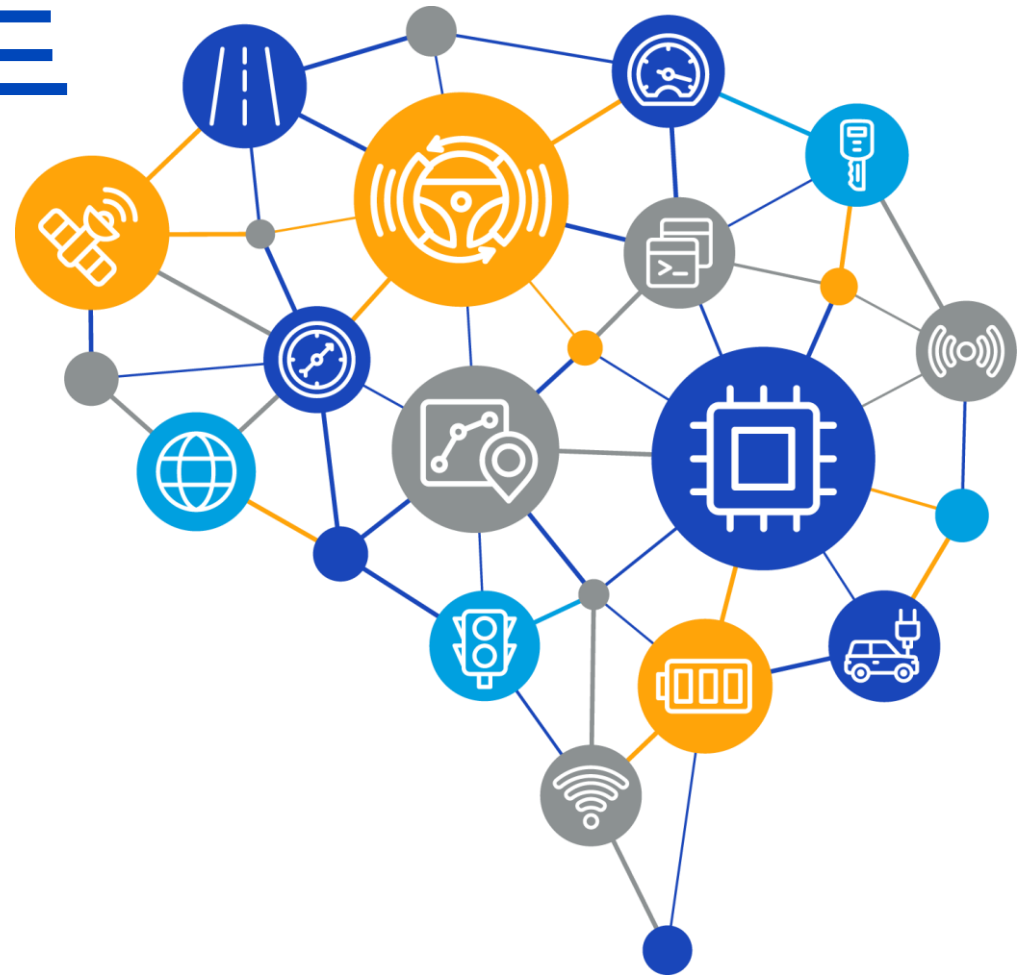


# 비주얼 스튜디오 코드 파이썬 환경과 실행



강환수 교수

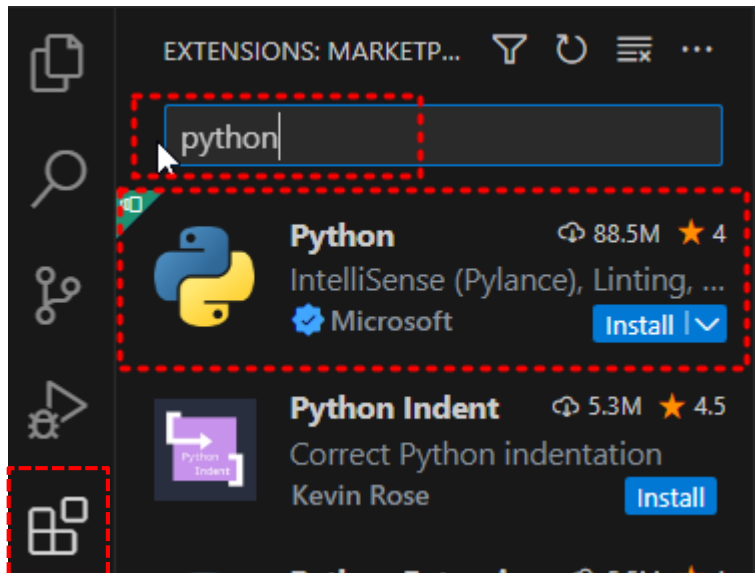
AI Experts  
Who Lead  
The Future

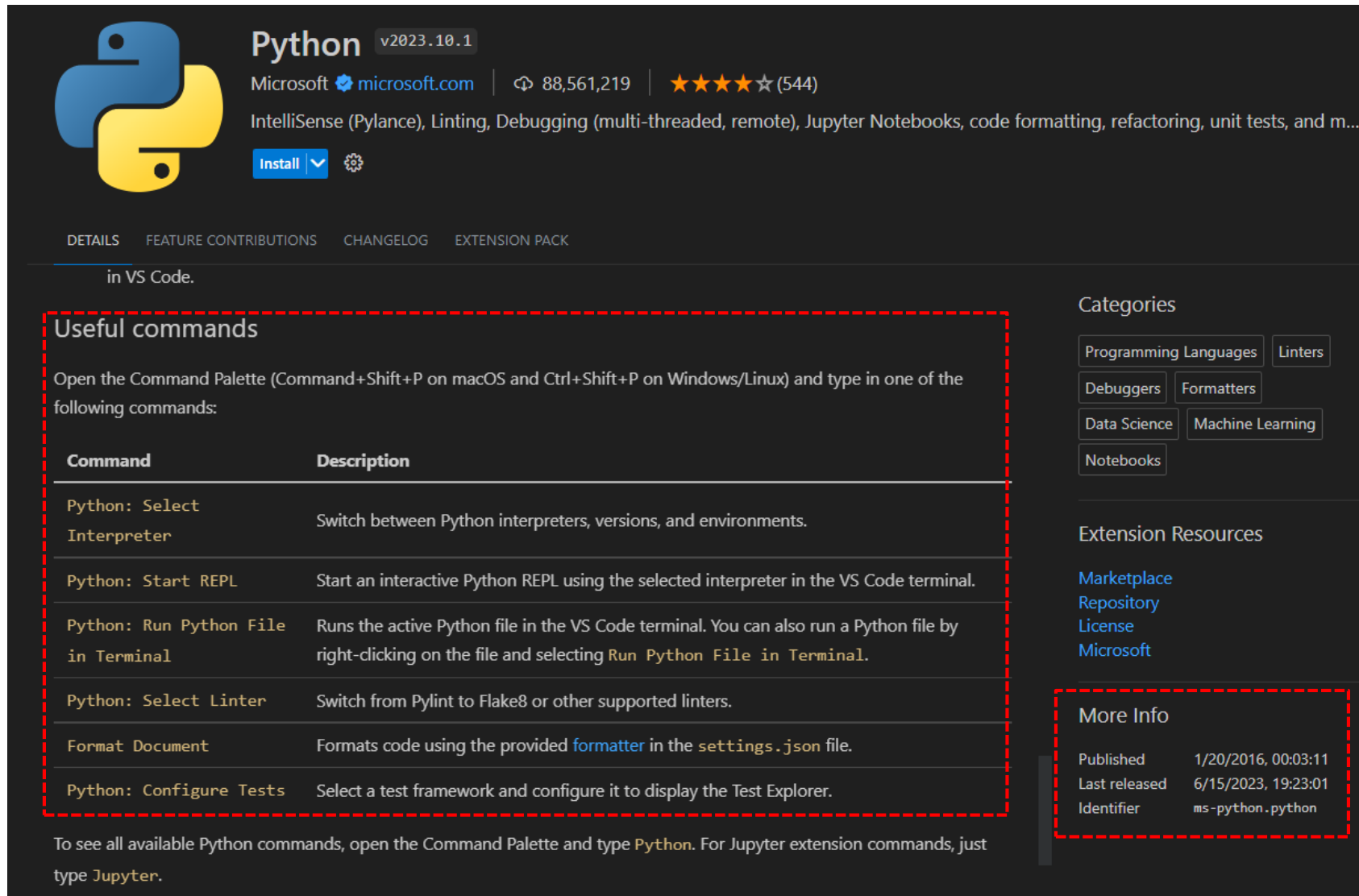
01

Vs code 파이썬 코드 실행 기초

- Vs code
  - 인터프리터가 반드시 필요
- 즐겨 활용하는 인터프리터
  - ① 표준 Python 인터프리터 설치
    - [python.org](https://python.org) 에서 Python을 설치
  - ② 또는 아나콘다 등의 배포판 설치

- Python 확장 설치





**Python** v2023.10.1

Microsoft [microsoft.com](https://microsoft.com) | 88,561,219 | ★★★★★ (544)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring, unit tests, and m...

[Install](#) [Settings](#)

[DETAILS](#) [FEATURE CONTRIBUTIONS](#) [CHANGELOG](#) [EXTENSION PACK](#)

in VS Code.

### Useful commands

Open the Command Palette (Command+Shift+P on macOS and Ctrl+Shift+P on Windows/Linux) and type in one of the following commands:

Command	Description
Python: Select Interpreter	Switch between Python interpreters, versions, and environments.
Python: Start REPL	Start an interactive Python REPL using the selected interpreter in the VS Code terminal.
Python: Run Python File in Terminal	Runs the active Python file in the VS Code terminal. You can also run a Python file by right-clicking on the file and selecting <b>Run Python File in Terminal</b> .
Python: Select Linter	Switch from Pylint to Flake8 or other supported linters.
Format Document	Formats code using the provided <a href="#">formatter</a> in the <code>settings.json</code> file.
Python: Configure Tests	Select a test framework and configure it to display the Test Explorer.

To see all available Python commands, open the Command Palette and type **Python**. For Jupyter extension commands, just type **Jupyter**.

### Categories

- Programming Languages
- Linters
- Debuggers
- Formatters
- Data Science
- Machine Learning
- Notebooks

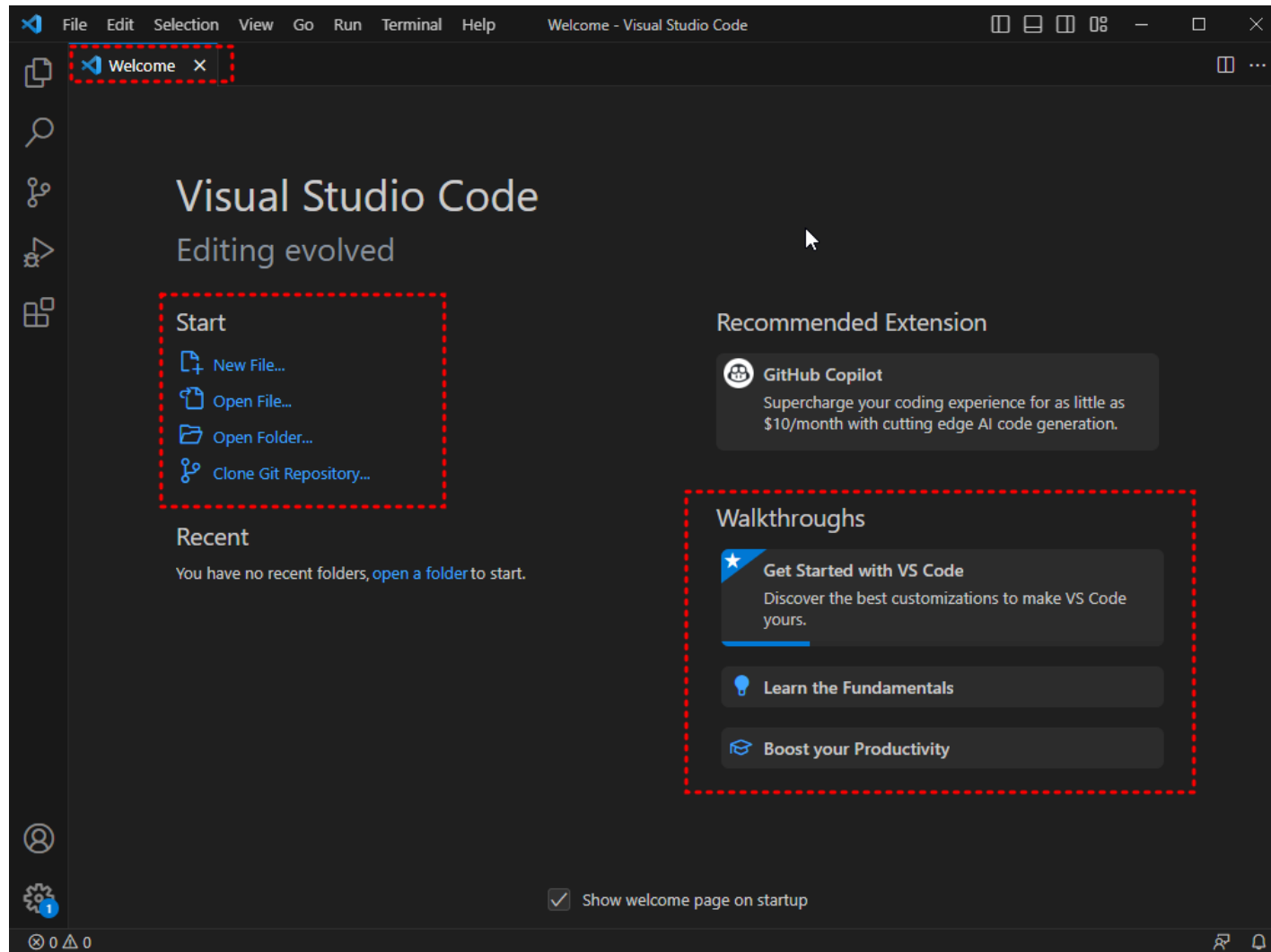
### Extension Resources

- [Marketplace](#)
- [Repository](#)
- [License](#)
- [Microsoft](#)

### More Info

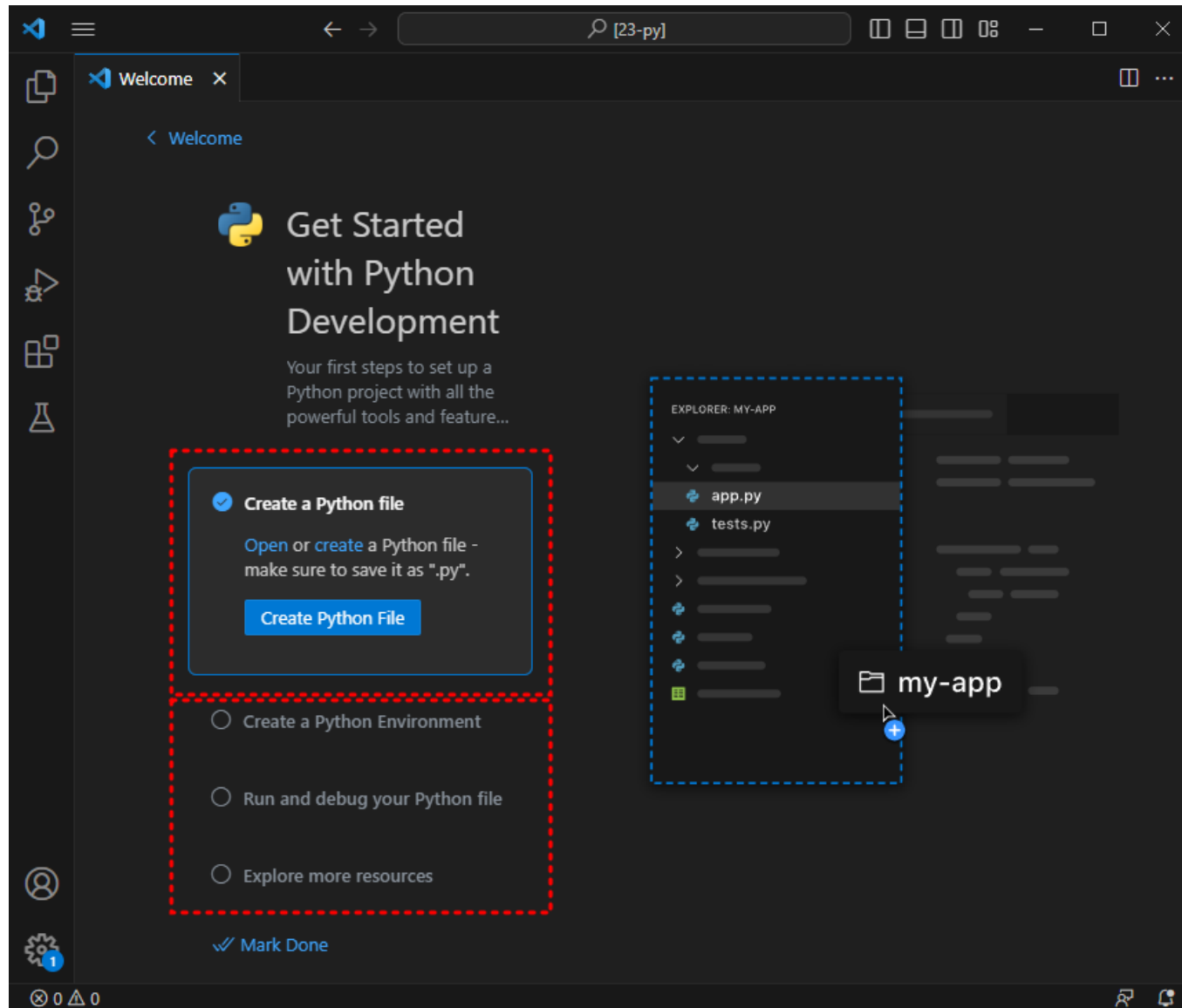
Published	1/20/2016, 00:03:11
Last released	6/15/2023, 19:23:01
Identifier	ms-python.python

- 다를 수 있음

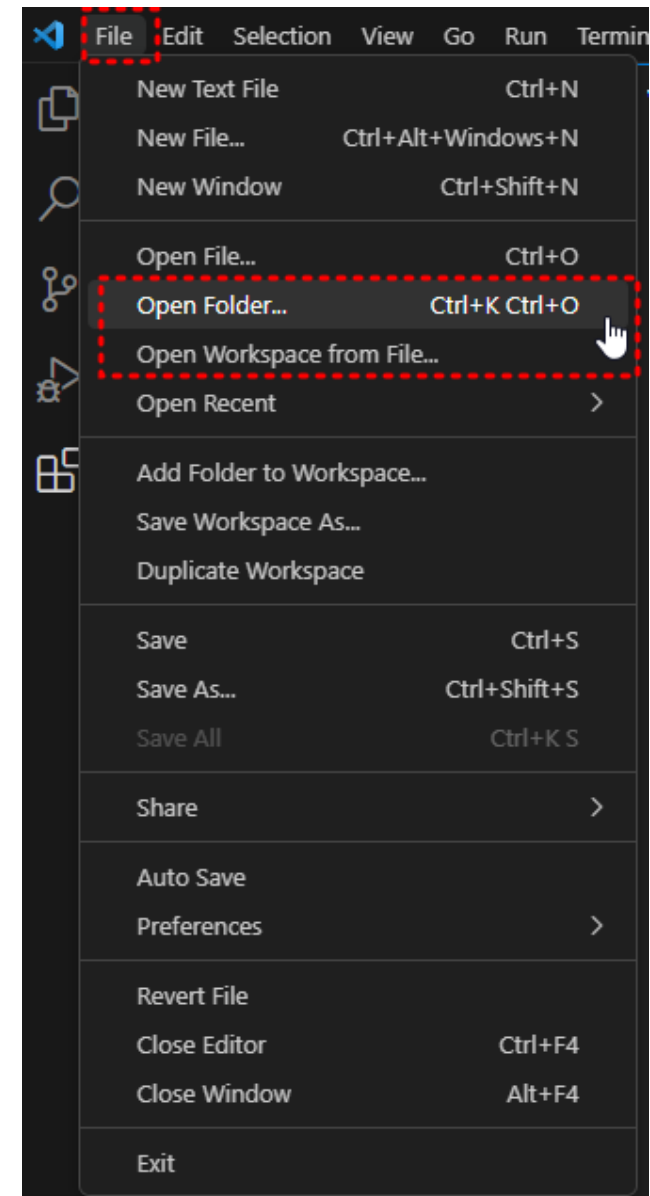
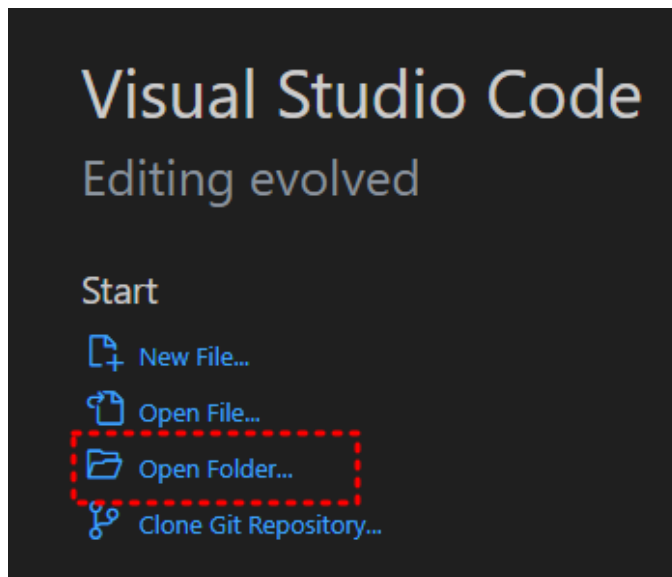


# Get Strted with vs Code

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language



- 폴더 지정
  - 우리 작업 폴더
    - D:\2023-github-codespace\code
  - Ctrl + K + Ctrl + O

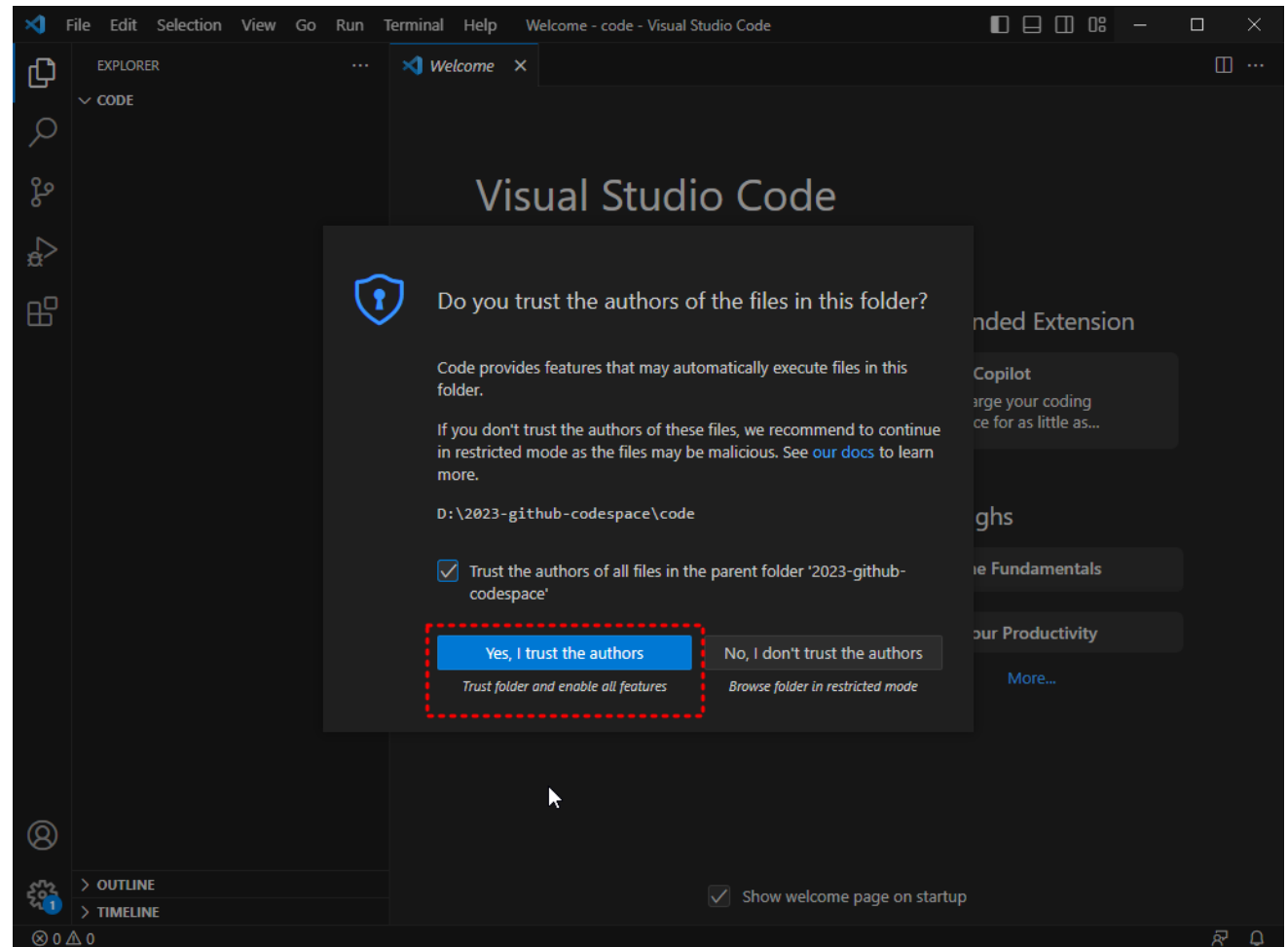
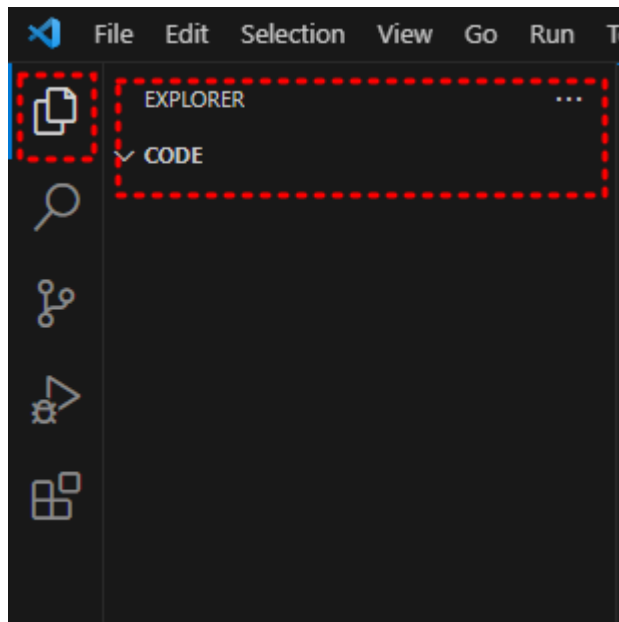




# 신뢰 확인

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- 열린 폴더



- **chatGPT**      파이썬을 처음 배우는 학생에게 도움이 되는  
코드 10줄 정도를 코딩해 줘, 설명도 해줘

# 1. Hello, World! 출력하기

```
print("Hello, World!")
```

# 2. 변수 사용하기

```
name = input("이름을 입력하세요: ")
```

```
print("안녕하세요, " + name + "님!")
```

# 3. 숫자 계산하기

```
num1 = 10
```

```
num2 = 5
```

```
result = num1 + num2
```

```
print("결과:", result)
```

# 4. 조건문 사용하기

```
age = int(input("나이를 입력하세요: "))
```

```
if age >= 18:
```

```
    print("성인입니다.")
```

```
else:
```

```
    print("미성년자입니다.")
```

# 5. 반복문 사용하기

```
for i in range(5):
```

```
    print("반복문 예제:", i)
```

# 6. 리스트 활용하기

```
fruits = ["사과", "바나나", "오렌지"]
```

```
print("과일 목록:", fruits)
```

# 7. 함수 정의하기

```
def square(x):
```

```
    return x * x
```

```
result = square(5)
```

```
print("결과:", result)
```

# 8. 파일 입출력

```
file_name = "example.txt"
```

```
with open(file_name, "w") as f:
```

```
    f.write("파일 입출력 예시")
```

# 9. 모듈 사용하기

```
import random
```

```
random_number = random.randint(1, 100)
```

```
print("랜덤 숫자:", random_number)
```

# 10. 예외 처리하기

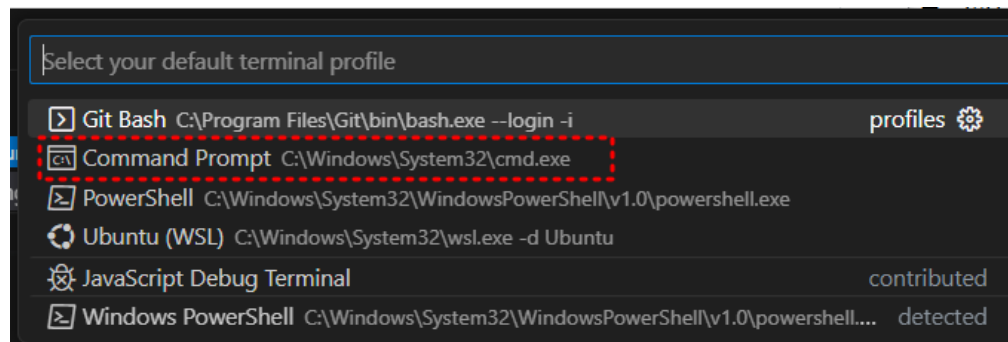
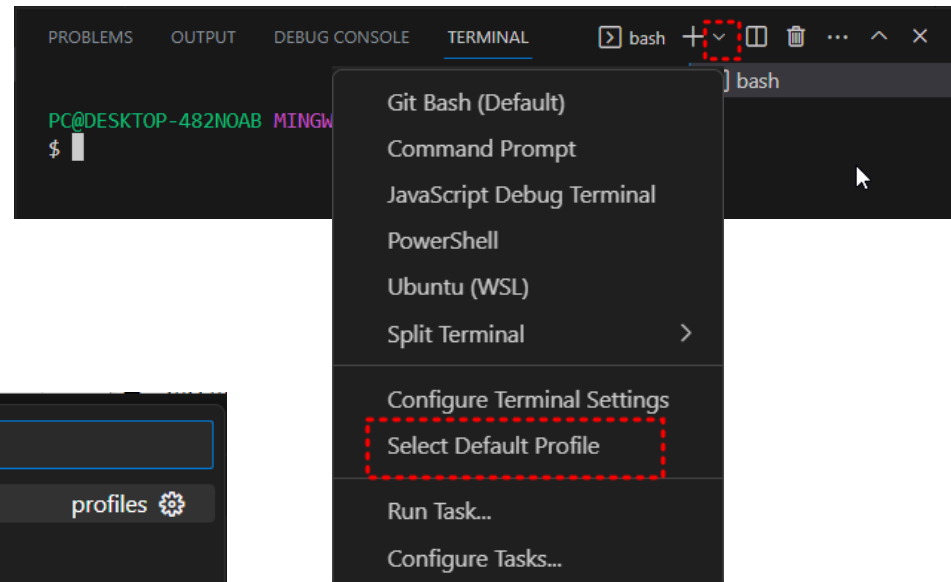
```
try:
```

```
    x = 10 / 0
```

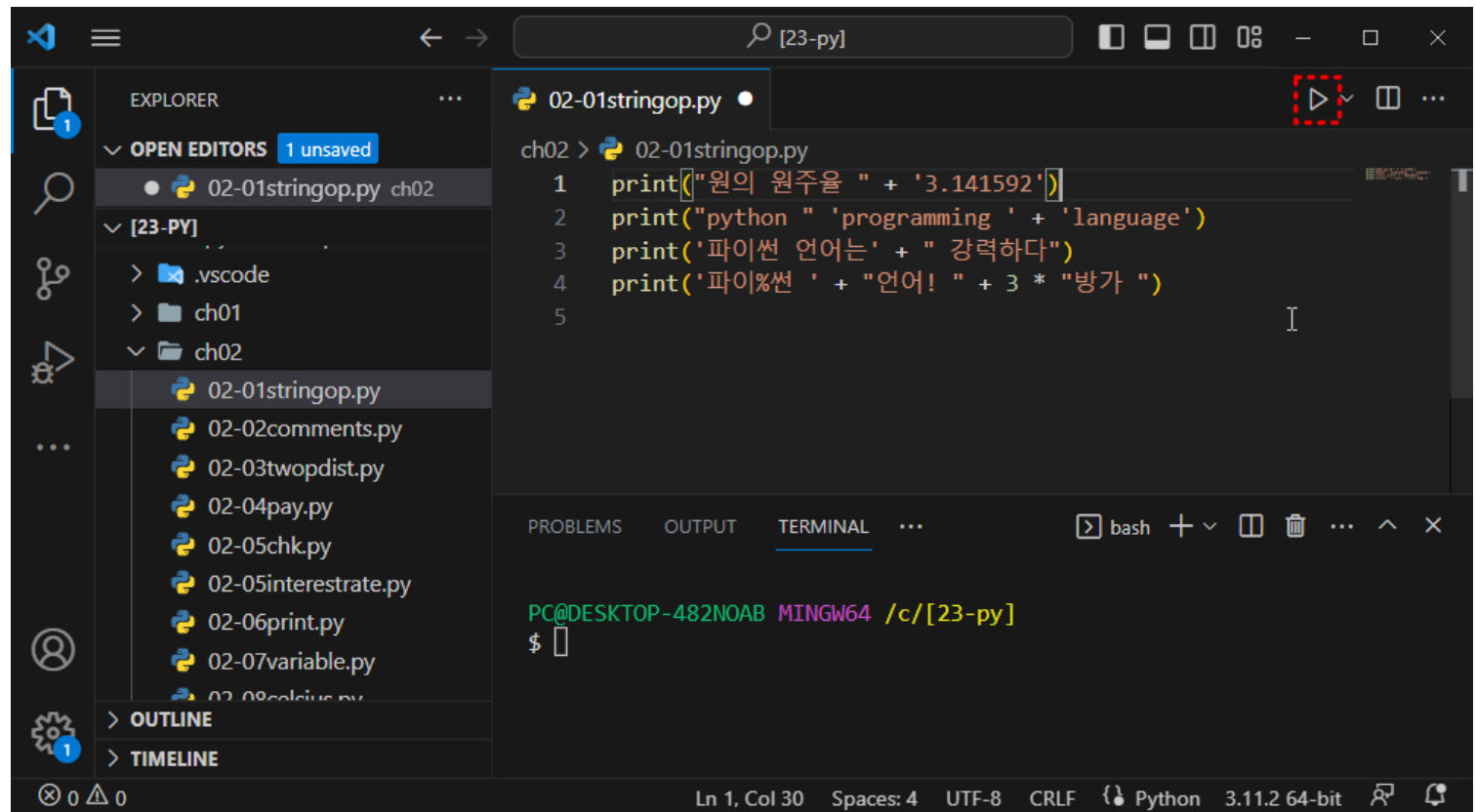
```
except ZeroDivisionError:
```

```
    print("0으로 나눌 수 없습니다.")
```

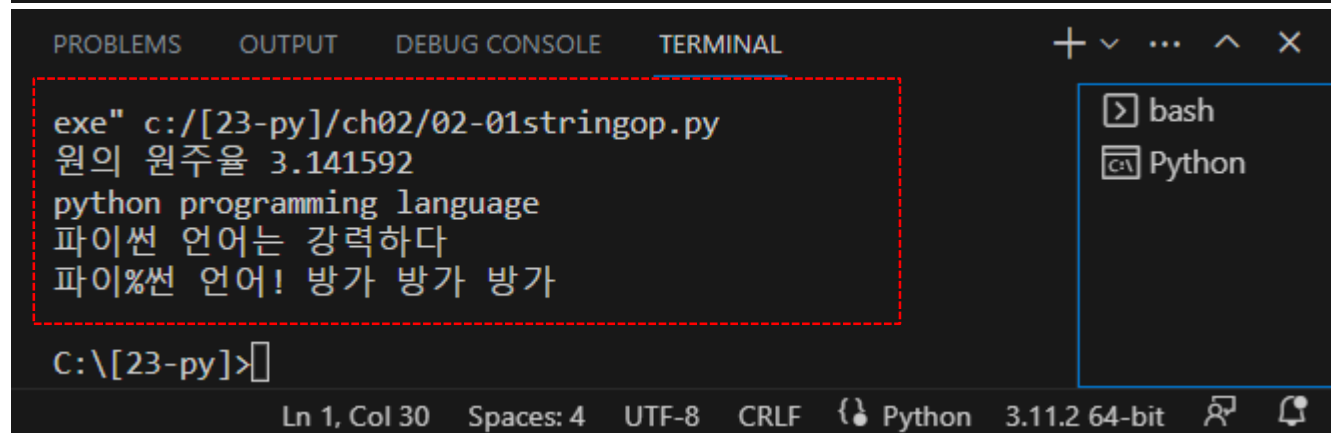
- 파일 생성
  - hello.py
- 터미널 실행
  - Ctrl + `(backquote)[왼쪽 Tab 키 위]
- 터미널 프로파일 설정
  - 터미널 아이콘 메뉴 아래 화살표 클릭
    - Select Default Profile
  - 원하는 셸 선택
    - Command Prompt



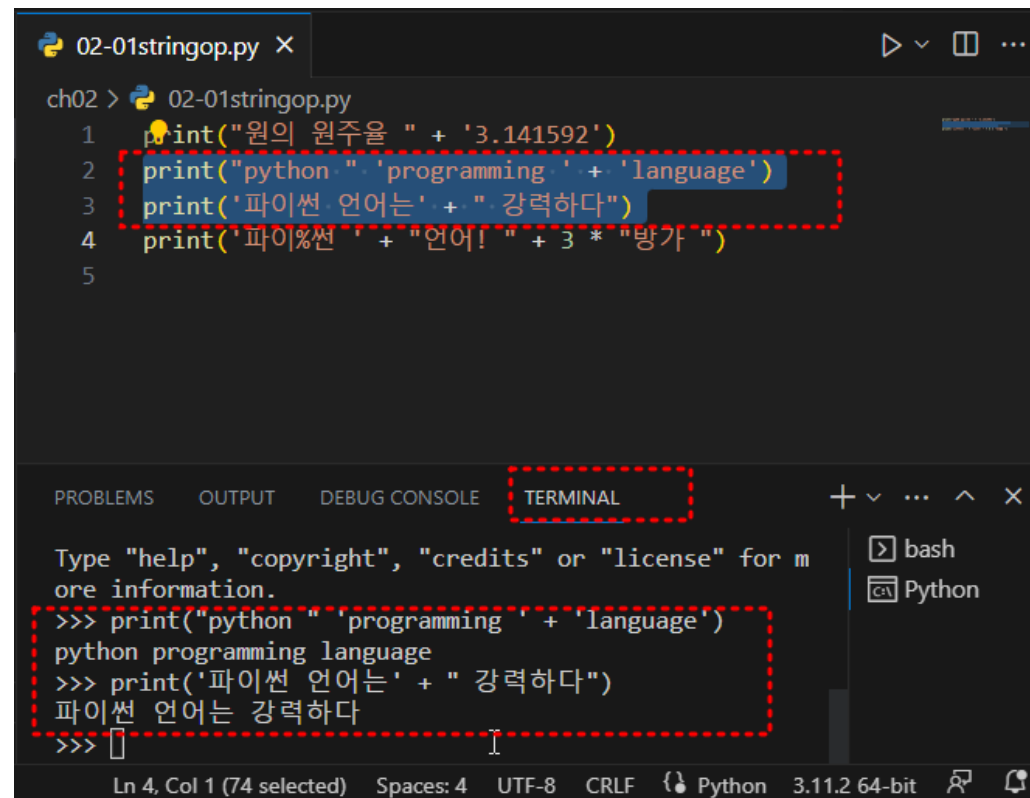
- 실행 아이콘 클릭



- 터미널에서 실행



- 편집기 창 클릭 > 마우스 오른쪽 버튼으로 클릭
  - Run Python | Run Python File in Terminal
- 하나 이상의 라인을 선택
  - 마우스 오른쪽 메뉴
    - Run Python | Run Selection/Line in Python Terminal
  - Shift + Enter

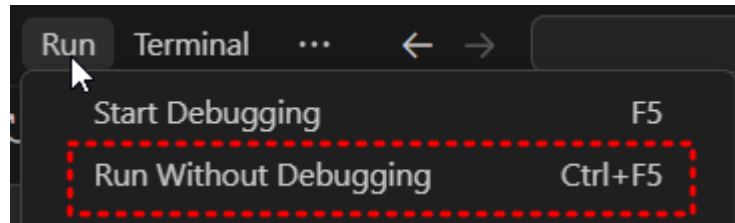


The screenshot shows a code editor with a file named '02-01stringop.py'. The code contains four lines of Python code. The second and third lines are selected. The terminal window at the bottom shows the output of the selected lines. The first line of code is commented out. The second line is highlighted in blue. The third line is highlighted in blue. The fourth line is not highlighted. The terminal window shows the output of the selected lines. The first line of output is 'python programming language'. The second line of output is '파이썬 언어는 강력하다'.

```
02-01stringop.py X
ch02 > 02-01stringop.py
1 print("원의 원주율 " + '3.141592')
2 print("python " + 'programming ' + 'language')
3 print('파이썬 언어는 ' + " 강력하다")
4 print('파이썬 ' + "언어! " + 3 * "방가 ")
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Type "help", "copyright", "credits" or "license" for more information.
>>> print("python " + 'programming ' + 'language')
python programming language
>>> print('파이썬 언어는 ' + " 강력하다")
파이썬 언어는 강력하다
>>>
```

- **Ctrl + F5**
  - Run | Run Without Debugging

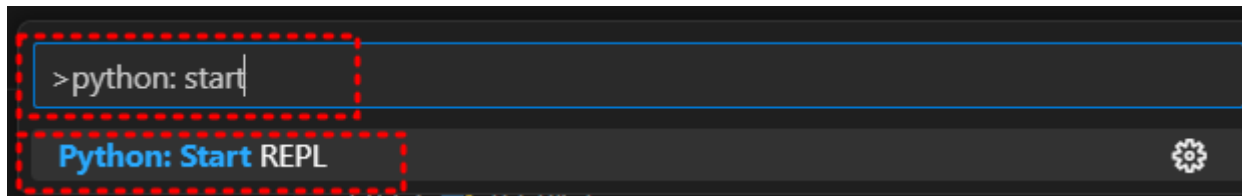


# Python 인터프리터에 대한 REPL 터미널 열기

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

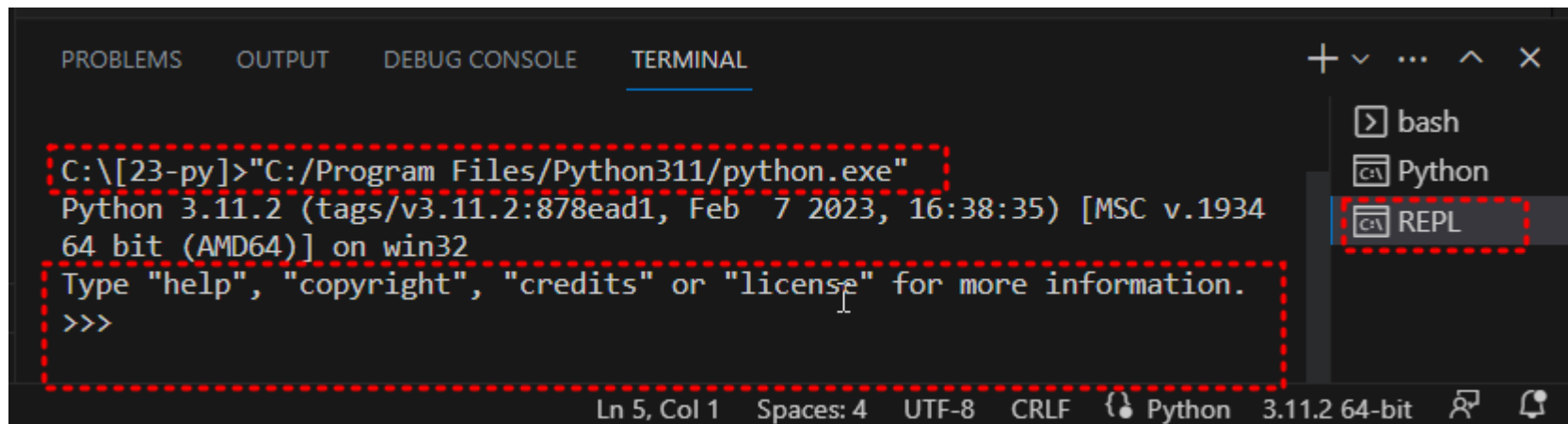
- **Command Palette**

- Ctrl+Shift+P



- **다음 입력**

- Python: Start REPL
- REPL에서
  - 코드 행을 한 번에 하나씩 입력하고 실행



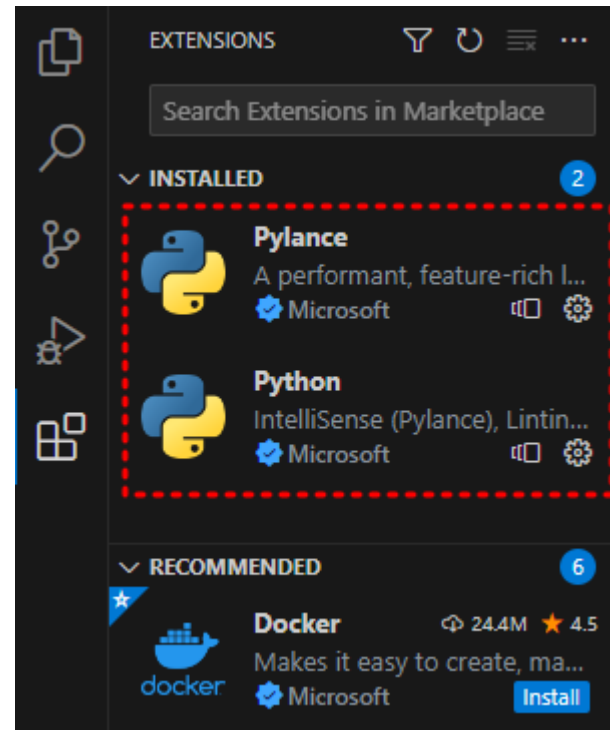
AI Experts  
Who Lead  
The Future

## 02

주피터 노트북에서 파이썬 실행

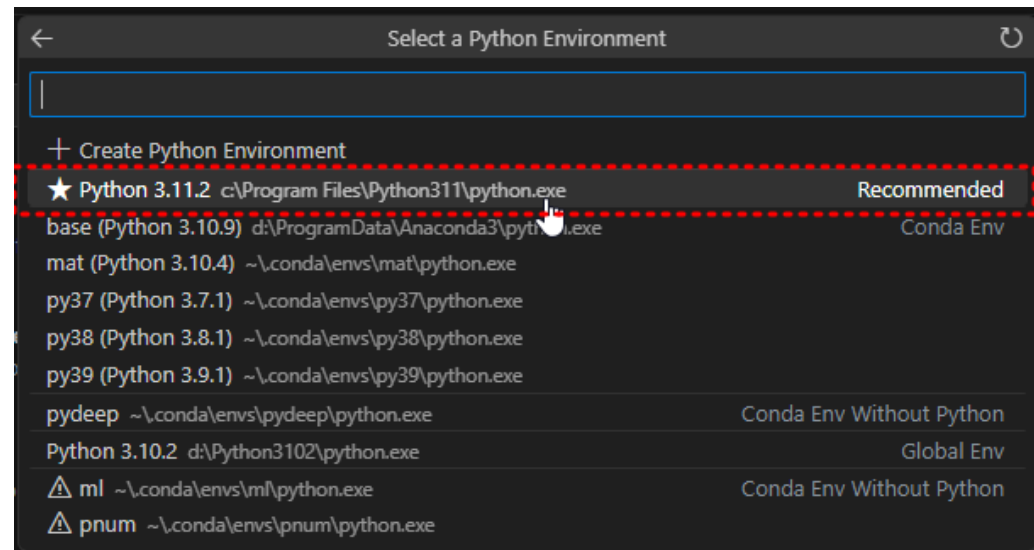
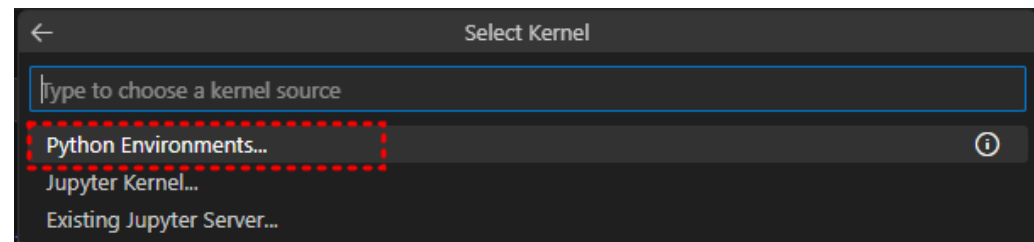
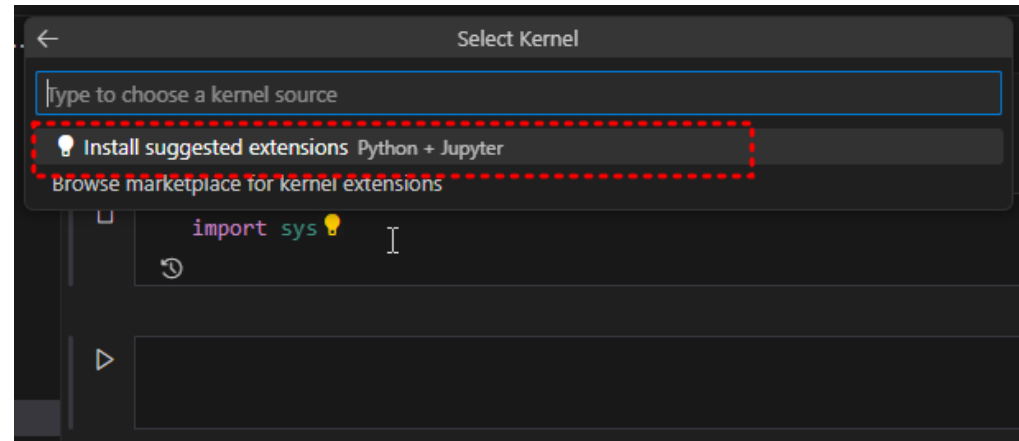


- 기본

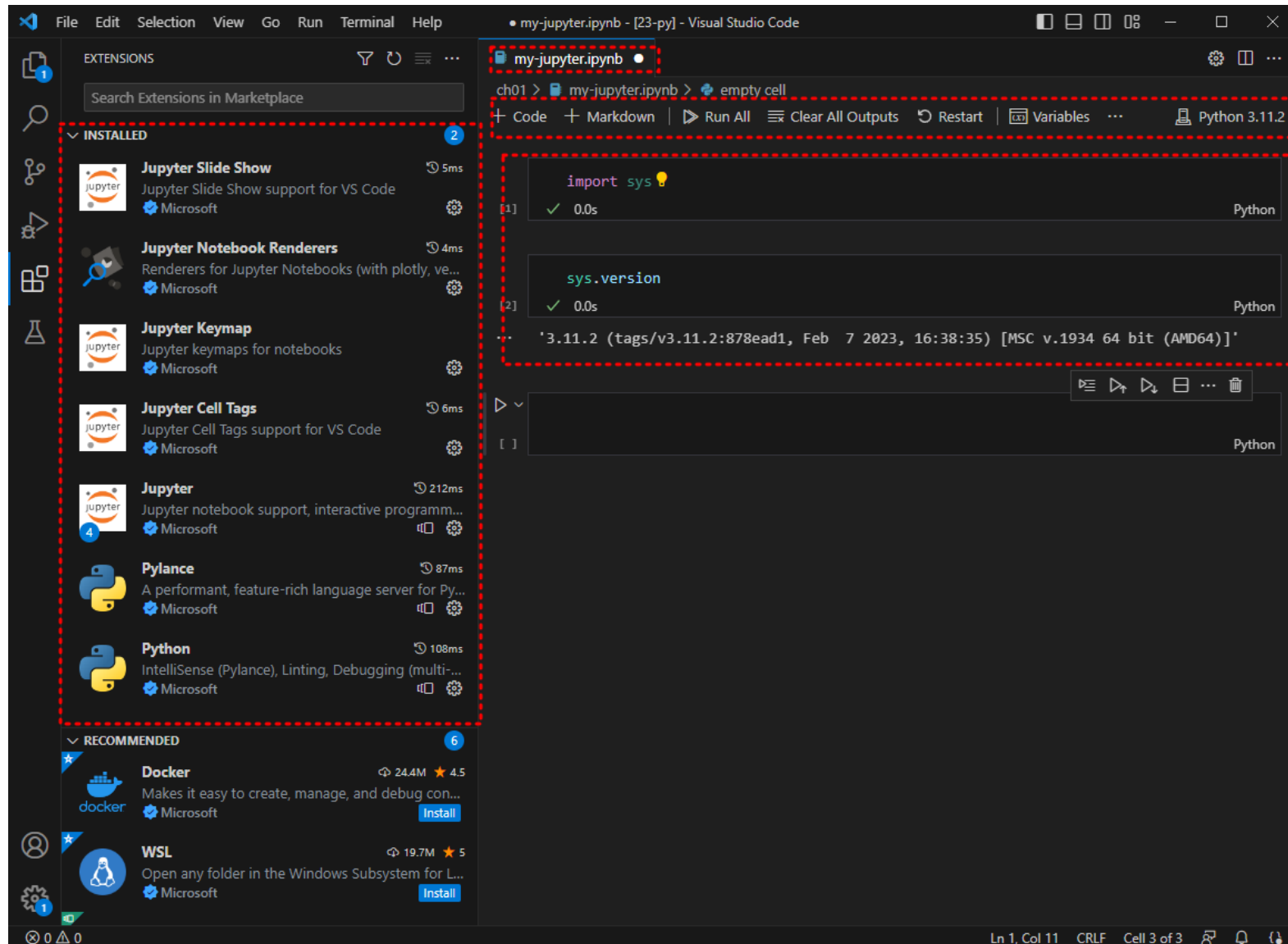


- 대화형이 강화된 인터프리터
  - 주피터(이전의 Ipython Notebook)
    - 마크다운 텍스트와 실행 가능한 파이썬 소스 코드를 노트북이라는 하나의 캔버스에 쉽게 결합할 수 있는 오픈 소스 프로젝트
    - 코드 셀로 작업

- 파일 my-jupyter.ipynb
- 첫 셀 코딩
  - import sys
    - Shift + Enter
- 선택 install suggested extensions
  - 주피터 관련 여러 개의 확장 설치
- 다시 실행
  - 선택 Python Environments...
  - 선택 Python 3.11.2



- 주피터 관련 여러 개의 확장 설치



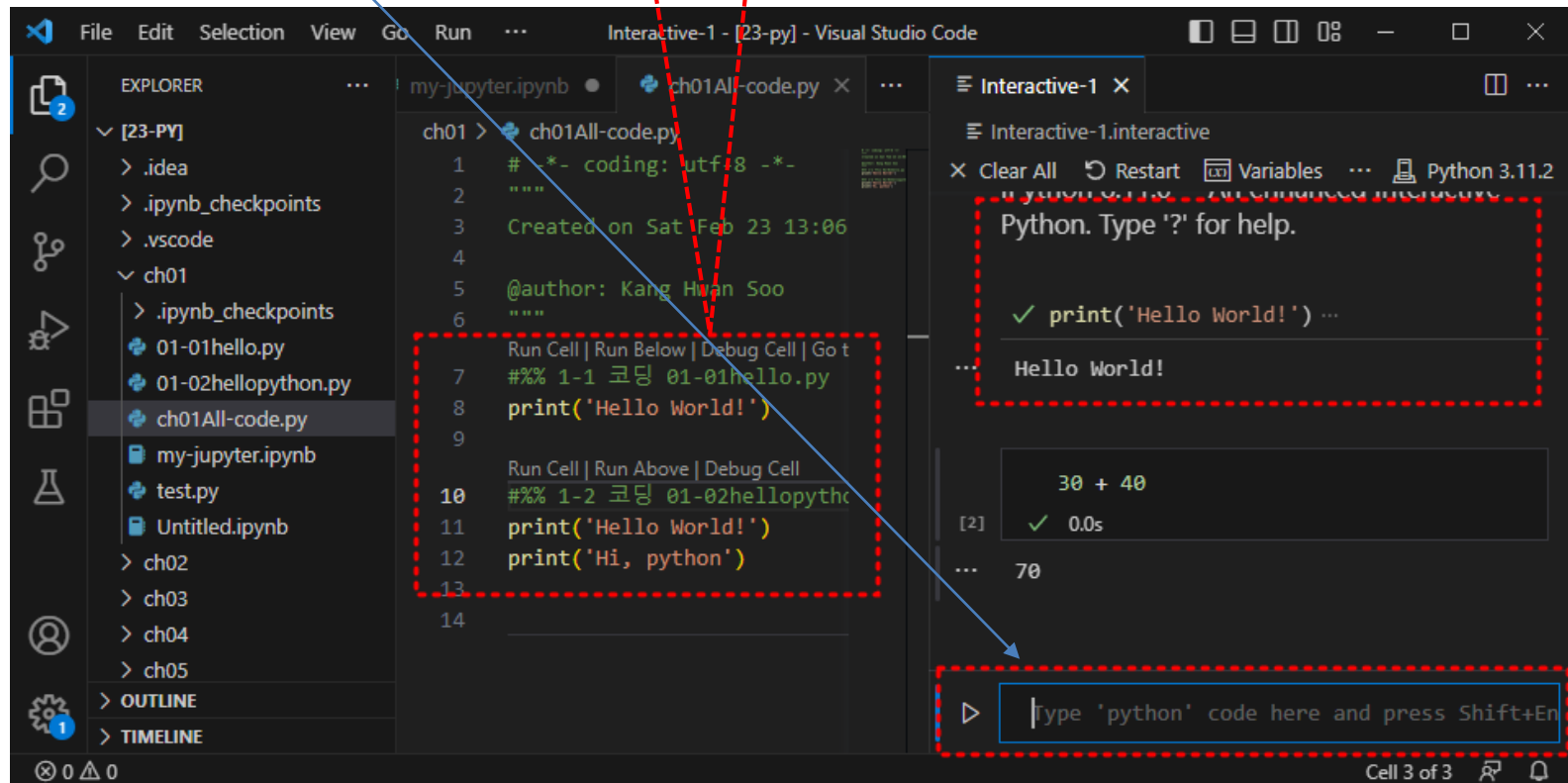
- 파일 \*.py
  - # %%

# 일반 파이썬 파일에서 주피터 코드 셀 실행

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

- # %%
  - 코드 셀 표시
- 실행
  - Shift + Enter
- 실행된 대화형 창의 맨 하부 셀
  - 코딩 후 실행
    - Ctrl + Enter 또는 Shift + Enter

코드 셀을 인식해야  
자동으로 메뉴가 보임



AI Experts  
Who Lead  
The Future

## 03

### 파이썬 실행 단축키 설정

- 편집기 창 클릭 > 마우스 오른쪽 버튼으로 클릭
  - Run Python | Run Python File in Terminal
- 또는 메뉴
  - Run | Run Without Debugging
- 또는 단축키
  - Ctrl + F5
- 새로운 단축키로 '파이썬 파일 실행'을 설정
  - Ctrl + shift + /

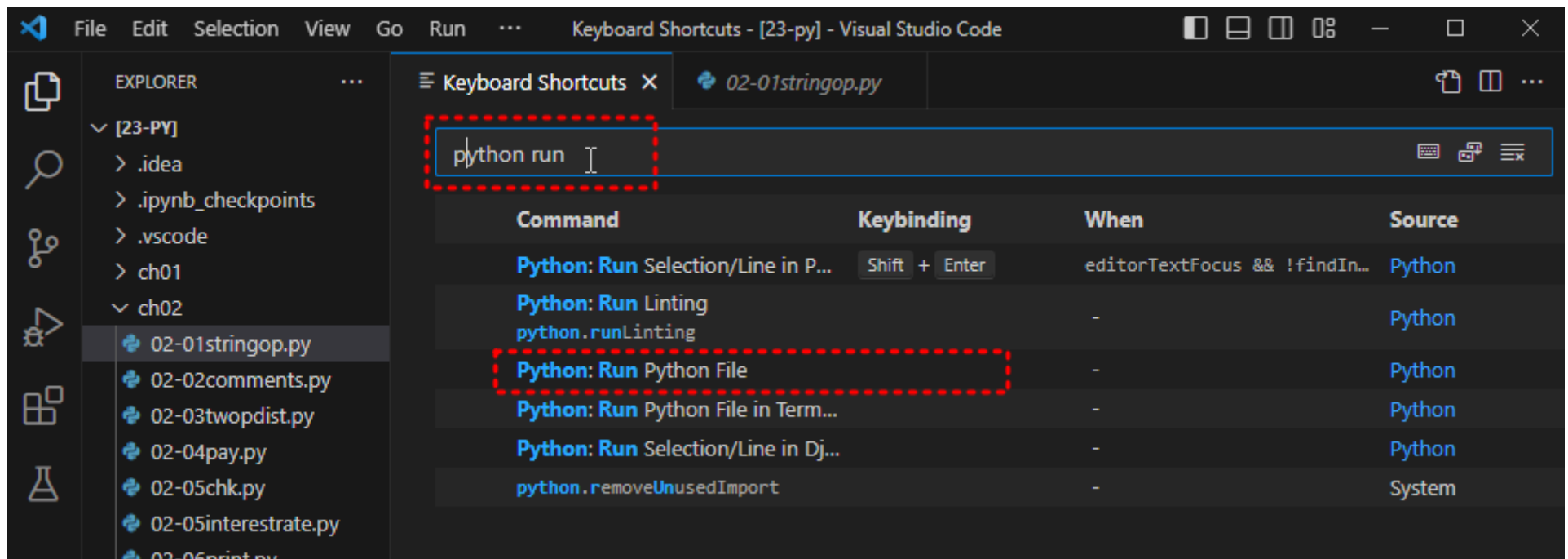


- 단축키

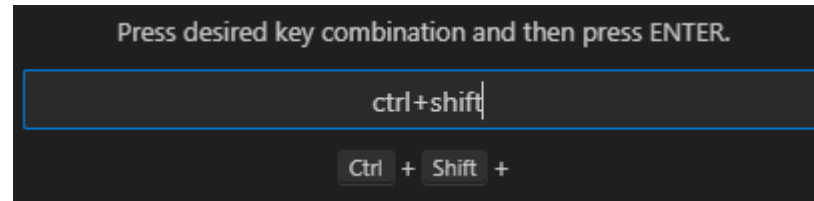
- Ctrl + K Ctrl + S
- 메뉴 파일 | 기본 설정 | 바로 가기 키

- 검색

- Python run python file



- 더블 클릭



- 단축키를 입력

- Ctrl + shift + /
- 이후 enter

Keyboard Shortcuts × 02-01stringop.py

python run

	Command	Keybinding	When	Source
	Python: Run Python File	Ctrl + Shift + /	-	User
	Python: Run Selection/Line in P...	Shift + Enter	editorTextFocus && !findIn...	Python
	Python: Run Linting python.runLinting		-	Python
	Python: Run Python File in Term...		-	Python
	Python: Run Selection/Line in Dj...		-	Python
	python.removeUnusedImport		-	System

# 실행 ctrl + shift + /

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with a folder named 'ch02' containing several Python files. The file '02-02comments.py' is selected and open in the editor. The editor shows the following code:

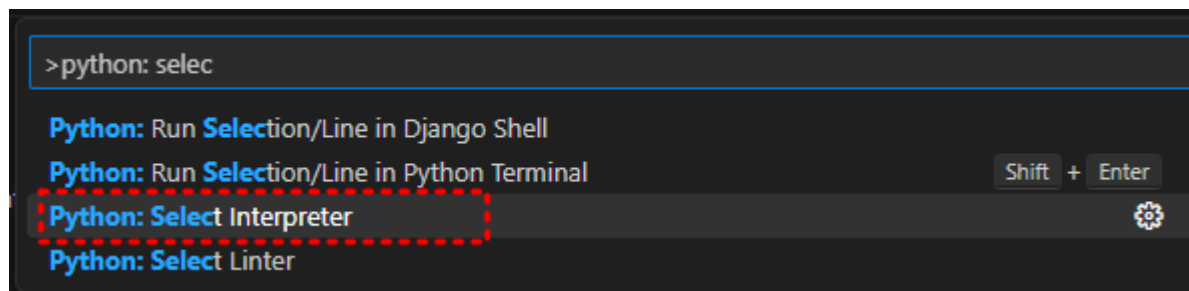
```
ch02 > 02-02comments.py
1  ''' 1-2comments.py
2      2019 3. by Kang Hwan Soo '''
3
4  print('# 이후는 주석') #한 줄에서 문장 이후에도 주석 사용이 가능
5  print('string: "python"') #큰 따옴표 내부에서 작은 따옴표는 문자열
6  print("number: 1 5 3.14") #문자열 내부에서 수도 문자열
7  print("string: 'python'") #작은 따옴표 내부에서 작은 따옴표는 문자열
8
```

The bottom panel shows the TERMINAL window with the following output:

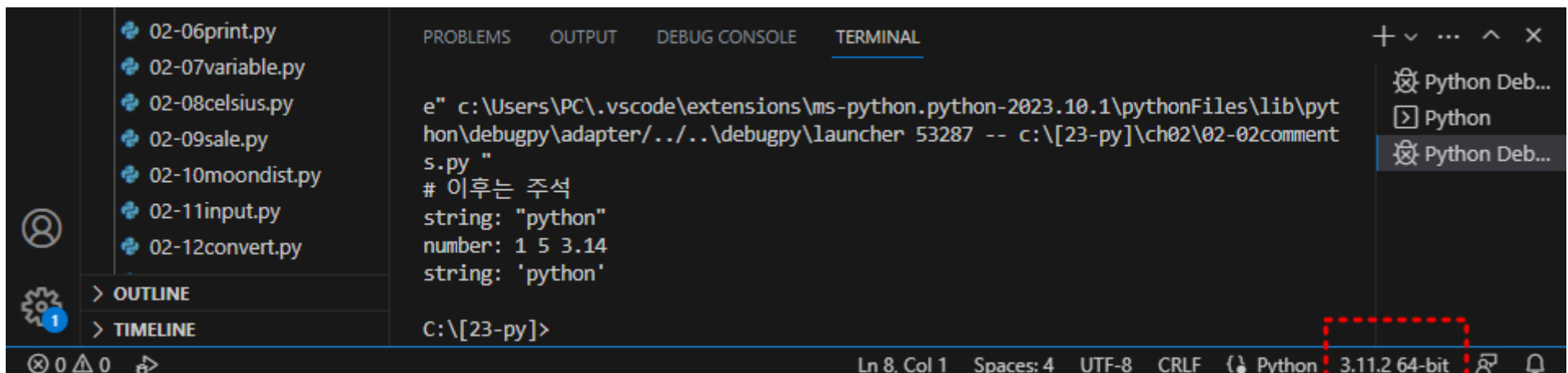
```
파이썬 언어는 강력하다
파이썬 언어! 방가 방가 방가
PS C:\Windows\System32\WindowsPowerShell\v1.0> & "C:/Program Files/Python311/pyt
hon.exe" c:/[23-py]/ch02/02-02comments.py
# 이후는 주석
string: "python"
number: 1 5 3.14
string: 'python'
PS C:\Windows\System32\WindowsPowerShell\v1.0>
```

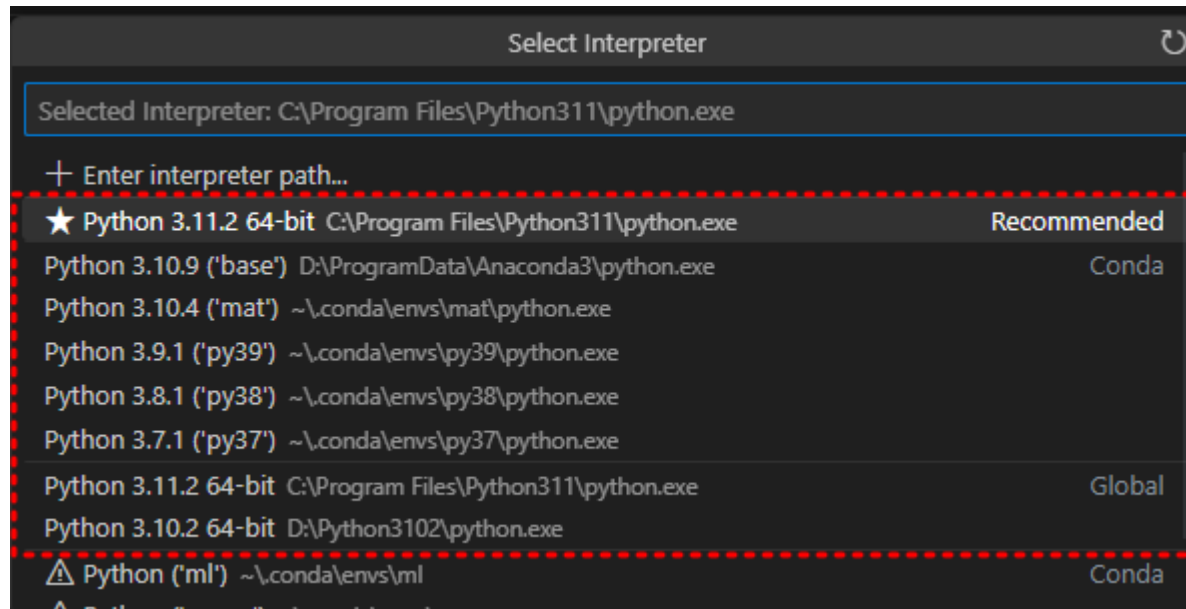
The status bar at the bottom indicates the current line and column (Ln 1, Col 1), the number of spaces (Spaces: 4), the encoding (UTF-8), the line ending (CRLF), the interpreter (Python 3.11.2 64-bit), and other settings.

- 환경 선택
  - 명령 팔레트(Ctrl+Shift+P)
    - Python: Select Interpreter 명령



- 상태바 우측 하단 현재 인터프리터 클릭





AI Experts  
Who Lead  
The Future

# 04

## 필요 외부 패키지 설치

- 설치 전 오류

```
ch09 > 09-08arrayop.py > ...
1  #%% 09-08arrayop.py 1차원 배열의 사칙 연산
2  import numpy as np
3
4  aryA = np.array([1, 2, 3, 4, 5])
5  aryB = np.array([6, 7, 8, 9, 10])
6
7  print(aryA + aryB)
8  print(aryA - aryB)
9  print(aryA * aryB)
10 print(aryA / aryB)
```

- Python 인터프리터 확인
  - > where python
  - > where pip
- 현재 설치 모듈 확인
  - > pip list
- 필요하면 pip 업그레이드
  - > python -m pip install --upgrade pip
- 모듈 numpy 설치와 확인
  - > pip install numpy
  - > pip show numpy

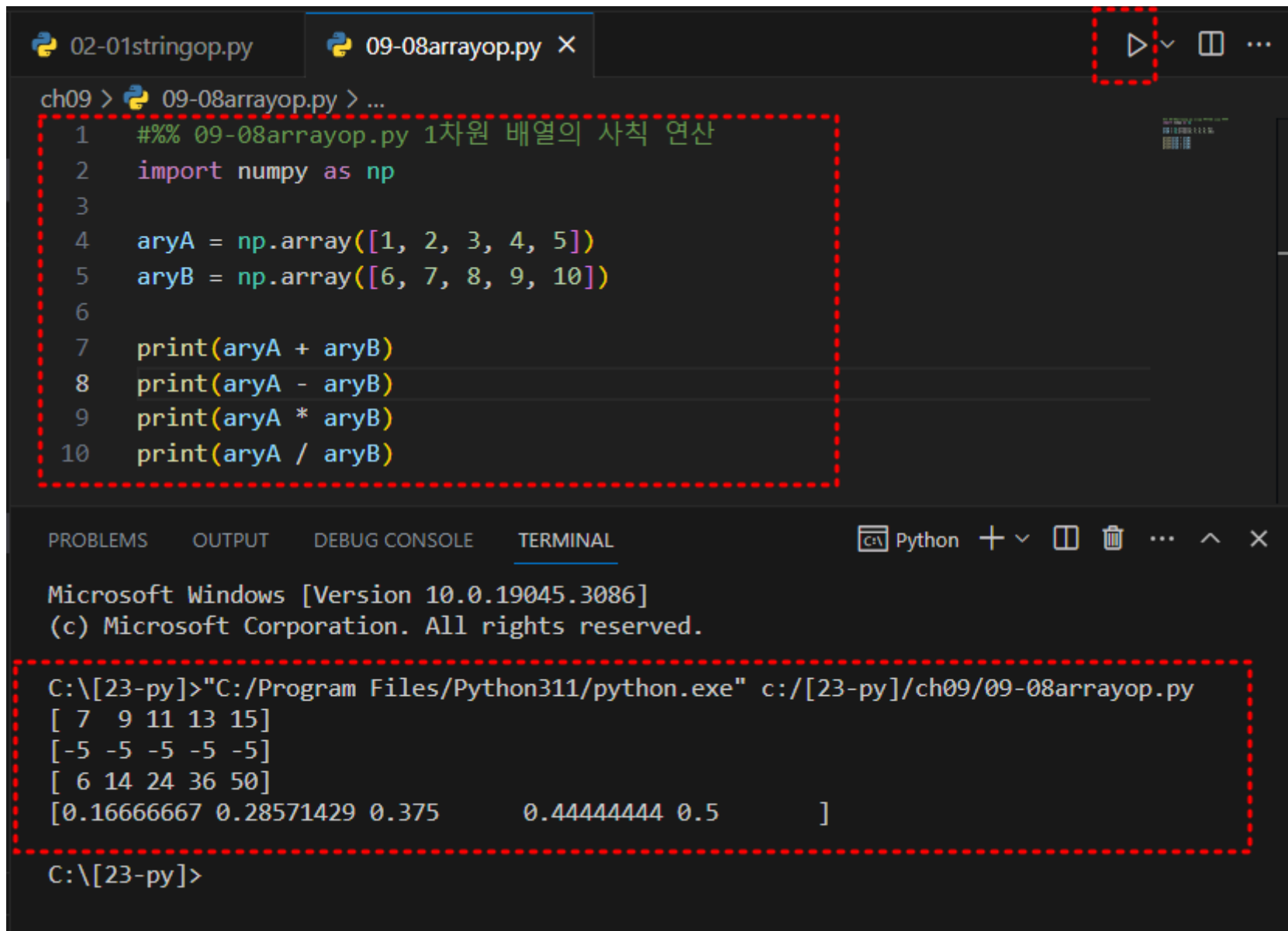
```
C:\[23-py]>where python
C:\Program Files\Python311\python.exe
D:\Python3102\python.exe
C:\Program Files\Python38\python.exe
C:\Users\PC\AppData\Local\Microsoft\WindowsApps\python.exe
```

```
C:\[23-py]>pip list
Package Version
-----
aiofiles 22.1.0
aiosqlite 0.18.0
anyio 3.6.2
argon2-cffi 21.3.0
```

```
C:\[23-py]>pip install numpy
Defaulting to user installation because normal site-packages is not writeable
Collecting numpy
  Downloading numpy-1.24.3-cp311-cp311-win_amd64.whl (14.8 MB)
    14.8/14.8 MB 19.9 MB/s eta 0:00:00
Installing collected packages: numpy
  WARNING: The script f2py.exe is installed in 'C:\Users\PC\AppData\Roaming\Python\Python311\Scripts' which is not on PATH.
    Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed numpy-1.24.3
```

```
C:\[23-py]>pip show numpy
Name: numpy
Version: 1.24.3
Summary: Fundamental package for array computing in Python
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD-3-Clause
Location: C:\Users\PC\AppData\Roaming\Python\Python311\site-packages
Requires:
Required-by:
```





The image shows a VS Code editor window with two tabs: '02-01stringop.py' and '09-08arrayop.py'. The '09-08arrayop.py' tab is active, showing a Python script. The script is enclosed in a red dashed box. The script defines two 1D arrays, 'aryA' and 'aryB', and performs element-wise addition, subtraction, multiplication, and division. The output of the script is shown in the terminal window at the bottom, also enclosed in a red dashed box. The terminal shows the command to run the script and the resulting output arrays.

```
ch09 > 09-08arrayop.py > ...
1  #%% 09-08arrayop.py 1차원 배열의 사칙 연산
2  import numpy as np
3
4  aryA = np.array([1, 2, 3, 4, 5])
5  aryB = np.array([6, 7, 8, 9, 10])
6
7  print(aryA + aryB)
8  print(aryA - aryB)
9  print(aryA * aryB)
10 print(aryA / aryB)
```

Microsoft Windows [Version 10.0.19045.3086]  
(c) Microsoft Corporation. All rights reserved.

```
C:\[23-py]>"C:/Program Files/Python311/python.exe" c:/[23-py]/ch09/09-08arrayop.py
[ 7  9 11 13 15]
[-5 -5 -5 -5 -5]
[ 6 14 24 36 50]
[0.16666667 0.28571429 0.375      0.44444444 0.5      ]

C:\[23-py]>
```

AI Experts  
Who Lead  
The Future

# 06

---

## 확장 설치

# Prettier - Code formatter

오픈소스 소프트웨어를 위한 깃과 깃허브 Python language

The screenshot displays the Visual Studio Code interface with the Prettier extension installed. The left sidebar shows the 'EXTENSIONS: MARKETPLACE' view with a search for 'prett'. The main editor area shows the 'Extension: Prettier - Code formatter' details page. The extension is version 2.3.0, published by Prettier, and has 33,277,582 downloads and a 5-star rating. It is described as a 'Code formatter using prettier'. The extension is enabled globally, and the 'Disable' and 'Uninstall' buttons are visible. The 'Details' tab is selected, showing the extension's description: 'Prettier is an opinionated code formatter. It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary.' The supported languages are listed as JavaScript, TypeScript, Flow, JSX, JSON, CSS, SCSS, Less, HTML, Vue, Angular, HANDLEBARS, Ember, Glimmer, GraphQL, Markdown, and YAML. The 'Categories' section shows 'Formatters'. The 'Extension Resources' section includes links to the Marketplace, Repository, License, and Prettier. The 'More Info' section shows the publication date as 1/11/2017.

File Edit Selection View Go Run ... Extension: Prettier - Code formatter - [23-py] - Visual Studio Code

EXTENSIONS: MARKETPLACE

prett

**Prettier - Code formatter** 267ms  
Code formatter using prettier  
Prettier

**Prettify JSON** 1.7M ★ 3  
Visual Studio Code Prettify JSON Extension  
Mohsen Azimi  
Install

**Prettier ESLint** 1.4M ★ 4  
A Visual Studio Extension to format JavaScript  
Rebecca Vest  
Install

**Prettier Now** 422K ★ 4  
VS Code plugin for Prettier Miscellaneous / ...  
Remi Marsal  
Install

**Prettier+** 250K ★ 4  
Prettier (code formatter) for the VS Code.  
Benas Svipas  
Install

**Prettier - Code formatter** 437K ★ 2.5  
Code formatter using prettier  
Simon Siefke

**Pretty Formatter** 230K ★ 3.5  
VS Code extension to format your code using  
mblode  
Install

**Prettier - Code formatter** v9.13.0  
Prettier prettier.io | 33,277,582 | ★★★★★  
Code formatter using prettier  
Disable Uninstall  
This extension is enabled globally.

DETAILS FEATURE CONTRIBUTIONS CHANGELOG RUNTIME STATUS

**Prettier Formatter for Visual Studio Code**

Prettier is an opinionated code formatter. It enforces a consistent style by parsing your code and re-printing it with its own rules that take the maximum line length into account, wrapping code when necessary.

JavaScript · TypeScript · Flow · JSX · JSON  
CSS · SCSS · Less  
HTML · Vue · Angular · HANDLEBARS · Ember · Glimmer  
GraphQL · Markdown · YAML  
Your favorite language?

**Categories**  
Formatters

**Extension Resources**  
Marketplace  
Repository  
License  
Prettier

**More Info**  
Published 1/11/2017,  
04:52:02

Python: Attach using Process Id ([23-py])

The screenshot shows the Visual Studio Code interface with the Python Indent extension installed. The left sidebar displays the 'EXTENSIONS: MARKETPLACE' view with a search for 'python inde'. The main editor area shows the 'Extension: Python Indent' details page.

**EXTENSIONS: MARKETPLACE**

- Python Indent** (Kevin Rose) - 30ms. **Install**
- Python Development Extensi...** (28K, 3 stars) - **Install**
- Python Paste And Indent** (61K, 3.5 stars) - **Install**
- Python extension pack** (22K) - **Install**
- Symdexer (Python)** (89) - **Install**
- Indent Nested Dictionary** (8K, 5 stars) - **Install**
- nudge** (235) - **Install**

**Extension: Python Indent** (v1.18.0) by Kevin Rose (5,370,158 downloads, 47 reviews)

Correct Python indentation

**Disable** **Uninstall** **Settings**

This extension is enabled globally.

**DETAILS** | FEATURE CONTRIBUTIONS | CHANGELOG | RUNTIME STATUS

**Python Indent**

Correct Python indentation in Visual Studio Code. See the extension on the [VSCode Marketplace](#) and its source code on [GitHub](#).

**Categories**

- Programming Languages
- Keymaps
- Formatters

**Extension Resources**

- Marketplace
- Repository
- License
- Kevin Rose

**More Info**

**demo.py**

```
data = {'a': 0,
        'b': [[1, 2],
               [3, 4]],
        'c': 5}

def hello(
    first:
```

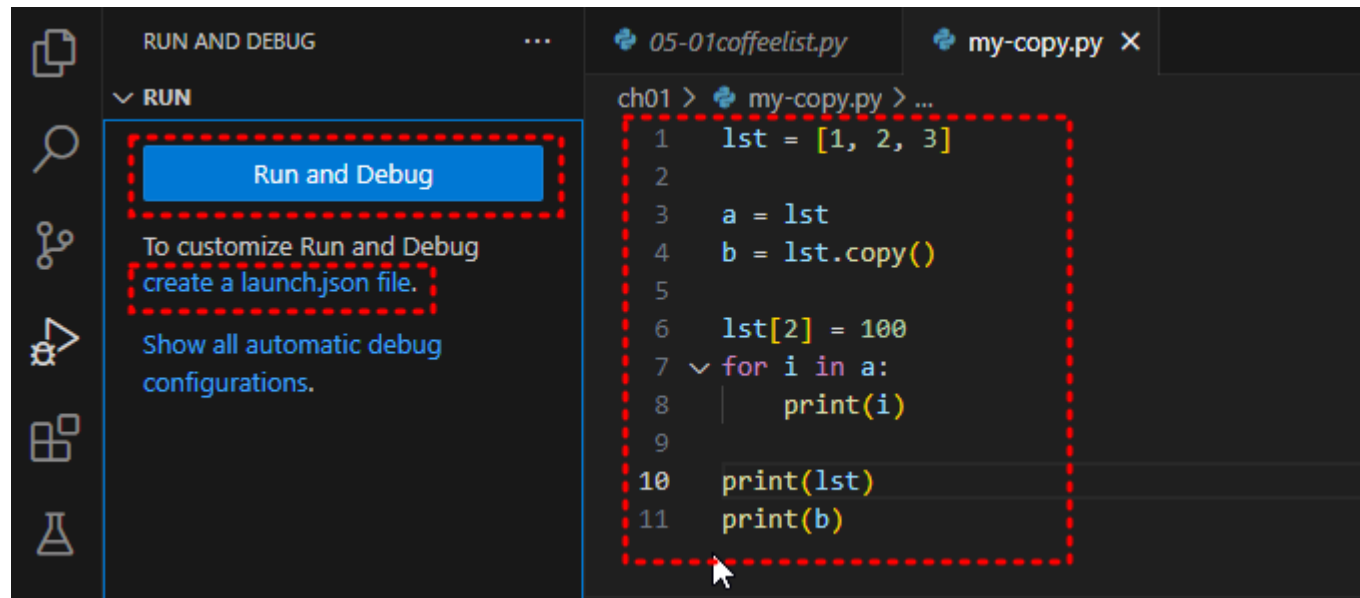
AI Experts  
Who Lead  
The Future

# 05

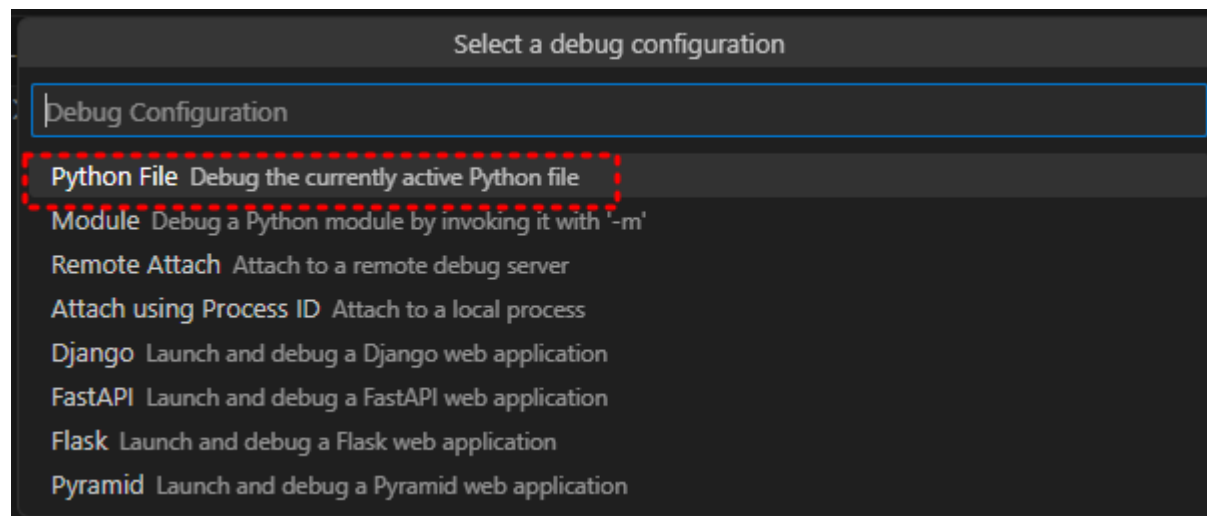
---

## 파이썬 디버깅

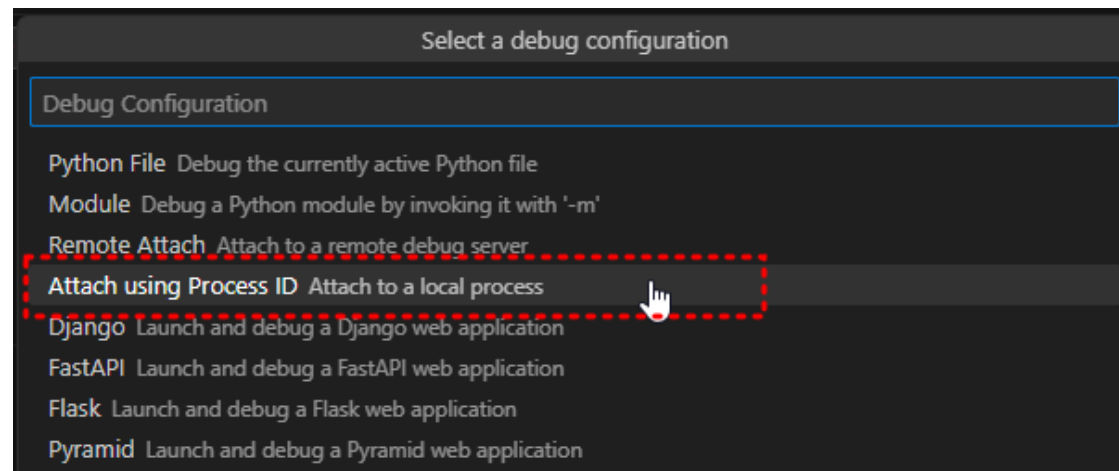
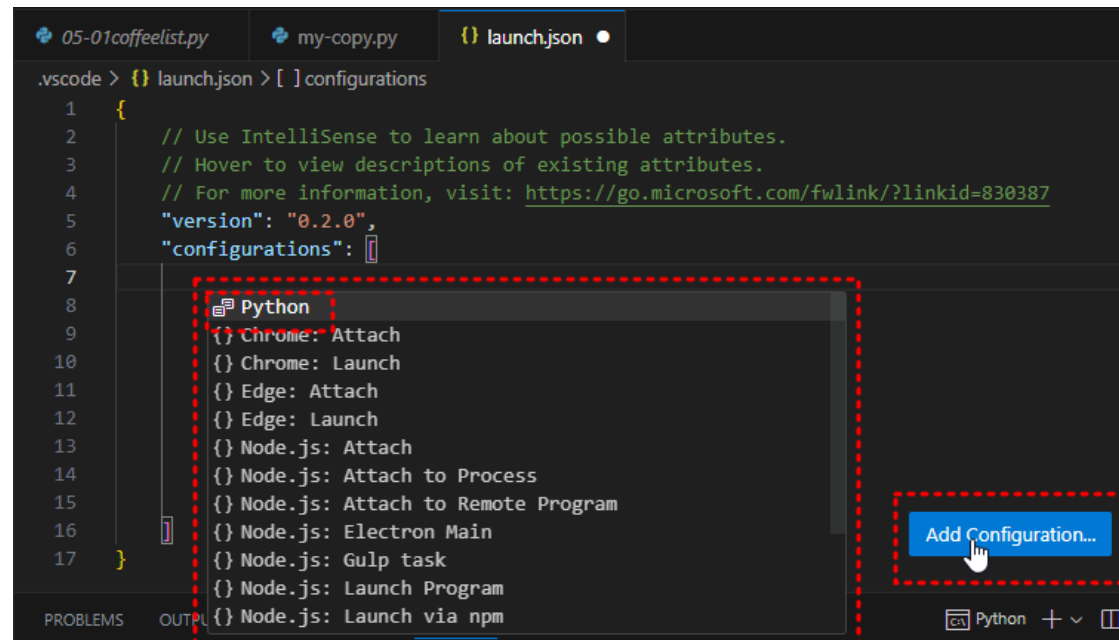
- 아이콘 Run and Debug



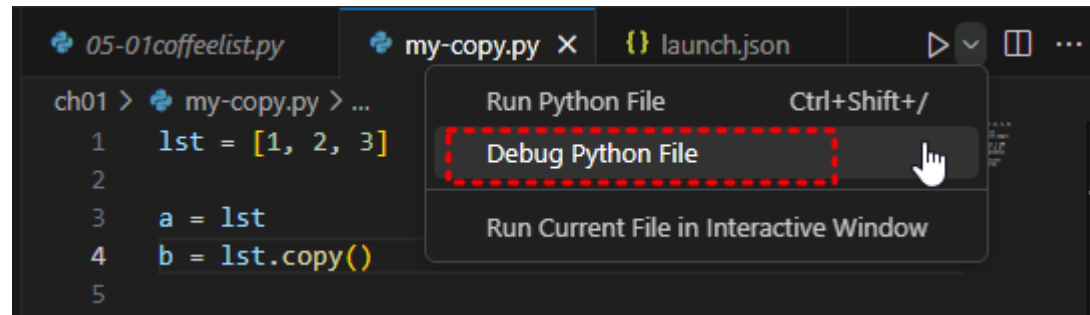
- Create a launch.json file



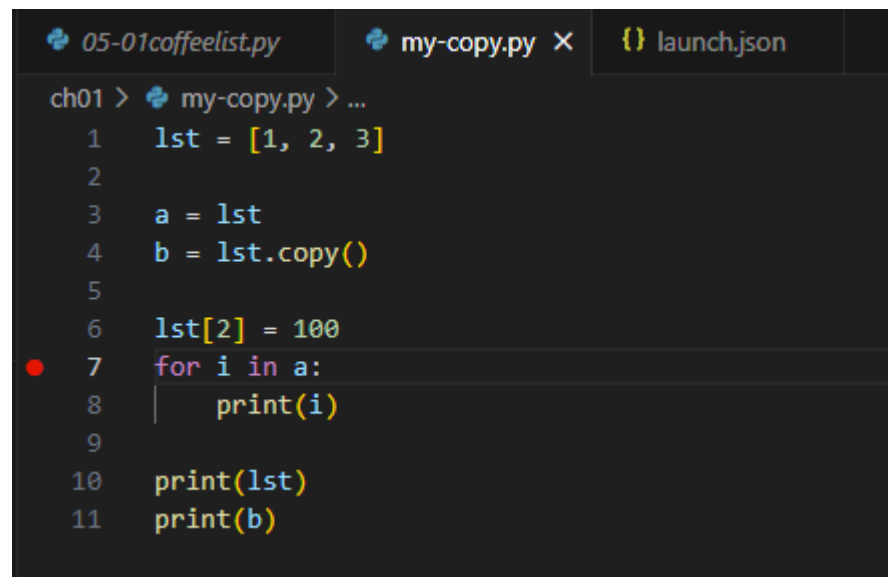
- Add Configurations



- Break point가 없으면 모두 실행

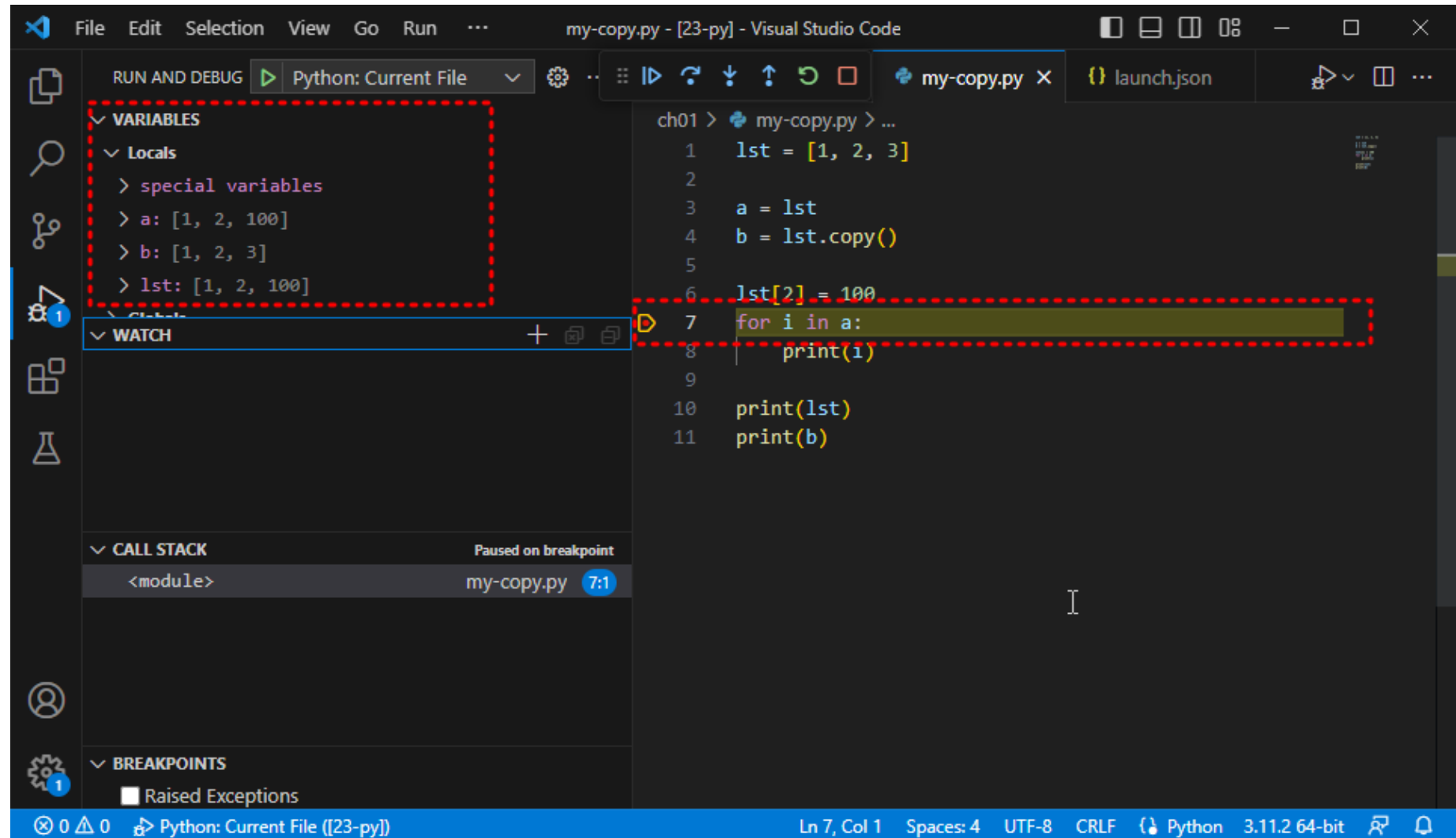


- Break point 지정





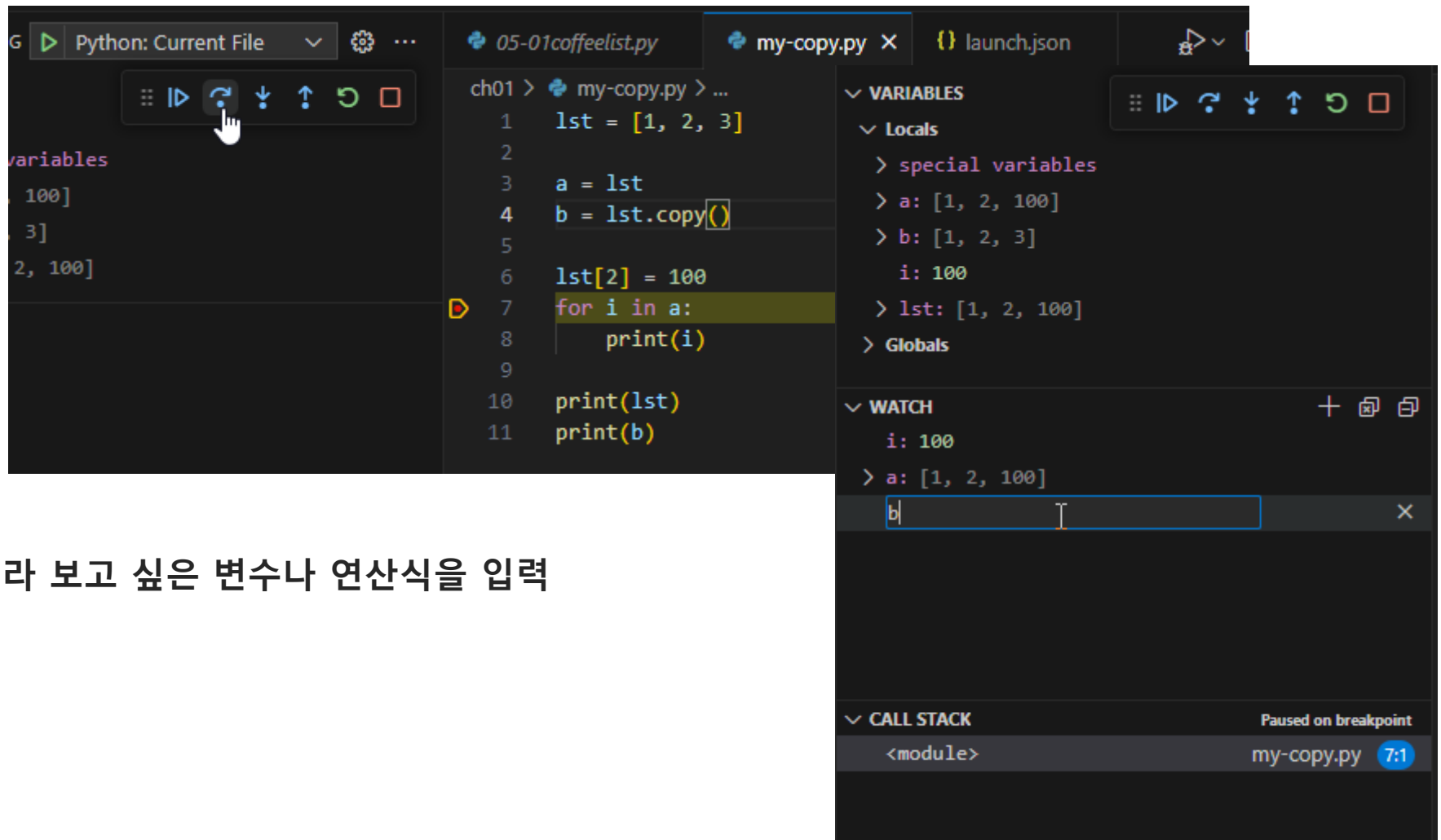
- F5 실행





액션.	설명.
계속/일시 중지 F5	<b>계속:</b> 다음 중단점까지 정상적인 프로그램/스크립트 실행을 재개합니다. <b>일시 중지:</b> 현재 줄에서 실행 중인 코드를 검사하고 줄별로 디버깅합니다.
스텝 오버 F10	구성요소 단계를 검사하거나 따르지 않고 단일 명령으로 다음 방법을 실행합니다.
스텝 인 F11	실행을 한 줄씩 수행하려면 다음 메서드를 입력합니다.
스텝 아웃 시프트+F11	메서드 또는 서브루틴 내부에 있는 경우, 단일 명령인 것처럼 현재 메서드의 나머지 행을 완료하여 이전 실행 컨텍스트로 돌아갑니다.
다시 시작 Ctrl+Shift+F5	현재 프로그램 실행을 종료하고 현재 실행 구성을 사용하여 디버깅을 다시 시작합니다.
이제 그만 시프트+F5	현재 프로그램 실행을 종료합니다.

- Step into F11
  - 한 줄씩 이동
- Step Over F5



- Watch
  - 실행때라 보고 싶은 변수나 연산식을 입력