

LECTURE 2-3_1

PYTHON PROGRAMMING

CODES AND STRUCTURES

DR. PRAPASSORN TANTIPHANWADI

INDUSTRIAL ENGINEERING, FACULTY OF ENGINEERING AT KHAMPAENGSAEN

DECEMBER 2565

DATA SCIENCE VS. PYTHON

- Install anaconda
- Install python, tensorflow

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



Products ▾ Pricing Solutions ▾ Resources ▾ Partners ▾ Blog Company ▾

Download Anaconda installers

Data science technology for a competitive edge.

A movement that brings together millions of data science practitioners, data-driven enterprises, and the open source community.

Anaconda Individual Edition

Download

For Windows

Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Started



CREATE ENVIRONMENT

File Help



Home

Environments

Learning

Community

Applications on base (root) Channels

CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch

Datalore Online Data Analysis To coding assistance by JetBr your Python notebooks i share them with yi Launch

Create new environment X

Name: py38

Location: C:\Users\HP\anaconda3\envs\py38

Packages: Python 3.8 R r

Cancel Create

Anaconda Navigator
File Help

ANACONDA.NAVIGATOR

Home

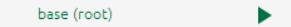
Environments

Learning

Community

Search Environments

base (root)



Installed Channels Update index... Search Packages

Installed

Channels

Update index...

Search Packages

Name T Description

<input checked="" type="checkbox"/> _ipyw_jlab_nb_ex...	
<input checked="" type="checkbox"/> alabaster	
<input checked="" type="checkbox"/> anaconda	
<input checked="" type="checkbox"/> anaconda-client	
<input checked="" type="checkbox"/> anaconda-project	
<input checked="" type="checkbox"/> anyio	
<input checked="" type="checkbox"/> appdirs	
<input checked="" type="checkbox"/> argh	
<input checked="" type="checkbox"/> argon2-cffi	
<input checked="" type="checkbox"/> asn1crypto	
<input checked="" type="checkbox"/> astroid	
<input checked="" type="checkbox"/> astropy	
<input checked="" type="checkbox"/> async-generator	
<input checked="" type="checkbox"/> async_generator	
<input checked="" type="checkbox"/> atomicwrites	
<input checked="" type="checkbox"/> attrs	
<input checked="" type="checkbox"/> autopep8	
<input checked="" type="checkbox"/> babel	
<input checked="" type="checkbox"/> backcall	
<input checked="" type="checkbox"/> backports	

362 packages available



CREATE ENVIRONMENT

Installed

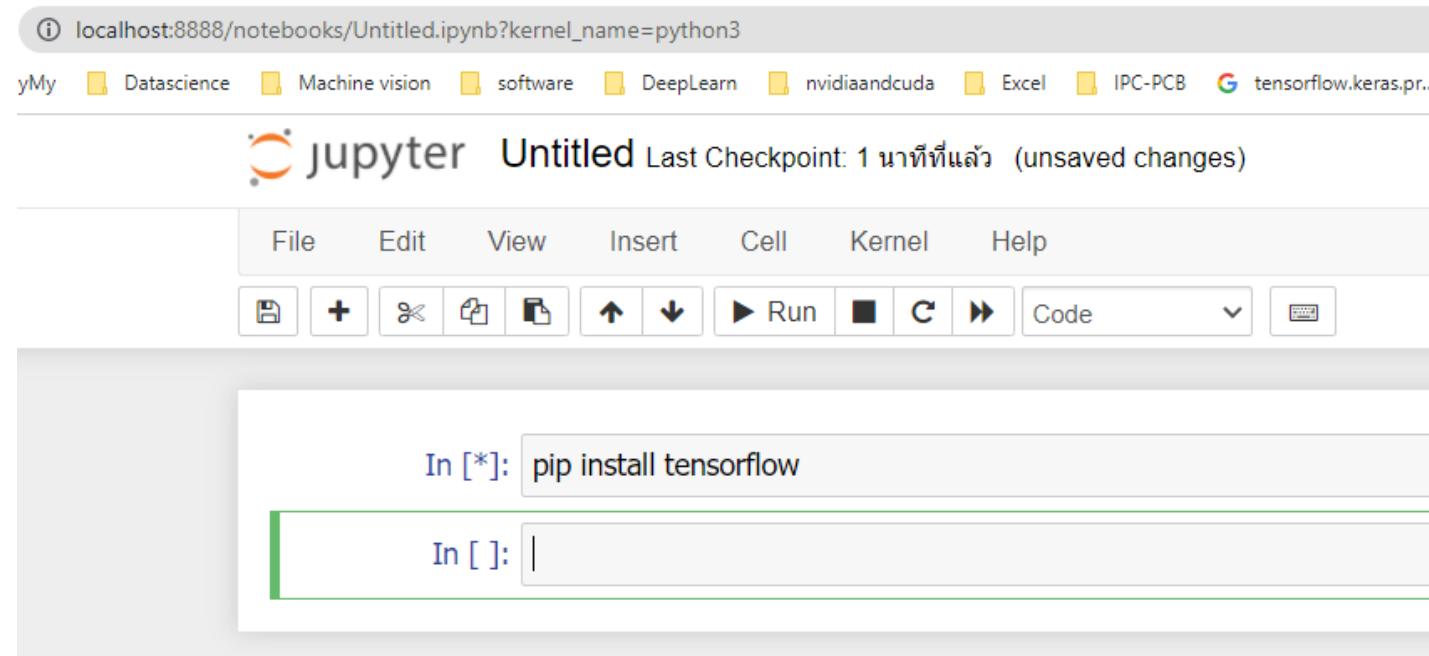
Name	T	Description
ca-certificates	○	
certifi	○	
openssl	○	
pip	○	
python	○	
setuptools	○	
sqlite	○	
vc	○	
vs2015_runtime	○	
wheel	○	
wincertstore	○	

Applications on py38

Channels

Datalore	IBM Watson Studio Cloud	CMD.exe Prompt	Glueviz	JupyterLab	Notebook
Launch	Launch	Install	Install	Install	Install
Orange 3	Powershell Prompt	PyCharm Professional	Qt Console	RStudio	Spyder
3.26.0	0.0.1	5.0.3	5.0.3	1.1.456	5.0.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	Run a Powershell terminal with your current environment from Navigator activated	A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	Scientific Python Development EnviRonment: Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Install	Install	Install	Install	Install	Install

CREATE ENVIRONMENT



n1-modules-0.2.8 requests-oauthlib-1.3.0 rsa-4.7.2 tensorboard-2.7.0 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.0 tensorflow-2.7.0 tensorflow-estimator-2.7.0 tensorflow-io-gcs-filesystem-0.21.0 termcolor-1.1.0 typing-extensions-3.10.0.2 werkzeug-2.0.2 wrapt-1.13.3
Note: you may need to restart the kernel to use updated packages.

In [2]: `import tensorflow as tf`

In [3]: `import tensorflow.keras`

In []: (empty cell)

DATA SCIENCE กับ PYTHON

PYTHON PROGRAMMING

- Easy to learn
- Cross platform
- Fast processing than R or Matlab
- Better in resource management
- Many libraries
- Package for other langage, such as C, Java

PYTHON LIBRARIES

Pandas

- Open source
- Arrange data into data structure – Series, Data Frame
- Steps – load, organize, manipulate, model, analyze

NumPy

- Abbr. from “Numeric Python”
- Open source
- Mathematics, calculation, science, engineering, statistics, business, graphic
- Manage array, matrices, vector, tensor
- Use together with SciPy (Scientific Python), Matplotlib (plotting library)

SciPy

- Work together with ndarray ឧប់ NumPy
- Compose of small packages of mathematics, Physics, Calculus, Linear Algebra, Fourier Series,...

PYTHON LIBRARIES

Matplotlib

- Use for Data Visualization
- Function for กราฟ แท่ง วงกลม 2 มิติ 3 มิติ
- Pyplot กำหนดรูปแบบตัวหนังสือ
- ใช้ร่วมกับ NumPy และ libraries อื่นๆสำหรับ Graphic

Seaborn

- สำหรับเพล็อกกราฟในการทำ Data Visualization ซึ่งเรียกใช้ Matplotlib
- กราฟค่าการกระจายตัวของข้อมูลทางสถิติ

JUPYTER NOTEBOOK และ GOOGLE COLAB

- เขียนโปรแกรมผ่าน IDE (Integrated Development Environment)
- IDE ประกอบด้วย
 - Source code editor
 - ตัวแปลภาษา
 - Debugger
 - Some libraries

Jupyter Notebook

- เป็น IDE ประเภทหนึ่ง
- Open source web application ที่ทำงานบน web browser ในลักษณะ client-server application
- การใช้งานผ่านทางโปรแกรม Anaconda ซึ่งมี libraries ค่อนข้างครบ
- การเรียกใช้งาน ไม่จำเป็นต้องเชื่อมต่อ internet

Google Colab

- Google Colaboratory
- ใช้งาน Jupyter Notebook บน cloud
- ไม่จำเป็นต้องติดตั้งโปรแกรมใดๆ
- การเรียกใช้งาน จะเป็นต้องเชื่อมต่อ internet
- ทุกคนใช้งานได้ฟรี เพียงแต่ต้องมีบัญชี Google Drive

ANACONDA

The screenshot shows the Anaconda website's header. It features the Anaconda logo, a navigation bar with links for Products, Pricing, Solutions, Resources, Blog (which is underlined in green), Company, and a prominent 'Get Started' button. The browser's address bar shows 'anaconda.com'. Below the header, there's a large banner with the text 'Data science technology for a better world.' and a subtext about a movement involving millions of practitioners, enterprises, and the open source community. On the left, there's a teal sidebar with links for 'Hello! Let's get started!', 'See all Anaconda products', 'Check out the latest in data science', 'Request an Anaconda demo', and 'Download Anaconda installers' (which is highlighted with a red border). The main content area has sections for Windows, Mac OS, and Linux installers.

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

Data science technology for a better world.

A movement that brings together millions of data science practitioners, data-driven enterprises, and the open source community.

Hello! Let's get started!

[See all Anaconda products](#)

science

Anaconda

[Check out the latest in data science](#)

[Request an Anaconda demo](#)

Install Anaconda on your laptop or desktop

[Download Anaconda installers](#)

Dr. Prapassorn Kamphamwadi. Copies are prohibited.

Windows

[Python 3.8
64-Bit Graphical Installer \(477 MB\)](#)

[32-Bit Graphical Installer \(409 MB\)](#)

Anaconda Installers

MacOS

[Python 3.8
64-Bit Graphical Installer \(440 MB\)](#)

[64-Bit Command Line Installer \(433 MB\)](#)

Linux

[Python 3.8
64-Bit \(x86\) I](#)

[64-Bit \(Power
MB\)](#)

11/29/2022

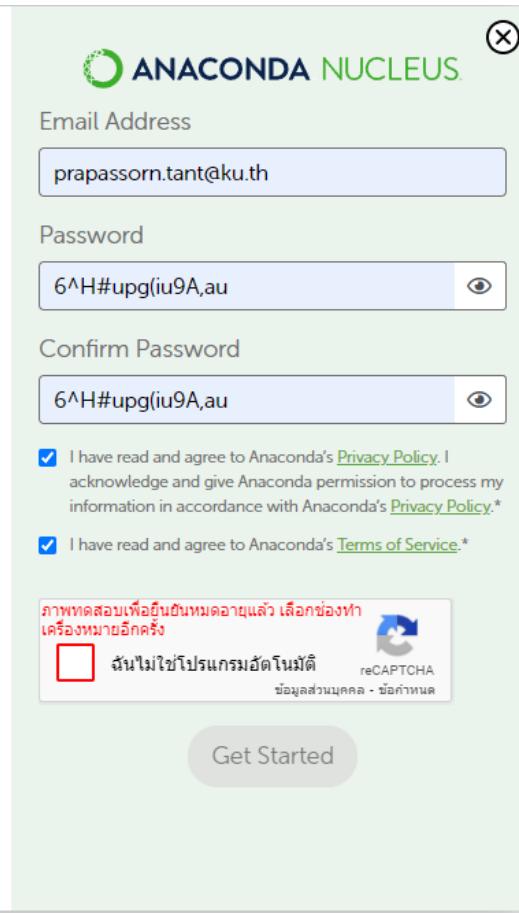
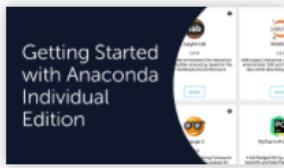
10

ANACONDA

Thank you for
downloading

Get Started with Anaconda Individual Edition

Available only in Anaconda's cloud, this free, hands-on Jupyter notebook provides an introduction to help you get started using this powerful tool.



Email Verification Issue

There was an issue verifying, please try again.

[Resend Email](#)



Welcome to Anaconda Nucleus.

To get started, please verify your email.

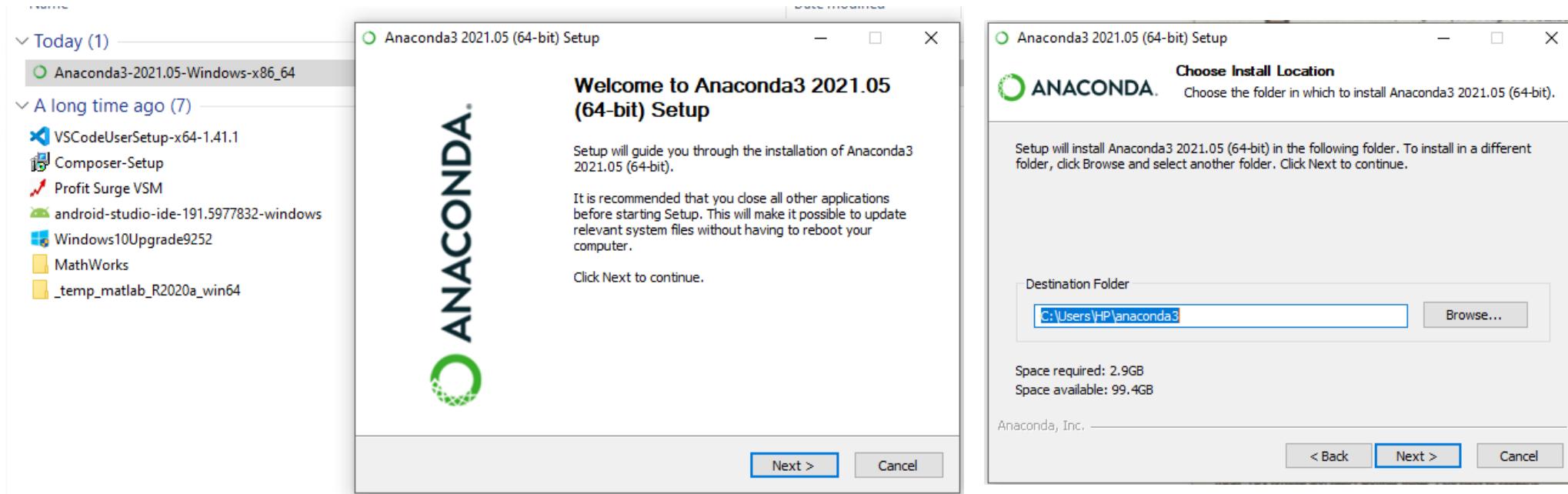
We do this as a security precaution to verify your credentials. If you did not sign up for Anaconda Nucleus, you can safely ignore this email.

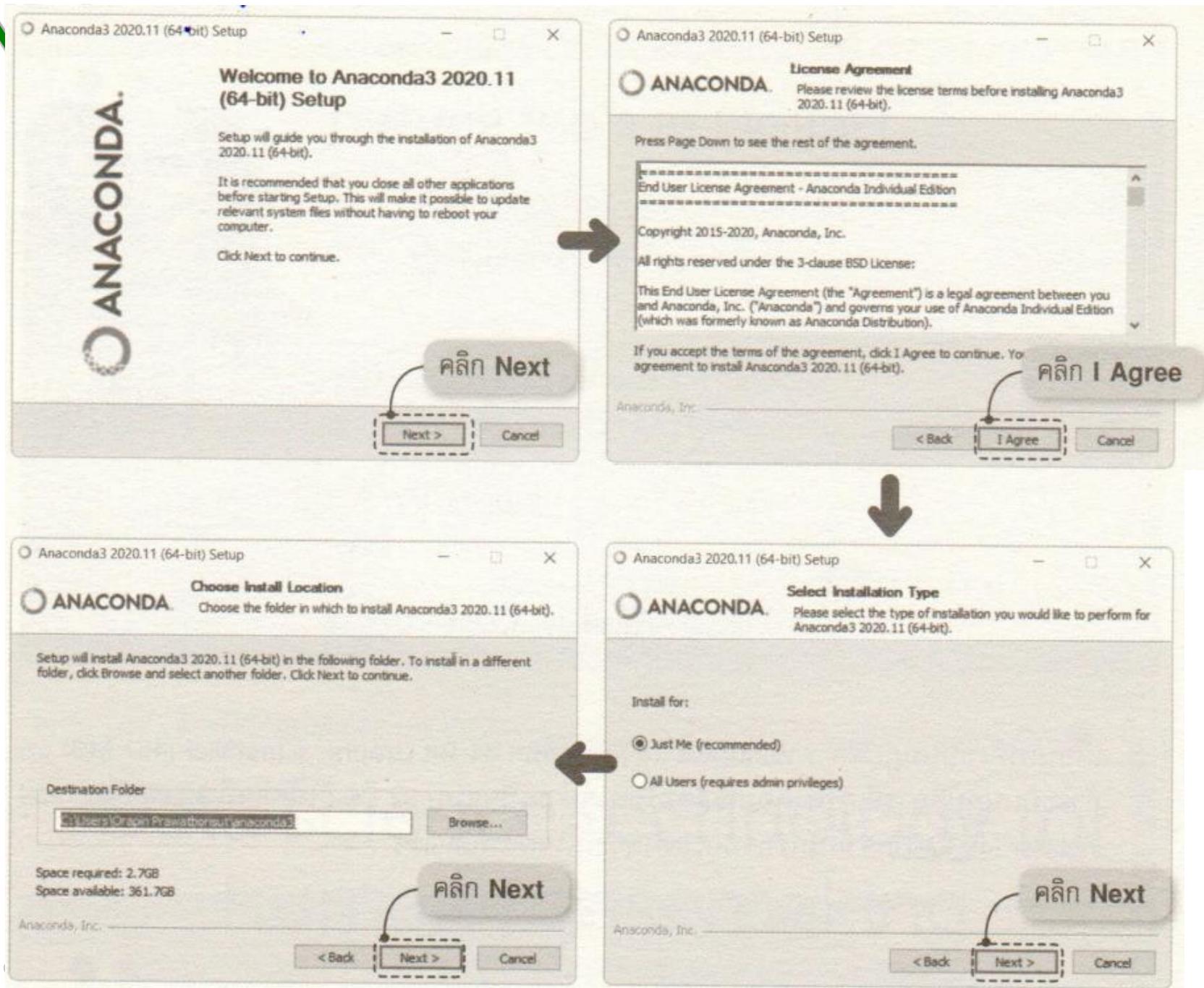
[Verify Email](#)

Anaconda Nucleus is a learning platform for everything data science related curated for students, hobbyists, and data science practitioners. Explore our library of learning content, including how-to videos and expert insights, all free for a limited time to Nucleus members!

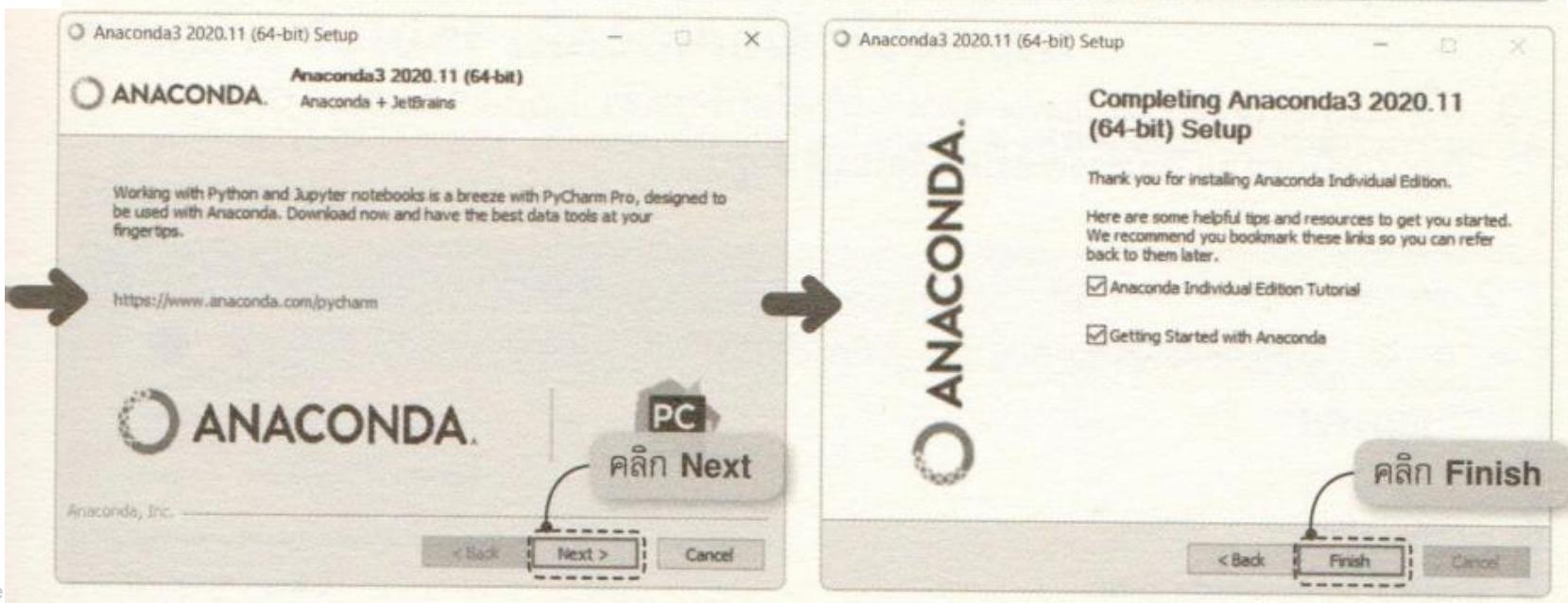
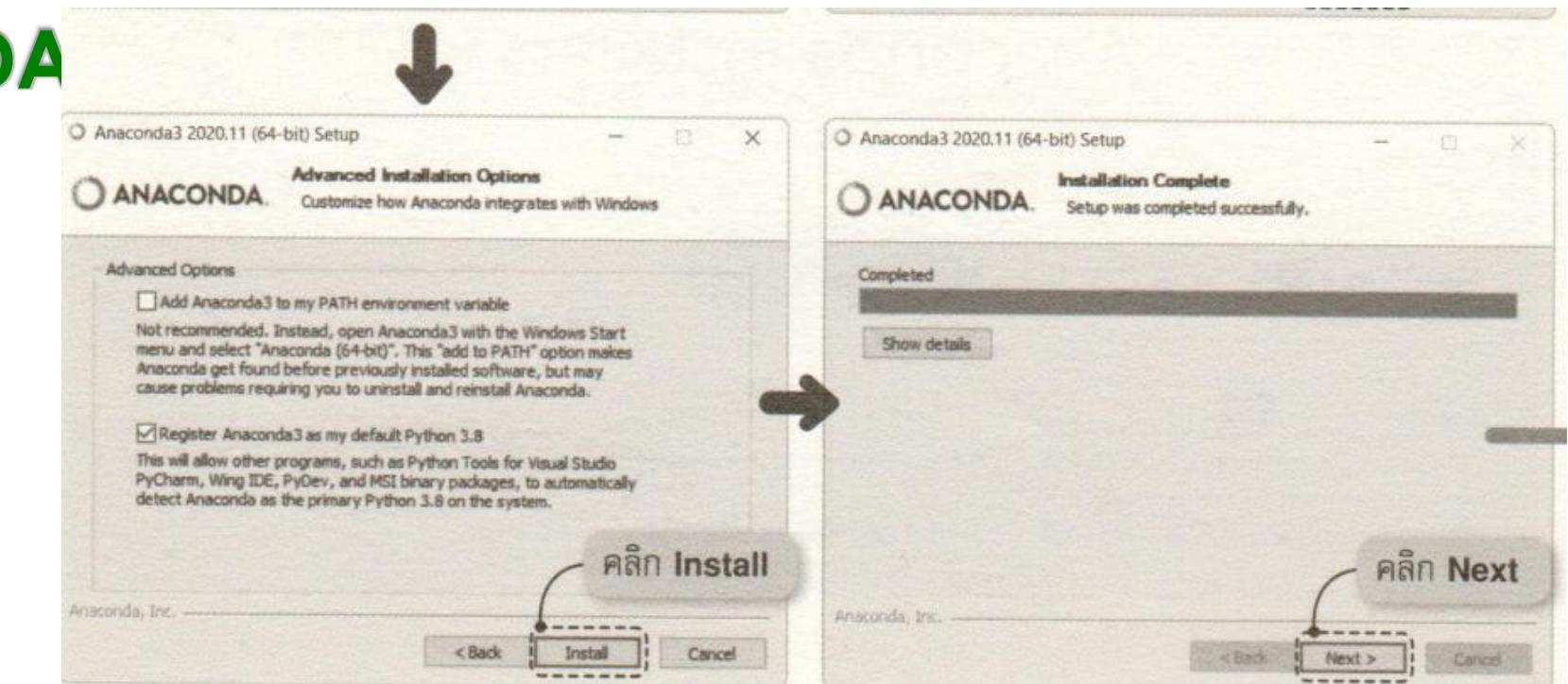
Whether you're just beginning your data science journey or a data science expert, we have content just for you. Take a look at some of our most popular content to kickstart your journey.

ANACONDA





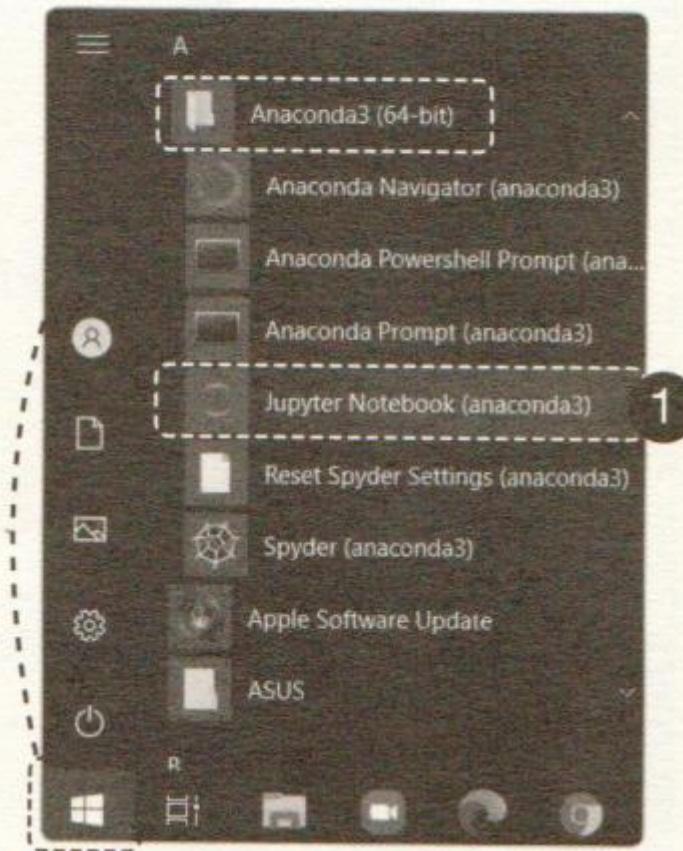
ANACONDA



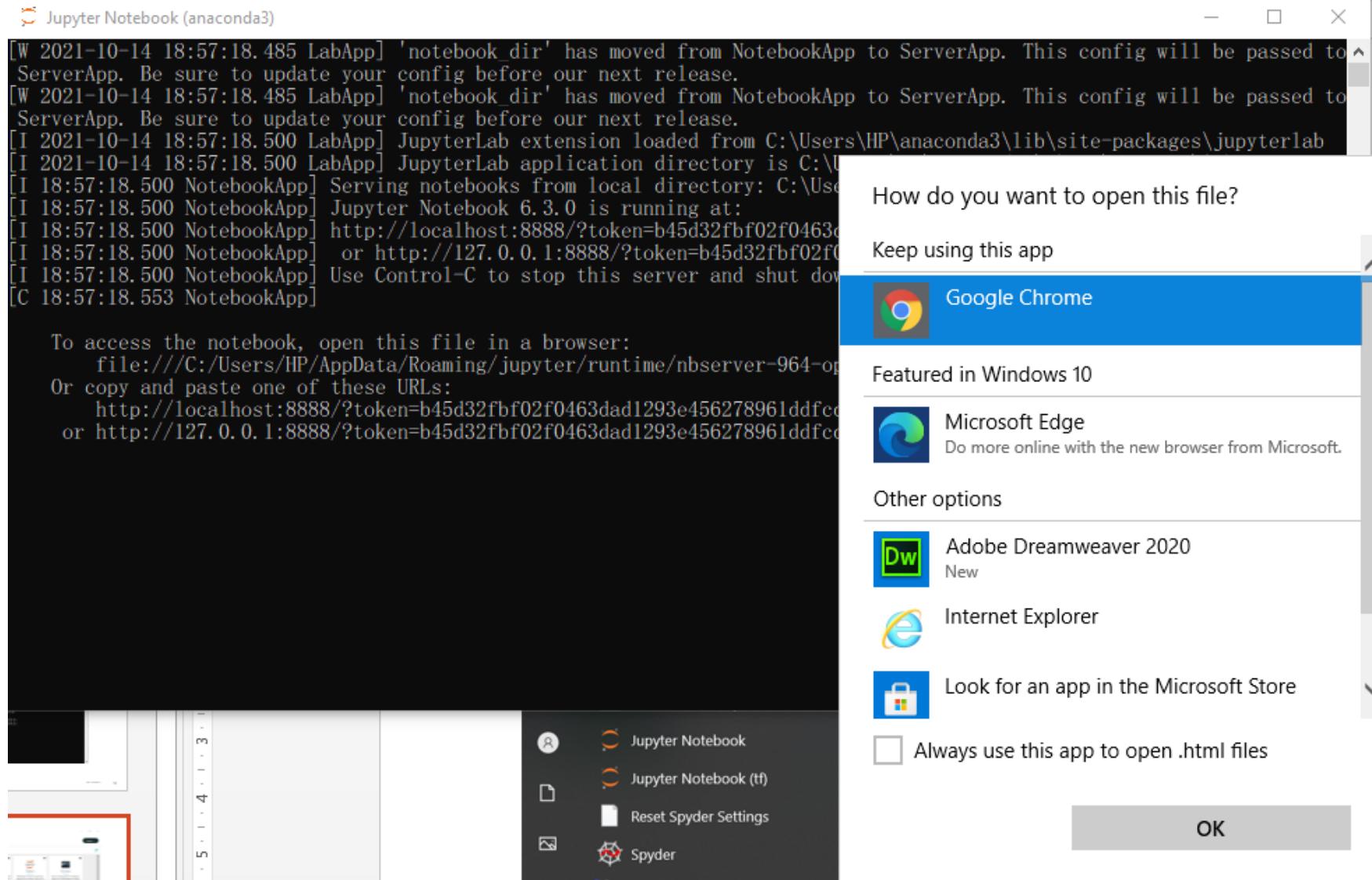
การเรียกใช้งาน Jupyter Notebook

การเรียกใช้งาน Jupyter Notebook มีขั้นตอนดังนี้

- คลิกปุ่ม Start และเลือกที่ Anaconda3 (64-bit)
จากนั้นคลิกเลือก Jupyter Notebook (anaconda3)
- Jupyter Notebook Server จะถูกสร้างขึ้นมา ดังรูป



ANACONDA



ANACONDA

The screenshot shows the Jupyter Notebook interface running on a local host. The top navigation bar includes icons for back, forward, search, and a purple profile icon labeled 'ยังผิดพลาด'. Below the bar, there are several folder icons: 'แอป' (App), 'MyMym', 'Datasience', 'Machine vision', 'software', and 'DeepLearn'. To the right of these are 'Next' and 'Recent' buttons.

The main title 'jupyter' is displayed with its logo. On the right side of the title bar are 'Quit' and 'Logout' buttons.

Below the title bar, there are three tabs: 'Files' (selected), 'Running', and 'Clusters'.

A message 'Select items to perform actions on them.' is centered on the page.

On the right side of the interface, there are buttons for 'Upload', 'New', and a refresh symbol.

The main content area displays a file browser with the following data:

	Name	Last Modified	File size
<input type="checkbox"/> 0	/		
<input type="checkbox"/>	anaconda3	12 นาทีที่แล้ว	
<input type="checkbox"/>	AndroidStudioProjects	2 ปีที่แล้ว	
<input type="checkbox"/>	Documents	2 ปีที่แล้ว	
<input type="checkbox"/>	MATLAB Drive	1 ปีที่แล้ว	
<input type="checkbox"/>	OneDrive	2 เดือนที่แล้ว	

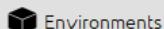
ANACONDA

ANACONDA.NAVIGATOR

Connect ▾



Home



Environments



Learning



Community

Applications on

base (root)

Channels

CMD.exe Prompt
0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

[Launch](#)

Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

[Launch](#)

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.

[Launch](#)

JupyterLab
0.35.3

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Launch](#)

Jupyter Notebook
5.7.4

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

[Launch](#)

Qt Console
4.4.3

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

[Launch](#)

Spyder
3.3.2

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

[Launch](#)

Glueviz
1.0.0

Multidimensional data visualization across files. Explore relationships within and among related datasets.

[Install](#)

Orange 3
3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

[Install](#)

PowerShell Prompt
0.0.1

Run a PowerShell terminal with your current environment from Navigator activated

[Install](#)

PyCharm Professional

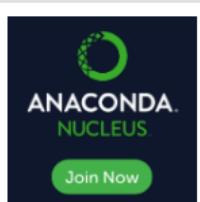
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

[Install](#)

RStudio
1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

[Install](#)



Join Now

JUPYTER NOTEBOOK

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the URL is `localhost:8888/tree`. The main area displays a file browser with a sidebar containing links to various local drives and cloud services like OneDrive, Google Drive, and OpenVPN. A context menu is open over the 'New' button, listing options for creating new notebooks: 'Notebook: Python 3', 'Python [cond]', 'Text File', 'Folder', and 'Terminal'. The bottom status bar shows the path `localhost:8888/tree#`, a search bar, and system icons for weather, battery, and network.

localhost:8888/tree#

Type here to search

Dr. Prapassorn Tantiprathanawadi. Copies are promoted.

File browser sidebar:

- My..my
- DataScience
- Machine Vision
- software
- Manufacturing
- MyBiz
- ADintrend.TV - ดูที่วีอ...
- Breaking the curse of...
- CNN vs small dataset...
- ເຮືອງອ່ານ

New Notebook Dialog:

- Notebook: Python 3
- Python [cond] Create a new notebook with Python 3
- Other:
 - Text File
 - Folder
 - Terminal

Recent Notebooks:

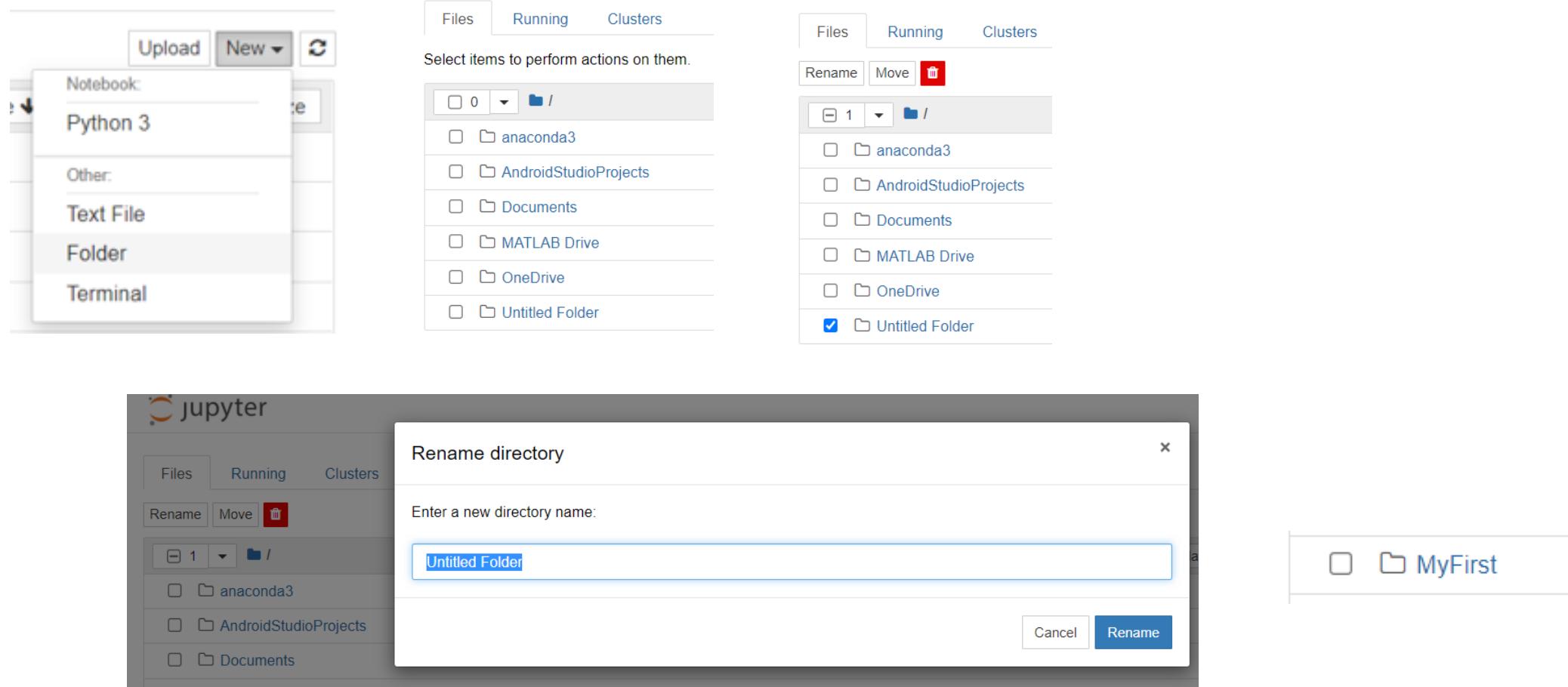
- 4 ชั่วโมงที่แล้ว
- 9 เดือนที่แล้ว
- 5 ปีที่แล้ว
- 1 ปีที่แล้ว
- 1 ปีที่แล้ว
- 2 เดือนที่แล้ว
- 5 ปีที่แล้ว
- 10 เดือนที่แล้ว
- 1 ปีที่แล้ว

JUPYTER NOTEBOOK

The screenshot shows a Jupyter Notebook interface. At the top left is the Jupyter logo and the title "Untitled". To its right is the message "Last Checkpoint: 17/03/2019 (unsaved changes)". On the far right are a Python logo icon and a "Logout" button. The top navigation bar includes links for File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of these are buttons for Trusted status and Python 3 kernel selection. Below the navigation is a toolbar with icons for file operations like Open, Save, and New, as well as cell controls like Run, Execute, and Cell Type. The main workspace contains a single code cell labeled "In []:".

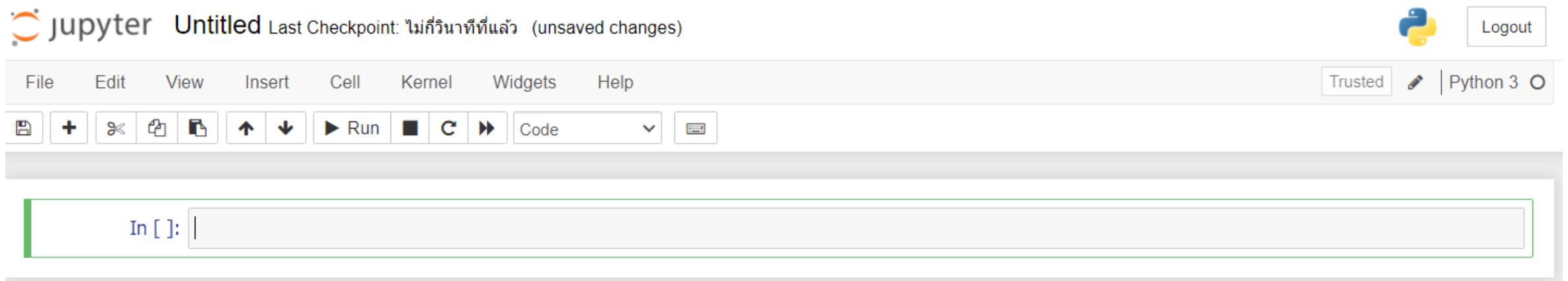
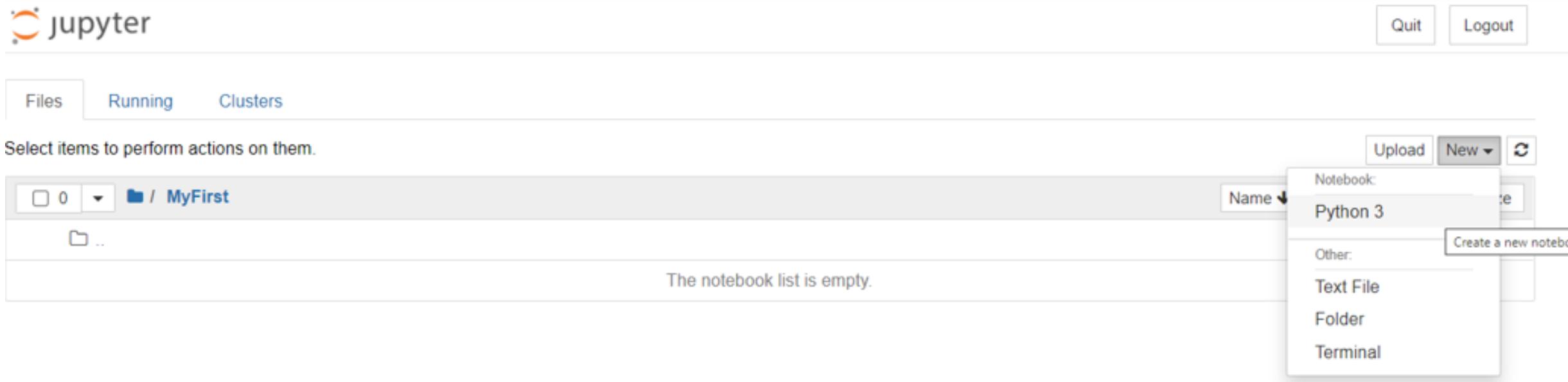
การจัดการ FILES และ FOLDERS

1. การสร้าง Folder เพื่อจัดเก็บโค๊ดโปรแกรม



การจัดการ FILES และ FOLDERS

2. การเขียนโปรแกรมลงบน Jupyter Notebook



การจัดการ FILES และ FOLDERS

2. การเขียนโปรแกรมลงบน Jupyter Notebook

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

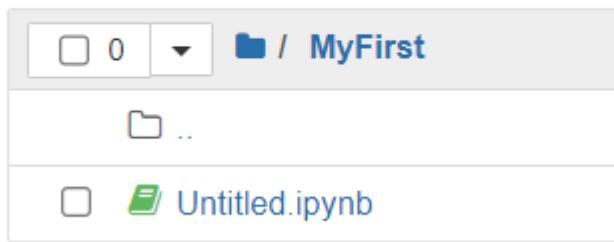
- Header:** jupyter Untitled Last Checkpoint: 3 นาทีที่แล้ว (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3
- Cell 1 (In [1]):** print("Hell world")
print("สวัสดี ชาวโลก")
Output: Hell world
สวัสดี ชาวโลก
- Cell 2 (In []):** print("Hell world!!!")
print("สวัสดี ชาวโลก!!!")
Output: (empty)
- Cell 3 (In [1]):** print("Hell world")
print("สวัสดี ชาวโลก")
Output: Hell world
สวัสดี ชาวโลก
- Cell 4 (In [2]):** print("Hell world!!!")
print("สวัสดี ชาวโลก!!!")
Output: Hell world!!!
สวัสดี ชาวโลก!!!
- Cell 5 (In []):** (empty)

การจัดการ FILES และ FOLDERS

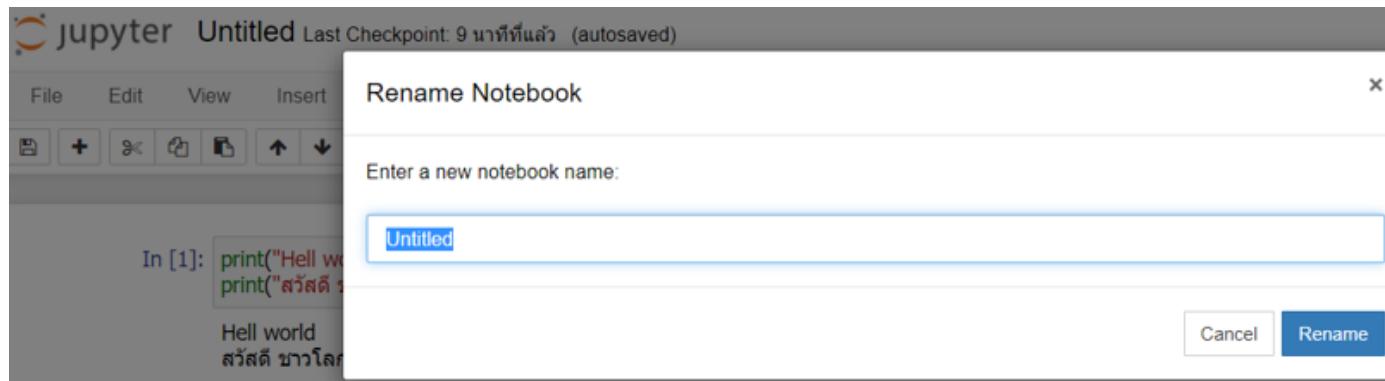
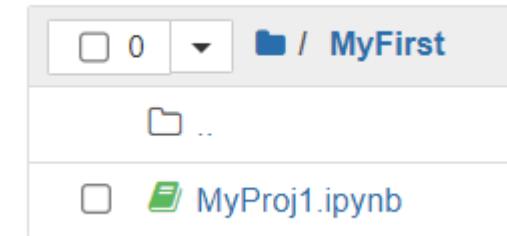
2. การบันทึกโค้ดโปรแกรมลงใน Jupyter Notebook

jupyter Untitled Last Checkpoint: 8 นาทีที่แล้ว (autosaved)

Select items to perform actions on them.



jupyter MyProj1 Last Checkpoint: 10 นาทีที่แล้ว (autosaved)



การจัดการ FILES และ FOLDERS

2. การเขียนโปรแกรมลงบน Jupyter Notebook

- การลบและเพิ่มเซลล์
- การคัดลอก / ตัดเซลล์เดิมไปไว้ยังเซลล์ใหม่
- การสลับที่เซลล์
- การรวมเซลล์และแตกเซลล์

3. การบันทึกโค้ดโปรแกรมใน Jupyter Notebook

- การบันทึกไฟล์เป็นชื่ออื่น Rename

4. การเปิดไฟล์โค้ดโปรแกรมใน Jupyter Notebook

5. การเขียน comment ใน Python

In [2]:

```
# MyFirstPython Program
# Create by Prapassorn Tantiphanwadi
print(1+1) # Test point
print("Hello Python") # Test print
```

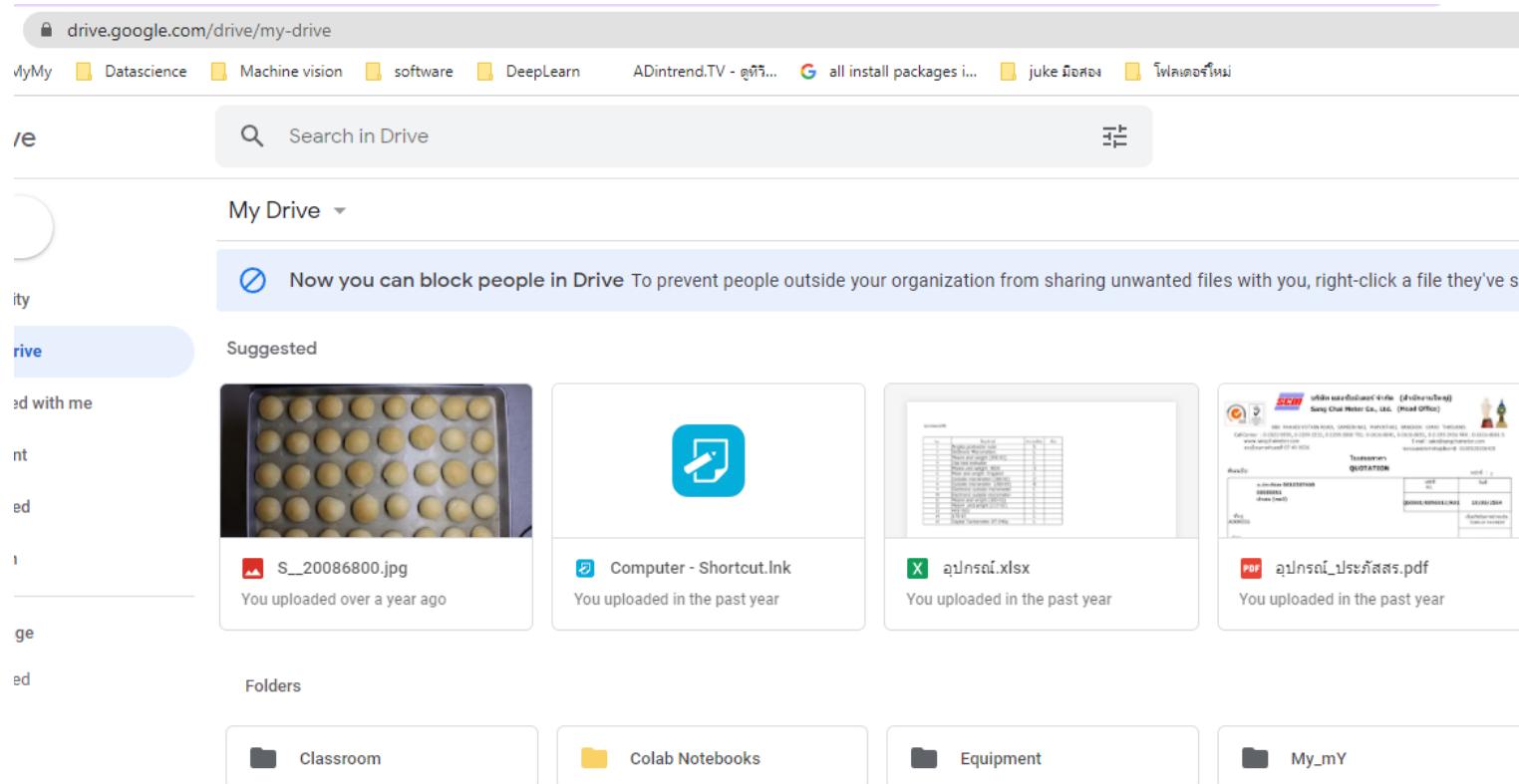
2
Hello Python

การเรียกใช้งาน GOOGLE COLAB

มีบัญชี Google Drive (สมัครได้ที่ <https://myaccount.google.com/?pli=1> และ create account)

การเรียกใช้งาน Google Colab

- เข้า <https://drive.google.com/> ล็อกอินเข้า google drive ก่อน



การเรียกใช้งาน GOOGLE COLAB

การเรียกใช้งาน Google Colab

2. เข้า <https://colab.research.google.com/> เลือก google drive และคลิก new notebook

The screenshot shows the Google Colab interface. At the top, there is a navigation bar with tabs: 'ตัวอย่าง' (Example), 'ล่าสุด' (Recent), 'Google ไดรฟ์' (Google Drive), 'GitHub', and 'อัปโหลด' (Upload). The 'Google ไดรฟ์' tab is currently selected. Below the navigation bar, there is a search bar labeled 'กรองสมุดบันทึก' (Filter notebook) with a magnifying glass icon. Underneath the search bar, there is a table listing three notebooks:

ชื่อ	เจ้าของ	เปิดล่าสุด	แก้ไขล่าสุด	操作
NNFashionMNIST.ipynb	Prapassorn TANTIPHANWADI	24 มี.ค. 2020	24 มี.ค. 2020	
Untitled1.ipynb	Prapassorn TANTIPHANWADI	24 มี.ค. 2020	24 มี.ค. 2020	
Untitled0.ipynb	Prapassorn TANTIPHANWADI	9 มี.ค. 2020	9 มี.ค. 2020	

At the bottom right of the interface, there are two buttons: 'สมุดบันทึกใหม่' (New notebook) and 'ยกเลิก' (Cancel).

การเรียนรู้ใช้งาน GOOGLE COLAB

การเรียนรู้ใช้งาน Google Colab

The screenshot shows the Google Colab interface. At the top, there's a browser-like header with back/forward buttons, a refresh icon, and a URL bar showing the address of the current notebook. Below the header is a navigation bar with links to various notebooks and sections like 'My Drive'.

The main area displays two Jupyter notebooks, both titled 'Untitled2.ipynb'. The first notebook has a status bar indicating it was last modified 'Just now'. The second notebook also has a similar status bar.

On the left side of each notebook, there are standard Jupyter notebook controls: a file menu, a code editor, a search bar, and a cell execution button. The code editor in the bottom-left notebook contains the Python command `print("Hello World!!!")`, which has been executed and displayed in the output cell below it.

To the right of the notebooks is a sidebar titled 'My Drive'. It features a message about blocking people from Drive. Below that are sections for 'Suggested' (showing the same 'Untitled2.ipynb' notebook and a 'Computer - Shortcut.lnk' entry) and 'Folders' (listing 'Classroom' and 'Colab Notebooks').

รูปแบบ PYTHON

6. กฎการตั้งชื่อ สำหรับ Function, Class, Method,...

- ชื่อที่ตั้งต้องขึ้นด้วย ตัวอักษร หรือ Underscore(_) เท่านั้น
- ชื่อที่ตั้ง ถ้ามีมากกว่า 1 คำ ให้ใช้ Underscore(_) เช่น Total_amount
- ชื่อที่ตั้งต้องประกอบไปด้วย ตัวอักษร ตัวเลข หรือ Underscore(_) เท่านั้น
- ชื่อที่ตั้งต้องไม่ซ้ำกับ keywords ในไพธอน
- ชื่อที่ตั้งเป็น Case sensitive
- ชื่อที่ตั้งมีความยาวเท่าไหร่ก็ได้ ไม่จำกัด

7. Keywords คือคำเฉพาะที่ส่วนไว

and	del	for	is	return	while
as	elif	from	lambda	or	with
assert	else	global	none	pass	yield
break	except	If	nonlocal	print	
class	false	import	not	try	
continue	finally	in	raise	true	

รูปแบบ PYTHON

8. ตัวแปร (Variables)

คือ ชื่อที่กำหนดขึ้น เพื่อใช้อ้างอิงข้อมูลบนหน่วยความจำ

ชื่อตัวแปร = ค่าที่ต้องการกำหนดให้ค่าว่าแปร

```
In [3]: # ตัวแปร (Variables)
a = 20
a = b = "Hello"
print(a)
print(b)
```

Hello
Hello

9. การรับและแสดงผลข้อมูล

- รับข้อมูล `input()`
- แสดงผลข้อมูล `print()`

```
In [4]: # การรับและแสดงผลข้อมูล
age = input("กรุณารอกรออายุ: ")
print("อายุ", age, "ปี")
```

กรุณารอกรออายุ: 20
อายุ 20 ปี

รูปแบบ PYTHON

10. ชนิดข้อมูล

- Numeric

1) Integer – จำนวนเต็มบวกหรือลบ

เลขฐานสิบ	เลขฐานสอง	เลขฐานแปด	เลขฐานสิบหก
10	0b1010	0o12	0xA
-11	-0b1011	-0o13	-0xB

2) Floating point numbers - เลขทศนิยม

10.0, 15.33, 9.109×10^9

```
In [1]: a = 10  
        b = 10.0  
        c = 5.5+2.1j
```

```
In [2]: print(type(a))  
        print(type(b))  
        print(type(c))
```

<class 'int'>
<class 'float'>
<class 'complex'>

3) Complex numbers - เชิงซ้อน $a+bj$

รูปแบบ PYTHON

10. ชนิดข้อมูล

- Boolean – ชนิดข้อมูลตรรกะ มี 2 ค่า จริง(True) เท็จ(False)

```
In [7]: e = 3>2  
f = 5<1
```

```
In [8]: print(e, type(e))  
print(f,type(f))
```

True <class 'bool'>
False <class 'bool'>

- String – ชนิดข้อมูลประเภท ข้อความ หรือ กลุ่มตัวอักษร

ถูกครอบด้วยเครื่องหมาย “Hello” หรือ ‘Hello’

ตัวอักษรในสตริงจะมีตำแหน่ง

H	E	L	L	O
---	---	---	---	---

ค่า index แบบบวก 0 1 2 3 4

ค่า index แบบลบ -5 -4 -3 -2 -1

```
In [3]: print("Hello")
```

Hello

```
In [4]: print('Hello')
```

Hello

```
In [5]: len("Hello")
```

Out[5]: 5

```
In [6]: len('Hello')
```

Out[6]: 5

รูปแบบ PYTHON

10. ชนิดข้อมูล

- List – ใช้เก็บข้อมูล ประเภทเดียวกันหรือต่างประเภทกันก็ได้ ข้อมูลสามารถซ้ำกันได้ สามารถเปลี่ยนแปลงหรือแก้ไขข้อมูลได้

- ข้อมูลใน List มีลำดับที่แน่นอนว่าข้อมูลใดอยู่ในตำแหน่งไหนเนื่องจากมีตำแหน่ง index กำกับข้อมูลอยู่โดยข้อมูลจะถูกครอบไว้ใน []

```
In [25]: # List
```

```
In [26]: price = [550, 1350.75, 820, 200]
fruit = ["apple", "orange", "kiwi", "grape", "banana"]
number = [2, 4, 6, 8]
amount = [250, 250, 350, 300]
```

```
In [27]: # กำหนด List ชื่อ List
```

```
student = [{"001": "apple", 20},
           {"002": "cherry", 30},
           {"003": "kiwi", 15}]
```

```
In [35]: len(fruit)
```

```
Out[35]: 5
```

```
In [36]: len(student)
```

```
Out[36]: 3
```

```
In [37]: len([250, 250, 350, 300])
```

```
Out[37]: 4
```

```
In [38]: fruit[0]
```

```
Out[38]: 'apple'
```

```
In [39]: fruit[-1]
```

```
Out[39]: 'banana'
```

- การค้นหาตัวอักษร และกลุ่มอักษรใน List
ค้นหาตัวอักษร ทำได้โดยระบุ index ของ List

```
In [46]: fruit[2:4]
```

```
Out[46]: ['kiwi', 'grape']
```

```
In [42]: fruit[-5:-3]
```

```
Out[42]: ['apple', 'orange']
```

```
fruit = ["apple", "orange", "kiwi", "grape", "banana"]
```

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

```
In [44]: fruit[:5]
```

```
Out[44]: ['apple', 'orange', 'kiwi', 'grape', 'banana']
```

```
In [45]: fruit[-4:]
```

```
Out[45]: ['orange', 'kiwi', 'grape', 'banana']
```

รูปแบบ PYTHON

10. ชนิดข้อมูล

➤ List

- การแก้ไขข้อมูลใน List ทำได้โดยระบุตำแหน่ง index ที่ต้องการแก้ไข
- การเพิ่มข้อมูลใน List: append(ข้อมูลที่ต้องการเพิ่มใน List) , insert(index, ข้อมูลที่ต้องการเพิ่มใน List)
- การลบข้อมูลใน List: list.remove ; del list[index]

```
In [6]: fruit = ["apple", "orange", "kiwi", "grape", "banana"]
In [7]: print(fruit)
['apple', 'orange', 'kiwi', 'grape', 'banana']
In [8]: fruit[0]
Out[8]: 'apple'
In [11]: fruit[0], fruit[4]
('apple', 'banana')
In [13]: fruit[0] = "cherry"
In [14]: print(fruit)
['cherry', 'orange', 'kiwi', 'grape', 'apple']

In [14]: print(fruit)
['cherry', 'orange', 'kiwi', 'grape', 'apple', 'durian']
In [16]: print(fruit)
['cherry', 'orange', 'kiwi', 'grape', 'apple', 'durian']
In [17]: fruit.insert(2, "mango")
In [18]: print(fruit)
['cherry', 'orange', 'mango', 'kiwi', 'grape', 'apple', 'durian']

In [19]: fruit.remove("orange")
In [20]: print(fruit)
['cherry', 'mango', 'kiwi', 'grape', 'apple', 'durian']

In [21]: del fruit[1]
In [22]: print(fruit)
['cherry', 'kiwi', 'grape', 'apple', 'durian']
```

➤ Tuple

- เป็นชนิดข้อมูลที่สามารถเก็บข้อมูลประเภทเดียวกันหรือต่างประเภทได้
- ข้อมูลสามารถซ้ำกันได้
- เมื่อกำหนดข้อมูลลงใน Tuple และไม่สามารถเปลี่ยนแปลงได้
- ข้อมูลมีลำดับแน่นอน เนื่องจากมีตำแหน่ง index กำกับอยู่
- ชนิดข้อมูลรอบด้วย ()

```
In [23]: price_tag = ("$6.57", "$10")
color = ("R", "G", "B")
num = (111, 222)
mix = ("R", "G", 111, 10.25)

In [24]: print(price_tag, color, num, mix)
('$6.57', '$10') ('R', 'G', 'B') (111, 222) ('R', 'G', 111, 10.25)
```

รูปแบบ PYTHON

10. ชนิดข้อมูล

➤ Set

- ชนิดของข้อมูลที่ไม่มีตำแหน่ง index กำกับข้อมูลอยู่
- ข้อมูลไม่มีลำดับที่แน่นอน
- เมื่อสร้าง Set แล้วไม่สามารถแก้ไขข้อมูลได้
- สามารถเพิ่มข้อมูลใหม่ลงไประบบ Set ได้ .add() ; update
- ลบด้วย .remove ; .discard
- ถูกรอบด้วย {}
- หาความยาวด้วย len()
- ค้นหาข้อมูลใน Set ด้วย in()

```
In [50]: days = {"Monday", "Tuesday"}
```

```
In [51]: len(days)
```

```
Out[51]: 2
```

```
In [52]: days.add("Wednesday")
```

```
In [53]: print(days)
```

```
{'Wednesday', 'Monday', 'Tuesday'}
```

```
In [54]: print("Monday" in days)
```

```
True
```

```
In [55]: print("Monday", "Thursday" in days)
```

```
Monday False
```

```
In [56]: days.remove("Wednesday")
```

```
In [57]: print(days)
```

```
{'Monday', 'Tuesday'}
```

```
In [58]: days.discard("Tuesday")
```

```
In [59]: print(days)
```

```
{'Monday'}
```

รูปแบบ PYTHON

10. ชนิดข้อมูล

➤ Dictionary

- ชนิดข้อมูลที่เก็บข้อมูลในรูปแบบ key และ value โดย key กับ value จะสัมพันธ์กัน
- ใช้ key เป็น index ในการเข้าถึง value ซึ่ง key แต่ละตัวจะไม่ซ้ำกัน
- สามารถเปลี่ยนแปลงข้อมูลได้
- ข้อมูลครอบด้วย {}
- รูปแบบ
ชื่อตัวแปร {
 key ตัวที่1 : value ตัวที่1
 key ตัวที่2 : value ตัวที่2}
- หาความยาว len()
- แก้ไขข้อมูล - ระบุตำแหน่ง key = ข้อมูลใหม่
- เพิ่มข้อมูล - ระบุตำแหน่ง key = ข้อมูลที่เพิ่ม
- ลบข้อมูล .pop(key) ; del[key]

```
In [99]: print(Grade_dict)
```

```
{'A': 'Eighty', 'B+': 75}
```

```
In [100]: Grade_dict["A"] = "Excellent"  
Grade_dict["B"] = "Good"
```

```
In [101]: print(Grade_dict)
```

```
{'A': 'Excellent', 'B+': 75, 'B': 'Good'}
```

```
In [102]: Grade_dict["C"] = "Fair"  
Grade_dict["D"] = "Poor"  
Grade_dict["F"] = "Fail"
```

```
In [103]: print(Grade_dict)
```

```
{'A': 'Excellent', 'B+': 75, 'B': 'Good', 'C': 'Fair', 'D': 'Poor', 'F': 'Fail'}
```

```
In [104]: Grade_dict.pop("B+")
```

```
Out[104]: 75
```

```
In [105]: print(Grade_dict)
```

```
{'A': 'Excellent', 'B': 'Good', 'C': 'Fair', 'D': 'Poor', 'F': 'Fail'}
```

```
In [106]: del Grade_dict["F"]
```

```
In [107]: print(Grade_dict)
```

```
{'A': 'Excellent', 'B': 'Good', 'C': 'Fair', 'D': 'Poor'}
```

รูปแบบ PYTHON

➤ ตัวดำเนินการ (Operator)

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic operators)

ตัวดำเนินการประเภทนี้จะใช้กับชนิดข้อมูลตัวเลข (Numeric) โดยแบ่งได้เป็น 7 ประเภท ดังนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
+	บวก	7+2	9
-	ลบ	7-2	5
*	คูณ	7*2	14
/	หาร	7/2	3.5
%	หารเอาเศษ	7%2	1 (เพราะ 7 หาร 2 เหลือเศษ 1)
**	ยกกำลัง	7**2	49
//	หารไม่เอาเศษ	7//2	3 (เพราะ 7 หาร 2 ได้ 3.5 จึงปัดเศษออกเหลือ 3)

In [39]:

```
a = 5<4
b = 10<20
print("a=", a)
print("b=", b)
print("a and b = ", a and b)
print("a or b = ", a or b)
print("not a = ", not a)
print("not b = ", not b)
```

```
a= False
b= True
a and b =  False
a or b =  True
not a =  True
not b =  False
```

ตัวดำเนินการทางตรรกศาสตร์ (Logical operators)

ตัวดำเนินการประเภทนี้จะใช้สำหรับตรวจสอบว่ามีค่าความจริงเป็น จริง (True) หรือ เท็จ (False) โดยแบ่งได้เป็น 3 ประเภท ดังนี้

- ตัวดำเนินการ **and** (และ) จะได้ค่าเป็น True เมื่อค่าอิوبอเระน์ A และ B เป็นจริงทั้งคู่
- ตัวดำเนินการ **or** (หรือ) จะได้ค่าเป็น True เมื่อค่าอิوبอเระน์ A หรือ B ตัวใดตัวหนึ่ง เป็นจริง
- ตัวดำเนินการ **not** (นิเสธ) ถ้าอิوبอเระน์เป็นจริง จะได้ค่าเป็น False แต่ถ้าอิوبอเระน์ เป็นเท็จ จะได้ค่าเป็น True

ผลลัพธ์ของตัวดำเนินการ and, or และ not สามารถสรุปได้ ดังตารางต่อไปนี้

A	B	A and B	A or B	not A	not B
True	True	True	True	False	False
True	False	False	True	False	True
False	True	False	True	True	False
False	False	False	False	True	True

รูปแบบ PYTHON

➤ ตัวดำเนินการ (Operator)

ตัวดำเนินการระดับบิต (Bitwise operators)

ตัวดำเนินการประเภทนี้จะใช้สำหรับเปรียบเทียบบิตของเลขฐานสอง โดยแบ่งได้เป็น 6 ประเภท ดังนี้

- &(and) หมายถึง ถ้าบิตที่นำมาเปรียบเทียบกันมีค่าเป็น 1 เหมือนกัน ผลลัพธ์จะได้ 1
- |(or) หมายถึง ถ้าบิตที่นำมาเปรียบเทียบกันมีบิตใดบิตหนึ่งเป็น 1 ผลลัพธ์จะได้ 1
- ^ (xor) หมายถึง ถ้าบิตที่นำมาเปรียบเทียบกันมีค่าเหมือนกัน ผลลัพธ์จะได้ 0 แต่ถ้าบิตที่นำมาเปรียบเทียบกันมีค่าต่างกัน ผลลัพธ์จะได้ 1 (ง่ายๆ ว่า เมื่อ 0 ต่าง 1)
- ~ (not) หมายถึง ถ้าบิตมีค่าเป็น 0 ผลลัพธ์จะได้ 1 แต่ถ้าบิตมีค่าเป็น 1 ผลลัพธ์จะได้ 0

ผลลัพธ์ของตัวดำเนินการ &, |, ^ และ ~ สามารถสรุปได้ ดังตารางด้านไปนี้

A	B	A & B	A B	~ A	~ B	A^B
1	1	1	1	0	0	0
1	0	0	1	0	1	1
0	1	0	1	1	0	1
0	0	0	0	1	1	0

- << (left shift) หมายถึง การเลื่อนบิตไปทางซ้ายโดยเติมบิต 0 เข้ามาทางด้านขวา เช่น ถ้าเลื่อนบิตออกไปทางซ้าย 2 บิต ก็จะเติมบิต 0 จำนวน 2 บิตเข้ามาทางด้านขวา
- >> (right shift) หมายถึง การเลื่อนบิตไปทางขวาโดยเติมบิต 0 เข้ามาทางด้านซ้าย เช่น ถ้าเลื่อนบิตออกไปทางขวา 2 บิต ก็จะเติมบิต 0 จำนวน 2 บิตเข้ามาทางด้านซ้าย

In [43]: a = 10
b = 2

In [44]: print("a = ", a)
print("b = ", b)
print("a & b = ", a & b)
print("a | b = ", a | b)
print("a ^ b = ", a ^ b)
print("~a = ", ~a)
print("~b = ", ~b)
print("a<<b = ", a<<b)
print("a>>b = ", a>>b)

a = 10
b = 2
a & b = 2
a | b = 10
a ^ b = 8
~a = -11
~b = -3
a<<b = 40
a>>b = 2

รูปแบบ PYTHON

➤ ตัวดำเนินการ (Operator)

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
<code> =</code>	or ก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a = 3$ คือ $a = a 3$ $a = 0100 0011$	0111 (มีค่าเท่ากับเลข 7 ฐานสิบ)
<code>^=</code>	xor ก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a ^= 3$ คือ $a = a ^ 3$ $a = 0100 0011$	0111 (มีค่าเท่ากับเลข 7 ฐานสิบ)
<code><<=</code>	left shift	สมมติ $a = 4$ $a <<= 3$ คือ $a = a << 3$ หมายถึง เลื่อนบิตไปทางซ้ายจำนวน 3 บิต และเดิมบิต 0 เข้ามาทางขวาจำนวน 3 บิต	0010 0000 (มีค่าเท่ากับ 32 ฐานสิบ)
<code>>>=</code>	right shift	สมมติ $a = 4$ $a >>= 3$ คือ $a = a >> 3$ หมายถึง เลื่อนบิตไปทางขวาจำนวน 3 บิต และเดิมบิต 0 เข้ามาทางซ้ายจำนวน 3 บิต	0000

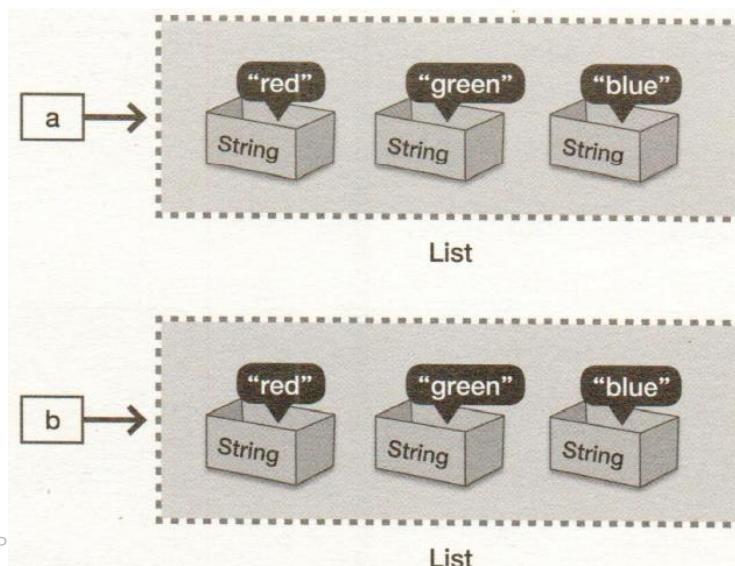
ตัวดำเนินการประเภทนี้ใช้สำหรับกำหนดค่าให้กับตัวแปร ซึ่งแบ่งออกได้เป็นหลายประเภท ดังนี้

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
<code>=</code>	กำหนดค่าให้ตัวแปร	$a = 3$	3
<code>+=</code>	บวกค่าก่อนกำหนดค่าให้ตัวแปร	สมมติ $a = 4$ $a += 3$ คือ $a = a + 3$ $a = 4 + 3$	7
<code>-=</code>	ลบค่าก่อนกำหนดค่าให้ตัวแปร	สมมติ $a = 4$ $a -= 3$ คือ $a = a - 3$ $a = 4 - 3$	1
<code>*=</code>	คูณค่าก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a *= 3$ คือ $a = a * 3$ $a = 4 * 3$	12
<code>/=</code>	หารค่าก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a /= 3$ คือ $a = a / 3$ $a = 4 / 3$	1.3333333333 333333
<code>%=</code>	หารเอาเศษก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a %= 3$ คือ $a = a \% 3$ $a = 4 \% 3$	1
<code>//=</code>	หารไม่เอาเศษก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a // = 3$ คือ $a = a // 3$ $a = 4 // 3$	1
<code>**=</code>	ยกกำลังก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ $a **= 3$ คือ $a = a ** 3$ $a = 4 ** 3$	64
<code>&=</code>	and ก่อนกำหนดค่าให้กับตัวแปร	สมมติ $a = 4$ (เลข 4 ฐานสิบมีค่าเท่ากับ 0100 ฐานสอง) $a &= 3$ คือ $a = a & 3$ (เลข 3 ฐานสิบมีค่าเท่ากับ 0011 ฐานสอง) $a = 0100 \& 0011$	0000 0000 0000

รูปแบบ PYTHON

➤ ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

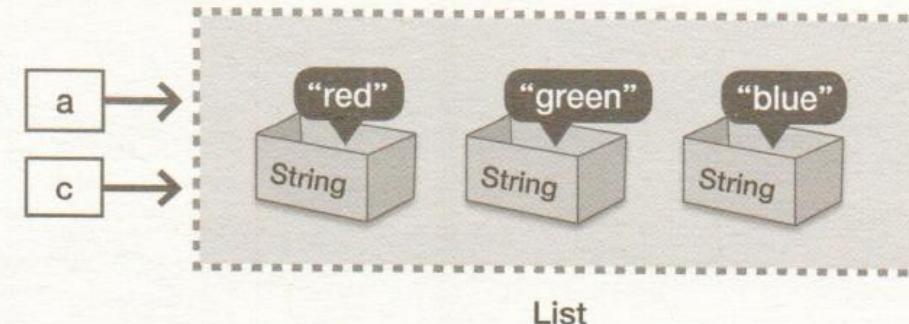
ตัวดำเนินการ	ความหมาย	ตัวอย่าง	ผลลัพธ์
<code>==</code>	เท่ากัน	<code>7==2</code>	<code>False</code>
<code>!=</code>	ไม่เท่ากัน	<code>7!=2</code>	<code>True</code>
<code>></code>	มากกว่า	<code>7>2</code>	<code>True</code>
<code><</code>	น้อยกว่า	<code>7<2</code>	<code>False</code>
<code>>=</code>	มากกว่าหรือเท่ากัน	<code>7>=2</code>	<code>True</code>
<code><=</code>	น้อยกว่าหรือเท่ากัน	<code>7<=2</code>	<code>False</code>



ตัวดำเนินการเอกลักษณ์ (Identity operators)

ตัวดำเนินการประเภทนี้ใช้สำหรับเปรียบเทียบว่าตัวแปรที่ไปยังออบเจ็คเดียวกันหรือไม่ ซึ่งแบ่งได้เป็น 2 ประเภท ดังนี้

- `is` จะให้ผลลัพธ์เป็นจริง (True) เมื่อตัวแปรที่ไปยังออบเจ็คเดียวกัน
- `is not` จะให้ผลลัพธ์เป็นจริง (True) เมื่อตัวแปรที่ไปยังออบเจ็คนั้นๆ ไม่เดียวกัน



List

จากรูป จะพบว่า `a` และ `c` ชี้ไปยังออบเจ็คเดียวกัน ดังนั้น

`a is c` จะให้ผลลัพธ์เป็น True

`a is not c` จะให้ผลลัพธ์เป็น False

In [45]: `a = ["red", "green", "blue"]`
`b = ["red", "green", "blue"]`
`c = a`

In [47]: `print(a is c)`
`print(a is b)`

True
False

จากรูป แม้ว่าตัวแปร `a` และ `b` จะมีค่าข้อมูลเหมือนกัน แต่จะพบว่า `a` และ `b` ชี้ไปยังออบเจ็คคนละตัวกัน ดังนั้น

`a is b` จะให้ผลลัพธ์เป็น False

`a is not b` จะให้ผลลัพธ์เป็น True

In [48]: `print(a is not c)`
`print(a is not b)`

False
True

รูปแบบ PYTHON

➤ ตัวดำเนินการสมาชิก (Membership Operators)

ตัวดำเนินการประเภทนี้ใช้สำหรับตรวจสอบว่ามีข้อมูลที่ระบุอยู่ในออบเจ็คหรือไม่ ซึ่งแบ่งได้เป็น 2 ประเภท ดังนี้

- `in` จะให้ผลลัพธ์เป็นจริง (True) เมื่อพบค่าข้อมูลในออบเจ็ค
- `not in` จะให้ผลลัพธ์เป็นจริง (True) เมื่อไม่พบค่าข้อมูลในออบเจ็ค

In [49]: `a = ["red", "green", "blue"]`

In [50]: `print("red" in a)`

True

In [51]: `print("blue" not in a)`

False

In [52]: `print("yellow" not in a)`

True

รูปแบบ PYTHON

➤ ลำดับความสำคัญของตัวดำเนินการ

ในนิพจน์หนึ่ง ๆ อาจมีตัวดำเนินการมากกว่า 1 ตัว และตัวดำเนินการแต่ละตัวมีลำดับความสำคัญแตกต่างกันออกไปดังตารางต่อไปนี้

ลำดับความสำคัญจากสูงไปต่ำ	ตัวดำเนินการ	ความหมาย
1 (ลำดับความสำคัญสูงสุด)	○	วงเล็บ
2	**	ยกกำลัง
3	~, +, -	ตัวดำเนินการระดับบิต not, เครื่องหมายบวกหน้าตัวเลข (unary plus) เช่น +5, เครื่องหมายลบ หน้าตัวเลข(unary minus) เช่น -5 Note : Unary Operators คือ การ ดำเนินการทางคณิตศาสตร์ที่มีการ กระทำกับໂປໂແຣນ໌ເພື່ອຕັດເຫຼີຍ
4	*, /, %, //	คูณ หาร หารເອເສັ້ນ หารໄມ່ເອເສັ້ນ
5	+, -	บวก ลบ

ลำดับความสำคัญจากสูงไปต่ำ	ตัวดำเนินการ	ความหมาย
6	>>, <<	ตัวดำเนินการระดับบิต right shift, left shift
7	&	ตัวดำเนินการระดับบิต and
8	^,	ตัวดำเนินการระดับบิต xor, or
9	<=, <, >, >=	น้อยกว่าเท่ากับ, น้อยกว่า, มากกว่า, มากกว่าเท่ากับ
10	==, !=	เท่ากับ, ไม่เท่ากับ
11	=, %=, /=, //=, -=, +=, *=, **=	ตัวดำเนินการกำหนดค่า
12	is, is not	ตัวดำเนินการเอกลักษณ์
13	in, not in	ตัวดำเนินการสมาชิก
14 (ลำดับความสำคัญต่ำสุด)	not, or, and	ตัวดำเนินการทางตรรกศาสตร์

รูปแบบ PYTHON

➤ การแปลงชนิดของข้อมูล

Implicit type conversion

คือ python interpreter จะแปลงชนิดของข้อมูลให้อัตโนมัติ โดยชนิดของข้อมูลที่มีความละเอียดต่ำกว่าจะถูกแปลงไปเป็นชนิดข้อมูลที่มีความละเอียดสูงกว่าเสมอ

เช่น ถ้าตัวแปร `a` มีเป็น integers ตัวแปร `b` เป็น floating-point ถ้านำ `a+b` ไปรอนจะทำ implicit type conversion แปลงชนิดข้อมูลก่อน เพราะ `a` กับ `b` มีชนิดข้อมูลคนละประเภทกัน

โดย floating-point มีความละเอียดสูงกว่า integers เพราะเลขทศนิยมเก็บข้อมูลได้ละเอียดกว่าเลขจำนวนเต็ม จึงแปลงชนิดข้อมูลของตัวแปร `a` จาก integers ไปเป็น floating-point แล้วบวกค่าตามปกติ ซึ่งผลลัพธ์ที่ได้จะมีชนิดข้อมูลเป็น floating-point

Explicit type conversion

Explicit type conversion เรียกอีกอย่างว่า type casting คือ ผู้เขียนโปรแกรมทำการแปลงชนิดข้อมูลด้วยตนเอง โดยไม่ทำให้ข้อมูลมีความละเอียดต่ำลง

เราสามารถเรียกใช้ฟังก์ชันต่างๆ ของ Python ในการแปลงชนิดข้อมูล ดังนี้

```
In [53]: a = 5  
b = 20.25  
c = a+b
```

```
In [55]: print(a, type(a))  
print(b, type(b))  
print(c, type(c))
```

```
5 <class 'int'>  
20.25 <class 'float'>  
25.25 <class 'float'>
```

รูปแบบ PYTHON

➤ การแปลงชนิดของข้อมูล → Explicit Type Conversion

ฟังก์ชัน	ความหมาย
int(x,base)	<p>แปลงค่าข้อมูลเป็นเลขจำนวนเต็ม x คือ ข้อมูลที่ต้องการแปลง base คือ เลขฐาน จะระบุเป็นฐานสอง ฐานสิบ ฐานแปด หรือฐานสิบหกก็ได้ ถ้าไม่ระบุจะหมายถึง ฐานสิบ)</p> <p>ตัวอย่าง :</p> <pre>int(20.75) แปลงเลขจำนวนจริง 20.75 เป็นเลขจำนวนเต็ม ผลลัพธ์ คือ 20</pre> <pre>int("1010") แปลงสตริง "1010" เป็นเลขจำนวนเต็ม ผลลัพธ์ คือ 1010</pre> <pre>int("1010",2) แปลงสตริง "1010" ฐานสองเป็นเลขจำนวนเต็ม ผลลัพธ์ คือ 10 (ฐานสิบ)</pre>
float(x)	<p>แปลงค่าข้อมูลเป็นเลขจำนวนจริง x คือ ข้อมูลที่ต้องการแปลง</p> <p>ตัวอย่าง :</p> <pre>float("25.75") แปลงสตริง "25.75" เป็นเลขจำนวนจริง ผลลัพธ์ คือ 25.75</pre> <pre>float(25) แปลงเลขจำนวนเต็ม 25 เป็นเลขจำนวนจริง ผลลัพธ์ คือ 25.0</pre> <pre>float(0b1010) แปลงเลขจำนวนเต็มฐานสอง 1010 เป็นเลข จำนวนจริง ผลลัพธ์ คือ 10.0</pre>
complex(real,imaginary)	<p>แปลงค่าข้อมูลเป็นเลขจำนวนเชิงซ้อน real คือ จำนวนจริง imaginary คือ จำนวนจินตภาพ</p> <p>ตัวอย่าง :</p> <pre>complex(2) แปลงเลขจำนวนเต็มเป็นเลขจำนวนเชิงซ้อน ผลลัพธ์ คือ (2+0j)</pre> <pre>complex(2,3) แปลงเลขจำนวนเต็มเป็นเลขจำนวนเชิงซ้อน ผลลัพธ์ คือ (2+3j)</pre> <pre>complex('1+2j') แปลงสตริงเป็นเลขจำนวนเชิงซ้อน ผลลัพธ์ คือ (1+2j)</pre>
str(x)	<p>แปลงค่าข้อมูลเป็นสตริง x คือ ข้อมูลที่ต้องการแปลง</p> <p>ตัวอย่าง :</p> <pre>str(20) แปลงเลขจำนวนเต็ม 20 เป็นสตริง ผลลัพธ์ คือ "20"</pre> <pre>str(20.25) แปลงเลขจำนวนจริง 20.25 เป็นสตริง ผลลัพธ์ คือ "20.25"</pre>

ฟังก์ชัน	ความหมาย
complex(real,imaginary)	<p>แปลงค่าข้อมูลเป็นเลขจำนวนเชิงซ้อน real คือ จำนวนจริง imaginary คือ จำนวนจินตภาพ</p> <p>ตัวอย่าง :</p> <pre>complex(2) แปลงเลขจำนวนเต็มเป็นเลขจำนวนเชิงซ้อน ผลลัพธ์ คือ (2+0j)</pre> <pre>complex(2,3) แปลงเลขจำนวนเต็มเป็นเลขจำนวนเชิงซ้อน ผลลัพธ์ คือ (2+3j)</pre> <pre>complex('1+2j') แปลงสตริงเป็นเลขจำนวนเชิงซ้อน ผลลัพธ์ คือ (1+2j)</pre>
str(x)	<p>แปลงค่าข้อมูลเป็นสตริง x คือ ข้อมูลที่ต้องการแปลง</p> <p>ตัวอย่าง :</p> <pre>str(20) แปลงเลขจำนวนเต็ม 20 เป็นสตริง ผลลัพธ์ คือ "20"</pre> <pre>str(20.25) แปลงเลขจำนวนจริง 20.25 เป็นสตริง ผลลัพธ์ คือ "20.25"</pre>

รูปแบบ PYTHON

➤ การแปลงชนิดของข้อมูล → Explicit Type Conversion

ฟังก์ชัน	ความหมาย	ฟังก์ชัน	ความหมาย
set(x)	<p>แปลงค่าข้อมูลเป็นเซ็ต</p> <p>x คือ ข้อมูลที่ต้องการแปลง จะเป็นสตริง ลิสต์ หรือ ทุกเพล็กซ์ ก็ได้ แต่ไม่สามารถกำหนดค่าเป็นเลขจำนวนเต็ม หรือเลขจำนวนจริงได้</p> <p>ตัวอย่าง :</p> <pre>set("abcde") แปลงสตริง "abcde" เป็นเซ็ต ผลลัพธ์ คือ {'d', 'b', 'a', 'e', 'c'}</pre> <pre>set([2,4,6,8]) แปลงลิสต์ [2,4,6,8] เป็นเซ็ต ผลลัพธ์ คือ {8, 2, 4, 6}</pre> <pre>set(("red","green","blue","pink")) แปลงทุกเพล็กซ์ ("red","green","blue","pink") เป็นเซ็ต ผลลัพธ์ คือ {'green', 'pink', 'blue', 'red'}</pre>	list(x)	<p>แปลงค่าข้อมูลเป็นลิสต์</p> <p>x คือ ข้อมูลที่ต้องการแปลง จะเป็นสตริง เซ็ต หรือทุกเพล็กซ์ ก็ได้ แต่ไม่สามารถกำหนดค่าเป็นเลขจำนวนเต็มหรือ เลขจำนวนจริงได้</p> <p>ตัวอย่าง :</p> <pre>list("python") แปลงสตริง "python" เป็นลิสต์ ผลลัพธ์ คือ ['p', 'y', 't', 'h', 'o', 'n']</pre> <pre>list(("car","bicycle","truck")) แปลงทุกเพล็กซ์ ("car","bicycle","truck") เป็นลิสต์ ผลลัพธ์ คือ ['car', 'bicycle', 'truck']</pre> <pre>list({"abc","def","ghi"}) แปลงเซ็ต {"abc","def","ghi"} เป็นลิสต์ ผลลัพธ์ คือ ['def', 'abc', 'ghi']</pre>
tuple(x)	<p>แปลงค่าข้อมูลเป็นทุกเพล็กซ์</p> <p>x คือ ข้อมูลที่ต้องการแปลง จะเป็นสตริง เซ็ต หรือลิสต์ ก็ได้ แต่ไม่สามารถกำหนดค่าเป็นเลขจำนวนเต็มหรือเลขจำนวน จริงได้</p> <p>ตัวอย่าง :</p> <pre>tuple("Hello") แปลงสตริง "Hello" เป็นทุกเพล็กซ์ ผลลัพธ์ คือ ('H', 'e', 'l', 'l', 'o')</pre> <pre>tuple([10.5,20.5,30.5]) แปลงลิสต์ [10.5,20.5,30.5] เป็นทุกเพล็กซ์ ผลลัพธ์ คือ (10.5, 20.5, 30.5)</pre> <pre>tuple({"tennis","baseball","basketball"}) แปลงเซ็ต {"tennis","baseball","basketball"} เป็นทุกเพล็กซ์ ผลลัพธ์ คือ ('baseball', 'basketball', 'tennis')</pre>	dict(x)	<p>แปลงค่าข้อมูลเป็นติกขันนาวี</p> <p>x คือ ข้อมูลที่ต้องการแปลงเป็นติกขันนาวี จะเป็นลิสต์ หรือทุกเพล็กซ์ ก็ได้ แต่ไม่สามารถกำหนดเป็นเลขจำนวนเต็ม เลขจำนวนจริง สตริงหรือเซ็ตได้</p> <p>ตัวอย่าง :</p> <pre>dict([[1,123],[2,124]]) แปลงลิสต์เป็นติกขันนาวี ผลลัพธ์ คือ {1: 123, 2: 124}</pre> <pre>dict(({"1","red"}, {"2", "green"}, {"3", "blue"})) แปลงทุกเพล็กซ์เป็นติกขันนาวี ผลลัพธ์ คือ {'1': 'red', '2': 'green', '3': 'blue'}</pre>

รู้แบบ PYTHON

➤ การแปลงชนิดของข้อมูล → Explicit Type Conversion

In [60]:

```
cake_price = input("Cake price: ")  
cookie_price = input("Cookie price: ")  
total_price = float(cake_price) + float(cookie_price)
```

Cake price: 80.50
Cookie price: 75.75

In [61]:

```
print("Cake price", cake_price, "บาท")  
print("Cookie price", cookie_price, "บาท")  
print("Total price", total_price, "บาท")
```

Cake price 80.50 บาท
Cookie price 75.75 บาท
Total price 156.25 บาท

HOMEWORK

Numeric

- 1) เขียนโปรแกรมสร้างตัวแปรชื่อ numeric1 และกำหนดค่าให้เท่ากับ 5 หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของ numeric1 ออกรมา
- 2) เขียนโปรแกรมรับอินพุต 1 ตัวที่เป็นจำนวนเต็ม หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของจำนวนที่รับอกรมา
- 3) เขียนโปรแกรมสร้างตัวแปรชื่อ numeric1 และกำหนดค่าให้เท่ากับ 5.0 หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของ numeric1 ออกรมา
- 4) เขียนโปรแกรมสร้างตัวแปรชื่อ numeric2 และกำหนดค่าให้เท่ากับ -3.1 หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของ numeric2 ออกรมา
- 5) เขียนโปรแกรมสร้างตัวแปรชื่อ numeric3 และกำหนดค่าให้เท่ากับ 0.0 หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของ numeric3 ออกรมา
- 6) เขียนโปรแกรมรับอินพุต 1 ตัวที่เป็นจำนวนจริง หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของจำนวนที่รับอกรมา
- 7) เขียนโปรแกรมรับอินพุต 2 ตัวที่เป็นจำนวนเต็ม(int1, int2) ให้คำนวณผลบวก ผลลบ ผลคูณ และผลหารของจำนวนที่รับมา หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของทุกผลลัพธ์อกรมา
- 8) เขียนโปรแกรมรับอินพุต 2 ตัวที่เป็นจำนวนจริง(float1, float2) ให้คำนวณผลบวก ผลลบ ผลคูณ และผลหารของจำนวนที่รับมา หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดข้อมูลของทุกผลลัพธ์อกรมา
- 9) เขียนโปรแกรมสร้างตัวแปรชื่อ logic1 และกำหนดค่าให้เท่ากับ True หลังจากนั้นให้ทำการพิมพ์ค่าและชนิดของข้อมูลของ logic1 ออกรมา
- 10) กำหนดให้ logic1=True และ logic2=False เขียนโปรแกรมเพื่อนำค่าความจริงทั้งสองมาดำเนินการด้วยคำสั่ง “และ” หลังจากนั้นให้ทำการพิมพ์ค่าความจริงที่ได้จากการดำเนินการอกรมา

HOMEWORK

String

- 11) เขียนโปรแกรมสร้างตัวแปรชื่อ string1 และกำหนดค่าให้เท่ากับ 'Python' หลังจากนั้นให้ทำการพิมพ์อักษรที่ index เป็น 0, 1, -1, -2 ออกมานะ
- 12) เขียนโปรแกรมสร้างตัวแปรชื่อ string1 และกำหนดค่าให้เท่ากับ 'Python' หลังจากนั้นให้ทำการพิมพ์อักษร 't', 'h' ออกมานะ โดยอ้างอิงตำแหน่งจาก string1
- 13) เขียนโปรแกรมสร้างตัวแปรชื่อ string1 และกำหนดค่าให้เท่ากับ 'Python' หลังจากนั้นให้ทำการพิมพ์อักษร
- ตั้งแต่ index ที่ 1 จนถึง index ที่ 3
 - ตั้งแต่ index ที่ 2 จนถึง index ที่ -2
 - ตั้งแต่ index ที่ -4 จนถึง index ที่ -2
- 14) เขียนโปรแกรมรับอินพุต 2 ตัวที่เป็นสายอักขระ (string1, string2) ให้ทำการตรวจสอบ string1 ที่รับมาว่ามี string2 ในนั้นหรือไม่ หากนั้นให้พิมพ์ค่าความจริงออกมา
- 15) เขียนโปรแกรมรับอินพุต 1 ตัวที่เป็นสายอักขระ (sentence) ให้ทำการตัดคำในสายอักขระโดยใช้ ' ' เป็นตัวตัดคำ หลังจากนั้นให้พิมพ์ผลลัพธ์จากการตัดคำออกมา

Example: sentence = 'We love Python'
จะได้ output = ['We', 'love', 'Python']

- 16) เขียนโปรแกรมรับอินพุต 2 ตัว ที่เป็นสายอักขระ (str1, str2) ให้ทำการต่อสายอักขระทั้งสอง (โดยนำ str2 ไปต่อหลัง str1) หลังจากนั้นให้พิมพ์ผลลัพธ์ออกมา

Example: str1 = 'Fri', str2 = 'day'
จะได้ output = Friday

HOMEWORK

List

17) เขียนโปรแกรมสร้าง list ที่มีสมาชิกเป็น 0, 1, 2, 'a', 'b', 'c' ตามลำดับ จากนั้น

- พิมพ์ค่าและชนิดข้อมูลของ list ที่สร้างอุ่นๆ
- กำหนดให้ `list1 = [0, 1, 2, 'a', 'b', 'c']` เขียนโปรแกรมพิมพ์สมาชิกที่ index เป็น 2, -3
- เขียนโปรแกรมพิมพ์สมาชิกที่ index เป็น 2, -3, 0, 'b' (โดยอ้างอิงจาก list1)
- เขียนโปรแกรมพิมพ์สมาชิก ตั้งแต่ index ที่ 2 จนถึง index ที่ 3 ; ตั้งแต่ index ที่ 1 จนถึง index ที่ -2 (โดยอ้างอิงจาก list1)
- เขียนโปรแกรมพิมพ์ [1, 2, 'a', 'b'] (โดยอ้างอิงจาก list1)

18) กำหนดให้ `list2 = ['ant', 'bird', 'cat', 'dog', 'eagle']` พิมพ์ค่าและชนิดข้อมูลของ list ที่สร้างอุ่นๆ

- เขียนโปรแกรมเพื่อเปลี่ยนค่าสมาชิกใน index ที่ 3 เป็น 'duck' และ เพิ่ม 'fish' เป็นตัวสุดท้าย หลังจากนั้นให้พิมพ์ list2 อุ่นๆ
- เขียนโปรแกรมสร้าง empty list และรับอินพุต 3 ตัวที่เป็นสายอักขระ (str1, str2, str3) จากนั้นให้ทำการเพิ่มสายอักขระทั้ง 3 ไป เป็นสมาชิกใน list โดยใช้ append จากนั้นให้พิมพ์ list อุ่นๆ

19) กำหนดให้ `list4 = [0, 4, 3, 2, 1]`

- เขียนโปรแกรมเพื่อเรียงลำดับข้อมูลใน list4 จากน้อยไปมาก หลังจากนั้นให้พิมพ์ list4 อุ่นๆ และเก็บค่าไว้ในตัวแปรชื่อ `sorted_list4` จากนั้นให้พิมพ์ sorted_list4 อุ่นๆ
- เขียนโปรแกรมเพื่อเรียงลำดับข้อมูลใน list4 มากไปน้อย หลังจากนั้นให้พิมพ์ list4 อุ่นๆ และเก็บค่าไว้ในตัวแปรชื่อ `sorted_list4` จากนั้นให้พิมพ์ sorted_list4 อุ่นๆ

HOMEWORK

List

20) กำหนดให้ `list5 = ['d', 'a', 'c', 'b', 'e']`

- เขียนโปรแกรมเพื่อเรียงลำดับข้อมูลใน `list5` ตามตัวอักษร หลังจากนั้นให้พิมพ์ `list5` ออกรมา
- เขียนโปรแกรมเพื่อเรียงลำดับข้อมูลใน `list5` ตามตัวอักษรแบบถอยหลัง และเก็บค่าไว้ในตัวแปรชื่อ `sorted_list5` จากนั้นให้พิมพ์ `sorted_list5` ออกรมา

21) กำหนดให้ `list1= ['a', 'b', 'c']` และ `list2= [0,1, 2]`

- เขียนโปรแกรมเพื่อนำสมาชิกทั้งหมดของ `list2` ไปต่อท้าย `list1` หลังจากนั้นให้พิมพ์ `list1` และ `list2` ออกรมา
- เขียนโปรแกรมเพื่อนำสมาชิกทั้งหมดของ `list2` ไปต่อท้าย `list1` และเก็บ `list` ที่ได้จากการดำเนินการไว้ในตัวแปรชื่อ `list3` หลังจากนั้นให้พิมพ์ `list1`, `list2` และ `list3` ออกรมา

22) กำหนดให้ `list1= [1, 2, 3, 1, 2, 3]`

- เขียนโปรแกรมลบสมาชิกใน `index` ที่ 1 ออกรจาก `list1` และพิมพ์ `list1` ออกรมา (ใช้ `del` และ `remove`)
- เขียนโปรแกรมลบ 3 ทั้งหมด ออกรจาก `list1` และพิมพ์ `list1` ออกรมา (ใช้ `del` และ `remove`)
- เขียนโปรแกรมลบสมาชิกทุกตัว และพิมพ์ `list1` ออกรมา
- กำหนดให้ `list1= [1, 2, 3, 'a', 'b', 'c']` เขียนโปรแกรมหาความยาวของ `list1` และพิมพ์อกรมา

HOMEWORK

Tuple

23) กำหนดให้ tuple1 = ('America', 'Brazil', 'China', 'Dominican', 'Egypt')

- เขียนโปรแกรมเพื่อพิมพ์ค่า ชนิดข้อมูล และหาความยาวของข้อมูล พิมพ์ออกมา
- เขียนโปรแกรมเพื่อพิมพ์สมาชิก index ที่ 1, -2,
- เขียนโปรแกรมเพื่อพิมพ์ 'China', 'Egypt'

24) กำหนดให้ tuple2 = ('One', 'Two', 'Three', 1, 2, 3)

- เขียนโปรแกรมเพื่อพิมพ์ tuple และหาความยาว และพิมพ์ออกมา
- เขียนโปรแกรมเพื่อพิมพ์ tuple ที่มีสมาชิกตั้งแต่ index ที่ 1 จนถึง index ที่ 3 พิมพ์ออกมา
- เขียนโปรแกรมเพื่อพิมพ์ tuple ที่มีสมาชิกตั้งแต่ index ที่ 2 จนถึง index ที่ -2 พิมพ์ออกมา
- เขียนโปรแกรมเพื่อพิมพ์ ('three', 1) พิมพ์ออกมา

HOMEWORK

Dictionary

24) เขียนโปรแกรมสร้าง dictionary

- เขียนโปรแกรมสร้าง dictionary ที่มี key เป็น 'first_name' และ 'last_name' โดยมี value เป็น 'John' และ 'Doe' ตามลำดับ จากนั้นให้ทำการพิมพ์ค่าและชนิดของข้อมูล dictionary ที่สร้างออกมา
- กำหนดให้ Dict1 = {'first_name' : 'John', 'last_name' : 'Doe'} เขียนโปรแกรมพิมพ์ค่า value ของ key 'first_name' ออกมานะ ; key 'last_name' ออกมานะ
- เขียนโปรแกรมพิมพ์ list ของ key ทั้งหมด ออกมานะ
- เขียนโปรแกรมพิมพ์ list ของ value ทั้งหมด ออกมานะ
- เขียนโปรแกรมเปลี่ยน value ของ key 'first_name' เป็น 'Jane' พิมพ์ dict1 ออกมานะ
- เขียนโปรแกรมสร้าง key ใหม่ชื่อ 'Age' และกำหนด value เป็น 32 พิมพ์ dict1 ออกมานะ
- เขียนโปรแกรมสร้าง key ใหม่ชื่อ 'Age' และ 'Hobby' กำหนด value เป็น 32 และ ['coding', 'studying'] ตามลำดับ พิมพ์ dict1 ออกมานะ
- เขียนโปรแกรมสร้าง empty dictionary และรับอินพุต 2 ตัวที่เป็นสายอักขระ (str1, str2) โดยให้ str1 เป็น key และ str2 เป็น value พิมพ์ dict1 ออกมานะ
- กำหนดให้ Dict1 = {'first_name' : 'John', 'last_name' : 'Doe', 'Age' : 32} เขียนโปรแกรมลบ key 'Age' พิมพ์ dict1 ออกมานะ
- เขียนโปรแกรมพิมพ์จำนวน key พิมพ์ dict1 ออกมานะ
- เขียนโปรแกรมลบข้อมูลทั้งหมด พิมพ์ dict1 ออกมานะ
- กำหนดให้ Dict1 = {'first_name' : 'John', 'last_name' : 'Doe', 'Age' : 32} เขียนโปรแกรมรับอินพุต 1 ตัวที่เป็นสายอักขระ (str1) ให้ทำการตรวจสอบว่า str1 เป็นหนึ่งใน value ของ Dict1 หรือไม่ ให้พิมพ์ค่าความจริงออกมานะ

HOMEWORK

Set

24) เขียนโปรแกรมสร้าง set

- เขียนโปรแกรมสร้าง set ที่มีสมาชิก เป็น 1, 2, 3, 'i', 'ii', 'iii' จากนั้นให้ทำการพิมพ์ค่าและชนิดของข้อมูล set ที่สร้างออกมา
- กำหนดให้ `set1 = {1, 2, 3, 'i', 'ii', 'iii'}` เขียนโปรแกรมพิมพ์สมาชิกทุกตัว ออกมา
- เขียนโปรแกรมเพิ่ม 'iv' และ 1 พิมพ์ `set1` ออกมา
- เขียนโปรแกรมพิมพ์ list ของ value ทั้งหมด ออกมา
- เขียนโปรแกรมสร้าง empty set และรับอินพุต 3 ตัวที่เป็นสายอักขระ (`str1, str2, str3`) ไปเป็นสมาชิกใน set สุดท้าย พิมพ์ set ออกมา
- เขียนโปรแกรมสร้าง empty set และรับอินพุต 3 ตัวที่เป็นสายอักขระ (`str1, str2, str3`) ไปเป็นสมาชิกใน set โดยใช้ method พิมพ์ `set` ออกมา
- กำหนดให้ `set1 = {1, 2, 3, 'i', 'ii', 'iii'}` เขียนโปรแกรมรับอินพุต 1 ตัวที่เป็นจำนวนเต็ม (`int1`) ให้ทำการตรวจสอบว่า `int1` เป็นสมาชิกของ `set1` หรือไม่ จากนั้นให้พิมพ์ค่าความจริงออกมา

HOMEWORK

Set

25) กำหนดให้ $\text{set1} = \{1, 2, 3, 4, 5\}$ และ $\text{set2} = \{3, 4, 5, 6\}$

- เขียนโปรแกรมพิมพ์ผลลัพธ์ที่เกิดจากการ union ระหว่าง set1 และ set2 ออกมานา
- เขียนโปรแกรมพิมพ์ผลลัพธ์ที่เกิดจากการ intersect ระหว่าง set1 และ set2 ออกมานา
- เขียนโปรแกรมพิมพ์ผลลัพธ์ที่เกิดจาก set1 - set2 ออกมานา
- เขียนโปรแกรมพิมพ์ผลลัพธ์ที่เกิดจาก set2 – set1 ออกมานา
- เขียนโปรแกรมพิมพ์ผลลัพธ์ที่เกิดจาก symmetric difference ระหว่าง set1 และ set2 ออกมานา

THE END