



# รายวิชา การเขียนโปรแกรมเชิงวัตถุ OOP(4122309)

บทที่ 3 แนะนำภาษาไพธอนและโปรแกรม VS-Code เบื้องต้น



โดย ผู้ช่วยศาสตราจารย์ ดร. นิตฺตพงษ์ ส่งเนียม  
สาขาวิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏพระนคร

# Agenda

- 3.1 แนะนำภาษาไพธอน(Python)
- 3.2 เครื่องมือที่ใช้ในการเขียนโปรแกรมภาษาไพธอน
- 3.3 วิธีการติดตั้ง VS-Code
- 3.4 วิธีการติดตั้ง Python
- 3.5 การใช้งาน VS-Code เบื้องต้น
- 3.6 วิทยากรณ์ทางภาษาไพธอนเบื้องต้น
- 3.7 การใช้งาน Google Colab เบื้องต้น



ไพธอน(Python) เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษา Python นั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 Python นั้นเป็นภาษาแบบ interprete ที่ถูกออกแบบโดยมีปรัชญาที่จะทำให้โค้ดอ่านได้ง่ายขึ้น และโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่

## ประวัติของภาษาโปรแกรม Python

สำหรับประวัติของภาษาโปรแกรม Python ได้เริ่มต้นขึ้นในเดือนธันวาคมปี 1989 โดยนาย Guido van Rossum โปรแกรมเมอร์ชาวดัตช์ ในตอนนั้นทำงานอยู่ที่สถาบันวิจัยแห่งชาติ Centrum Wiskunde & Informatica (CWI) ซึ่งเป็นสถาบันวิจัยทางด้านคณิตศาสตร์และวิทยาการคอมพิวเตอร์ในเมืองอัมสเตอร์ดัม ประเทศเนเธอร์แลนด์ ในเวลานั้น Guido ต้องพัฒนาโปรแกรมสำหรับผู้ดูแลระบบ เพื่อใช้ในโครงการ Amoeba ซึ่งเป็นโครงการเกี่ยวกับระบบปฏิบัติการแบบกระจาย (Distributed operating system) อย่างไรก็ตามเขารู้สึกว่าภาษาโปรแกรม ABC, C และ Bourne shell มีข้อจำกัดมากมาย ทั้งเรื่องใช้เวลาในการพัฒนานานมากและไม่สามารถตอบโจทย์หลายประการ ดังนั้น Guido จึงได้ตัดสินใจเริ่มพัฒนาภาษาโปรแกรมระดับสูงขึ้นมาใหม่เพื่อใช้งานเองเป็นงานอดิเรก โดยนำเอาสิ่งที่ชอบในภาษา ABC มาพัฒนาลงไปเป็นภาษาโปรแกรม Python รวมถึงได้พัฒนาส่วนอื่น ๆ เพิ่มเติมเข้าไป และในเวลาต่อมาจึงได้เผยแพร่ Python 1.0 เวอร์ชันแรกในปี 1994 หากเทียบกับภาษา Java ที่ได้ทำการเผยแพร่เวอร์ชันแรกในปี 1996 จะเห็นได้ว่าภาษา Python มีอายุมากกว่าภาษา Java ถึง 2 ปี



## ประวัติของภาษาโปรแกรม Python (ต่อ)

สำหรับที่มาของชื่อภาษาโปรแกรม Python นั้นไม่ได้มีที่มาเกี่ยวข้องกับงูเหมือนกับชื่อของมันแต่อย่างใด แต่ในช่วงที่ตัดสินใจเลือกชื่อนั้น ชื่อแรกที่เข้ามาในความคิดของ Guido ก็คือ มอนตี ไพธอน: ละครสัตว์เหินหวา (Monty Python's Flying Circus) ซึ่งเป็นชื่อรายการโทรทัศน์ทางช่อง BBC แนวตลกชื่อดังจากฝั่งอังกฤษที่เขาชื่นชอบมาก ๆ โดยเขาให้เหตุผลว่า “Python” เป็นชื่อที่สั้น จำได้ง่าย ฉีกแนวนิดๆ และดูลึกลับ ในตอนนั้นโดยทั่วไปมักจะนิยมเอาชื่อของบุคคลที่มีชื่อเสียงมาใช้เป็นชื่อภาษาโปรแกรมคอมพิวเตอร์ เช่น Ada, Pascal และ Eiffel ถึงแม้ว่าทีมนักแสดงในรายการจะไม่ได้มีชื่อเสียงทางด้านวิทยาศาสตร์และเทคโนโลยี แต่ก็เป็นที่ชื่นชอบในกลุ่มชาว Geek อย่างมาก รวมถึงกลุ่มคนที่ทำงานใน CWI ก็มักจะนิยมเอาชื่อรายการทีวีโชว์มาตั้งชื่อในงานของตัวเองอีกด้วย นี่คือเหตุผลที่นำไปของชื่อภาษา Python นอกจากนั้น Guido ยังใช้ชื่อของนักแสดงตลกชาวอังกฤษชื่อดังและเป็นหนึ่งในสมาชิกผู้ก่อตั้งทีม Monty Python ที่ชื่อ Eric Idle มาใช้เป็นชื่อ IDE หรือเครื่องมือที่ใช้ในการพัฒนาโปรแกรมว่า “IDLE” อีกด้วย

## ตัวแปลภาษาของไพธอน (Python interpreter)

Python interpreter เป็นตัวแปลภาษาของภาษา Python เพื่อให้สามารถรันโค้ด Python ได้ ซึ่งได้มากับไลบรารีมาตรฐานที่สามารถใช้งานได้ฟรี ซึ่งดาวน์โหลดได้ที่ <https://www.python.org/> มันเป็นโปรแกรมแบบ Open source และ binary ที่มีให้ดาวน์โหลดในแพลตฟอร์มที่ได้รับความนิยม นอกจากนี้ interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดภาษา Python และเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที นอกจากนี้ Python interpreter ยังสามารถนำมาเขียนร่วมกับฟังก์ชันใหม่ที่ถูกพัฒนามาจากภาษา C และ C++ เพื่อเพิ่มความสามารถการทำงานของภาษา Python และเหมาะสำหรับเป็นภาษาในการสร้าง Extension และแอปพลิเคชันที่ปรับแต่งได้

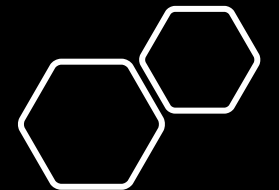


python

TM

3.2

เครื่องมือที่ใช้ในการเขียนภาษา Python

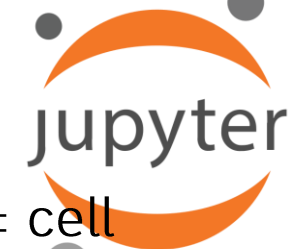




การเขียนโปรแกรมภาษา Python มีเครื่องมือที่ใช้ในการเขียนโปรแกรมหลากหลายเครื่องมือ การทำงานของแต่ละเครื่องมือก็จะแตกต่างกันไป มีทั้งลักษณะการ Run โปรแกรมทีละบรรทัด หรือการ Run โปรแกรมทั้งโปรแกรม

ลักษณะของเครื่องมือที่ใช้ในการเขียน Python นั้นมีทั้งเครื่องมือที่ทำงานบนเครื่องคอมพิวเตอร์ และเครื่องมือที่ทำงานบน Server





## Jupyter Notebook

Jupyter Notebook คือ หน้าเว็บที่ประกอบด้วย ช่อง ๆ cell เรียงต่อกันลงไป โดยแต่ละ cell สามารถเป็นเนื้อหา static content ต่าง ๆ เช่น ข้อความ รูปภาพ กราฟ วิดีโอ เสียง หรือ เป็นโค้ดโปรแกรม คอมพิวเตอร์ ภาษา Python ที่สามารถรันคำสั่งประมวลผล แสดงผลลัพธ์ออกมาได้จริง ๆ

ซึ่งโดยปกติการเขียนโปรแกรมทั่วไป เราจะมองไฟล์ Source Code 1 ไฟล์ เป็นหน่วยเดียว เราไม่สามารถแยกสั่งรันบางบรรทัดได้ง่าย ๆ และเรามักจะแยก Source Code กับเอกสารออกจากกัน หรือบางทีก็จะใส่เป็นข้อความง่าย ๆ ไม่มีรูปประกอบไม่มีตัวหนาตัวเอียง ไว้ใน Comment ตอนต้นของไฟล์ หรือต้นฟังก์ชัน

แต่ Jupyter Notebook เอาทั้งสองอย่างมาใส่ cell เรียงไว้ในหน้าเว็บเดียวกัน เราสามารถเข้าถึงหน้าเว็บนี้โดยไม่ต้องติดตั้งโปรแกรม แค่มีเว็บเบราว์เซอร์ เราสามารถแก้ไขโค้ด สั่งรัน cell เห็นผลลัพธ์ได้ cell หนึ่งที่ เราสามารถสั่งรัน cell ไหน ข้ามไปมา แก้ไปแก้มา ได้อย่างอิสระ และการแสดงผลใน cell ที่เป็น content ก็สามารถทำได้ rich กว่าแค่ข้อความ text ใน comment สามารถใส่รูปประกอบ คลิปวิดีโอ ใส่ ลิงค์อ้างอิง ใส่สมการคณิตศาสตร์ ใส่กราฟ/ตารางเปรียบเทียบ, etc.

## Google Colab

Google colab เป็นโฮสต์โปรแกรม Jupyter notebook บน Cloud ของ Google ชื่อเต็ม คือ Google Colaboratory โดยใช้ภาษา python3 เป็นภาษาหลักที่ใช้ในการเขียน และรันงานบน colab นี้

**\*\*\*Google colab มีข้อดีคือไม่จำเป็นต้องติดตั้ง Library ก็สามารถใช้งาน Library ได้**

ทำไมถึงสร้าง Machine Learning บน Google Colab การสร้าง Model จะต้องใช้คอมพิวเตอร์ที่มีความเร็วหรือ ประสิทธิภาพสูง เพื่อลดระยะเวลาในการ ประมวลผลของเครื่อง

Google Colab มีความเร็วหลายเท่าถ้าเทียบกับคอมพิวเตอร์ที่ใช้ทั่วไป ตัวอย่างเช่น

คอมพิวเตอร์ทั่วไป = การ Train Model 1 ครั้ง ใช้เวลาประมาณ 2-3 ชั่วโมง

Google Colab = การ Train Model 1 ครั้ง ใช้เวลาประมาณ 30-35 นาที

และ การ Train Model จะต้อง ทำหลายร้อยครั้ง เท่ากับการที่จะได้ Model ที่มีประสิทธิภาพจะต้องใช้เวลานานมาก

**\*\*\*Google Colab มีข้อดี คือ มีความเร็วของ CPU และ GPU ใช้ได้ฟรี สามารถเชื่อมต่อกับ Google Drive ได้รองรับ Tensorflow**

ข้อเสียคือ อาจจะหยุดการทำงานได้ มีเวลาการรันต่อครั้งที่ 12 ชั่วโมง



## VS-Code

วิชวลสตูดิโอโค้ด (อังกฤษ: Visual Studio Code) เป็นโปรแกรมแก้ไขซอร์สโค้ดที่พัฒนาโดยไมโครซอฟท์สำหรับ Windows, Linux และ macOS มีการสนับสนุนสำหรับการดีบั๊ก การควบคุม Git ในตัว และ GitHub การเน้นไวยากรณ์ การเติมโค้ดอัจฉริยะ ตัวอย่าง และ code refactoring มันสามารถปรับแต่งได้หลายอย่าง ให้ผู้ใช้สามารถเปลี่ยนธีม แป้นพิมพ์ลัด การตั้งค่า และติดตั้งส่วนขยายที่เพิ่มฟังก์ชันการทำงานเพิ่มเติม ซอร์สโค้ดนั้นฟรีและโอเพนซอร์สและเผยแพร่ภายใต้สิทธิ์การใช้งาน MIT ไบนารีที่คอมไพล์แล้วเป็นฟรีแวร์และฟรีสำหรับการใช้ส่วนตัวหรือเพื่อการค้า

ที่มา : <https://th.wikipedia.org/wiki/วิชวลสตูดิโอโค้ด>

3.2

<https://code.visualstudio.com/>

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs Download

Version 1.68 is now available! Read about the new features and fixes from May.

Code editing.  
Redefined.

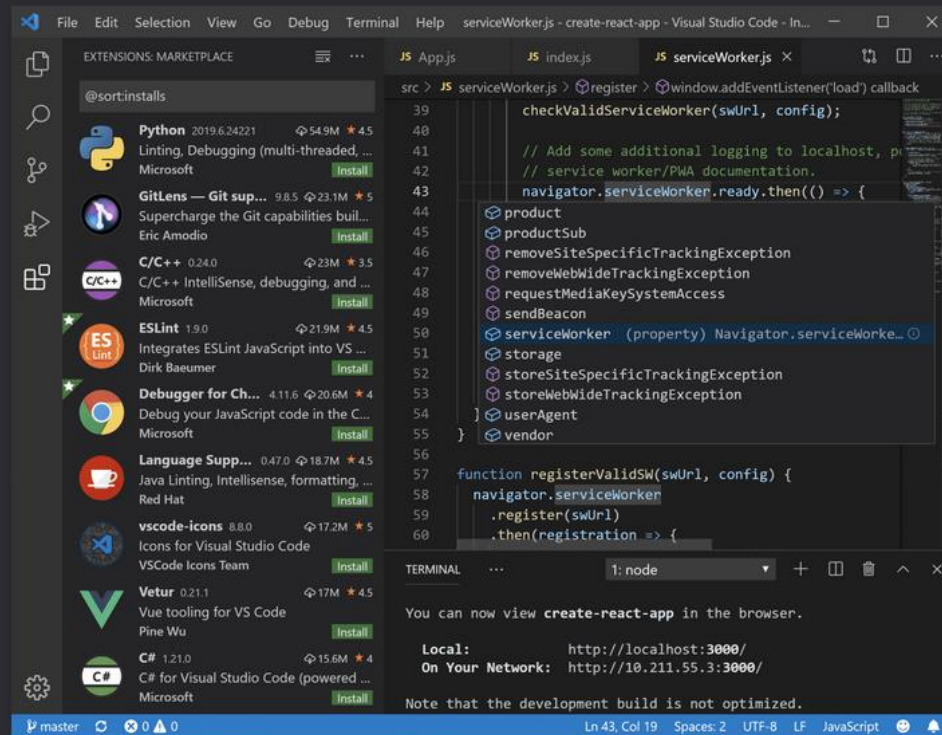
Free. Built on open source. Runs everywhere.

Download for Windows

Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its  
[license and privacy statement.](#)



IntelliSense



Run and Debug



Built-in Git










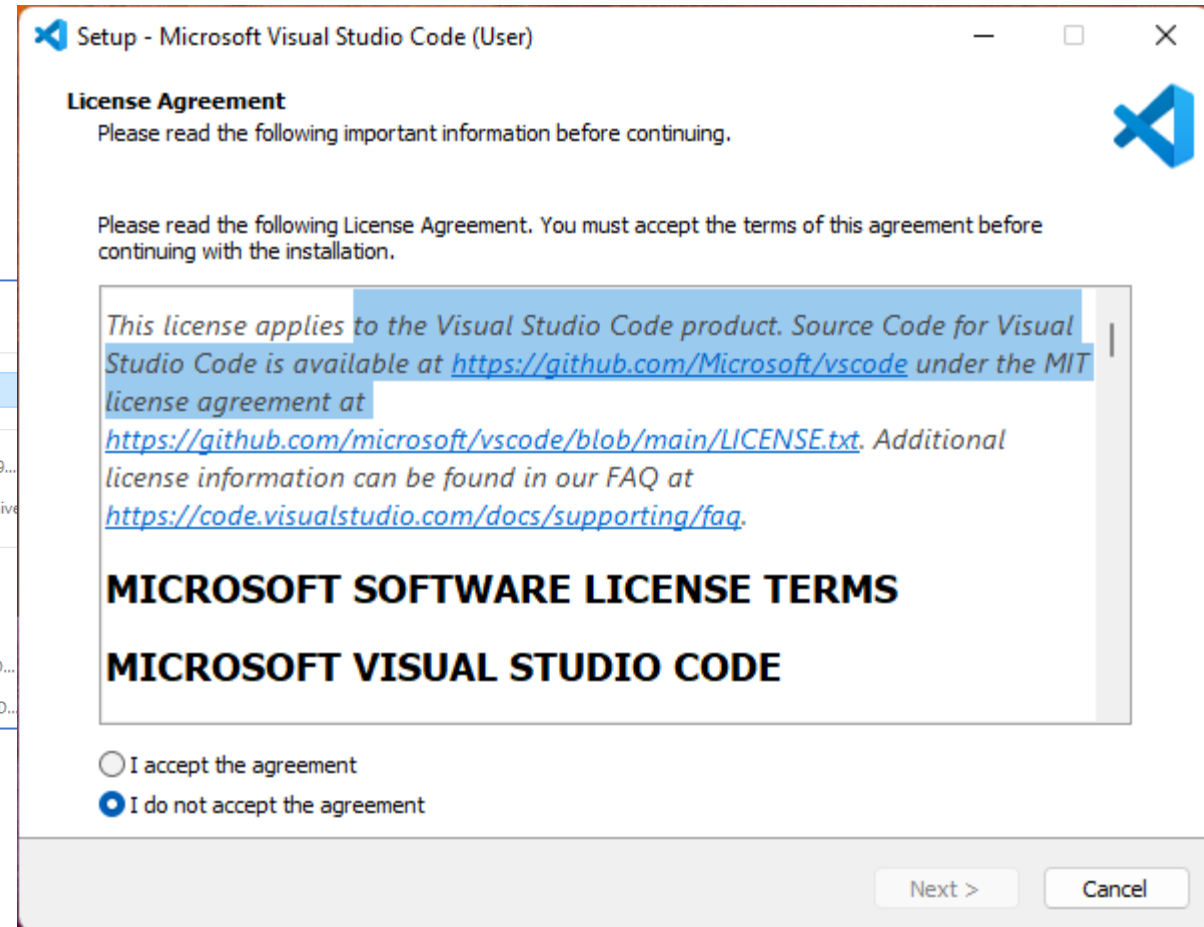
Extensions



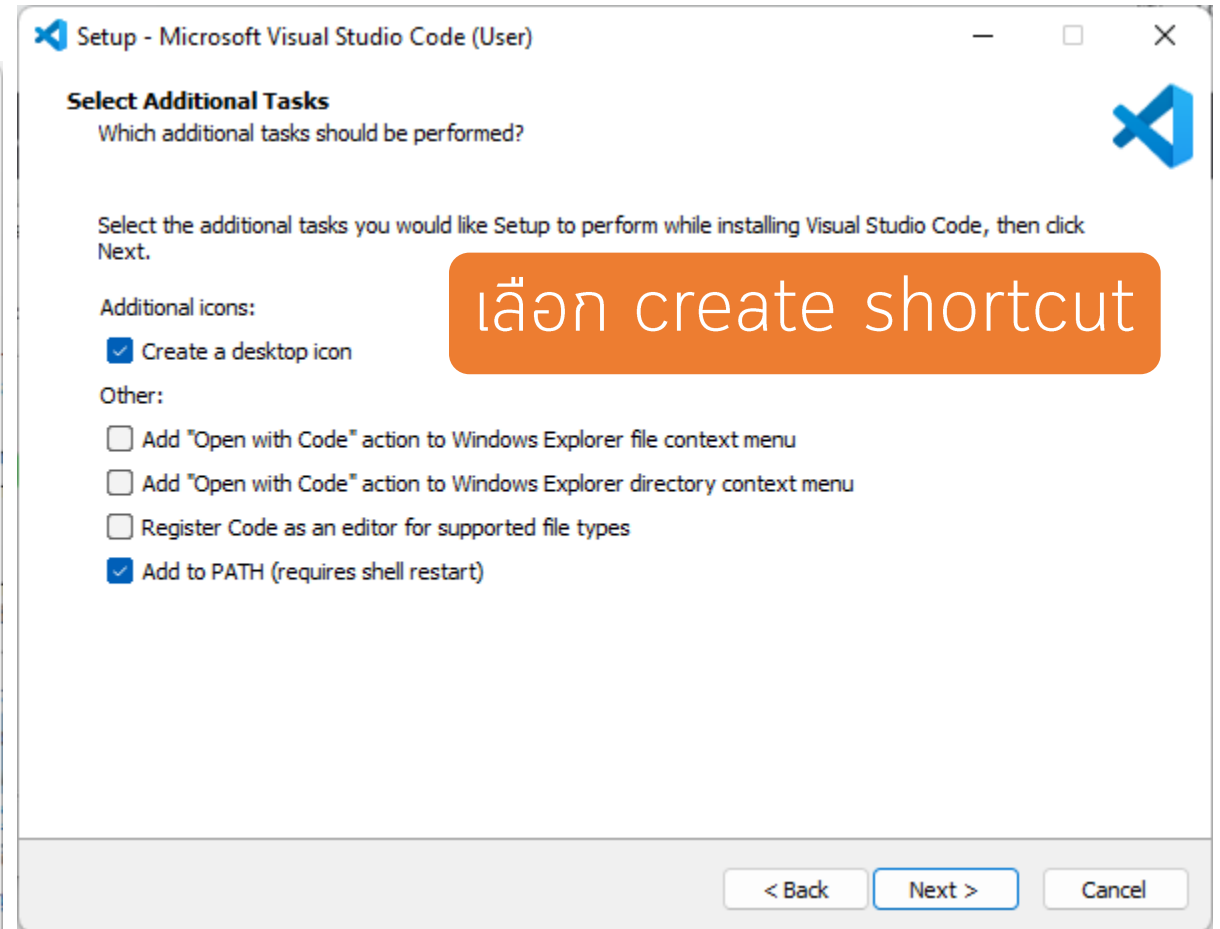
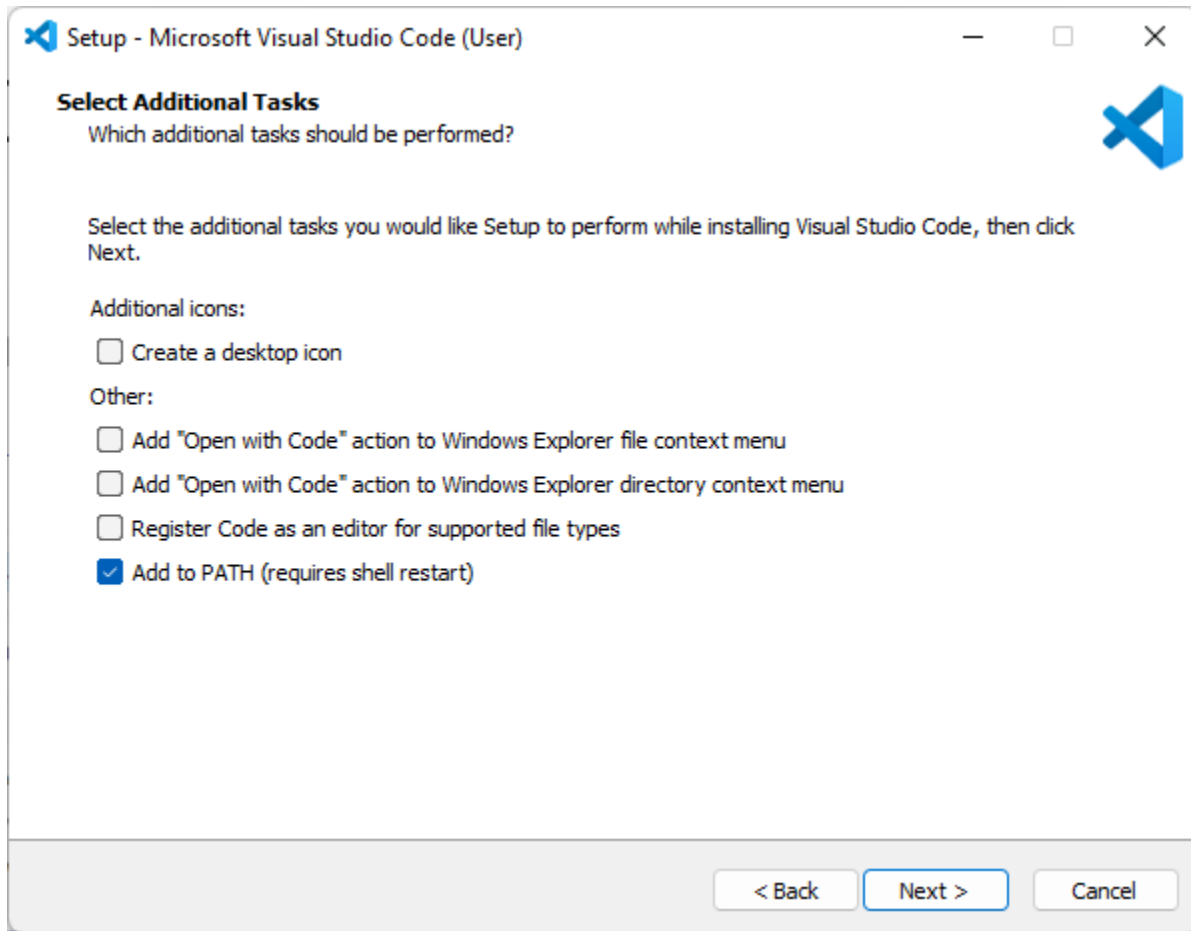
## วิธีการติดตั้ง VS-code

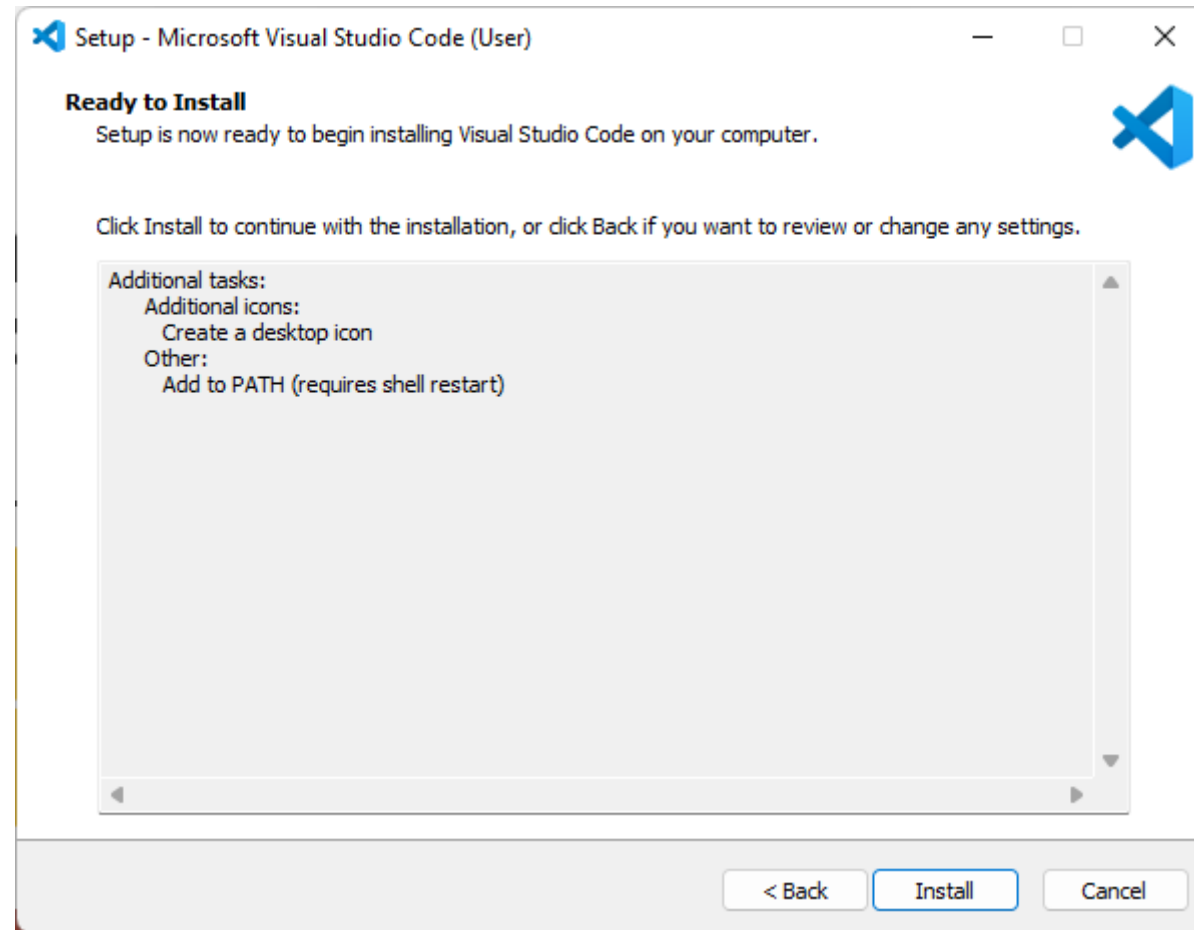
- ดับเบิลคลิกไฟล์ที่ดาวน์โหลด

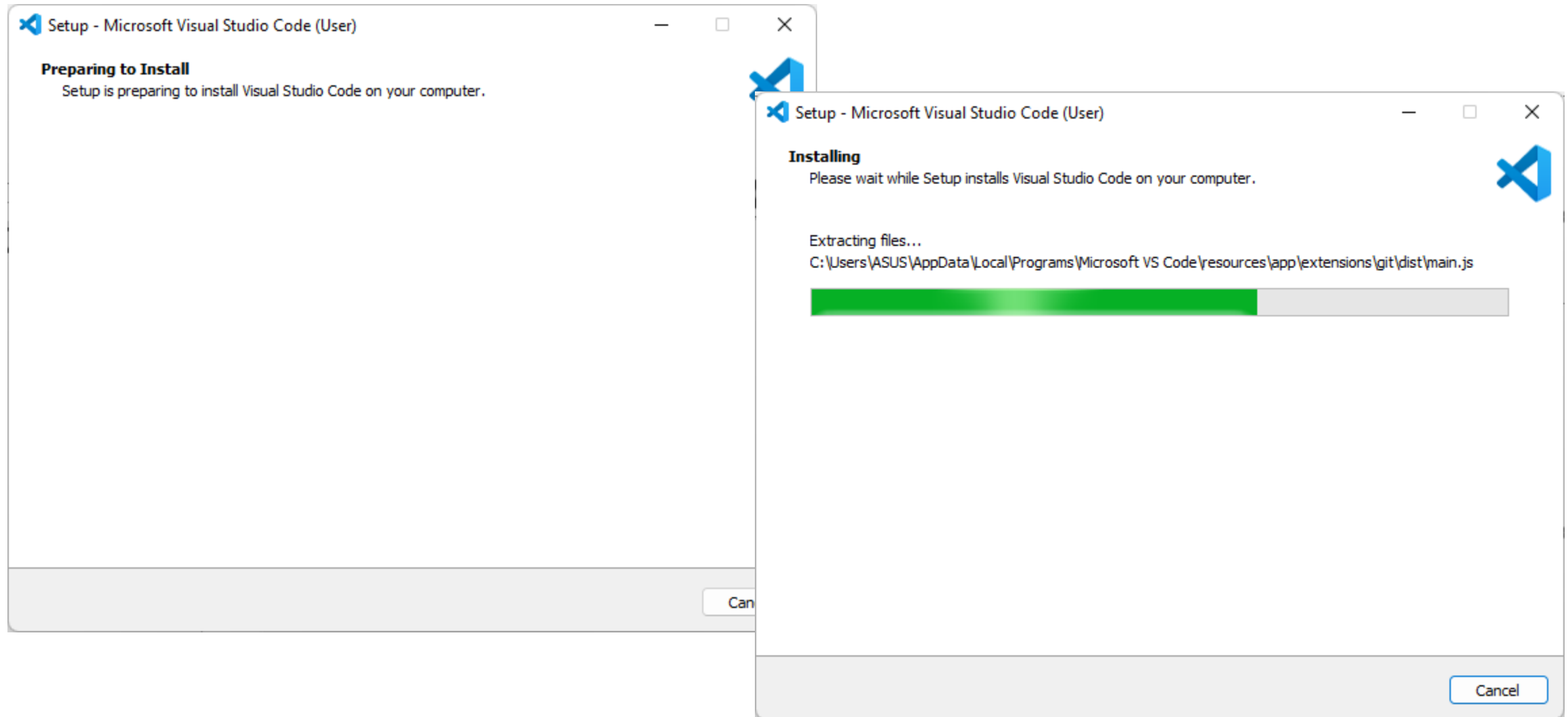
Name	Date modified	Type
▼ Today (1)		
 VSCodeUserSetup-x64-1.68.1.exe	6/26/2022 22:48	Application
▼ Last week (2)		
 78123d4690ecbf0c9a0603adc69779cf.doc	6/22/2022 11:25	Microsoft Word 9...
 เอกสารหลักฐาน SAR-2564 (21 มิย 2564)-20220620T061514Z-001.zip	6/20/2022 13:17	WinRAR ZIP archive
▼ Earlier this month (10)		
 ทีมสัหายขายลูกชิ้น_ประกาศนิมัตกรกิจกรรม PNRU Smart Start Idea by GSB 2565	6/2/2022 15:45	File folder
 รางวัลชมเชย_ประกาศนิมัตกรกิจกรรม PNRU Smart Start Idea by GSB 2565	6/2/2022 15:47	File folder
 แบบเสนอชื่อผู้ทรงคุณวุฒิเป็นกบปรจำ65 มศ.ด.pdf	6/4/2022 16:18	Adobe Acrobat D...
 มคอ7_64.docx	6/13/2022 23:04	Microsoft Word D...

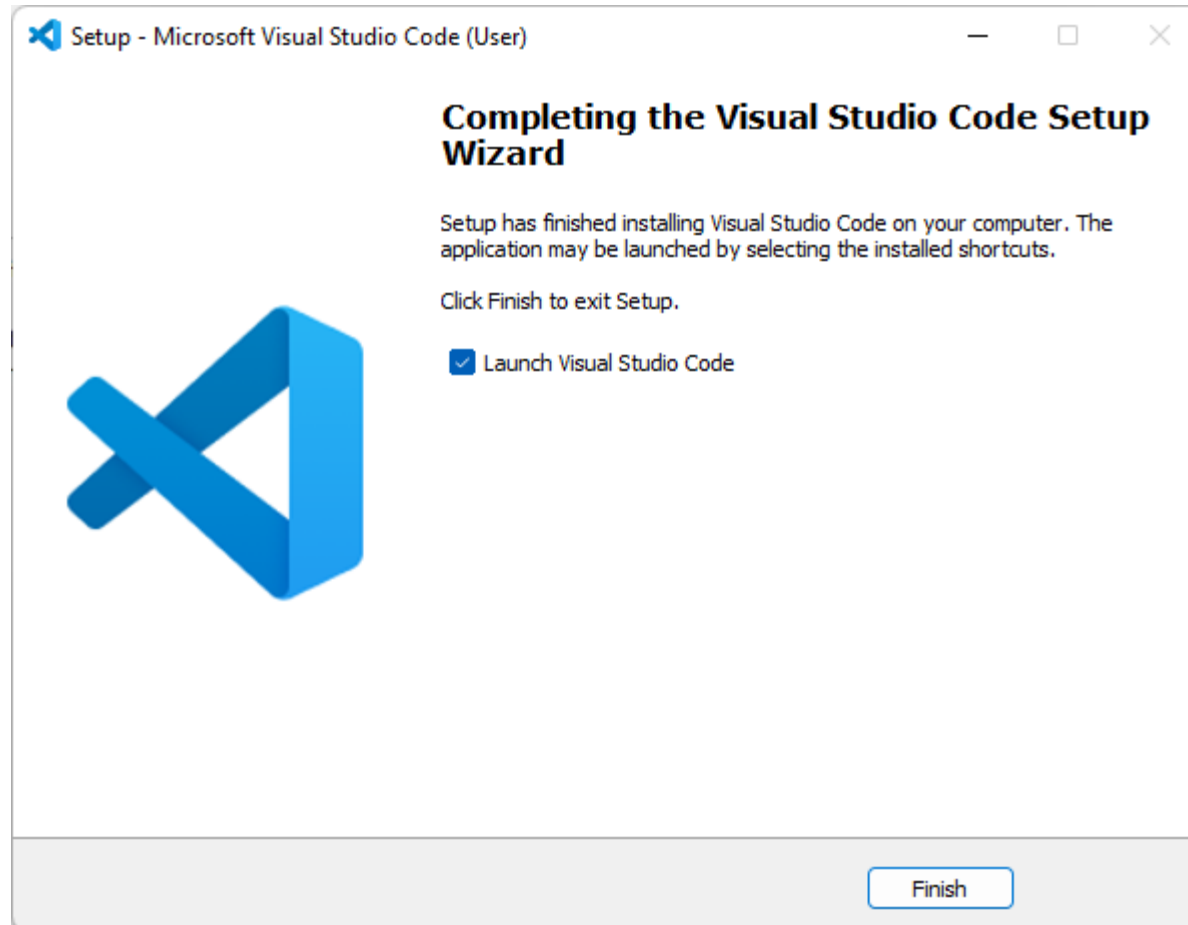


- กด Next แล้วปรากฏหน้าต่างดังรูป









# October 2022 (version 1.73)

**Update 1.73.1:** The update addresses these [issues](#).

Welcome to the October 2022 release of Visual Studio Code. There are many updates in this version that we hope you'll like, some of the key highlights include:

- **Search include/exclude folders** - Quickly set folders to include/exclude in the tree view.
- **Command Center mode shortcuts** - Displays commonly used modes before recent files list.
- **Merge editor improvements** - Access Combination option to auto merge conflicts.
- **Markdown automatic link updates** - File and image links updated on rename/move.
- **More audio cues** - Task completed or failed, Terminal Quick Fix available.
- **vscode.dev protected branch workflow** - Create new branch when committing to protected branch.
- **New Python extensions** - New standalone extensions for isort, Pylint, and Flake8.
- **Dev Container Templates** - Create new Dev Containers based on existing templates.

If you'd like to read these release notes online, go to [Updates](#) on [code.visualstudio.com](https://code.visualstudio.com).

**Insiders:** Want to try new features as soon as possible? You can download the nightly [Insiders](#) build and try the latest updates as soon as they are available.



OUTPUT   TERMINAL   JUPYTER   DEBUG CONSOLE   PROBLEMS

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

+   v   ^   x

Pyth...

Pyth...

Pyth...

Pyth...



# October 2022 (version 1.73)

**Update 1.73.1:** The update addresses these [issues](#).

Welcome to the October 2022 release of Visual Studio Code. There are many updates in this version that we hope you'll like, some of the key highlights include:

- **Search include/exclude folders** - Quickly set folders to include/exclude in the tree view.
- **Command Center mode shortcuts** - Displays commonly used modes before recent files list.
- **Merge editor improvements** - Access Combination option to auto merge conflicts.
- **Markdown automatic link updates** - File and image links updated on rename/move.
- **More audio cues** - Task completed or failed, Terminal Quick Fix available.
- **vscode.dev protected branch workflow** - Create new branch when committing to protected branch.
- **New Python extensions** - New standalone extensions for isort, Pylint, and Flake8.
- **Dev Container Templates** - Create new Dev Containers based on existing templates.

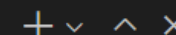
If you'd like to read these release notes online, go to [Updates](#) on [code.visualstudio.com](https://code.visualstudio.com).

**Insiders:** Want to try new features as soon as possible? You can download the nightly [Insiders](#) build and try the latest updates as soon as they are available.



OUTPUT   TERMINAL   JUPYTER   DEBUG CONSOLE   PROBLEMS

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>



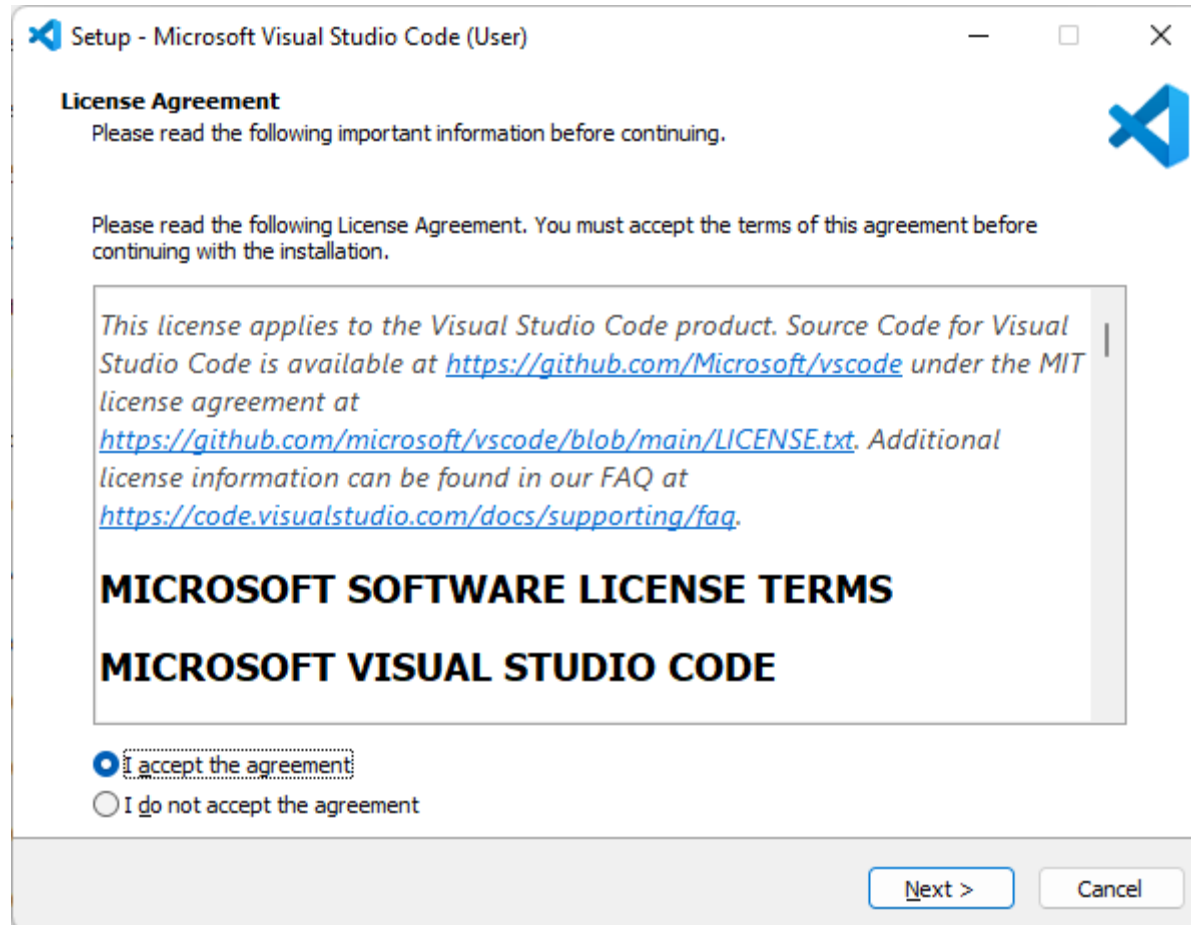
Python...

Python...

Python...

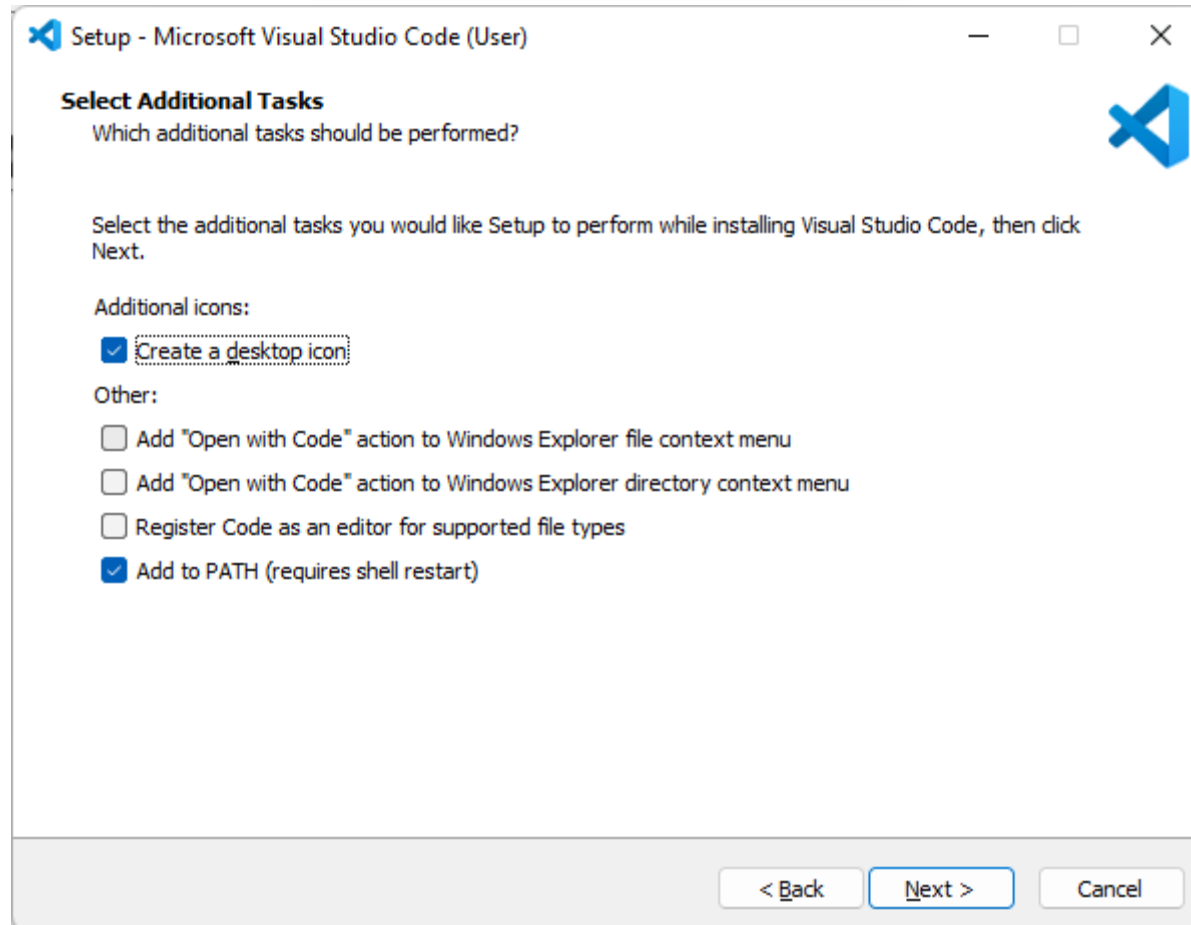
Python...



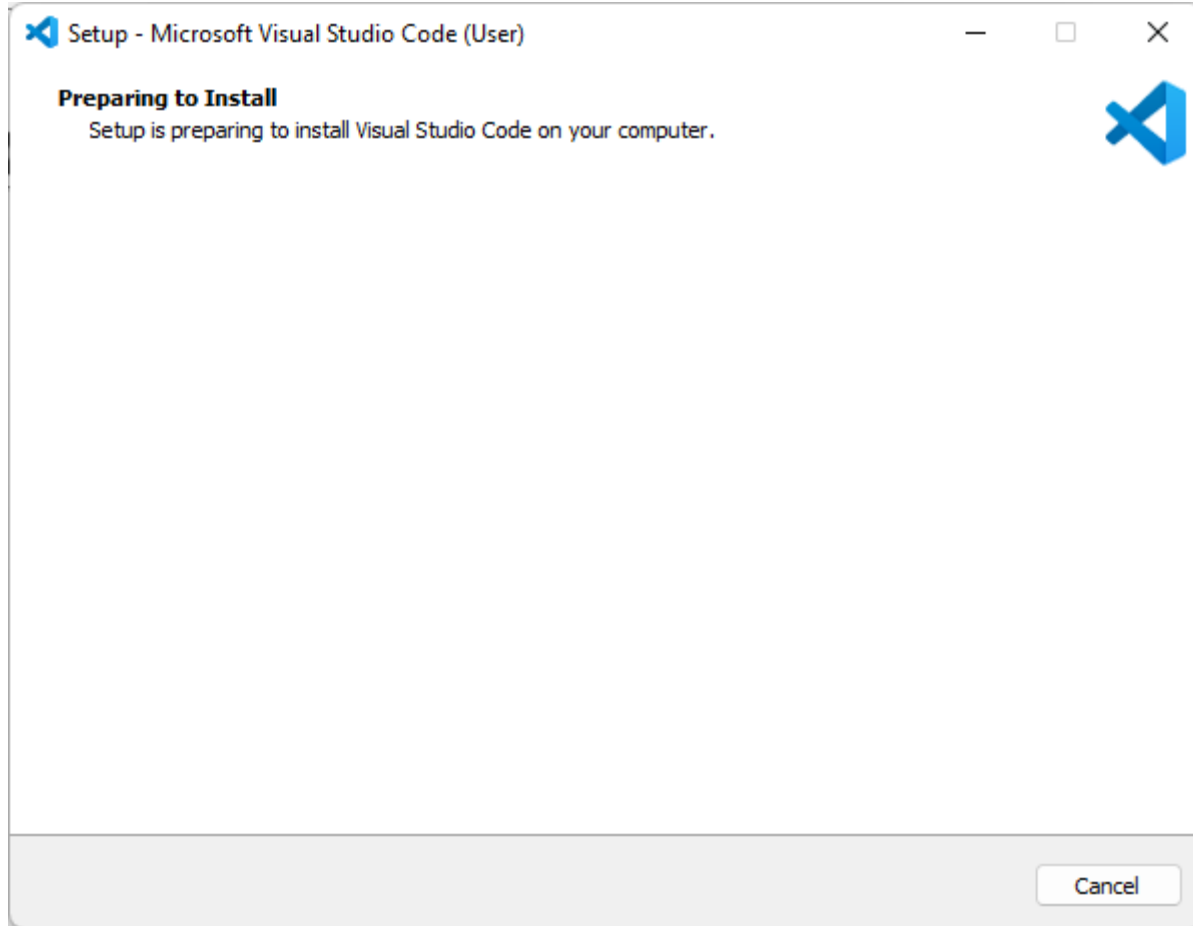




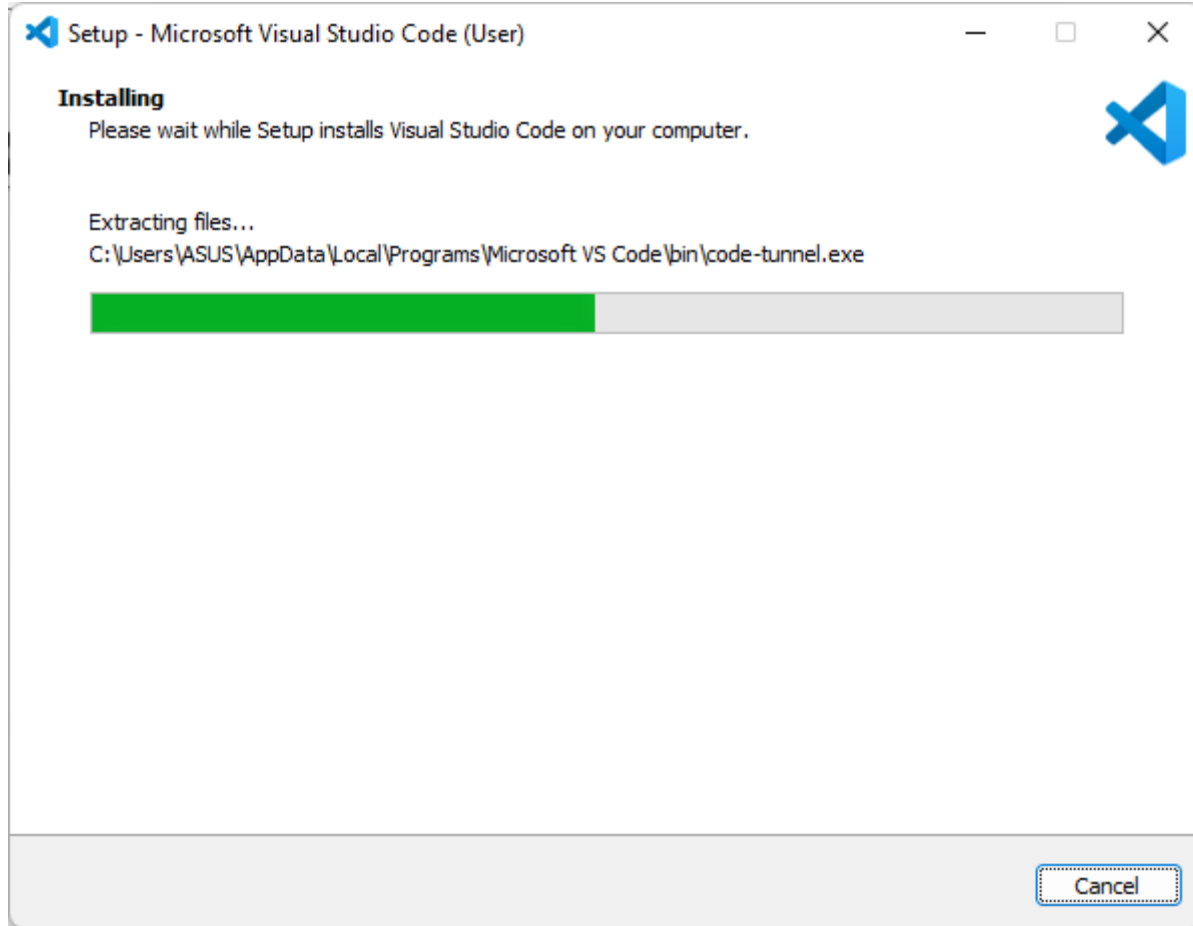
# 3.3



# 3.3



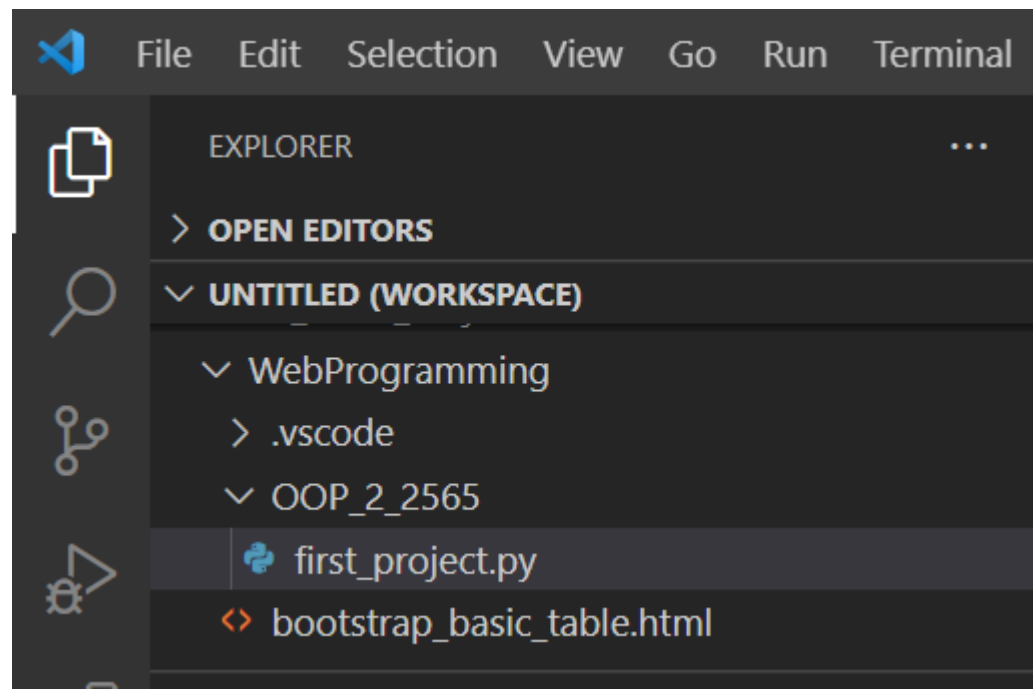
# 3.3



3.3

# การสร้างไฟล์ .py first\_project.py

- การสร้างไฟล์ .py

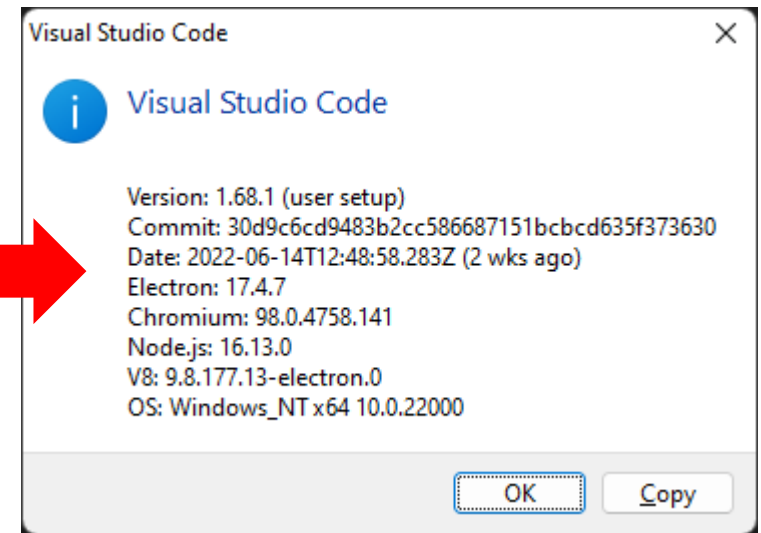
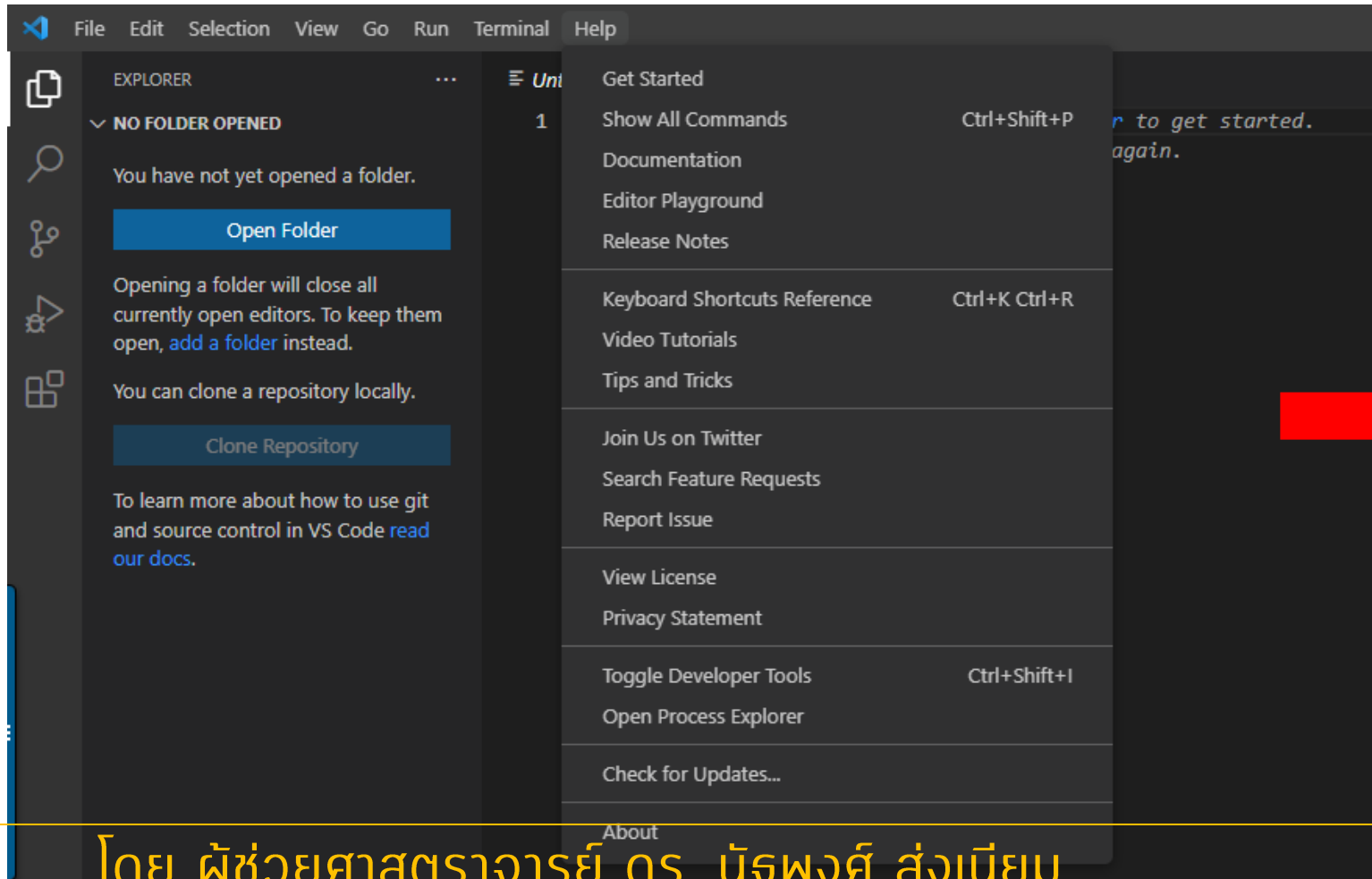




## การเปิดใช้งานโปรแกรม VS-Code

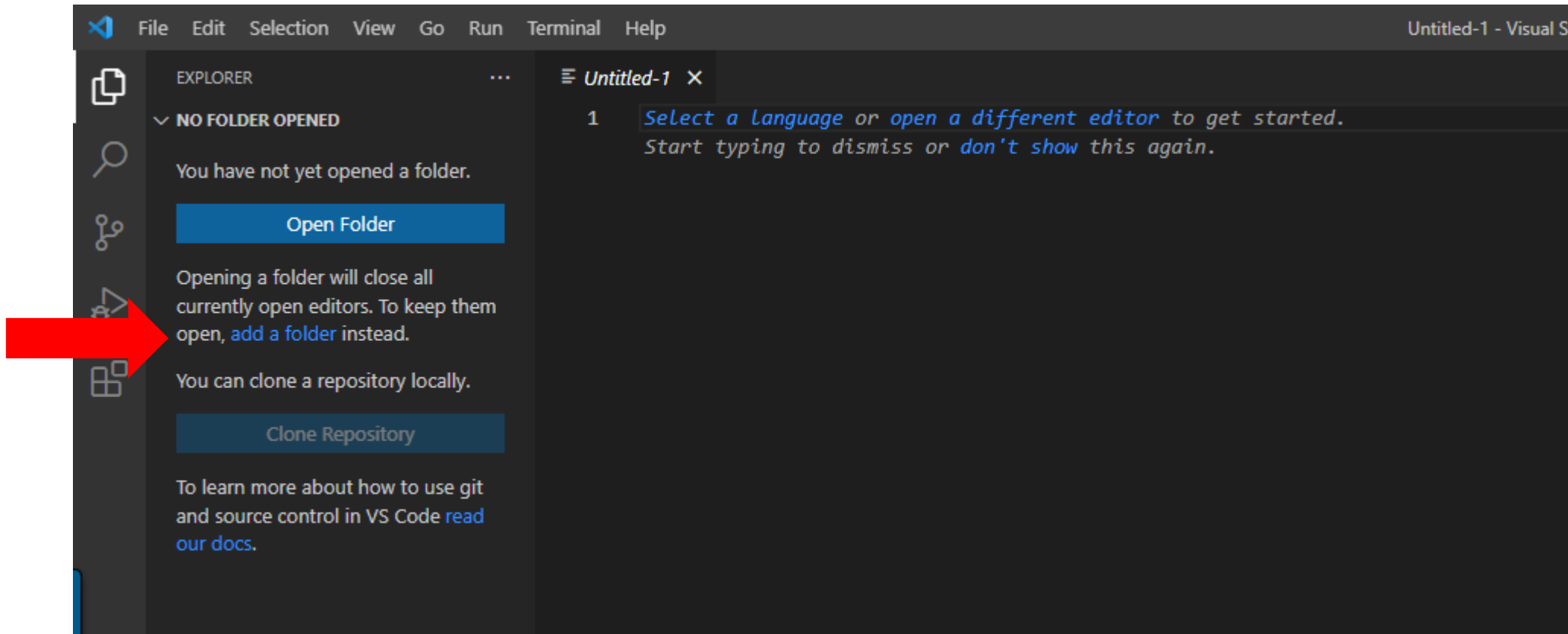
- Double click ที่ ไอคอนโปรแกรม VS-Code ดังรูป

## วิธีการตรวจสอบรุ่นของ Vs-code



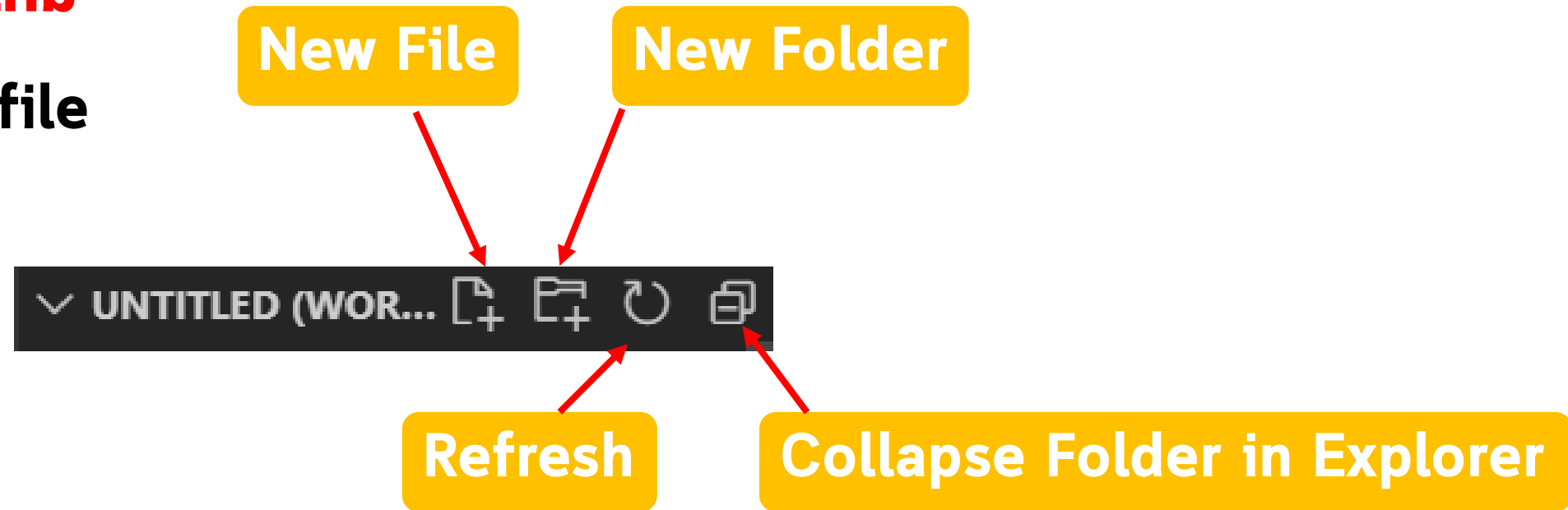
### การเพิ่ม folder

- คลิกที่ Add Folder



### การสร้างไฟล์ใหม่

- คลิกที่ปุ่ม new file

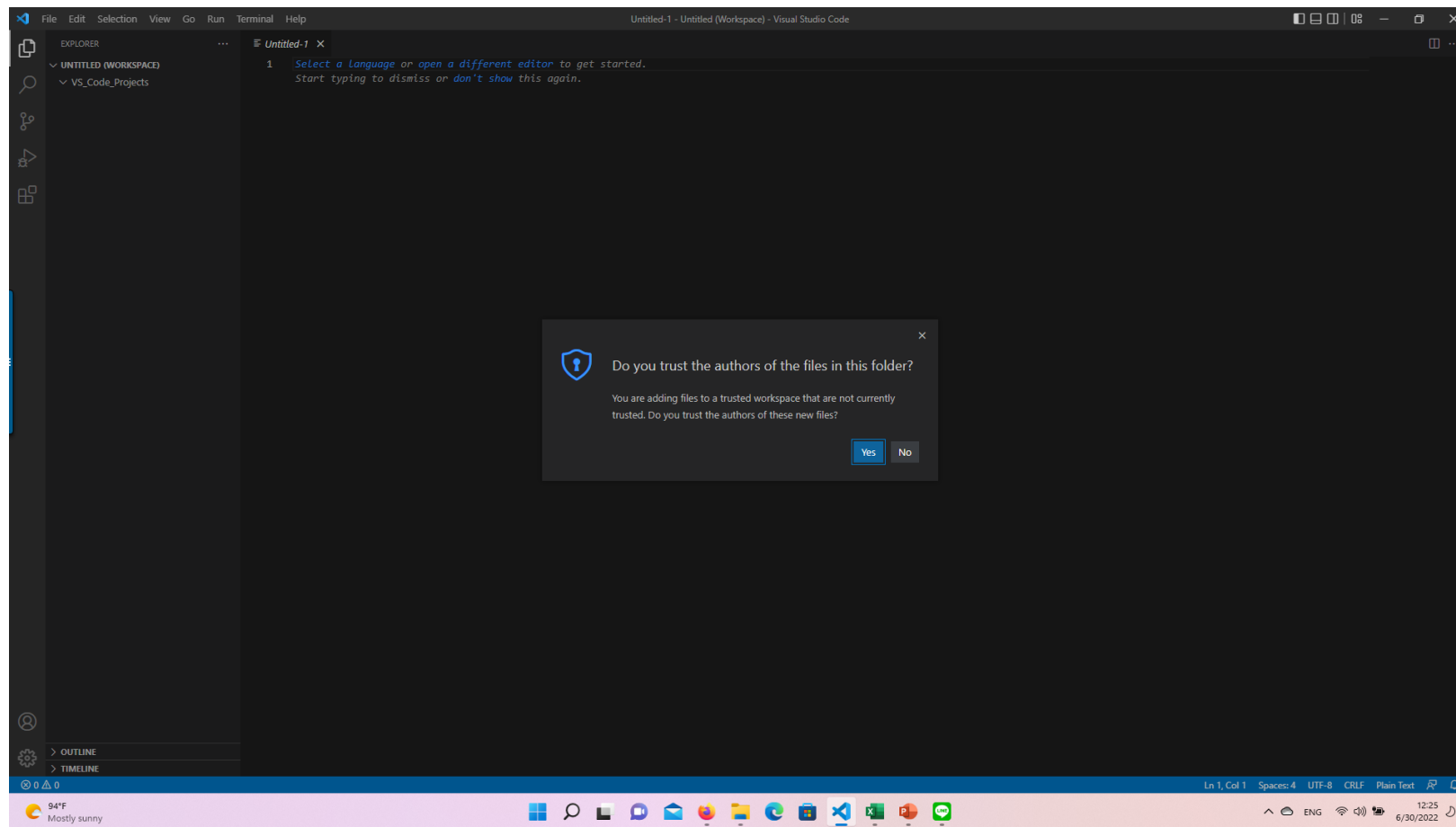




## 3.5 การใช้งาน VS-code เบื้องต้น



### จะได้ Folder : OOP\_Projects



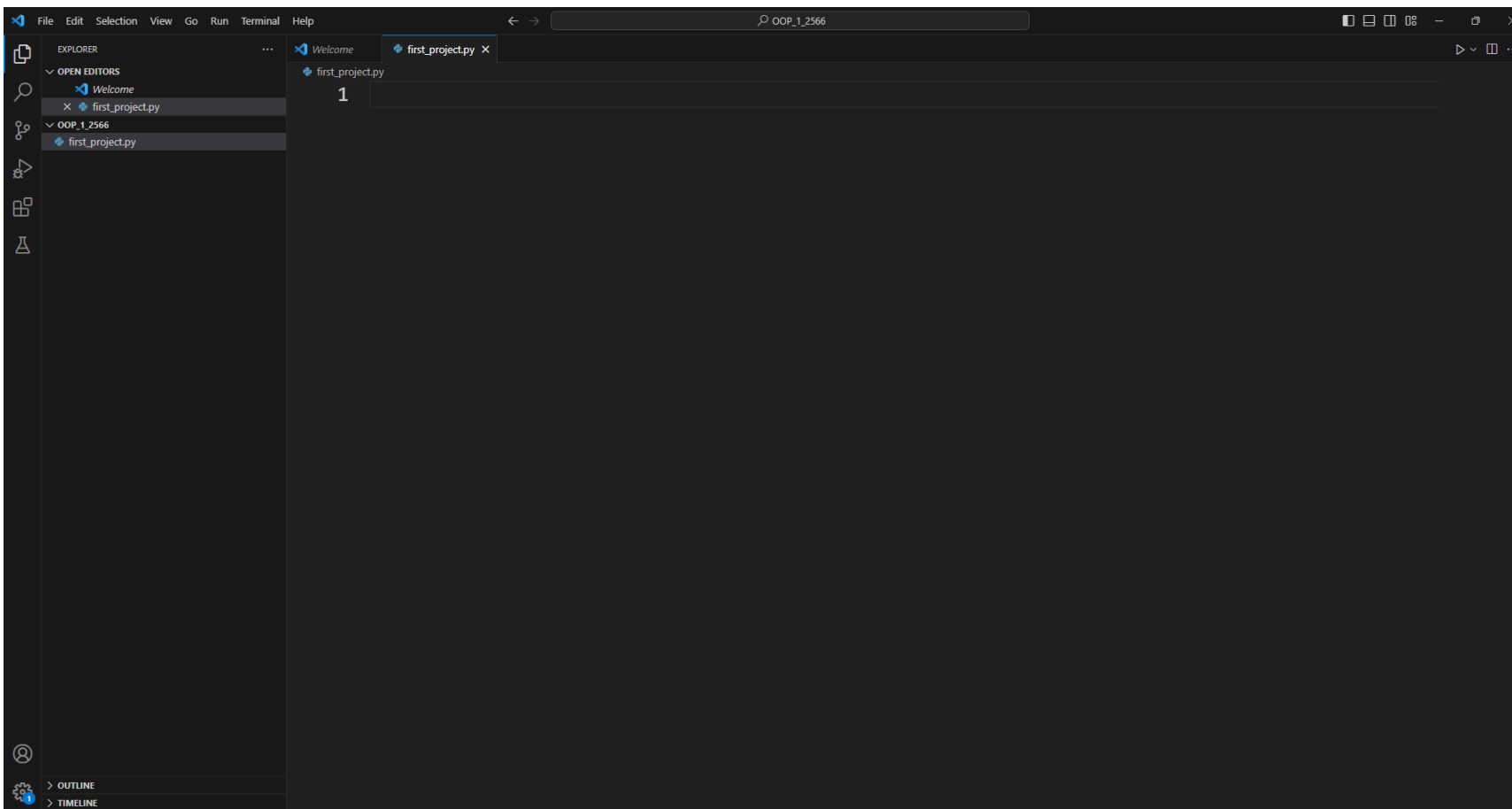
โดย ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่งเนียม



## 3.5 การใช้งาน VS-code เบื้องต้น



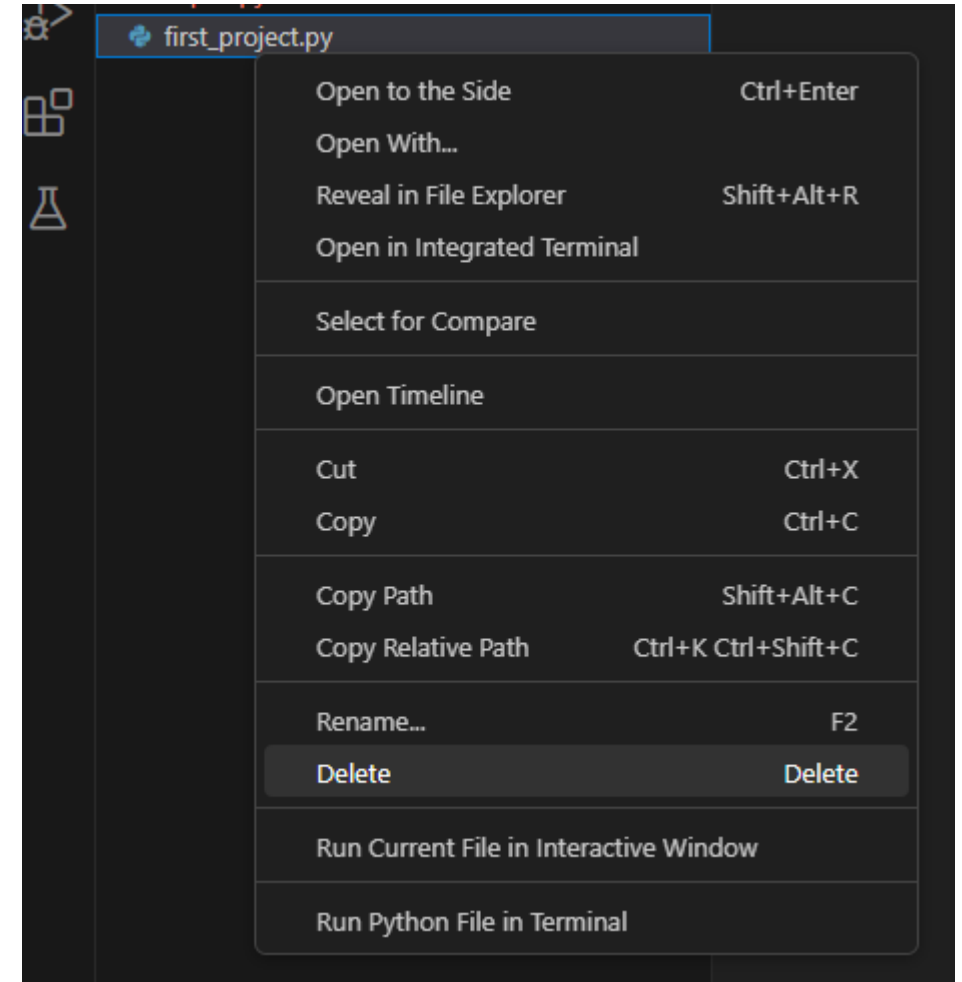
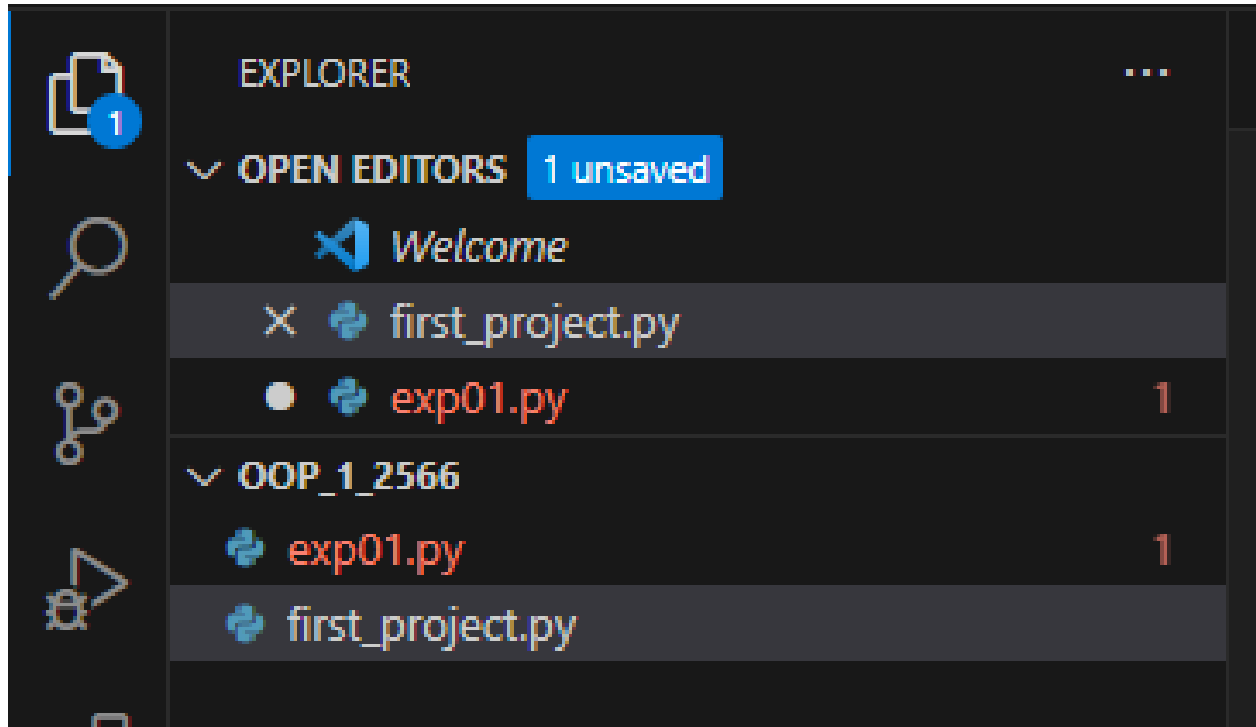
จะได้ Folder : OOP\_1\_2566



โดย ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่งเนียม

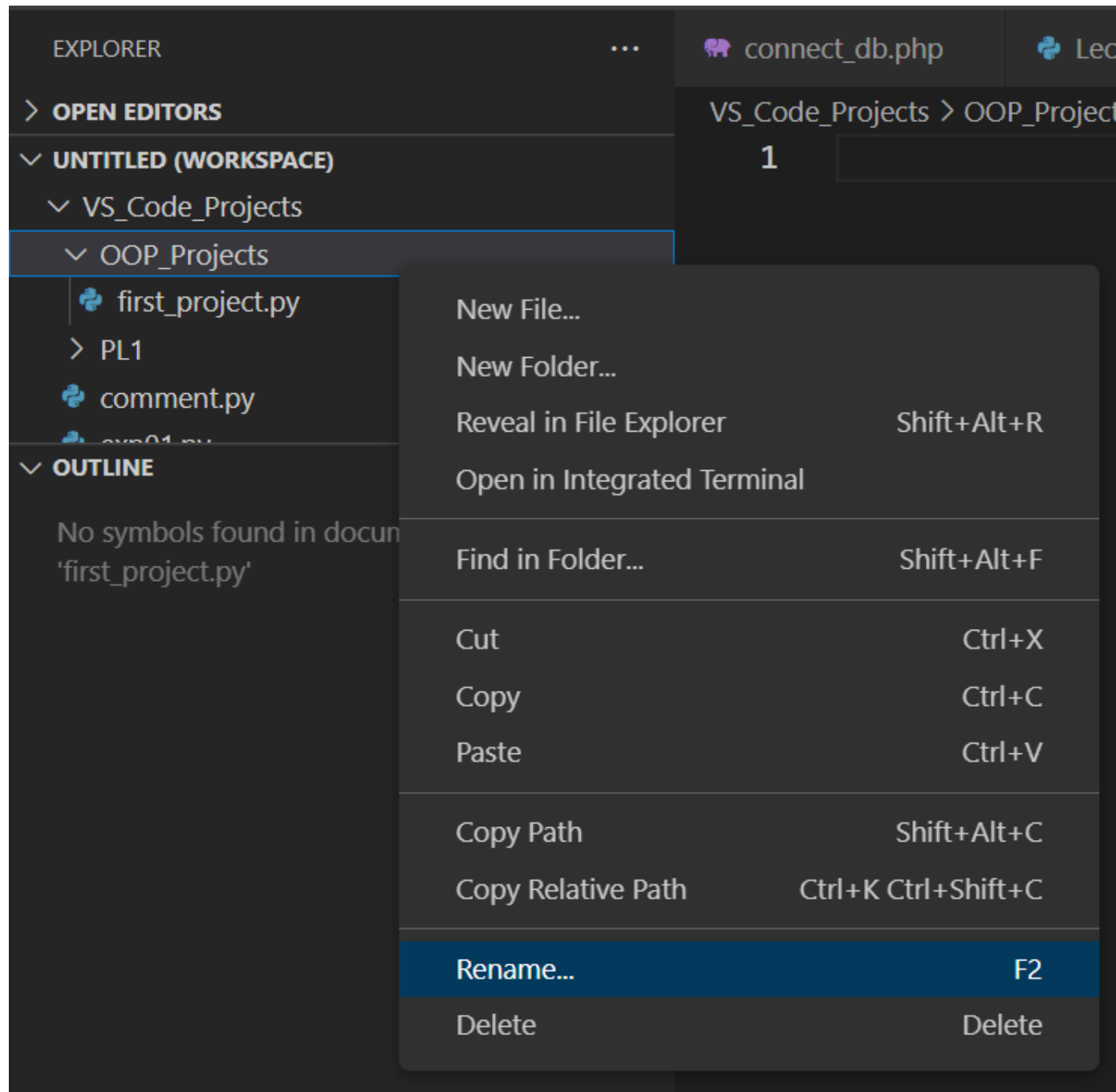


# First\_project.py



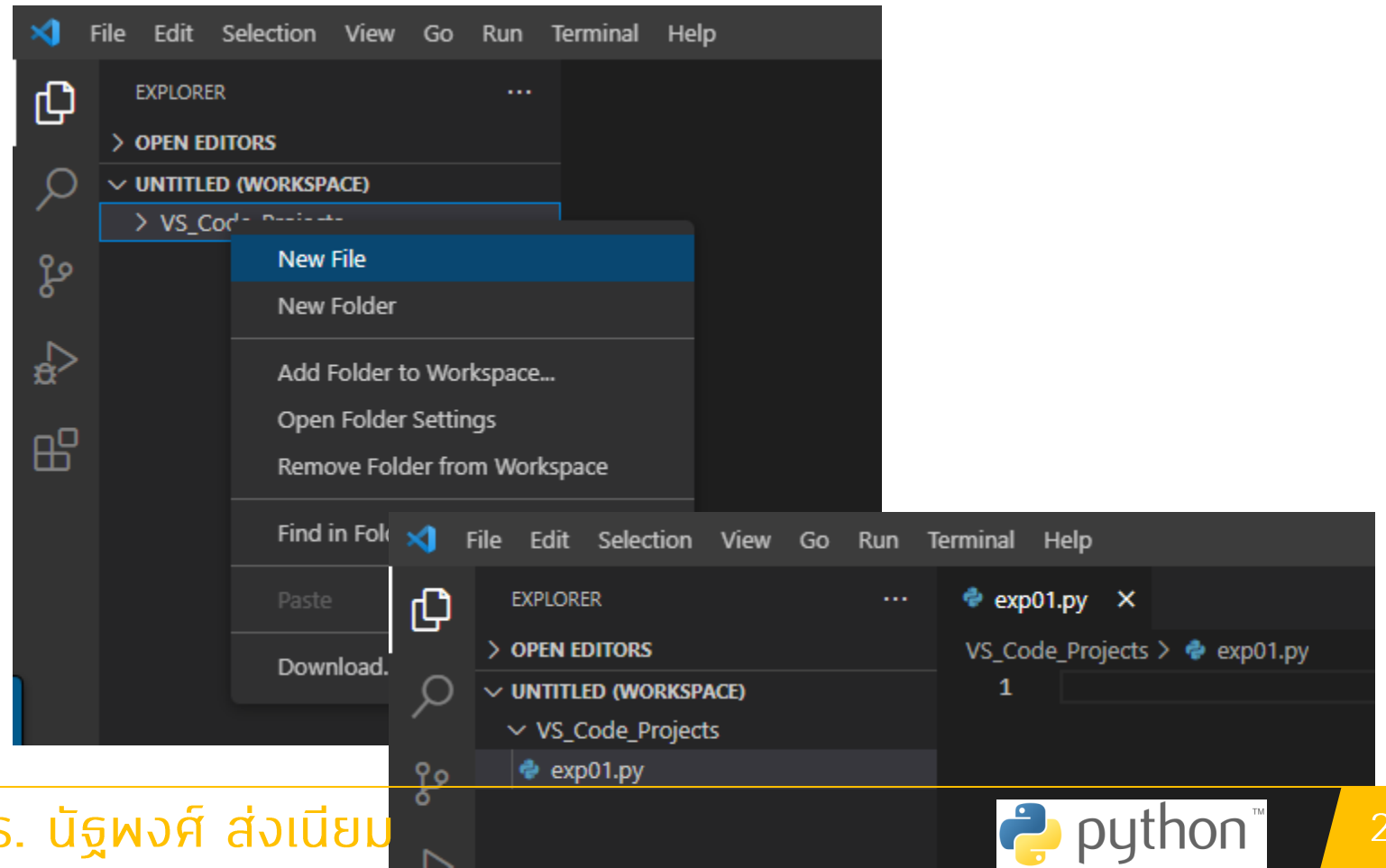
# การเปลี่ยนชื่อ Folder

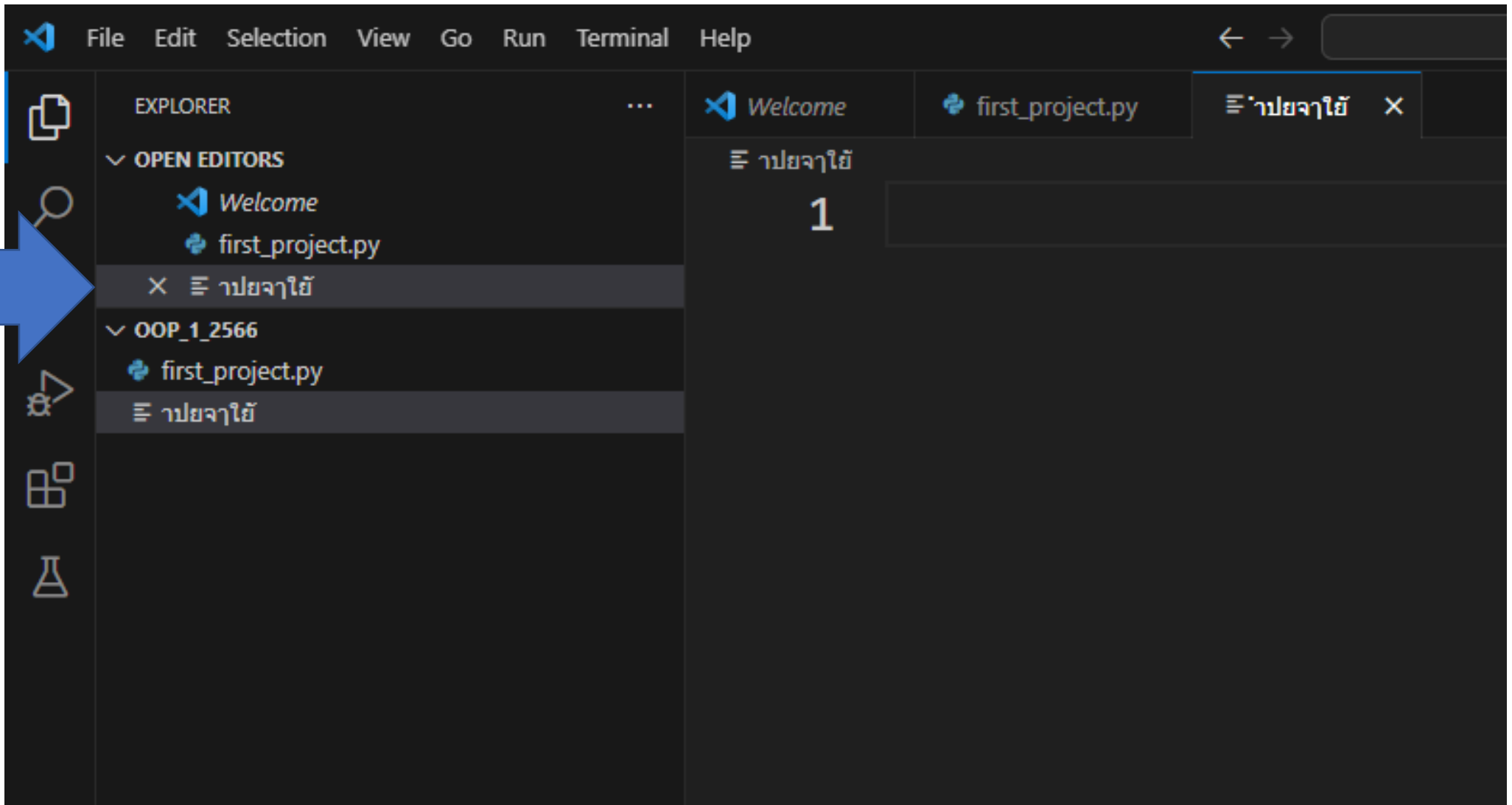
- คลิกขวา
- เลือก Rename



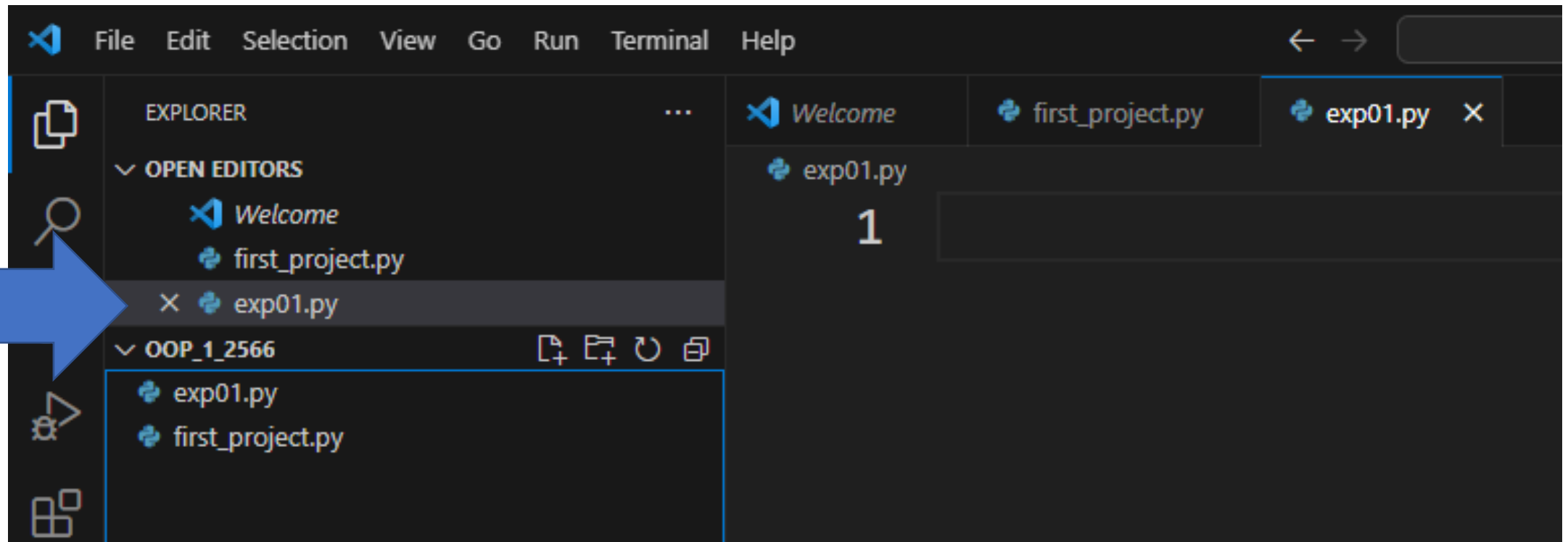
## การสร้างไฟล์ใหม่

- คลิกขวาที่ Folder
- เลือก new file
- ตั้งชื่อ exp01.py

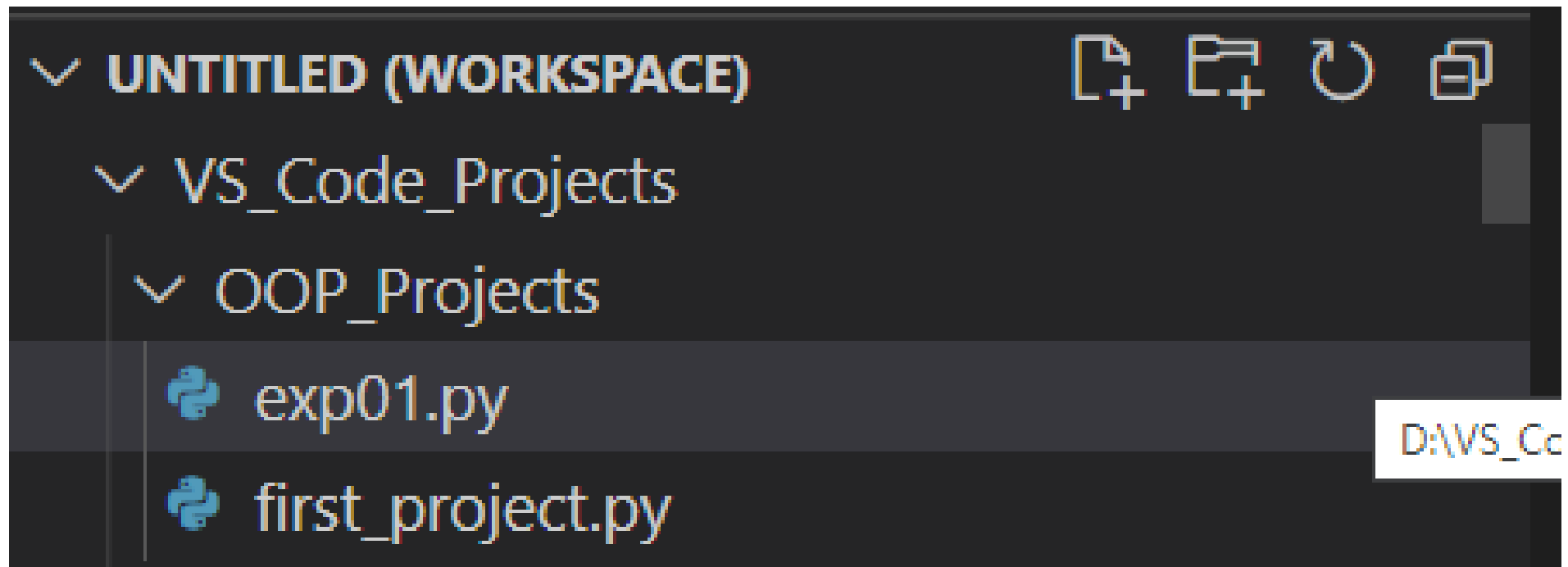




# Rename



# สร้างไฟล์ใหม่

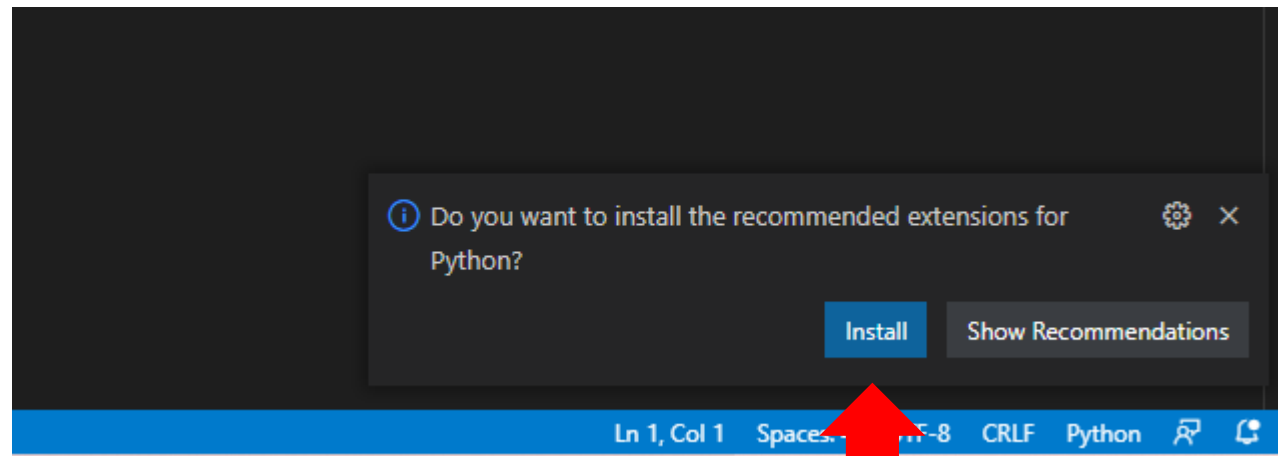




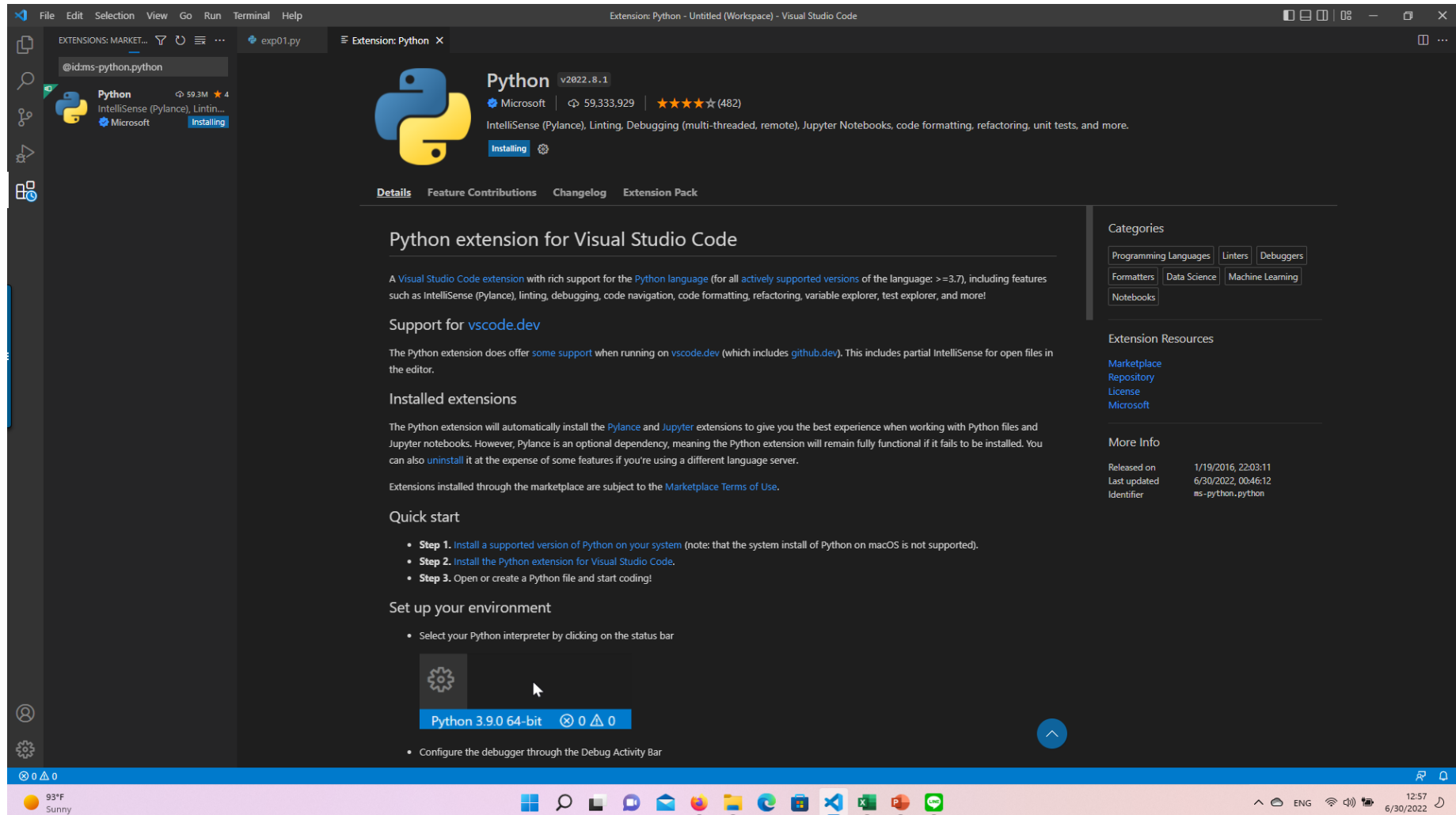
### การติดตั้ง extension ของ Python

Vs-code จะถามว่าต้องการทำการติดตั้ง extension ของ python หรือไม่ เพื่อที่จะเรียกใช้งานโครงสร้างของ python

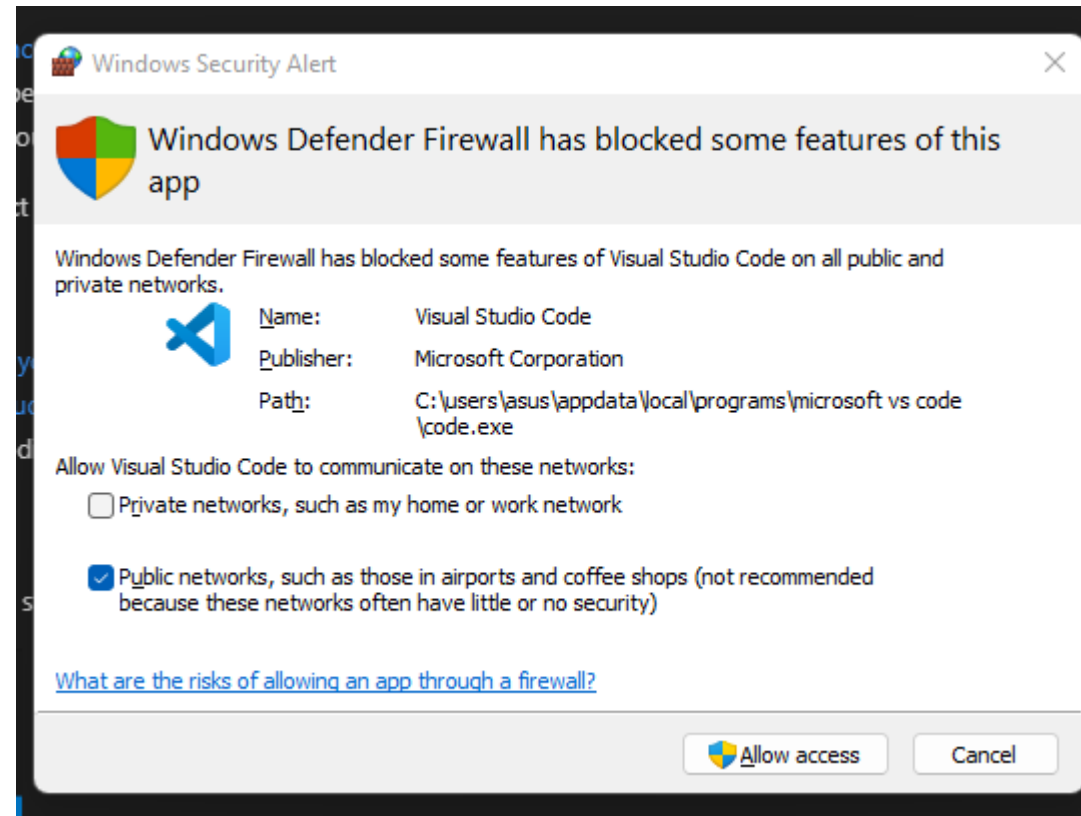
- กด install ไป



- จะได้



# กด Allow access



EXTENSIONS: MARKETPLACE

@category:debuggers Python

- Python** 71M ★ 4  
IntelliSense (Pylance), Linting, Debugging (multi-threaded, r...  
Microsoft [Install](#)
- Python-Preview** 1M ★ 4.5  
Provide Preview for Python Execution.  
dongli [Install](#)
- AREPL for python** 550K ★ 5  
real-time python scratchpad  
Almenon [Install](#)
- Python Image Preview** 238K ★ 5  
Numpy, Pillow, OpenCV, Matplotlib, Plotly, ImageIO, Scikit I...  
윤대희 [Install](#)
- Python C++ Debugger** 164K ★ 5  
Extension for starting the Python & C++ debugger on the sa...  
BeniBenj [Install](#)
- GeeCode Python** 77K  
Linting, Debugging (multi-threaded, remote), Intellisense, co...  
GeeCode Team [Install](#)
- LiveCode for python** 68K ★ 5  
Real-time python variable evaluation  
xirider [Install](#)
- Python Resource Monitor** 42K ★ 5  
A resource monitor for the Python Debugger  
kaih2o [Install](#)
- Python Quick Print** 46K ★ 5  
Quickly print out log messages  
AhadCove [Install](#)
- Python Debugging Address Prompt** 32K  
Prompt for host/port before starting a Python debugging se...  
Alexander Sagen [Install](#)
- Quick-Python-Print** 9K ★ 5  
Quickly handle print operations : Insertion(include print out ...  
WeidaWang [Install](#)
- Python Quick Logger** 3K  
Wrap variables with stdlib Logging instances: log.info().  
Brian Crant [Install](#)
- NI Python DataPlugins** 8K ★ 5  
Create and export NI DataPlugins in Python  
NI [Install](#)
- Python Colorful Print** 732 ★ 5  
Python print colorful out log messages  
Ali Yaman [Install](#)
- Console Wrap PHP/JS/Python** 6K  
This extension helps you easily create log statements to print

exp01.py

```
OOP_Python_Projects > exp01.py
1 # My Python
2 name = input('What is your name \n')
3 print ('Hi , %s' % name)
4 print ('Welcome to Python')
```



Do you want to install the recommended extensions for Python?

[Install](#) [Show Recommendations](#)

EXTENSIONS: MARKETPLACE

@id:ms-python.python

**Python**  
IntelliSense (Pylance), Linting, Debugging (multi-threaded, r...  
Microsoft

547ms

Get Started

exp01.py

Get Started

Get started with Python development

Your first steps to set up a Python project with all the powerful tools and features that the Python extension has to offer!

- Create a Python file  
Open or create a Python file - make sure to save it as ".py".  
**Create Python File**
- Create a Python Environment
- Run and debug your Python file
- Explore more resources
- Mark Done

EXPLORER: MY-APP

- app.py
- tests.py

my-app

Get Started

Get started with Jupyter Notebooks

Your first steps to set up a Jupyter project with all the powerful tools and features that the Jupyter Extension has to offer!

- Create or open a Jupyter Notebook
- Select a Jupyter kernel  
Click Select Kernel at the top right of the notebook and choose the environment in which to run your kernel. Or, with a notebook opened and in focus, open the Command Palette and run the command:  
Notebook: Select Notebook Kernel.  
**Open Command Palette to Select a Jupyter Kernel**  
Tip: Use keyboard shortcut (Ctrl+Shift+P)
- Explore data and debug
- Find out more!
- Mark Done

Select kernel for 'Jupyter Notebook.ipynb'

- Python 3.7.10 64-bit ('base': conda)  
~\Anaconda3\python.exe

คุณกำลังนำเสนอต่อทุกคน

คลิกที่นี่เพื่อกลับไปยัง Hangouts วิดีโอเมื่อคุณพร้อมที่จะหยุดการนำเสนอ

via meet.google.com

Exp01.py



# comment

```
Get Started exp01.py x
OOP_Python_Projects > exp01.py > ...
1 # My Python
2 name = input('What is your name \n')
3 print ('Hi , %s' % name)
4 print ('Welcome to Python')
```



Home



Apps



Gaming



# Python 3.10

Python Software Foundation



27.15 MB of 39.57 MB downloaded..

4.3 ★

Average

14

Ratings

Developer tools



Library



Help

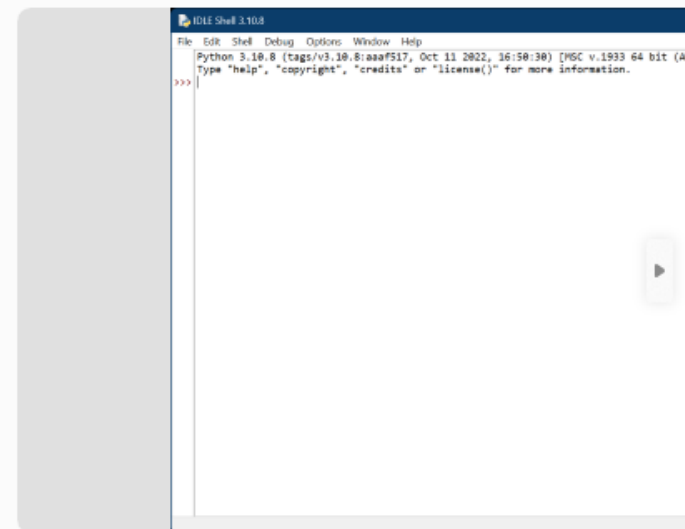
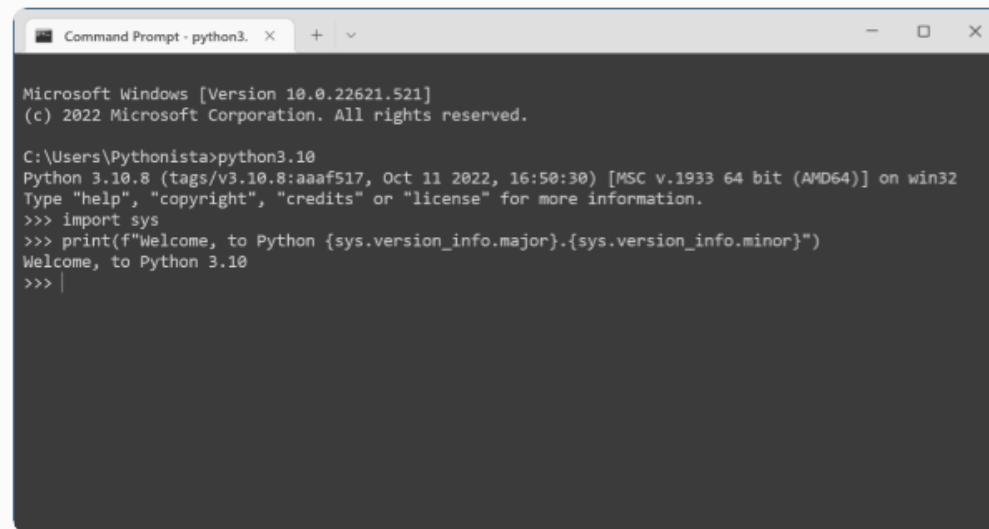


3+

Age Rating: IARC: 3+



## Screenshots



## Description

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

=====

Expected your existing copy of Python to run? Either update your PATH to raise its priority, or open "Manage App Execution Aliases" from Start to disable the shortcuts.

=====

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://...>

[Show more](#)

## Ratings and reviews

EXTENSIONS: MARKETPLACE

@id:ms-python.python

**Python**  
IntelliSense (Pylance), Linting, Debugging (multi-threaded, r...  
Microsoft

547ms

Settings gear icon

Get Started

OOP\_Python\_Projects > python

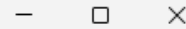
Select at workspace level

```

1 # My Python
2 name = input('What is your name \n')
3 print ('Hi , %s' % name)
4 print ('Welcome to Python')
```

Select the workspace folder to set the interpreter





Home



Apps



Gaming



# Python 3.10

Python Software Foundation

You can find these apps in your All apps list.

4.3 ★

Average

14

Ratings

Developer tools



Library



Help

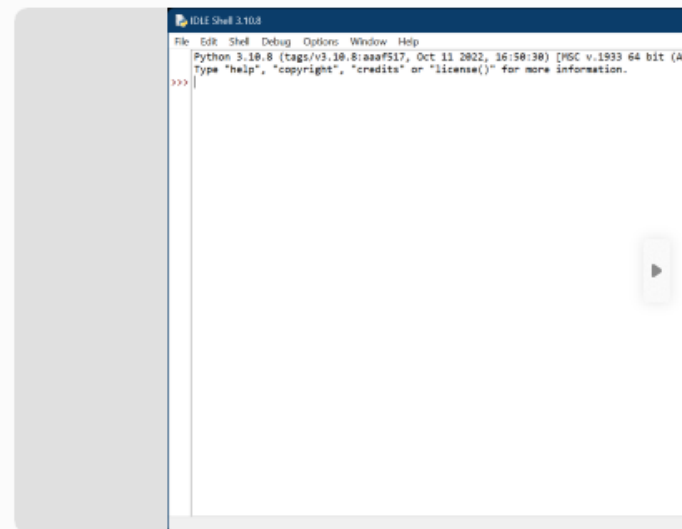
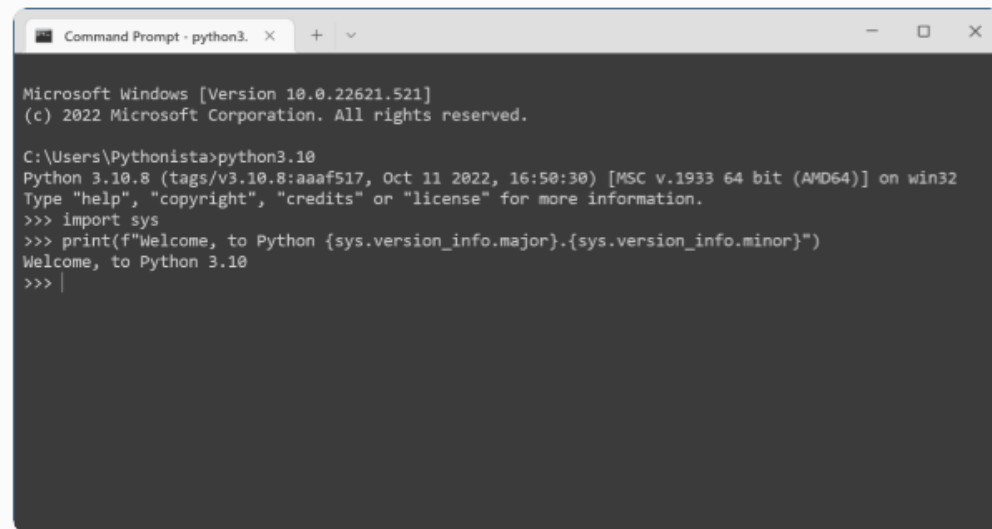


3+

Age Rating: IARC. 3+



## Screenshots



## Description

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

=====

Expected your existing copy of Python to run? Either update your PATH to raise its priority, or open "Manage App Execution Aliases" from Start to disable the shortcuts.

=====

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://...>

[Show more](#)

## Ratings and reviews

EXTENSIONS: MARKETPLACE

@id:ms-python.python

**Python**  
 IntelliSense (Pylance), Linting, Debugging (multi-threaded, r...  
 Microsoft

```

OOP_Python_Projects > exp01.py > ...
1 # My Python
2 name = input('What is your name \n')
3 print ('Hi , %s' % name)
4 print ('Welcome to Python')
  
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell  
 Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```

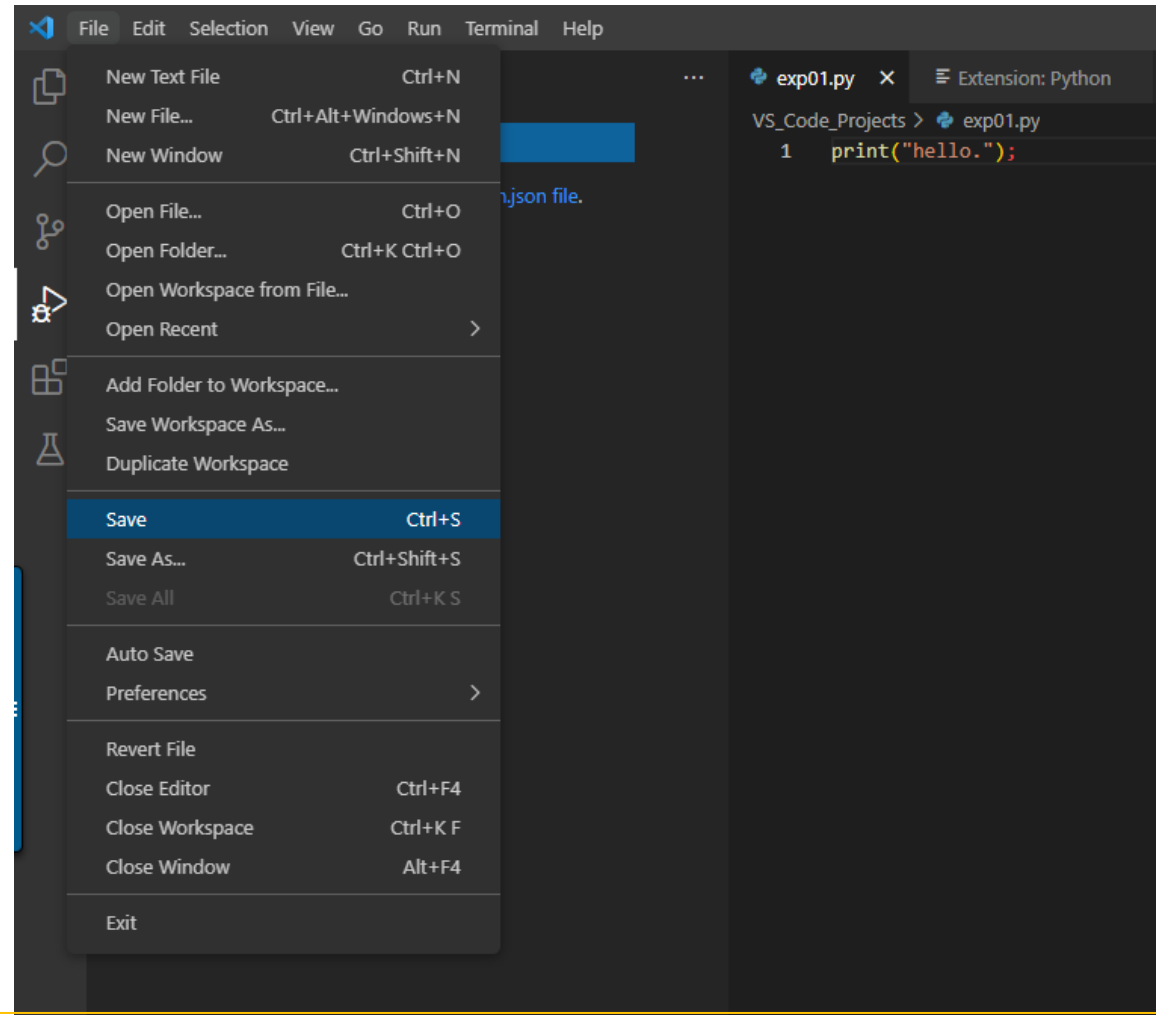
PS D:\OOP_Python_Projects> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe d:/OOP_Python_Projects/exp01.py
What is your name
nattapong
Hi , nattapong
Welcome to Python
PS D:\OOP_Python_Projects>
  
```

⊗ An Invalid Python interpreter is selected, please try changing it ⊗  
 to enable features such as IntelliSense, linting, and debugging.

Source: Python (Extension) Select Python Interpreter

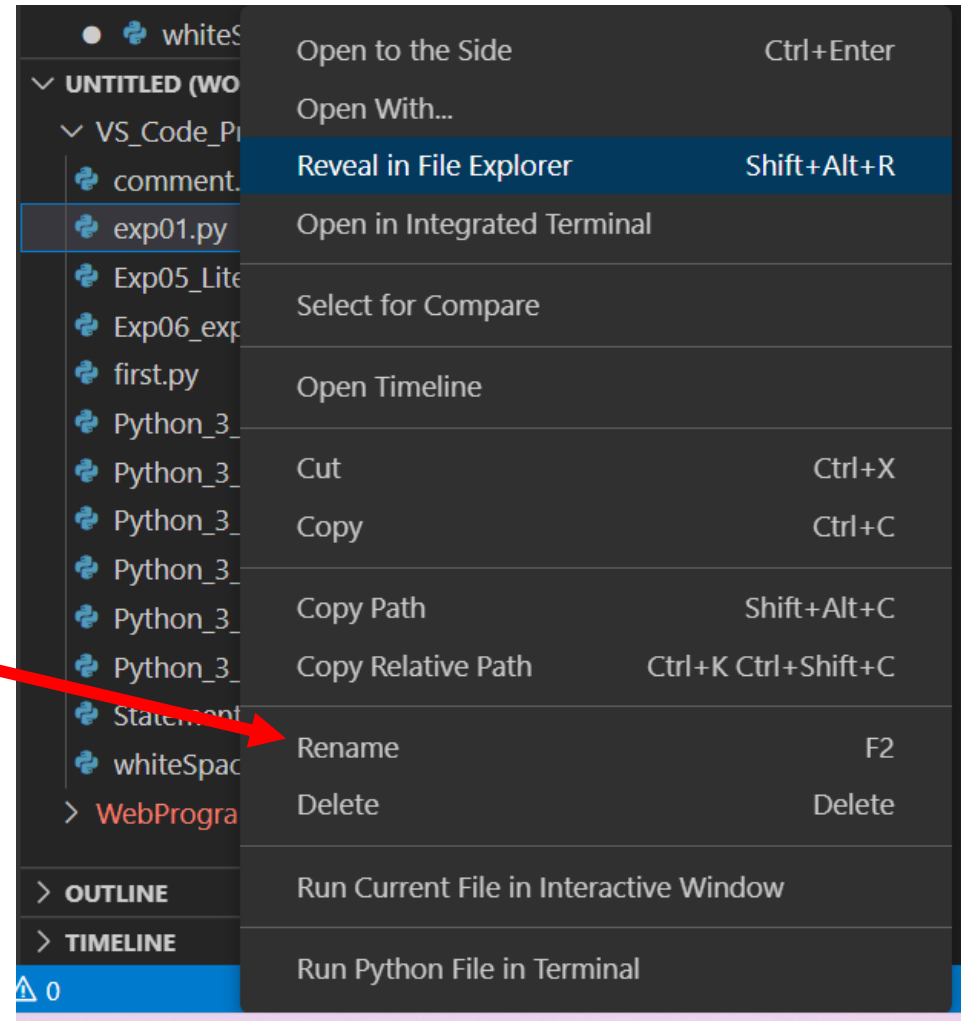
## การบันทึก Save

- Save หรือ
- Save as หรือ
- กด Ctrl + S



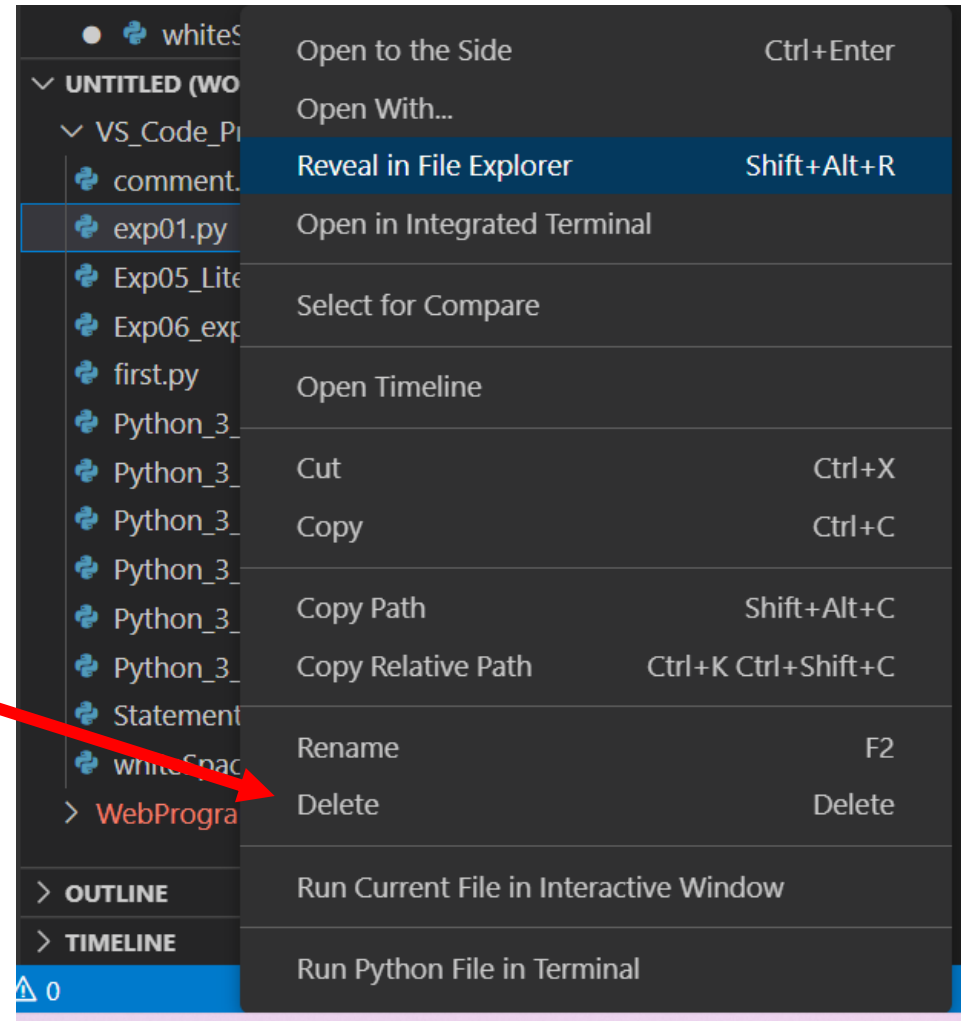
## การเปลี่ยนชื่อไฟล์

- คลิกขวาที่ชื่อไฟล์ที่ต้องการเปลี่ยน
- เลือก Rename
- พิมพ์ชื่อไฟล์ใหม่ที่ต้องการ



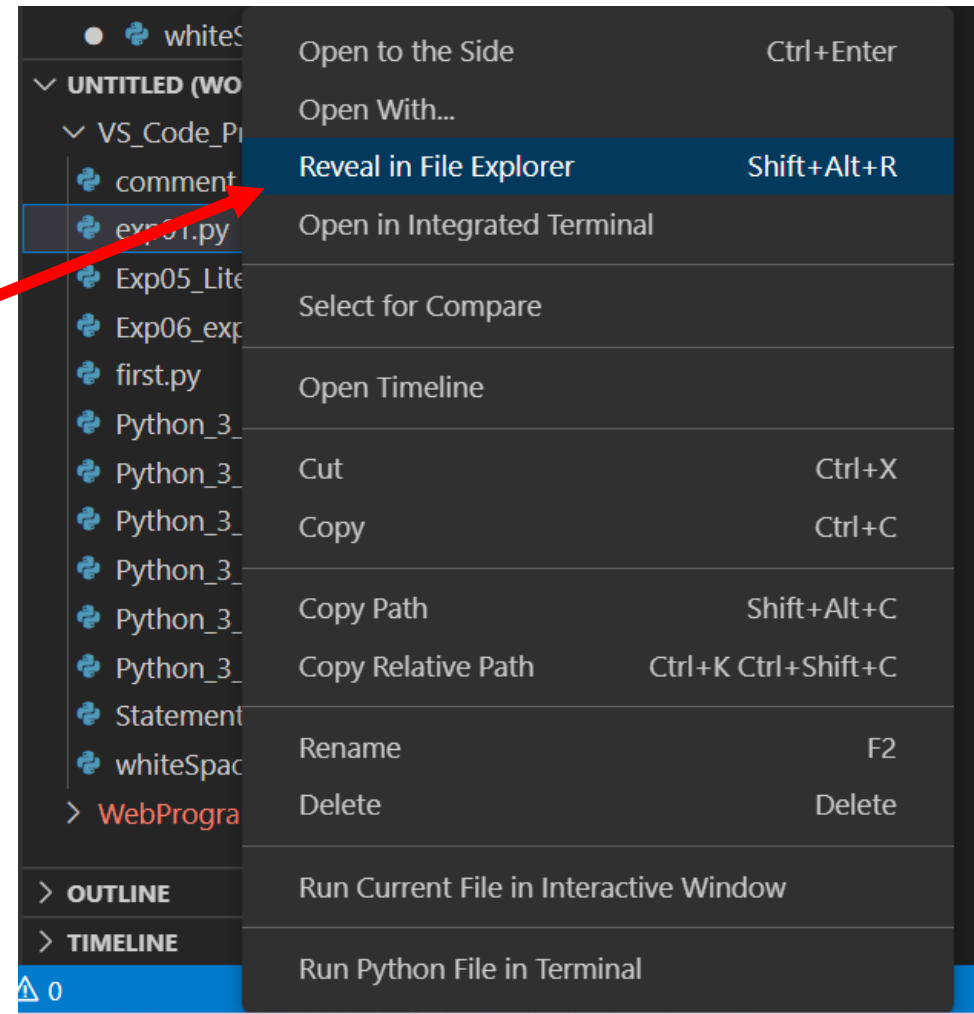
## การลบไฟล์

- คลิกขวาที่ชื่อไฟล์ที่ต้องการลบ
- เลือก Delete



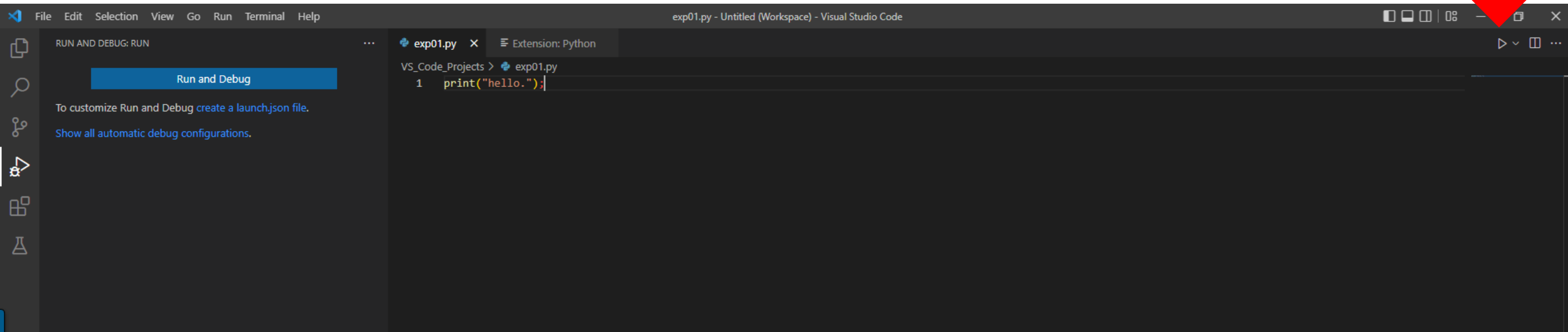
## การเปิดไฟล์ใน Explorer

- คลิกขวาที่ชื่อไฟล์ที่ต้องการ
- เลือก Reveal in File Explorer
- จะปรากฏหน้าต่าง File Explorer ขึ้นมา



# กด RUN

- คลิกที่ปุ่ม RUN ด้านขวาบนดังรูป



### ผลการรัน

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE  PROBLEMS

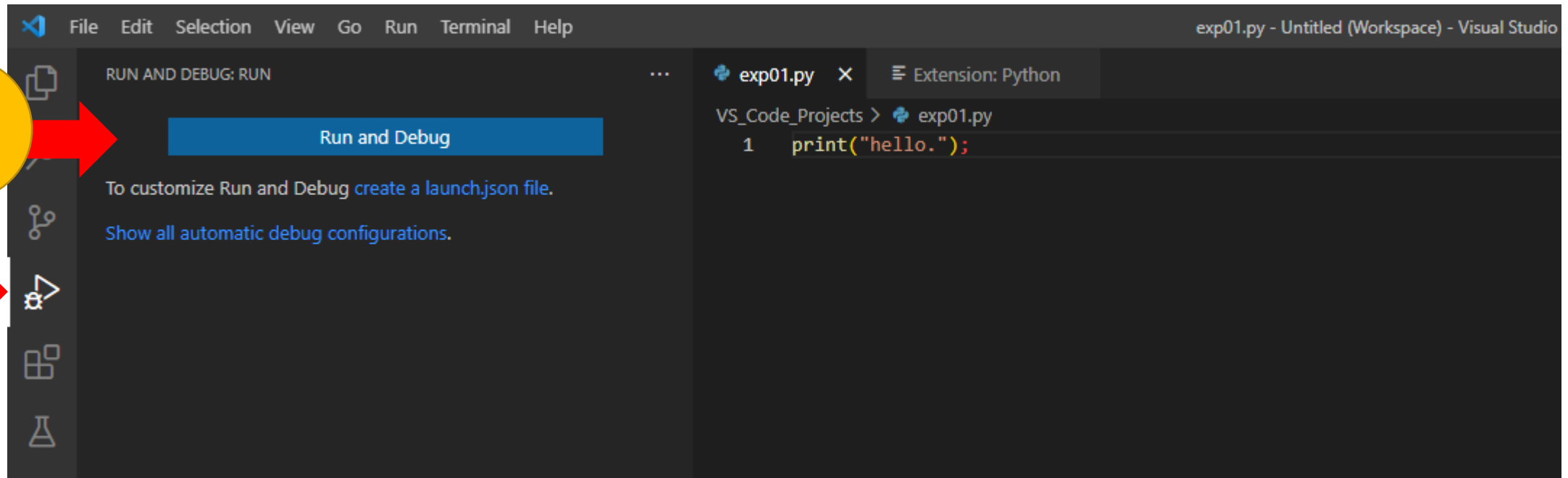
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\VS_Code_Projects> conda activate
PS D:\VS_Code_Projects> & C:/ProgramData/Anaconda3/python.exe d:/VS_Code_Projects/exp01.py
hello.
PS D:\VS_Code_Projects>
```



### กด RUN



EXPLORER ...

NO FOLDER OPENED

You have not yet opened a folder.

[Open Folder](#)

Opening a folder will close all currently open editors. To keep them open, [add a folder](#) instead.

You can clone a repository locally.

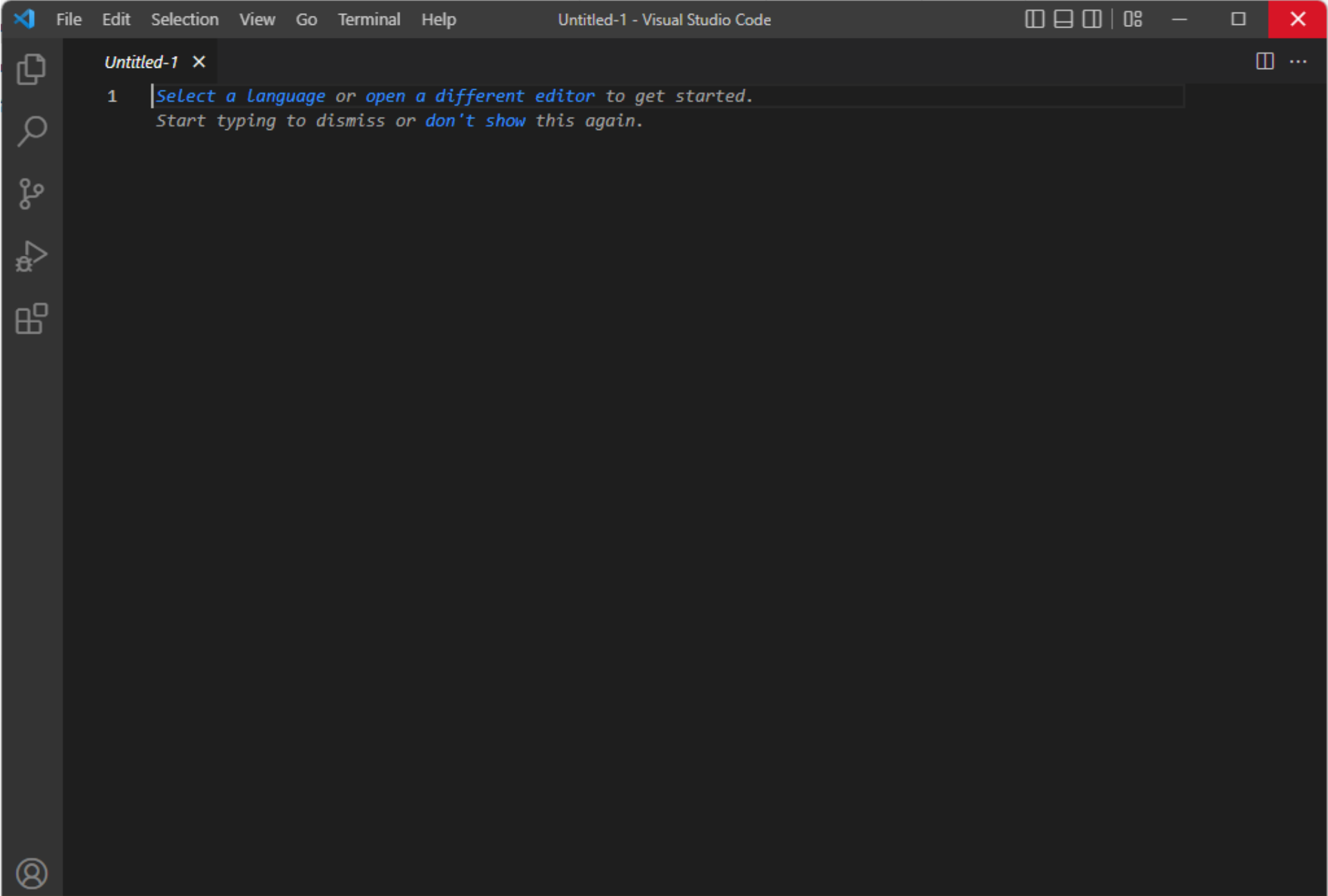
[Clone Repository](#)

To learn more about how to use git and source control in VS Code [read our docs](#).

```

1  Select a language or open a different editor to get started.
   Start typing to dismiss or don't show this again.

```

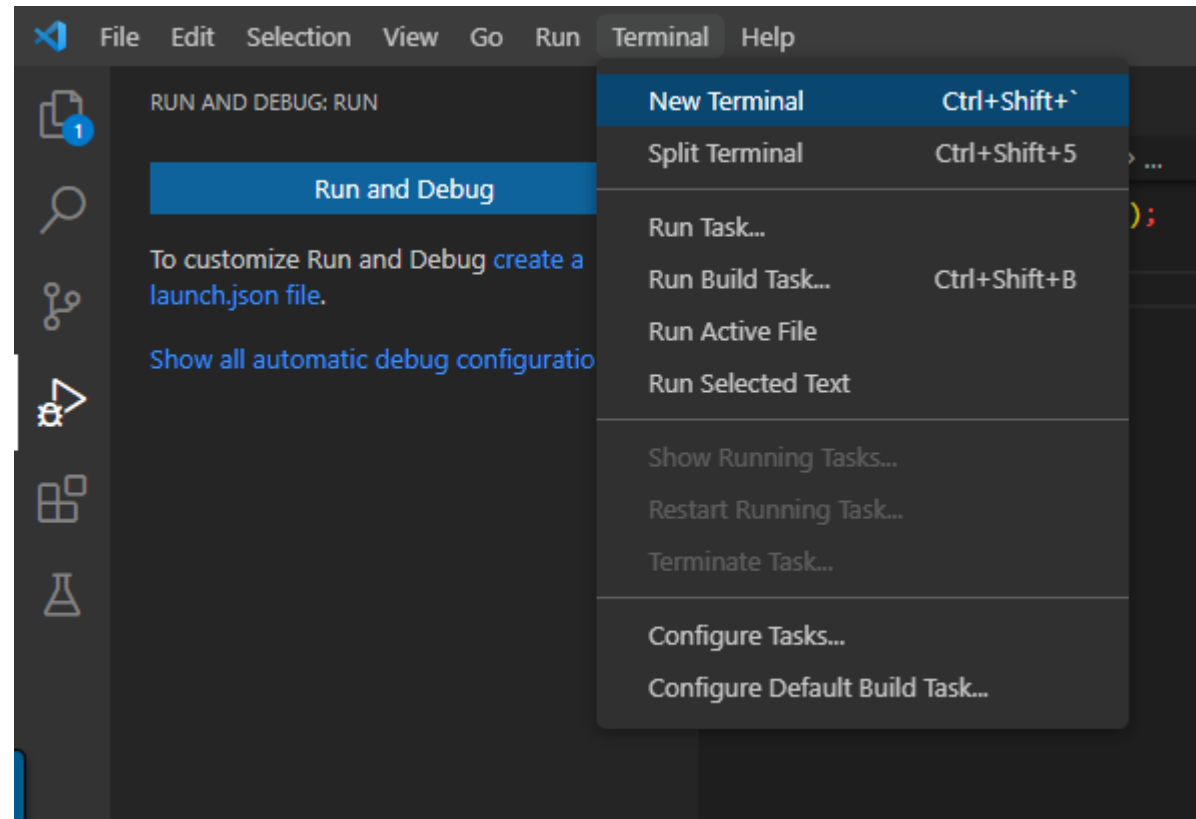


โดย ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่งเนียม



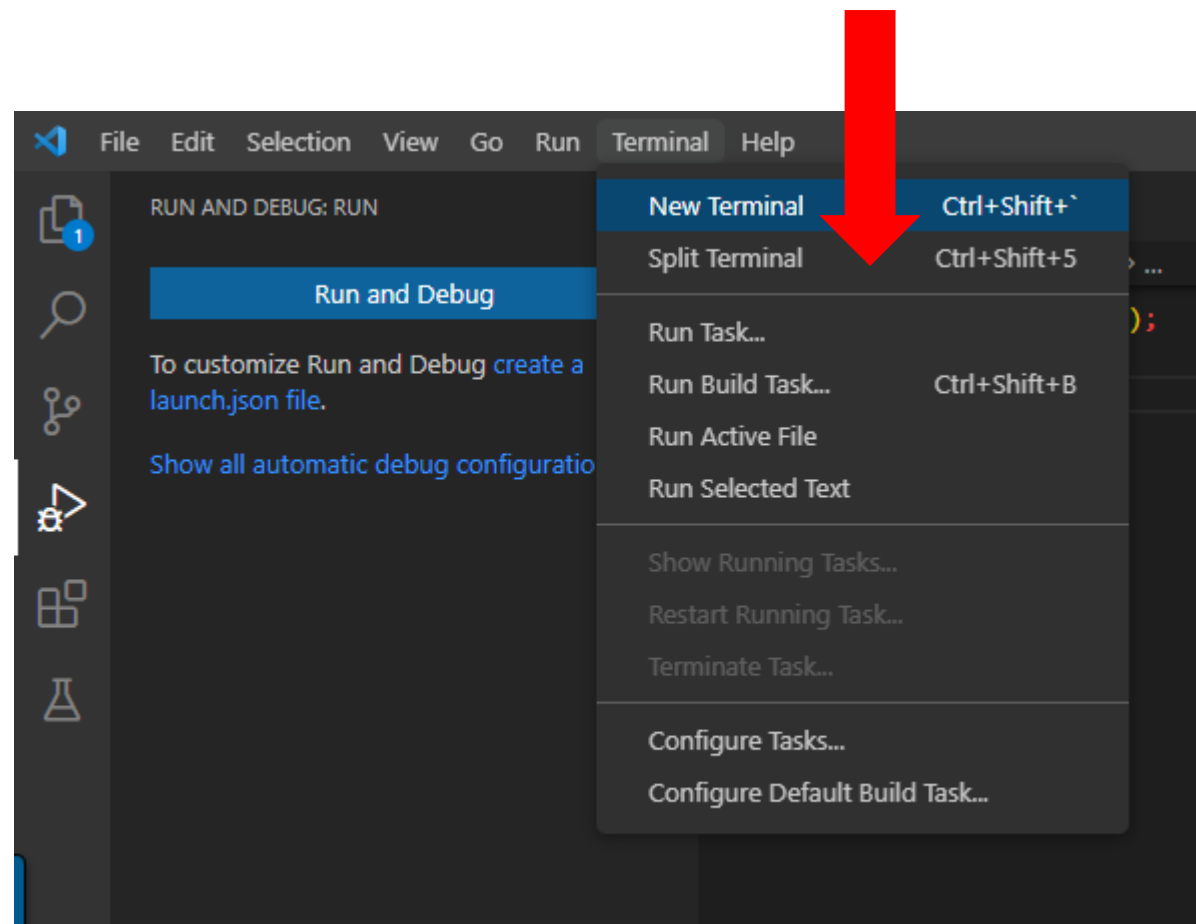
### การใช้งาน Terminal

- การเปิดหน้าต่าง Terminal
  - ไปที่เมนู Terminal
  - เลือก new Terminal



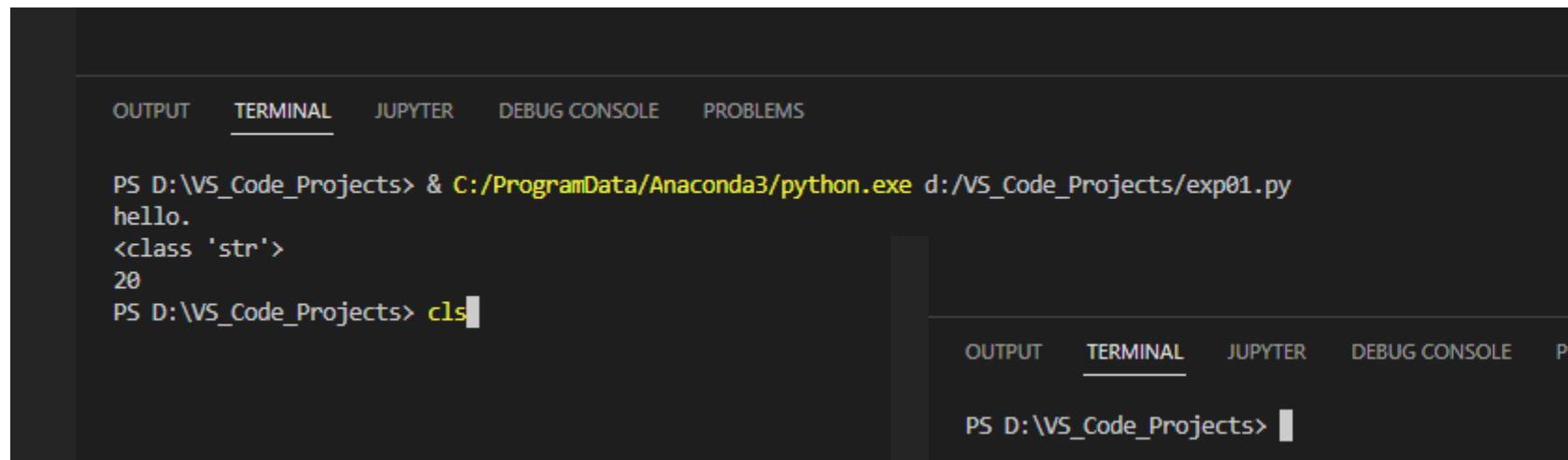
### การใช้งาน Terminal

- การแบ่งหน้าต่าง Terminal
  - ไปที่เมนู Terminal
  - เลือก split Terminal



### การใช้งาน Terminal

- การล้างหน้าจอ
  - Cls
  - Clear



```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE  PROBLEMS

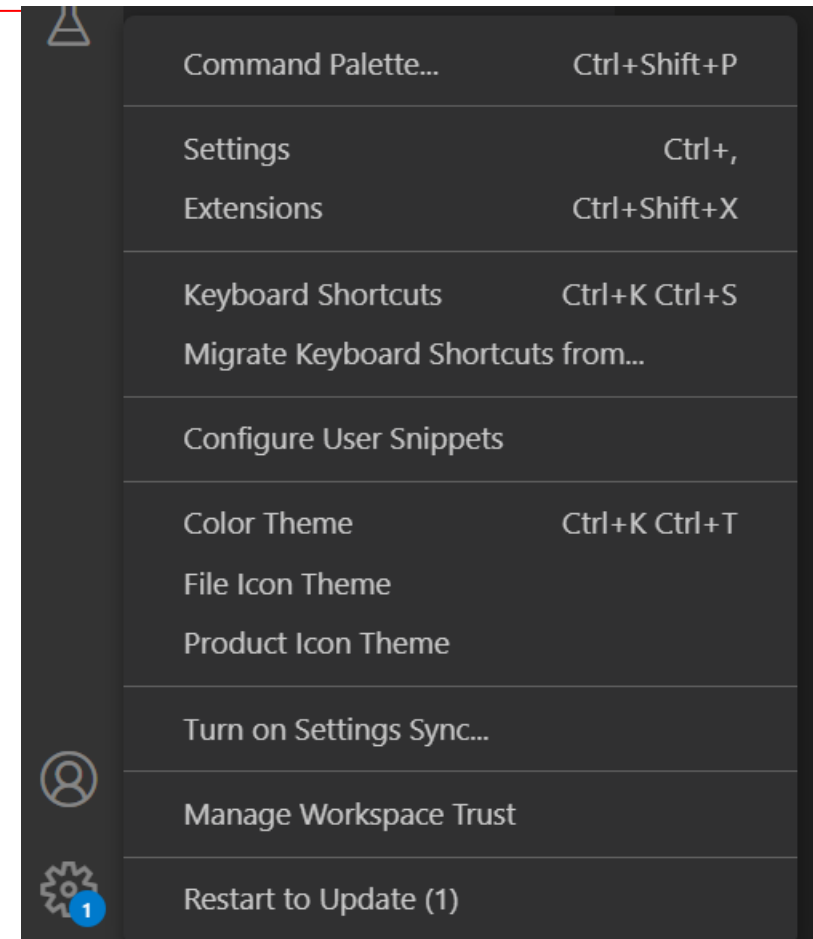
PS D:\VS_Code_Projects> & C:/ProgramData/Anaconda3/python.exe d:/VS_Code_Projects/exp01.py
hello.
<class 'str'>
20
PS D:\VS_Code_Projects> cls
```

```
OUTPUT  TERMINAL  JUPYTER  DEBUG CONSOLE  PROBLEMS

PS D:\VS_Code_Projects> 
```

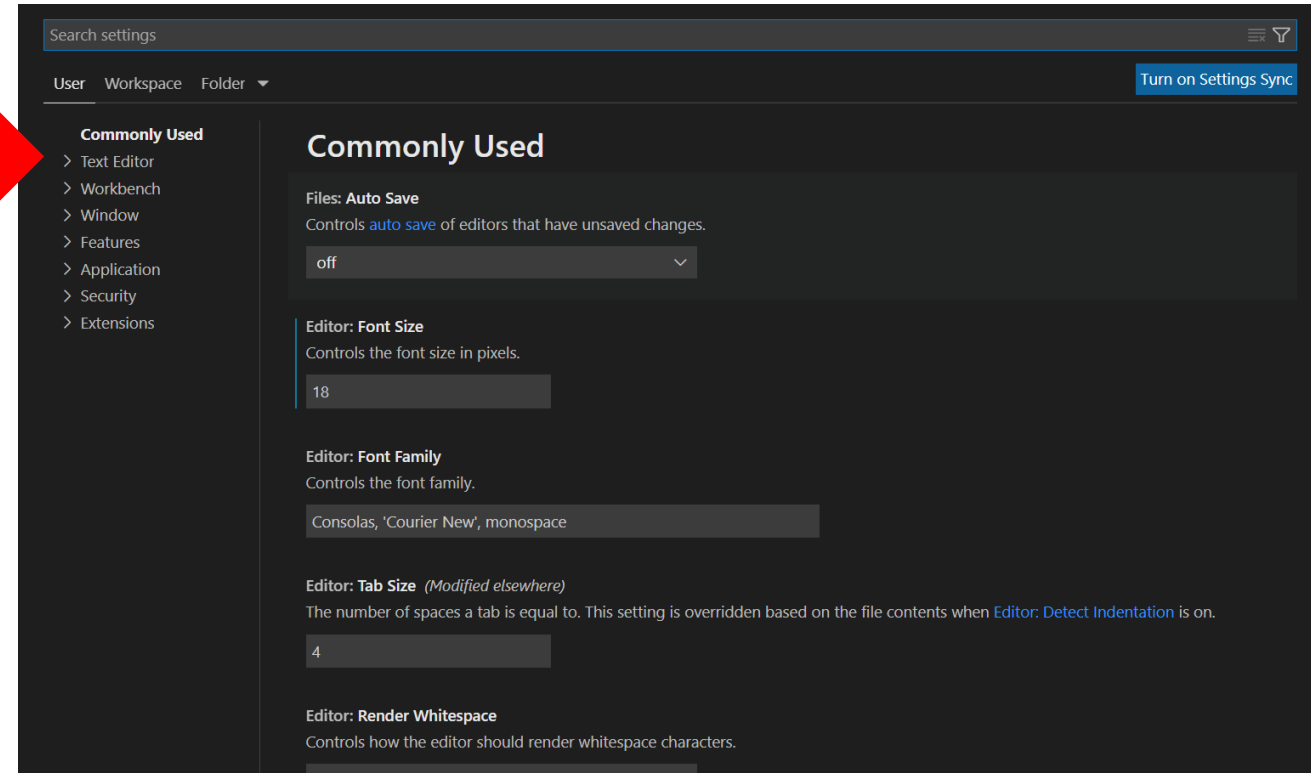
### การเปลี่ยนขนาดตัวอักษรในหน้า Code

1. ไปคลิกที่ปุ่ม 
2. เลือก Setting (หรือกด ctrl+,)
3. จะปรากฏหน้าต่าง Setting ขึ้นมา
4. เลือก Text Editor




## การเปลี่ยนขนาดตัวอักษรในหน้า Code

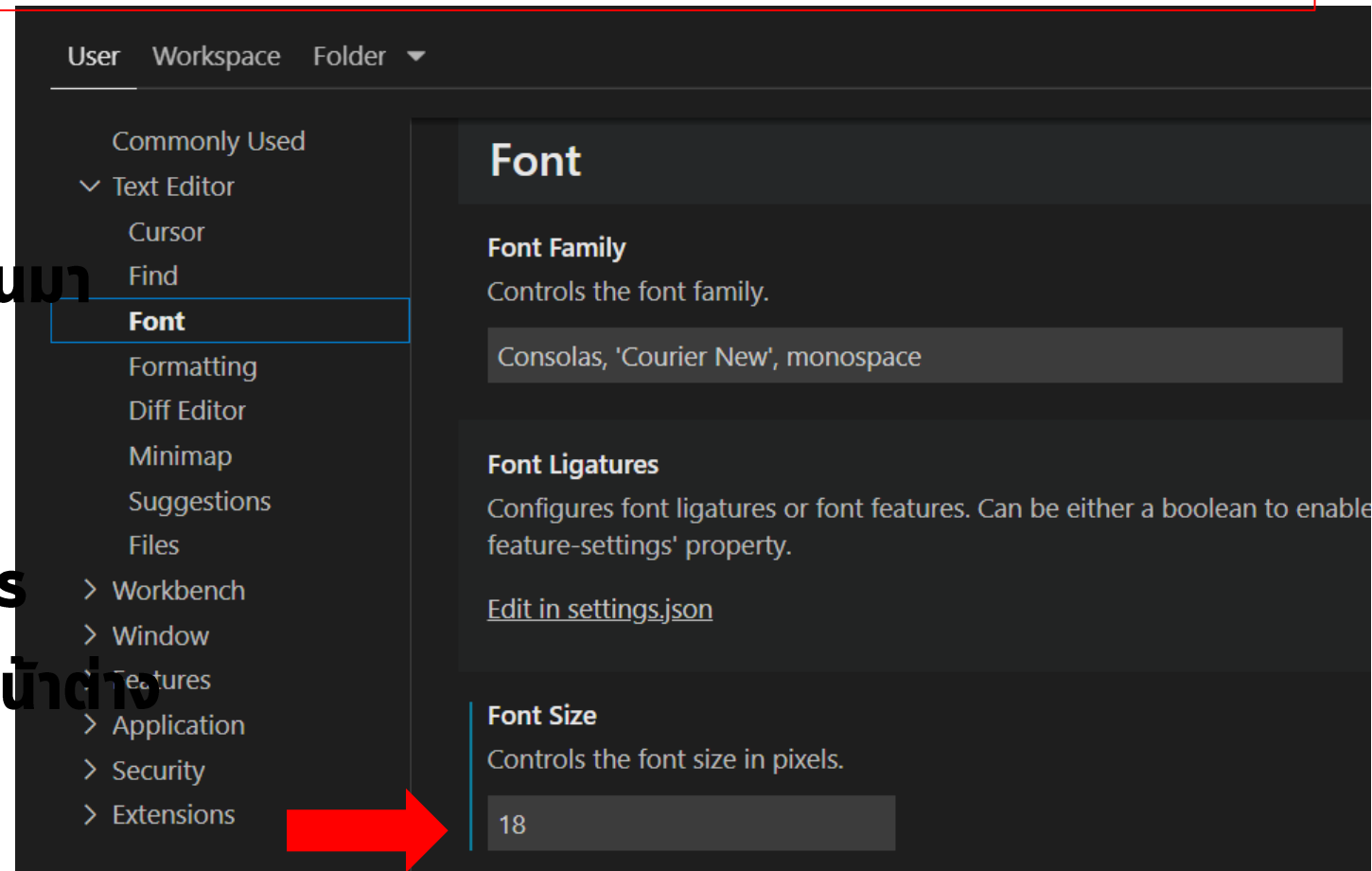
1. ไปคลิกที่ปุ่ม 
2. เลือก Setting
3. จะปรากฏหน้าต่าง Setting ขึ้นมา
4. เลือก Text Editor
5. เลือก Font
6. กำหนดขนาดตัวอักษรที่ต้องการ





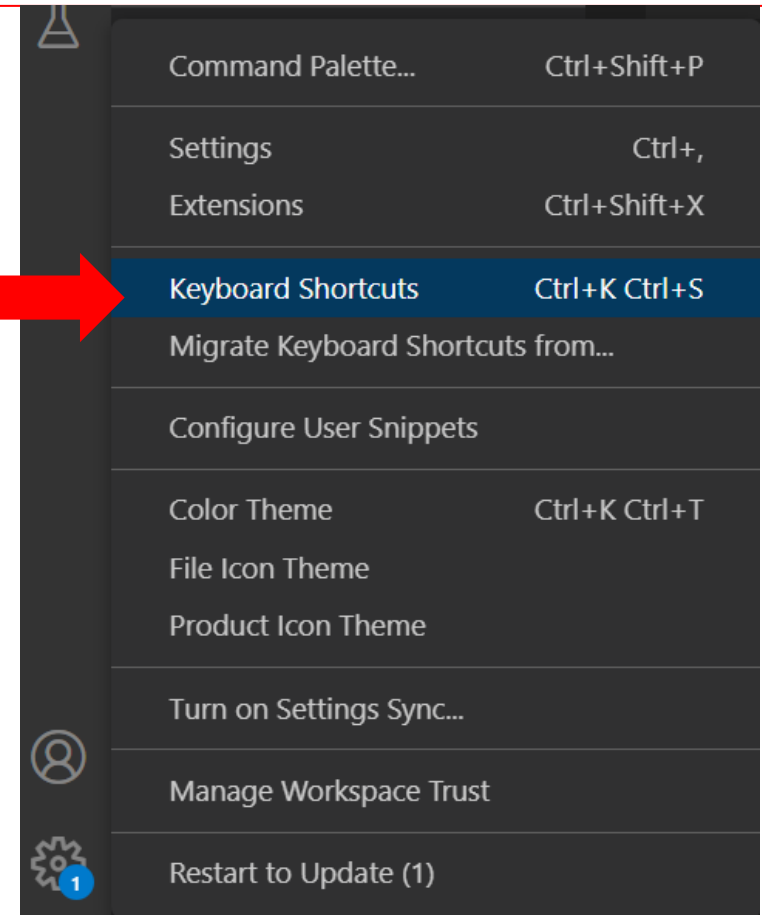
## การเปลี่ยนขนาดตัวอักษรในหน้า Code

1. ไปคลิกที่ปุ่ม 
2. เลือก Setting
3. จะปรากฏหน้าต่าง Setting ขึ้นมา
4. เลือก Text Editor
5. เลือก Font
6. กำหนดขนาดตัวอักษรที่ต้องการ
7. คลิกที่เครื่องหมาย X เพื่อปิดหน้าต่าง



### การดู keyboard shortcut

1. ไปคลิกที่ปุ่ม 
2. เลือก keyboard shortcut
3. จะปรากฏหน้าต่าง keyboard shortcut ขึ้นมา 



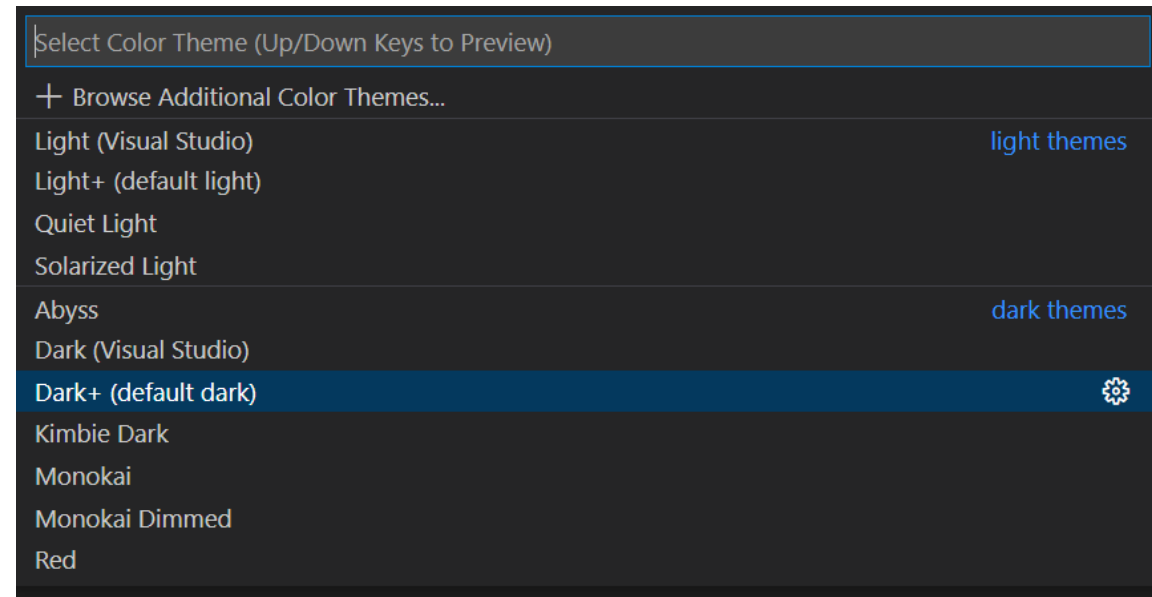
## การดู keyboard shortcut

1. ไปคลิกที่ปุ่ม 
2. เลือก keyboard shortcut
3. จะปรากฏหน้าต่าง keyboard shortcut ขึ้นมา

Command	Keybinding	When	Source
Add Cursor Above	Ctrl + Alt + UpArrow	editorTextFocus	Default
Add Cursor Below	Ctrl + Alt + DownArrow	editorTextFocus	Default
Add Cursors to Line Ends	Shift + Alt + I	editorTextFocus	Default
Add Line Comment	Ctrl + K Ctrl + C	editorTextFocus && !editorReadOnly	Default
Add Selection To Next Find Match	Ctrl + D	editorFocus	Default
Auto Fix...	Shift + Alt + .	editorTextFocus && !editorReadOnly && supportedCodeAc...	Default
Calls: Show Call Hierarchy	Shift + Alt + H	editorHasCallHierarchyProvider	Default
Cancel Selection Anchor	Escape	editorTextFocus && selectionAnchorSet	Default
Change All Occurrences	Ctrl + F2	editorTextFocus && !editorReadOnly	Default
Change Language Mode	Ctrl + K M	InnotebookEditorFocused	Default
Close Exception Widget	Escape	exceptionWidgetVisible	Default
Close Window	Ctrl + Shift + W	-	Default
Close Window	Alt + F4	-	Default
Copy	Ctrl + Insert	-	Default
Copy	Ctrl + C	-	Default
Copy Line Down	Shift + Alt + DownArrow	editorTextFocus && !editorReadOnly	Default
Copy Line Up	Shift + Alt + UpArrow	editorTextFocus && !editorReadOnly	Default
Create: New File...	Ctrl + Alt + Windows + N	-	Default
Cursor Undo	Ctrl + U	textInputFocus	Default
Cut	Shift + Delete	-	Default
Cut	Ctrl + X	-	Default
Debug: Continue	F5	debugState == 'stopped'	Default
Debug: Disconnect	Shift + F5	focusedSessionIsAttach && inDebugMode	Default

### การเปลี่ยน Color Theme

1. ไปคลิกที่ปุ่ม 
2. เลือก Color Theme
3. จะปรากฏหน้าต่าง Color Theme ขึ้นมา
4. เลือก Theme ที่ต้องการ



# โปรแกรมแรก Exp01.py

```
# My first Python program  
name = input('What is your name?\n')
```

```
print  
print
```

ในตัวอย่าง เป็นโปรแกรมในการรับชื่อและ  
แสดงข้อความทักทายออกทางหน้าจอ ในการ  
รันโปรแกรม คุณสามารถทำได้หลายวิธี แต่ที่

exp01.py

exp02.py

VS\_Code\_Projects > exp02.py > ...

```
1 # My first Python program  
2 name = input('What is your name?\n')  
3 print ('Hi, %s.' % name)  
4 print ('Welcome to Python.')
```

# Code และผลการทำงาน

```
exp01.py  exp02.py  ●
VS_Code_Projects > exp02.py > ...
1  # My first Python program
2  name = input('What is your name?\n')
3  print ('Hi, %s.' % name)
4  print ('Welcome to Python.')
```

```
PS D:\VS_Code_Projects> & C:/Users/AS/
cts/exp02.py
What is your name?
nattapong
Hi, nattapong.
Welcome to Python.
PS D:\VS_Code_Projects> █
```

# การ Comment ในภาษาไพธอน

```
# My first Python program

'''
This is a multiline comment
'''

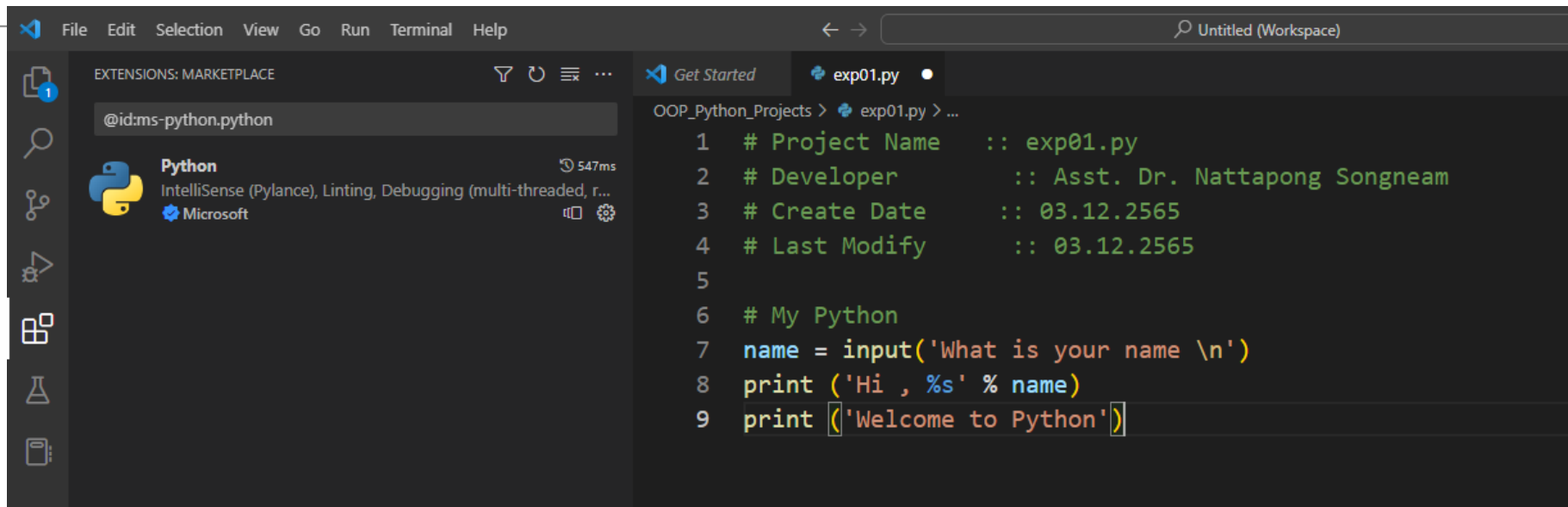
print ('Hello Python.') # Inline comment
```

คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whitespace หรือโค้ดของโปรแกรมหรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายโค้ดที่

```
VS_Code_Projects > exp03.py
● 1 # My first Python program
2
3 '''
4 This is a multiline comment
5 ตรงนี้จะเป็น comment แบบหลายบรรทัด
6 '''
7
8 print ('Hello Python.') # Inline comment
9
```

# ใส่ comment ในตอนต้นของไฟล์ทุกครั้ง

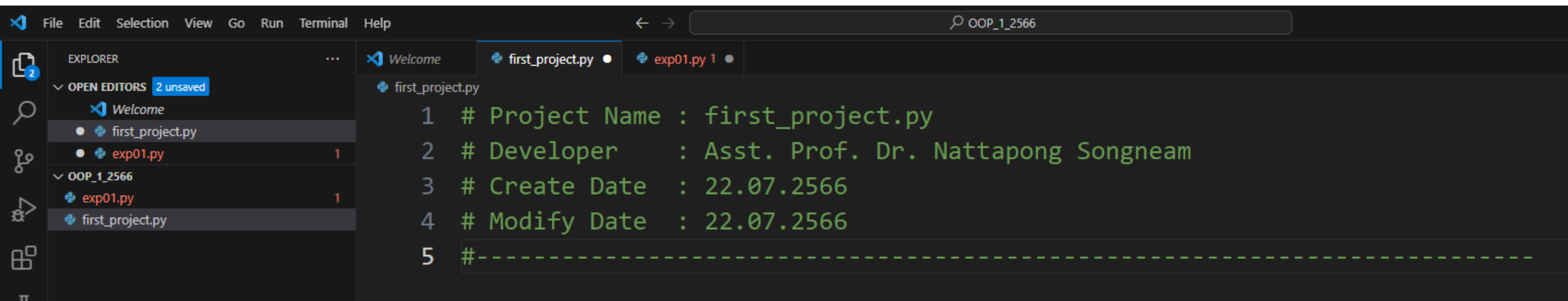
- **# Project Name**    :: **exp01.py**
- **# Developer**        :: **Asst.Prof. Dr. Nattapong Songneam**
- **# Create Date**        :: **03.12.2565**
- **# Last Modify**        :: **15.07.2566**



The screenshot shows a code editor interface with a dark theme. The left sidebar displays the 'EXTENSIONS: MARKETPLACE' section, showing the 'Python' extension by Microsoft. The main editor area shows a file named 'exp01.py' with the following content:

```
1 # Project Name    :: exp01.py
2 # Developer        :: Asst. Dr. Nattapong Songneam
3 # Create Date     :: 03.12.2565
4 # Last Modify     :: 03.12.2565
5
6 # My Python
7 name = input('What is your name \n')
8 print ('Hi , %s' % name)
9 print ('Welcome to Python')
```





# Statement

สร้างไฟล์ใหม่และตั้งชื่อนี้

`Exp02_Statement.py`

```
name = input('What is your name?\n')  
print ('Hi, %s.' % name);  
print ('Welcome to Python.'); print ('Do you love it?')
```

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอน (;) ในการจบคำสั่งการทำงาน แต่อย่างไรก็ตาม ในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้ โดยการใช้เครื่องหมายเซมิโคลอน

# Statement

```
Welcome  first_project.py  exp01.py 1
first_project.py > ...
1 # Project Name : first_project.py
2 # Developer    : Asst. Prof. Dr. Nattapong Songneam
3 # Create Date  : 22.07.2566
4 # Modify Date  : 22.07.2566
5 #-----
6 name = input('What is your name?\n')
7 print ('Hi, %s.' % name);
8 print ('Welcome to Python. '); print ('Do you love it?')
9
```

# ภาษาที่มีการทำงานแบบ Interpreter

- VB , Python
- 1 คำสั่ง 1 บรรทัด ไม่มีเครื่องหมาย ; ในการจบประโยคคำสั่ง เหมือนภาษา C,C++ ,C# ,java

Java, C#

```
int x = 10;
```

```
int x
```

```
=
```

```
10
```

```
;
```

# ประโยค (Statement)

Exp02\_Statement.py

```
name = input('What is your name?\n')  
print ('Hi, %s.' % name);  
print ('Welcome to Python.'); print ('Do you love it?')
```

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอน (;) ในการจบคำสั่งการทำงาน แต่อย่างไรก็ตาม ในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้ โดยการใช้เครื่องหมายเซมิโคลอน

## Indentation and while space

สร้างไฟล์ใหม่และตั้งชื่อ

Exp03\_whiteSpace.py

```
n = int(input ('Input an integer: '))

if (n > 0):
    print ('x is positive number')
    print ('Show number from 0 to %d' % (n - 1))

else:
    print ('x isn\'t positive number')

for i in range(n):
    print(i)
```

ในตัวอย่าง โปรแกรมของเรานั้นประกอบไปด้วย 3 บล็อกที่สร้างจาก 3 คำสั่ง คำสั่งแรกคือ If ในบล็อกนี้มีสองคำสั่งย่อยอยู่ภายใน ที่หัวของบล็อกจะต้องมีเครื่องหมายโคลอน (:) กำหนดหลังคำสั่ง if ในการเริ่มต้นบล็อกเสมอ อีกสองบล็อกสุดท้ายนั้นเป็นคำสั่ง else และ for ซึ่งแต่ละบล็อกมีหนึ่งคำสั่งย่อยอยู่ภายใน ในภาษา Python นั้นเข้มงวดกับช่องว่างภายในบล็อกมาก นั้นหมายความว่าทุกคำสั่งย่อยภายในบล็อกนั้นต้องมีจำนวนช่องว่างเท่ากันเสมอ

# Indentation and while space

```
exp01.py > ...
1 # Project Name : exp01.py
2 # Developer : Asst. Prof. Dr. Nattapong Songneam
3 # Create Date : 22.07.2566
4 # Modify Date : 22.07.2566
5 #-----
6 n = int(input('Plz. Input an Integer : ')) # การ casting เป็นชนิดจำนวนเต็ม
7 if (n > 0):
8     print ('n is positive number')
9     print ('Show number from 0 to %d' % (n - 1))
10 else:
11     print ('n isn\'t positive number')
12
13 for i in range(n):
14     print(i)
15
```

# Indentation and while space

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\OOP_1_2566> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe d:/OOP_1_2566/exp01.py
Plz. Input an Integer : 5
n is positive number
Show number from 0 to 4
0
1
2
3
4
PS D:\OOP_1_2566> █
```



# Indentation and while space

## VS-Code

VS\_Code\_Projects > exp05.py > ...

```
• 1 n = int(input ('Input an integer: '))
  2
  3 √ if (n > 0):
  4     print ('x is positive number')
  5     print ('Show number from 0 to %d' % (n - 1))
  6
  7 √ else:
  8     print ('x isn\'t positive number')
  9
 10 √ for i in range(n):
 11     print(i)
 12
```

```
PS D:\VS_Code_Projects> & C:/
cts/exp05.py
Input an integer: 5
x is positive number
Show number from 0 to 4
0
1
2
3
4
PS D:\VS_Code_Projects>
```

# Literals

## Exp03\_Literals.py

```
VS_Code_Projects > Exp05_Literals.py > ...  
1 a = 1  
2 b = -1.64E3  
3 c = True  
4 d = "http://www.siam2dev.net"  
5 e = 'A'  
6 print (d);
```

ในการเขียนโปรแกรม Literal คือตัวอักษรหรือเครื่องหมายที่ใช้แสดงค่าของค่าคงที่ในโปรแกรม ในภาษา Python นั้นมี Literal ของข้อมูลประเภทต่างๆ เช่น Integer Floating-point number และ String หรือแม้กระทั่งตัวอักษรและ boolean นี้เป็นตัวอย่างของการกำหนด Literal ให้กับตัวแปรในภาษาไพธอน

# Expressions

## exp04\_expression.py

```
1 a = 4
2 b = 5
3 # Boolean expressions
4 print(a == 4)
5 print(a == 5)
6 print(a == 4 and b == 5)
7 print(a == 4 and b == 8)
8 # Non-boolean expressions
9 print(a + b)
10 print(a + 2)
11 print(a * b)
12 print(((a * a) + (b * b)) / 2)
13 print("Python " + "Language")
14
```

```
cts/Exp06_expression.py
True
False
True
False
9
6
20
20.5
Python Language
```

### Expression คือการ

แปร (หรือค่าคงที่) และตัวดำเนินการ โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงาน ในภาษาไพธอนนั้นมี Expression อยู่สองแบบ แบบแรกคือ Boolean expression เป็นการกระทำกันระหว่างตัวแปรและตัวดำเนินการเปรียบเทียบค่าหรือตัวดำเนินการตรรกศาสตร์ และจะได้ผลลัพธ์เป็น Boolean

และแบบที่สองคือ Expression ทางคณิตศาสตร์ เป็นการกระทำกันระหว่างตัวแปรและตัวดำเนินการคณิตศาสตร์ และจะได้รับค่าใหม่เป็นตัวเลขหรือค่าที่ไม่ใช่ Boolean นี่เป็นตัวอย่างของ Expressions ในภาษา Python

## การรับค่าและการแสดงผลบนหน้าจอ

Keyword เป็นคำที่ถูกสงวนไว้ในการเขียนโปรแกรมภาษาไพธอน จะไม่สามารถใช้คำสั่งเหล่านี้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน คลาส หรือตัวระบุชื่อ ใด ๆ ที่กำหนดขึ้นโดยโปรแกรมเมอร์ นี่เป็นรายการของ Keyword ในภาษาไพธอน

<b>False</b>	<b>None</b>	<b>True</b>	<b>and</b>
<b>as</b>	assert	break	class
<b>continue</b>	def	del	elif
<b>else</b>	except	finally	for
<b>from</b>	global	if	import
<b>in</b>	is	lambda	nonlocal
<b>not</b>	or	pass	raise
<b>return</b>	try	while	with
<b>yield</b>			

## Keywords

การรับค่าและการแสดงผลพื้นฐานในภาษาไพธอน ซึ่งเป็นสิ่งสำคัญที่โปรแกรมต้องมีสำหรับการติดต่อกับผู้ใช้ การรับค่าคือการรับข้อมูลจากภายนอกโดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ส่วนการแสดงผลนั้นจะเป็นบน Console ถ้าหากคุณรันไพธอนบนเว็บเซิร์ฟเวอร์ การแสดงผลจะเป็นทางเว็บเบราว์เซอร์ และการรับค่าจะเป็นทาง URI หรือเว็บฟอร์ม

### การแสดงผลด้วยฟังก์ชัน print()

รูปแบบ

```
print(value, ..., sep = ' ', end = '\n');
```

## การรับค่าและการแสดงผลในภาษาไพธอน

การรับค่าและการแสดงผลพื้นฐานในภาษไพธอน ซึ่งเป็นสิ่งสำคัญที่โปรแกรมต้องมีสำหรับการติดต่อกับผู้ใช้ การรับค่าคือการรับข้อมูลจากภายนอกโดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ส่วนการแสดงผลนั้นจะเป็นบน Console ถ้าหากคุณรันไพธอนบนเว็บเซิร์ฟเวอร์ การแสดงผลจะเป็นทางเว็บเบราว์เซอร์ และการรับค่าจะเป็นทาง URI หรือเว็บฟอร์ม

การแสดงผลด้วยฟังก์ชัน print()

### Exp05.py

```
VS_Code_Projects > Python_3_01.py > ...
1 print("Hello Python")
2 print("My name is Mateo")
3 print("Mercury", "Venus", "Earth")
4 name = "marcuscode.com"
5 year = 2017
6 print(name)
7 print(year)
8
```

### ผลการทำงาน

```
PS D:\VS_Code_Projects> & C:\
cts/Python_3_01.py
Hello Python
My name is Mateo
Mercury Venus Earth
marcuscode.com
2017
PS D:\VS_Code_Projects> |
```

```
1 # Project Name : exp05_print.py
2 # Developer    : Asst. Prof. Dr. Nattapong Songneam
3 # Create Date  : 22.07.2566
4 # Modify Date  : 22.07.2566
```

```
5 #-----
```

```
6 print('Hello Python')
```

```
7 Invalid character "\ue37" in token Pylance
```

```
8 View Problem \(Alt+F8\) No quick fixes available
```

```
9 website_name = "www.siam2dev.com"
```

```
10
```

exp05.py > ...

```
1 # Project Name : exp05_print.py
2 # Developer    : Asst. Prof. Dr. Nattapong Songneam
3 # Create Date  : 22.07.2566
4 # Modify
5 #-----
6 print('He
7 print("My
8 print("วัน
9 web_name
10 year = 2017
11 print(web)
12
13
```

(\*values: object, sep: str  
| None = " ", end: str |  
None = "\n", file:  
SupportsWrite[str] | None  
= None, flush:  
Literal[False] = False) ->

site ไม่ได้

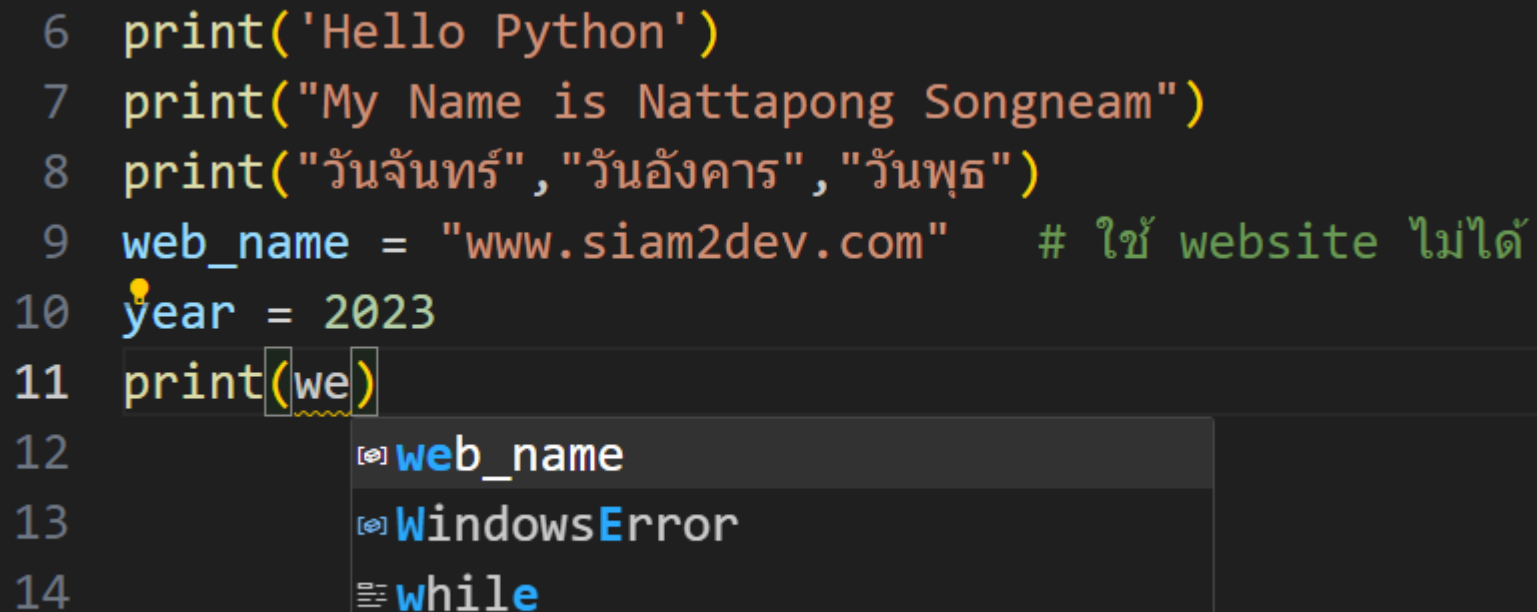
[e] web\_name >



# กด TAB

- สำหรับพิมพ์ตัวแปรแบบอัตโนมัติ

```
6 print('Hello Python')
7 print("My Name is Nattapong Songneam")
8 print("วันจันทร์", "วันอังคาร", "วันพุธ")
9 web_name = "www.siam2dev.com" # ใช้ website ไม่ได้
10 year = 2023
11 print(web_name)
12
13
14
```



The image shows a code editor with a Python script. The script contains several lines of code, including a variable assignment for 'web\_name' and a function call 'print(web\_name)'. A dropdown menu is open below the function call, showing suggestions for 'web\_name', 'WindowsError', and 'while'. The 'web\_name' suggestion is highlighted, indicating that the code editor has recognized the variable and is suggesting its value.

exp05.py > ...

```
1 # Project Name : exp05_print.py
2 # Developer   : Asst. Prof. Dr. Nattapong Songneam
3 # Create Date  : 22.07.2566
4 # Modify Date  : 22.07.2566
5 #-----
6 print('Hello Python')
7 print("My Name is Nattapong Songneam")
8 print("วันจันทร์", "วันอังคาร", "วันพุธ")
9 web_name = "www.siam2dev.com" # ใช้ website ไม่ได้
10 year = 2023
11 print(web_name)
12 print(year)
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\OOP_1_2566> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.10.exe d:/OOP_1_2566/exp05.py
Hello Python
My Name is Nattapong Songneam
วันจันทร์ วันอังคาร วันพุธ
www.siam2dev.com
2023
PS D:\OOP_1_2566> |
```

```
# Project Name : exp05_print.py
```

```
1 # Project Name : exp05_print.py
2 # Developer   : Asst. Prof. Dr. Nattapong Songneam
3 # Create Date : 22.07.2566
4 # Modify Date : 22.07.2566
5 #-----
6 print('Hello Python')
7 print("My Name is Nattapong Songneam")
8 print("วันจันทร์","วันอังคาร","วันพุธ")
9 web_name = "www.siam2dev.com" # ใช้ website ไม่ได้
10 year = 2023
11 print(web_name)
12 print(year)
```

## การรับค่าและการแสดงผลในภาษาไพธอน

นอกจากนี้ เรายังสามารถใช้ keyword อาร์กิวเมนต์สำหรับกำหนดการแสดงผลเพื่อแบ่งแต่ละอาร์กิวเมนต์ และการแสดงผลในตอนท้ายของฟังก์ชัน นี่เป็นตัวอย่างการใช้งาน

การแสดงผลด้วยฟังก์ชัน print()

**Exp08.py**

**ผลการทำงาน**

```
VS_Code_Projects > Python_3_02.py
1 print("Mercury", "Venus", "Earth", sep=', ')
2 print("One", end=' ')
3 print("Two", end=' ')
4 print("Three", end=' ')
5
```

```
PS D:\VS_Code_Projects> & C:/Users/
cts/Python_3_02.py
Mercury, Venus, Earth
One Two Three
PS D:\VS_Code_Projects> █
```

# การรับค่าและการแสดงผลในภาษาไพธอน

นอกจากนี้ฟังก์ชัน `print()` นั้นยังสามารถส่งพารามิเตอร์ในรูปแบบของ String Formatting ได้โดยการใช้รูปแบบการแปลงข้อมูล ที่คล้ายกับการจัดรูปแบบการแสดงผลในภาษา C นี่เป็นตัวอย่างการใช้งาน

## การแสดงผลด้วยฟังก์ชัน `print()`

### Exp09.py

```
VS_Code_Projects > Python_3_03.py > ...
1 lang = "Python"
2 version = 3.6
3 print("%s language" % lang)
4 print("Version %f" % version)
5 print("%d" % 123)
6 print("%s %f %d" % (lang, version, 123))
7
```

### ผลการทำงาน

```
OUTPUT  TERMINAL  JUPYTER  DEBU
PS D:\VS_Code_Projects> & C:/U
cts/Python_3_03.py
Python language
Version 3.600000
123
Python 3.600000 123
PS D:\VS_Code_Projects> █
```

# การรับค่าจาก Keyboard ด้วยฟังก์ชัน input()

การติดต่อกับผู้ใช้ในอีกรูปแบบหนึ่งคือการรับค่า โดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ในภาษา Python เราใช้ฟังก์ชัน input() สำหรับการรับค่า String จากทางคีย์บอร์ด มาดูตัวอย่างการรับค่าจากผู้ใช้ในภาษาไพธอน

การรับข้อมูลด้วยฟังก์ชัน input()

**Exp10.py**

```
VS_Code_Projects > Python_3_05.py > ...  
1 name = input("Enter your name: ")  
2 print("Hello " + name)  
3
```

**ผลการทำงาน**

```
PS D:\VS_Code_Projects> & C:/Users/  
cts/Python_3_05.py  
Enter your name: nattapong  
Hello nattapong  
PS D:\VS_Code_Projects> █
```

# การรับค่าจาก Keyboard ด้วยฟังก์ชัน input()

อย่างไรก็ตาม ในการรับค่าด้วยฟังก์ชัน input() นั้นจะใช้กับการรับค่าที่เป็น String เท่านั้น และในการที่จะรับข้อมูลประเภทอื่นๆ เช่น ตัวเลข เราสามารถใช้ฟังก์ชันที่มากับภาษาไพธอน ในการแปลงข้อมูลจาก String ไปเป็นข้อมูลประเภทอื่นได้ ต่อไปมาดูตัวอย่างการรับค่าตัวเลขในภาษาไพธอน

## การรับข้อมูลด้วยฟังก์ชัน input()

### Exp11.py

### ผลการทำงาน

```
VS_Code_Projects > Python_3_06.py > ...
```

```
1 a = int(input("Enter first number: "))
2 b = int(input("Enter second number: "))
3 print("a + b = %d" % (a + b))
4
```

```
PS D:\VS_Code_Projects> & C:/Python_3_06.py
Enter first number: 1
Enter second number: 2
a + b = 3
PS D:\VS_Code_Projects>
```

## 3.6

# Agenda

- 1) ไวยากรณ์ของภาษา Python
- 2) เริ่มต้นเขียนโปรแกรมภาษาไพธอนด้วย VS-Code





### 3.6.1 ไวยากรณ์ของภาษา Python

ภาษาไพธอนถูกพัฒนาขึ้นมาโดยมีความตั้งใจว่าจะให้เป็นภาษาที่อ่านง่าย มันถูกออกแบบมาให้มีโครงสร้างที่ไม่ซับซ้อน โดยมักจะใช้คำในภาษาอังกฤษในขณะที่ภาษาอื่นใช้เครื่องหมายและวรรคตอน นอกจากนี้ Python ยังมีข้อยกเว้นของโครงสร้างทางภาษาน้อยกว่าอื่น ๆ อย่างภาษา C และ Pascal เป็นต้น

### 3.6.2 ตัวแปลภาษาของไพธอน (Python Interpreter)

Python interpreter เป็นตัวแปลภาษาของภาษา Python เพื่อให้สามารถรันโค้ดไพธอนได้ ซึ่งได้มากับไลบรารีมาตรฐานที่สามารถใช้งานได้ฟรี ซึ่งดาวน์โหลดได้ที่ <https://www.python.org/> มันเป็นโปรแกรมแบบ source และ binary ที่มีให้ดาวน์โหลดในแพลตฟอร์มที่ได้รับความนิยม นอกจากนี้ interpreter ยังสนับสนุนการเขียนโปรแกรมกับ Interactive shell ซึ่งเป็นการเขียนโค้ดภาษา Python และเห็นผลลัพธ์การทำงานของคำสั่งได้ในทันที นอกจากนี้ Python interpreter ยังสามารถนำมาเขียนร่วมกับฟังก์ชันใหม่ที่ถูกพัฒนามาจากภาษา C และ C++ เพื่อเพิ่มความสามารถการทำงานของภาษา Python และเหมาะสำหรับเป็นภาษาในการสร้าง Extension และแอปพลิเคชันที่ปรับแต่งได้

### 3.6.3 โครงสร้างของภาษาไพธอน (Python Structure)

ในบทนี้ คุณจะได้เรียนรู้และทำความเข้าใจเกี่ยวกับโครงสร้างของภาษาไพธอน ในภาษาคอมพิวเตอร์นั้นมีโครงสร้างของภาษาเช่นเดียวกับภาษาของมนุษย์เรา ซึ่งสิ่งเหล่านี้ถูกกำหนดเพื่อเป็นรูปแบบและวิธีในการเขียนโปรแกรมในภาษาไพธอน มันใช้สำหรับควบคุมวิธีที่คุณจะเขียนโค้ดของคุณเพื่อให้เข้าใจโดยตัวแปลภาษาหรือคอมไพเลอร์

### 3.6.4 Module

ในตัวอย่างโปรแกรมรับชื่อ เป็นโปรแกรมแรกของเราในบทเรียน Python นี้ เราได้บันทึกเป็นไฟล์ที่ชื่อว่า *first.py* ซึ่งไฟล์ของภาษาไพธอน นั้นจะเรียกว่า Module ซึ่ง Module จะประกอบไปด้วยคลาส ฟังก์ชัน และตัวแปรต่างๆ และนอกจากนี้เรายังสามารถ import โมดูลอื่นเข้ามาในโปรแกรมได้ ซึ่งโมดูลอาจจะอยู่ภายใน package ซึ่งเป็นเหมือนไฟล์เดอร์ที่เอาไว้จัดกลุ่มของ Module ในตัวอย่าง *first.py* จึงเป็นโมดูลของโปรแกรมแรกของเรา

### 3.6.5 Comment

คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whitespace หรือโค้ดของโปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายข้อผิดพลาดที่เราเขียนขึ้นและมันไม่มีผลต่อการทำงานของโปรแกรม นี่เป็นตัวอย่างการคอมเมนต์ในภาษา Python

#### ไฟล์ Comment.py

```
VS_Code_Projects > comment.py
1  # My first Python program with Python
2  '''
3      This is a multiline comment
4      การหมายเหตุแบบหลายบรรทัด
5  '''
6  print ('Hello Python.') # Inline comment
7
```

#### ผลการทำงาน

```
PS D:\VS_Code_Projects> & C:/U
cts/comment.py
Hello Python.
PS D:\VS_Code_Projects>
```



### 3.6.7 WhiteSpace

ในภาษา Python นั้นใช้ White space และ Tab สำหรับกำหนดบล็อกของโปรแกรม เช่น คำสั่ง If Else For หรือการประกาศฟังก์ชัน ซึ่งคำสั่งเหล่านี้เป็นคำสั่งแบบบล็อก โดยจำนวนช่องว่างที่ใช้ันั้นต้องเท่ากัน มาตรฐานของบล็อกคำสั่งในภาษาไพธอน

#### ไฟล์ whiteSpace.py

```
VS_Code_Projects > whiteSpace.py > ...
1  n = int(input ('Input an integer: '))
2  if (n > 0):
3      print ('x is positive number')
4      print ('Show number from 0 to %d' % (n - 1))
5  else:
6      print ('x isn\'t positive number')
7  for i in range(n):
8      print(i)
9
```

#### ผลการทำงาน

```
Show number from 0 to 1
PS D:\VS_Code_Projects> & C:/Use
cts/whiteSpace.py
Input an integer: 5
x is positive number
Show number from 0 to 4
0
1
2
3
4
```

### 3.6.7 WhiteSpace

ตัวอย่างการเว้นช่องไม่เท่ากันในคำสั่ง if

ไฟล์ `whiteSpace.py`

ผลการทำงาน

```
VS_Code_Projects > whiteSpace.py > ...
1  n = int(input ('Input an integer: '))
2  if (n > 0):
3      print ('x is positive number')
4      print ('Show number from 0 to %d' % (n - 1))
5  else:
6      print ('x isn\'t positive number')
7  for i in range(n):
8      print(i)
9
```



## เริ่มต้นเขียนโปรแกรมภาษาไพธอนด้วย VS-Code

- เปิดโปรแกรม Vs-Code ขึ้นมา
- สร้างไฟล์ชื่อ `first.py`

### 3.7.1 โปรแกรมแรก **First.py**

```
# My first Python program
name = input('What is your name?\n')
print ('Hi, %s.' % name)
print ('Welcome to Python.')
```

ในตัวอย่าง เป็นโปรแกรมในการรับชื่อและแสดงข้อความทักทายออกทางหน้าจอ ในการรันโปรแกรม คุณสามารถรันได้หลายวิธี แต่ที่แนะนำคือการใช้ Python shell ให้คุณเปิด Python shell ขึ้นมาแล้วกดสร้างไฟล์ใหม่โดยไปที่ *File -> New File* จะปรากฏกล่อง Text editor ของภาษา Python ขึ้นมาเพื่อรันโปรแกรม **Run -> Run Module** หรือกด **F5** โปรแกรมจะเปลี่ยนกลับไปยัง Python shell และเริ่มต้นทำงาน

## 3.7.2 การ Comment ในภาษาไพธอน

```
# My first Python program  
  
...  
This is a multiline comment  
...  
  
print ('Hello Python.') # Inline comment
```

คอมเมนต์ในภาษา Python นั้นเริ่มต้นด้วยเครื่องหมาย # คอมเมนต์สามารถเริ่มต้นที่ตำแหน่งแรกของบรรทัดและหลังจากนั้นจะประกอบไปด้วย Whitespace หรือโค้ดของโปรแกรม หรือคำอธิบาย ซึ่งโดยทั่วไปแล้วคอมเมนต์มักจะใช้สำหรับอธิบายข้อผิดพลาดที่เราเขียนขึ้นและมันไม่มีผลต่อการทำงานของโปรแกรม นี่เป็นตัวอย่างการคอมเมนต์ในภาษา Python

### 3.7.3 ประโยค (Statement)

Exp02\_Statement.py

```
name = input("What is your name?\n")
print ('Hi, %s.' % name);
print ("Welcome to Python."); print ("Do you love it?")
```

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอน (;) ในการจบคำสั่งการทำงาน แต่อย่างไรก็ตาม ในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน

### 3.7.4 ประโยค (Statement)

Exp02\_Statement.py

```
name = input('What is your name?\n')
print ('Hi, %s.' % name);
print ('Welcome to Python.');
```

Statement คือคำสั่งการทำงานของโปรแกรม แต่ละคำสั่งในภาษา Python นั้นจะแบ่งแยกด้วยการขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจากภาษา C และ Java ซึ่งใช้เครื่องหมายเซมิโคลอน (;) ในการจบคำสั่งการทำงาน แต่อย่างไรก็ตาม ในภาษา Python นั้นคุณสามารถมีหลายคำสั่งในบรรทัดเดียวกันได้โดยการใช้เครื่องหมายเซมิโคลอน

## 3.7.5 Indentation and while space

### Exp03\_whiteSpace.py

```
n = int(input ('Input an integer: '))

if (n > 0):
    print ('x is positive number')
    print ('Show number from 0 to %d' % (n - 1))

else:
    print ('x isn\'t positive number')

for i in range(n):
    print(i)
```

ในตัวอย่าง โปรแกรมของเรานั้นประกอบไปด้วย 3 บล็อกที่สร้างจาก 3 คำสั่ง คำสั่งแรกคือ If ในบล็อกนี้มีสองคำสั่งย่อยอยู่ภายใน ที่หัวของบล็อกจะต้องมีเครื่องหมายโคลอน (: ) กำหนดหลังคำสั่ง if ในการเริ่มต้นบล็อกเสมอ อีกสองบล็อกสุดท้ายนั้นเป็นคำสั่ง else และ for ซึ่งแต่ละบล็อกมีหนึ่งคำสั่งย่อยอยู่ภายใน ในภาษา Python นั้นเข้มงวดกับช่องว่างภายในบล็อกมาก นั้นหมายความว่าทุกคำสั่งย่อยภายในบล็อกนั้นต้องมีจำนวนช่องว่างเท่ากันเสมอ

## 3.7.6 Literals

### Exp05\_Literals.py

```
VS_Code_Projects > Exp05_Literals.py > ...
```

```
1 a = 1
2 b = -1.64E3
3 c = True
4 d = "http://www.siam2dev.net"
5 e = 'A'
6 print (d);
```

ในการเขียนโปรแกรม Literal คือ ตัวอักษรหรือเครื่องหมายที่ใช้แสดงค่าของค่าคงที่ในโปรแกรม ในภาษา Python นั้นมี Literal ของข้อมูลประเภทต่างๆ เช่น Integer Floating-point number และ String หรือแม้กระทั่งตัวอักษรและ boolean นี่เป็นตัวอย่างของการกำหนด Literal ให้กับตัวแปรในภาษาไพธอน

## 3.7.7 Expressions

Exp06\_expression.py

```
1 a = 4
2 b = 5
3 # Boolean expressions
4 print(a == 4)
5 print(a == 5)
6 print(a == 4 and b == 5)
7 print(a == 4 and b == 8)
8 # Non-boolean expressions
9 print(a + b)
10 print(a + 2)
11 print(a * b)
12 print(((a * a) + (b * b)) / 2)
13 print("Python " + "Language")
14
```

```
cts/Exp06_expression.py
True
False
True
False
9
6
20
20.5
Python Language
```

### Expression คือการ

แปร (หรือค่าคงที่) และตัวดำเนินการ โดยค่าเหล่านี้จะมีตัวดำเนินการสำหรับควบคุมการทำงาน ในภาษาไพธอนนั้นมี Expression อยู่สองแบบ แบบแรกคือ Boolean expression เป็นการกระทำกันระหว่างตัวแปรและตัวดำเนินการเปรียบเทียบค่าหรือตัวดำเนินการตรรกศาสตร์ และจะได้ผลลัพธ์เป็น Boolean

และแบบที่สองคือ Expression ทางคณิตศาสตร์ เป็นการกระทำกันระหว่างตัวแปรและตัวดำเนินการคณิตศาสตร์ และจะได้รับค่าใหม่เป็นตัวเลขหรือค่าที่ไม่ใช่ Boolean นี่เป็นตัวอย่างของ Expressions ในภาษา Python



### 3.7.8 การรับค่าและการแสดงผลบนหน้าจอ

Keyword เป็นคำที่ถูกสงวนไว้ในการเขียนโปรแกรมภาษาไพธอน จะไม่สามารถใช้คำสั่งเหล่านี้ในการตั้งชื่อตัวแปร ชื่อฟังก์ชัน คลาส หรือตัวระบุชื่อ ใด ๆ ที่กำหนดขึ้นโดยโปรแกรมเมอร์ นี่เป็นรายการของ Keyword ในภาษาไพธอน

<b>False</b>	<b>None</b>	<b>True</b>	<b>and</b>
<b>as</b>	assert	break	class
<b>continue</b>	def	del	elif
<b>else</b>	except	finally	for
<b>from</b>	global	if	import
<b>in</b>	is	lambda	nonlocal
<b>not</b>	or	pass	raise
<b>return</b>	try	while	with
<b>yield</b>			

### 3.7.9 การแสดงผลด้วยฟังก์ชัน print()

การรับค่าและการแสดงผลพื้นฐานในภาษาไพธอน ซึ่งเป็นสิ่งสำคัญที่โปรแกรมต้องมีการติดต่อกับผู้ใช้ การรับค่าคือการรับข้อมูลจากภายนอกโดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ส่วนการแสดงผลนั้นจะเป็นบน Console ถ้าหากคุณรันไพธอนบนเว็บเซิร์ฟเวอร์ การแสดงผลจะเป็นทางเว็บเบราว์เซอร์แทน และการรับค่าจะเป็นทาง URI หรือเว็บฟอร์ม

#### การแสดงผลด้วยฟังก์ชัน print()

รูปแบบ

```
print(value, ..., sep = ' ', end = '\n');
```

## อักขระพิเศษ escape character

- `\n` การขึ้นบรรทัดใหม่
- `\t` TAB
- `\'`
  - `print(' x is\'t positive number')`

## 3.7.10 การรับค่าและการแสดงผลในภาษาไพธอน

การรับค่าและการแสดงผลพื้นฐานในภาษาไพธอน ซึ่งเป็นสิ่งสำคัญที่โปรแกรมต้องมีการติดต่อกับผู้ใช้ การรับค่าคือการรับข้อมูลจากภายนอกโดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ส่วนการแสดงผลนั้นจะเป็นบน Console ถ้าหากคุณรันไพธอนบนเว็บเซิร์ฟเวอร์ การแสดงผลจะเป็นทางเว็บเบราว์เซอร์ และการรับค่าจะเป็นทาง URI หรือเว็บฟอร์ม

การแสดงผลด้วยฟังก์ชัน `print()`

### Python\_3\_01.py

```
VS_Code_Projects > Python_3_01.py > ...
1 print("Hello Python")
2 print("My name is Mateo")
3 print("Mercury", "Venus", "Earth")
4 name = "marcuscode.com"
5 year = 2017
6 print(name)
7 print(year)
8
```

### ผลการทำงาน

```
PS D:\VS_Code_Projects> & C:\
cts/Python_3_01.py
Hello Python
My name is Mateo
Mercury Venus Earth
marcuscode.com
2017
PS D:\VS_Code_Projects>
```

### 3.7.11 การรับค่าและการแสดงผลในภาษาไพธอน

นอกจากนี้ เรายังสามารถใช้ keyword อาร์กิวเมนต์สำหรับกำหนดการแสดงผลเพื่อแบ่งแต่ละอาร์กิวเมนต์ และการแสดงผลในตอนที่ท้ายของฟังก์ชัน นี่เป็นตัวอย่างการใช้งาน

การแสดงผลด้วยฟังก์ชัน `print()`

**Python\_3\_02.py**

**ผลการทำงาน**

```
VS_Code_Projects > Python_3_02.py
1 print("Mercury", "Venus", "Earth", sep=', ' )
2 print("One", end=' ')
3 print("Two", end=' ')
4 print("Three", end=' ')
5
```

```
PS D:\VS_Code_Projects> & C:/Users/...
cts/Python_3_02.py
Mercury, Venus, Earth
One Two Three
PS D:\VS_Code_Projects> █
```

### 3.7.12 การรับค่าและการแสดงผลในภาษาไพธอน

นอกจากนี้ฟังก์ชัน `print()` นั้นยังสามารถส่งพารามิเตอร์ในรูปแบบของ String Formatting ได้โดยการใช้รูปแบบการแปลงข้อมูล ที่คล้ายกับการจัดรูปแบบการแสดงผลในภาษา C นี่เป็นตัวอย่างการใช้งาน

การแสดงผลด้วยฟังก์ชัน `print()`

#### Python\_3\_03.py

```
VS_Code_Projects > Python_3_03.py > ...
1 lang = "Python"
2 version = 3.6
3 print("%s language" % lang)
4 print("Version %f" % version)
5 print("%d" % 123)
6 print("%s %f %d" % (lang, version, 123))
7
```

#### ผลการทำงาน

```
OUTPUT  TERMINAL  JUPYTER  DEBU
PS D:\VS_Code_Projects> & C:/U
cts/Python_3_03.py
Python language
Version 3.600000
123
Python 3.600000 123
PS D:\VS_Code_Projects>
```

## 3.7.12 การรับค่าและการแสดงผลในภาษาไพธอน

นอกจากนี้ฟังก์ชัน `print()` นั้นยังสามารถส่งพารามิเตอร์ในรูปแบบของ String Formatting ได้โดยการใช้รูปแบบการแปลงข้อมูล ที่คล้ายกับการจัดรูปแบบการแสดงผลในภาษา C นี่เป็นตัวอย่างการใช้งาน

### การแสดงผลด้วยฟังก์ชัน

`print()`

<code>%s</code>	พิมพ์ข้อความ
<code>%d</code>	พิมพ์จำนวนเต็ม
<code>%f</code>	พิมพ์ทศนิยม

### 3.7.13 การรับค่าจาก Keyboard ด้วยฟังก์ชัน input()

การติดต่อกับผู้ใช้ในอีกรูปแบบหนึ่งคือการรับค่า โดยทั่วไปแล้วมักจะเป็นการรับค่าทางคีย์บอร์ด ในภาษา Python เราใช้ฟังก์ชัน input() สำหรับการรับค่า String จากทางคีย์บอร์ด มาดูตัวอย่างการรับค่าจากผู้ใช้ในภาษาไพธอน

การรับข้อมูลด้วยฟังก์ชัน input()

#### Python\_3\_05.py

```
VS_Code_Projects > Python_3_05.py > ...  
1 name = input("Enter your name: ")  
2 print("Hello " + name)  
3
```

#### ผลการทำงาน

```
PS D:\VS_Code_Projects> & C:/Users/...  
cts/Python_3_05.py  
Enter your name: nattapong  
Hello nattapong  
PS D:\VS_Code_Projects>
```



### 3.7.13 การรับค่าจาก Keyboard ด้วยฟังก์ชัน input()

อย่างไรก็ตาม ในการรับค่าด้วยฟังก์ชัน input() นั้นจะใช้กับการรับค่าที่เป็น String เท่านั้น และในการที่จะรับข้อมูลประเภทอื่นๆ เช่น ตัวเลข เราสามารถใช้ฟังก์ชันที่มากับภาษาไพธอน ในการแปลงข้อมูลจาก String ไปเป็นข้อมูลประเภทอื่นได้ ต่อไปมาดูตัวอย่างการรับค่าตัวเลขในภาษาไพธอน

การรับข้อมูลด้วยฟังก์ชัน input()

**Python\_3\_06.py**

**ผลการทำงาน**

```
VS_Code_Projects > Python_3_06.py > ...
```

```
1 a = int(input("Enter first number: "))
2 b = int(input("Enter second number: "))
3 print("a + b = %d" % (a + b))
4
```

```
PS D:\VS_Code_Projects> & C:\Python3\python.exe D:\VS_Code_Projects\Python_3_06.py
Enter first number: 1
Enter second number: 2
a + b = 3
PS D:\VS_Code_Projects>
```

### 3.7.14 ตัวอย่างการวิเคราะห์โจทย์ปัญหา

1. จงเขียนโปรแกรมเพื่อรับค่าตัวเลขมา 3 จำนวน แล้วหาค่าเฉลี่ย
2. จงเขียนโปรแกรมเพื่อคำนวณหาพื้นที่สามเหลี่ยม
3. จงเขียนโปรแกรมเพื่อคำนวณหาพื้นที่วงกลม
4. จงเขียนโปรแกรมเพื่อปริมาตรทรงกระบอก
5. จงคำนวณหาจำนวนวินาทีใน 1 ปี
6. จงคำนวณหาค่า ปีแสง (Light year)

**\*\*\* ส่งใน google drive ในแบบฝึกหัด บทที่ 3**

### 3.7.14 ตัวอย่างการวิเคราะห์โจทย์ปัญหา

1. จงเขียนโปรแกรมเพื่อรับค่าตัวเลขมา 3 จำนวน แล้วหาค่าเฉลี่ย
  1. สิ่งที่ต้องการ : ค่าเฉลี่ยของตัวเลข 3 จำนวน
  2. รูปแบบผลลัพธ์ : ค่าเฉลี่ยของตัวเลข 3 จำนวน เท่ากับ **XX.XX**
  3. ข้อมูลนำเข้า ตัวเลขใด ๆ 3 จำนวน
  4. ตัวแปรที่ใช้ num1 , num2 และ num3 แทน ตัวเลขทั้ง 3 จำนวน
  5. วิธีการประมวลผล
    1. เริ่มต้น
    2. กำหนดตัวแปร 2.1 num1 , num2, num3 เป็นจำนวนเต็ม / ทศนิยม  
2.2 กำหนด sum เป็นจำนวนเต็ม
    3. รับค่าตัวแปรแต่ละตัว
    4. หาผลรวมตัวเลข num1 + num2 + num3 เก็บไว้ใน sum
    5. หาค่าเฉลี่ย avg = sum /3
    6. แสดงผล avg
  2. สิ้นสุด

### 3.7.14 ตัวอย่างการวิเคราะห์โจทย์ปัญหา

#### 1. จงเขียนโปรแกรมเพื่อรับค่าตัวเลขมา 3 จำนวน แล้วหาค่าเฉลี่ย

```
exe01.py ×
VS_Code_Projects > PL1 > exe01.py > ...
1 # exe01.py
2 num1 = input("Plz. Enter number #1")
3 num2 = input("Plz. Enter number #2")
4 num3 = input("Plz. Enter number #3")
5 sum = num1 + num2 + num3
6 print (sum)
```

### Logical Error ?

```
Windows PowerShell
Copyright (C) Microsoft Corporation

Install the latest PowerShell for all versions:
https://aka.ms/WindowsPowerShell

PS D:\VS_Code_Projects> & C:/Users/.../Python/Scripts/PL1/exe01.py
Plz. Enter number #1 1
Plz. Enter number #2 2
Plz. Enter number #3 3
123
```

## Code

### 1. จงเขียนโปรแกรมเพื่อรับค่าตัวเลขมา 3 จำนวน แล้วหาค่าเฉลี่ย

```
VS_Code_Projects > PL1 > exe01.py > ...
```

```
1 # exe01.py
2 num1 = int(input("Plz. Enter number #1 : "))
3 num2 = int(input("Plz. Enter number #2 : "))
4 num3 = int(input("Plz. Enter number #3 : "))
5 sum = num1 + num2 + num3
6 avg = sum / 3
7 print (avg)
```

```
PS D:\VS_Code_Projects> & C:/
cts/PL1/exe01.py
Plz. Enter number #1 : 1
Plz. Enter number #2 : 2
Plz. Enter number #3 : 3
2.0
PS D:\VS_Code_Projects> █
```

# แบบฝึกหัด บทที่ 3

1. จงเขียน Code แนะนำตัวเอง โดยมีลักษณะของผลลัพธ์ดังนี้

Name : Nattapong Songneam

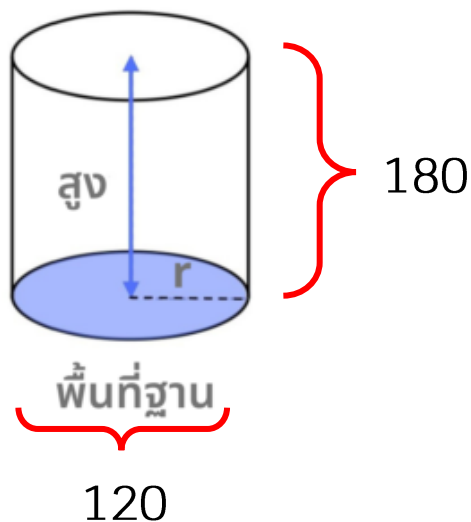
Student ID : 6510122115000

Tel : 099-087-4433

Email : [siam2dev@hotmail.com](mailto:siam2dev@hotmail.com)

2. จงเขียน Code หาปริมาตรทรงกระบอก ดังนี้

$$\text{ปริมาตร} = \pi r^2 h$$



## อ้างอิง

<https://www.9experttraining.com/articles/python-คืออะไร>

<https://medium.com/@moungsiri/โครงสร้างภาษา-python-53cc38a51462>

<http://marcuscode.com/lang/python/variables-and-types>

<http://cms576.bps.in.th/group11/introduction-to-computer-programming>

<https://www.mindphp.com/บทเรียนออนไลน์/83-python/2398-python-operator.html>

# **การใช้งาน Google Colab เบื้องต้น ในการเขียนโปรแกรมภาษา Python**



## Google Colab คืออะไร

- Google Colab ชื่อเต็มคือ Google Colaboratory เป็นบริการ Software as a Service (SaaS) โฮสต์โปรแกรม Jupyter Notebook บน Cloud จาก Google
- Google Colab ใช้ยังไง เราสามารถใช้ Google Colab สร้าง Notebook เขียนโปรแกรมภาษา Python ได้ฟรี ๆ และแถมยังมี GPU, TPU ให้เราใช้ได้ฟรีอีก ทีละ 12 ชั่วโมง ใจดีสุด ๆ
- ขณะนี้ Google Colab มี GPU ให้เราใช้ ดังนี้
  - Nvidia Tesla K80
  - Nvidia Tesla T4
  - Nvidia Tesla P100 (ดีที่สุด)
- โดยเราสามารถ เช็คสเปคของ GPU ได้ด้วยคำสั่ง `!nvidia-smi` ถ้าไม่พอใจสามารถ เลือกเมนู Runtime / Factory reset runtime เพื่อเปลี่ยนเครื่อง อาจจะได้ GPU ที่ดีขึ้น ถ้ามีเครื่องว่าง

ทำการเข้าไปที่ Link: <https://colab.research.google.com/>  
เลือก File > New Notebook

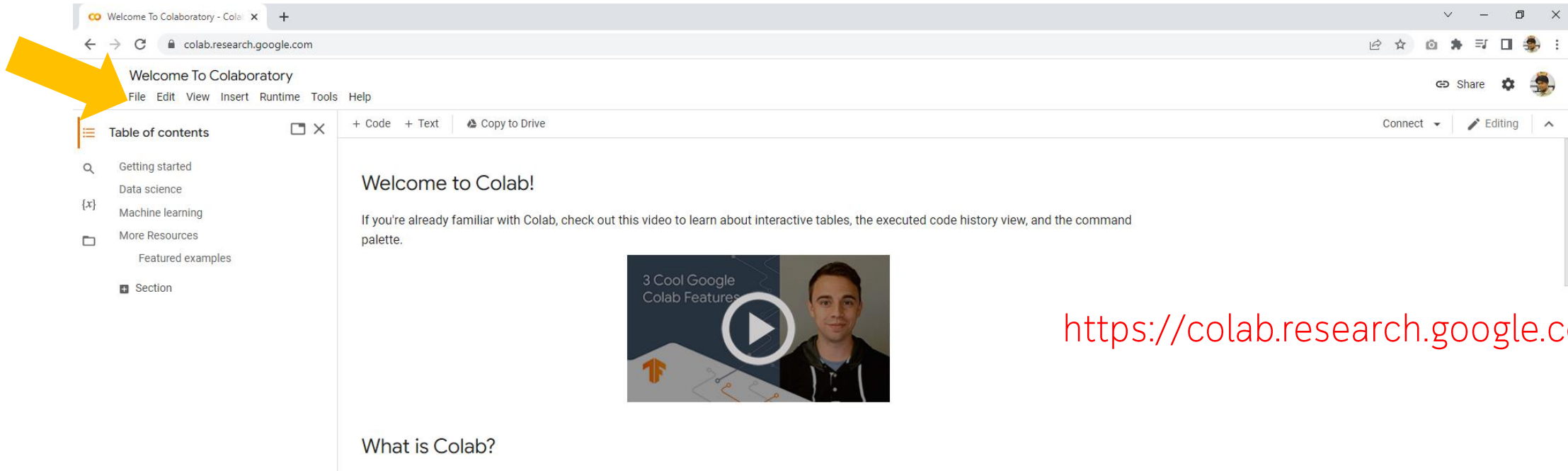


Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
  - Featured examples
  - Section

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

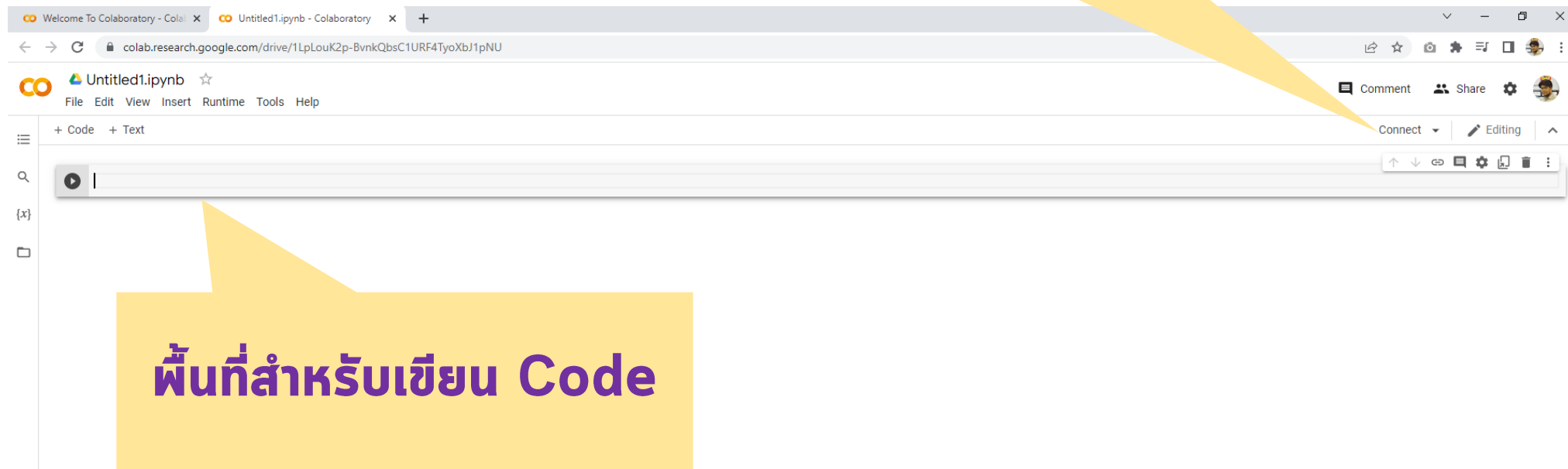
3 Cool Google Colab Features

What is Colab?

<https://colab.research.google.com/>

จากนั้นจะได้หน้าจอที่ไว้ใช้สำหรับเขียน Code ภาษา Python

สามารถกดปุ่ม Connect เพื่อ  
เชื่อมต่อก่อน Run โปรแกรม



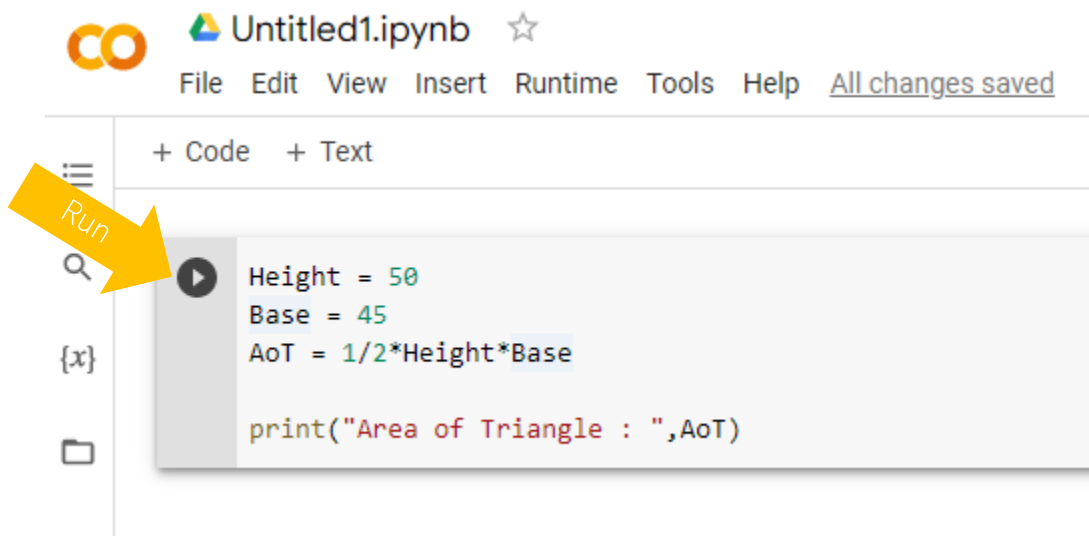
พื้นที่สำหรับเขียน Code

ทดสอบโปรแกรมคำนวณหาพื้นที่สามเหลี่ยม

ตัวอย่าง : ความสูง (Height) = 50 , ฐาน (Base) = 45

จากสูตร :  $\frac{1}{2} * \text{ฐาน} * \text{สูง}$

ทำการคลิกปุ่ม Run เพื่อดูผลลัพธ์



Untitled1.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

+ Code + Text

```
Height = 50
Base = 45
AoT = 1/2*Height*Base

print("Area of Triangle : ",AoT)
```

Run

Output



Untitled1.ipynb ☆

File Edit View Insert Runtime Tools Help

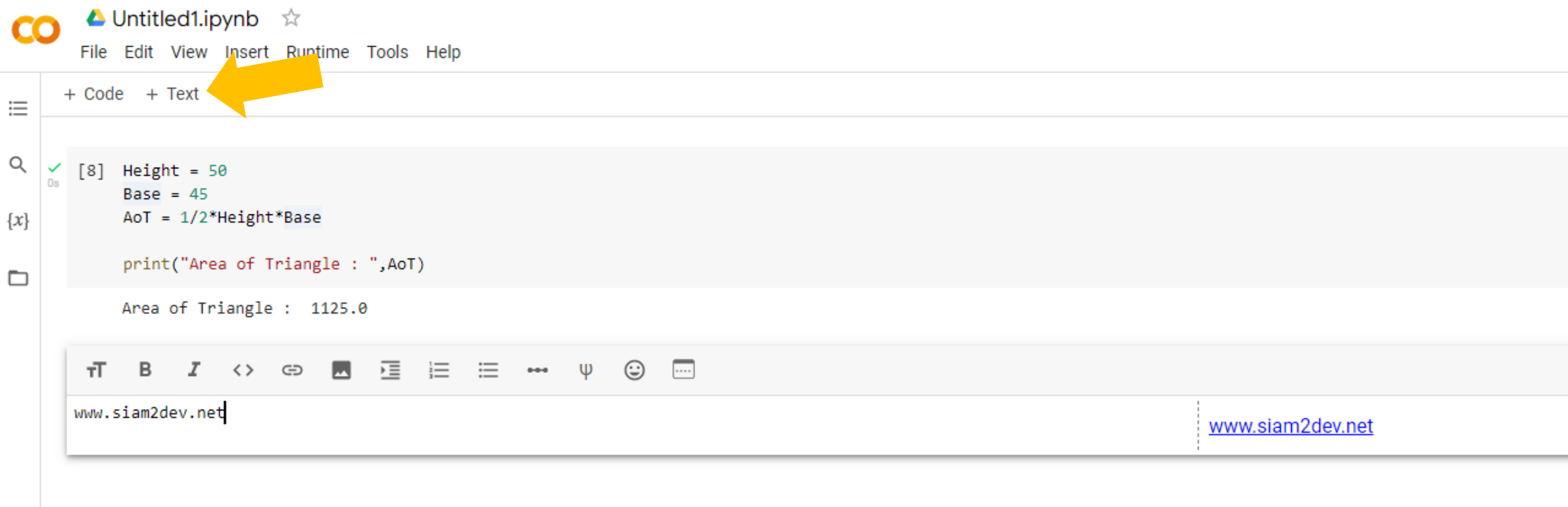
+ Code + Text

```
Height = 50
Base = 45
AoT = 1/2*Height*Base

print("Area of Triangle : ",AoT)
```

Area of Triangle : 1125.0

สามารถทำการเพิ่มช่องข้อความได้ด้วยการกดที่ปุ่ม Text เพื่อใช้ในการพิมพ์ข้อความหรือแบ่งหัวข้อมาทำงานจากตัวอย่างทำการเพิ่มข้อความว่า [www.siam2dev.com](http://www.siam2dev.com)



The screenshot shows the Google Colab interface. At the top, there is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the menu bar, there are two tabs: '+ Code' and '+ Text'. A yellow arrow points to the '+ Text' tab. The main area contains a code cell with the following code:

```
[8] Height = 50
Base = 45
AoT = 1/2*Height*Base

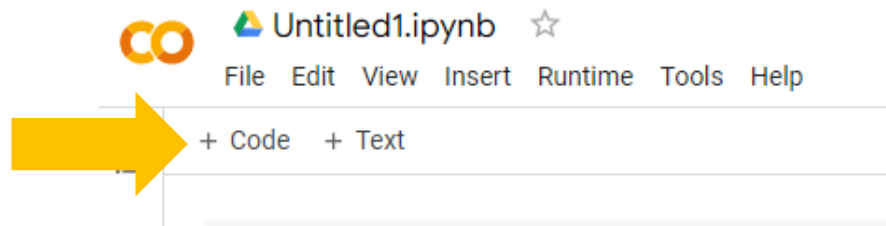
print("Area of Triangle : ",AoT)

Area of Triangle : 1125.0
```

Below the code cell, there is a text cell containing the text [www.siam2dev.net](http://www.siam2dev.net). The text cell has a rich text editor toolbar above it with various icons for text formatting and insertion.

## ทดสอบการใช้งาน String

ก่อนอื่นต้องทำการเพิ่มบรรทัดสำหรับเขียน Code โดยคลิกที่ปุ่ม **+ Code**



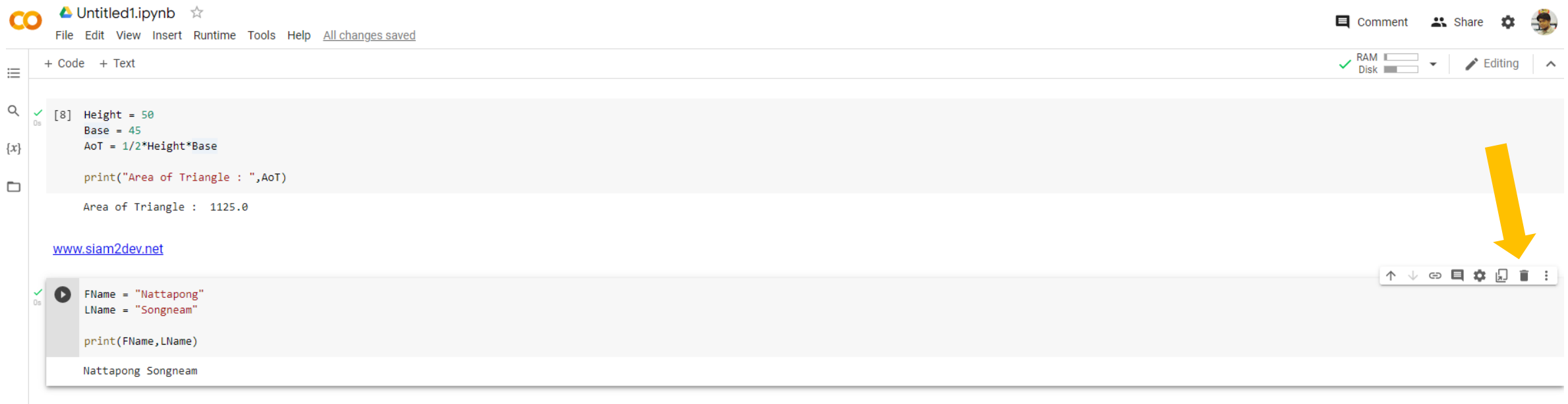
จากนั้นทำการสร้างตัวแปรเพื่อเก็บข้อความ ดังนี้ และแสดงผลลัพธ์ ชื่อ และนามสกุล  
FName = "Nattapong"  
Lname = "Songneam"

```
FName = "Nattapong"  
LName = "Songneam"  
  
print(FName, LName)
```

Output

```
✓ 0s  
FName = "Nattapong"  
LName = "Songneam"  
  
print(FName, LName)  
  
Nattapong Songneam
```

## หากต้องการลบบรรทัดสามารถกดได้ที่ปุ่มรูปถังขยะด้านขวา



The screenshot shows the Google Colab interface for a file named 'Untitled1.ipynb'. The top navigation bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', along with 'Comment', 'Share', and a settings icon. The main workspace contains two code cells. The first cell has the following code: 

```
[8] Height = 50
Base = 45
AoT = 1/2*Height*Base

print("Area of Triangle : ",AoT)
```

 and the output is 'Area of Triangle : 1125.0'. The second cell has the code: 

```
FName = "Nattapong"
LName = "Songneam"

print(FName, LName)
```

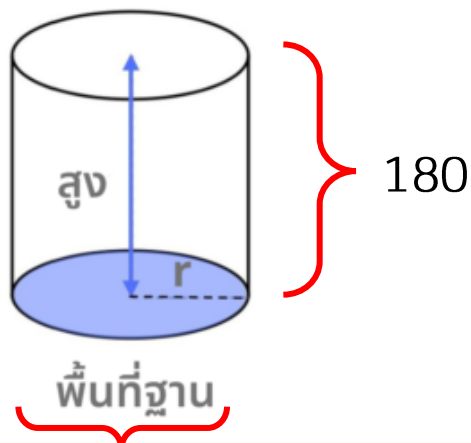
 and the output is 'Nattapong Songneam'. A yellow arrow points to a trash icon in the bottom right corner of the code editor area, which is used for deleting lines of code.

## 1. จงเขียน Code แนะนำตัวเอง โดยมีลักษณะของผลลัพธ์ดังนี้

```
Name : Nattapong Songneam  
Student ID : 6510122115000  
Tel : 099-087-4433  
Email : siam2dev@hotmail.com
```

## 2. จงเขียน Code หาปริมาตรทรงกระบอก ดังนี้

$$\text{ปริมาตร} = \pi r^2 h$$





## อ้างอิง

- <https://www.9experttraining.com/articles/python-คืออะไร>
- <https://medium.com/@moungsiri/โครงสร้างภาษา-python-53cc38a51462>
- <http://marcuscode.com/lang/python/variables-and-types>
- <http://cms576.bps.in.th/group11/introduction-to-computer-programming>
- <https://www.mindphp.com/บทเรียนออนไลน์/83-python/2398-python-operator.html>
- <https://th.wikipedia.org/wiki/วิซวลสตูดิโอโค้ด>