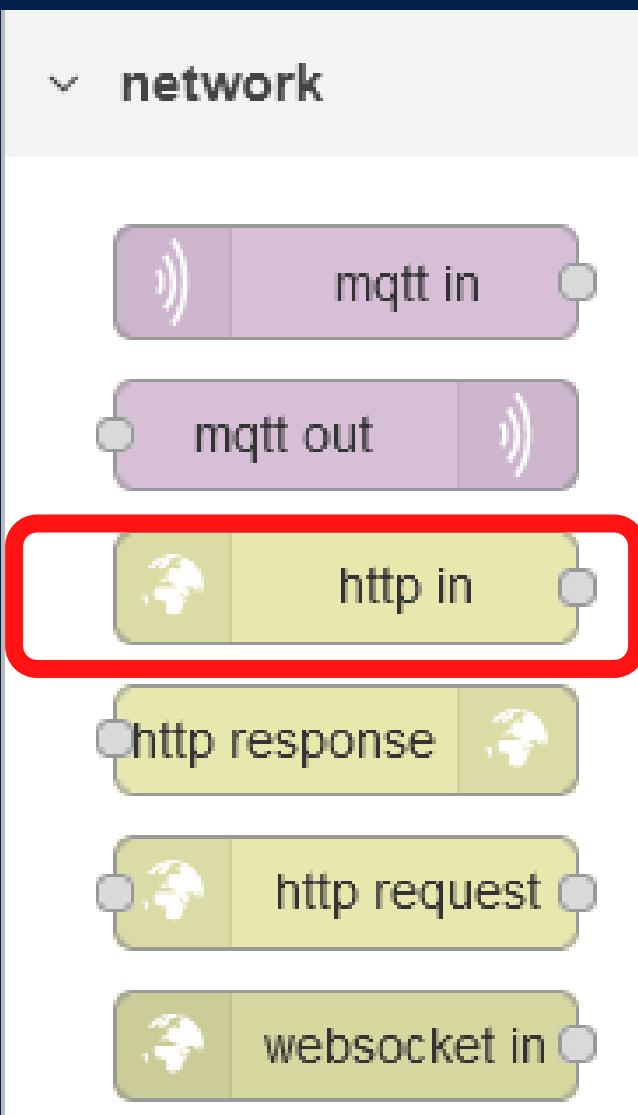
The configuration dialog for the 'Receive Data' node is shown. It includes the following fields:

- Method: POST
- Accept file uploads?
- URL: /receive-data
- Name: Receive Data

A red box highlights the 'URL' input field, with the text "config your URL you need" displayed in red to its right.

Q3

Configure padlet in node red



double-click in nodes Receive Data

The configuration dialog for the 'Receive Data' node is shown. It includes the following fields:

- Method: POST
- Accept file uploads?
- URL: /receive-data
- Name: Receive Data

A red box highlights the 'URL' input field, with the text "config your URL you need" displayed in red to its right.

Q3

Hand Count Coding

Our Wisdom

URL

```
1 import cv2
2 import mediapipe as mp
3 import requests
4 url = 'http://127.0.0.1:1880/receive-data' URL
5
6 mp_drawing = mp.solutions.drawing_utils
7 mp_drawing_styles = mp.solutions.drawing_styles
8 mp_hands = mp.solutions.hands
9
10 capture = cv2.VideoCapture(0)
11 with mp_hands.Hands(
12     model_complexity=0,
13     min_detection_confidence=0.5,
14     min_tracking_confidence=0.5) as hands:
15     while capture.isOpened():
16         success, image = capture.read()
17         if not success:
18             print('Ignored empty webcam\'s frame')
19             continue
20         image.flags.writeable = False
21         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
22         results = hands.process(image)
23
24         image.flags.writeable = True
25         image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
26
27         fingerCount = 0
```

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

```
29     if results.multi_hand_landmarks:
30         for hand_landmarks in results.multi_hand_landmarks:
31             handIndex = results.multi_hand_landmarks.index(hand_landmarks)
32             handLabel = results.multi_handedness[handIndex].classification[0].label
33
34             handLandmarks = []
35
36             for landmarks in hand_landmarks.landmark:
37                 handLandmarks.append([landmarks.x, landmarks.y])
38
39             if handLabel == "Left" and handLandmarks[4][0] > handLandmarks[3][0]:
40                 fingerCount = fingerCount + 1
41             elif handLabel == "Right" and handLandmarks[4][0] < handLandmarks[3][0]:
42                 fingerCount = fingerCount + 1
43
44             if handLandmarks[8][1] < handLandmarks[6][1]:
45                 fingerCount = fingerCount + 1
46             if handLandmarks[12][1] < handLandmarks[10][1]:
47                 fingerCount = fingerCount + 1
48             if handLandmarks[16][1] < handLandmarks[14][1]:
49                 fingerCount = fingerCount + 1
50             if handLandmarks[20][1] < handLandmarks[18][1]:
51                 fingerCount = fingerCount + 1
52
53             mp_drawing.draw_landmarks(
54                 image,
55                 hand_landmarks,
56                 mp_hands.HAND_CONNECTIONS,
57                 mp_drawing_styles.get_default_hand_landmarks_style(),
58                 mp_drawing_styles.get_default_hand_connections_style()
59             )
60
```

Q4

FingerCount Send To NodeRed Coding

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

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

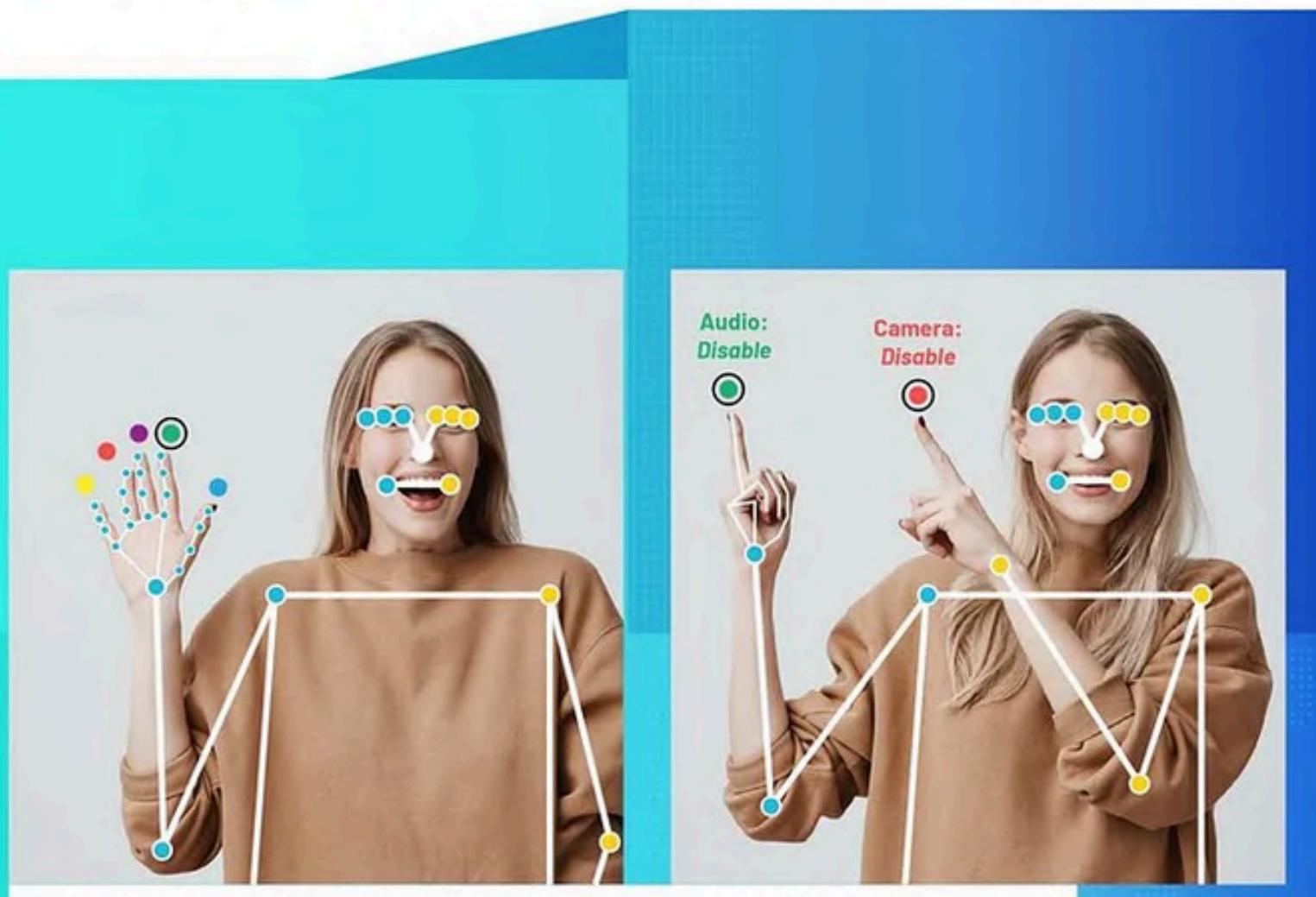


github code :

https://github.com/TheBoyZ/Cod eBotix_Camp_Image

Intelligence

Google's MediaPipe Holistic Interface

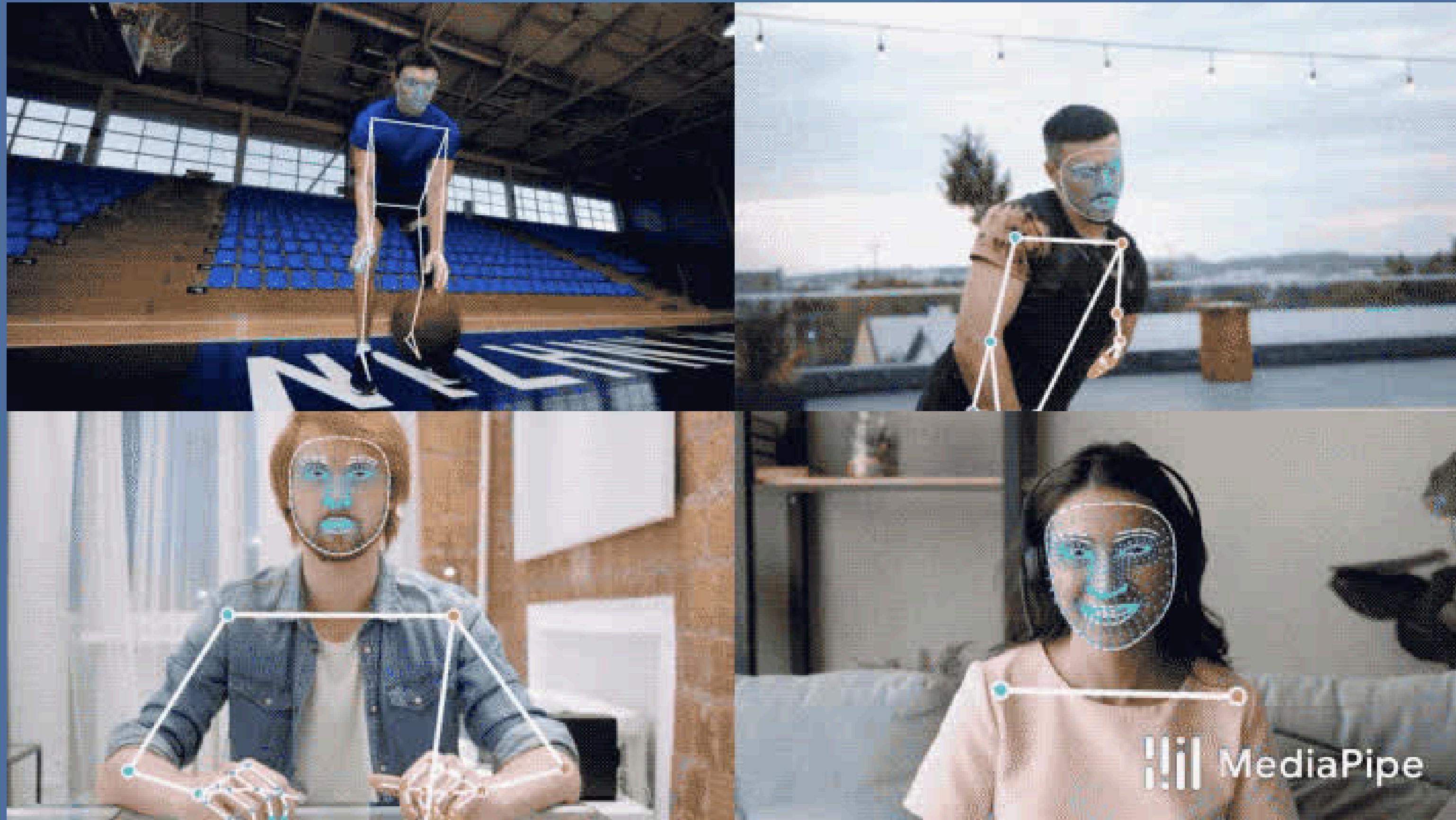


This gesture control and command technique will take us to the next level. And allow us to use new innovations that other devices cannot do

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 នីរ ម៉ោច្ចាយ ហើយ ខ្លា

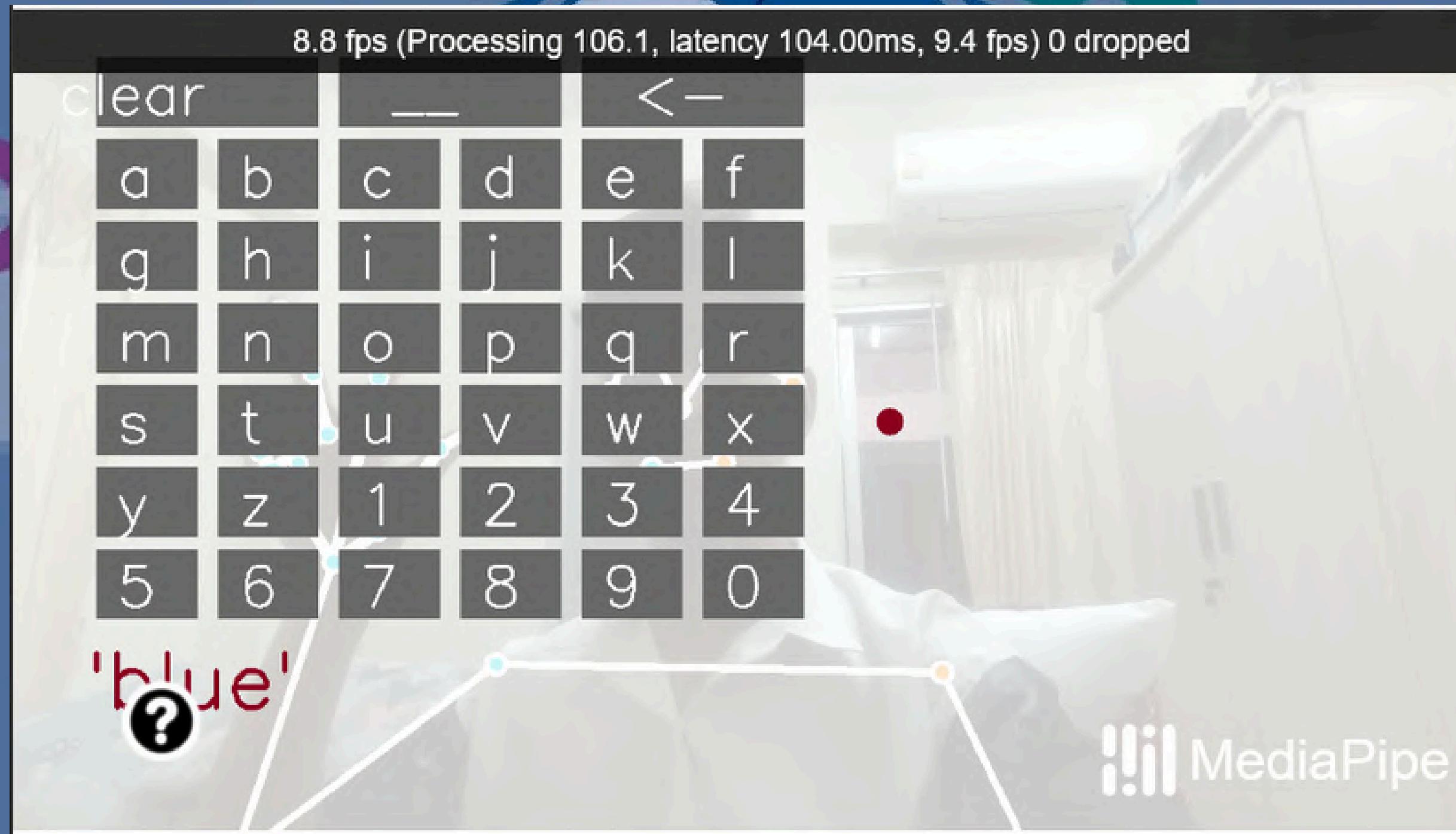
2

3 នីរ ម៉ោច្ចាយ

3

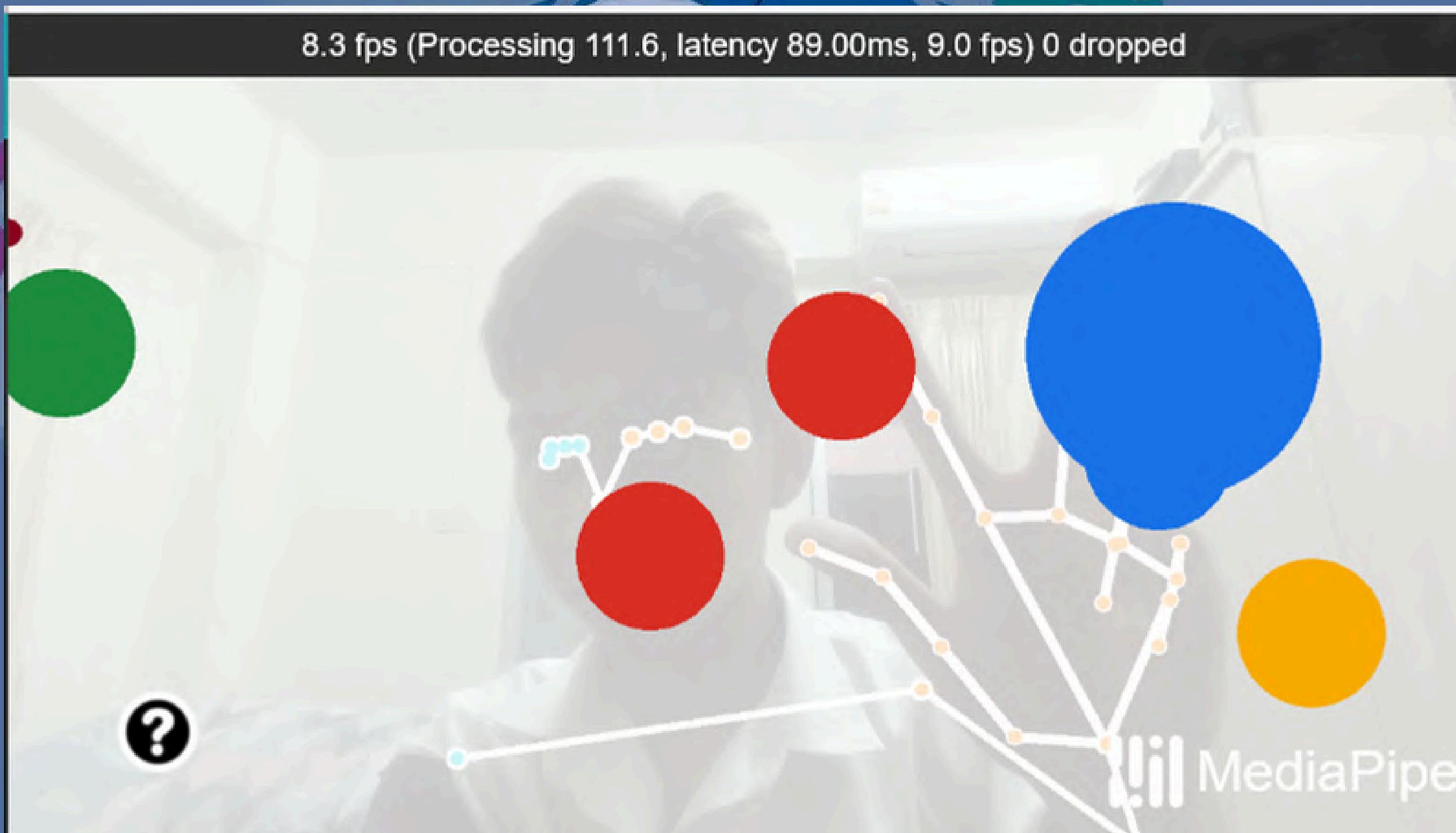
3 នីរ ម៉ោខ្លា

Holistic demo



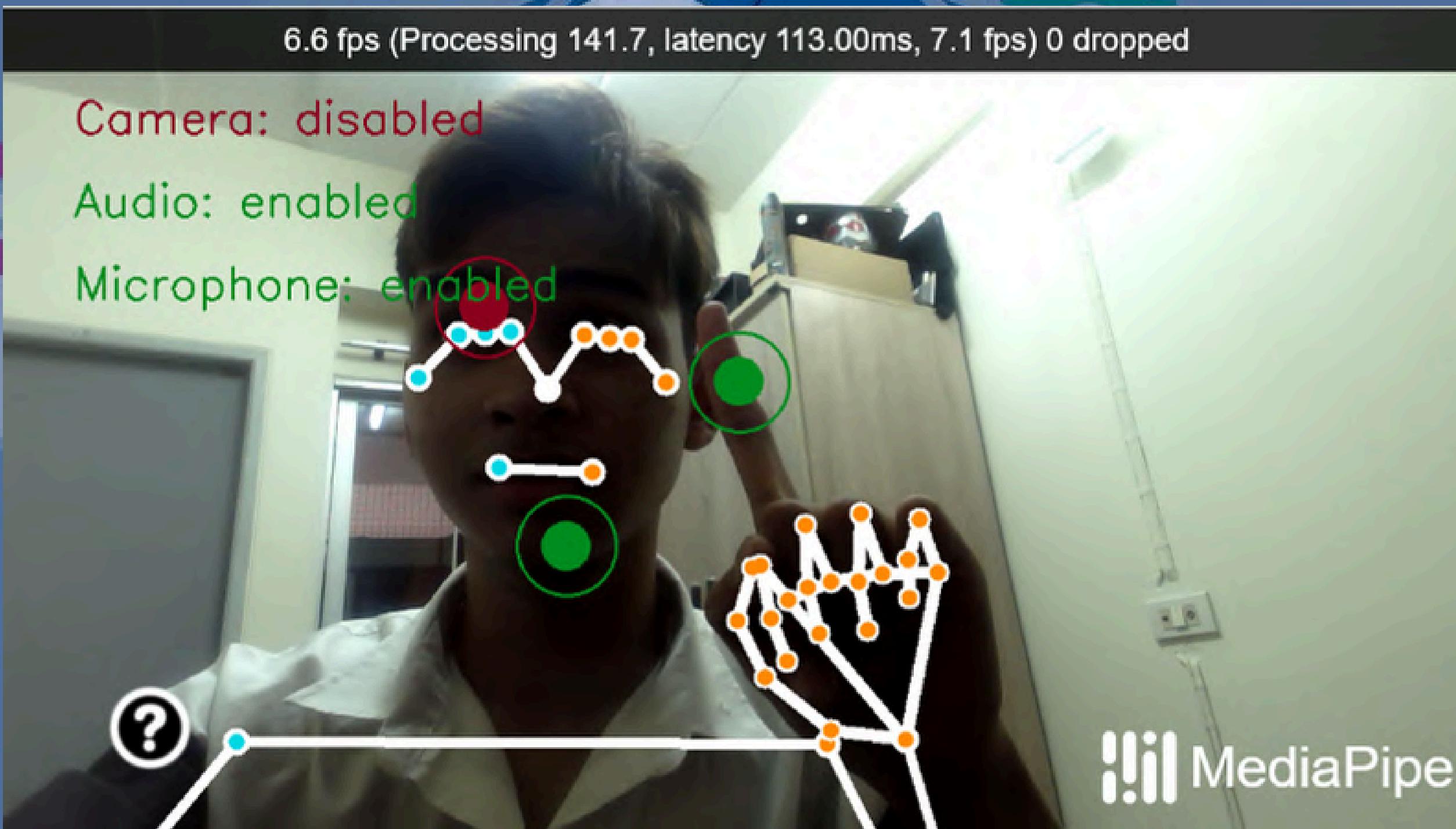
ลองเขียนชื่อตัวเอง เป็นภาษาอังกฤษ

Holistic demo



สามารถใช้มือแทน เม้าส์ในการลากกรุ๊ปสีจัดหมวดหมู่ได้

Holistic demo



Disable / Enable mode

Anaconda Navigator



Anaconda

<https://www.anaconda.com> ::

Anaconda | The World's Most Popular Data Science Platform

Anaconda is the birthplace of Python data science. We are a movement of data scientists, data-driven enterprises, and open source communities.

Free Download

Conda is an open-source package

The screenshot shows the Anaconda Distribution website with a dark green background featuring a stylized brain and neural network graphic. At the top, there is a navigation bar with the Anaconda logo, 'ANACONDA.', and links for 'Enterprise', 'Pricing', 'Resources', and 'About'. Below the navigation, the text 'Anaconda Distribution' is displayed in green. A large white button with the text 'Free Download' in bold is centered. Below this button, the text 'Everything you need to get started in data science on your workstation.' is written. To the right of this text, there is a bulleted list of four features, each preceded by a green checkmark: 'Free distribution install', 'Thousands of the most fundamental DS, AI, and ML packages', 'Manage packages and environments from desktop application', and 'Deploy across hardware and software platforms'. At the bottom of the section, there are two green buttons: 'Code in the Cloud' and 'Download'. Below these buttons, there is a link 'Get Additional Installers' followed by icons for Windows, Mac, and Linux.

ANACONDA. Enterprise Pricing Resources About

Anaconda Distribution

Free Download

Everything you need to get started in data science on your workstation.

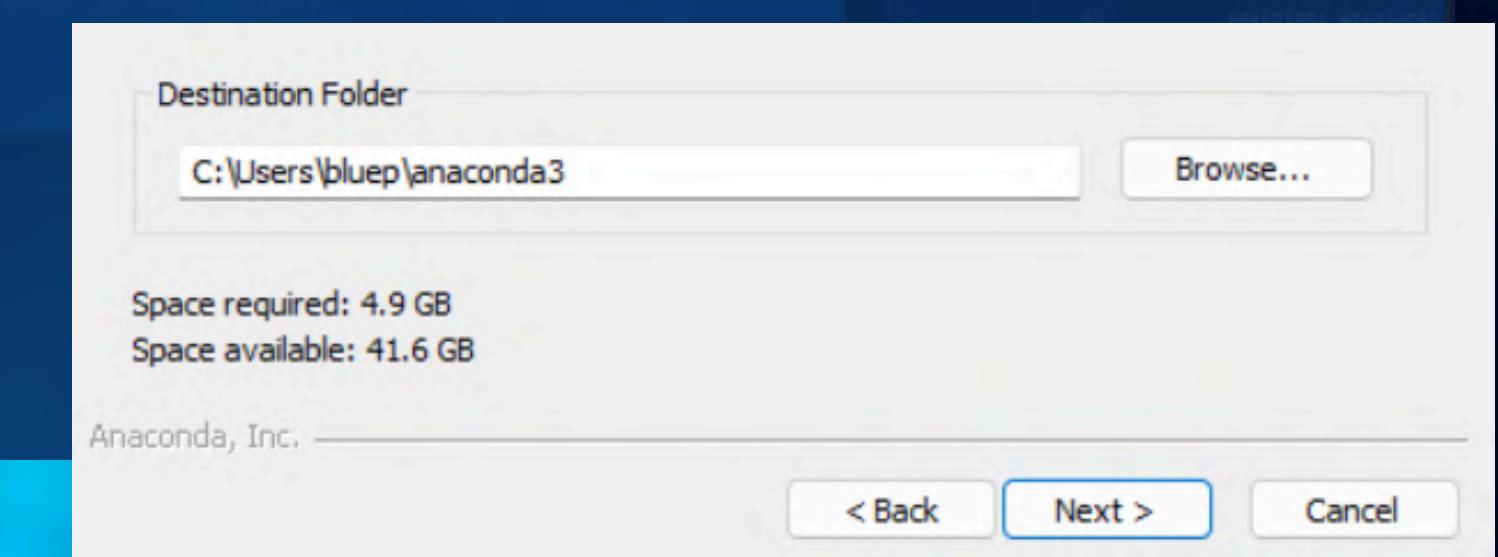
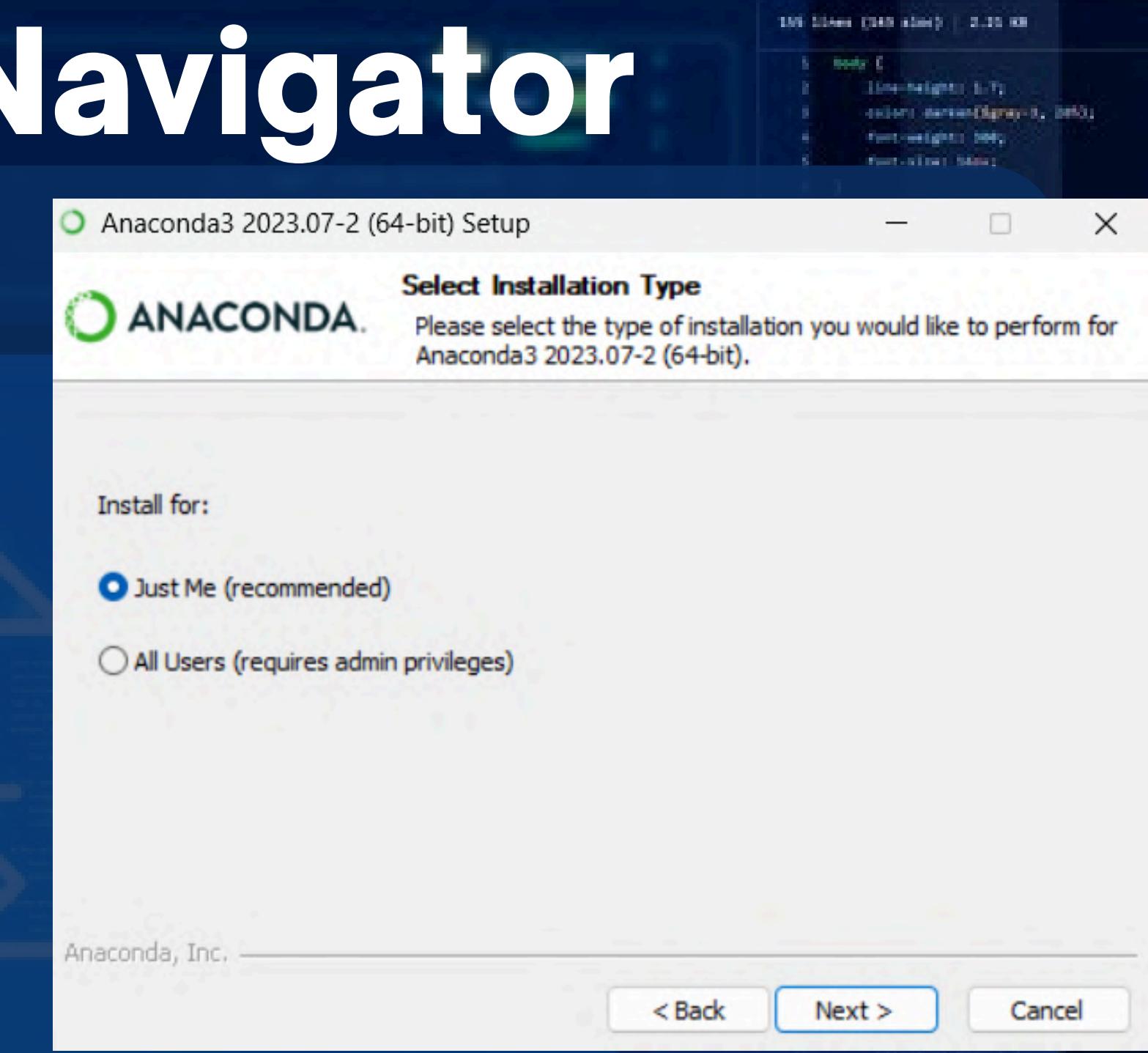
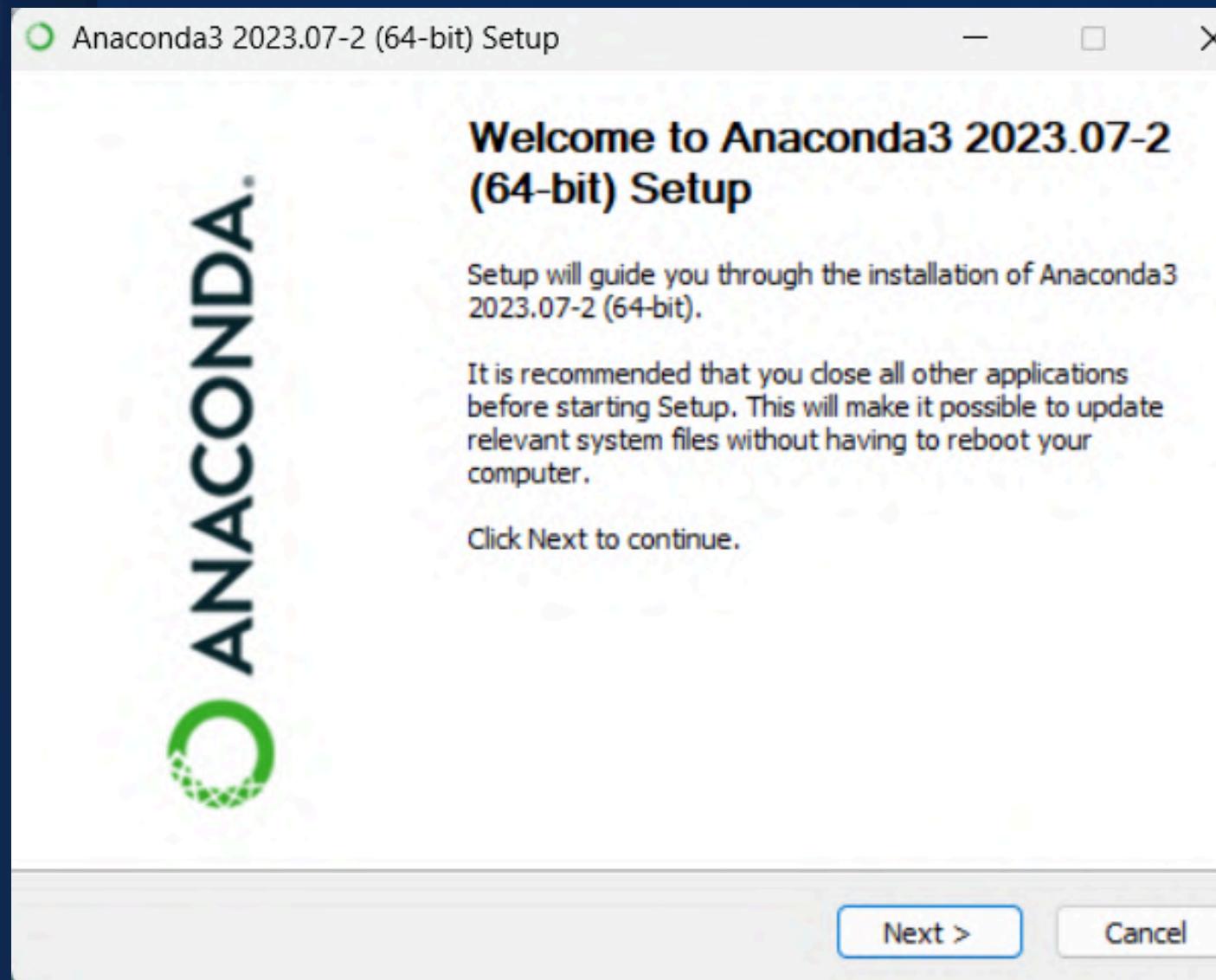
- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Code in the Cloud Download

Get Additional Installers

Windows Mac Linux

Anaconda Navigator



Anaconda Navigator

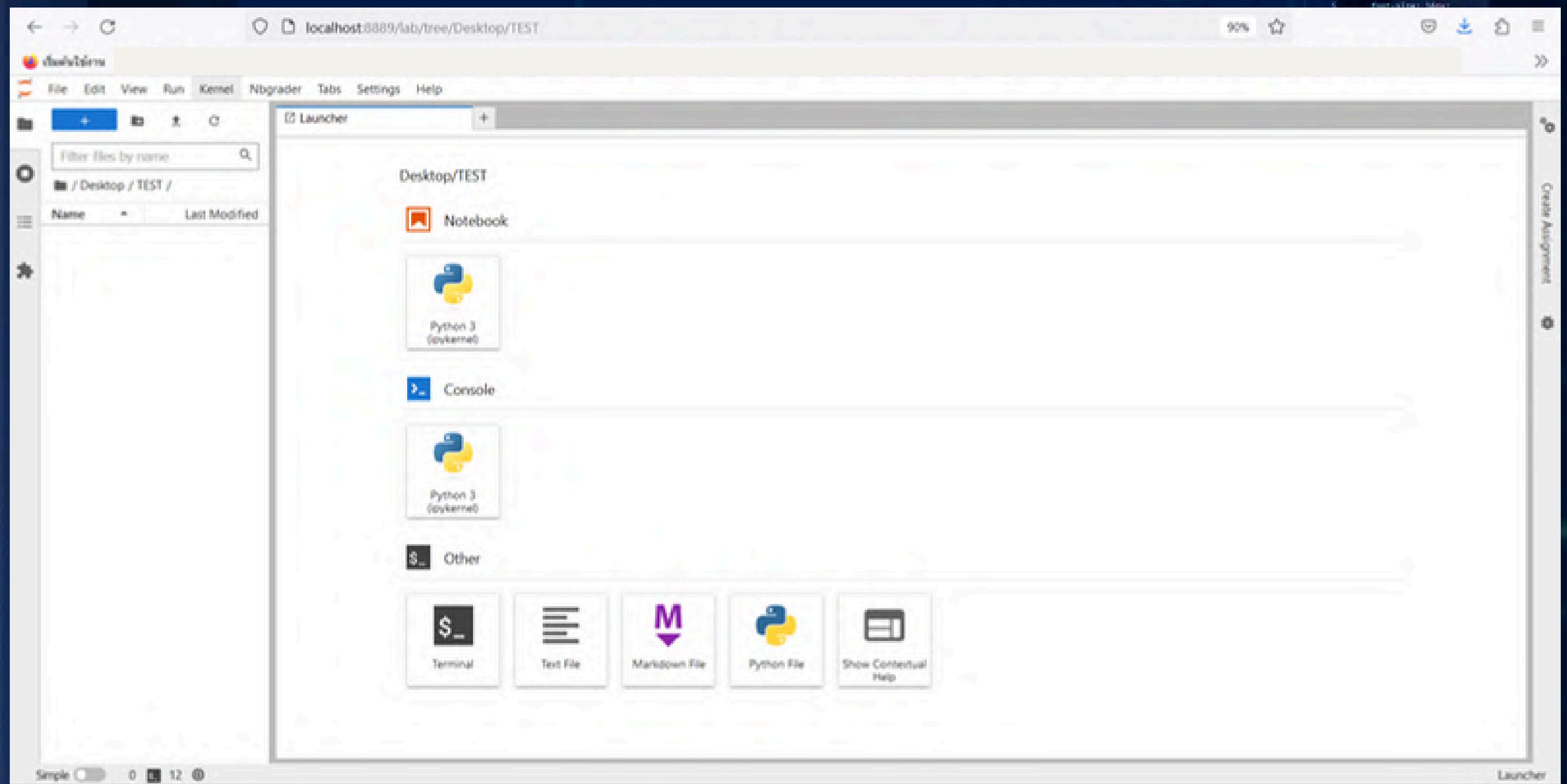
The screenshot shows the Anaconda Navigator application window. On the left is a sidebar with navigation links: Home, Environments, Learning, and Community. A prominent callout box highlights the "Anaconda Notebooks" section, which says "Cloud notebooks with hundreds of packages ready to code" and has a "Learn More" button. Below this are links for "A Full Python IDE directly from the browser", "Documentation", and "Anaconda Blog". Social media icons for Twitter, LinkedIn, and GitHub are at the bottom.

The main area displays a grid of 12 tool cards:

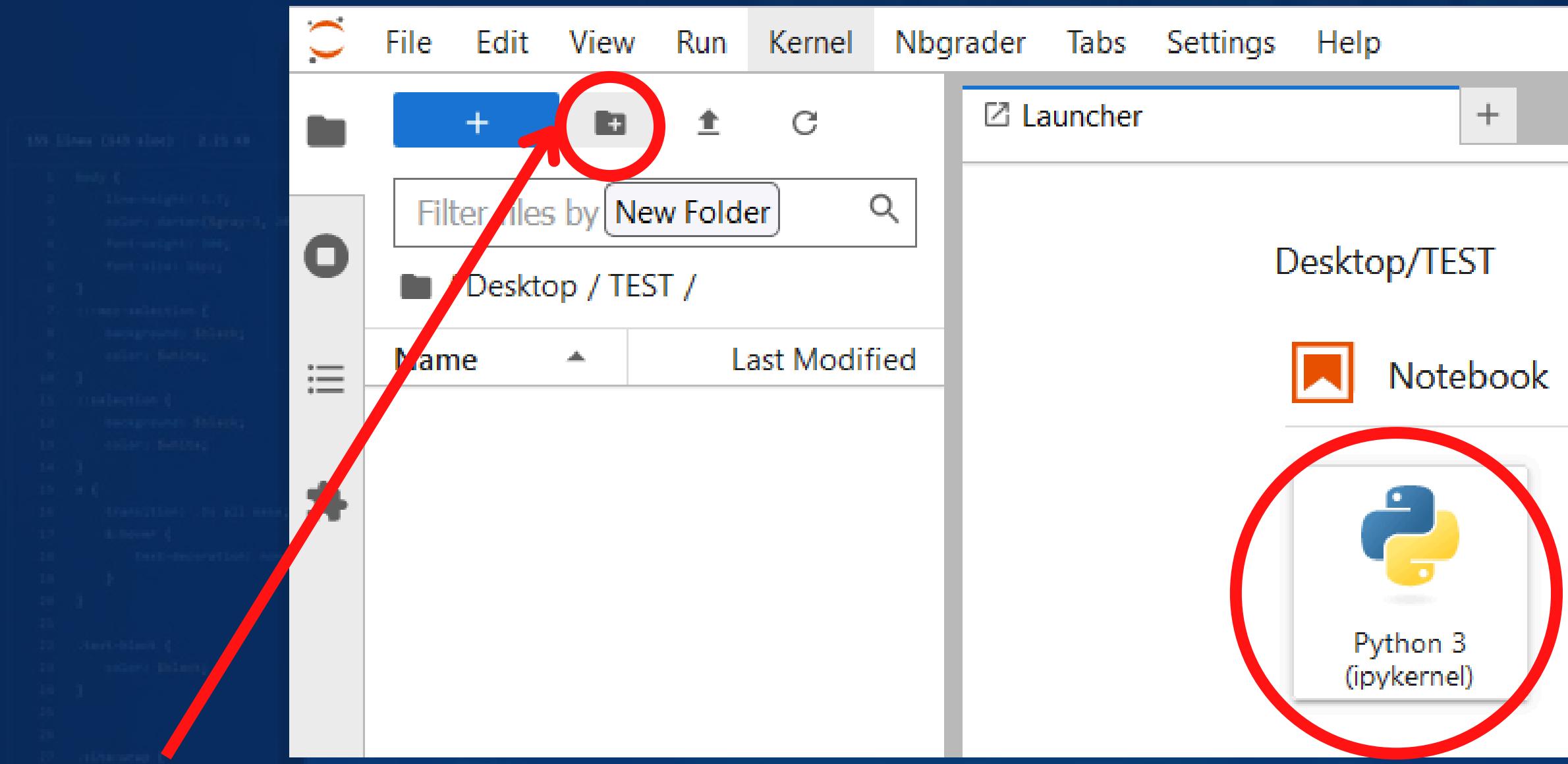
- Datavant**: An IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment. Includes "Install" and "Launch" buttons.
- CMDShell Prompt**: Run a command-line terminal with your current environment from Navigator activated. Includes "Launch" button.
- JupyterLab**: An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook architecture. Includes "Launch" button.
- Notebook**: Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Includes "Launch" button.
- Powershell Prompt**: Run a PowerShell terminal with your current environment from Navigator activated. Includes "Launch" button.
- Qt Console**: PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical controls, and more. Includes "Launch" button.
- Spyder**: Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features. Includes "Launch" button.
- VS Code**: Streamlined code editor with support for development operations like debugging, task running and version control. Includes "Launch" button.
- Databricks**: Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Databricks online for free. Includes "Launch" button.
- IBM Watson Studio Cloud**: IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models, Prepare data and build models, using open source data science tools or visual modeling. Includes "Launch" button.
- ORACLE Cloud Infrastructure**: Oracle Data Science Service. OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on the cloud with your favorite open-source tools. Includes "Launch" button.
- Globoviz**: Multidimensional data visualization across files. Explore relationships within and among related datasets. Includes "Install" button.
- Orange 3**: Component-based data mining framework. Data visualization and data analysis for machine learning. Includes "Launch" button.
- PyCharm Professional**: A full-fledged IDE by JetBrains for both scientific and web Python development. Includes "Launch" button.
- RStudio**: A set of integrated tools designed to help you be more productive with R. Includes RStudio Server Pro. Includes "Launch" button.

A small floating terminal window in the top right corner shows a command-line interface with the output of "ls" command.

Anaconda Navigator



Anaconda Navigator

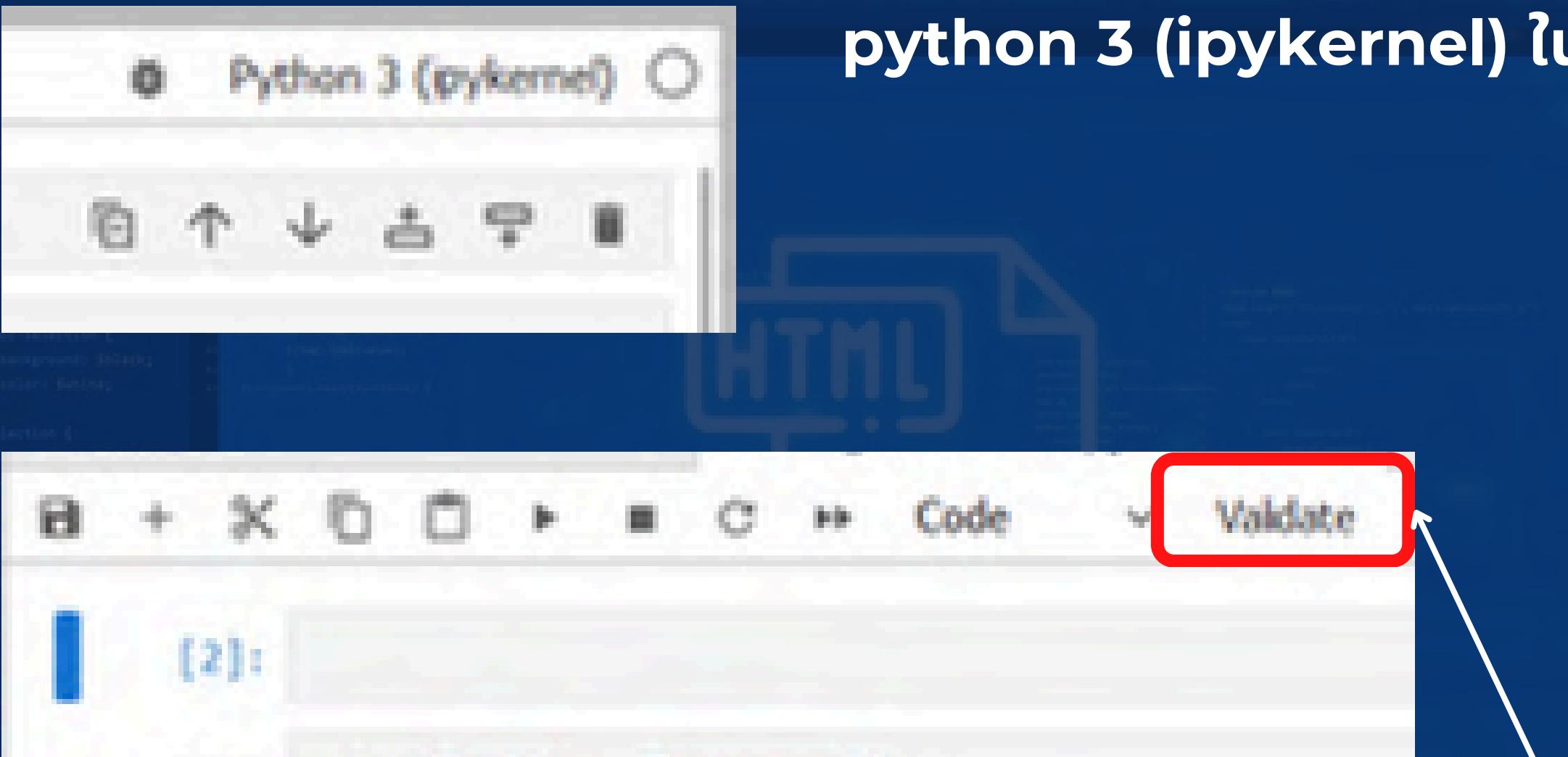


สร้าง Floder ใหม่

สร้าง สร้างไฟล์ .ipynb เพื่อเขียน
โปรแกรมใน jupyter notebook

Anaconda Navigator

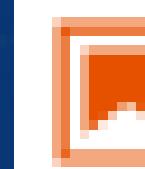
python 3 (ipykernel) ในการรันโค้ด



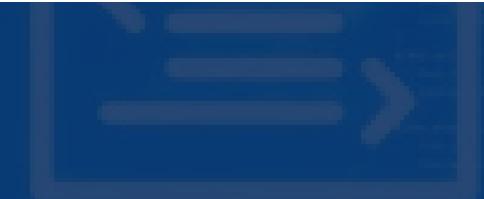
ใช้สำหรับการทดสอบ save ไฟล์
ตรวจสอบความถูกต้องทั้งหมด

Jupyter Notebook

image processing



ch1_helloworld.ipynb



Jupyter Notebook

image processing

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

- ทดสอบ test jupyter Lab 1

[6]: a = 4
[7]: b = 2

part 1

[9]: c = a // b
c
[9]: 2

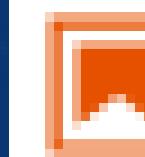
[11]: c = a * b
c
[11]: 8

part2

[12]: print(c)
8
[13]: b = 10
[14]: b
[14]: 10
```

Jupyter Notebook

image processing



ch2_output_python.ipynb



Jupyter Notebook

image processing

```
[1]: 2 ** 5  
[1]: 32
```

comment

```
[2]: n = 20.5  
      dn = 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 = 'theeranet'  
      sur = 'chuaiipayung'  
      name + sur
```

```
[16]: 'theeranet chuaiipayung'
```

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
```

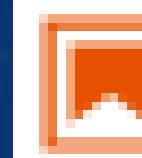
```
[19]: print('Hi {} {:.1f}'.format(name , sc))  
Hi Blue 3.7
```

```
[21]: b = 280903  
print('data = {:.2f}'.format(b))  
data = 280,903.00
```

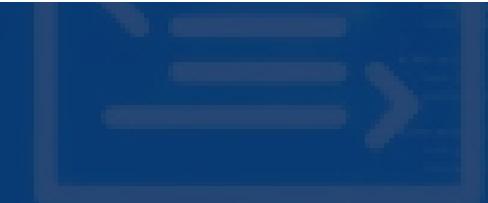
```
[ ]:
```

Jupyter Notebook

image processing



ch3_basic_python.ipynb



Jupyter Notebook

image processing

The screenshot shows a Jupyter Notebook interface with the following content:

- Python เบื้องต้น** (Python Basics) section expanded.
- In [3]:**

```
name = 'theeramet'  
name = 'scimath'  
print(name)
```

 Output: scimath
- In [4]:** เลือก操作符ดำเนิน (Operators)
+ * /
% หารเหลือ
// หารปั๊บเหลือ
** ยกกำเนิด
- In [5]:** a = 10
- In [6]:** b = a / 2
b
- In [7]:** 5.0
- In [8]:** 5 // 2
- In [9]:** 2
- In [10]:** 5 % 2
- In [11]:** 1
- In [12]:** 2 ** 3
- In [13]:** 8

On the right side, there is a sidebar with file navigation:

- File
- File highlight (0-7)
- Search (Search Displayed, Search, Find, Find next, Find previous, Find and replace, Find and replace next, Find and replace previous)
- Font weight (Bold)
- Font size (16px)

Jupyter Notebook

image processing

Variable

```
[16]:  
    ราก = 100  
    รากที่ = ราก / 4  
    ผลลัพธ์ = รากที่  
    print(ผลลัพธ์)  
  
25.0
```

```
[17]:  
    2r = 4  
    pass
```

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

Condition

```
[1]:  
    > < >= <= !=
```

```
[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
```

```
[29]:  
    leftFootprint = {  
        "x": 100, "y": 100},
```

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)
```

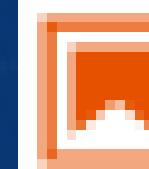
```
27
```

```
171
```

```
18.4
```

Jupyter Notebook

image processing



ch4_library.ipynb



Jupyter Notebook

image processing

The screenshot shows a Jupyter Notebook interface with three code cells displayed:

- [2]:** `!pip show pandas`
Output:
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`
Output:
packages in environment at C:\Users\bluep\anaconda3:

Name Version Build Channel
pandas 1.5.3 py310h4ded8f06_0
- [4]:** `!pip show matplotlib`
Output:
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
```

	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

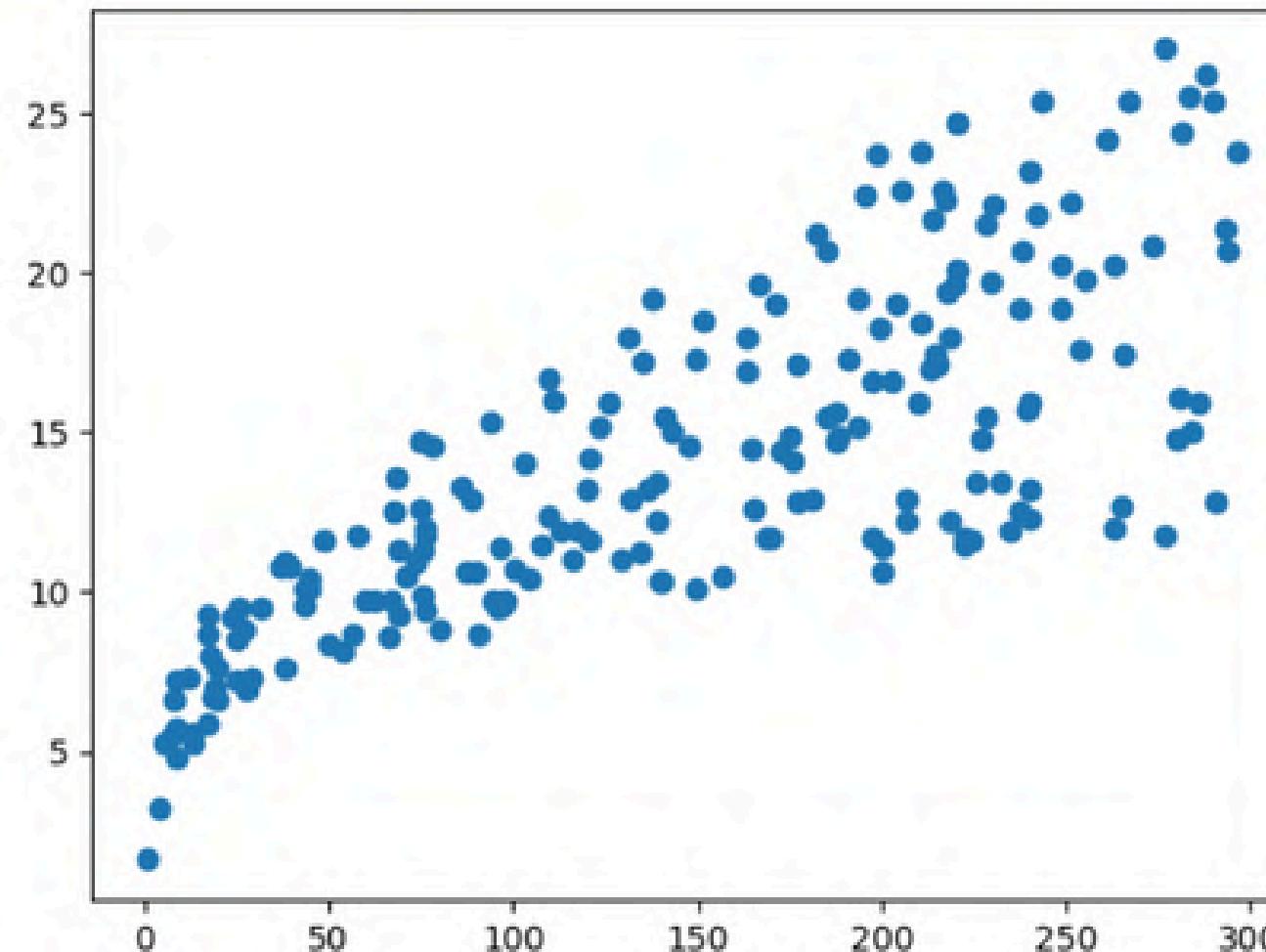
Jupyter Notebook

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],  
             [8, 9, 1, 7, 2]])
```

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]]))
```

```
100 lines (248 chars) | 0.00 KB  
1 body {  
2   line-height: 1.7;  
3   color: inherit; display: block;  
4   font-weight: 600;  
5   font-size: 0.9em;  
6 }  
7 .l-line{color:#000;}
```

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)

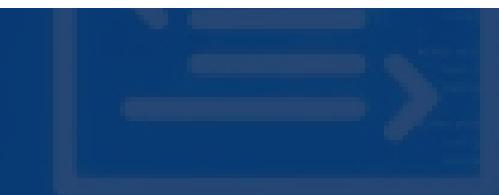
[ ]:
```

Jupyter Notebook

image processing



ch5_matplotlib.ipynb



Jupyter Notebook

image processing

Matplotlib

ໃຫ້ຮັບກິນກົນ 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
```

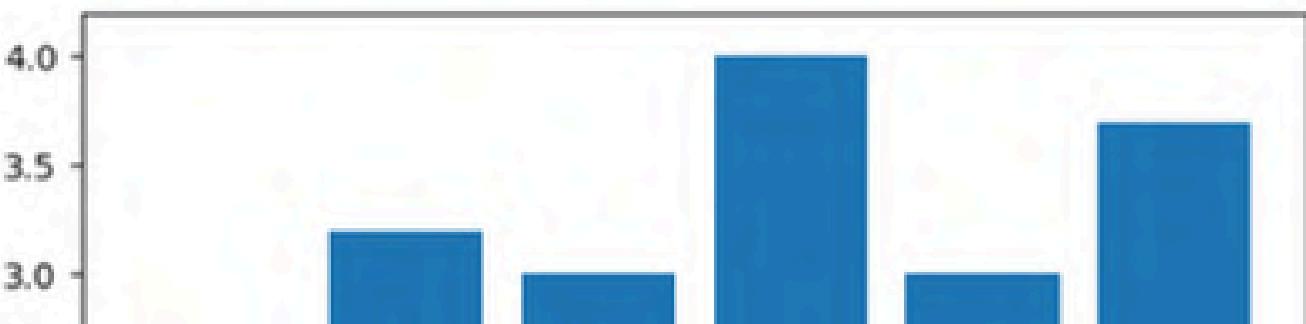
```
[5]: import matplotlib.pyplot as plt
```

```
from matplotlib import pyplot as plt # ເມືອນແບບນີ້ໄດ້
```

Bar

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

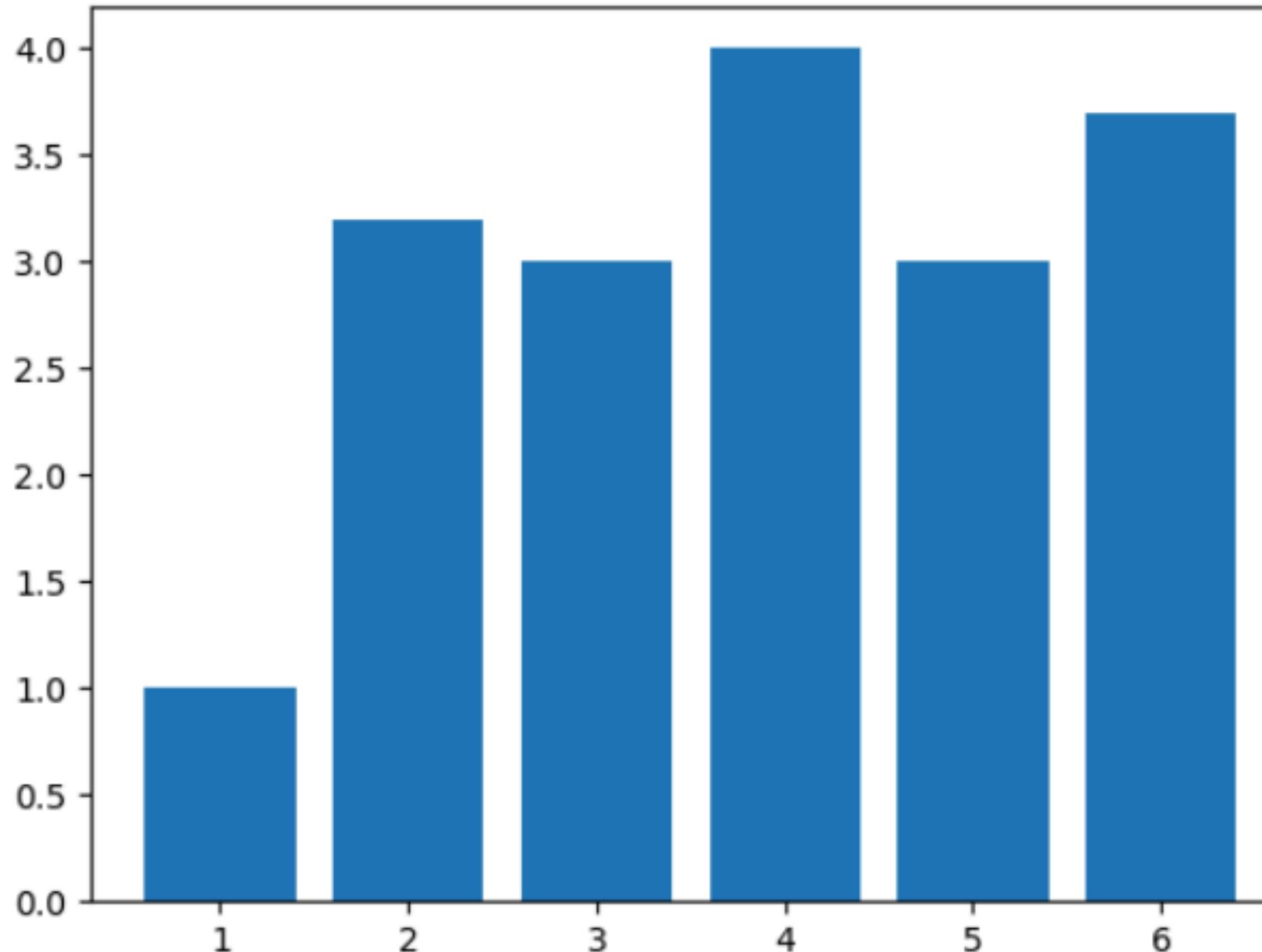
```
[7]: plt.bar(x, y_data)  
plt.show() # ພັດທະນາກຳ x ແລະ y ພາວີ່ງ
```



Jupyter Notebook

image processing

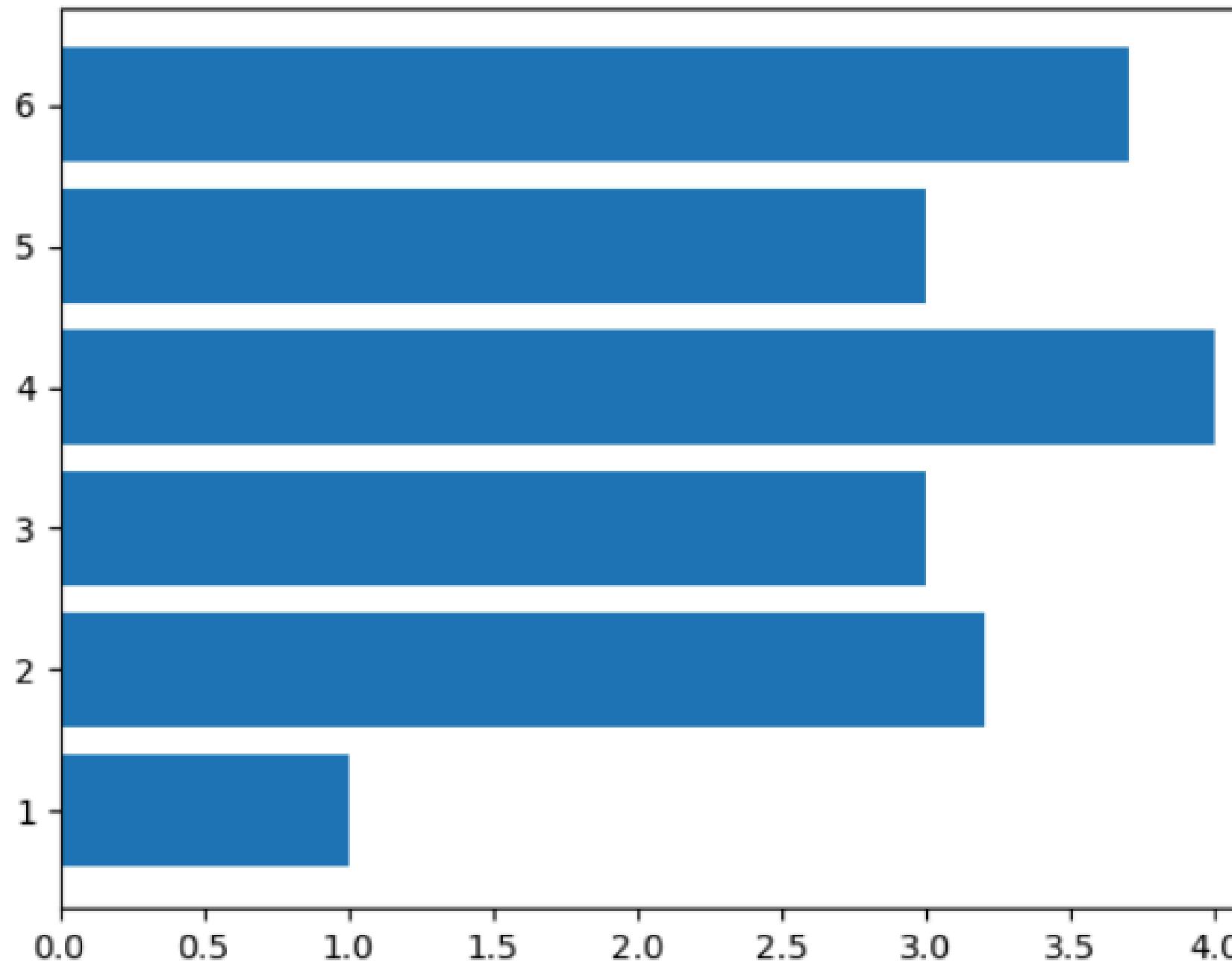
```
[7]: plt.bar(x, y_data)
plt.show() # พล็อตกราฟแกน x แกน y แนวตั้ง
```



Jupyter Notebook

image processing

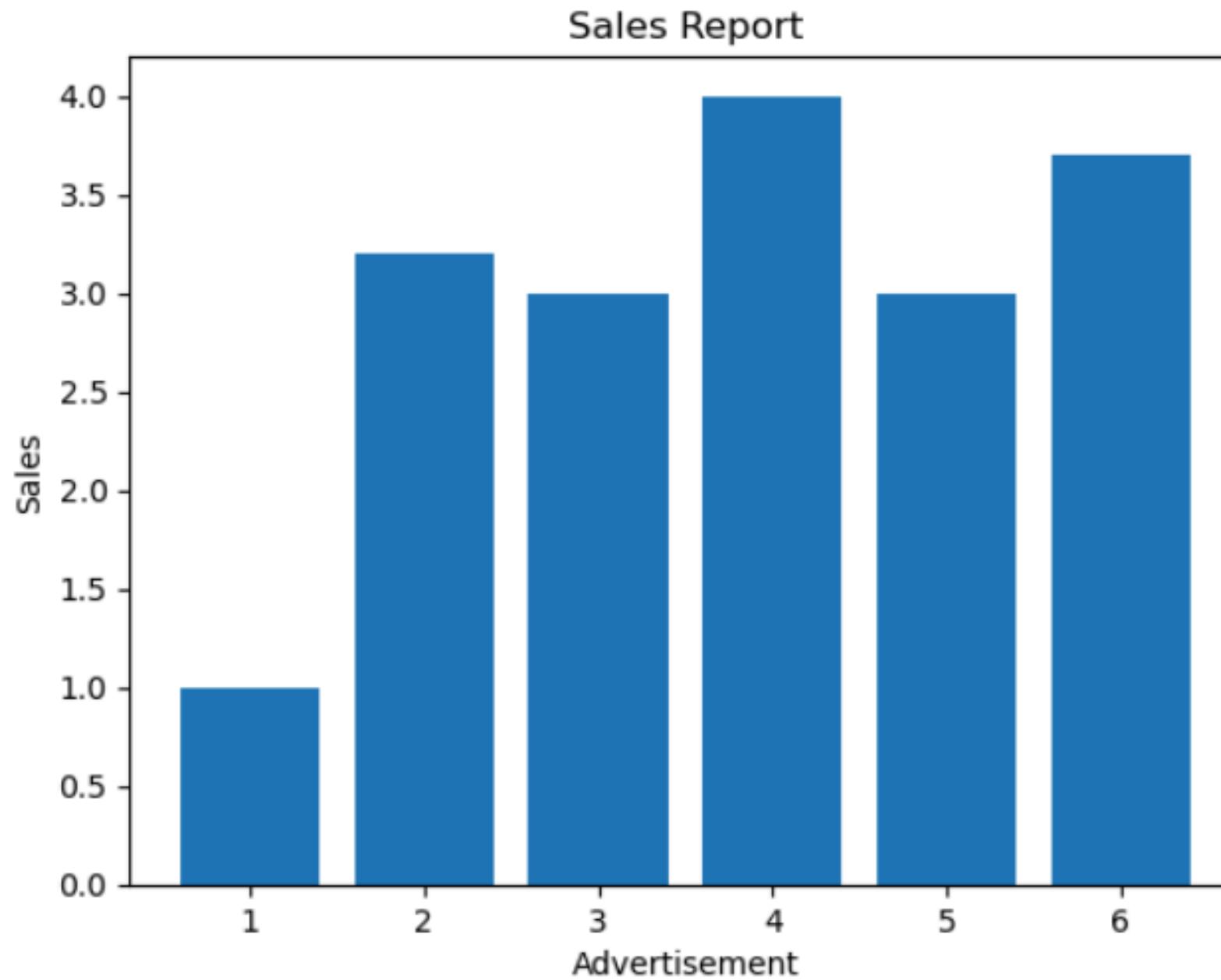
```
[8]: plt.barh(x, y_data) # พล็อตกราฟแท่ง แนวนอน  
plt.show()
```



Jupyter Notebook

image processing

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

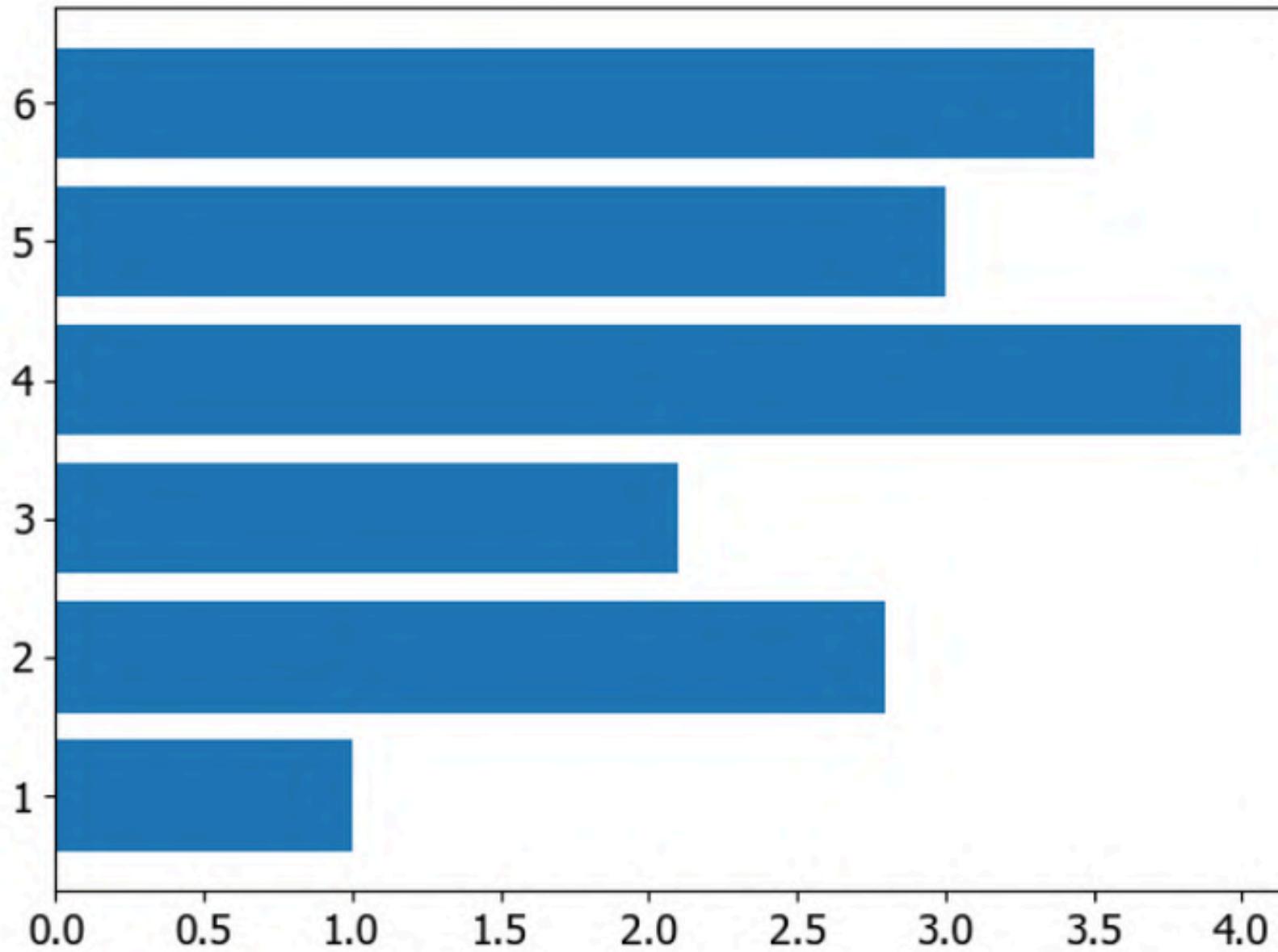


Jupyter Notebook

image processing

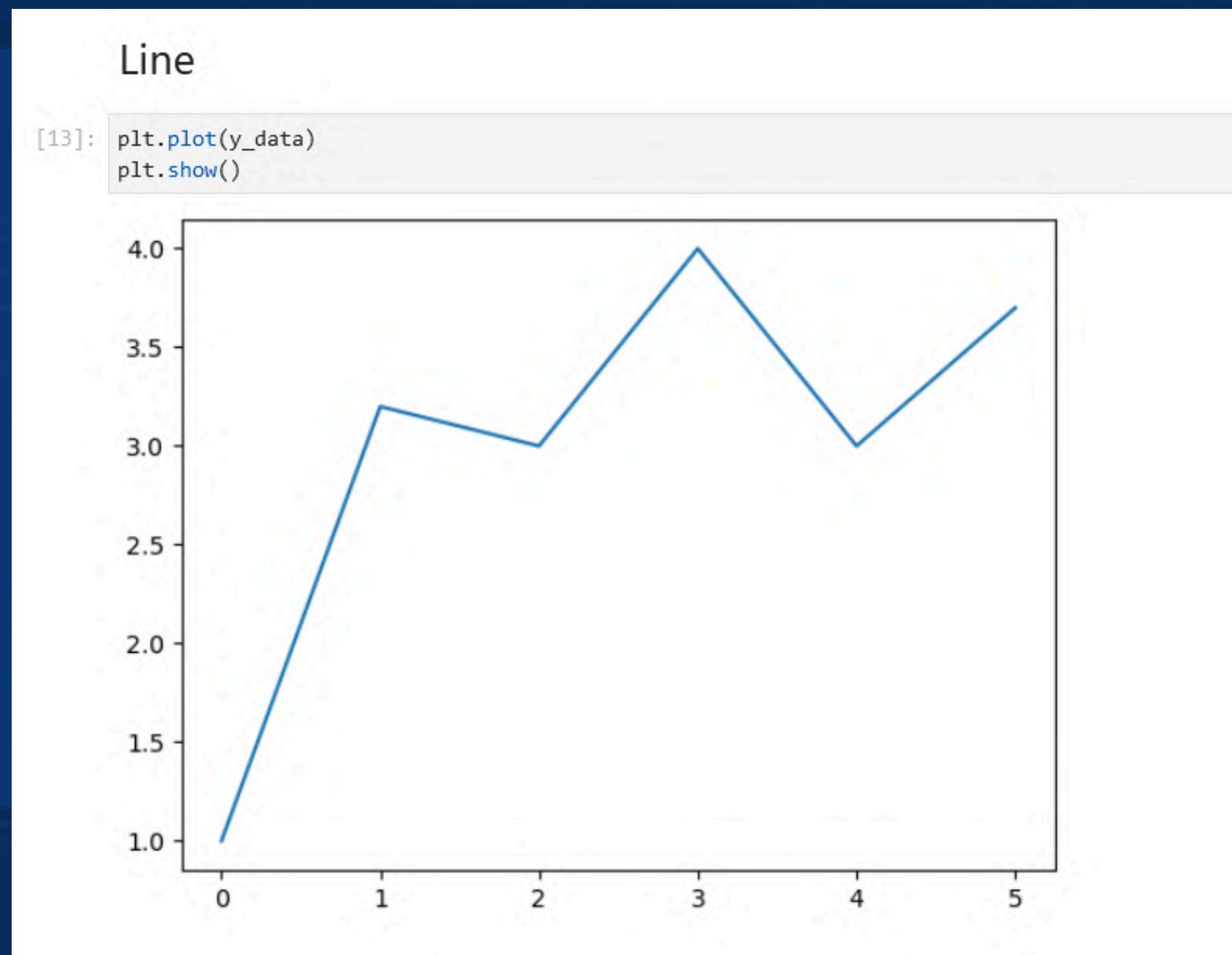
save

```
[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

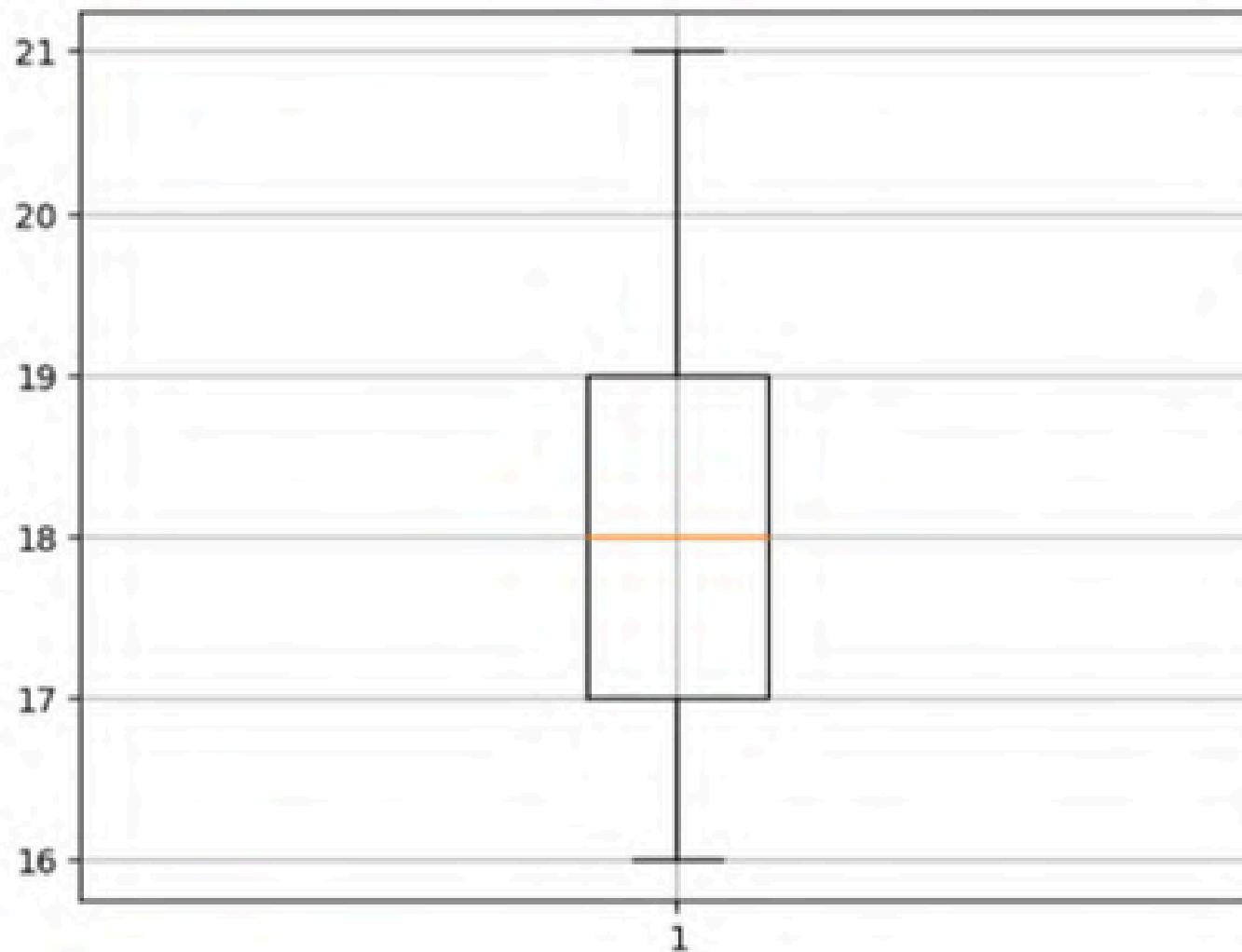


Jupyter Notebook

image processing

Box plot

```
[15]: age = [16,17,17,20,20,19,19,18,18,17,16,19,19,19,18,17,21]
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]: # เรียนรู้การเขียนป้าย (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(xs, ydata, color='c', width=0.5, label='ลูกค้าเดิม')
plt.bar([x + 0.2 for x in xs], # ข้อมูลที่เพิ่ม
        ydata2, width=0.5, color='orange', alpha=0.8, label='ลูกค้าใหม่')

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

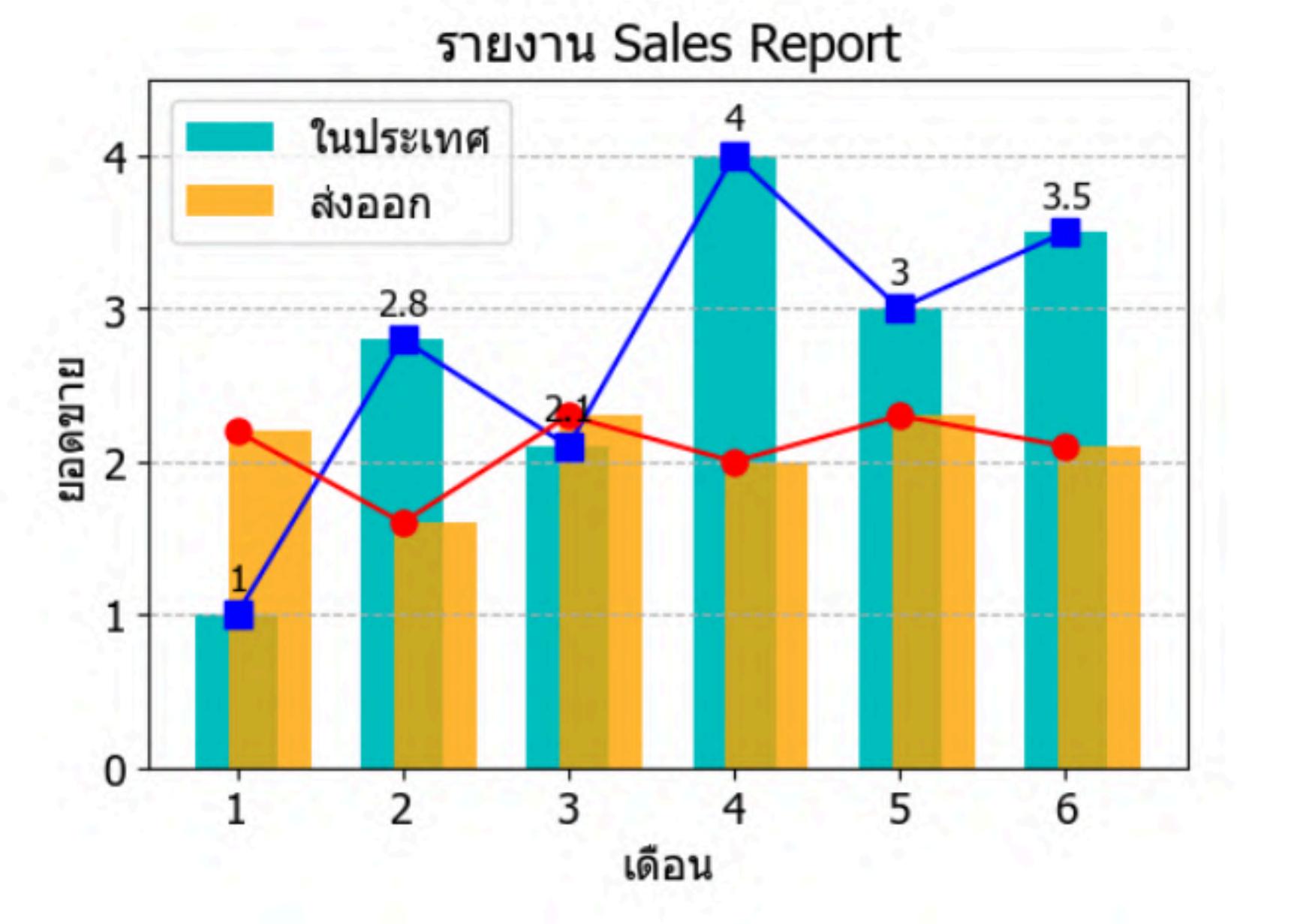
plt.xlabel("พื้นที่")
plt.ylabel("ยอดขาย")
plt.title("รายงาน Sales Report")

for tx,ty in list(zip(xs, ydata)):
    plt.annotate(ty, # ตัวอักษร label
                 (tx, ty), # point ที่ต้องการ label
                 textcoords="offset points", # position ของตัวอักษร
                 xytext=(0, 8), # ระยะทาง points (x,y)
                 ha='center',
                 fontsize=12) # font

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

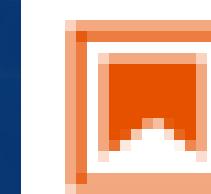
Jupyter Notebook

image processing

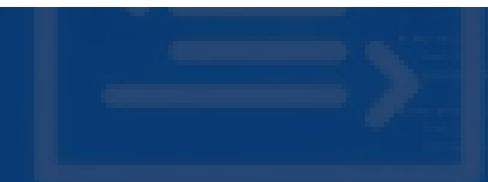


Jupyter Notebook

image processing



ch6 seaborn.ipynb



Jupyter Notebook

image processing

Seaborn

```
[1]: conda list seaborn
```

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

```
[40]: xdata = [1,2,3,4,5,6]          # មែលូកខ្លាងបន្ទាន់ x (តែងចាប់ពី 1 ដល់លម្អិតគាំទ្រ)
ydata = [1,2.8,2.1,4,3,3.5]      # មែលូកបន្ទាន់ y (x នាមីជាបន្ទាន់មែលូកខ្លាងបន្ទាន់)
```

```
sns.barplot(xdata, ydata)
```

```
-----
TypeError                                 Traceback (most recent call last)
Cell In[40], line 4
      1 xdata = [1,2,3,4,5,6]          # មែលូកខ្លាងបន្ទាន់ x (តែងចាប់ពី 1 ដល់លម្អិតគាំទ្រ)
      2 ydata = [1,2.8,2.1,4,3,3.5]      # មែលូកបន្ទាន់ y (x នាមីជាបន្ទាន់មែលូកខ្លាងបន្ទាន់)
----> 4 sns.barplot(xdata, ydata)

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

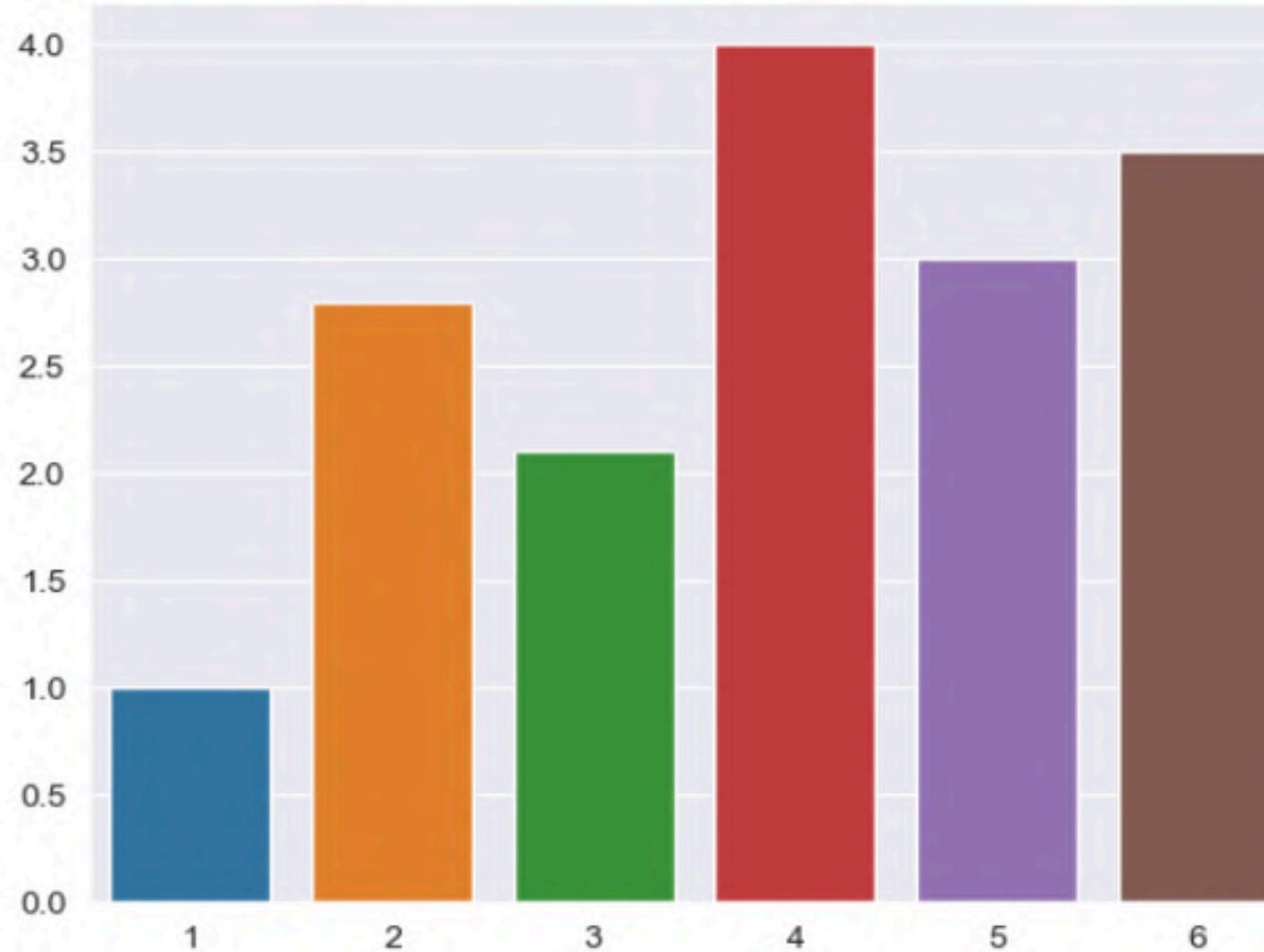
```
[22]: sns.barplot(x=xdata, y=ydata)
plt.show()
```

The figure shows a bar plot with six bars. The x-axis is labeled with integers from 1 to 6. The y-axis ranges from 2.5 to 4.0. The bars have the following approximate heights: bar 1 (~2.8), bar 2 (~4.0), bar 3 (~3.0), and bar 4 (~3.5). Bars 5 and 6 are very low, near the baseline.

Jupyter Notebook

image processing

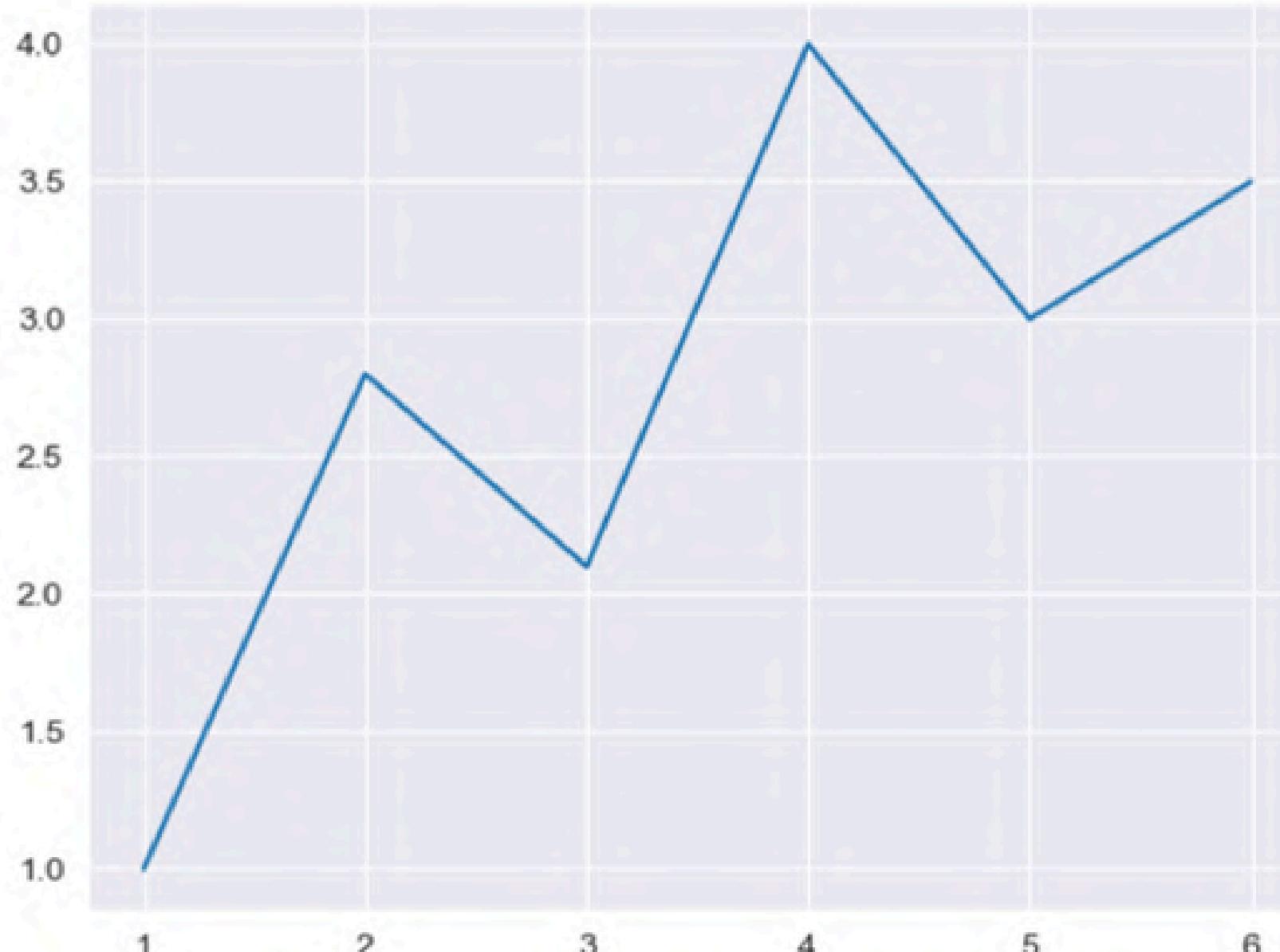
```
[22]: sns.barplot(x=xdata, y=ydata)
plt.show()
```



Jupyter Notebook

image processing

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

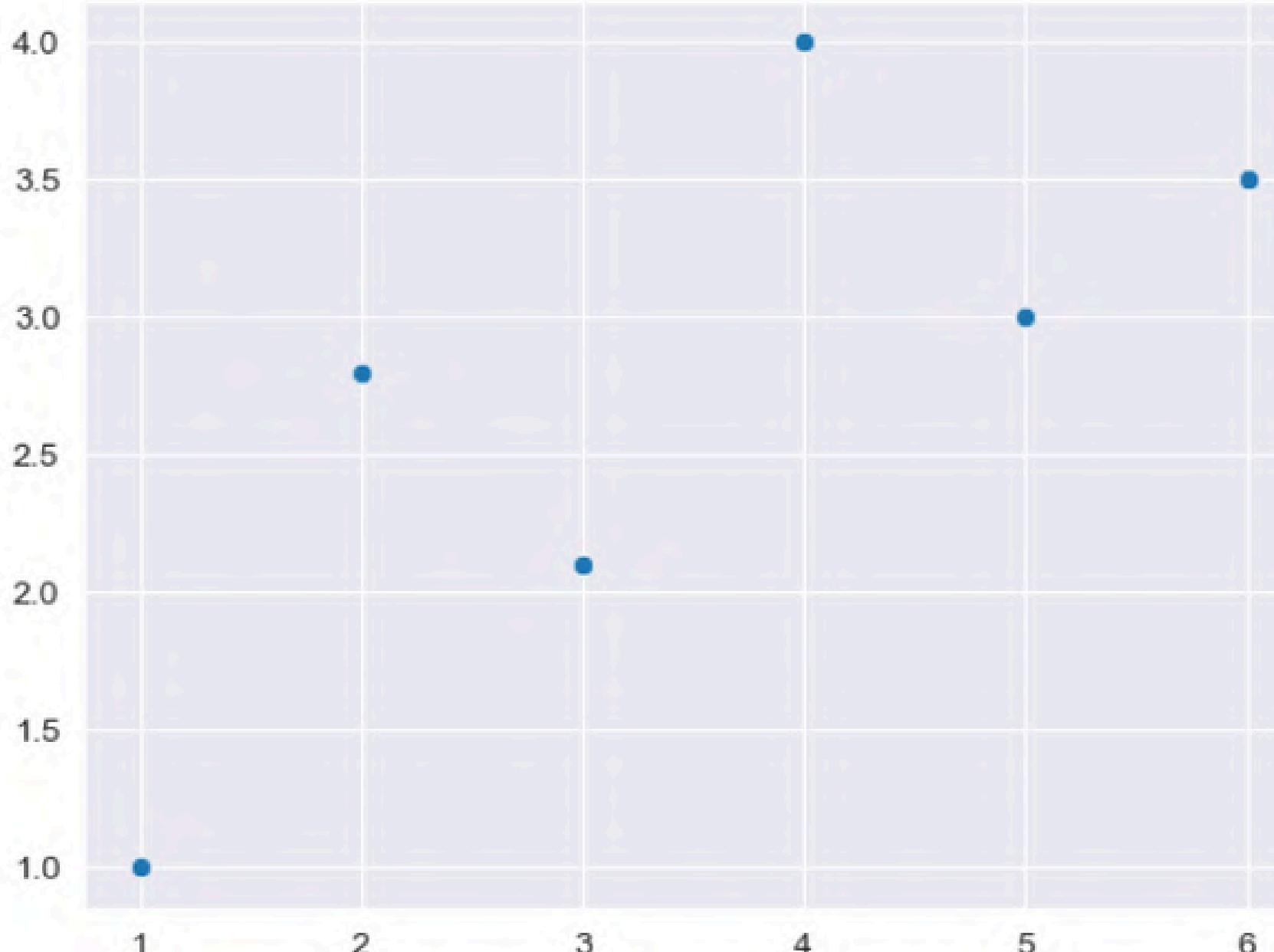


Jupyter Notebook

image processing

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

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



Jupyter Notebook

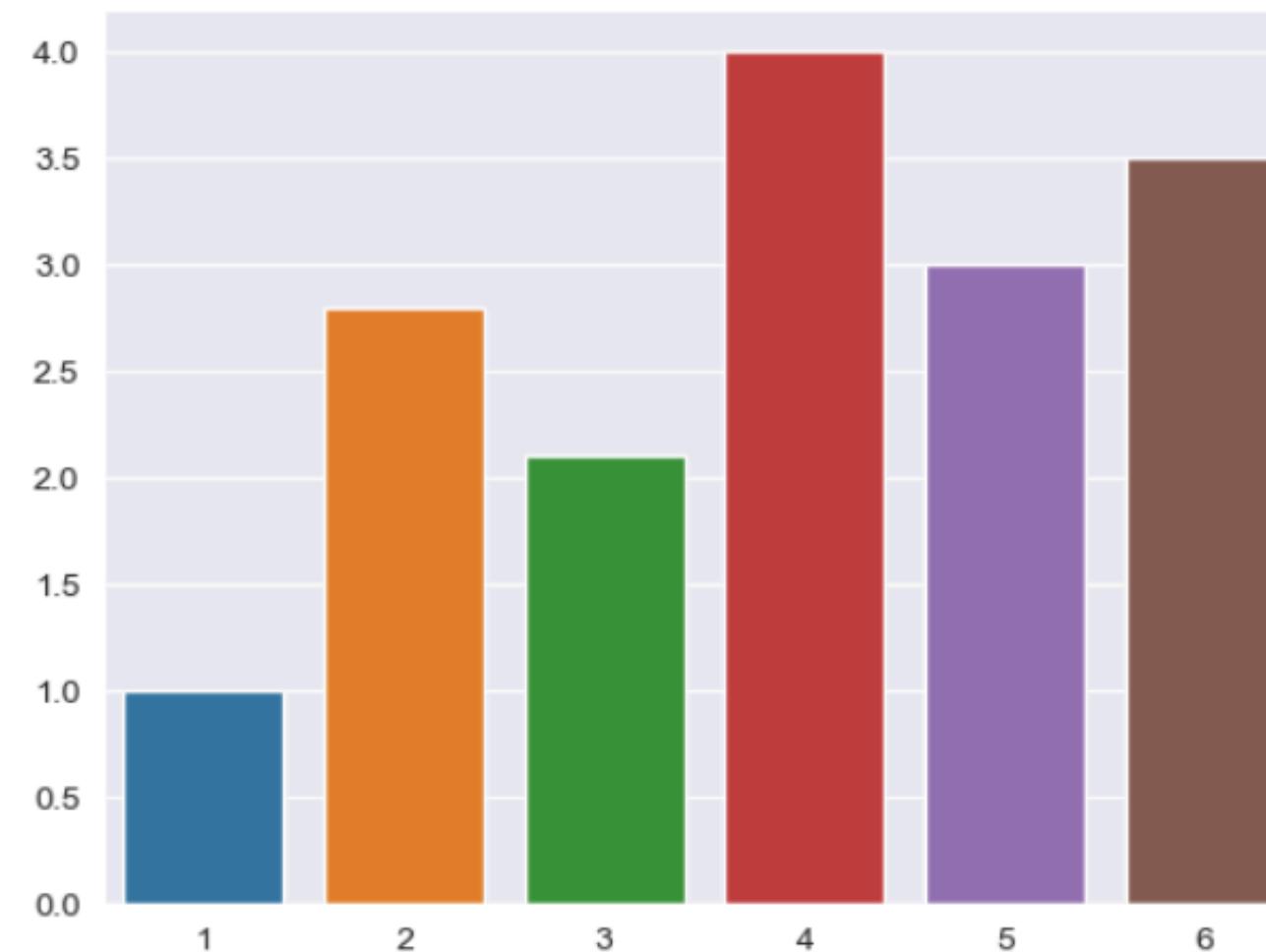
image processing

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: >
```

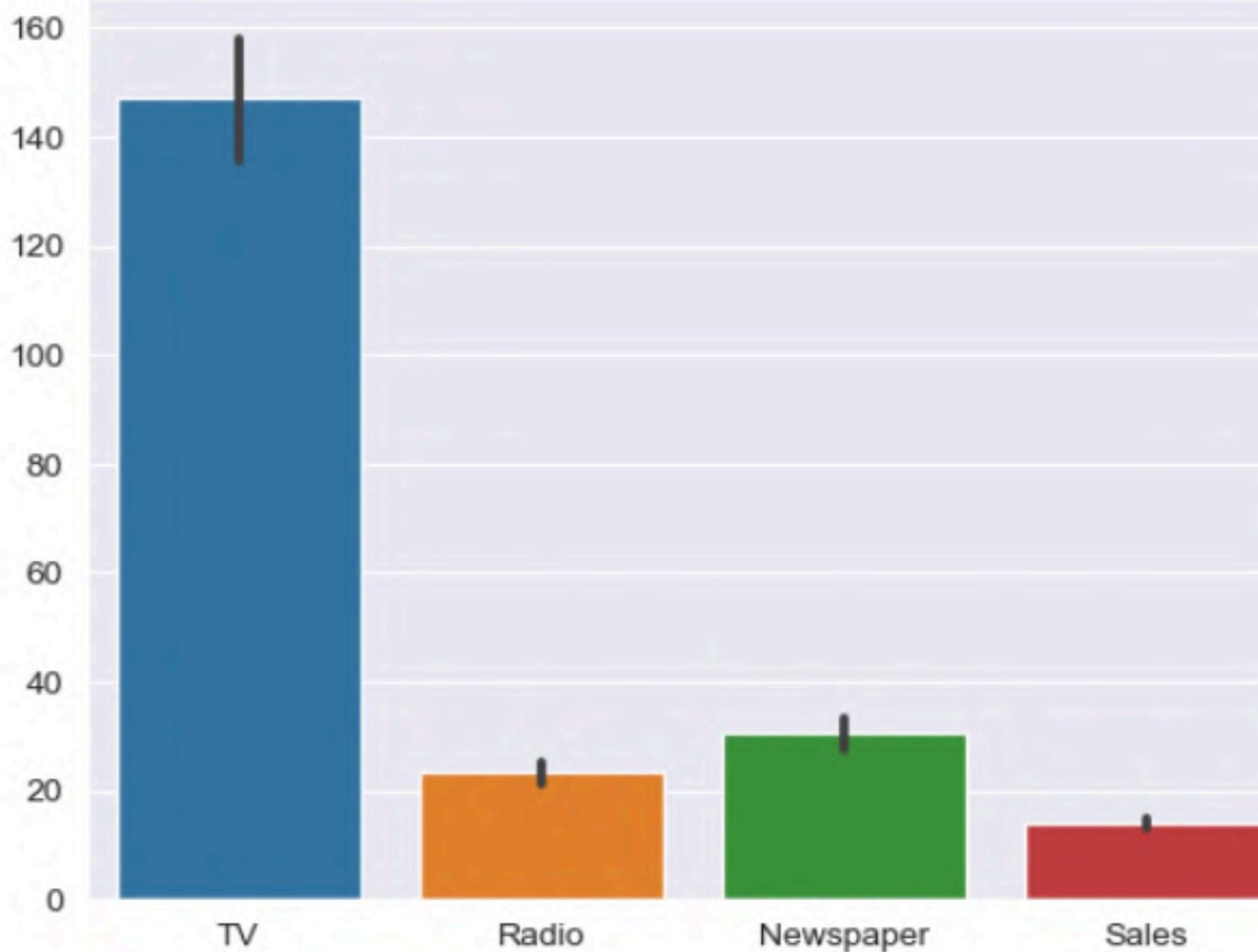


Jupyter Notebook

image processing

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

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



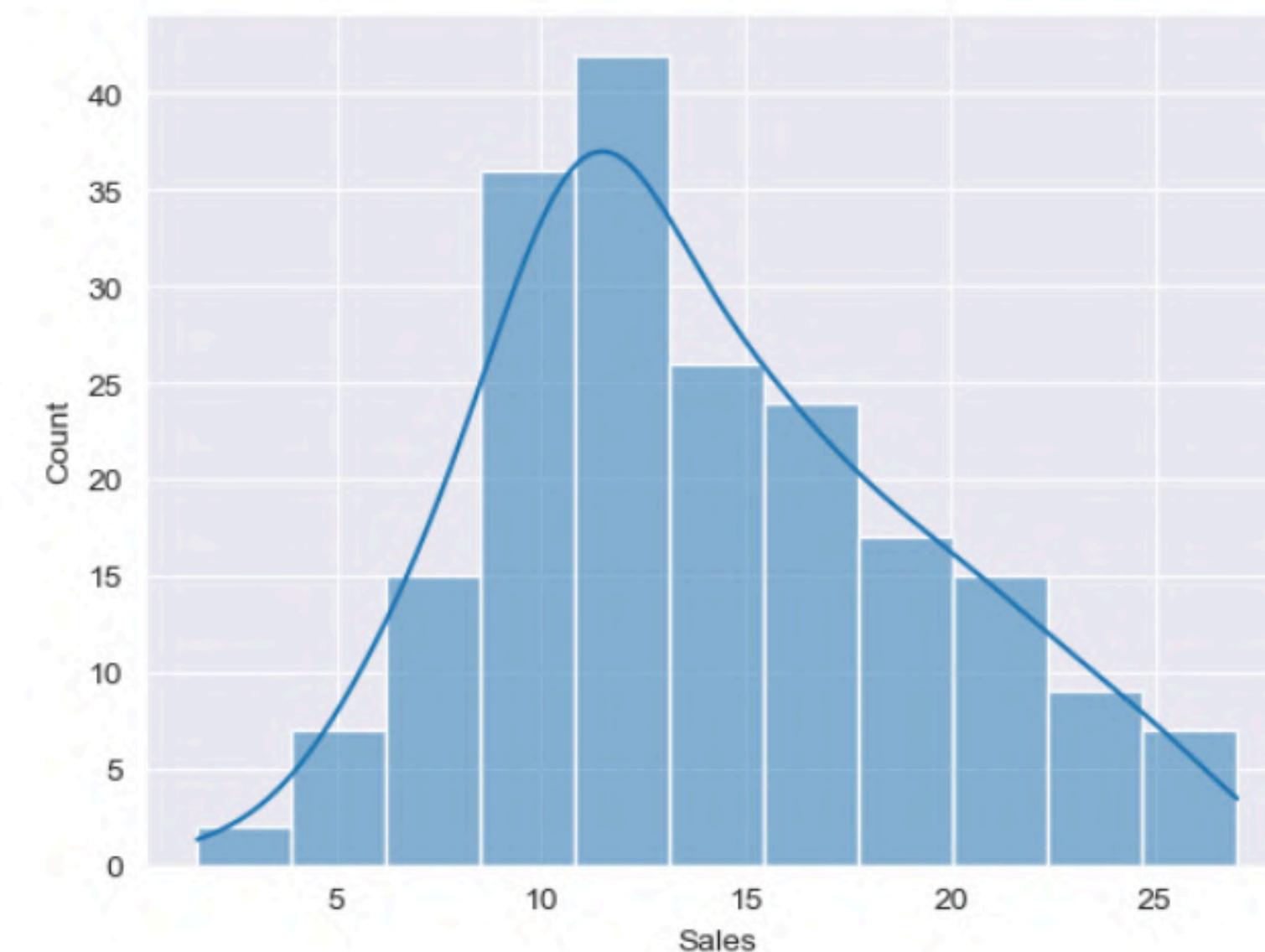
Jupyter Notebook

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)

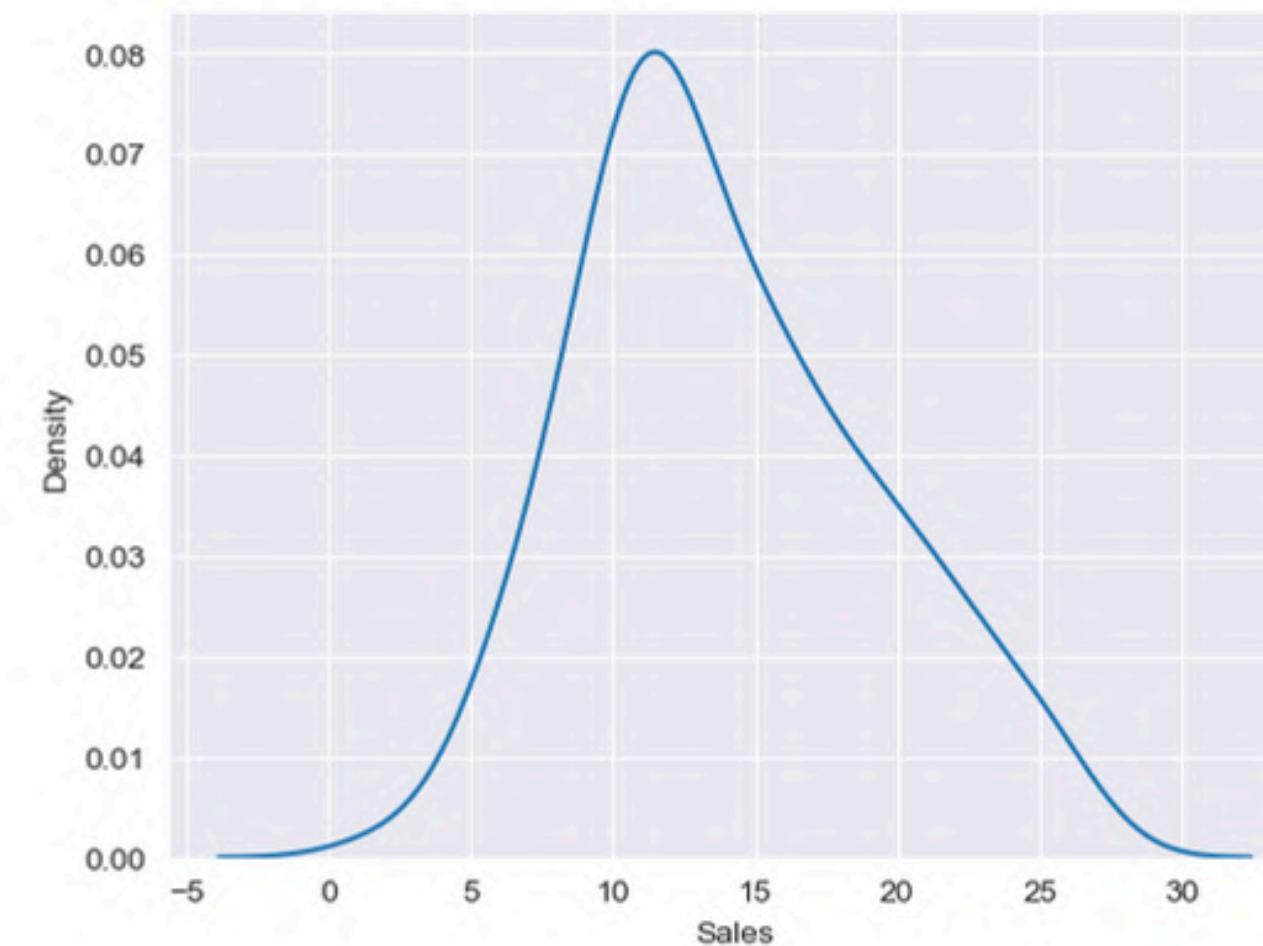
```
[33]: sns.kdeplot(df.Sales, shade=False)
```

```
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

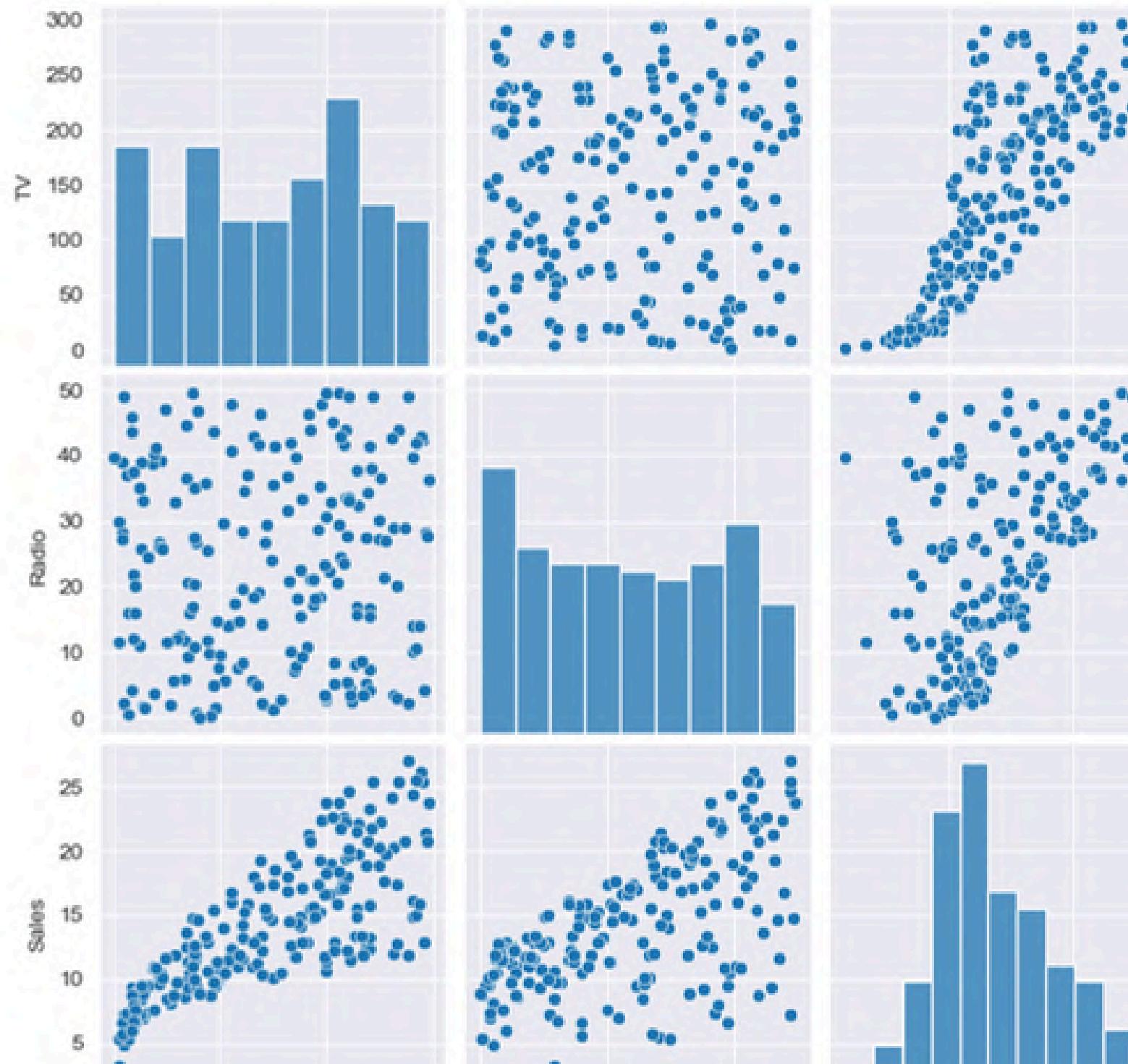


Jupyter Notebook

image processing

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

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



Jupyter Notebook

image processing

Irish dataset

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

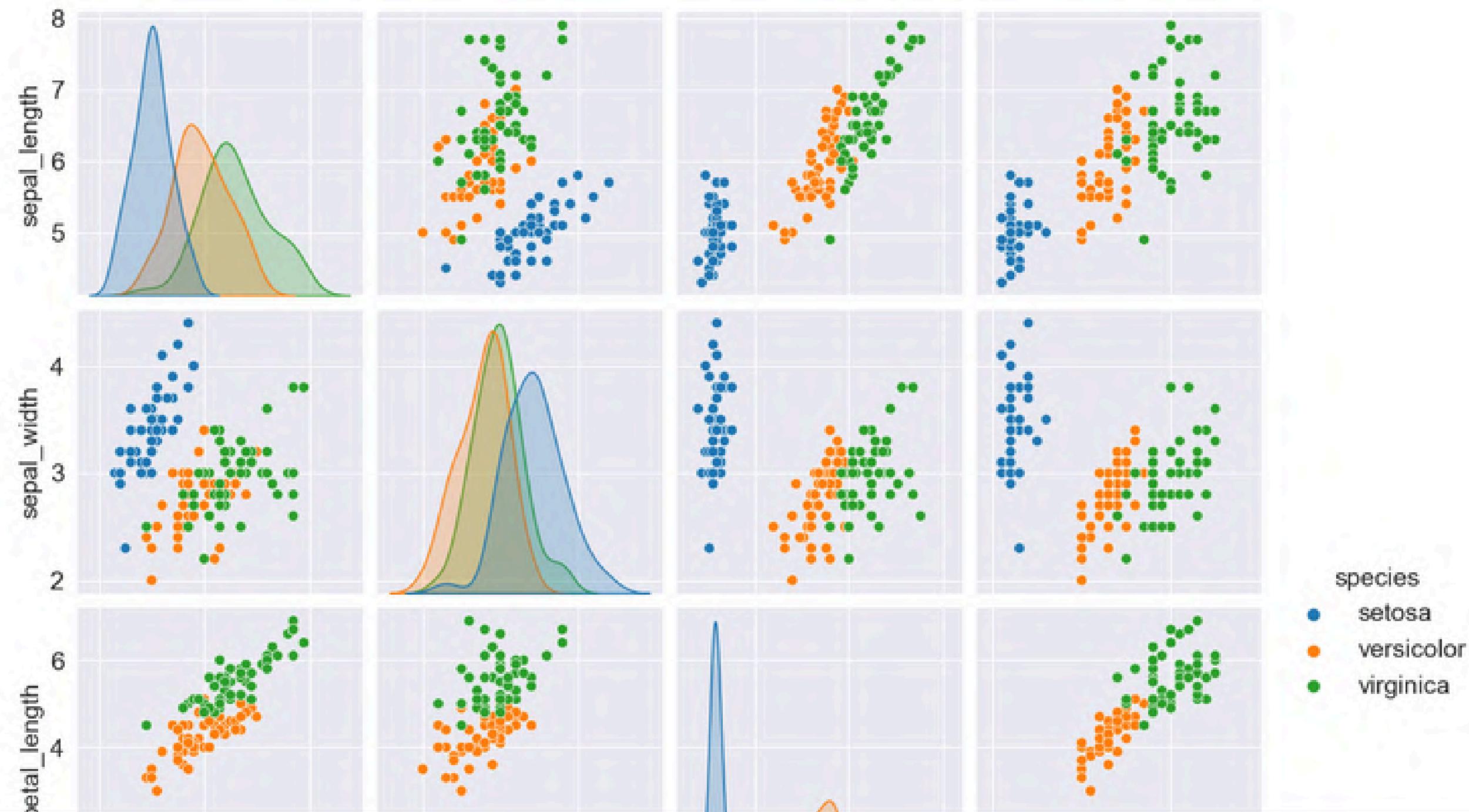
	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

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

Jupyter Notebook

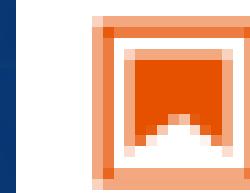
image processing

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



Jupyter Notebook

image processing



ch7_model.ipynb



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()
```

A scatter plot showing the relationship between Investment (X-axis) and Profit (Y-axis). The X-axis ranges from 1 to 6, and the Y-axis ranges from 1.0 to 3.5. The data points show a clear positive linear correlation.

Investment	Profit
1.0	1.0
1.8	1.3
3.0	2.2
4.1	2.5
5.2	2.8
6.0	3.6

Jupyter Notebook

image processing

Machine Learning (w/ 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
```

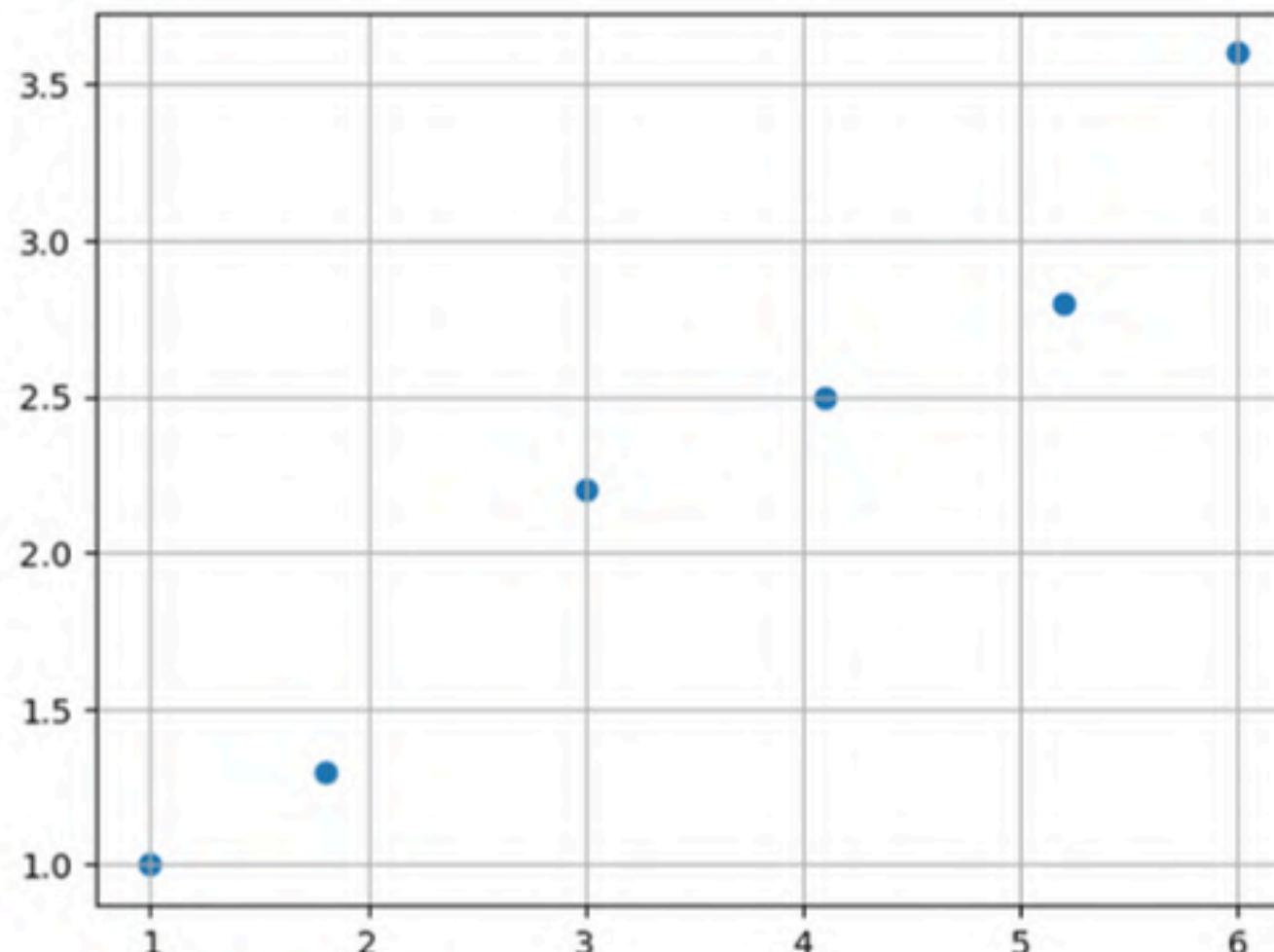
```
[8]: array([1. , 1.3, 2.2, 2.5, 2.8, 3.6])
```

Jupyter Notebook

image processing

Data Visualization

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



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

```
[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]] # ไม่จำเป็นต้องเรียงตาม  
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],  
[3],  
[7.]])
```

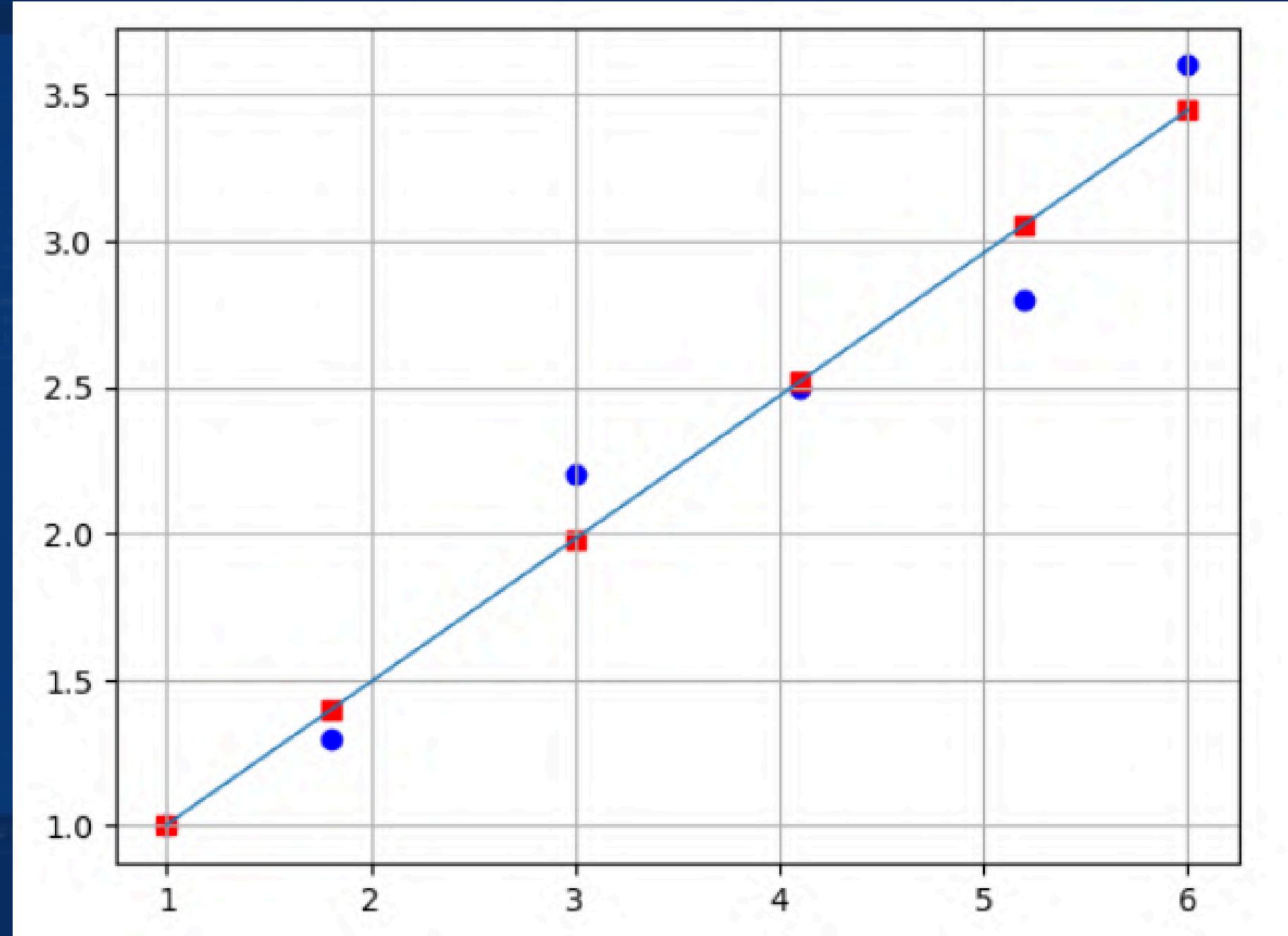
Jupyter Notebook

image processing

```
[28]: # គិតរបាយអត្ថបទ input នូវ នៅតម្លៃ  
for i, y in enumerate(y_predict):  
    print('x={:.4f} 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')      # ម៉ោងទីតាំង (real data points)  
plt.plot(x_data, predict, linewidth=1)        # ការពារ (prediction)  
plt.scatter(x_data, predict, color='r', marker='s')  
# plt.savefig('modell.png', dpi=100)  
plt.show()
```

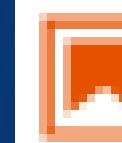
Jupyter Notebook

image processing



Jupyter Notebook

image processing



ch8_use_model.ipynb



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
      (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]: # วนรอบและ 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

image classification to AI :

EXIT

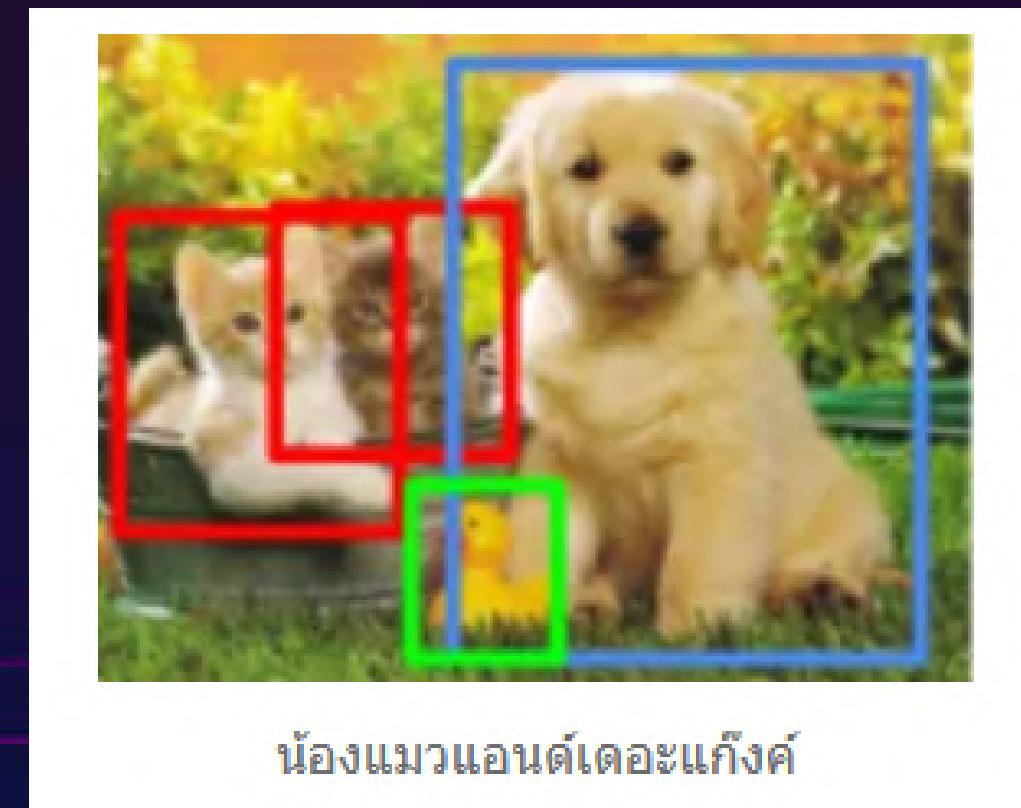
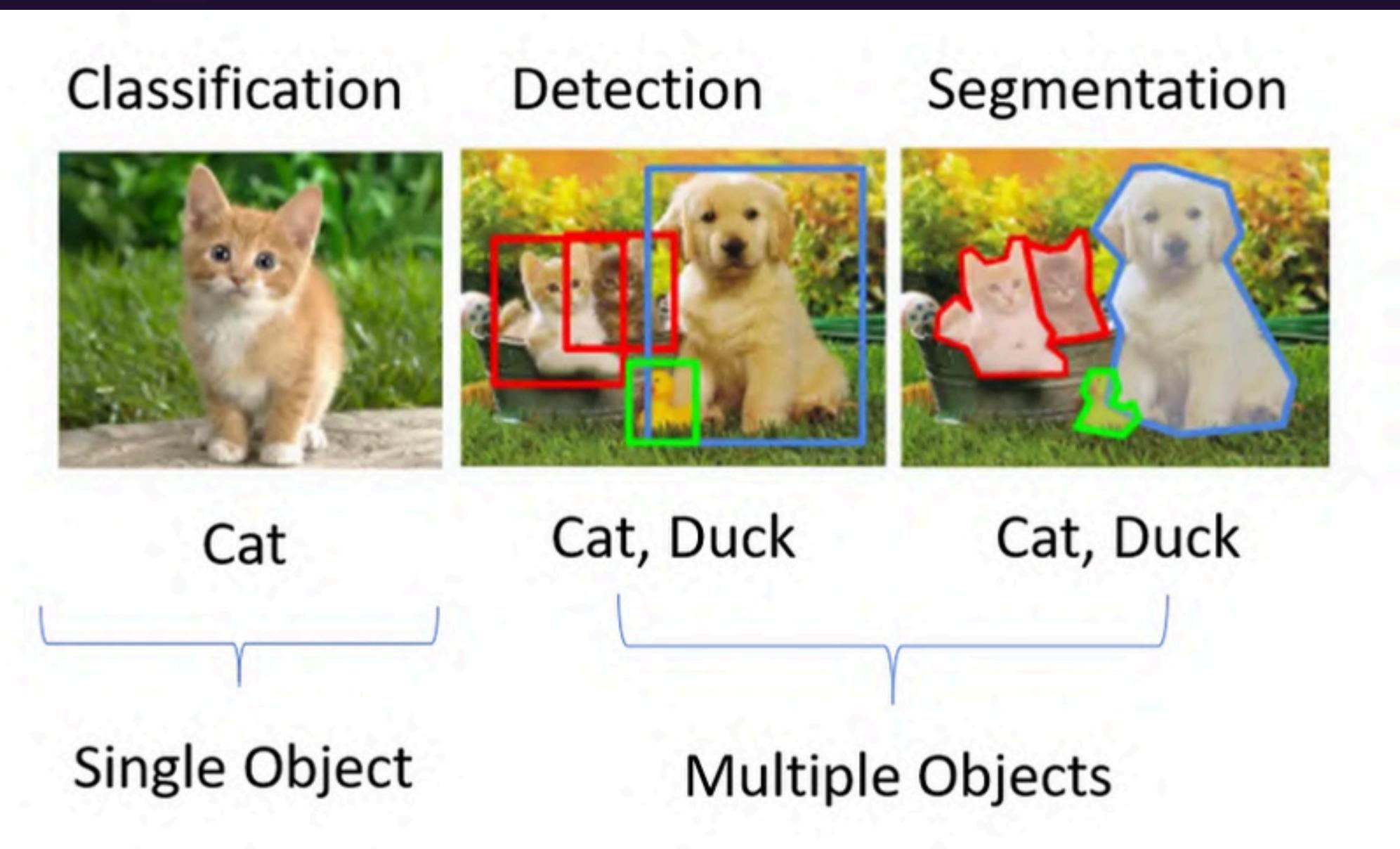


IMAGE CLASSIFICATION



EXIT

IMAGE CLASSIFICATION

Medical Sector

X-ray

AI

Focus

Segmentation

Image

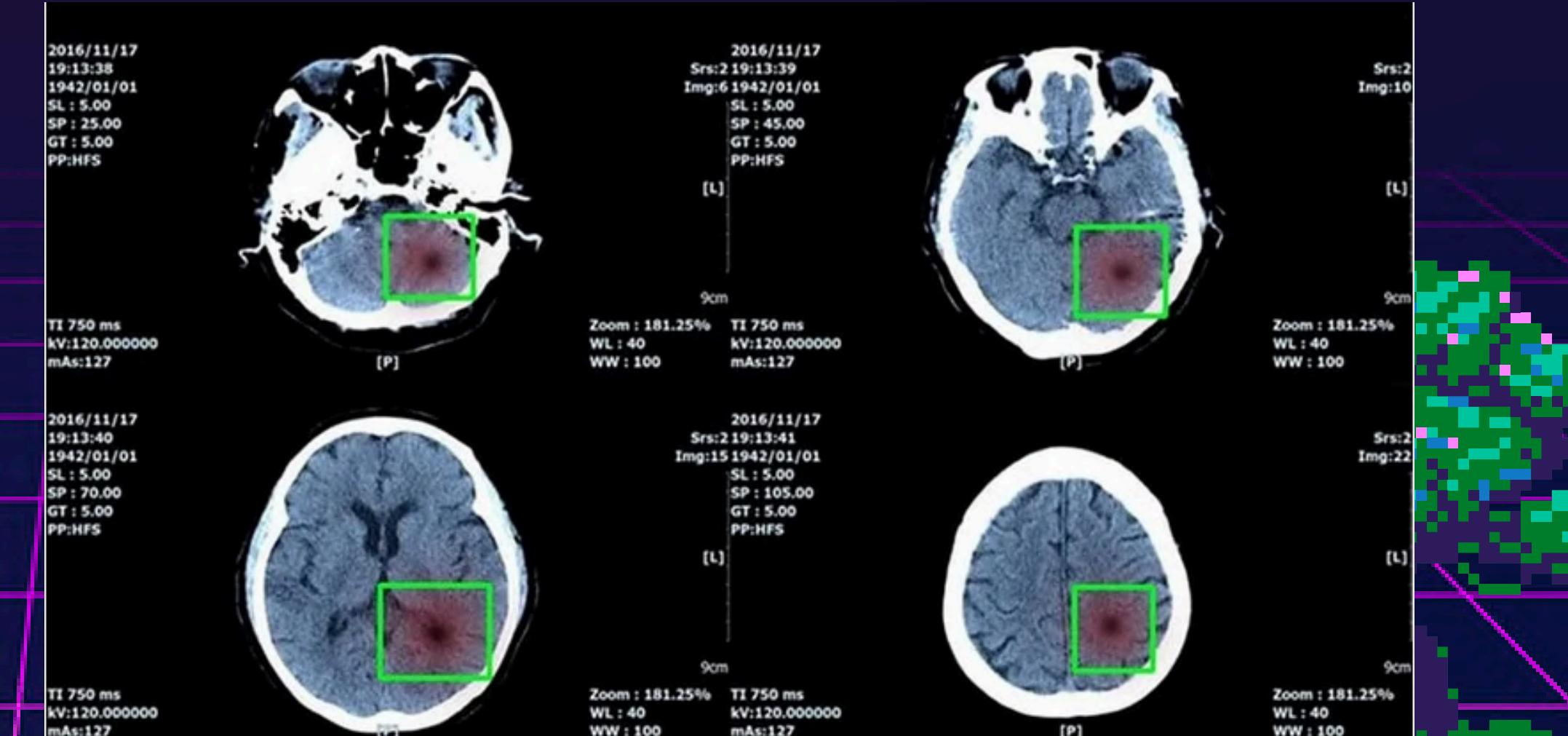
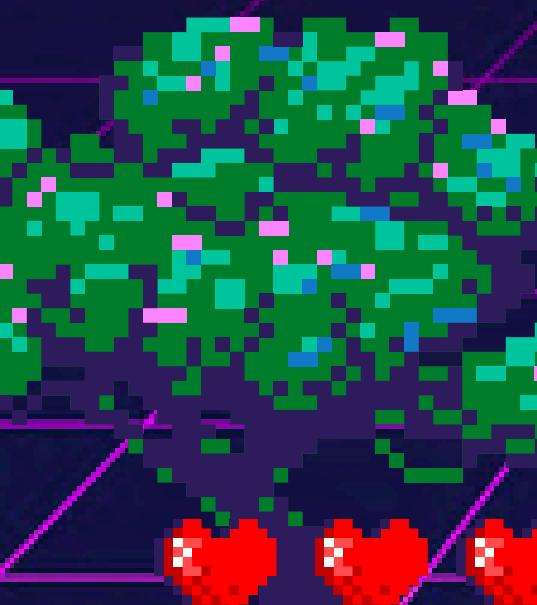


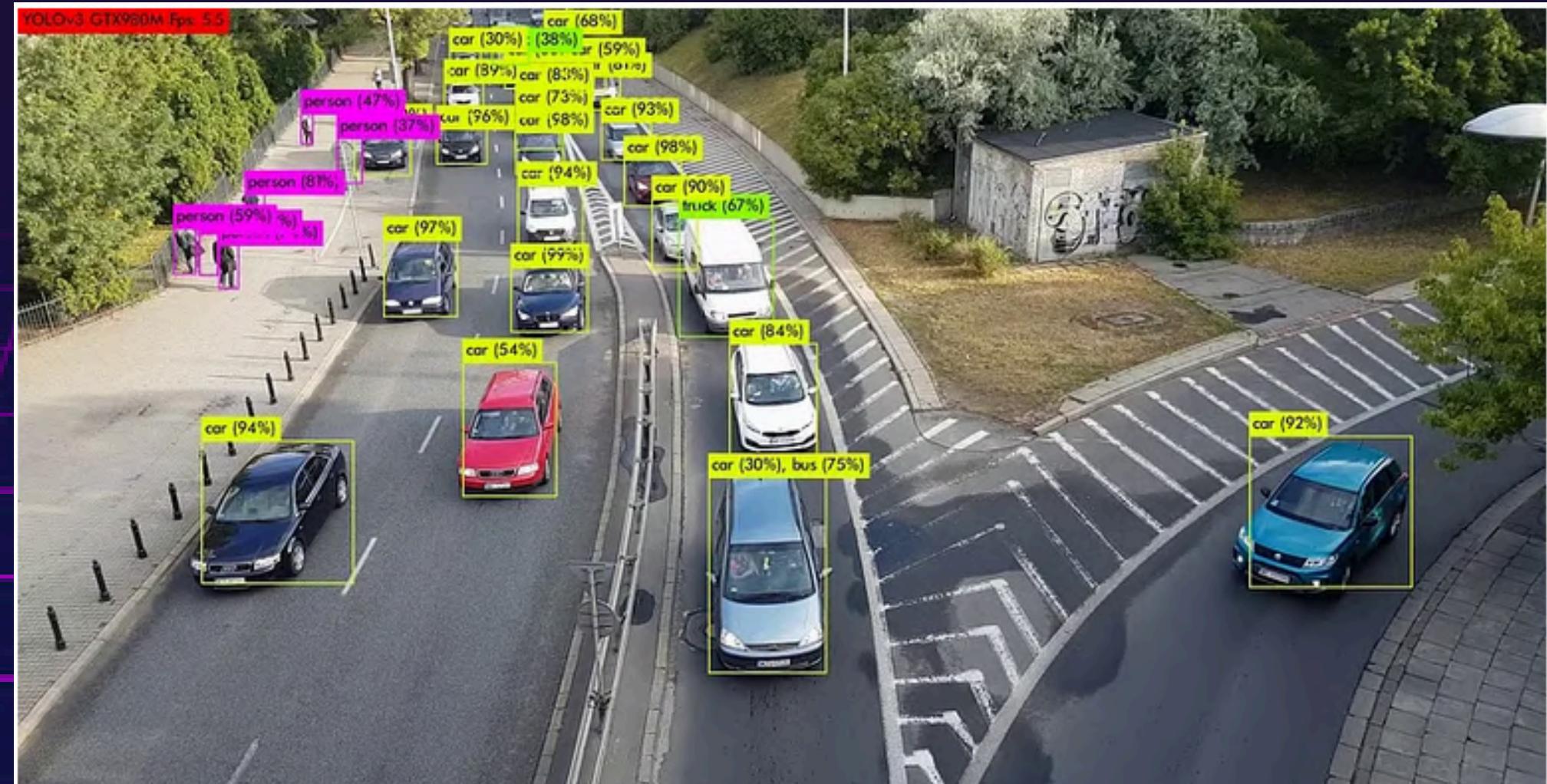
IMAGE CLASSIFICATION

Traffic Sector

AI

(

)



EXIT

YOLOv5 គីឡៅតិ៍ ?

'You only look once'

Version 10
ultralytics

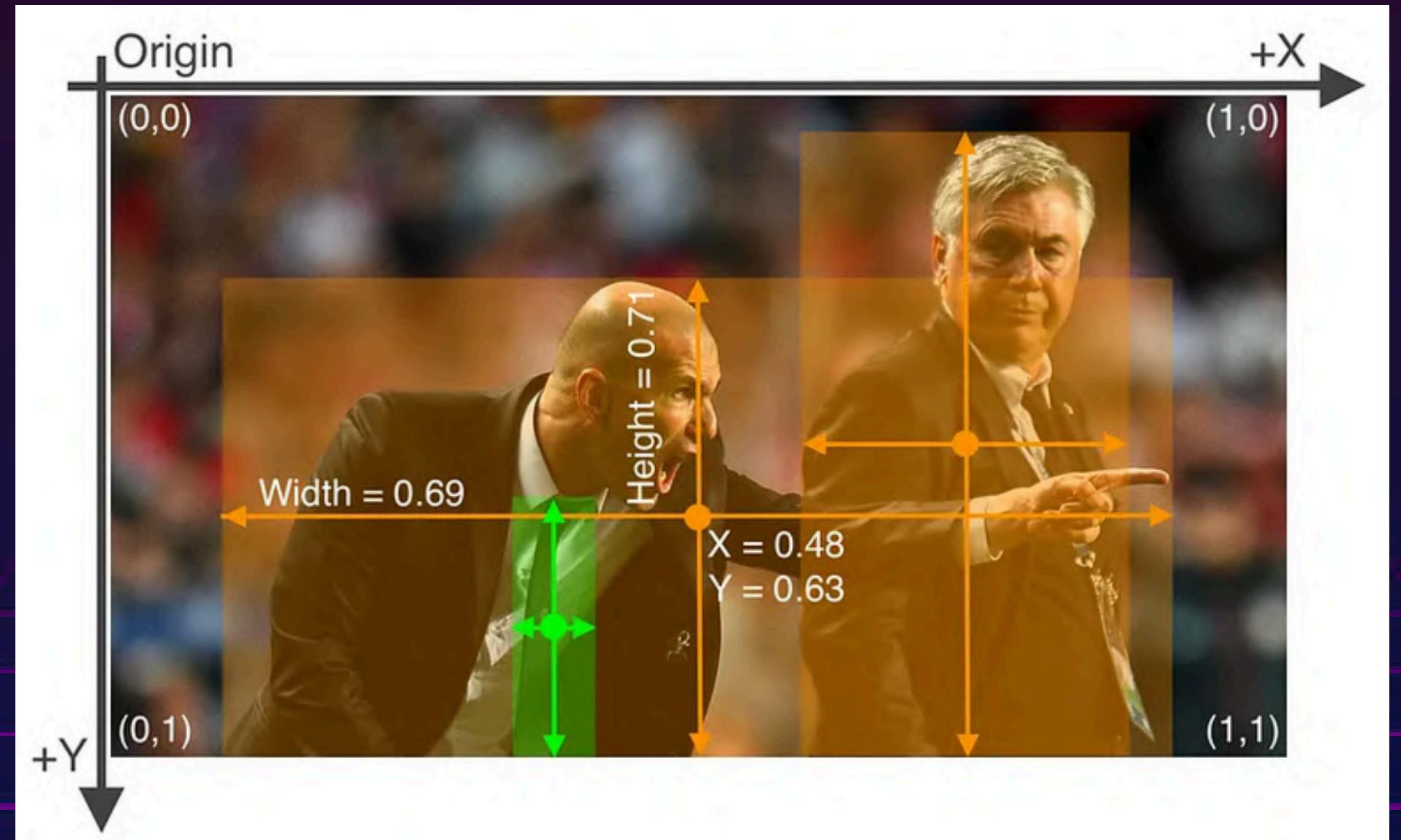
Image Detection

YOLO

EXIT

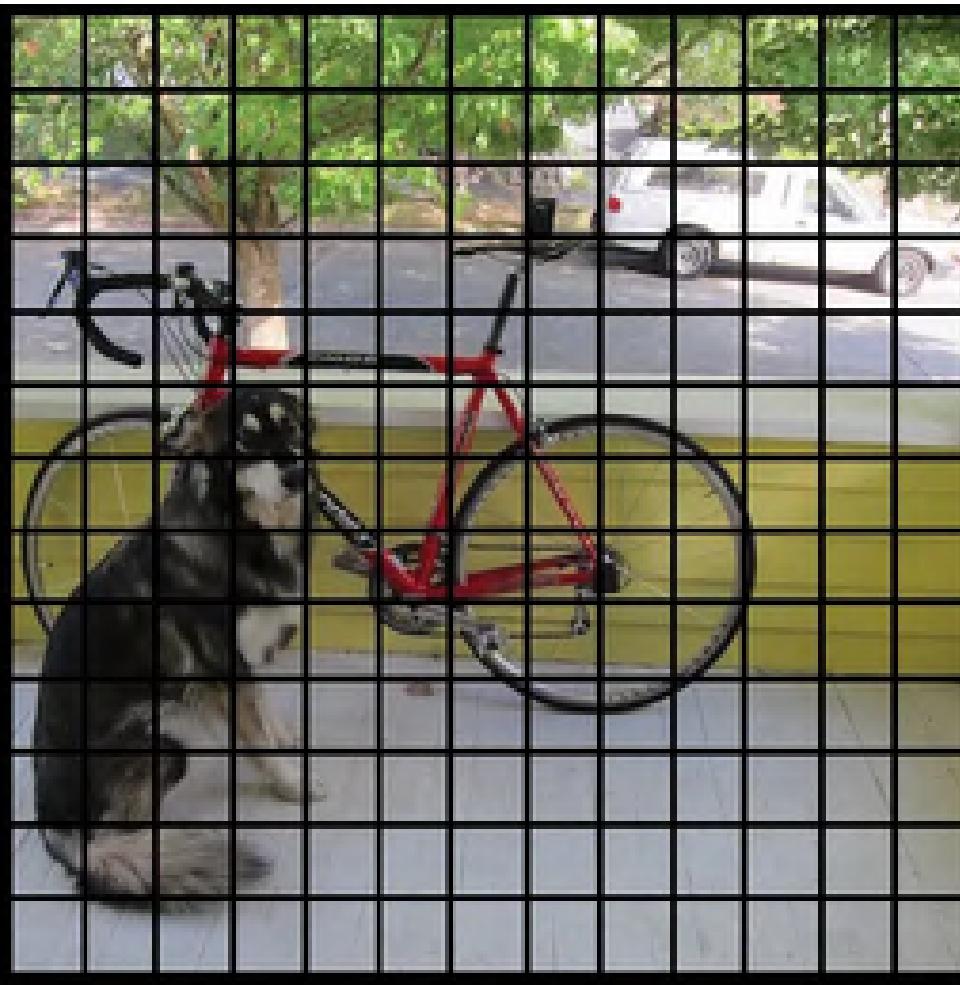


YOLOV5 គីឡូច្ចារៈ ?

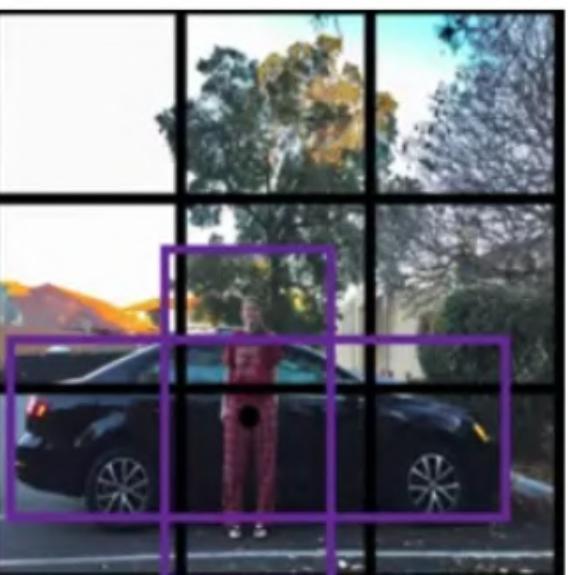


EXIT

YOLOV5 គីឡៅតូ ?

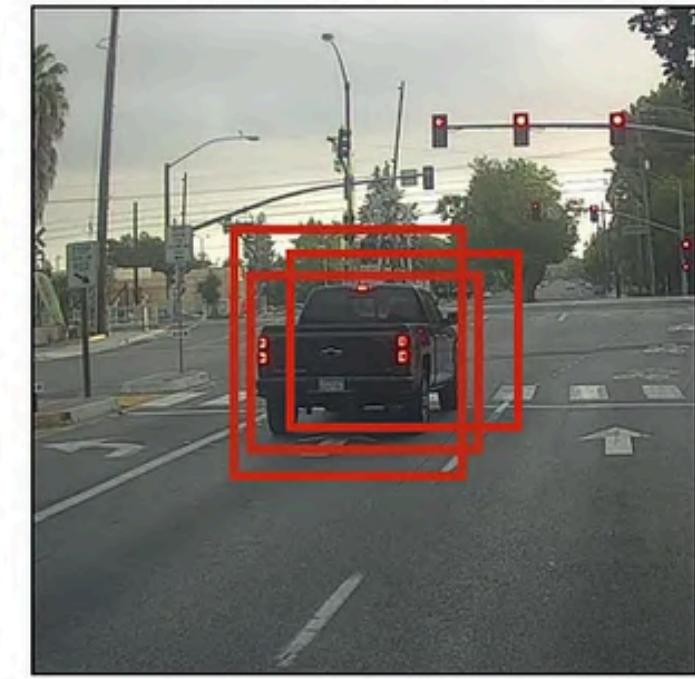


grid
elements pixel



EXIT

Before non-max suppression



Non-Max
Suppression



After non-max suppression



layer predict
intersection

การติดตั้ง YOLOV5

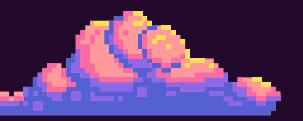
5

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

medium yolov5 :



EXIT



ນາສົດຕິ່ງ YOLOV5

1.

The screenshot shows the official Git website. At the top left is the Git logo with the tagline "git --local-branching-on-the-cheap". On the right is a search bar. Below the header, there's a navigation menu with links to "About", "Documentation", "Downloads" (which is highlighted in red), "GUI Clients", "Logos", and "Community". The main content area is titled "Downloads" and features three download links: "macOS", "Windows", and "Linux/Unix". To the right, a computer monitor displays the "Latest source Release 2.42.0" with a "Release Notes (2023-08-21)" link and a "Download for Windows" button. A note at the bottom states: "Older releases are available and the Git source repository is on GitHub."

EXIT

ការតាំង 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

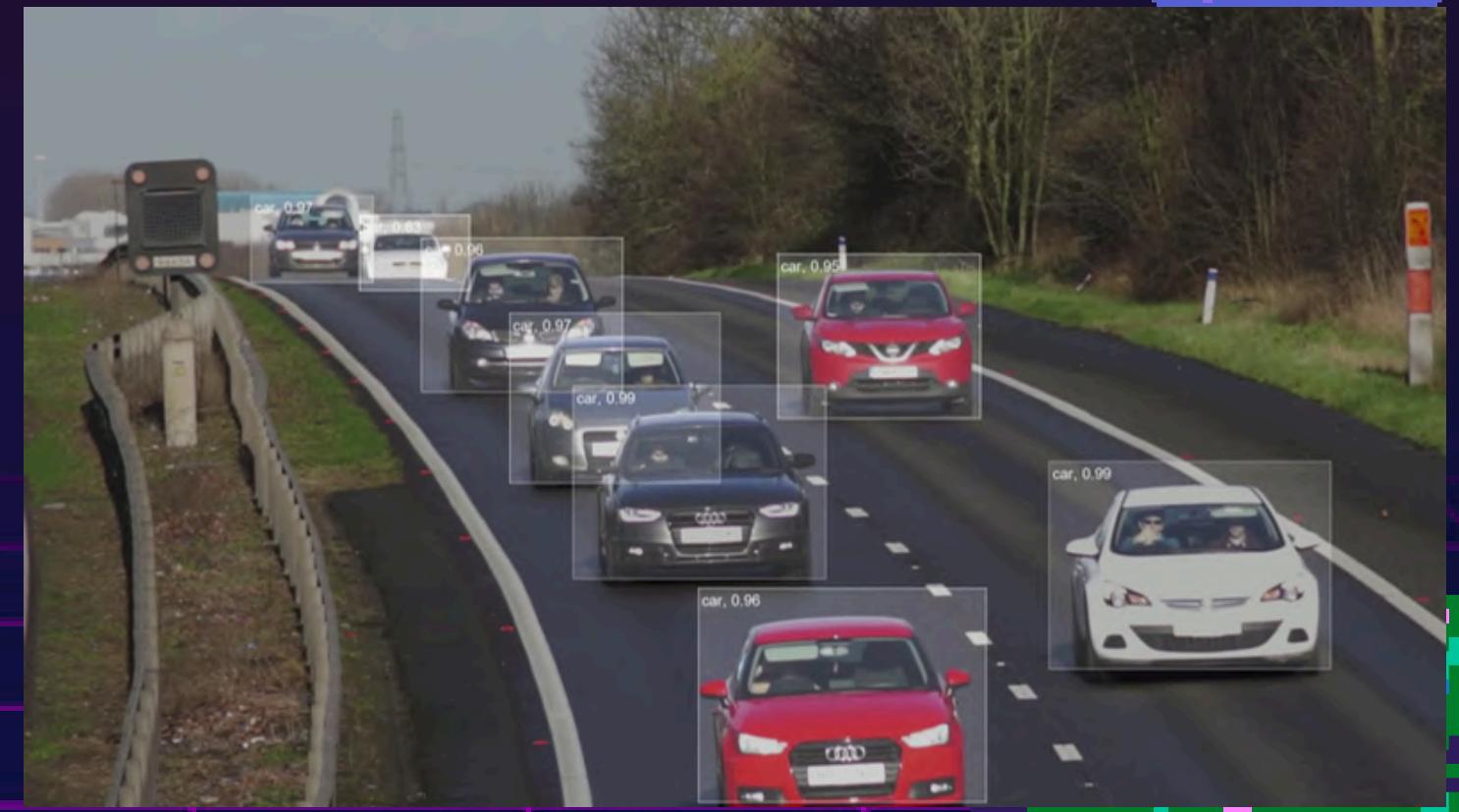
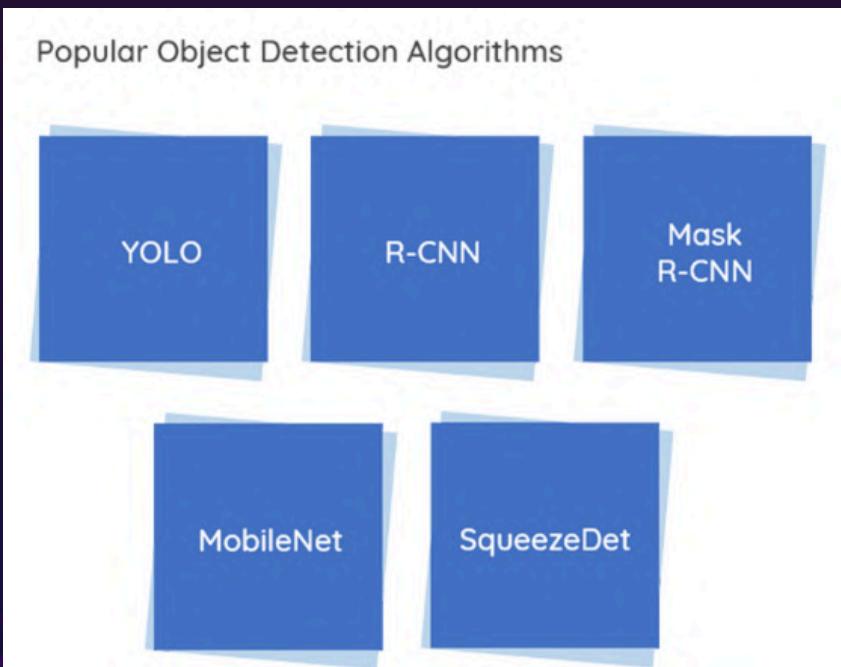
EXIT

nms? YOLOV3

Classification Detect

Person

0 - 1 (100%)



EXIT

nst 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

ms? YOLOV3

0 - 1%

(CODE) Part 1

```
• detection_notEsp32cam.py > detect_person
1 import cv2
2 import numpy as np
3
4 # Load YOLOv3 model and class names
5 net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
6 with open("coco.names", "r") as f:
7     classes = [line.strip() for line in f.readlines()]
8
9 def detect_person(frame):
10     blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
11     net.setInput(blob)
12     outs = net.forward(net.getUnconnectedOutLayersNames())
```

Floder

EXIT

coco.names

detection_person.py

yolov3.cfg

yolov3.weights

nms? 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)
```

EXIT

mst's YOLOV3

0 - 1%

(CODE) Part 3

```
35     for i in range(len(boxes)):
36         if i in indexes:
37             x, y, w, h = boxes[i]
38             label = str(classes[class_ids[i]])
39             confidence = confidences[i]
40             cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
41             cv2.putText(frame, f'{label} {confidence:.2f}', (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
42
43     return frame
44
```

EXIT

nms? YOLOV3

0 - 1%

(CODE) Part 4

```
45 # Capture video from your webcam (change the argument to the video file if needed)
46 cap = cv2.VideoCapture(0)
47
48 while True:
49     ret, frame = cap.read()
50
51     if frame is not None:
52         frame = detect_person(frame)
53         cv2.imshow("Person Detection", frame)
54
55     if cv2.waitKey(1) & 0xFF == ord('q'):
56         break
57
58 cap.release()
59 cv2.destroyAllWindows()
```

EXIT

nms? YOLOV3

(Result)



EXIT



Thank You!

End of Session Image Processing