

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



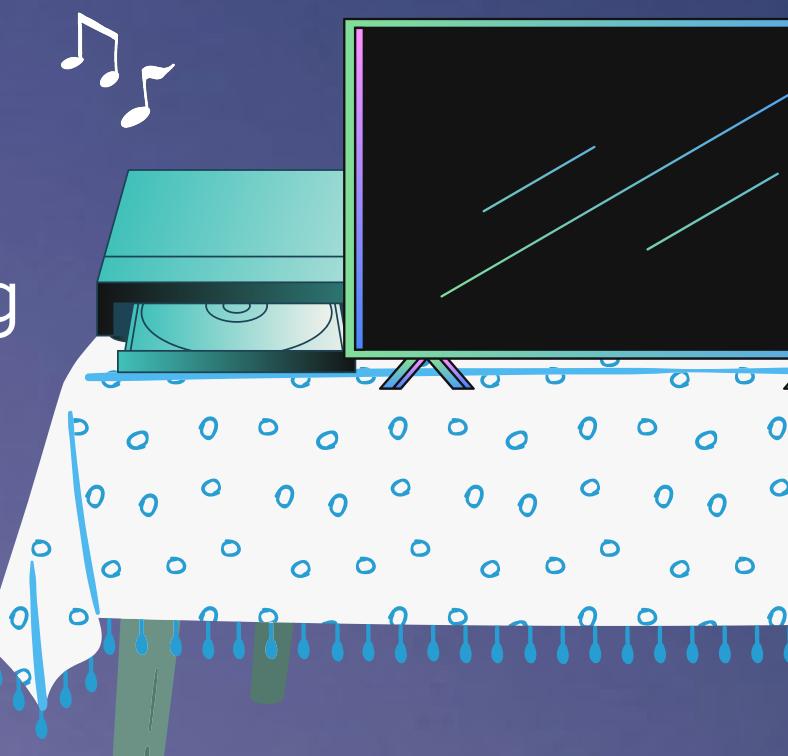
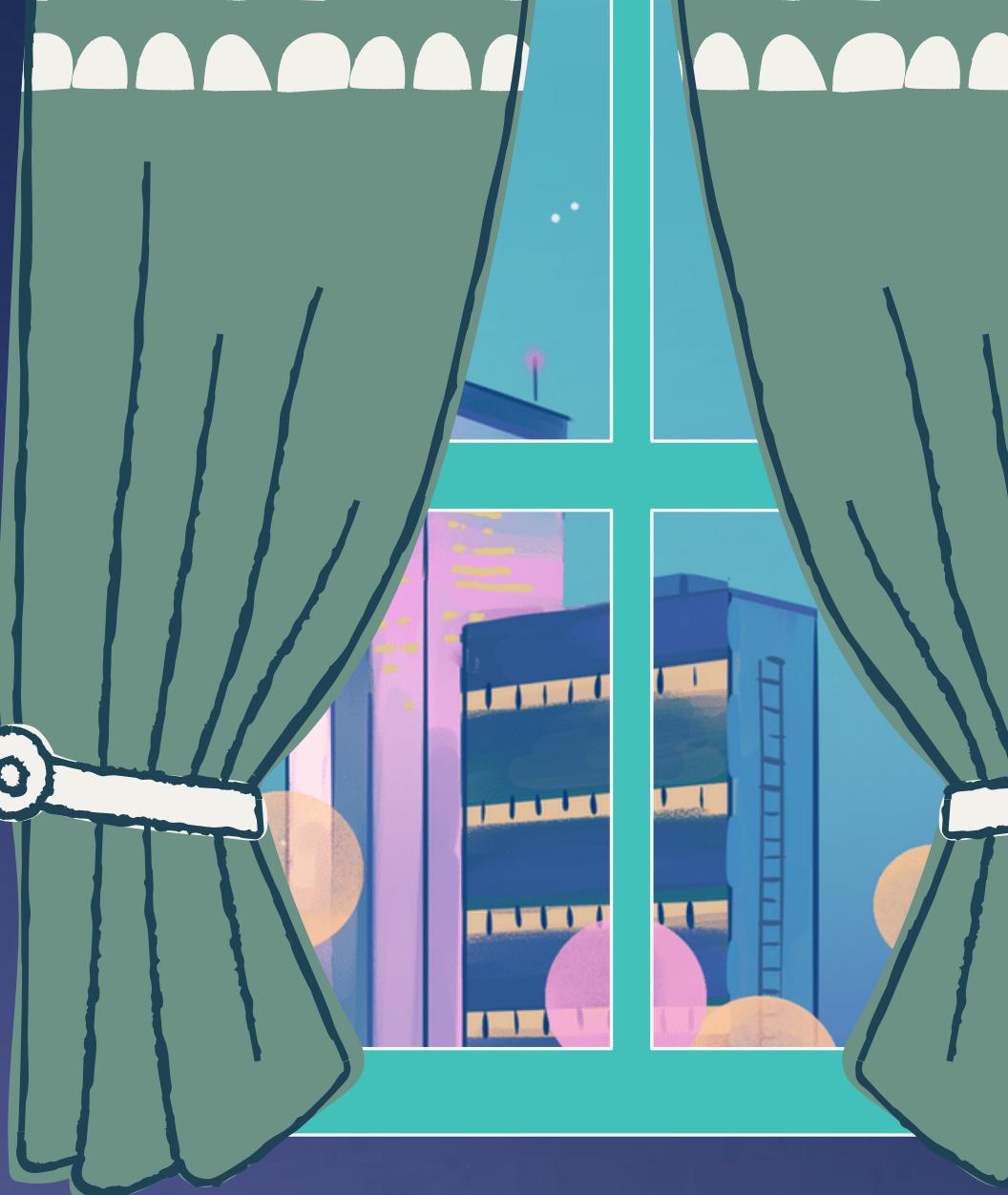
# Let's get to know each other first



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

```
pip3 install opencv-python
```

```
pip3 install numpy
```



We'll use Visual Studio Code

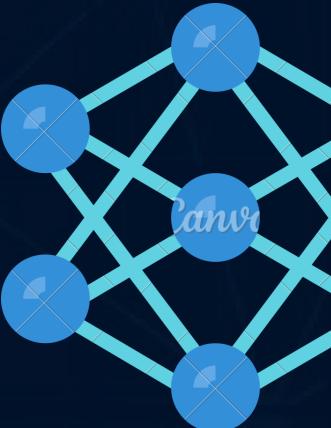


- Create new file your project (.py file)

# Q3

## Coding

## Model (face , eye , etc.)



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

```
1 import cv2
2 import numpy as np
3 cascade = cv2.CascadeClassifier('data/haarcascades/haarcascade_frontalface_default.xml')
4
5 cap = cv2.VideoCapture(0) → Initialize the webcam
6 while True:
7     ret, frame = cap.read() → Read a frame from the webcam
8     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) → Convert the frame to grayscale
9     objects = cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5, minSize=(30, 30))
10    for (x, y, w, h) in objects:
11        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
12
13    cv2.imshow('Object Detection', frame) → Perform object detection
14
15    if cv2.waitKey(1) & 0xFF == ord('q'): → When press key 'q' it will exist
16        break
17
18    cap.release()
19    cv2.destroyAllWindows()
```

# 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	May 29
ch1_hello.ipynb	May 29
data	May 29
Object_Detection_yolov5.ipynb	Aug 12

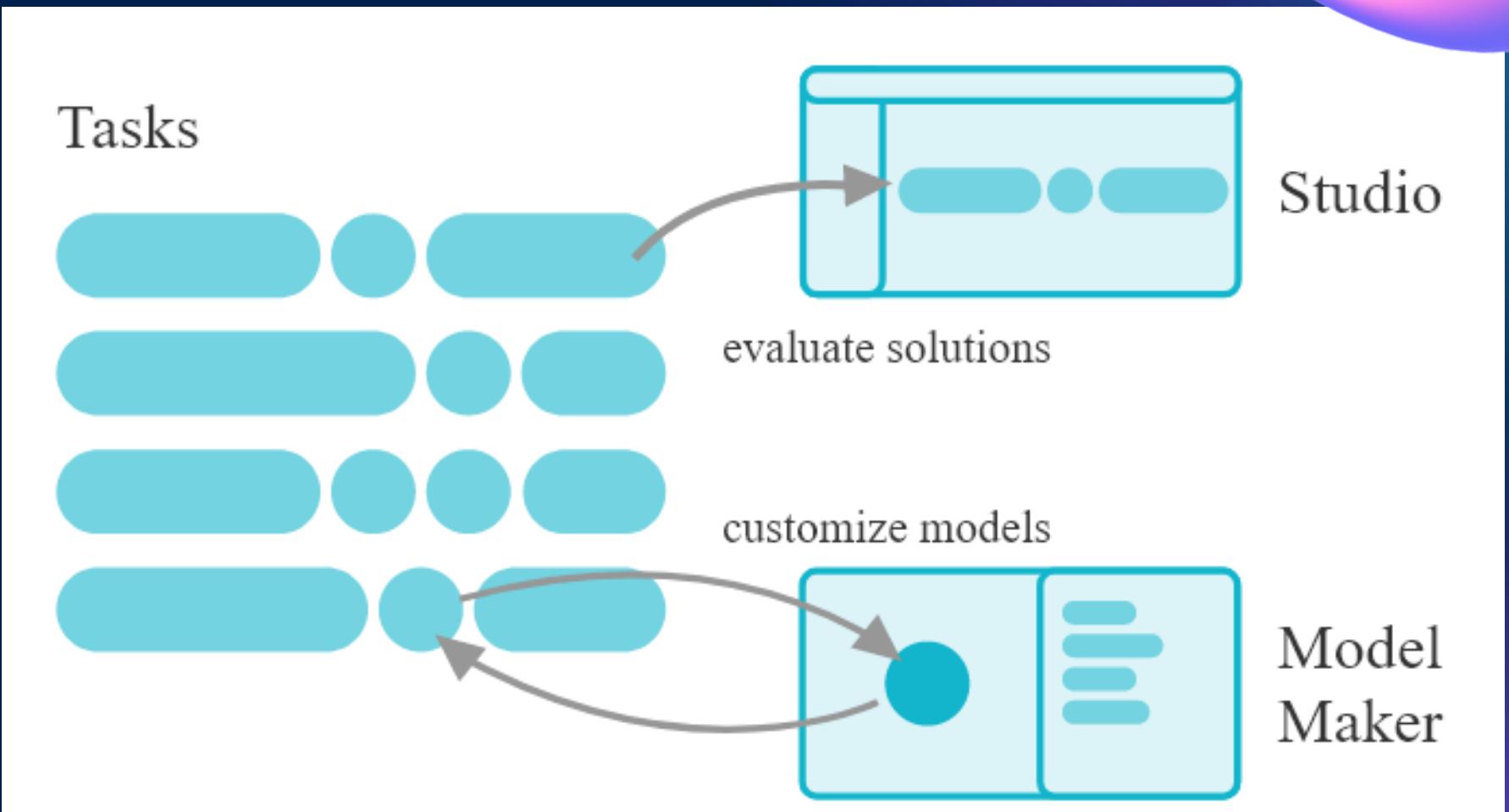
can try on others object example detection on this link : <https://shorturl.at/ejszl>

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

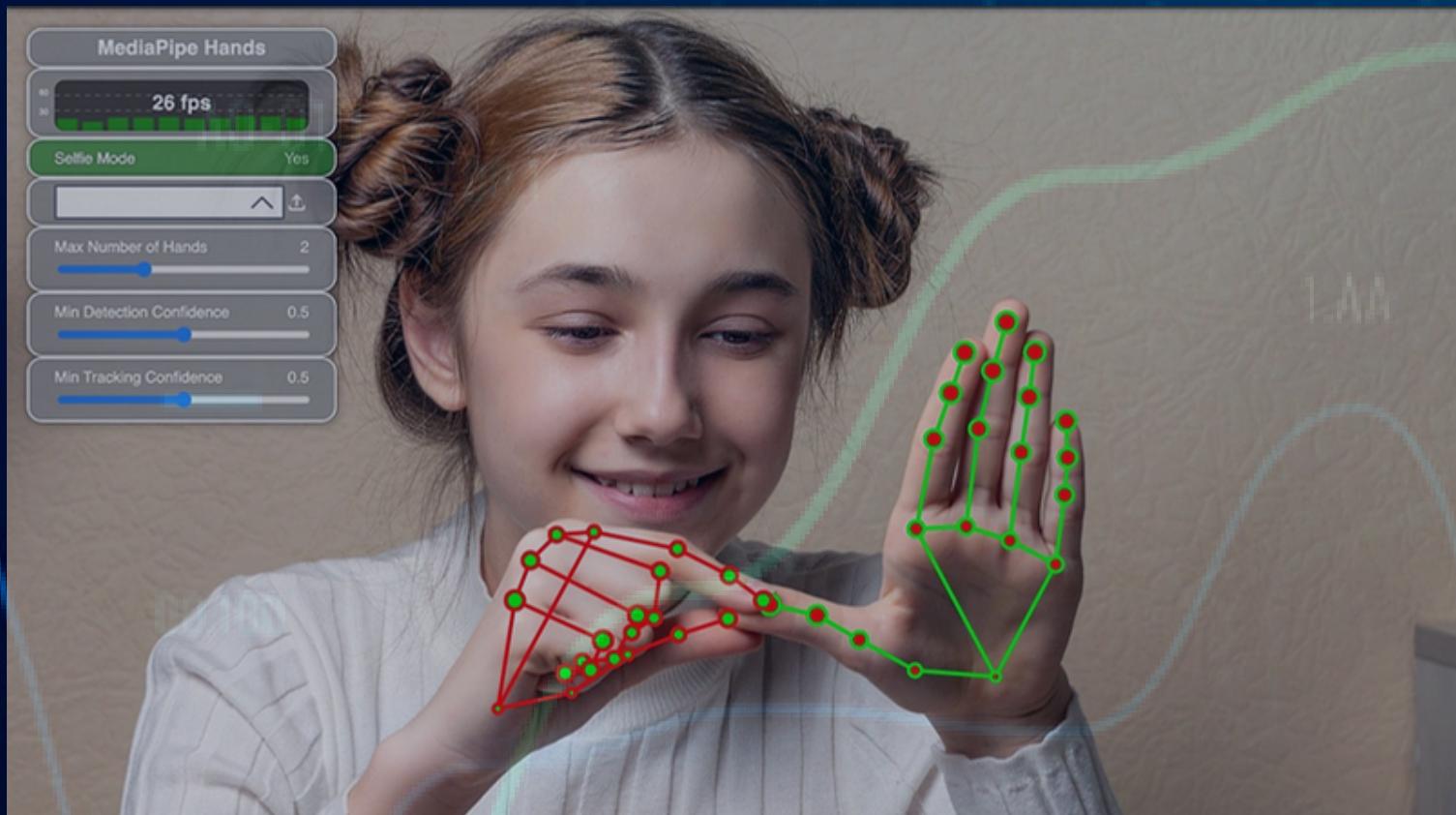
**NEXT PAGE >>**

# 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.

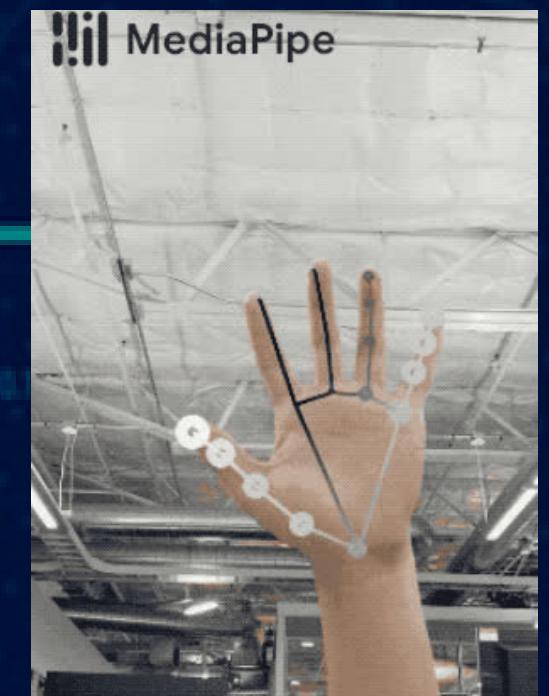


# MediaPipe Hands

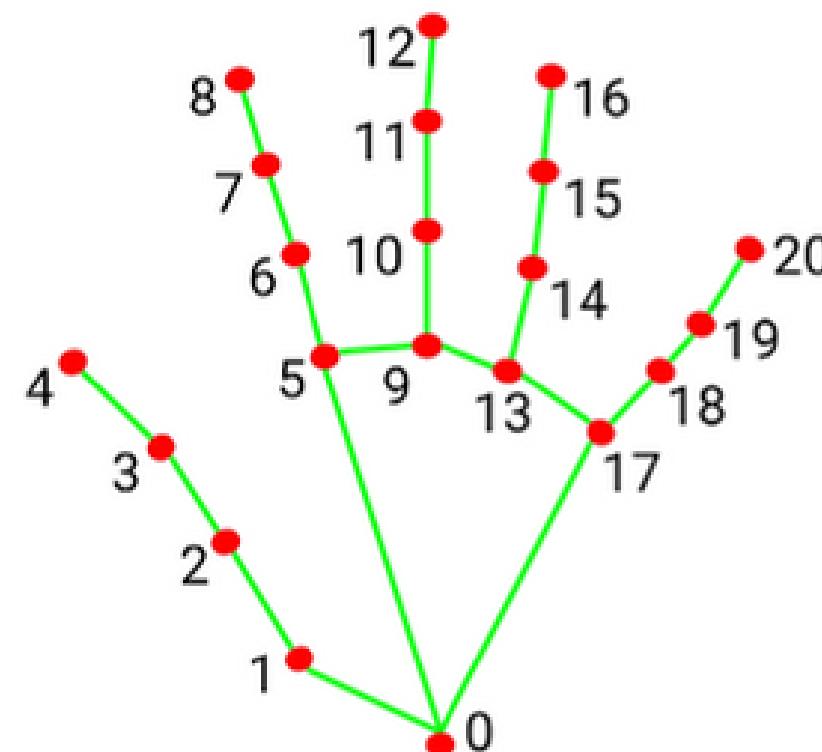


Today we will learn to use MediaPipe, which is a Machine Learning Solutions or a ready-made program from Google that can be used to do Hand Tracking accurately. And as fast as real-time tracking.

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.

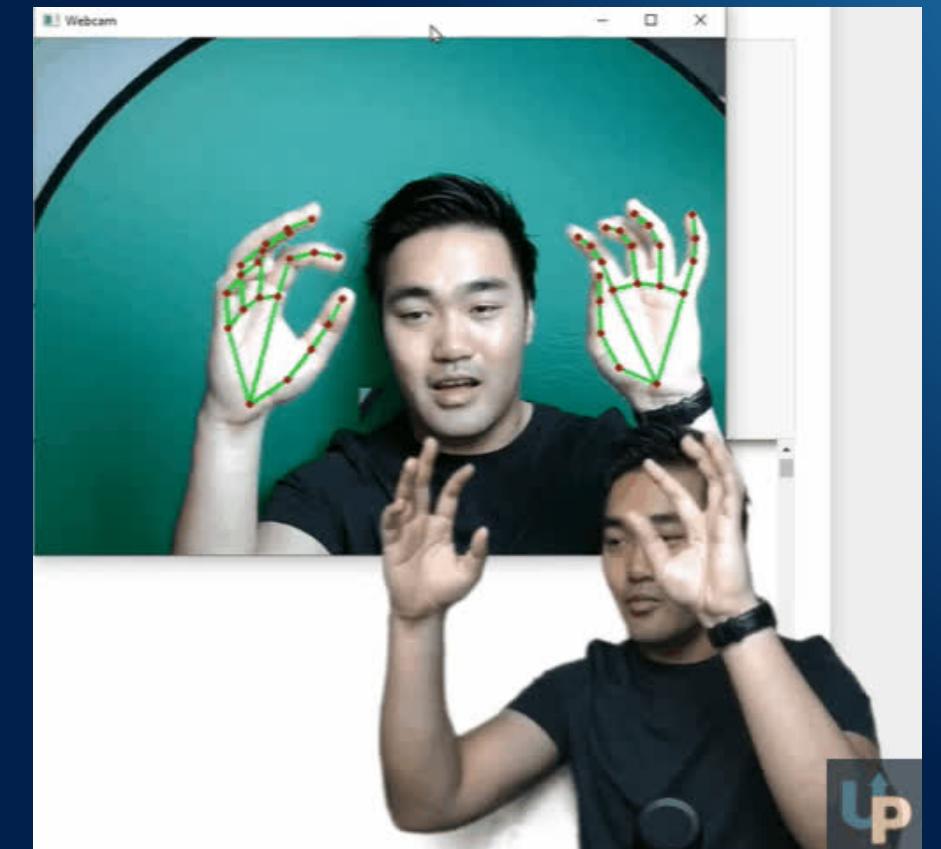
# Hand Tracking

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

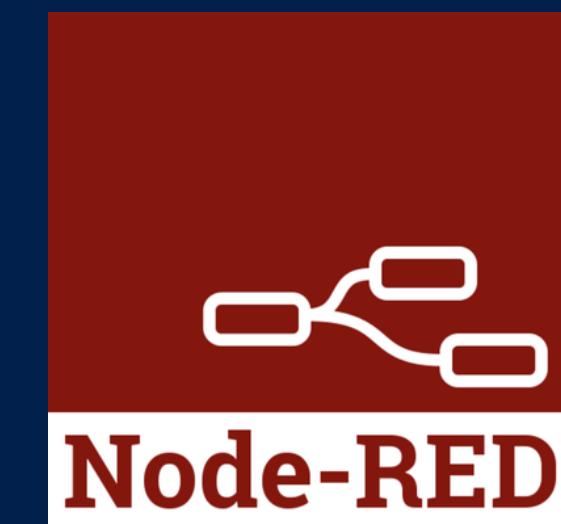
**Q1**

```
pip3 install mediapipe  
pip3 install opencv-python
```

**Q2**

install NodeRed

```
pip3 install requests
```

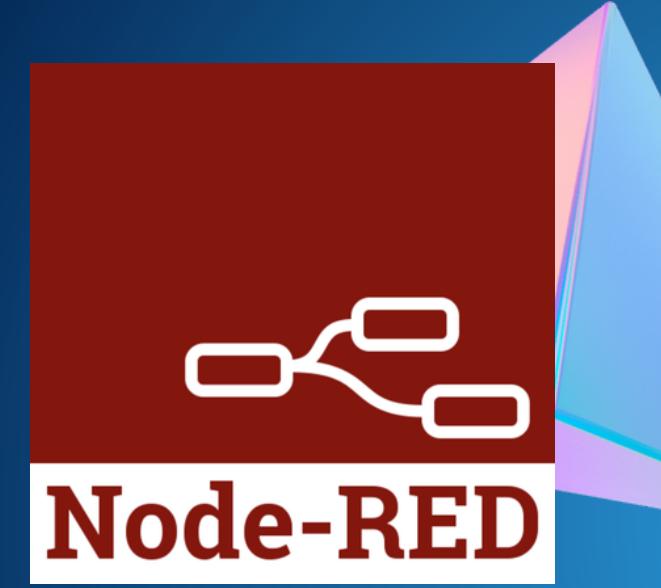


# Hand Count Finger Send To NodeRed

Q2 |

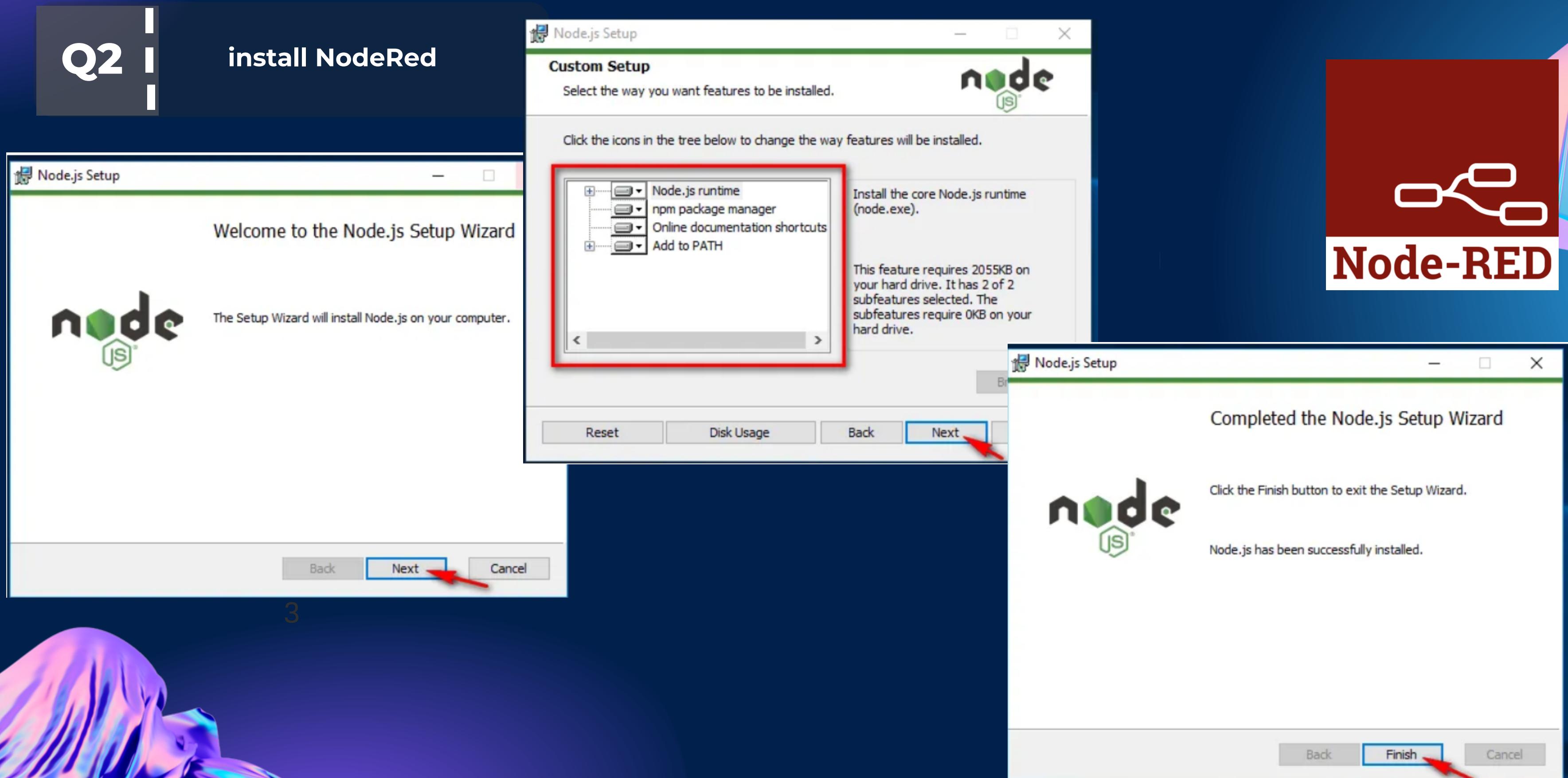
install NodeRed

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 states: "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Two large green buttons are prominently featured: "Download for Windows (x64)" and "10.15.0 LTS Recommended For Most Users". The "10.15.0 LTS" button is highlighted with a red border. To its right is another button: "11.7.0 Current Latest Features". Below these buttons, there are links for "Other Downloads", "Changelog", and "API Docs" for both versions. A note below the buttons says: "Or have a look at the [Long Term Support \(LTS\) schedule](#)." At the bottom, there's a call to action: "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

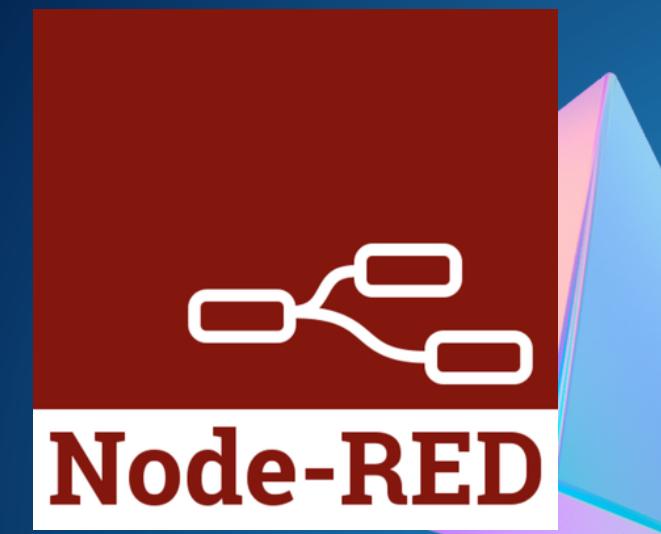
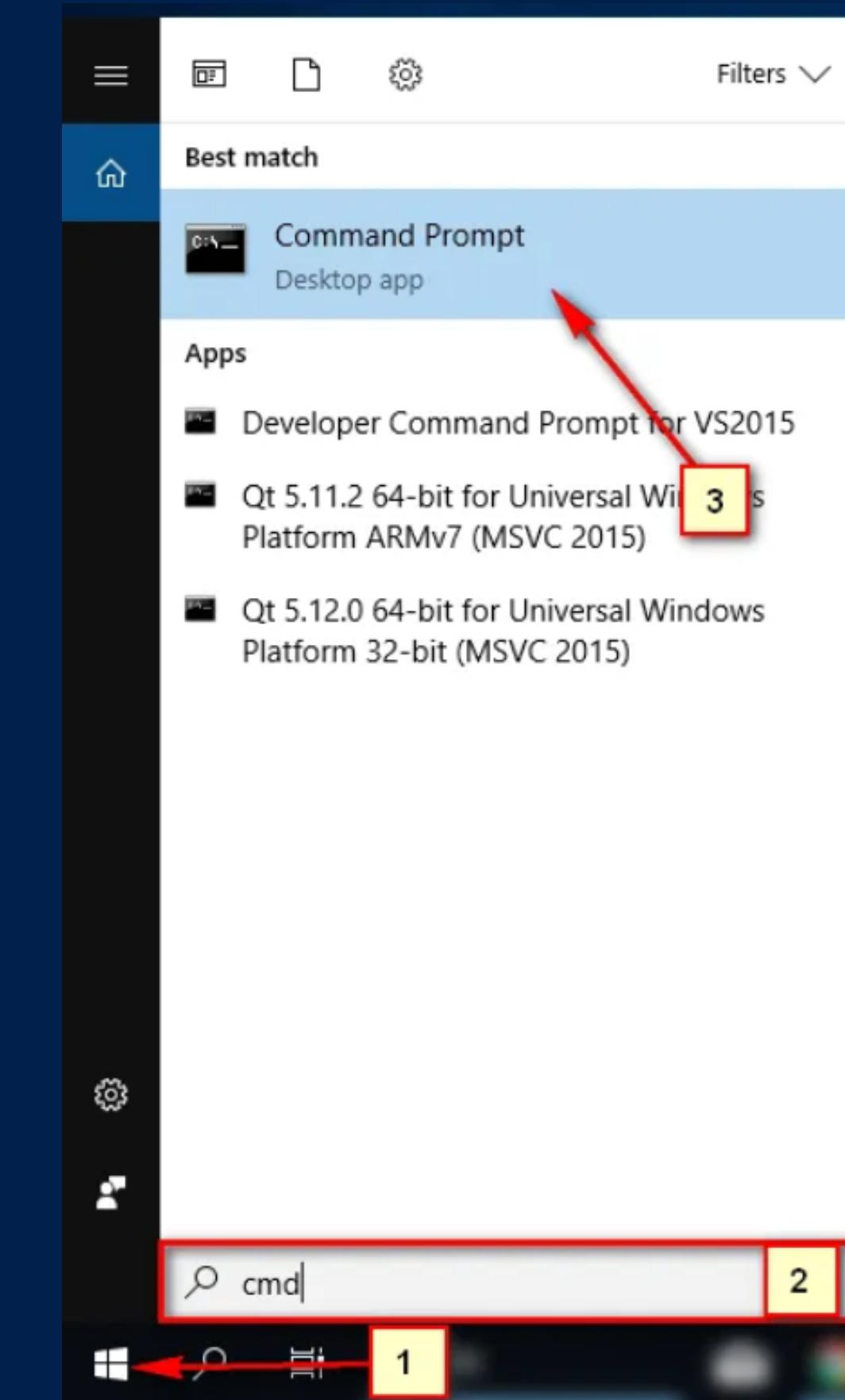


## 2. Check Version NodeJS with CMD

3

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

3

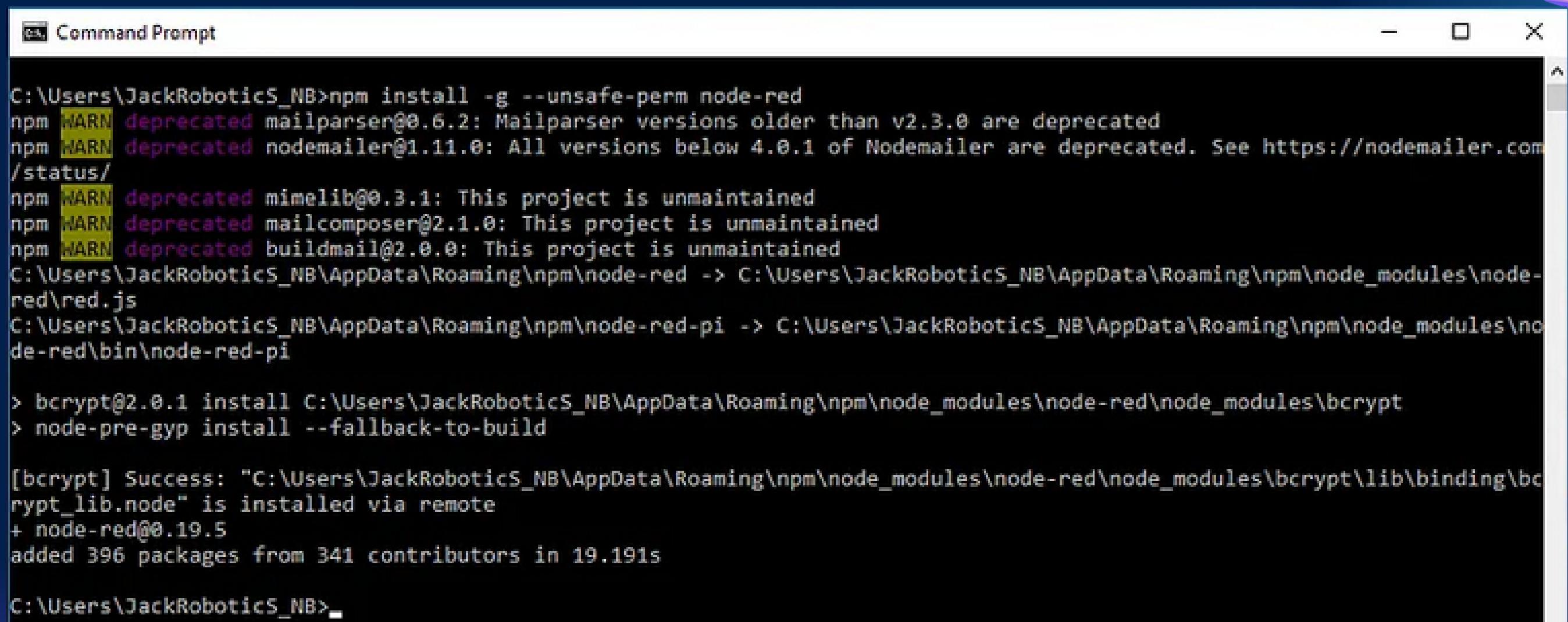


# Hand Count Finger Send To NodeRed

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]

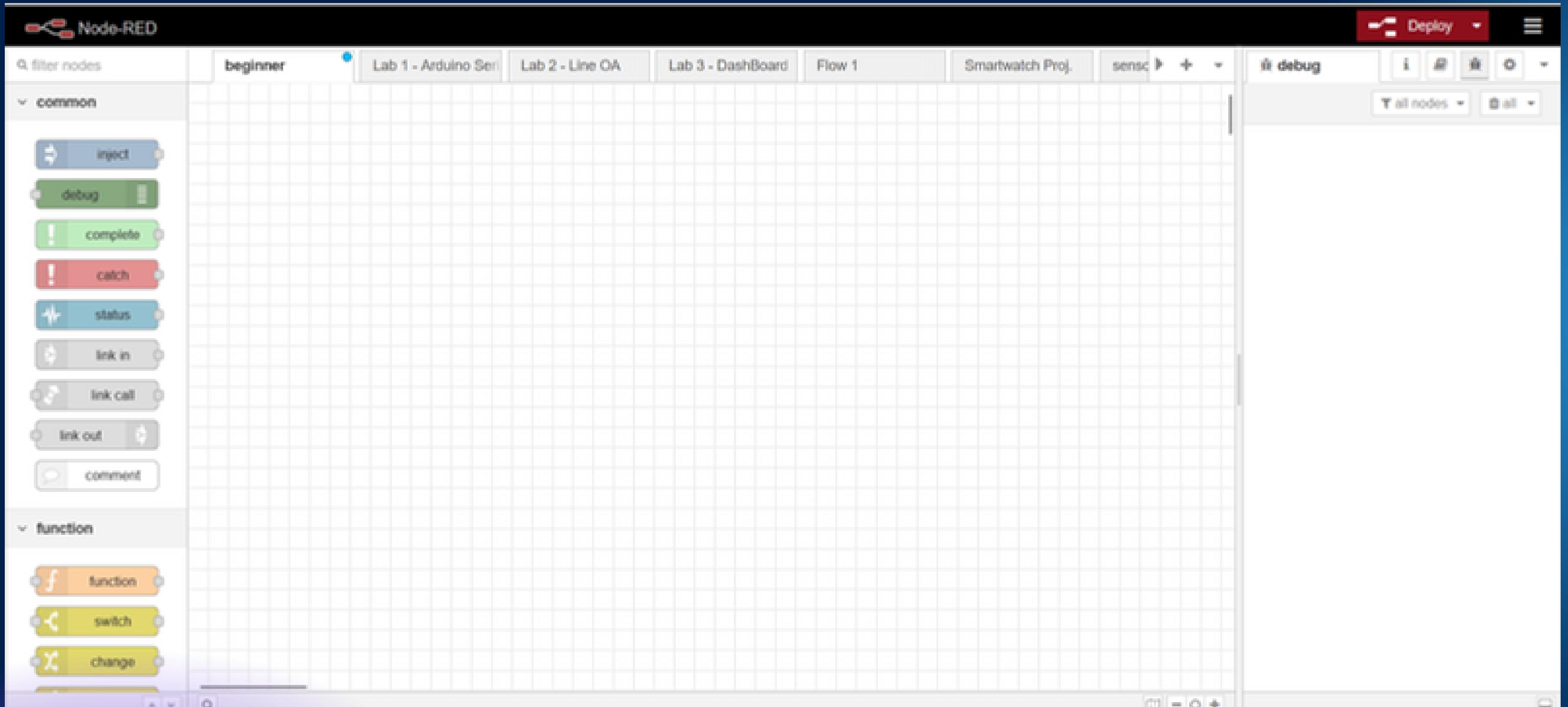
Welcome to Node-RED
-----
22 Jan 20:04:21 - [info] Node-RED version: v0.19.5
22 Jan 20:04:21 - [info] Node.js  version: v10.15.0
22 Jan 20:04:21 - [info] Windows_NT 10.0.17134 x64 LE
22 Jan 20:04:21 - [info] Loading palette nodes
22 Jan 20:04:22 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
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

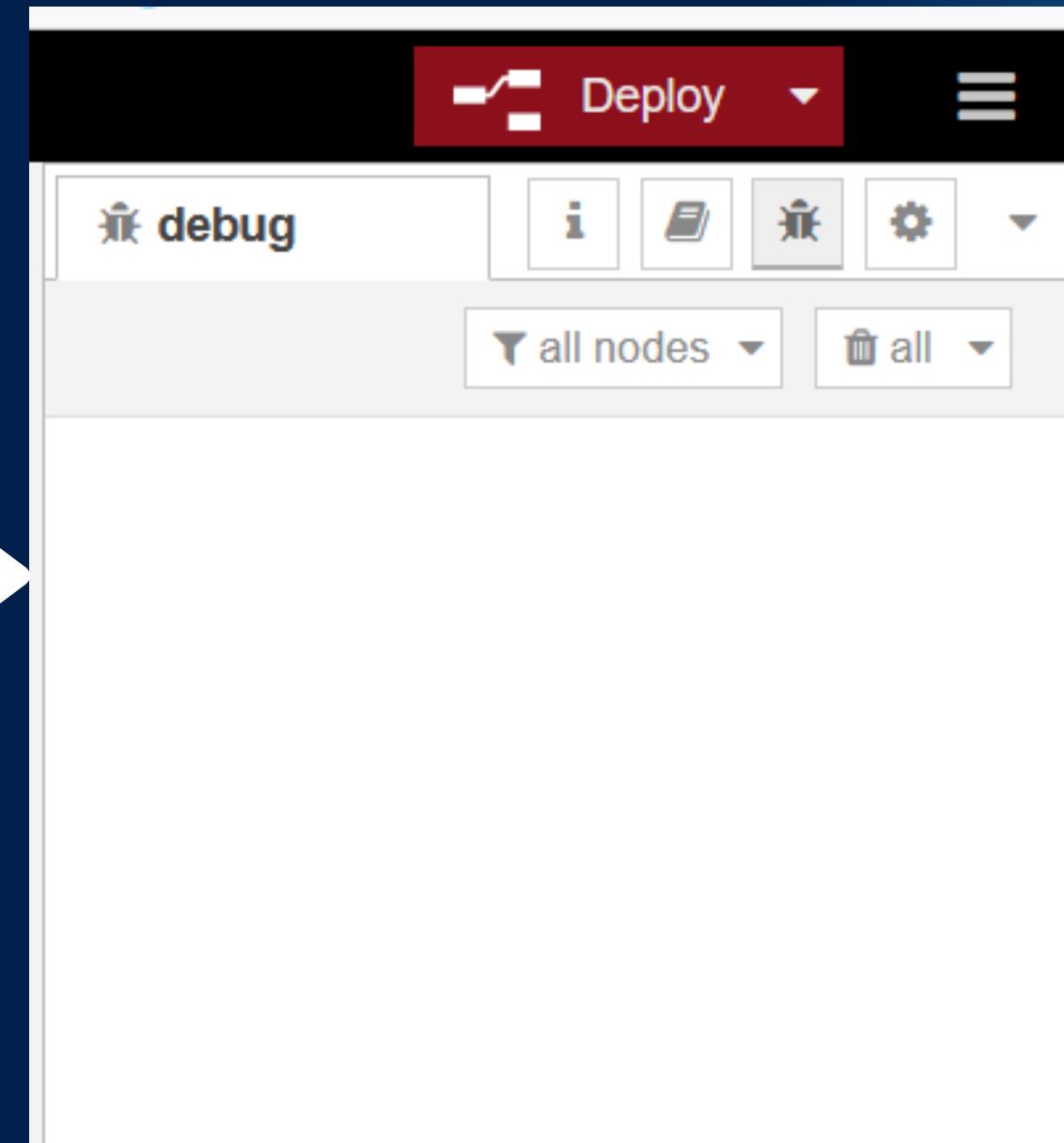


# Workspace **Node-Red**

# Deploy Button



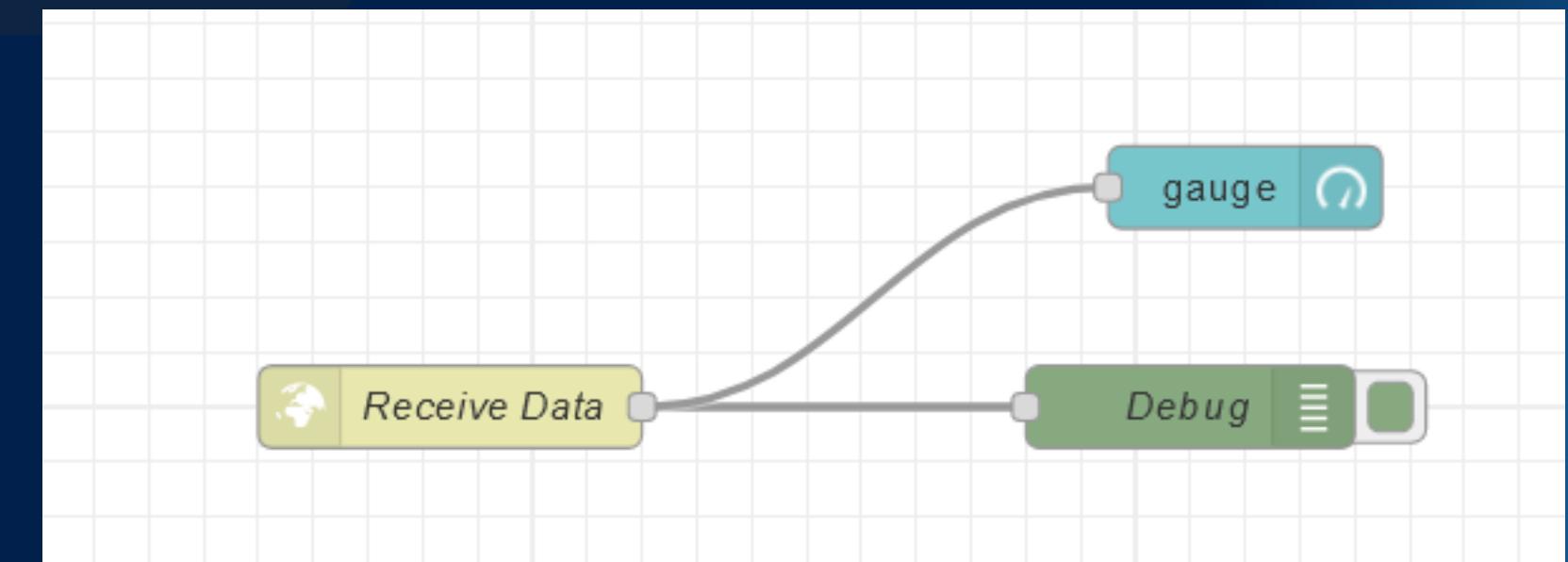
- บุ่มนี้ไว้กดทุกครั้งเวลา อัพเดต **block code**  
หลังแก้ไขเสร็จ



นำไว้ดู Debug เวลาเมื่อค่าเข้ามาจากการส่งข้อมูล  
**Data Value / Struct after processing**

# Q3

## Configure padlet in node red



**double-click in nodes Receive Data**

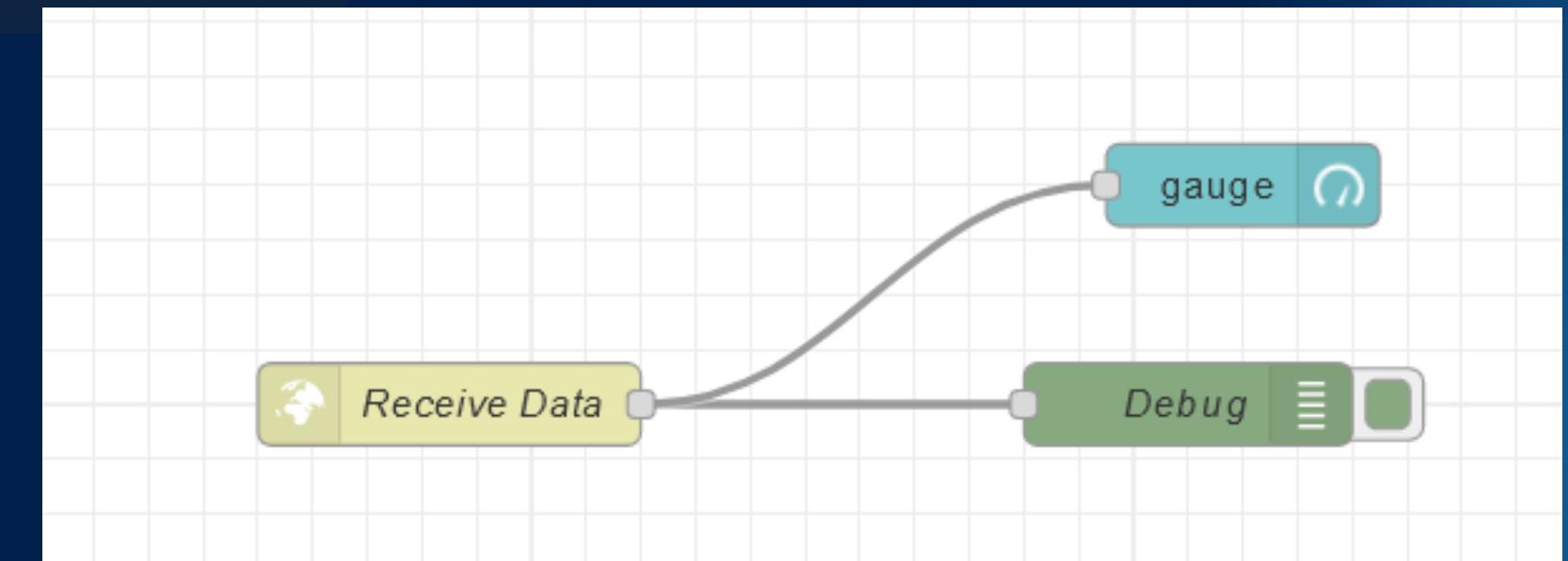
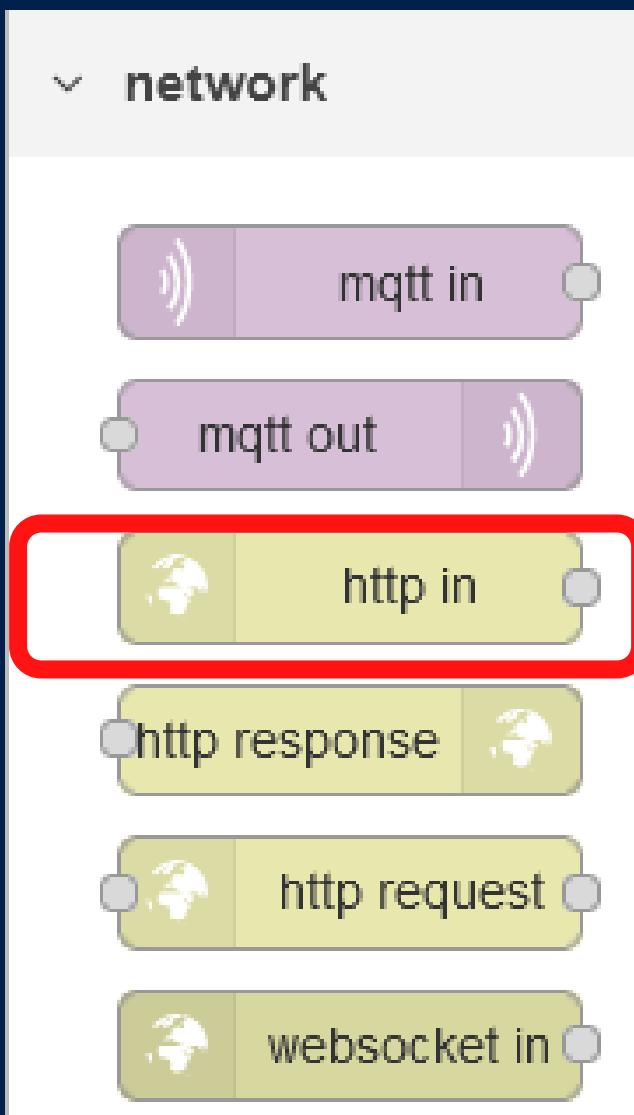
The configuration dialog for the 'Receive Data' node is shown. It includes fields for Method (set to POST), Accept file uploads? (unchecked), URL (set to '/receive-data'), and Name (set to 'Receive Data'). The URL field is highlighted with a red border, and the text 'config your URL you need' is displayed next to it.

Method	POST
Accept file uploads?	<input type="checkbox"/>
URL	/receive-data
Name	Receive Data

**config your URL you need**

# Q3

## Configure padlet in node red



**double-click in nodes Receive Data**

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

- Method: POST
- Accept file uploads?:
- URL: /receive-data **config your URL you need**
- Name: Receive Data

# Q3

## Hand Count Coding

# Our Wisdom

```
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)
```

**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()
```

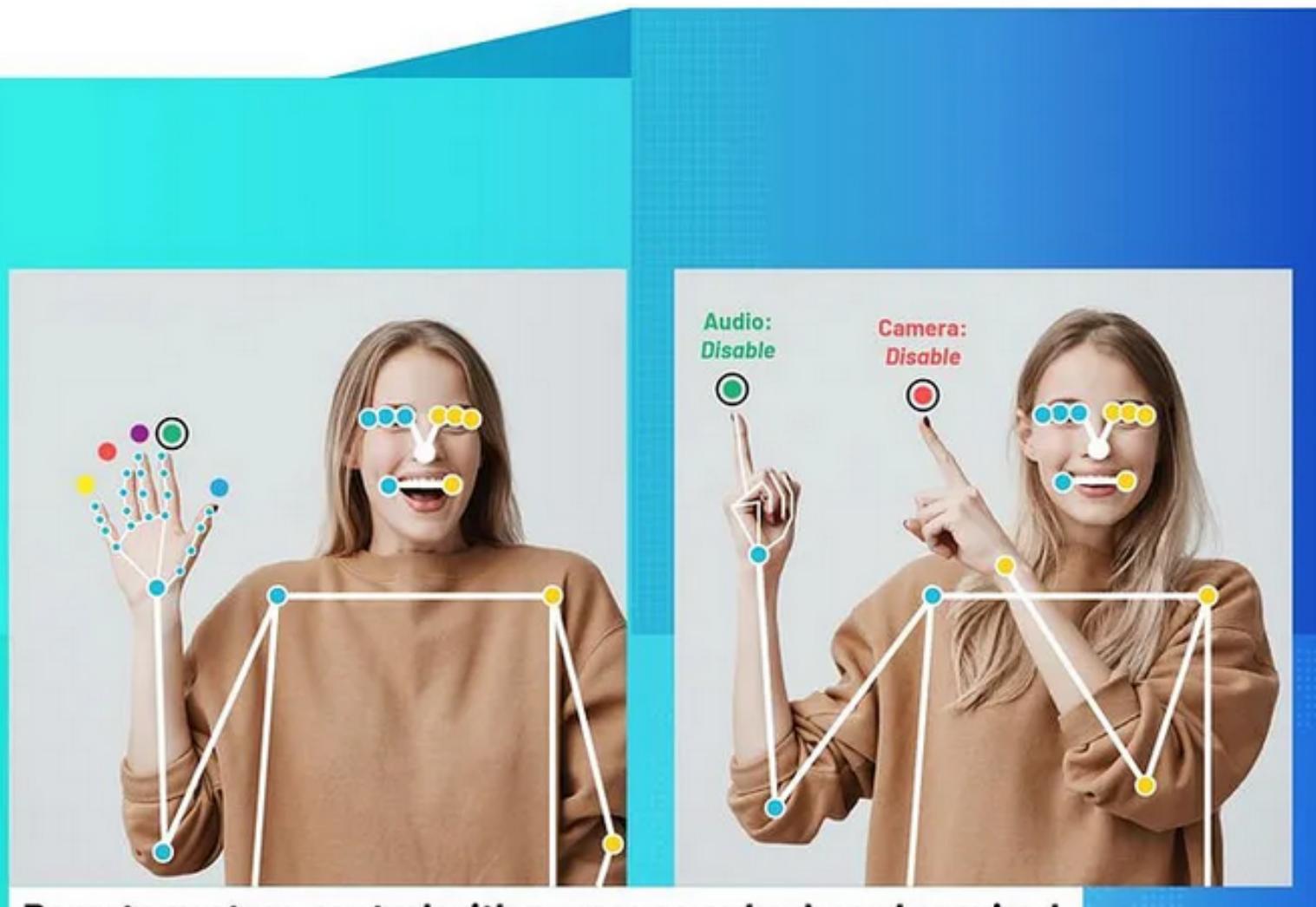


**github code :**

**[https://github.com/TheedBoyZ/Cod eBotix\\_Camp\\_Image](https://github.com/TheedBoyZ/Cod eBotix_Camp_Image)**

# Intelligence

## Google's MediaPipe Holistic Interface

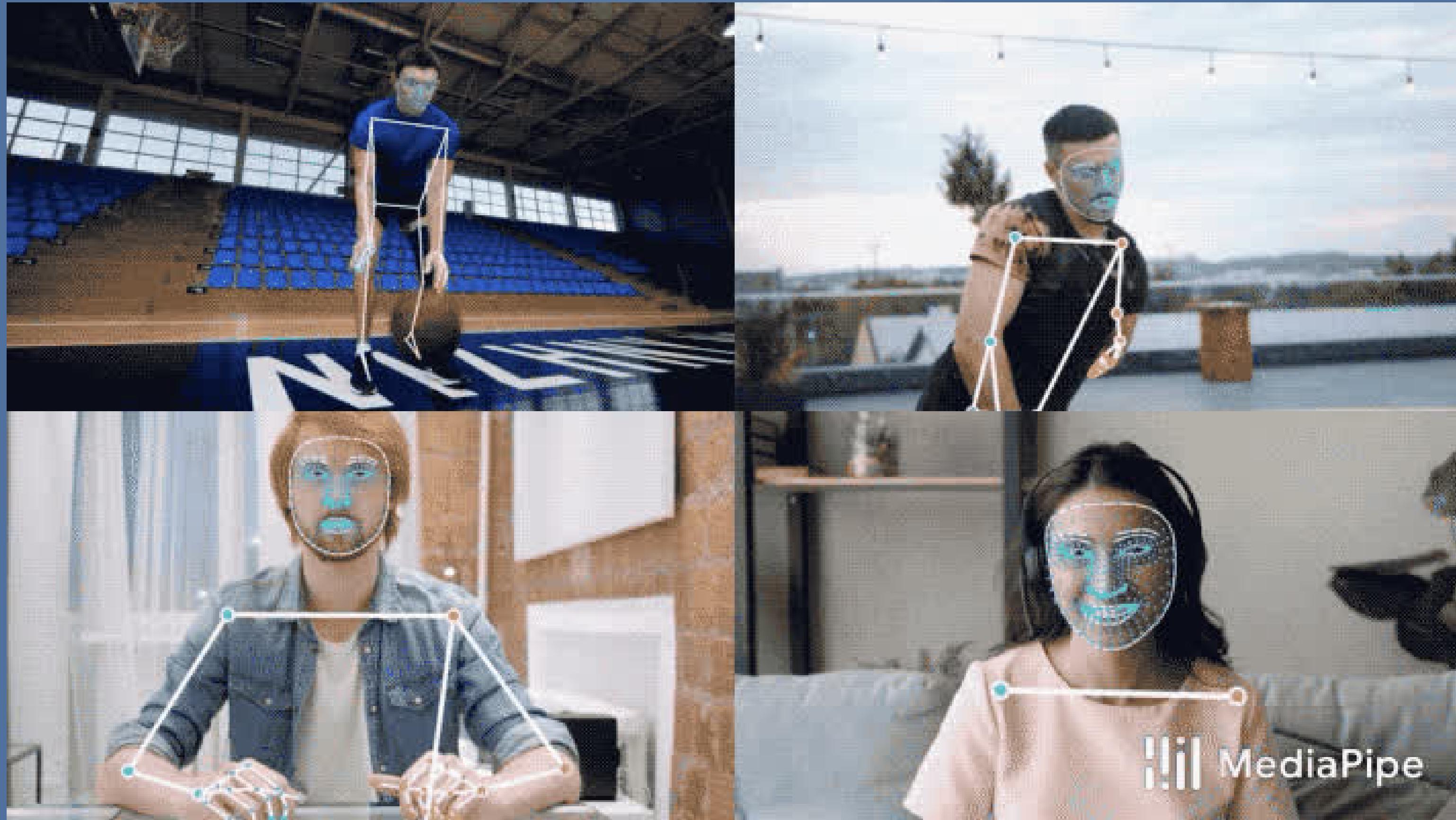


This gesture control and command technique will take us to the next level. And allow us to use new innovations that other devices cannot do

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

[https://mediapipe.dev/demo/holistic\\_remote/](https://mediapipe.dev/demo/holistic_remote/)





[https://mediapipe.dev/demo/holistic\\_remote/](https://mediapipe.dev/demo/holistic_remote/)

# Holistic demo

ໂຄມດກາຣໃຊ້ງານຕ່າງໆ

1

1 ນຶ່ງ ມື້ອໜ້າຍ ຮັບ ບວາ

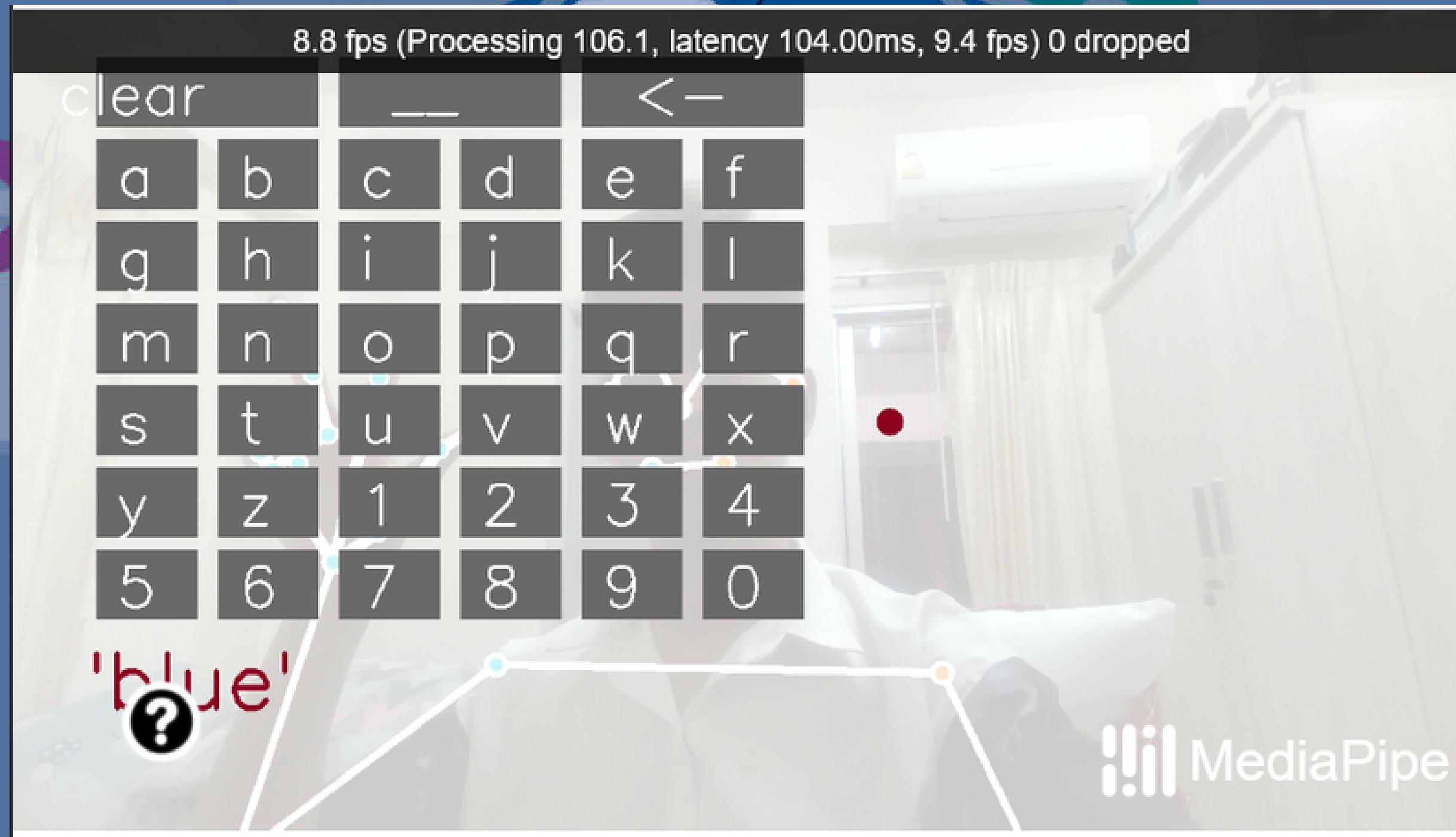
2

3 ນຶ່ງ ມື້ອໜ້າຍ

3

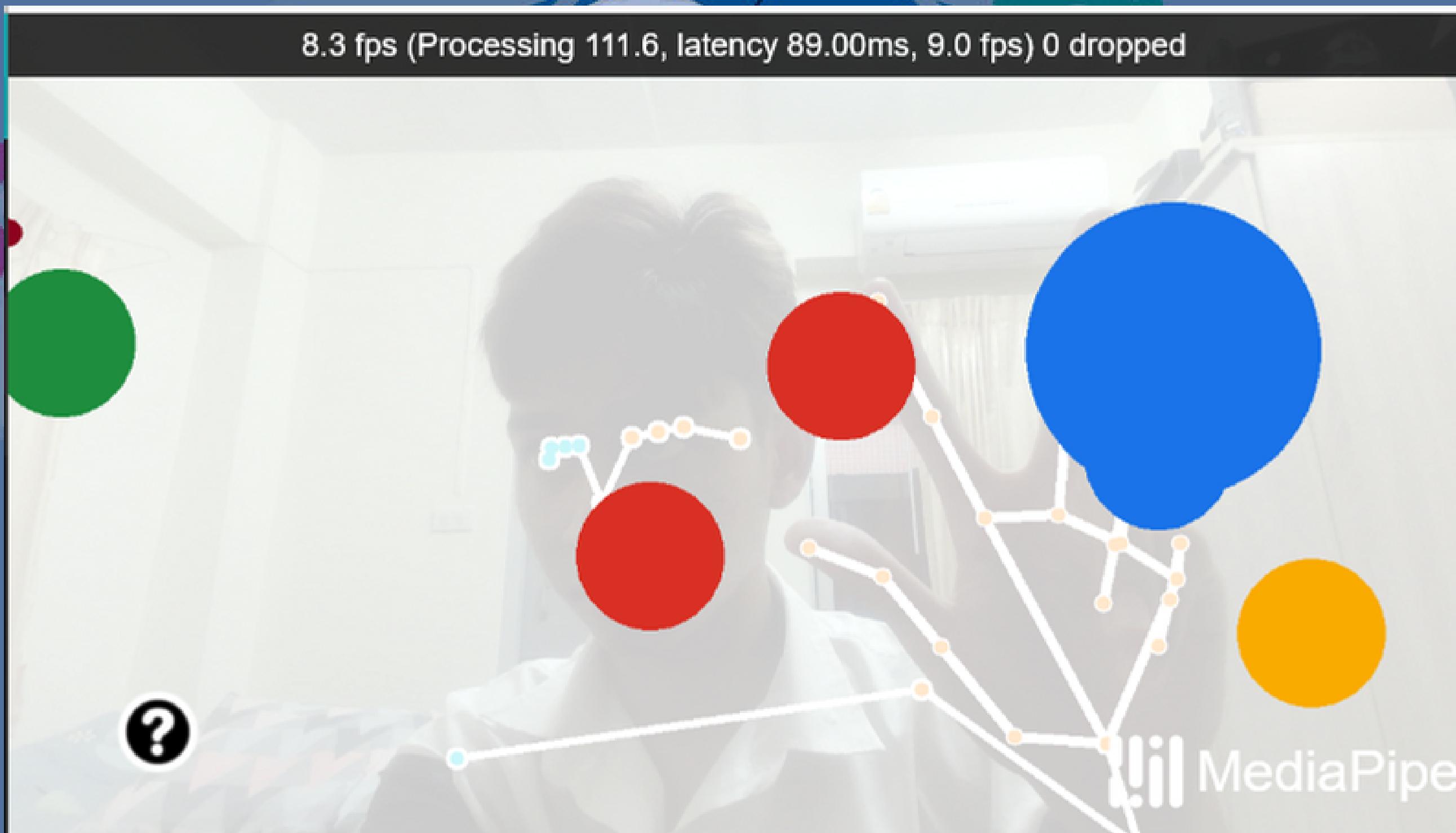
3 ນຶ່ງ ມື້ບວາ

# Holistic demo



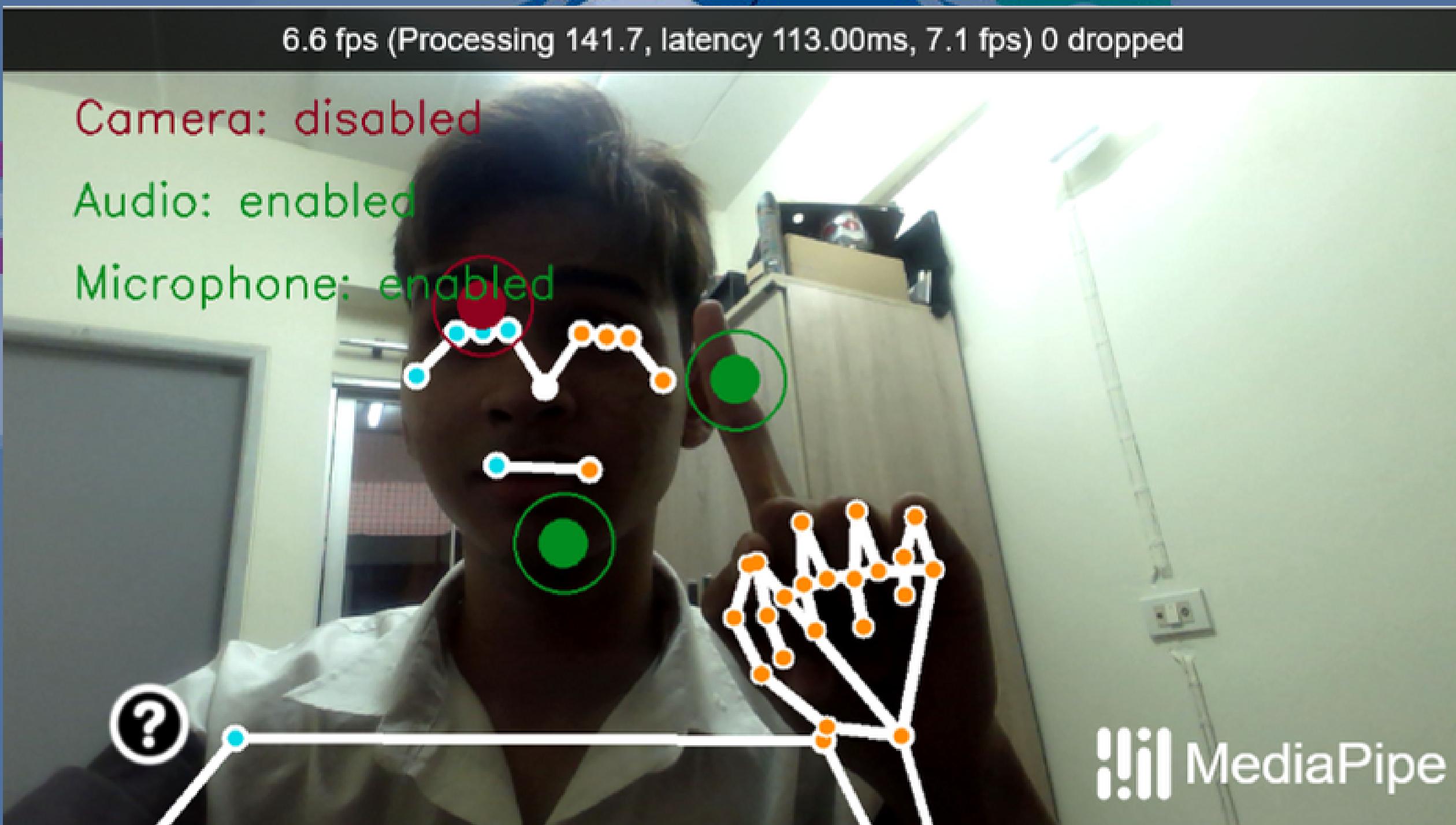
ลองเขียนชื่อตัวเอง เป็นภาษาอังกฤษ

# 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



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

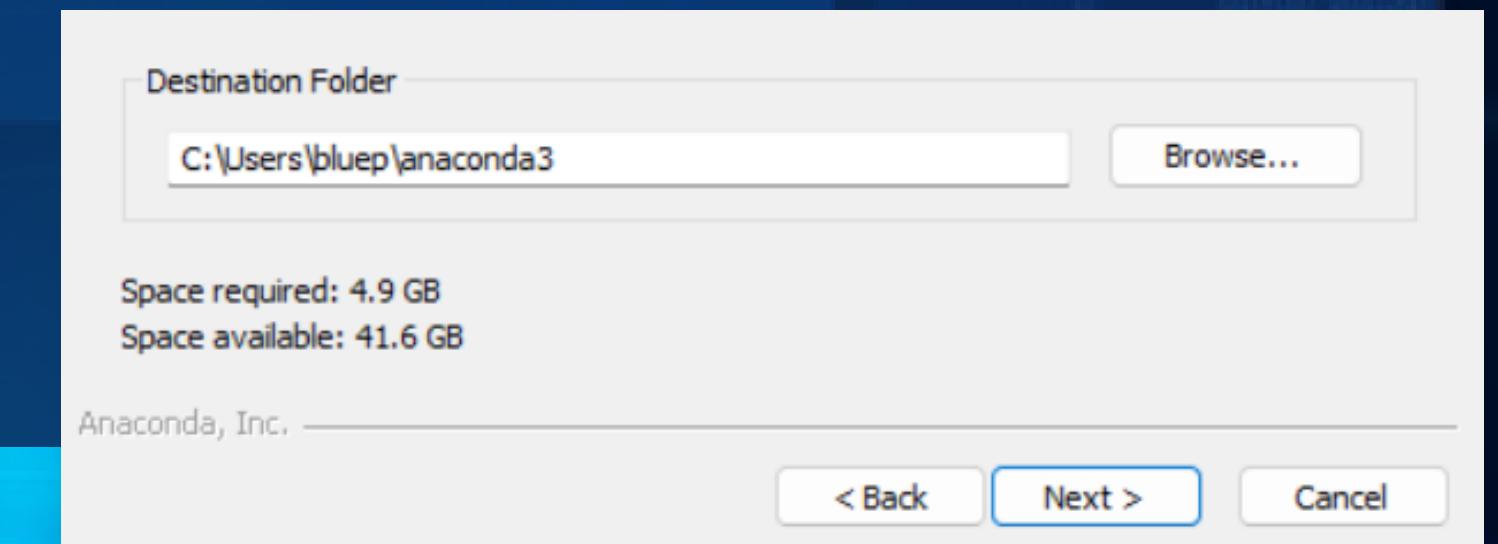
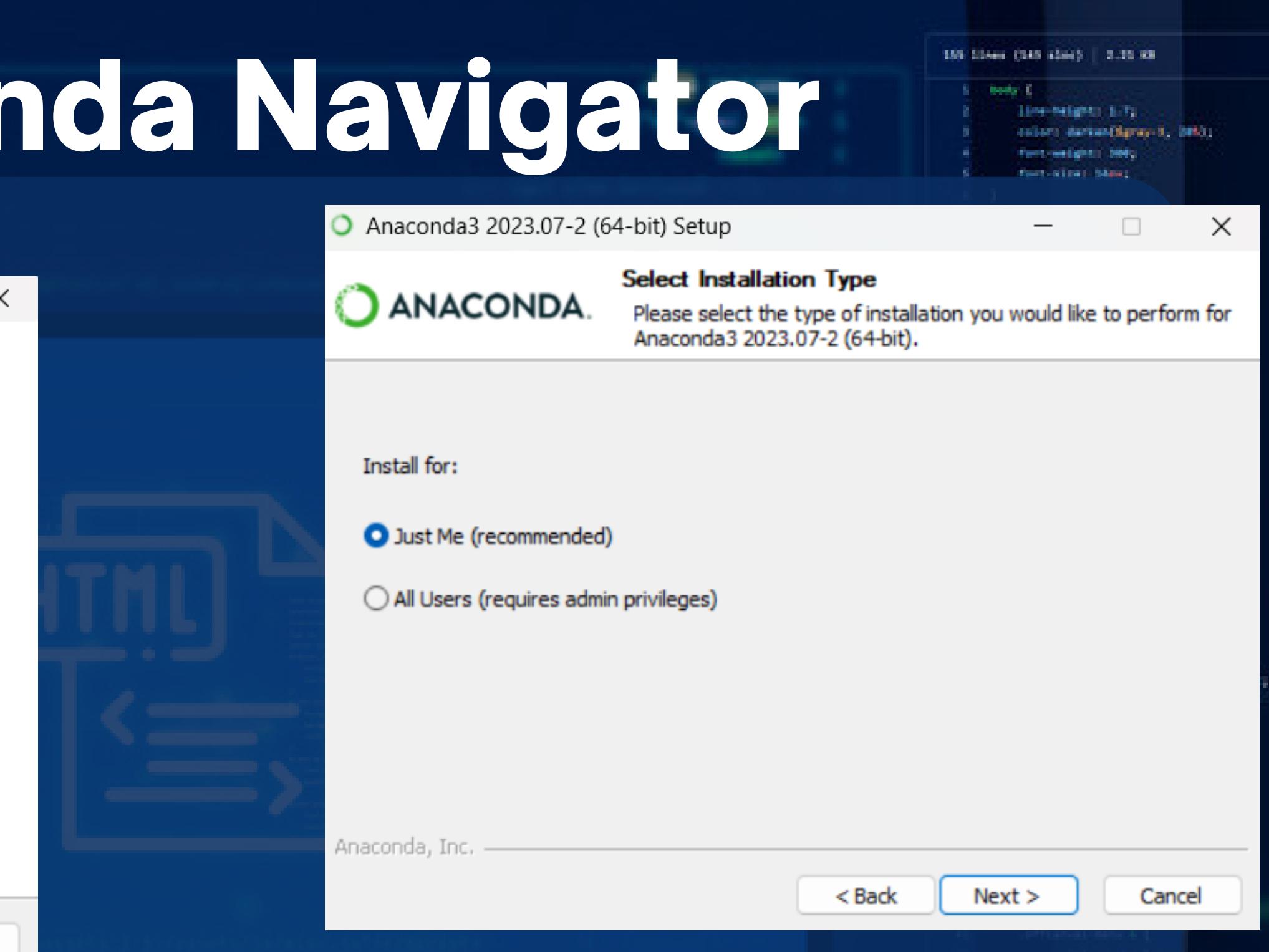
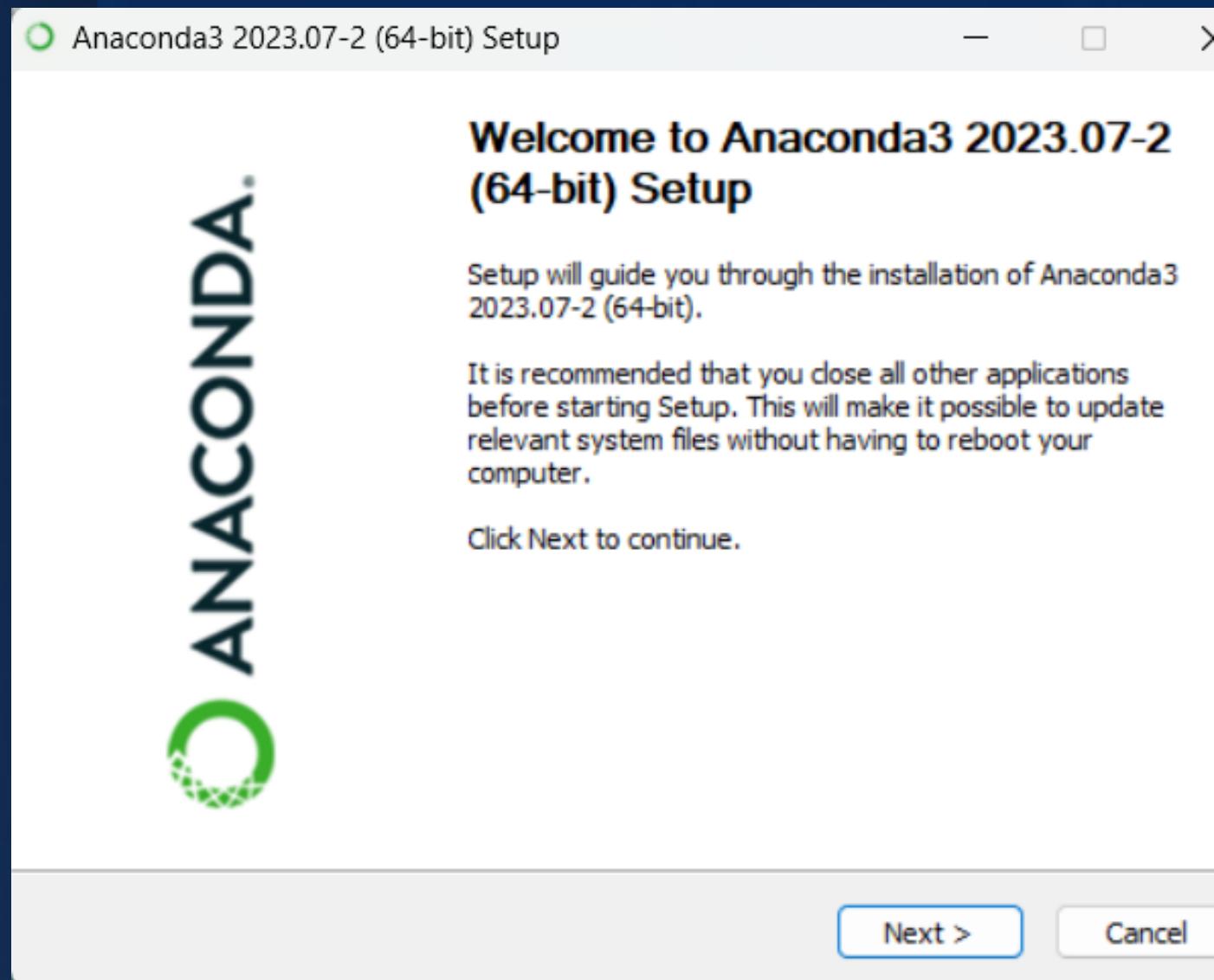
Code in the Cloud

Download

Get Additional Installers



# Anaconda Navigator



# Anaconda Navigator

The image shows the Anaconda Navigator application interface. The main window displays a grid of 12 data science tools, each with a icon, name, version, and a brief description. A sidebar on the left provides links to 'Anaconda Notebooks', 'Documentation', and 'Anaconda Blog'. A top navigation bar includes 'File' and 'Help' menus, and a 'Connect' button. A status bar at the bottom shows system information.

**Left Sidebar:**

- Home
- Environments
- Learning
- Community

**Anaconda Notebooks:** Cloud notebooks with hundreds of packages ready to code. [Learn More](#)

**Bottom Links:**

- A Full Python IDE directly from the browser
- Documentation
- Anaconda Blog

**Top Status Bar:**

100% (0.0s total) 0.00 KB

**Right Panel (Status Bar):**

- Code Editor (Python 3)
- Line height: 1.5px
- Color scheme: Gray 1, 100%
- Font weight: 300px
- Font size: 14px

**Central Grid (Applications):**

Icon	Name	Version	Description	Action Buttons
	DataSpell	0.0.1	DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the Intelligent Python and AI coding assistance of PyCharm in the user-friendly environment.	<a href="#">Install</a> <a href="#">Launch</a>
	CMDline Prompt	0.1.1	Run a command-line terminal with your current environment from Navigator activated.	<a href="#">Launch</a>
	JupyterLab	0.1.1	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and architecture.	<a href="#">Launch</a>
	Notebook	4.8.2	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	<a href="#">Launch</a>
	Powershell Prompt	0.0.1	Run a PowerShell terminal with your current environment from Navigator activated.	<a href="#">Launch</a>
	IPyConsole	0.0.0	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical controls, and more.	<a href="#">Launch</a>
	Spyder	3.5.1	Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.	<a href="#">Launch</a>
	VS Code	1.82.2	Streamlined code editor with support for development operations like debugging, task running and version control.	<a href="#">Launch</a>
	Datablone	0.0.0	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 Datablone online for free.	<a href="#">Launch</a>
	IBM Watson Studio Cloud	0.0.0	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.	<a href="#">Launch</a>
	ORACLE Cloud Infrastructure	0.0.0	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.	<a href="#">Launch</a>
	Gloviz	1.2.4	Multidimensional data visualization across files. Explore relationships within and among related datasets.	<a href="#">Install</a>
	Orange 3	3.32.0	Component-based data mining framework. Data visualization and data analysis for machine learning.	
	PyCharm Professional	2021.1.1	A full-fledged IDE by JetBrains for both scientific and web Python development.	
	RStudio	1.4.106	A set of integrated tools designed to help you be more productive with R. Includes R	

# Anaconda Navigator

The screenshot shows the Anaconda Navigator application window. At the top, there is a header bar with a back/forward button, a search bar containing "localhost:8889/lab/tree/Desktop/TEST", a zoom level indicator (90%), and several icons. Below the header is a menu bar with File, Edit, View, Run, Kernel, Navigator, Tabs, Settings, and Help.

The main area is divided into two sections: a file browser on the left and a launcher on the right.

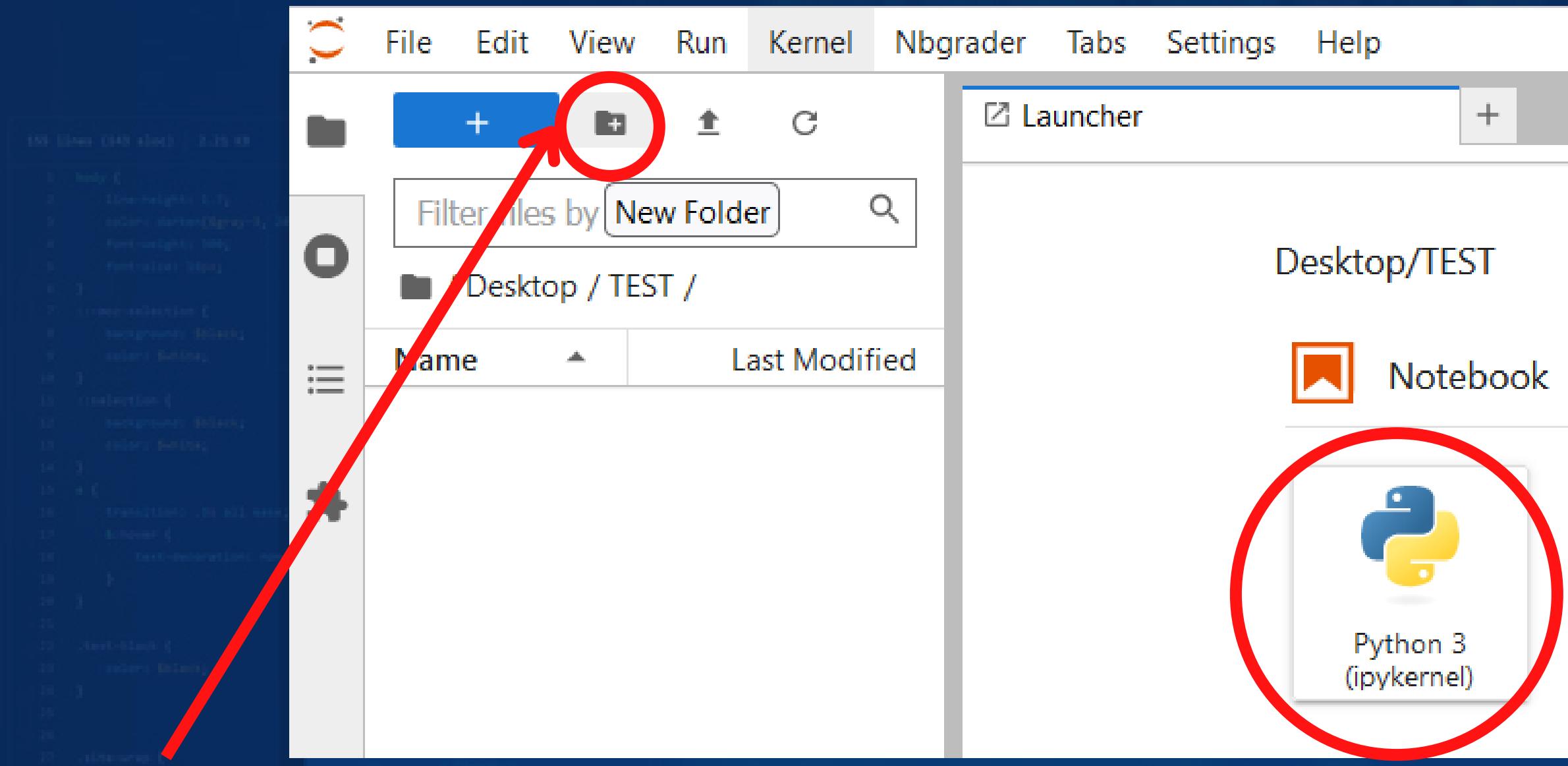
**File Browser:** The left panel displays a tree view of the directory structure under "Desktop/TEST". A search bar at the top of this panel allows filtering files by name. The columns shown are Name and Last Modified.

**Launcher:** The right panel is titled "Launcher" and contains the following sections:

- Notebook:** Shows a Python 3 (system) kernel icon.
- Console:** Shows a Python 3 (system) kernel icon.
- Other:** Shows icons for Terminal, Text File, Markdown File, Python File, and Show Content Help.

At the bottom of the window, there is a toolbar with various icons and a status bar indicating "100 lines (ctrl+shift) | 2.00 MB".

# Anaconda Navigator

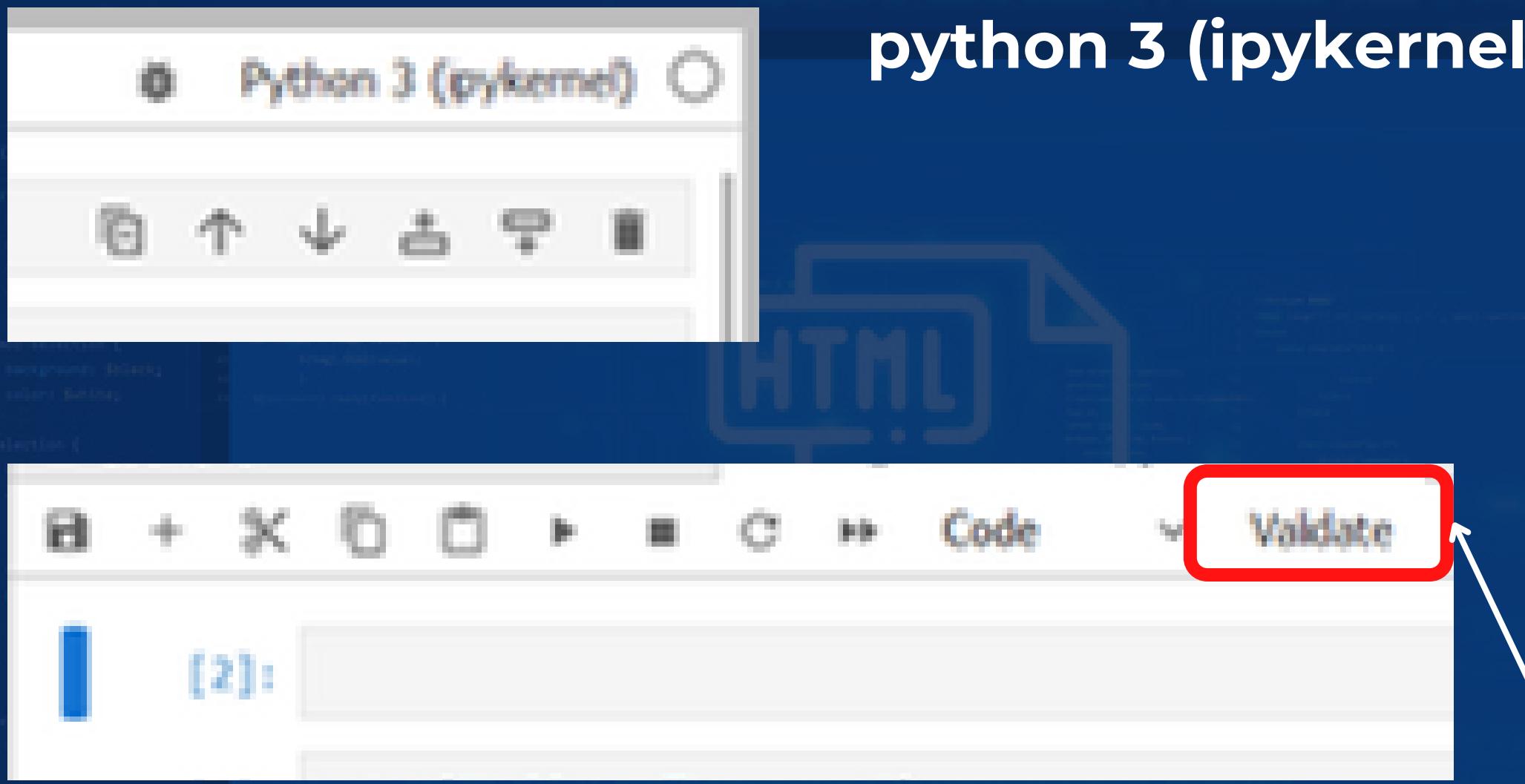


สร้าง Floder ใหม่

สร้าง สร้างไฟล์ .ipynb เพื่อเขียน  
โปรแกรมใน jupyter notebook

# Anaconda Navigator

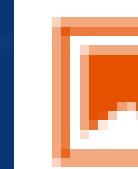
python 3 (ipykernel) ในการรันโค้ด



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

```
[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)
```

```
[12]: 8
```

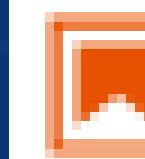
```
[13]: b = 10
```

```
[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
```

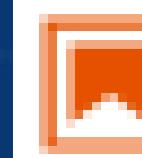
```
[19]: print('Hi {} {:.1f}'.format(name , sc))  
      Hi Blue 3.7
```

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

```
[ ]:
```

# Jupyter Notebook

## image processing



ch3\_basic\_python.ipynb



# Jupyter Notebook

## image processing

Python เบื้องต้น

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

scimath
```

```
[4]: เลือก操作คำนวณ (Operators)
+ * / 
% หารเหลือเศษ
// หารปีผลเศษ
** ยกกำเนิด
```

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

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

```
[6]: 5.0
```

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

```
[10]: 2
```

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

```
[11]: 1
```

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

```
[12]: 8
```

# Jupyter Notebook

## image processing

### Variable

```
[16]: raca = 100  
quai = raca / 4  
naam = quai  
print(quai)
```

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
```

```
100 lines (diff 100) 2.00 KB  
1 body {  
2   line-height: 1.5;  
3   color: #333333; font-size: 1em;  
4   font-weight: 400;  
5   font-style: normal;  
6 }  
7 .CodeMirror-line {  
8   background-color: black;  
9 }
```

# 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
```

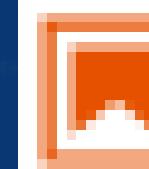
```
[64]: for i in book_lish.values():  
      print(i)
```

```
27  
171  
18.4
```

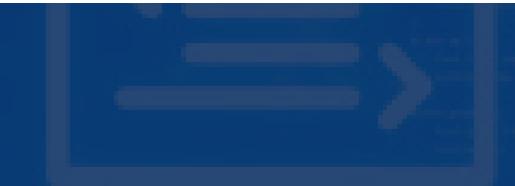
```
100 lines (diff 100) | 2.00 KB  
1 body {  
2   background-color: #f0f0f0;  
3   color: inherit; font-family: inherit;  
4   font-weight: inherit;  
5   font-size: inherit;  
6 }  
7 .cell {  
8   border: 1px solid #ccc;  
9   padding: 10px;  
10  background-color: white;  
11  color: black;  
12 }  
13 .cell .cell_type {  
14   margin-bottom: 10px;  
15 }  
16 .cell .cell_input {  
17   border: 1px solid #ccc;  
18   padding: 5px;  
19 }  
20 .cell .cell_output {  
21   margin-top: 10px;  
22 }  
23 .cell .cell_text {  
24   margin-top: 10px;  
25 }  
26 .cell .cell_code {  
27   margin-top: 10px;  
28 }  
29 .cell .cell_html {  
30   margin-top: 10px;  
31 }  
32 .cell .cell_error {  
33   margin-top: 10px;  
34 }  
35 .cell .cell_stdout {  
36   margin-top: 10px;  
37 }  
38 .cell .cell_stderr {  
39   margin-top: 10px;  
40 }  
41 .cell .cell_stderr {  
42   margin-top: 10px;  
43 }  
44 .cell .cell_stderr {  
45   margin-top: 10px;  
46 }  
47 .cell .cell_stderr {  
48   margin-top: 10px;  
49 }  
50 .cell .cell_stderr {  
51   margin-top: 10px;  
52 }  
53 .cell .cell_stderr {  
54   margin-top: 10px;  
55 }  
56 .cell .cell_stderr {  
57   margin-top: 10px;  
58 }  
59 .cell .cell_stderr {  
60   margin-top: 10px;  
61 }  
62 .cell .cell_stderr {  
63   margin-top: 10px;  
64 }  
65 .cell .cell_stderr {  
66   margin-top: 10px;  
67 }  
68 .cell .cell_stderr {  
69   margin-top: 10px;  
70 }  
71 .cell .cell_stderr {  
72   margin-top: 10px;  
73 }  
74 .cell .cell_stderr {  
75   margin-top: 10px;  
76 }  
77 .cell .cell_stderr {  
78   margin-top: 10px;  
79 }  
80 .cell .cell_stderr {  
81   margin-top: 10px;  
82 }  
83 .cell .cell_stderr {  
84   margin-top: 10px;  
85 }  
86 .cell .cell_stderr {  
87   margin-top: 10px;  
88 }  
89 .cell .cell_stderr {  
90   margin-top: 10px;  
91 }  
92 .cell .cell_stderr {  
93   margin-top: 10px;  
94 }  
95 .cell .cell_stderr {  
96   margin-top: 10px;  
97 }  
98 .cell .cell_stderr {  
99   margin-top: 10px;  
100 }  
101 .cell .cell_stderr {  
102   margin-top: 10px;  
103 }  
104 .cell .cell_stderr {  
105   margin-top: 10px;  
106 }  
107 .cell .cell_stderr {  
108   margin-top: 10px;  
109 }  
110 .cell .cell_stderr {  
111   margin-top: 10px;  
112 }  
113 .cell .cell_stderr {  
114   margin-top: 10px;  
115 }  
116 .cell .cell_stderr {  
117   margin-top: 10px;  
118 }  
119 .cell .cell_stderr {  
120   margin-top: 10px;  
121 }  
122 .cell .cell_stderr {  
123   margin-top: 10px;  
124 }  
125 .cell .cell_stderr {  
126   margin-top: 10px;  
127 }  
128 .cell .cell_stderr {  
129   margin-top: 10px;  
130 }  
131 .cell .cell_stderr {  
132   margin-top: 10px;  
133 }  
134 .cell .cell_stderr {  
135   margin-top: 10px;  
136 }  
137 .cell .cell_stderr {  
138   margin-top: 10px;  
139 }  
140 .cell .cell_stderr {  
141   margin-top: 10px;  
142 }  
143 .cell .cell_stderr {  
144   margin-top: 10px;  
145 }  
146 .cell .cell_stderr {  
147   margin-top: 10px;  
148 }  
149 .cell .cell_stderr {  
150   margin-top: 10px;  
151 }  
152 .cell .cell_stderr {  
153   margin-top: 10px;  
154 }  
155 .cell .cell_stderr {  
156   margin-top: 10px;  
157 }  
158 .cell .cell_stderr {  
159   margin-top: 10px;  
160 }  
161 .cell .cell_stderr {  
162   margin-top: 10px;  
163 }  
164 .cell .cell_stderr {  
165   margin-top: 10px;  
166 }  
167 .cell .cell_stderr {  
168   margin-top: 10px;  
169 }  
170 .cell .cell_stderr {  
171   margin-top: 10px;  
172 }  
173 .cell .cell_stderr {  
174   margin-top: 10px;  
175 }  
176 .cell .cell_stderr {  
177   margin-top: 10px;  
178 }  
179 .cell .cell_stderr {  
180   margin-top: 10px;  
181 }  
182 .cell .cell_stderr {  
183   margin-top: 10px;  
184 }  
185 .cell .cell_stderr {  
186   margin-top: 10px;  
187 }  
188 .cell .cell_stderr {  
189   margin-top: 10px;  
190 }  
191 .cell .cell_stderr {  
192   margin-top: 10px;  
193 }  
194 .cell .cell_stderr {  
195   margin-top: 10px;  
196 }  
197 .cell .cell_stderr {  
198   margin-top: 10px;  
199 }  
200 .cell .cell_stderr {  
201   margin-top: 10px;  
202 }  
203 .cell .cell_stderr {  
204   margin-top: 10px;  
205 }  
206 .cell .cell_stderr {  
207   margin-top: 10px;  
208 }  
209 .cell .cell_stderr {  
210   margin-top: 10px;  
211 }  
212 .cell .cell_stderr {  
213   margin-top: 10px;  
214 }  
215 .cell .cell_stderr {  
216   margin-top: 10px;  
217 }  
218 .cell .cell_stderr {  
219   margin-top: 10px;  
220 }  
221 .cell .cell_stderr {  
222   margin-top: 10px;  
223 }  
224 .cell .cell_stderr {  
225   margin-top: 10px;  
226 }  
227 .cell .cell_stderr {  
228   margin-top: 10px;  
229 }  
230 .cell .cell_stderr {  
231   margin-top: 10px;  
232 }  
233 .cell .cell_stderr {  
234   margin-top: 10px;  
235 }  
236 .cell .cell_stderr {  
237   margin-top: 10px;  
238 }  
239 .cell .cell_stderr {  
240   margin-top: 10px;  
241 }  
242 .cell .cell_stderr {  
243   margin-top: 10px;  
244 }  
245 .cell .cell_stderr {  
246   margin-top: 10px;  
247 }  
248 .cell .cell_stderr {  
249   margin-top: 10px;  
250 }  
251 .cell .cell_stderr {  
252   margin-top: 10px;  
253 }  
254 .cell .cell_stderr {  
255   margin-top: 10px;  
256 }  
257 .cell .cell_stderr {  
258   margin-top: 10px;  
259 }  
260 .cell .cell_stderr {  
261   margin-top: 10px;  
262 }  
263 .cell .cell_stderr {  
264   margin-top: 10px;  
265 }  
266 .cell .cell_stderr {  
267   margin-top: 10px;  
268 }  
269 .cell .cell_stderr {  
270   margin-top: 10px;  
271 }  
272 .cell .cell_stderr {  
273   margin-top: 10px;  
274 }  
275 .cell .cell_stderr {  
276   margin-top: 10px;  
277 }  
278 .cell .cell_stderr {  
279   margin-top: 10px;  
280 }  
281 .cell .cell_stderr {  
282   margin-top: 10px;  
283 }  
284 .cell .cell_stderr {  
285   margin-top: 10px;  
286 }  
287 .cell .cell_stderr {  
288   margin-top: 10px;  
289 }  
290 .cell .cell_stderr {  
291   margin-top: 10px;  
292 }  
293 .cell .cell_stderr {  
294   margin-top: 10px;  
295 }  
296 .cell .cell_stderr {  
297   margin-top: 10px;  
298 }  
299 .cell .cell_stderr {  
300   margin-top: 10px;  
301 }  
302 .cell .cell_stderr {  
303   margin-top: 10px;  
304 }  
305 .cell .cell_stderr {  
306   margin-top: 10px;  
307 }  
308 .cell .cell_stderr {  
309   margin-top: 10px;  
310 }  
311 .cell .cell_stderr {  
312   margin-top: 10px;  
313 }  
314 .cell .cell_stderr {  
315   margin-top: 10px;  
316 }  
317 .cell .cell_stderr {  
318   margin-top: 10px;  
319 }  
320 .cell .cell_stderr {  
321   margin-top: 10px;  
322 }  
323 .cell .cell_stderr {  
324   margin-top: 10px;  
325 }  
326 .cell .cell_stderr {  
327   margin-top: 10px;  
328 }  
329 .cell .cell_stderr {  
330   margin-top: 10px;  
331 }  
332 .cell .cell_stderr {  
333   margin-top: 10px;  
334 }  
335 .cell .cell_stderr {  
336   margin-top: 10px;  
337 }  
338 .cell .cell_stderr {  
339   margin-top: 10px;  
340 }  
341 .cell .cell_stderr {  
342   margin-top: 10px;  
343 }  
344 .cell .cell_stderr {  
345   margin-top: 10px;  
346 }  
347 .cell .cell_stderr {  
348   margin-top: 10px;  
349 }  
350 .cell .cell_stderr {  
351   margin-top: 10px;  
352 }  
353 .cell .cell_stderr {  
354   margin-top: 10px;  
355 }  
356 .cell .cell_stderr {  
357   margin-top: 10px;  
358 }  
359 .cell .cell_stderr {  
360   margin-top: 10px;  
361 }  
362 .cell .cell_stderr {  
363   margin-top: 10px;  
364 }  
365 .cell .cell_stderr {  
366   margin-top: 10px;  
367 }  
368 .cell .cell_stderr {  
369   margin-top: 10px;  
370 }  
371 .cell .cell_stderr {  
372   margin-top: 10px;  
373 }  
374 .cell .cell_stderr {  
375   margin-top: 10px;  
376 }  
377 .cell .cell_stderr {  
378   margin-top: 10px;  
379 }  
380 .cell .cell_stderr {  
381   margin-top: 10px;  
382 }  
383 .cell .cell_stderr {  
384   margin-top: 10px;  
385 }  
386 .cell .cell_stderr {  
387   margin-top: 10px;  
388 }  
389 .cell .cell_stderr {  
390   margin-top: 10px;  
391 }  
392 .cell .cell_stderr {  
393   margin-top: 10px;  
394 }  
395 .cell .cell_stderr {  
396   margin-top: 10px;  
397 }  
398 .cell .cell_stderr {  
399   margin-top: 10px;  
400 }  
401 .cell .cell_stderr {  
402   margin-top: 10px;  
403 }  
404 .cell .cell_stderr {  
405   margin-top: 10px;  
406 }  
407 .cell .cell_stderr {  
408   margin-top: 10px;  
409 }  
410 .cell .cell_stderr {  
411   margin-top: 10px;  
412 }  
413 .cell .cell_stderr {  
414   margin-top: 10px;  
415 }  
416 .cell .cell_stderr {  
417   margin-top: 10px;  
418 }  
419 .cell .cell_stderr {  
420   margin-top: 10px;  
421 }  
422 .cell .cell_stderr {  
423   margin-top: 10px;  
424 }  
425 .cell .cell_stderr {  
426   margin-top: 10px;  
427 }  
428 .cell .cell_stderr {  
429   margin-top: 10px;  
430 }  
431 .cell .cell_stderr {  
432   margin-top: 10px;  
433 }  
434 .cell .cell_stderr {  
435   margin-top: 10px;  
436 }  
437 .cell .cell_stderr {  
438   margin-top: 10px;  
439 }  
440 .cell .cell_stderr {  
441   margin-top: 10px;  
442 }  
443 .cell .cell_stderr {  
444   margin-top: 10px;  
445 }  
446 .cell .cell_stderr {  
447   margin-top: 10px;  
448 }  
449 .cell .cell_stderr {  
450   margin-top: 10px;  
451 }  
452 .cell .cell_stderr {  
453   margin-top: 10px;  
454 }  
455 .cell .cell_stderr {  
456   margin-top: 10px;  
457 }  
458 .cell .cell_stderr {  
459   margin-top: 10px;  
460 }  
461 .cell .cell_stderr {  
462   margin-top: 10px;  
463 }  
464 .cell .cell_stderr {  
465   margin-top: 10px;  
466 }  
467 .cell .cell_stderr {  
468   margin-top: 10px;  
469 }  
470 .cell .cell_stderr {  
471   margin-top: 10px;  
472 }  
473 .cell .cell_stderr {  
474   margin-top: 10px;  
475 }  
476 .cell .cell_stderr {  
477   margin-top: 10px;  
478 }  
479 .cell .cell_stderr {  
480   margin-top: 10px;  
481 }  
482 .cell .cell_stderr {  
483   margin-top: 10px;  
484 }  
485 .cell .cell_stderr {  
486   margin-top: 10px;  
487 }  
488 .cell .cell_stderr {  
489   margin-top: 10px;  
490 }  
491 .cell .cell_stderr {  
492   margin-top: 10px;  
493 }  
494 .cell .cell_stderr {  
495   margin-top: 10px;  
496 }  
497 .cell .cell_stderr {  
498   margin-top: 10px;  
499 }  
500 .cell .cell_stderr {  
501   margin-top: 10px;  
502 }  
503 .cell .cell_stderr {  
504   margin-top: 10px;  
505 }  
506 .cell .cell_stderr {  
507   margin-top: 10px;  
508 }  
509 .cell .cell_stderr {  
510   margin-top: 10px;  
511 }  
512 .cell .cell_stderr {  
513   margin-top: 10px;  
514 }  
515 .cell .cell_stderr {  
516   margin-top: 10px;  
517 }  
518 .cell .cell_stderr {  
519   margin-top: 10px;  
520 }  
521 .cell .cell_stderr {  
522   margin-top: 10px;  
523 }  
524 .cell .cell_stderr {  
525   margin-top: 10px;  
526 }  
527 .cell .cell_stderr {  
528   margin-top: 10px;  
529 }  
530 .cell .cell_stderr {  
531   margin-top: 10px;  
532 }  
533 .cell .cell_stderr {  
534   margin-top: 10px;  
535 }  
536 .cell .cell_stderr {  
537   margin-top: 10px;  
538 }  
539 .cell .cell_stderr {  
540   margin-top: 10px;  
541 }  
542 .cell .cell_stderr {  
543   margin-top: 10px;  
544 }  
545 .cell .cell_stderr {  
546   margin-top: 10px;  
547 }  
548 .cell .cell_stderr {  
549   margin-top: 10px;  
550 }  
551 .cell .cell_stderr {  
552   margin-top: 10px;  
553 }  
554 .cell .cell_stderr {  
555   margin-top: 10px;  
556 }  
557 .cell .cell_stderr {  
558   margin-top: 10px;  
559 }  
560 .cell .cell_stderr {  
561   margin-top: 10px;  
562 }  
563 .cell .cell_stderr {  
564   margin-top: 10px;  
565 }  
566 .cell .cell_stderr {  
567   margin-top: 10px;  
568 }  
569 .cell .cell_stderr {  
570   margin-top: 10px;  
571 }  
572 .cell .cell_stderr {  
573   margin-top: 10px;  
574 }  
575 .cell .cell_stderr {  
576   margin-top: 10px;  
577 }  
578 .cell .cell_stderr {  
579   margin-top: 10px;  
580 }  
581 .cell .cell_stderr {  
582   margin-top: 10px;  
583 }  
584 .cell .cell_stderr {  
585   margin-top: 10px;  
586 }  
587 .cell .cell_stderr {  
588   margin-top: 10px;  
589 }  
590 .cell .cell_stderr {  
591   margin-top: 10px;  
592 }  
593 .cell .cell_stderr {  
594   margin-top: 10px;  
595 }  
596 .cell .cell_stderr {  
597   margin-top: 10px;  
598 }  
599 .cell .cell_stderr {  
600   margin-top: 10px;  
601 }  
602 .cell .cell_stderr {  
603   margin-top: 10px;  
604 }  
605 .cell .cell_stderr {  
606   margin-top: 10px;  
607 }  
608 .cell .cell_stderr {  
609   margin-top: 10px;  
610 }  
611 .cell .cell_stderr {  
612   margin-top: 10px;  
613 }  
614 .cell .cell_stderr {  
615   margin-top: 10px;  
616 }  
617 .cell .cell_stderr {  
618   margin-top: 10px;  
619 }  
620 .cell .cell_stderr {  
621   margin-top: 10px;  
622 }  
623 .cell .cell_stderr {  
624   margin-top: 10px;  
625 }  
626 .cell .cell_stderr {  
627   margin-top: 10px;  
628 }  
629 .cell .cell_stderr {  
630   margin-top: 10px;  
631 }  
632 .cell .cell_stderr {  
633   margin-top: 10px;  
634 }  
635 .cell .cell_stderr {  
636   margin-top: 10px;  
637 }  
638 .cell .cell_stderr {  
639   margin-top: 10px;  
640 }  
641 .cell .cell_stderr {  
642   margin-top: 10px;  
643 }  
644 .cell .cell_stderr
```

# Jupyter Notebook

## image processing



### ch4\_library.ipynb



# 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           py310h4ded8f06_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
```

	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

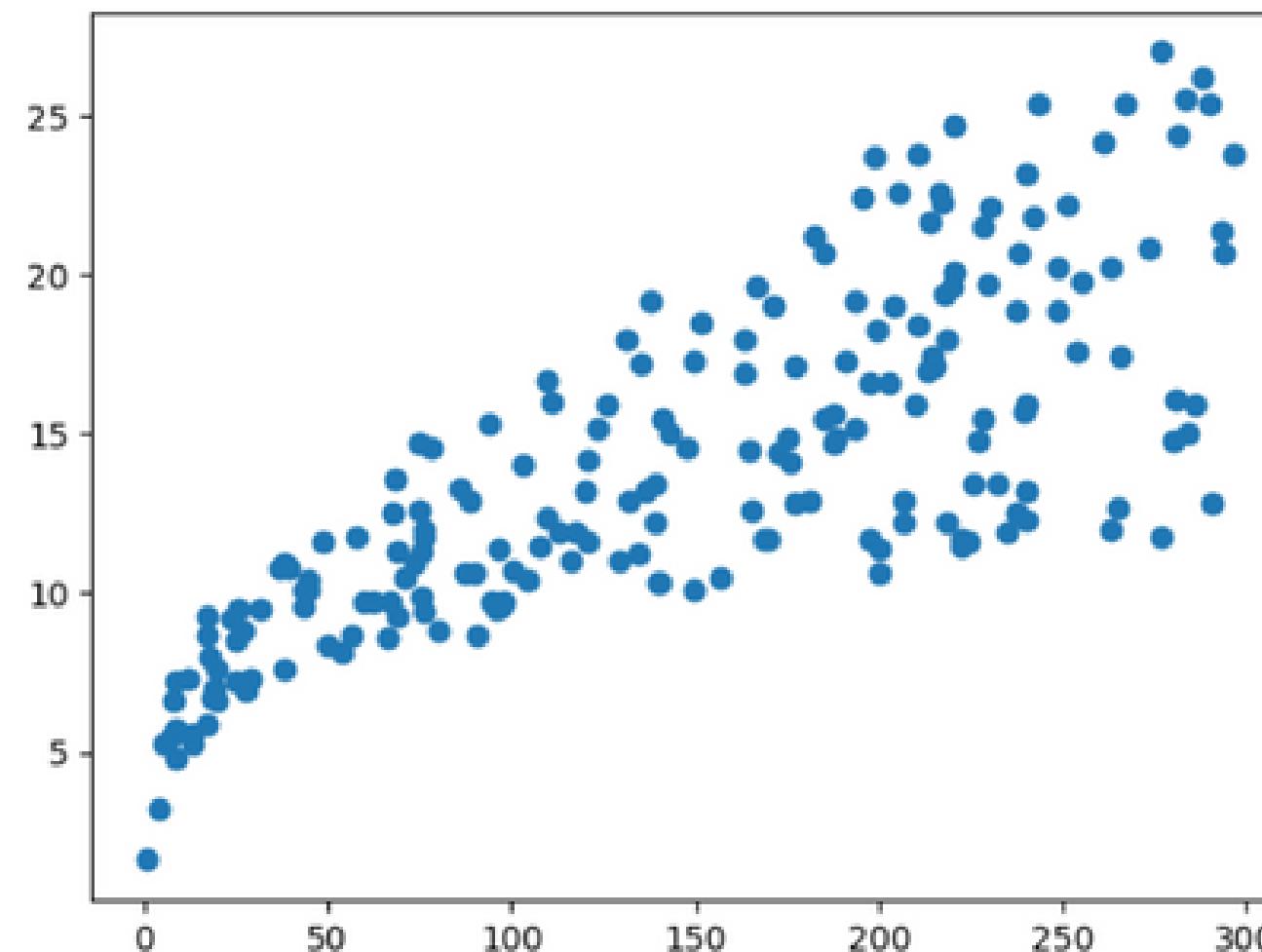
# Jupyter Notebook

## 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]]))
```

# Jupyter Notebook

## image processing

Random and unique

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

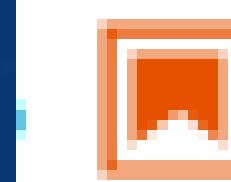
[38]: array([0.85592474, 0.00815198, 0.04118109, 0.78303578])

[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

ใช้ร่วมกับ 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
```

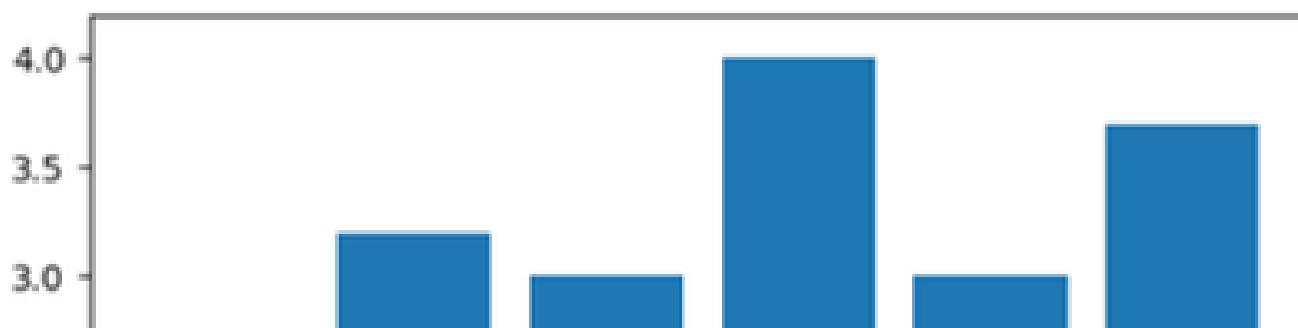
```
[5]: import matplotlib.pyplot as plt
```

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

#### Bar

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

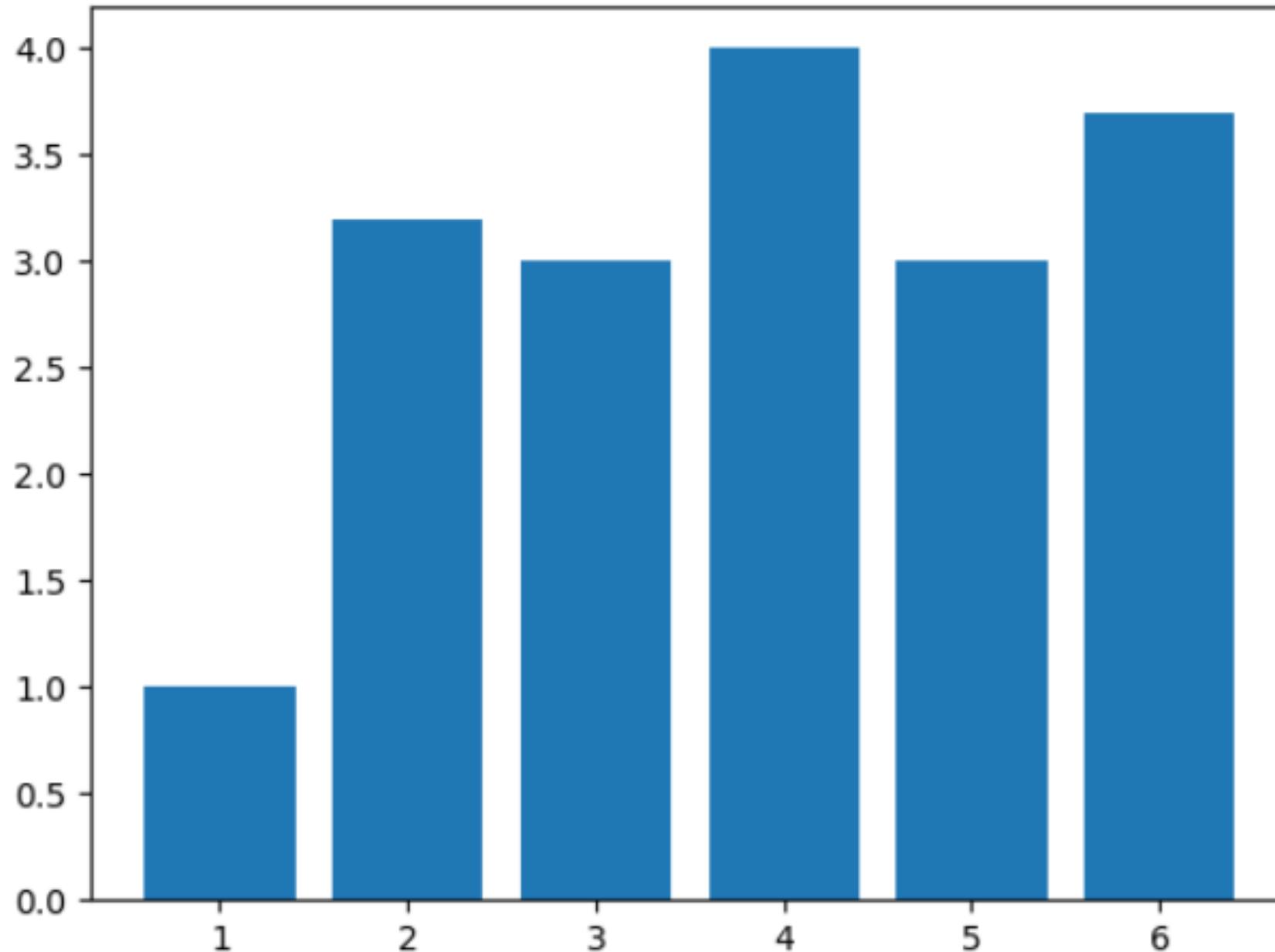
```
[7]: plt.bar(x, y_data)  
plt.show() # ผลลัพธ์จะเห็น x หรือ y บนแก้ว
```



# Jupyter Notebook

## image processing

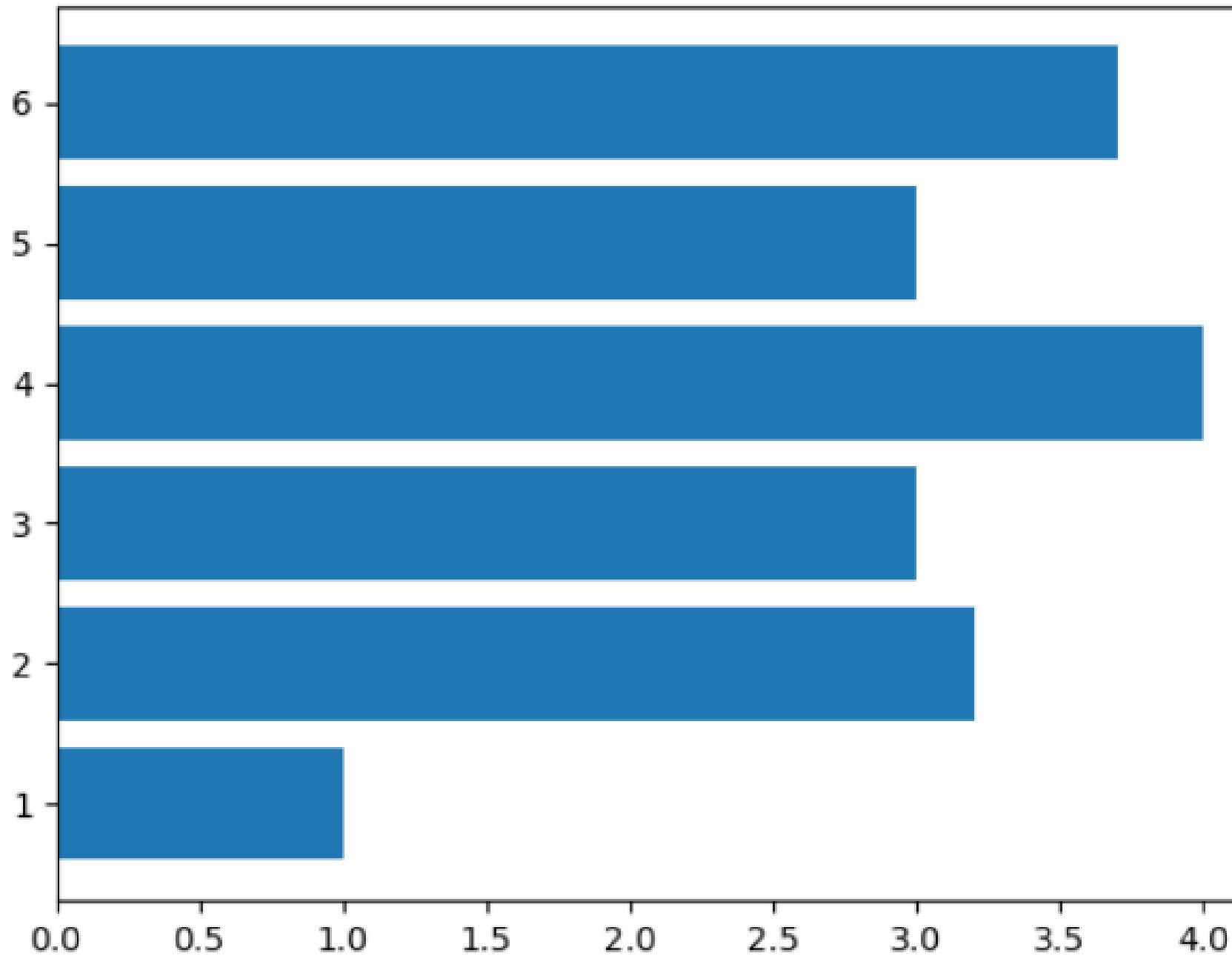
```
[7]: plt.bar(x, y_data)
plt.show() # พล็อตกราฟแกน x แกน y แนวตั้ง
```



# Jupyter Notebook

## image processing

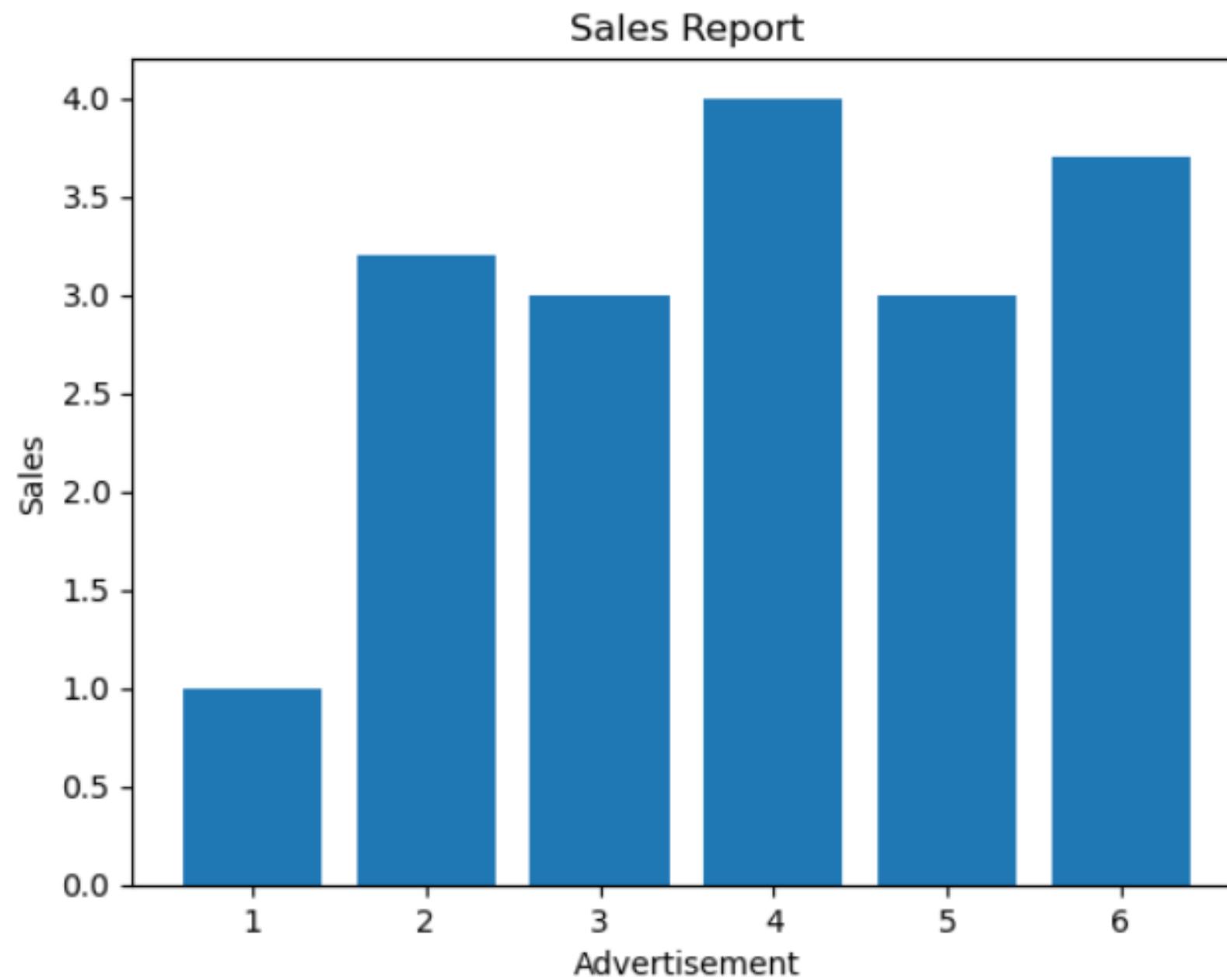
```
[8]: plt.barh(x, y_data) # พล็อตกราฟแท่ง แนวนอน  
plt.show()
```



# Jupyter Notebook

## image processing

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

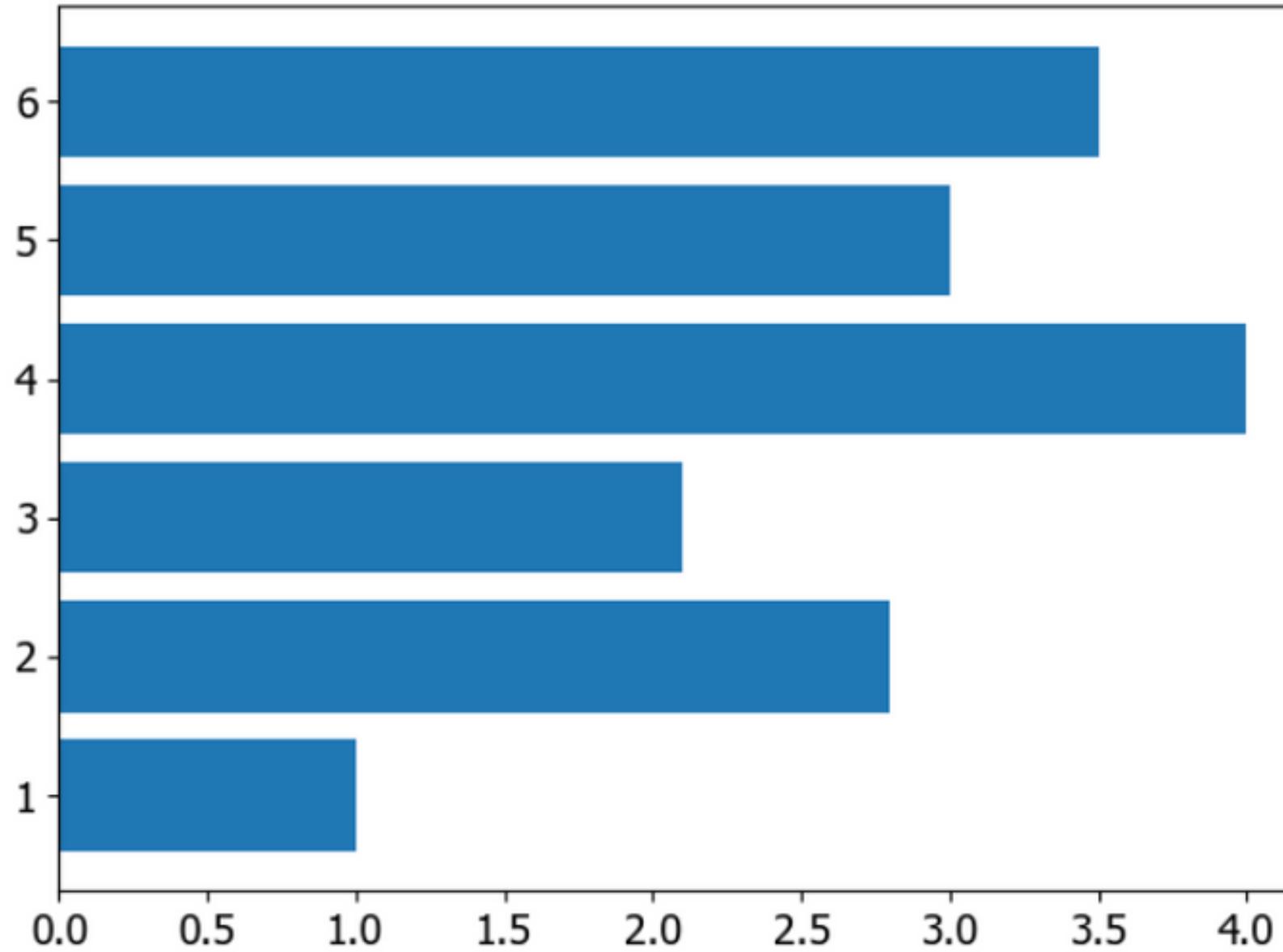


# Jupyter Notebook

## image processing

save

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

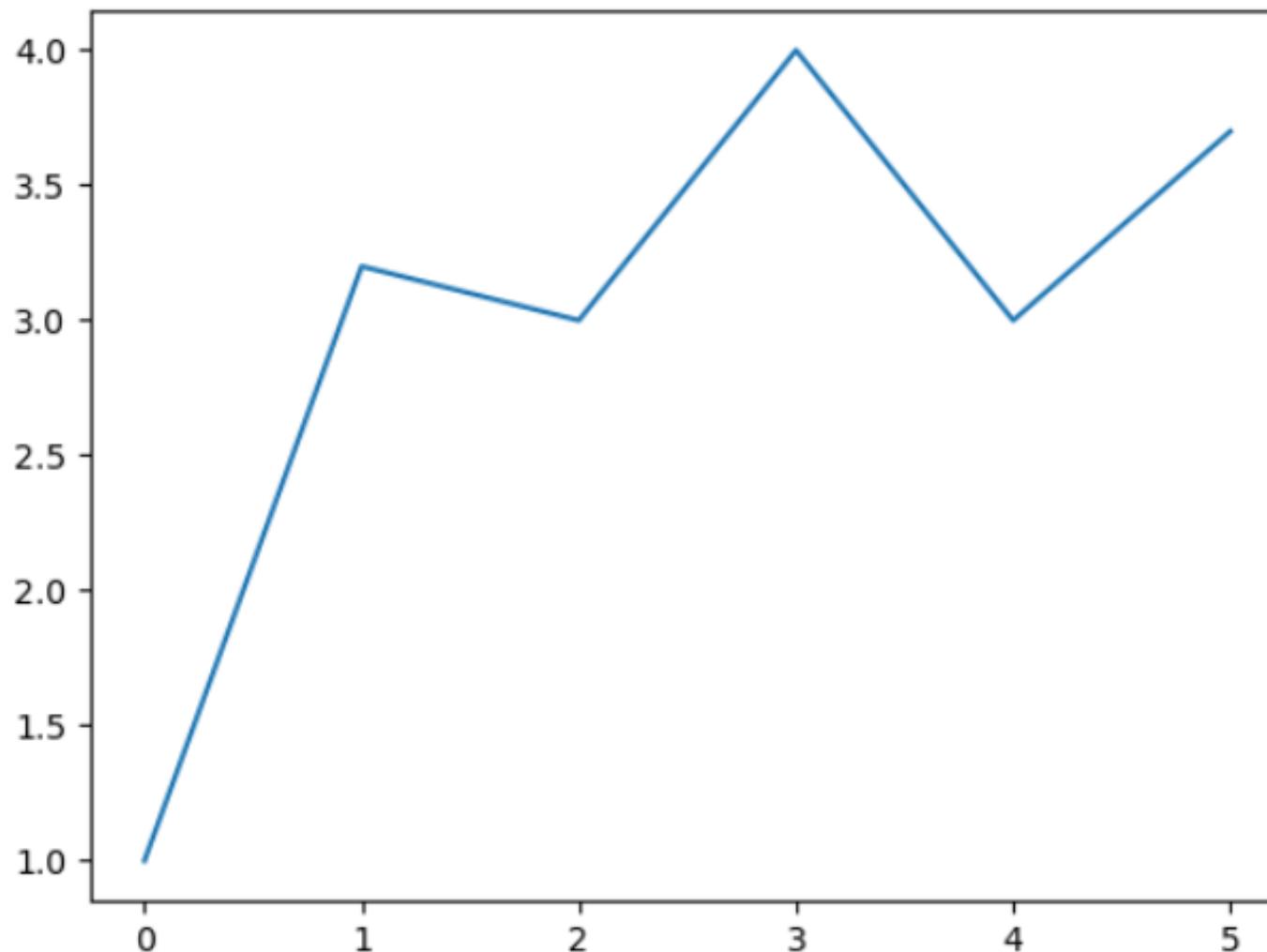


# Jupyter Notebook

## image processing

Line

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

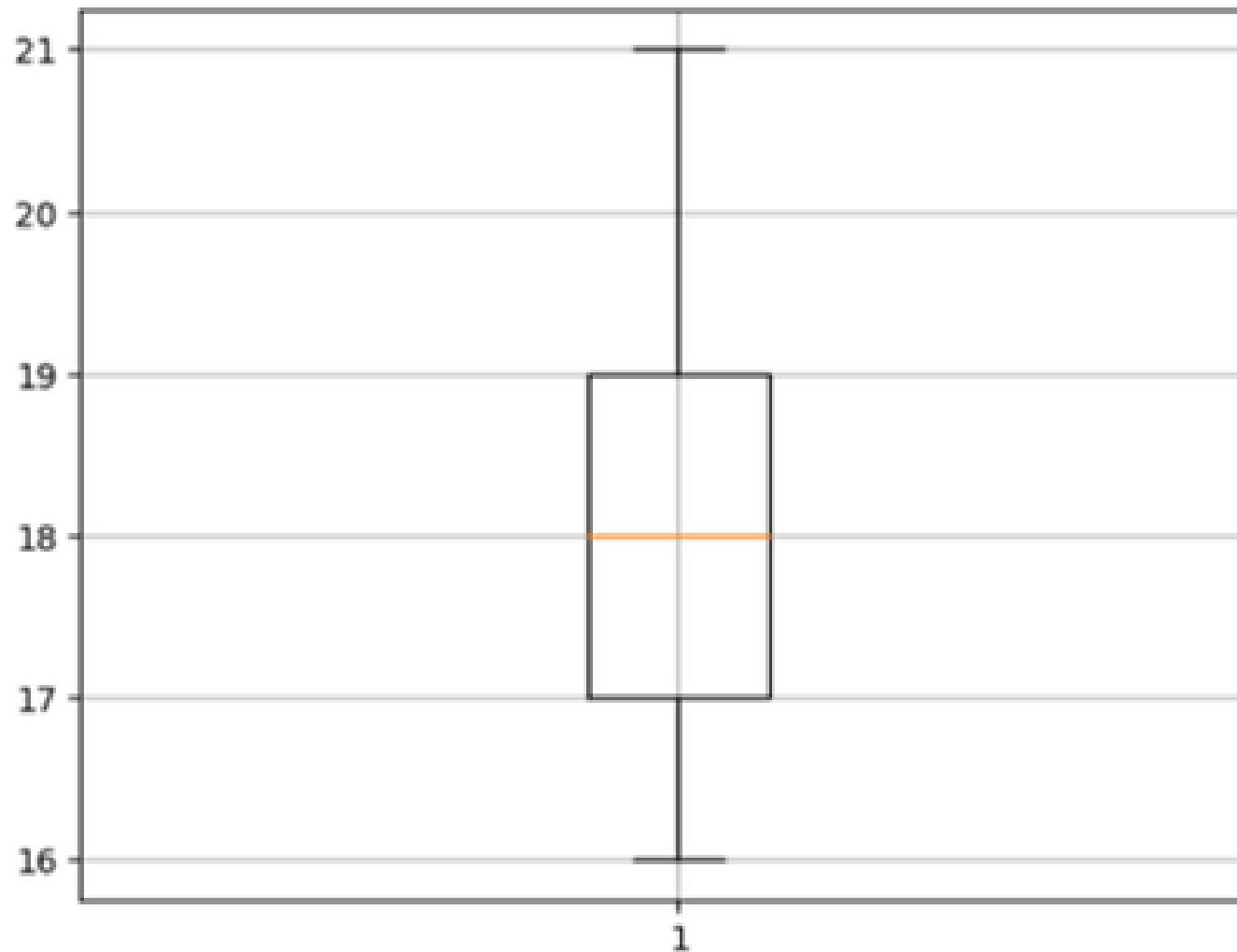


# Jupyter Notebook

## image processing

Box plot

```
[15]: age = [16,17,17,20,20,19,19,18,18,17,16,19,19,19,18,17,21]
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]: # เชื่อมรากฐานเมืองต่างๆ (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='Lumpini')
plt.bar([x + 0.2 for x in xs], # ขยับบาร์ไปทางขวา
        ydata2, width=0.5, color='orange', alpha=0.8, label='Suan')

plt.plot(xs, ydata, color='b', marker='s', markersize=8) #plot ภูมิภาคที่ 1
plt.plot(xs, ydata2, color='r', marker='o', markersize=8)

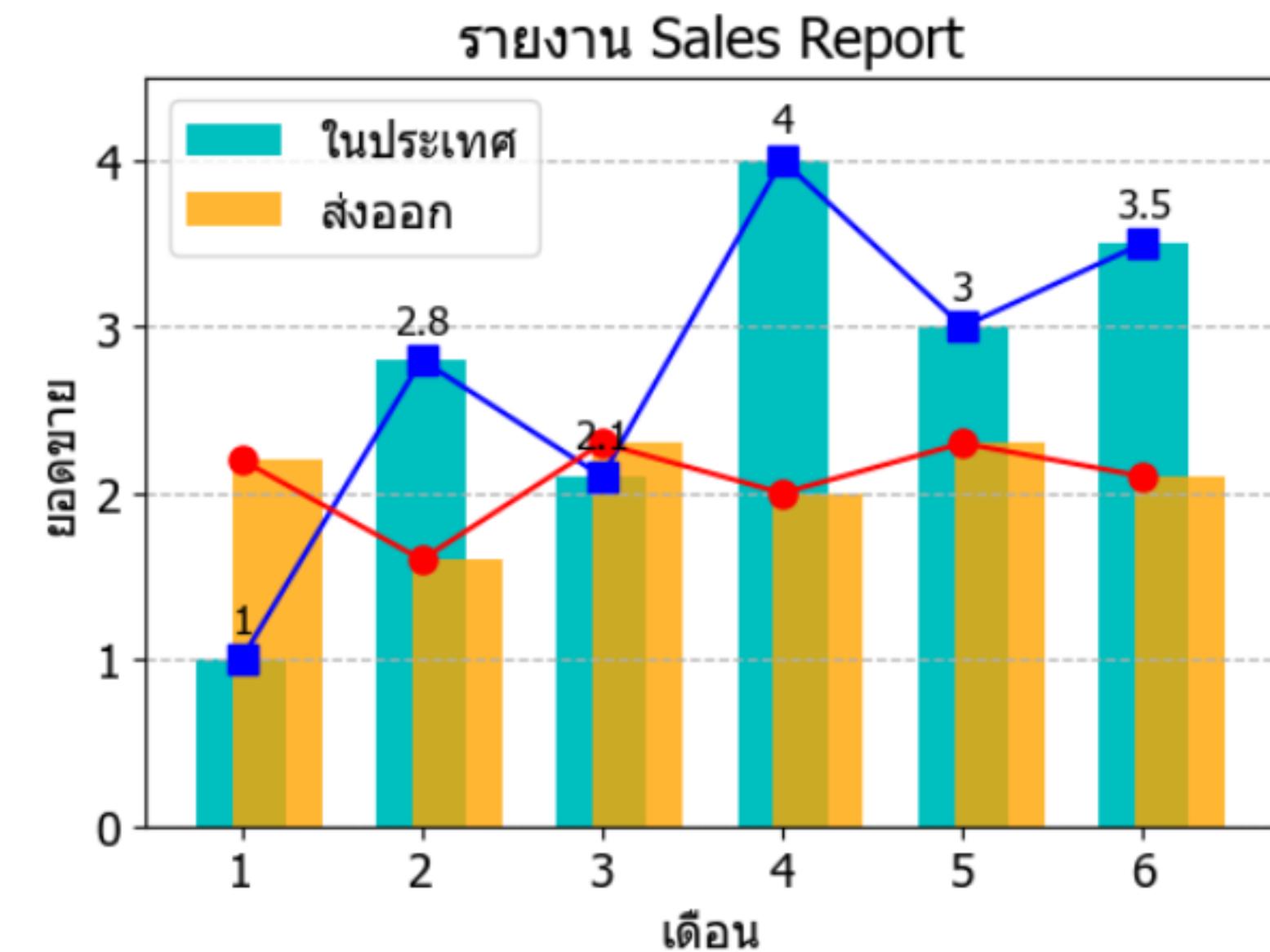
plt.xlabel("พื้นที่")
plt.ylabel("ยอดขาย")
plt.title("รายงาน Sales Report")

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

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

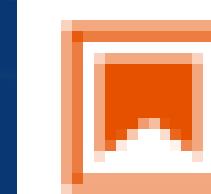
# Jupyter Notebook

## image processing

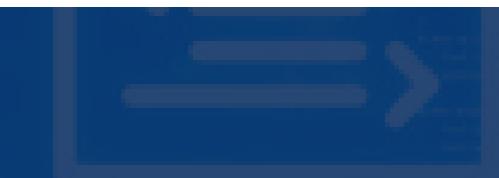


# Jupyter Notebook

## image processing



ch6 seaborn.ipynb



# 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]      # մասկություն x (ծառչացի լուսապատճեմություն)  
       ydata = [1,2.8,2.1,4,3,3.5]    # մասկություն y (x և y մասմիջանականությունը)  
  
       sns.barplot(xdata, ydata)
```

```
-----  
TypeError                                 Traceback (most recent call last)  
Cell In[40], line 4  
      1 xdata = [1,2,3,4,5,6]      # մասկություն x (ծառչացի լուսապատճեմություն)  
      2 ydata = [1,2.8,2.1,4,3,3.5]    # մասկություն y (x և y մասմիջանականությունը)  
----> 4 sns.barplot(xdata, ydata)
```

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

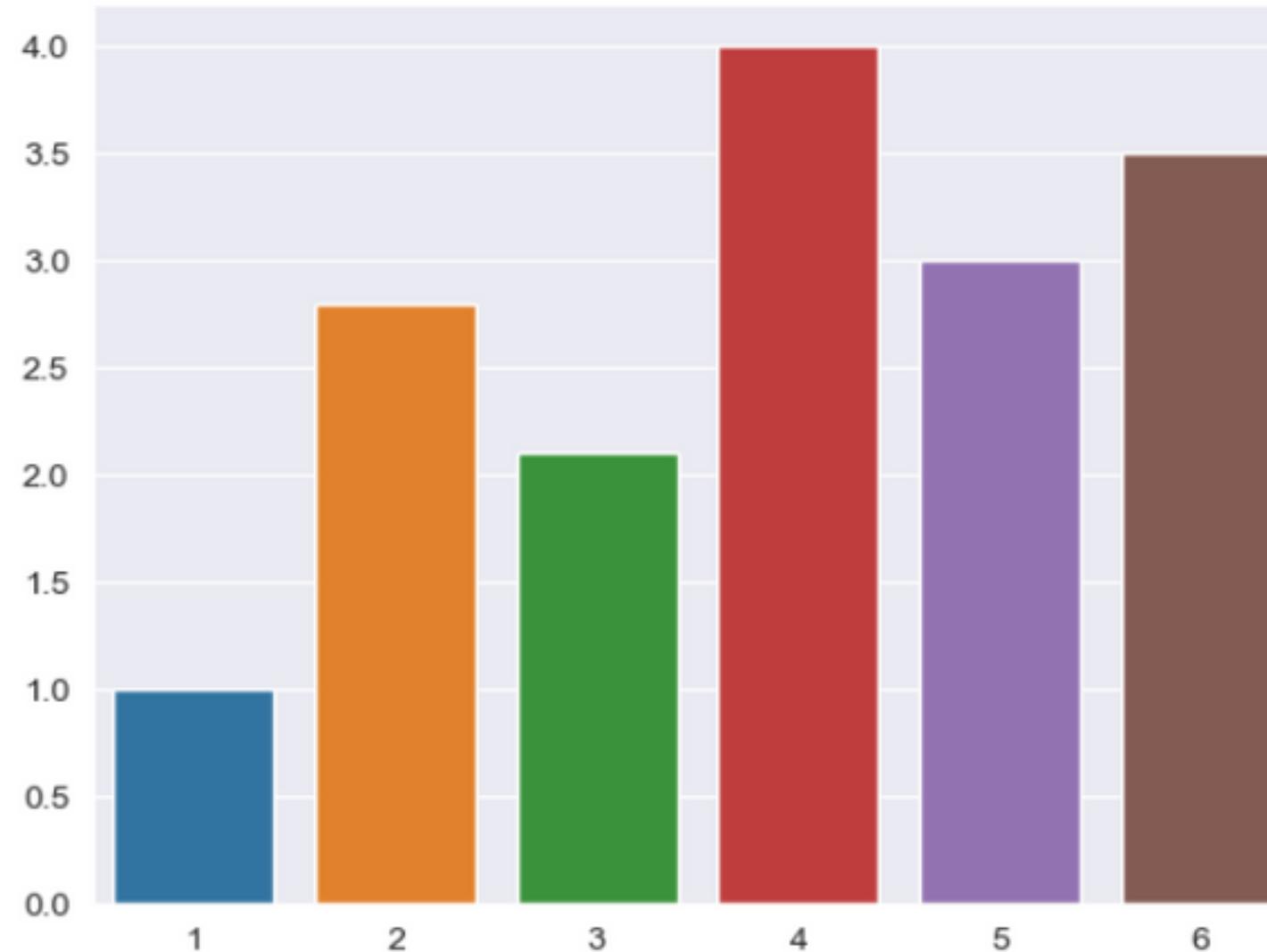
```
[22]: sns.barplot(x=xdata, y=ydata)  
       plt.show()
```



# 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()
```

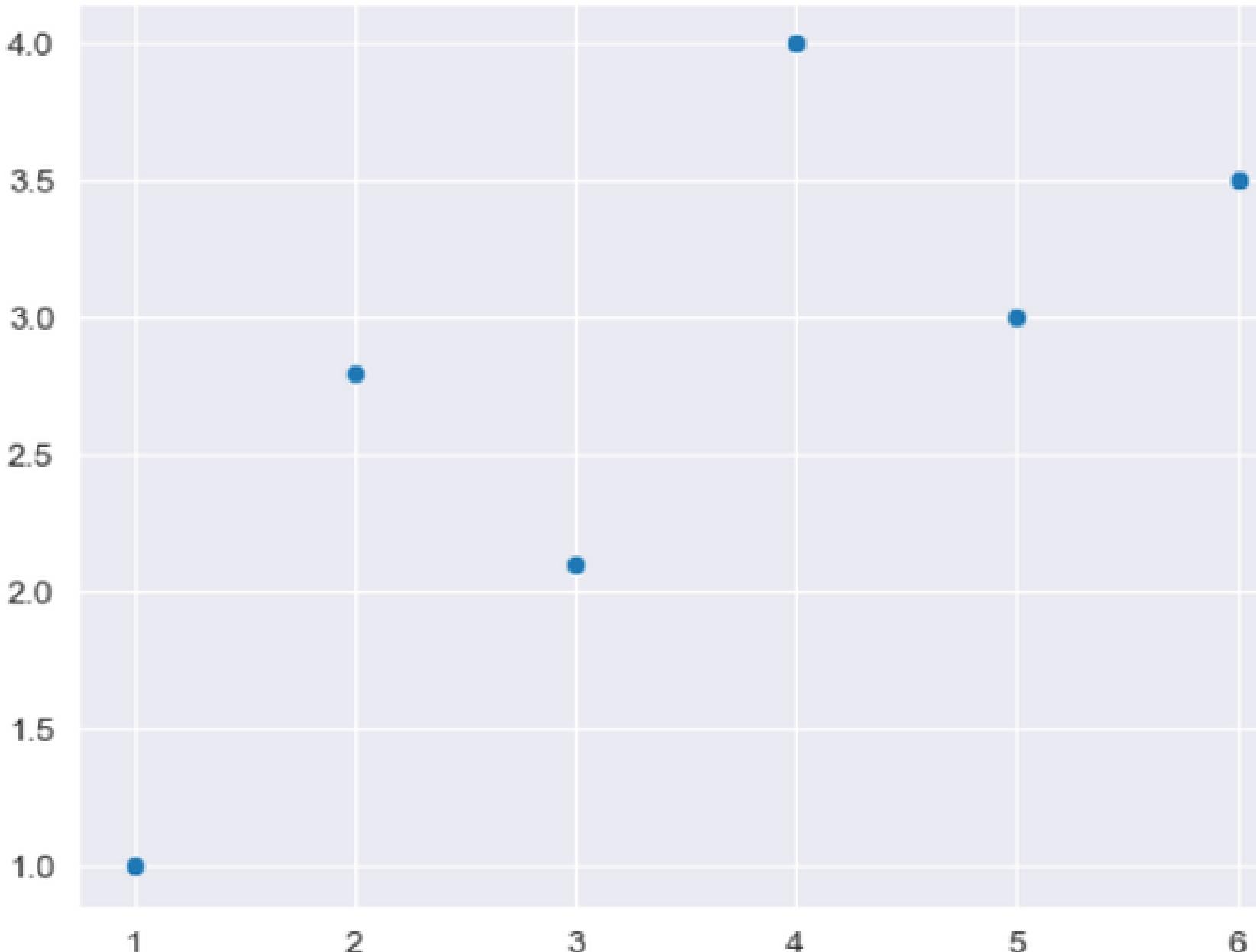


# Jupyter Notebook

## image processing

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

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



# Jupyter Notebook

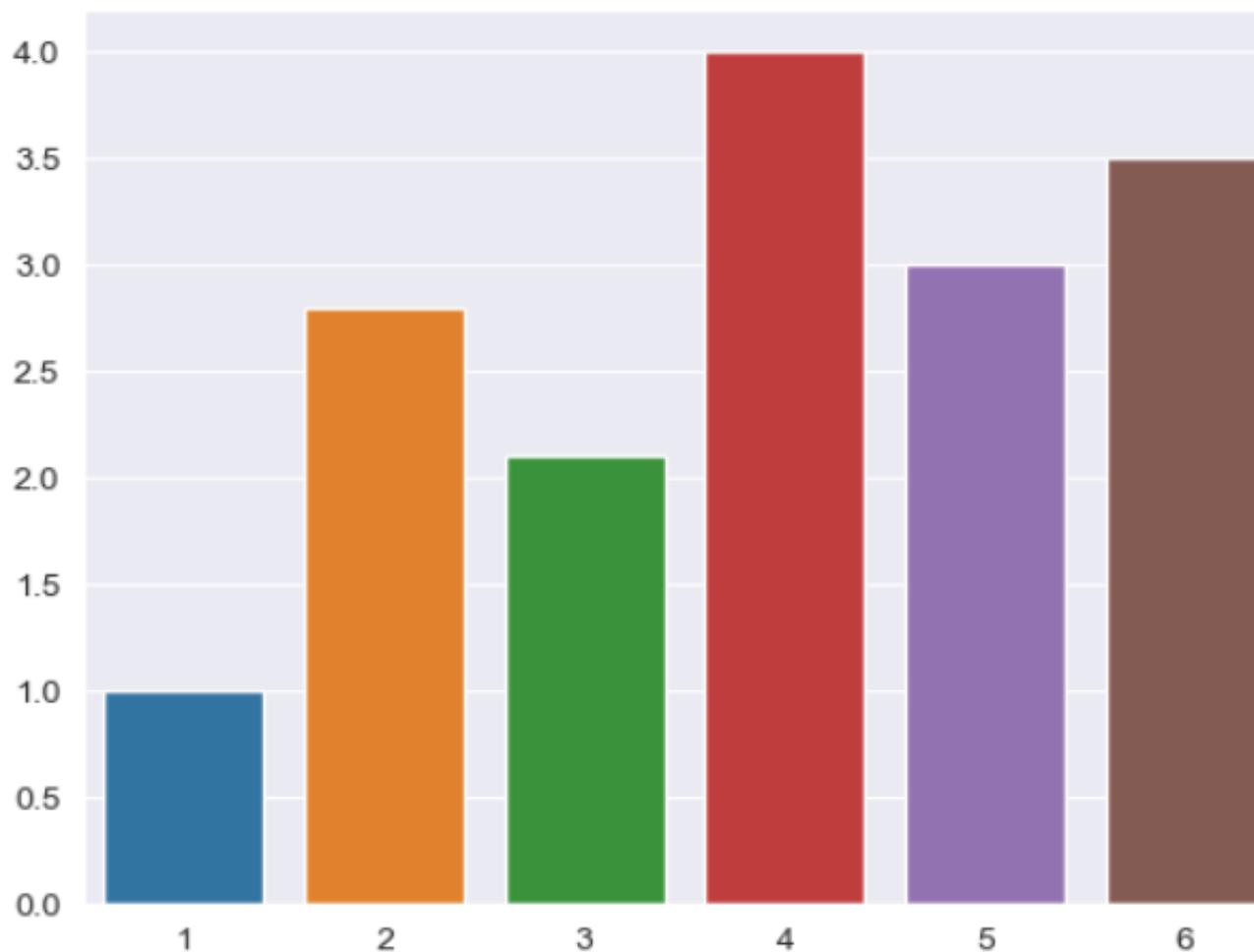
## image processing

### 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: >
```

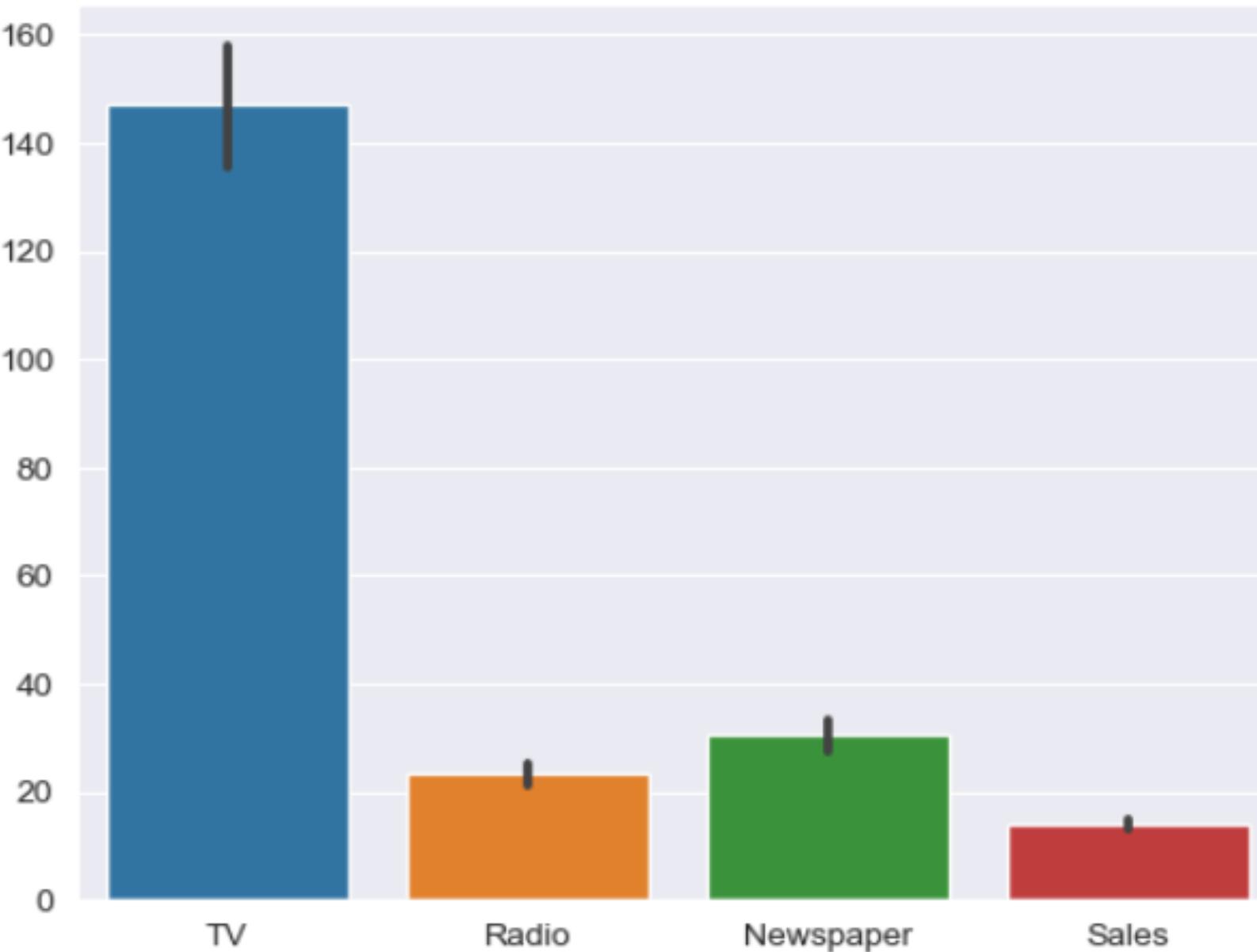


# Jupyter Notebook

## image processing

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

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



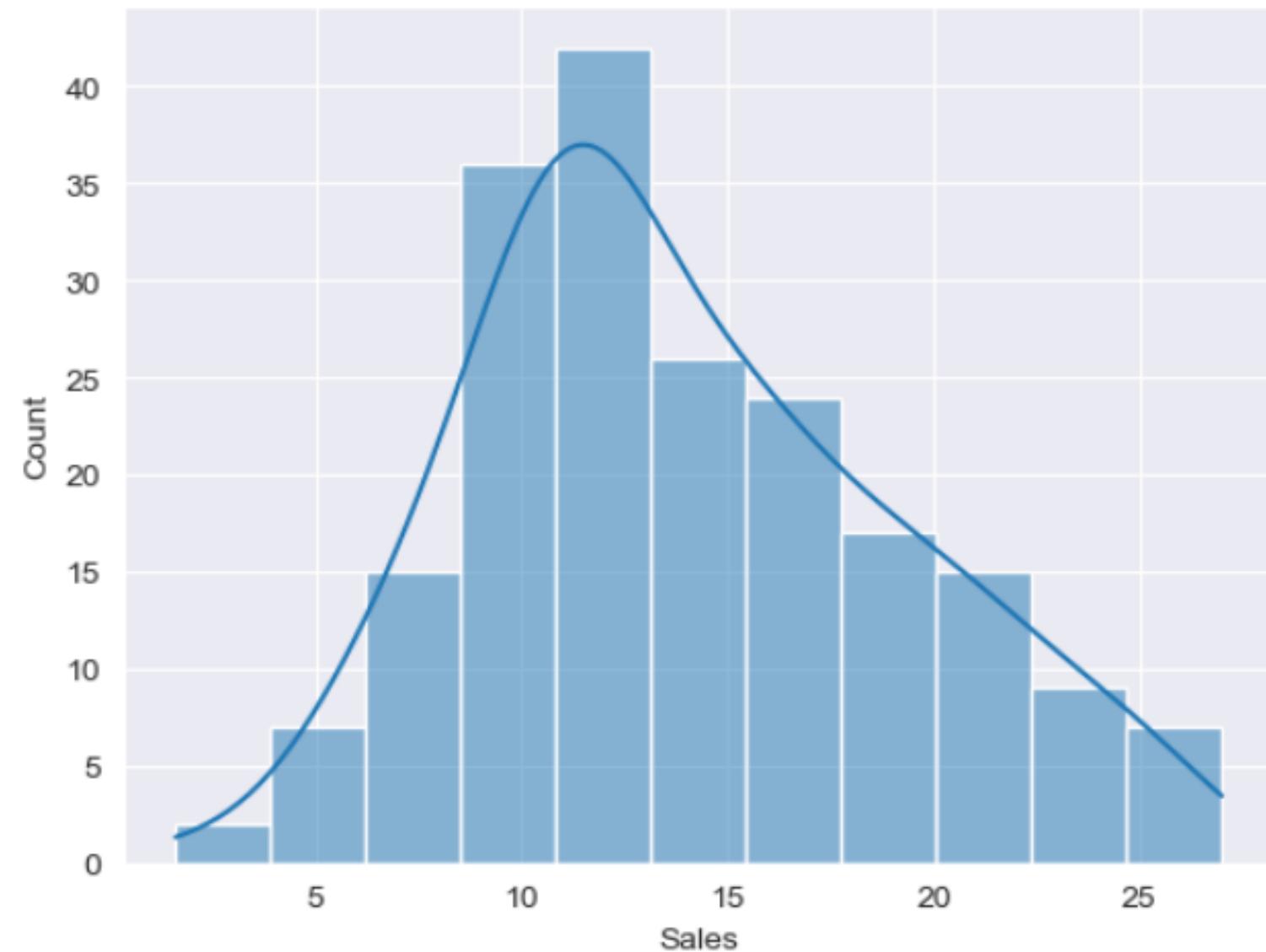
# Jupyter Notebook

## 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)

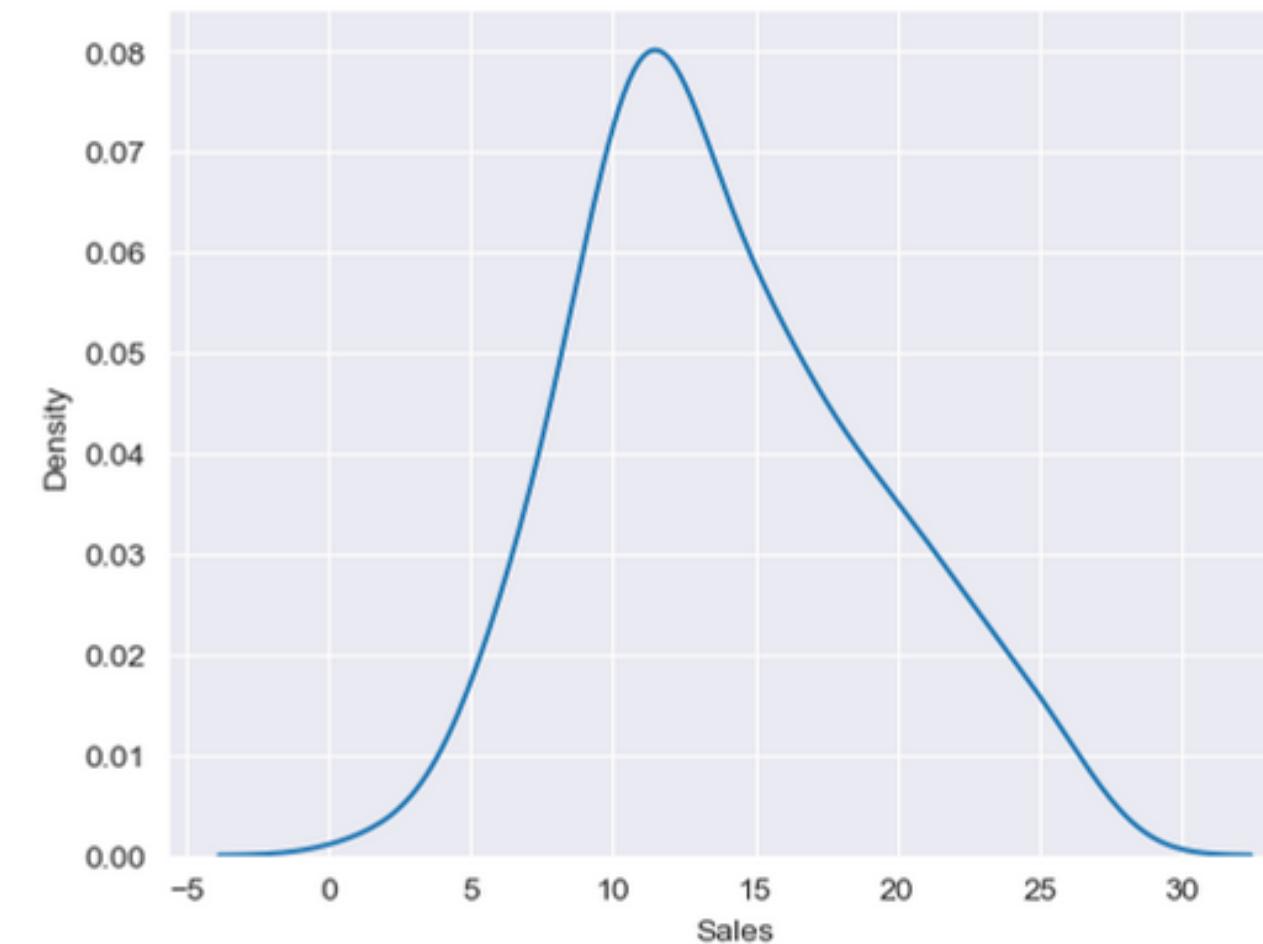
```
[33]: sns.kdeplot(df.Sales, shade=False)
```

```
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.
```

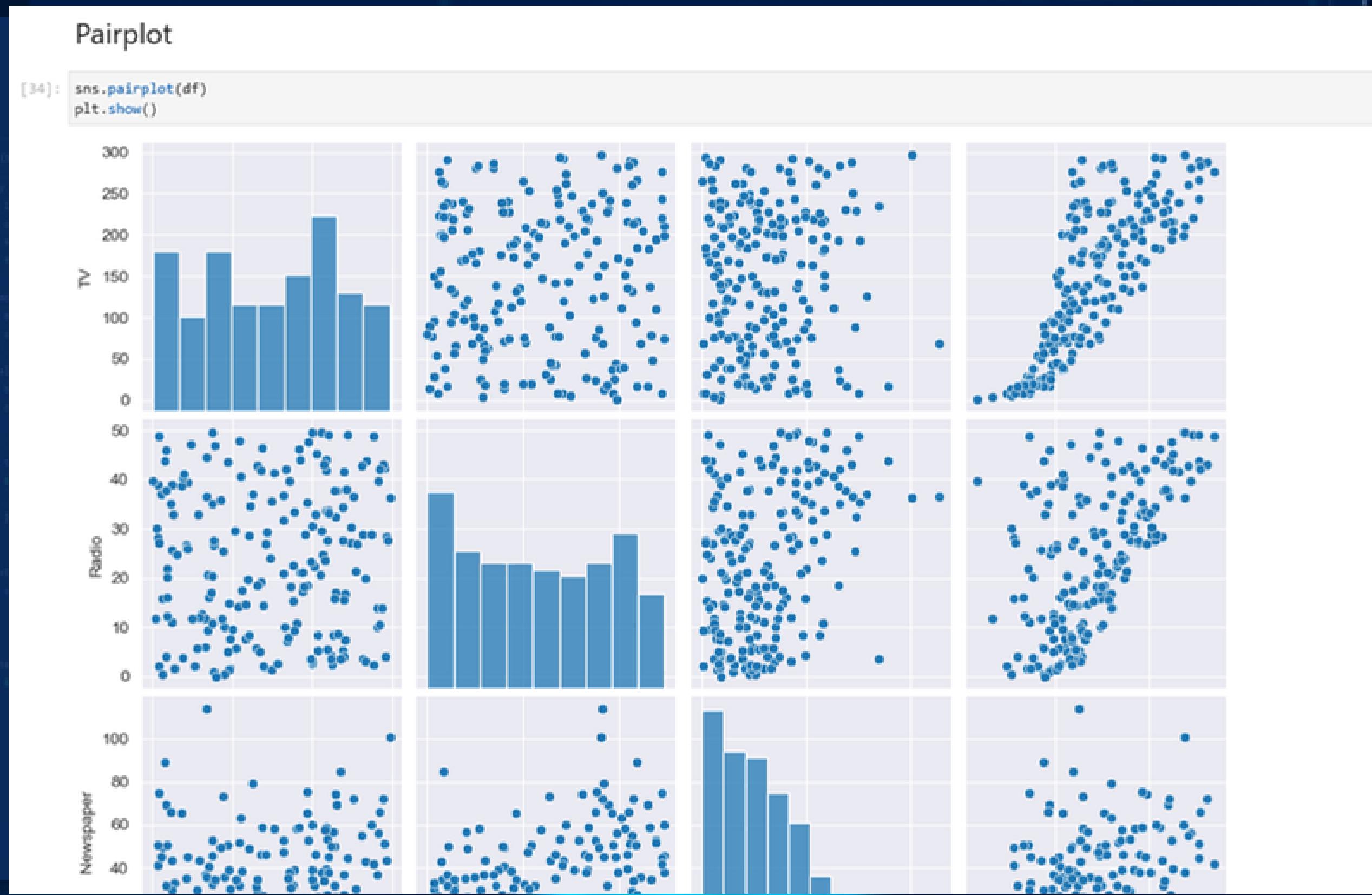
```
sns.kdeplot(df.Sales, shade=False)
```

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



# Jupyter Notebook

## image processing

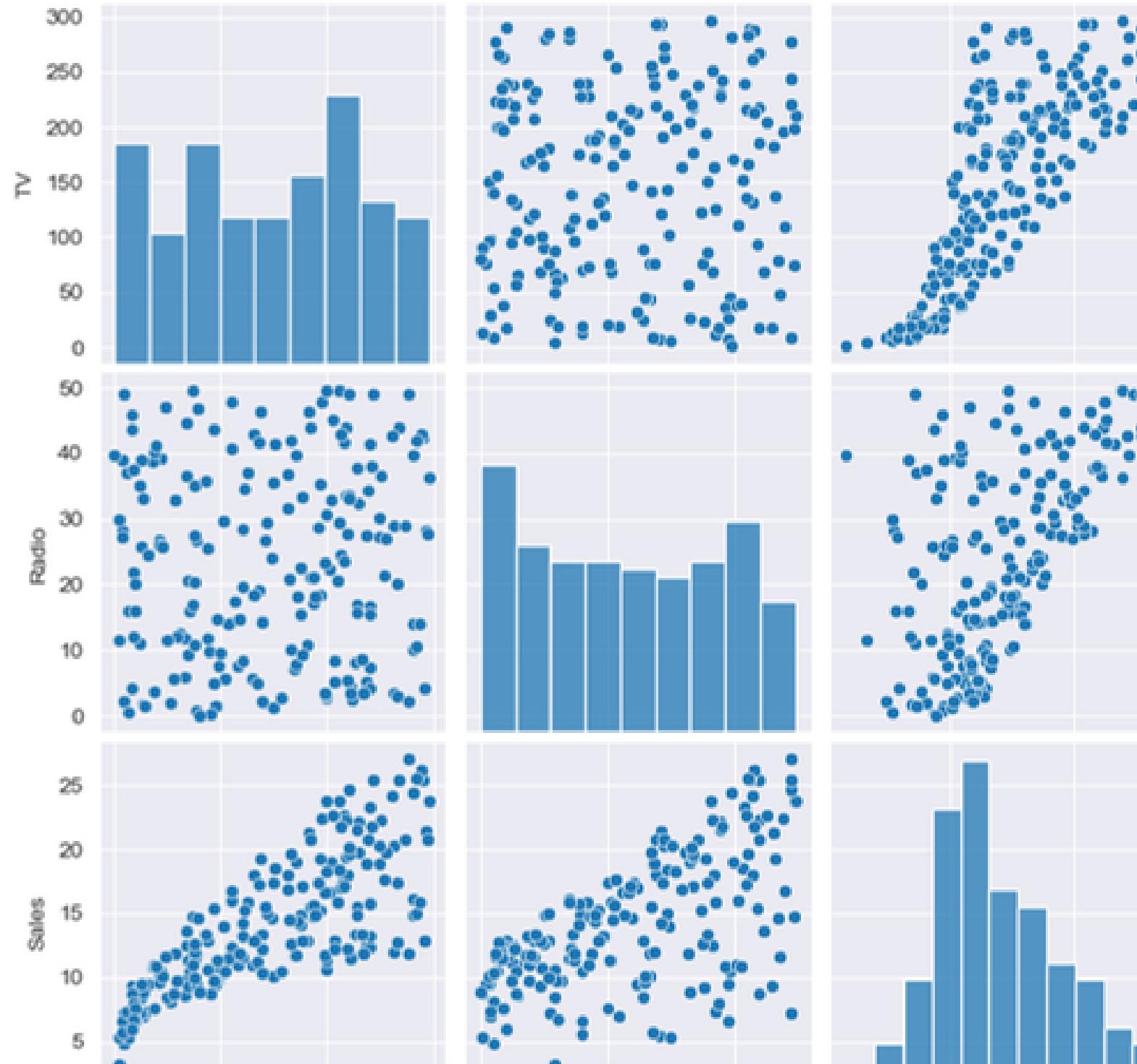


# Jupyter Notebook

## 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()
```

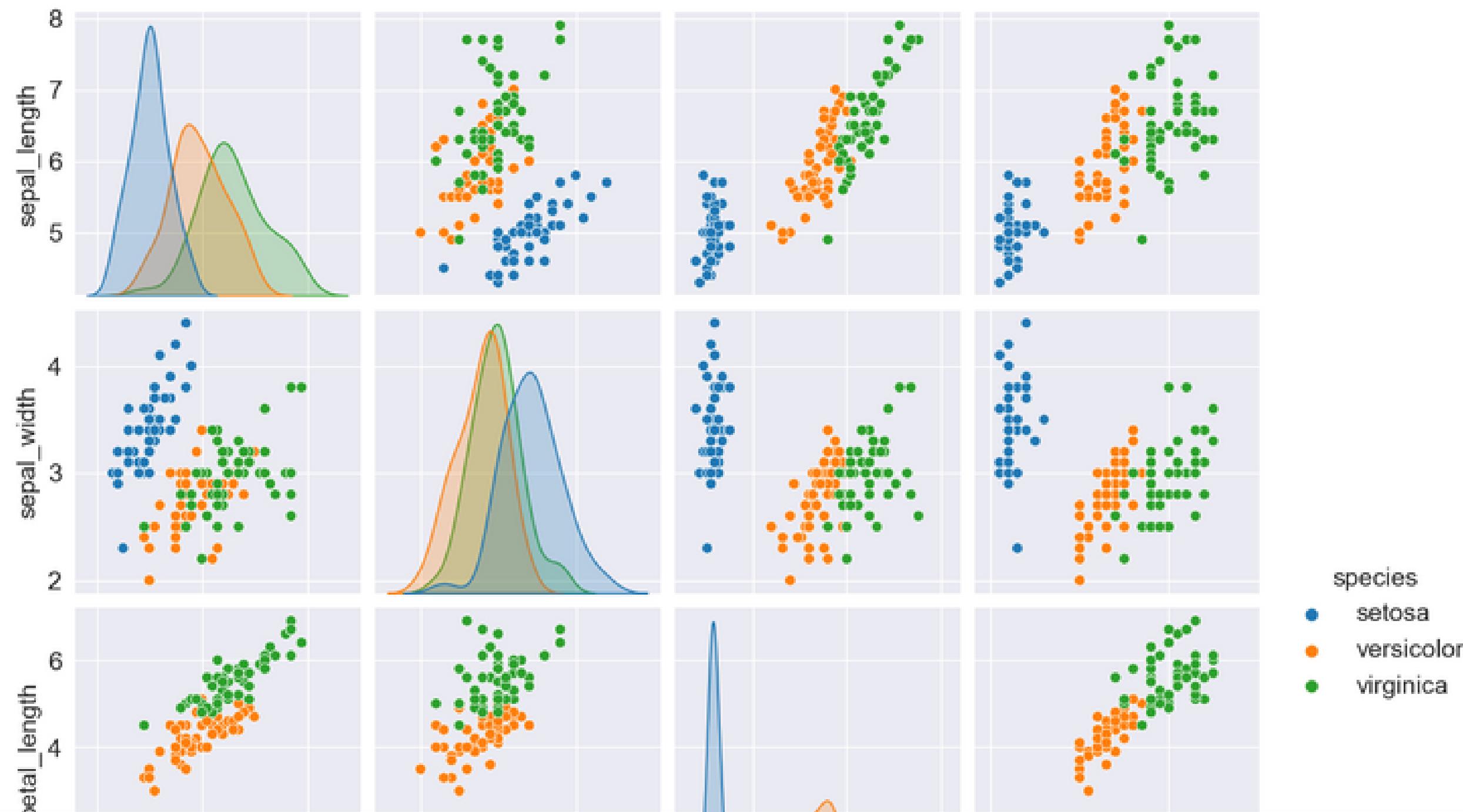
	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

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

# Jupyter Notebook

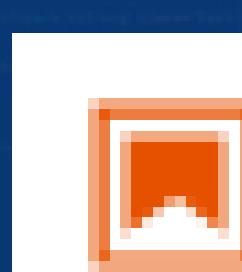
## image processing

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

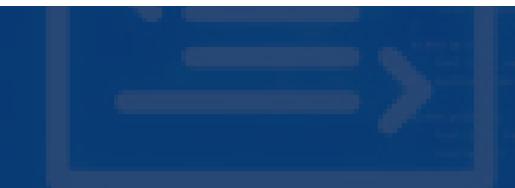


# Jupyter Notebook

## image processing



ch7\_model.ipynb

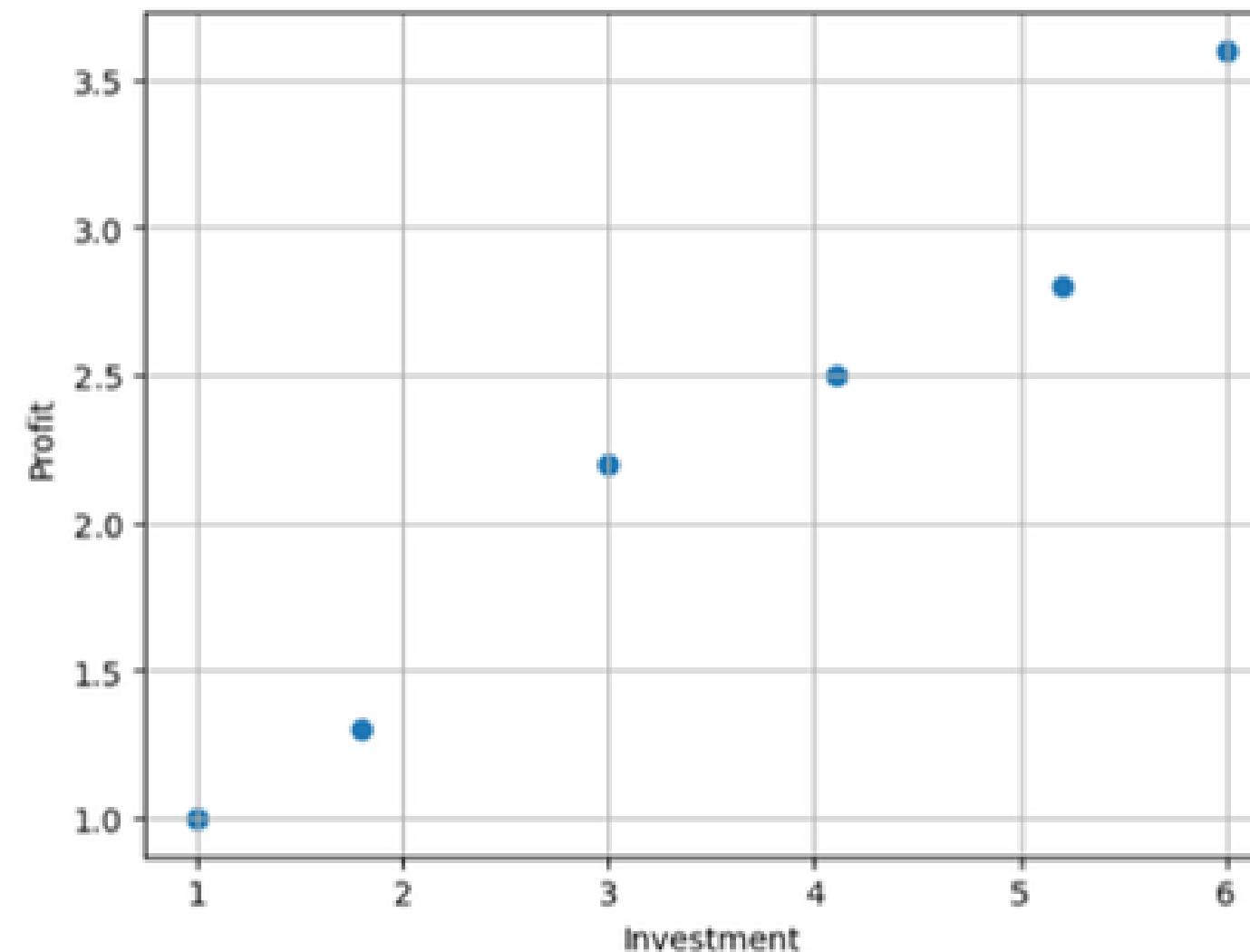


# Jupyter Notebook

## 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()
```



# 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
```

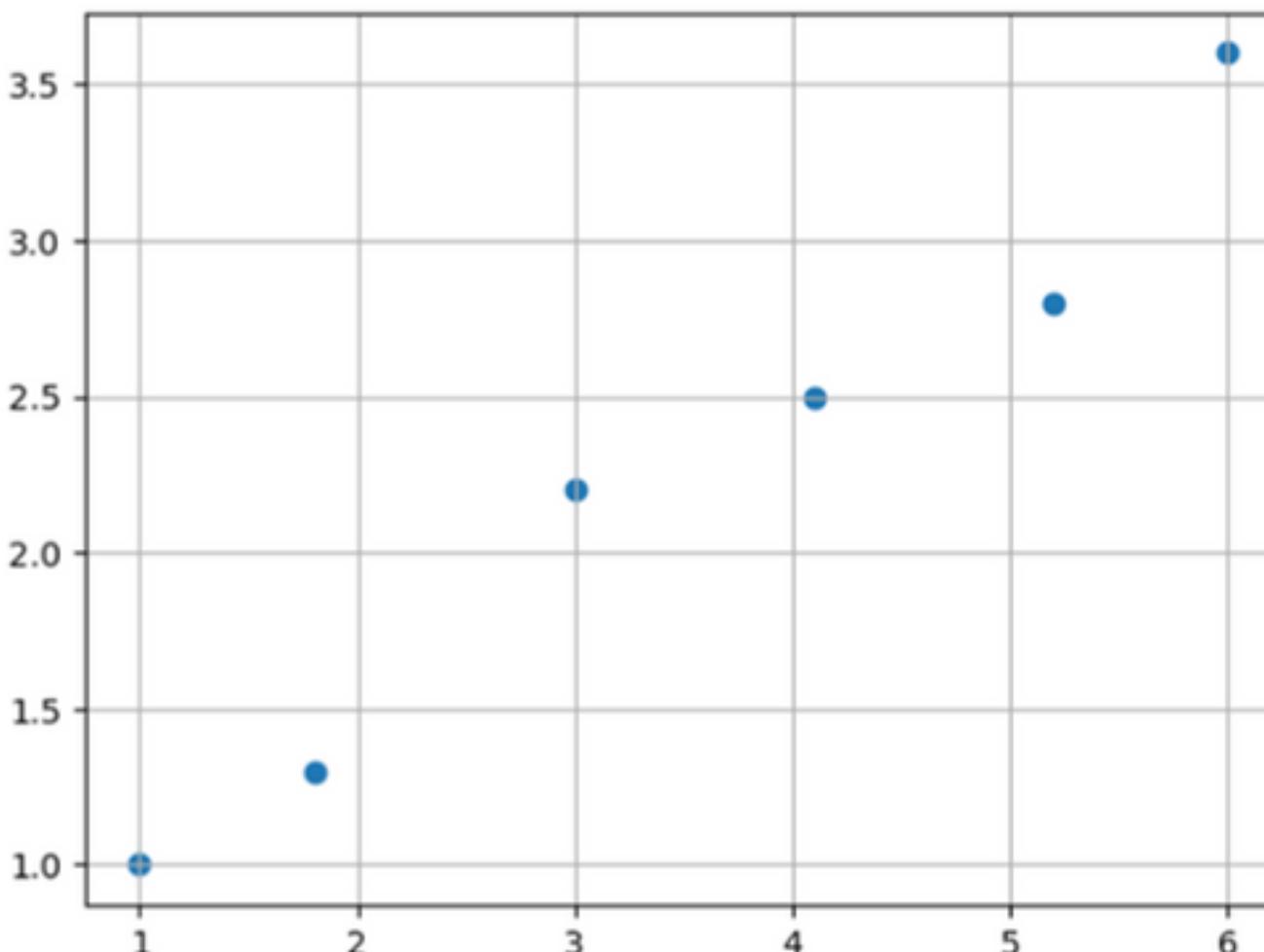
```
[8]: array([1. , 1.3, 2.2, 2.5, 2.8, 3.6])
```

# Jupyter Notebook

## image processing

### Data Visualization

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



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

```
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]] # 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. ]])
```

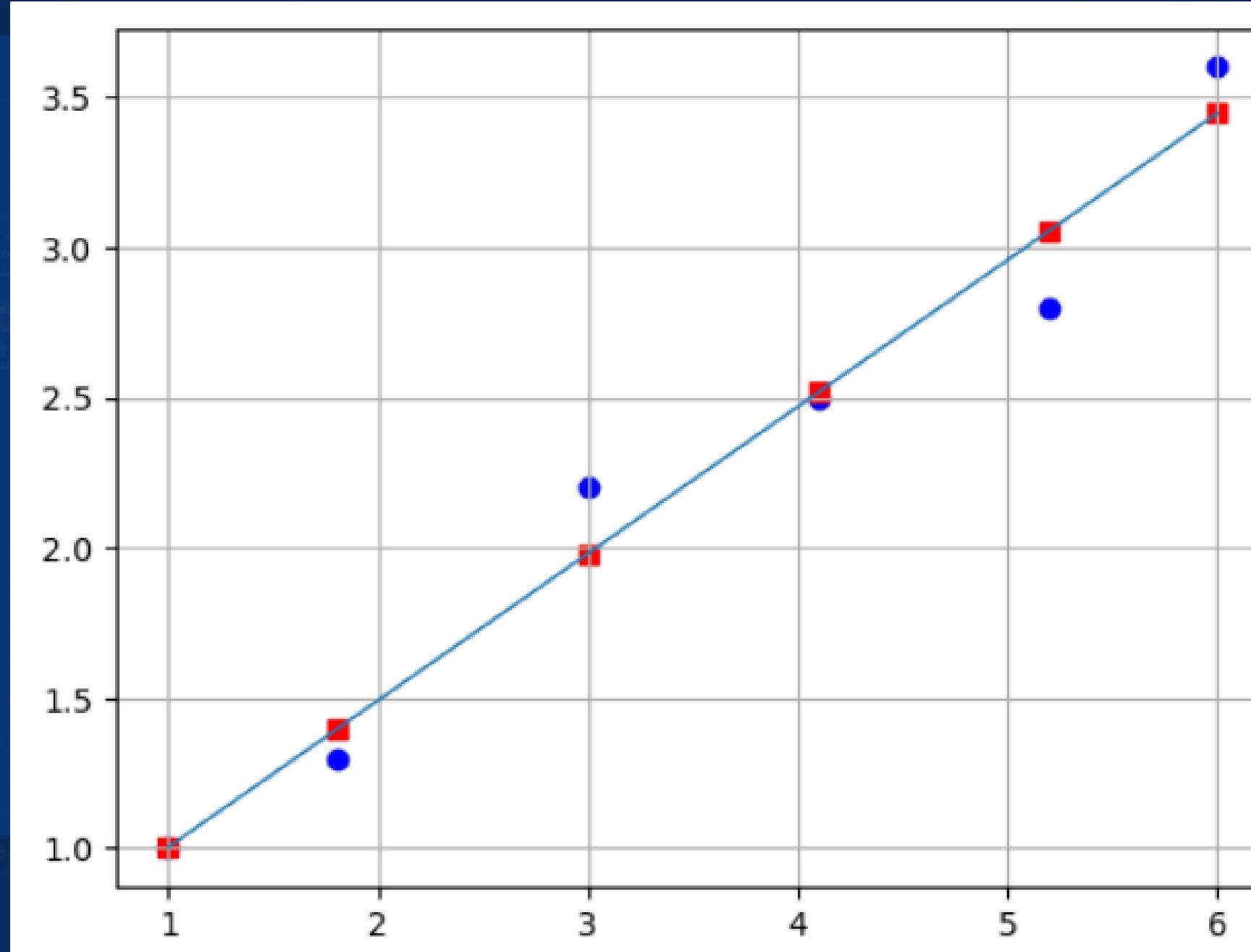
# Jupyter Notebook

## image processing

```
[28]: # ວິນກອນມັດຄົງ input ແລະ ນຳເລັດທີ່  
for i, y in enumerate(y_predict):  
    print('x={0} y={1:.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')      # ນຳມັດຈິງ (real data points)  
plt.plot(x_data, predict, linewidth='1')       # ທຳນາວຸ (prediction)  
plt.scatter(x_data, predict, color='r', marker='s')  
# plt.savefig('modell.png', dpi=100)  
plt.show()
```

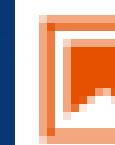
# Jupyter Notebook

## 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
[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])

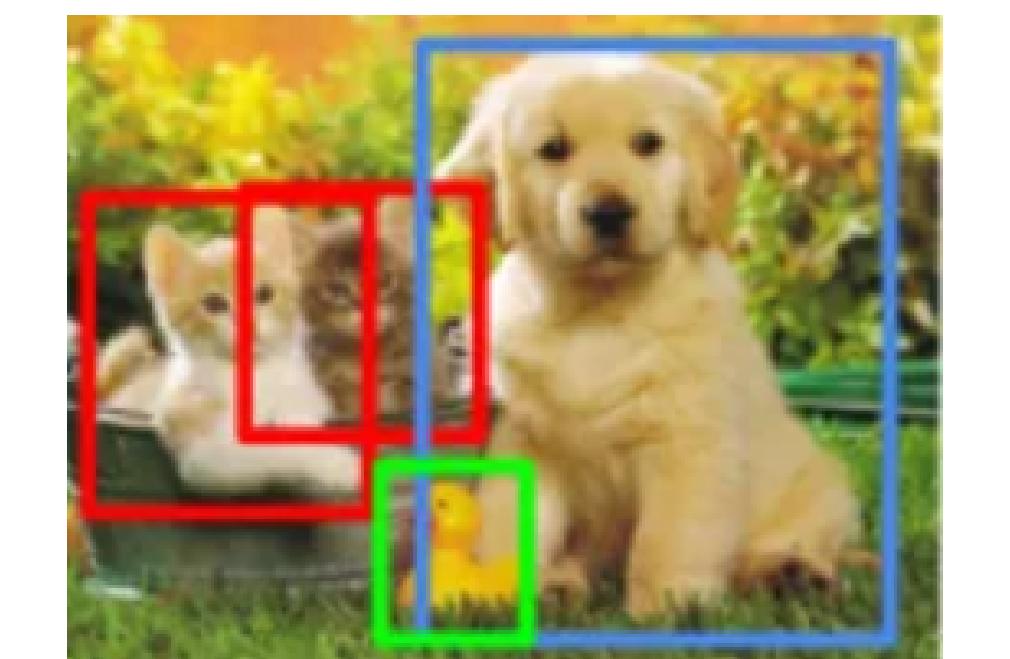
[5]: # วนรอบและ input และ ผลลัพธ์
for i, y in enumerate(y_predict):
    print('x={:.4f} {:.4f}'.format(x_input[i], y))

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

เช่นเดียวกับหลักการของ AI เราต้องส่งรูปเข้าไปพร้อมกับผลลัพธ์ว่า นี่คือรูปแบบ เพื่อให้มันเรียนรู้เรื่อยๆ และนี่คืองานหลักของสิ่งที่เรียกว่า **Image Classification**

แล้วแม่มันอยู่ตรงไหนของรูปหละ? ถ้าได้โจทย์มาเป็นแบบนี้ จะเป็นโจทย์อีกแนวที่เรียกว่า **Image Detection** โดยมันจะบอกถึงตำแหน่งของรูปด้วย



น้องแมวแอนด์เดอะแก๊งค์

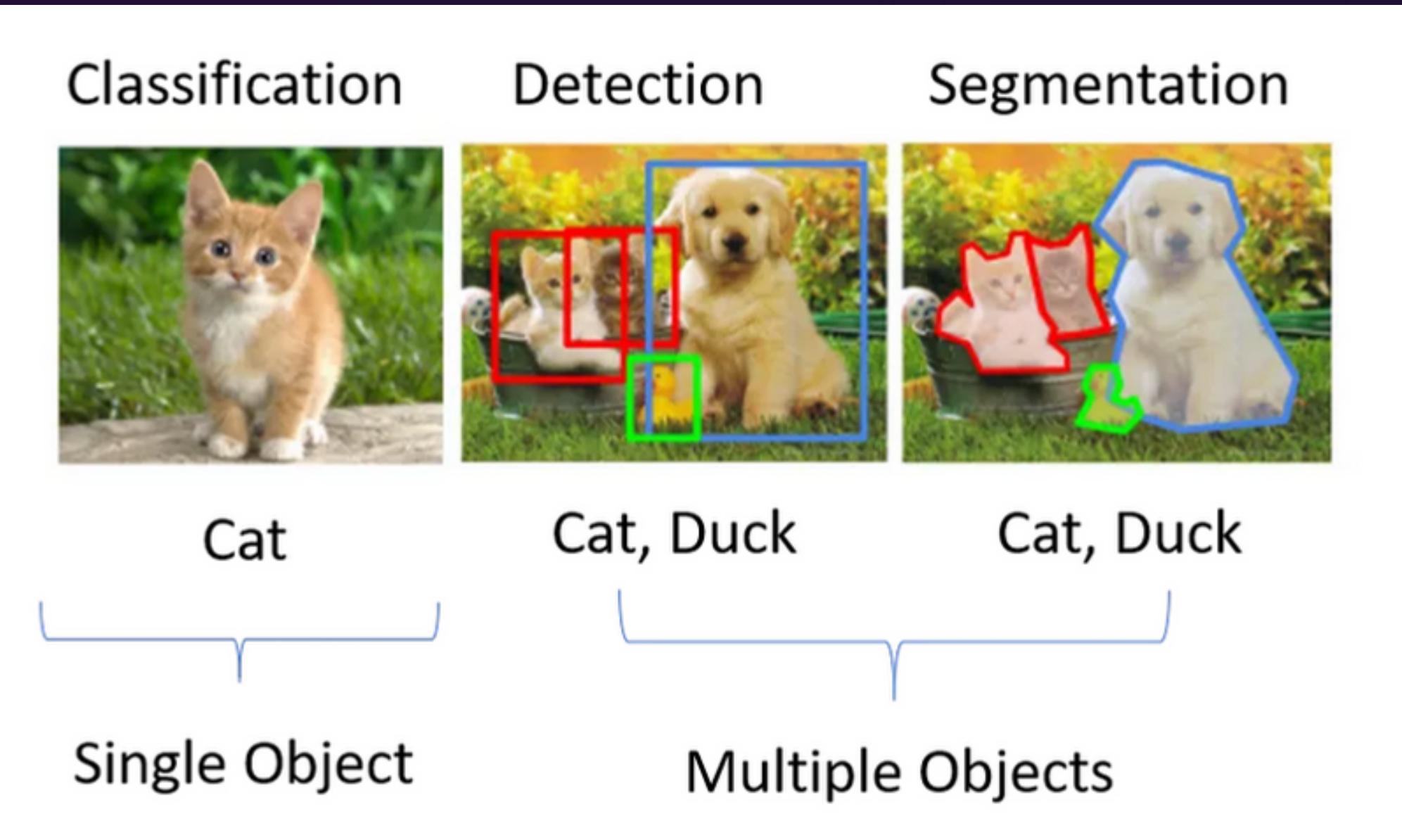
บทความ ความรู้ image classification to AI :

EXIT



# IMAGE CLASSIFICATION

ภาพรวมของโจทย์โดยรวม คือ

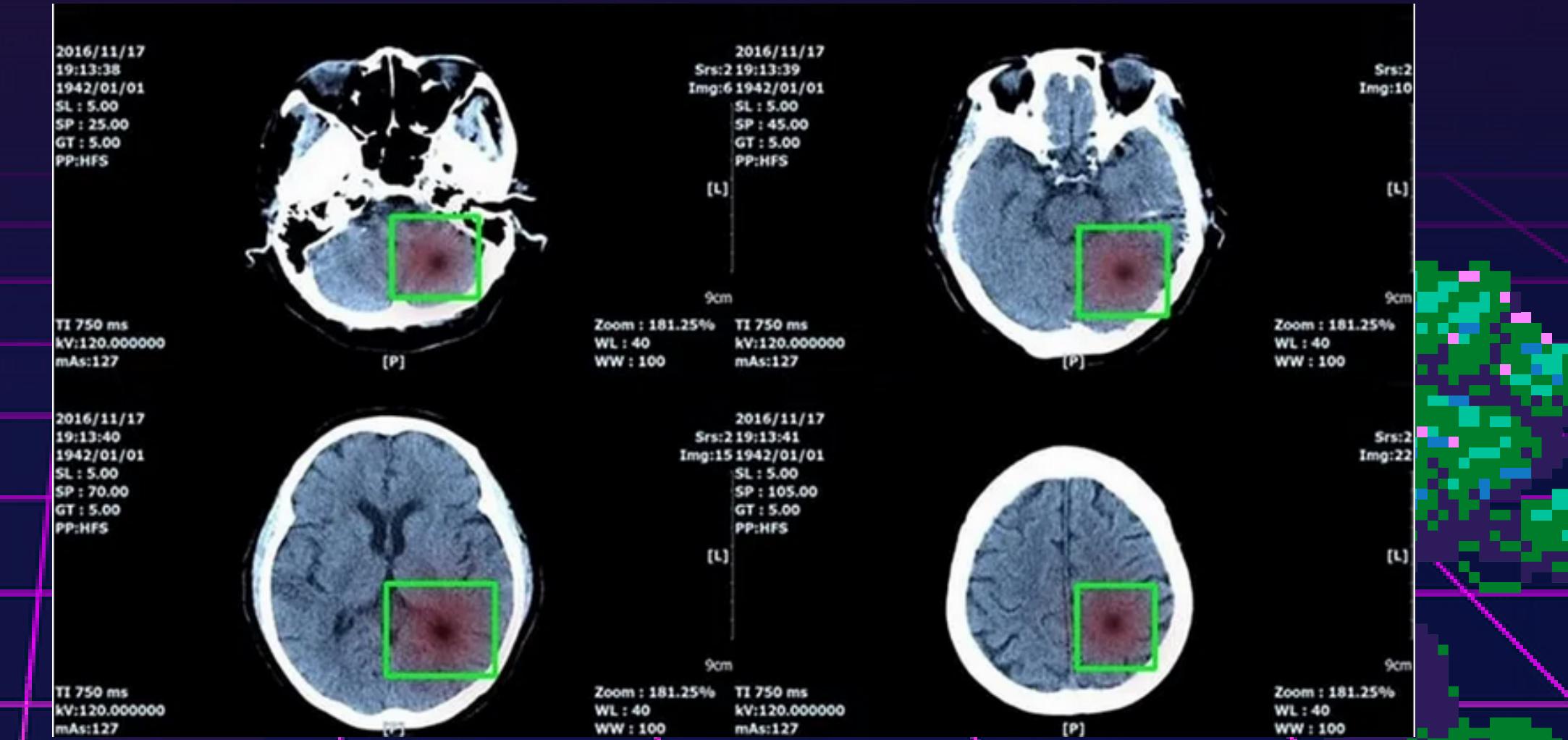


EXIT

# IMAGE CLASSIFICATION

## Medical Sector

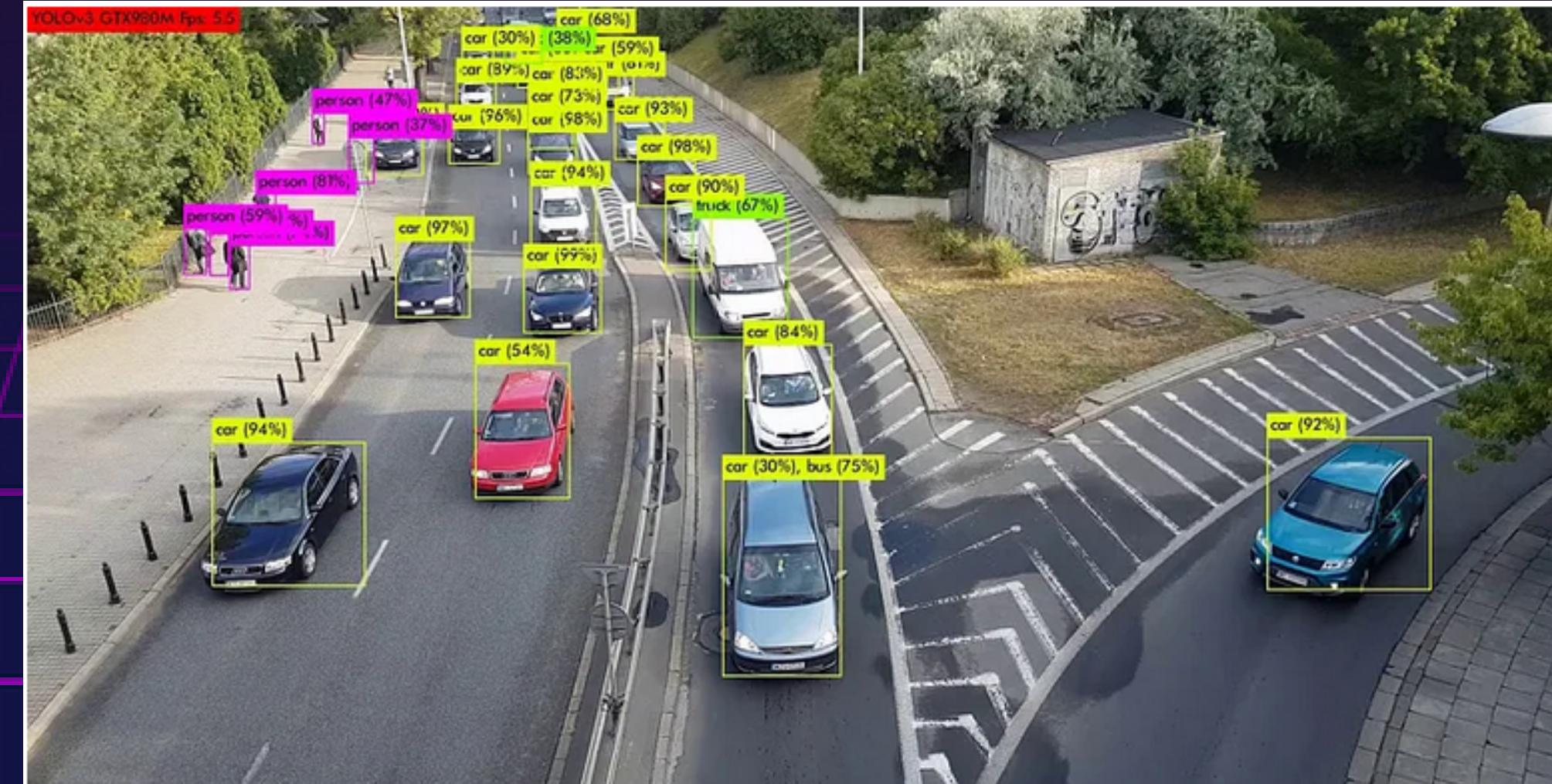
บางครั้งเวลาที่มีรูป X-ray ออกมาก ในการตรวจเช็ครูปนั้นจะต้องใช้หมอน  
เฉพาะทางที่มีจำกัด ถ้าเรามี AI คอยช่วย Focus หรือตีกรอบส่วนที่มี  
ปัญหา จะลดเวลาในส่วนนี้มาก รวมถึงอาจจะใช้งานร่วมกับ Image  
Segmentation ได้ จากรูปด้านล่าง เขาวิเคราะห์เรื่องตำแหน่งของเนื้อ  
อกในสมองครับ



# IMAGE CLASSIFICATION

## Traffic Sector

อีกตัวอย่าง เวลาที่เราขับรถบนท้องถนน ก็จะมีกล้องที่ค่อยเช็คความเร็วหรืออาจ จะตรวจสอบว่ารถติดไหม เราสามารถนำ AI นี้ไปช่วยได้ว่ามีรถบนถนนเยอะหรือ ไม่แล้วไปเชื่อมกับระบบอื่น (รวมไปถึงรถกำลังจะเข้าเส้นทิบใหม)



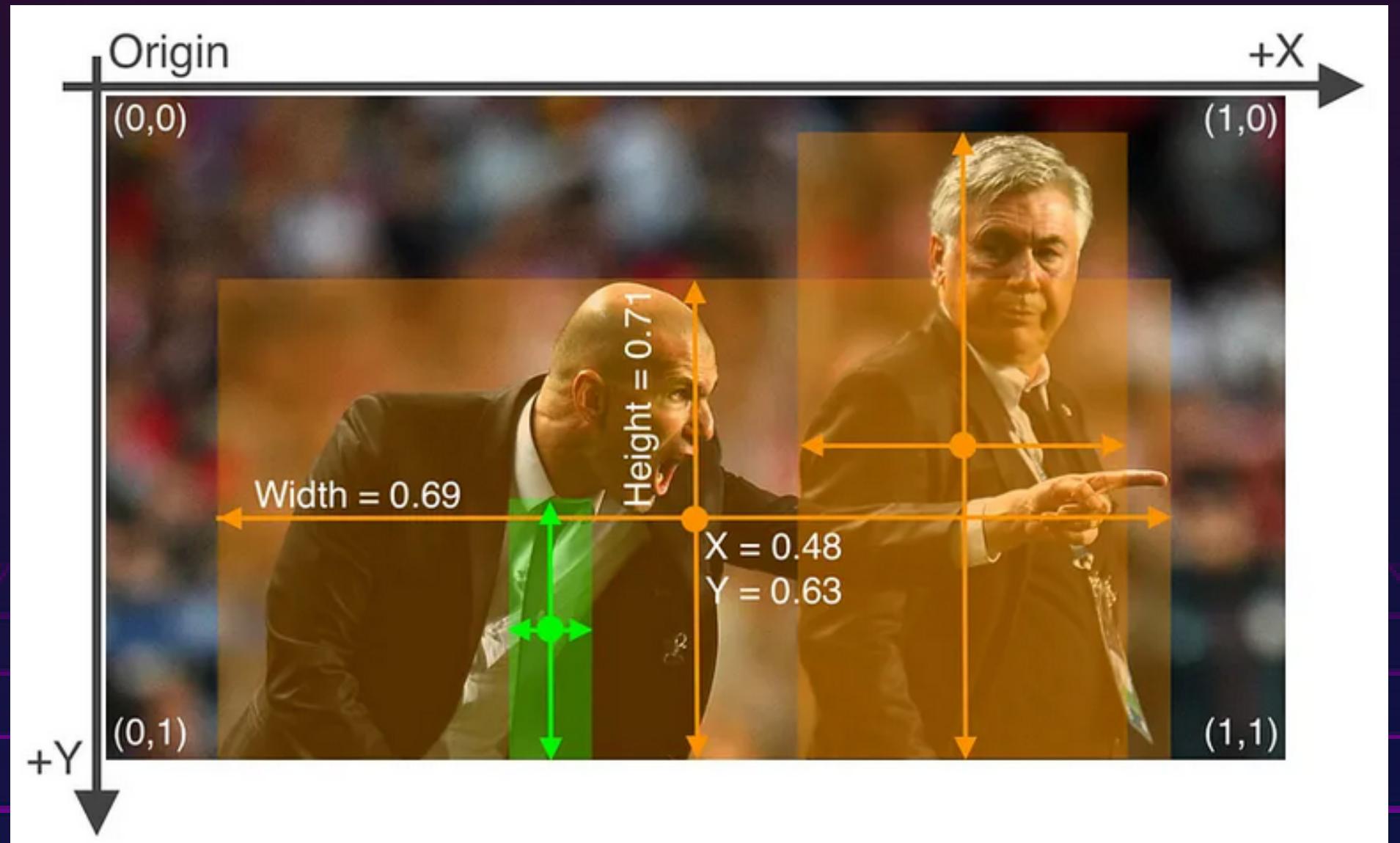
# YOLOv5 คืออะไร ?

มันคือ ‘You only look once’ ซึ่งจะเป็นหลักการทำงานของมัน และในปัจจุบันมีถึง Version 5 แล้ว โดยตัว YOLO นี้คือสถาปัตยกรรมที่ทาง ultralytics ได้ออกแบบไว้เพื่อทำ Image Detection ได้อย่างรวดเร็วและมีประสิทธิภาพ

YOLOv5 by 

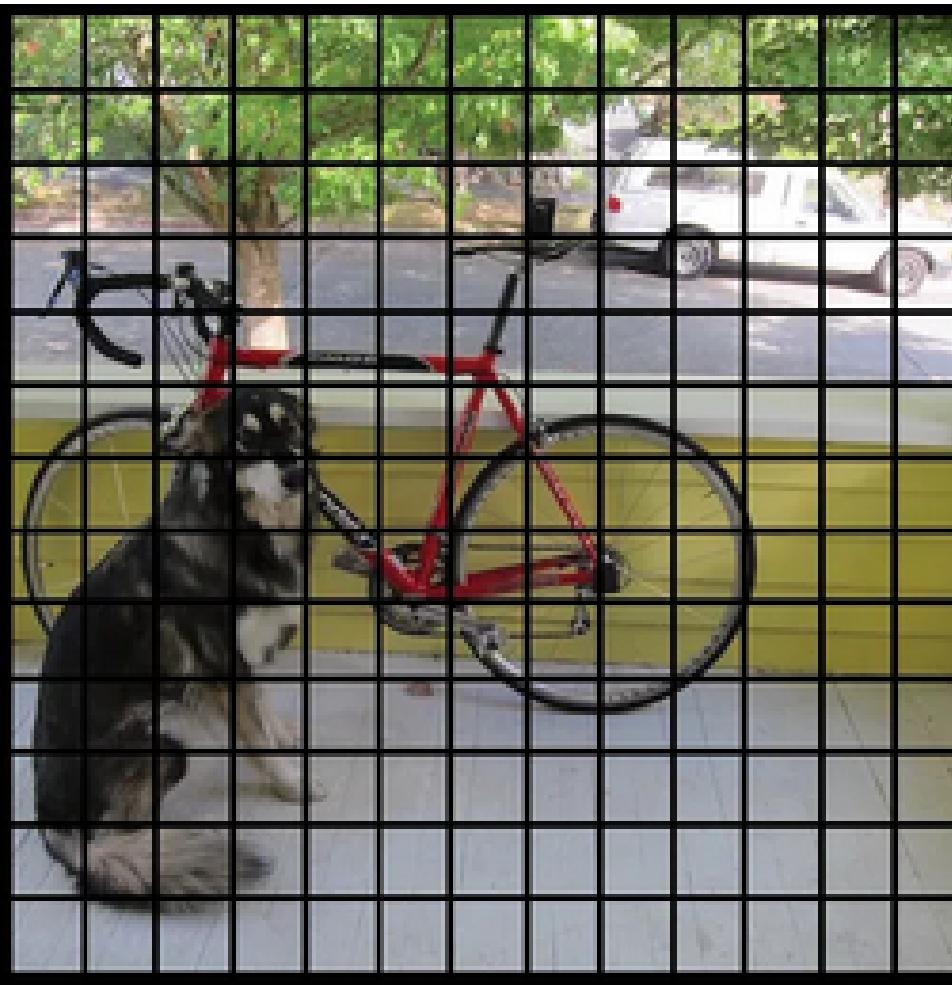
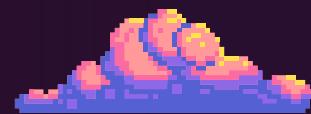


# YOLOV5 គីឡូច្ចិក ?

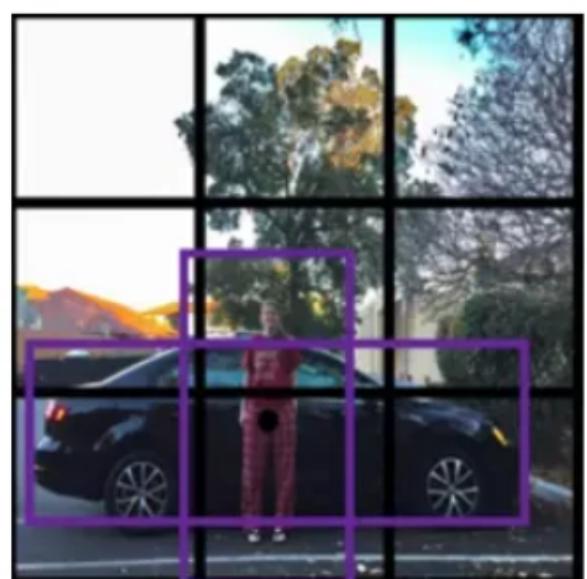


EXIT

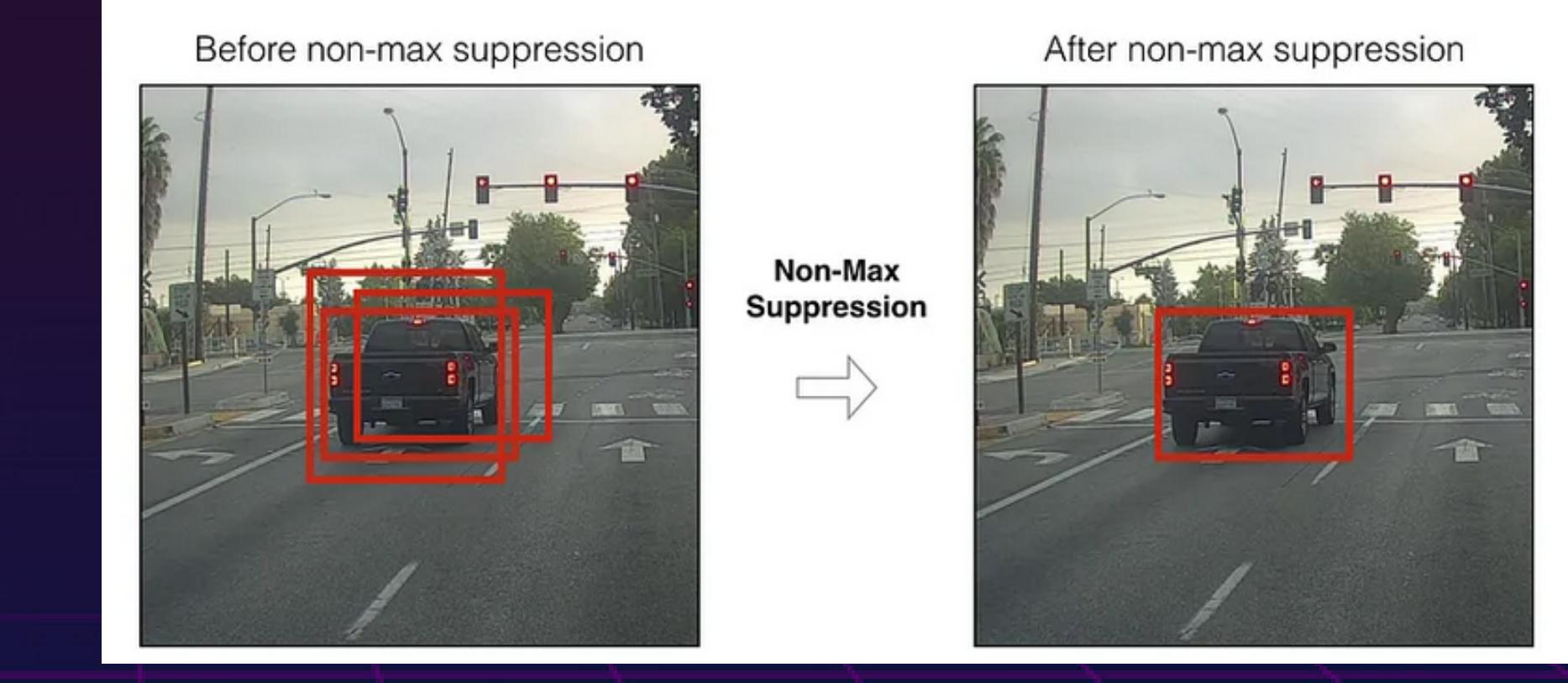
# YOLOV5 คืออะไร ?



การ grid ในแต่ละส่วนของภาพ  
เพื่อ ดู elements pixel



EXIT



layer ที่ predict ได้  
มีการจัดกระทำ intersection

# การติดตั้ง YOLOV5

5 ขั้นตอน)

- 1) Image Detection
- 2) การดึงชุดข้อมูลและจัดเตรียม (Data Gathering and Collecting)
- 3) การกำหนดผลลัพธ์ (Data Labeling) และการทำ Image Augmentation )
- 4) การสร้างโมเดลและวัดผล (Modelling)
- 5) การนำโมเดลไปประยุกต์ใช้ (Deployment)

บทความ medium yolov5 :

EXIT



# ការតាមទី 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". To the right is a search bar. On the left side of the main content area, there's a sidebar with links: "About", "Documentation", "Downloads" (which is highlighted in red), "GUI Clients", "Logos", and "Community". The main content area has a large title "Downloads" and three download links: "macOS", "Windows", and "Linux/Unix". Below these links, a note states: "Older releases are available and the Git source repository is on GitHub." To the right of the main content, there's a monitor icon displaying a teal window titled "Latest source Release 2.42.0" with a "Release Notes (2023-08-21)" link and a "Download for Windows" button. The background of the slide features a dark purple gradient with pixelated green bushes at the bottom.

EXIT

# การติดตั้ง YOLOV5

1.

ดาวน์โหลด YOLOv5 เราสามารถเริ่มดาวน์โหลดและติดตั้ง Library ที่ต้องใช้

```
## 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



# Thank You!

**End of Session AI**