

# 오픈소스 SW 입문

w6 VS Code



2023-0504, 강의실(과학관213)

강사 : 박노현

<https://github.com/nparkcourage/2023-kau-0504>



한국항공대학교

## 목차

### 7교시

15:00

#### ◆ VS Code 사용준비

- VS Code 소개
- VS Code 설치
- WSL에서 VS Code 사용

### 8교시

16:00

#### ◆ VS Code 사용하기

- VS Code 도커 컨테이너 연결
- VS Code에서 Python 사용
- VS Code에서 github 사용

### 9교시

17:00

#### ◆ 예제 실행

- OpenAI 사용하기
- DALL-E 사용 예제
- ChatGPT 사용 예제

## 첫째 시간

### 7교시

15:00

#### ◆ VS Code 사용준비

- VS Code 소개
- VS Code 설치
- WSL에서 VS Code 사용

### 8교시

16:00

#### ◆ VS Code 사용하기

- VS Code 도커 컨테이너 연결
- VS Code에서 Python 사용
- VS Code에서 github 사용

### 9교시

17:00

#### ◆ 예제 실행

- OpenAI 사용하기
- DALL-E 사용 예제
- ChatGPT 사용 예제

## VS Code 소개

### VS Code?

- VS Code(Visual Studio Code)는 마이크로소프트에서 개발한 무료 및 오픈소스 통합 개발 환경(IDE)
- <https://github.com/microsoft/vscode>
- License : MIT License
- 다운로드 사이트 : <https://code.visualstudio.com/Download>
- 온라인 사용 : <https://vscode.dev/>
- github에서 사용 : 소스파일을 열고 -> .

### VS Code의 특징

- 크로스 플랫폼 : Windows, Linux, Mac OS
- 경량 : 설치 모듈과 실행 모듈의 자원 사용이 비교적 적음(< eclipse, IntelliJ, pycharm)
- 확장성 : 플러그인으로 기능 확장
- 원격 접속 : 클라이언트/서버 구조로 원격 접속 사용 지원

### 많이 사용하는 언어

- 대부분의 언어 지원
- Javascript : Type Script
- python

# W6 VS Code : 사용준비

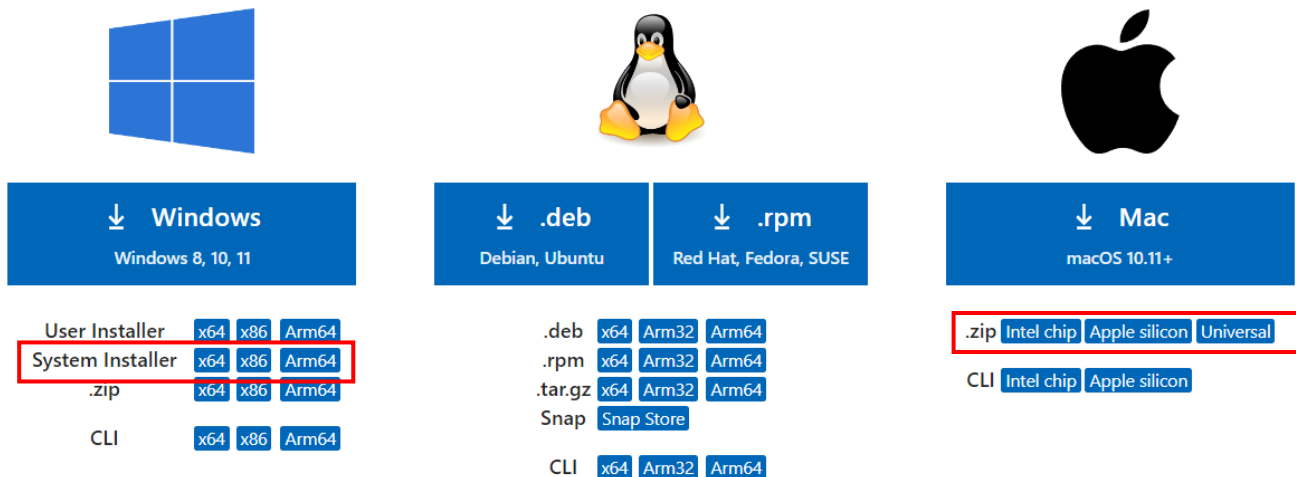
## VS Code 설치

### 설치하기

- 다음 다운로드 사이트에서 자신의 os에 맞는 설치파일을 다운로드하여 설치
- 다운로드 사이트 : <https://code.visualstudio.com/Download>

## Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



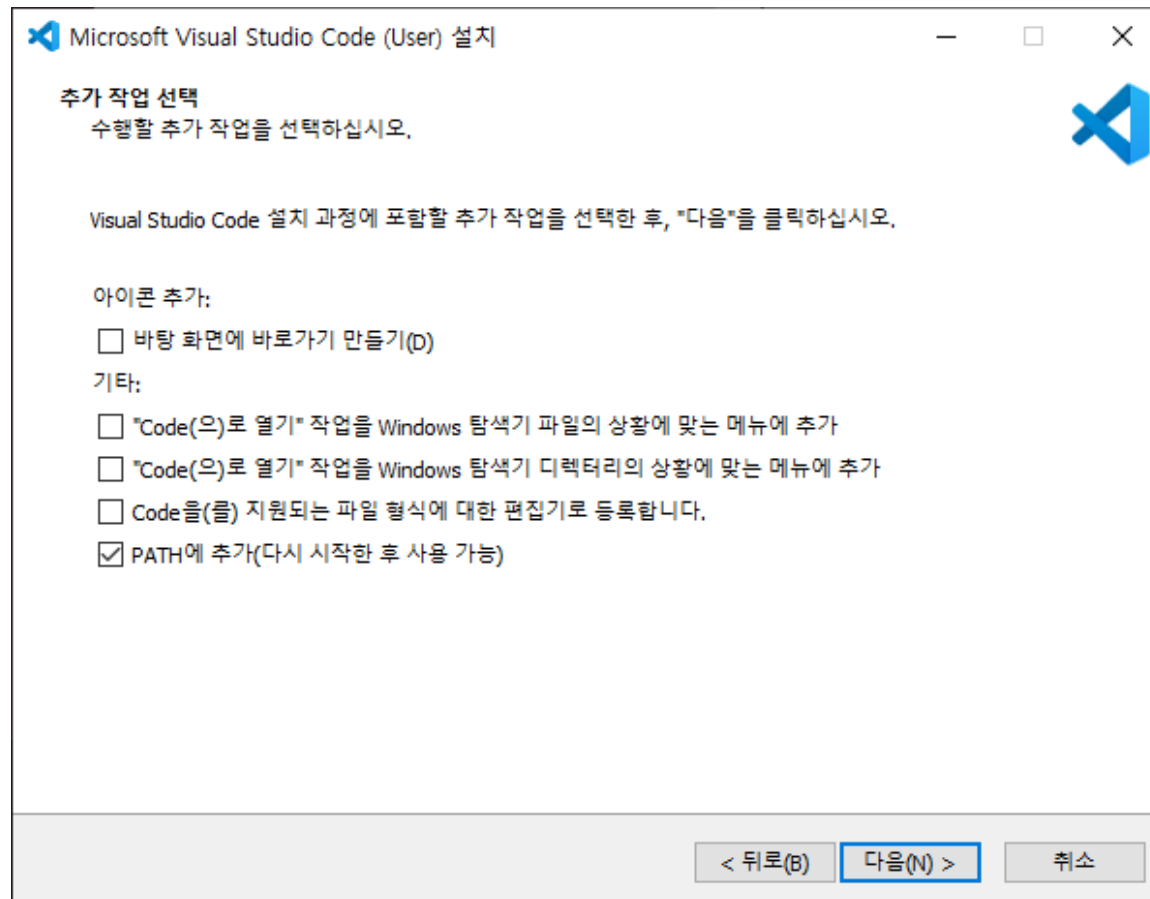
The screenshot displays the Visual Studio Code download page with three main sections: Windows, Linux, and Mac. Each section has a download button and a list of available installers. The 'System Installer' for Windows and the '.zip' file for Mac are highlighted with red boxes.

OS	Installer Type	Architecture
Windows	User Installer	x64, x86, Arm64
	System Installer	x64, x86, Arm64
	.zip	x64, x86, Arm64
	CLI	x64, x86, Arm64
Linux	.deb	x64, Arm32, Arm64
		x64, Arm32, Arm64
	.rpm	x64, Arm32, Arm64
		x64, Arm32, Arm64
	.tar.gz	x64, Arm32, Arm64
	Snap	Snap Store
	CLI	x64, Arm32, Arm64
	Mac	.zip
CLI		Intel chip, Apple silicon

## VS Code 설치

### 설치하기(Windows ) - 옵션 선택

- 설치 중에 추가 작업 선택 메시지가 표시되면 PATH에 추가 옵션을 선택
- WSL에서 실행할 때 PATH 필요

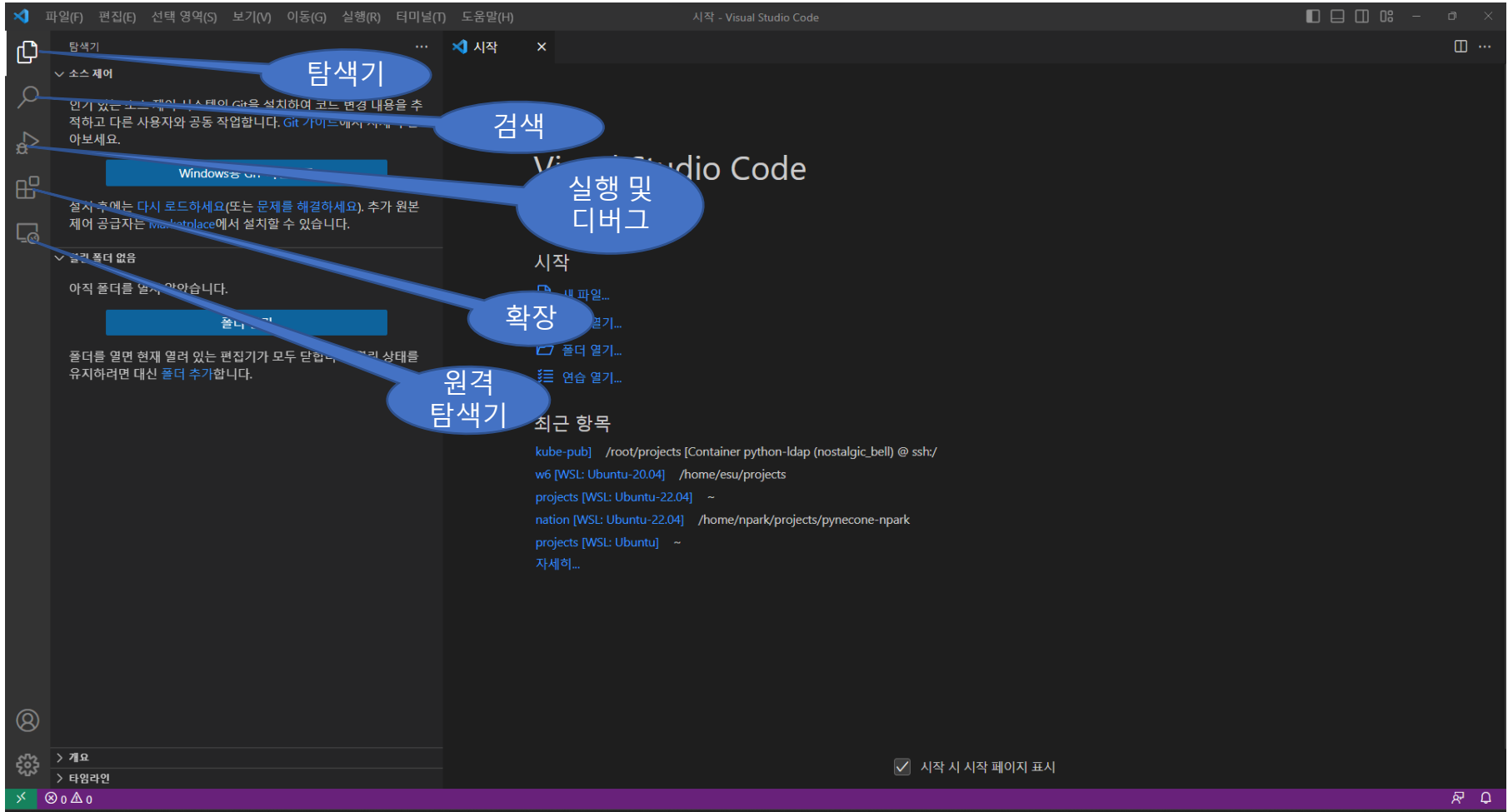


# W6 VS Code : 사용준비

## VS Code 설치

### 주요 메뉴

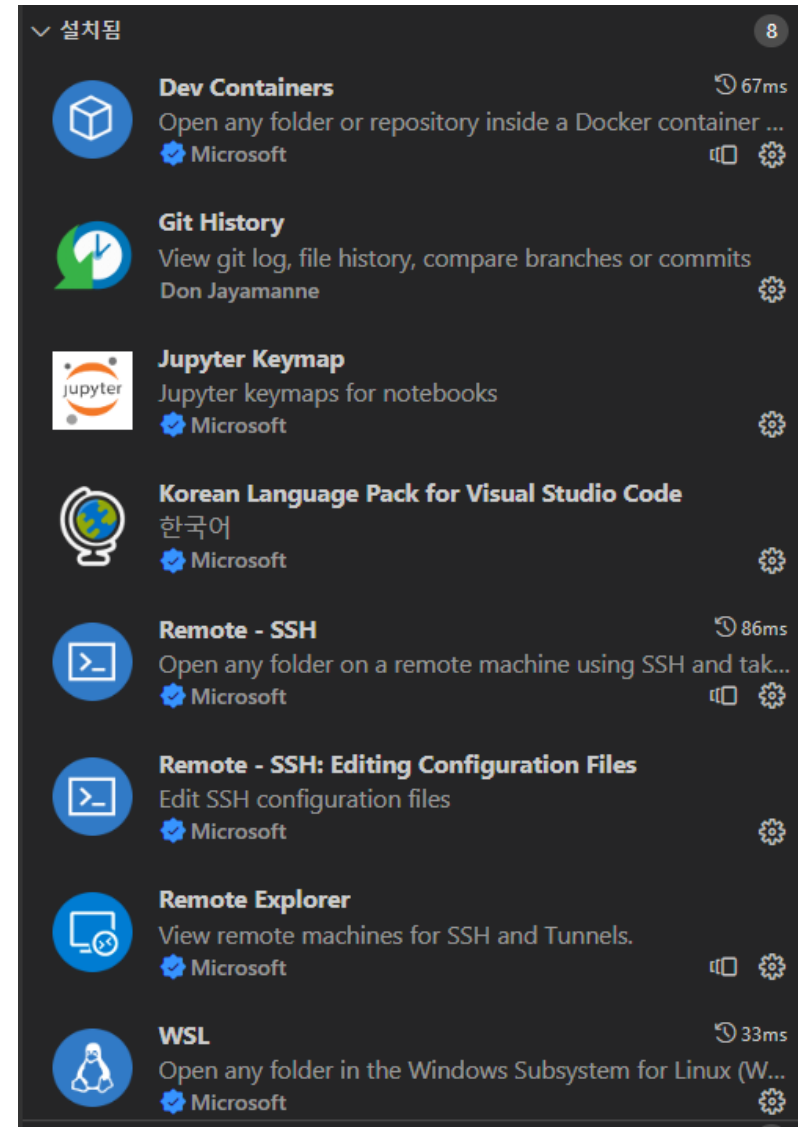
- CTRL+SHIFT+P : 명령 팔레트



## VS Code 설치

### 확장(Plugin) 설치

- 기본(Local)에 다음의 확장들을 (전역 설정으로) 설치
- Korean Language Pack for Visual Studio Code(한국어팩)
- Dev Containers
  - Docker 사용지원
- Remote - SSH
  - Remote - SSH: Editing Configuration Files
  - Remote Explorer
- WSL
- Git History





# W6 VS Code : 사용준비

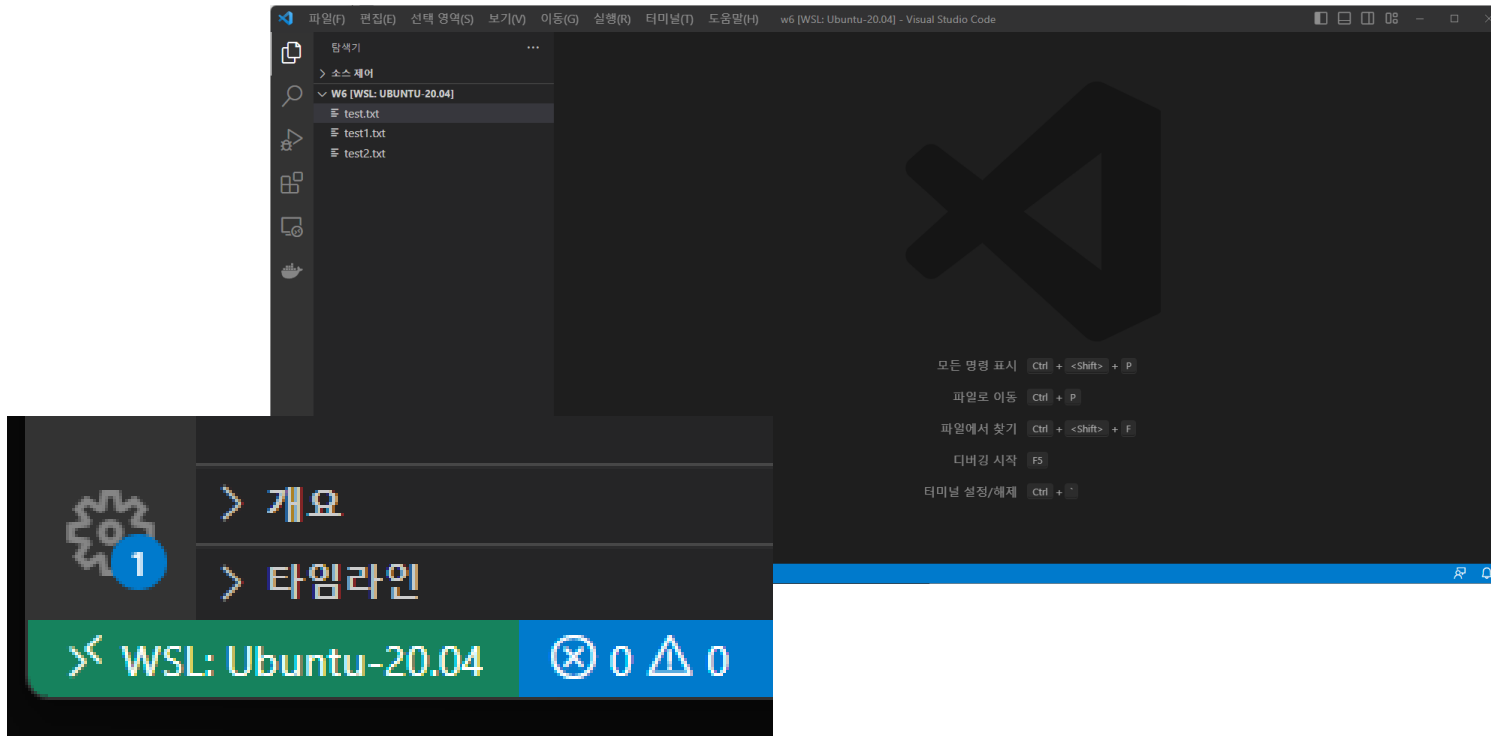
## WSL에서 VS Code 사용

### WSL에서 실행

- WSL 터미널 열기
- 프로젝트 디렉토리 만들기
- 프로젝트 디렉토리로 이동
- VS Code 실행

### WSL에서 실행

```
mkdir -p ~/projects  
cd ~/projects  
code .
```

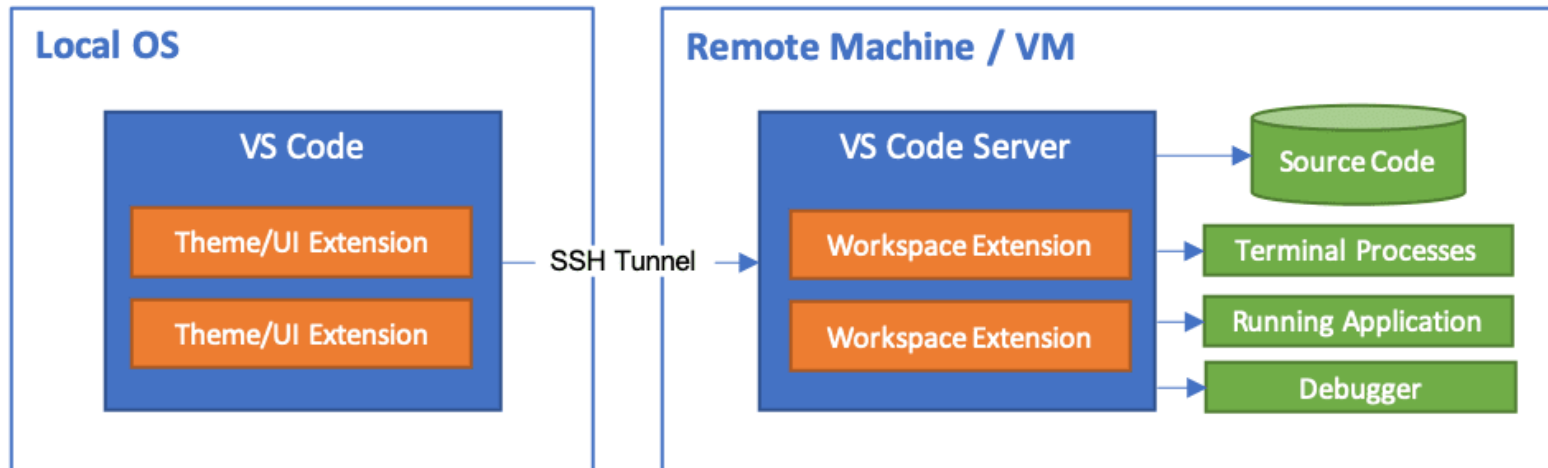


<https://learn.microsoft.com/ko-kr/windows/wsl/tutorials/wsl-vscode>

## WSL에서 VS Code 사용

### VS Code 원격 사용

- VS Code는 클라이언트/서버 구조의 시스템으로 원격 사용을 지원
- Client(GUI)는 윈도우에서 실행
- 서버는 원격(WSL, 원격 시스템 등)에서 실행
- WSL에서 사용할 경우 code가 경로에 있어야 함
  - ex) which code -> /mnt/c/Program Files/Microsoft VS Code/bin/code
- 트러블 슈팅
- [https://code.visualstudio.com/docs/remote/troubleshooting#\\_fixing-problems-with-the-code-command-not-working](https://code.visualstudio.com/docs/remote/troubleshooting#_fixing-problems-with-the-code-command-not-working)



<https://code.visualstudio.com/docs/remote/ssh>

## 둘째 시간

7교시

15:00

### ◆ VS Code 사용준비

- VS Code 소개
- VS Code 설치
- WSL에서 VS Code 사용

8교시

16:00

### ◆ VS Code 사용하기

- VS Code 도커 컨테이너 연결
- VS Code에서 Python 사용
- VS Code에서 github 사용

9교시

17:00

### ◆ 예제 실행

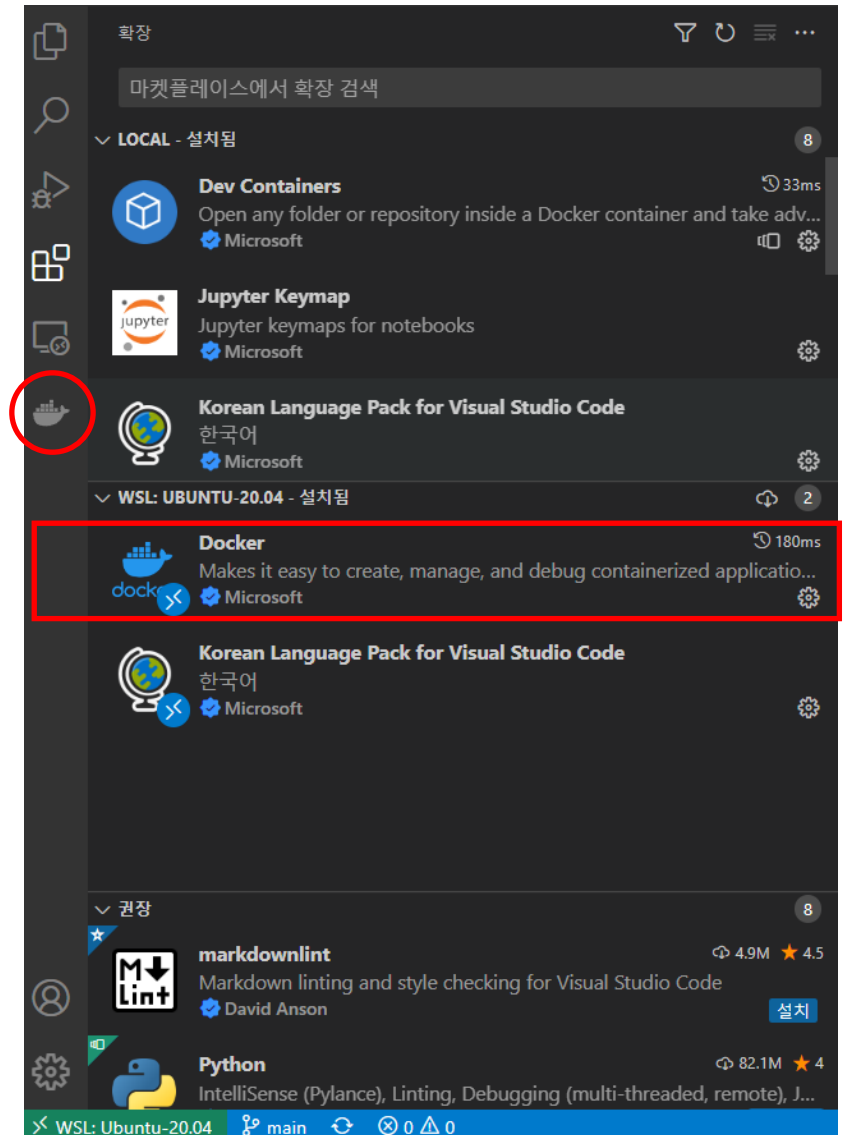
- OpenAI 사용하기
- DALL-E 2 사용 예제
- ChatGPT 사용 예제

# W6 VS Code : 사용하기

## 도커 컨테이너 연결

### 도커 플러그인 설치

- VS Code의 확장에서 WSL 부분에 Docker Plugin 설치



# W6 VS Code : 사용하기

## 도커 컨테이너 연결

### 컨테이너 실행

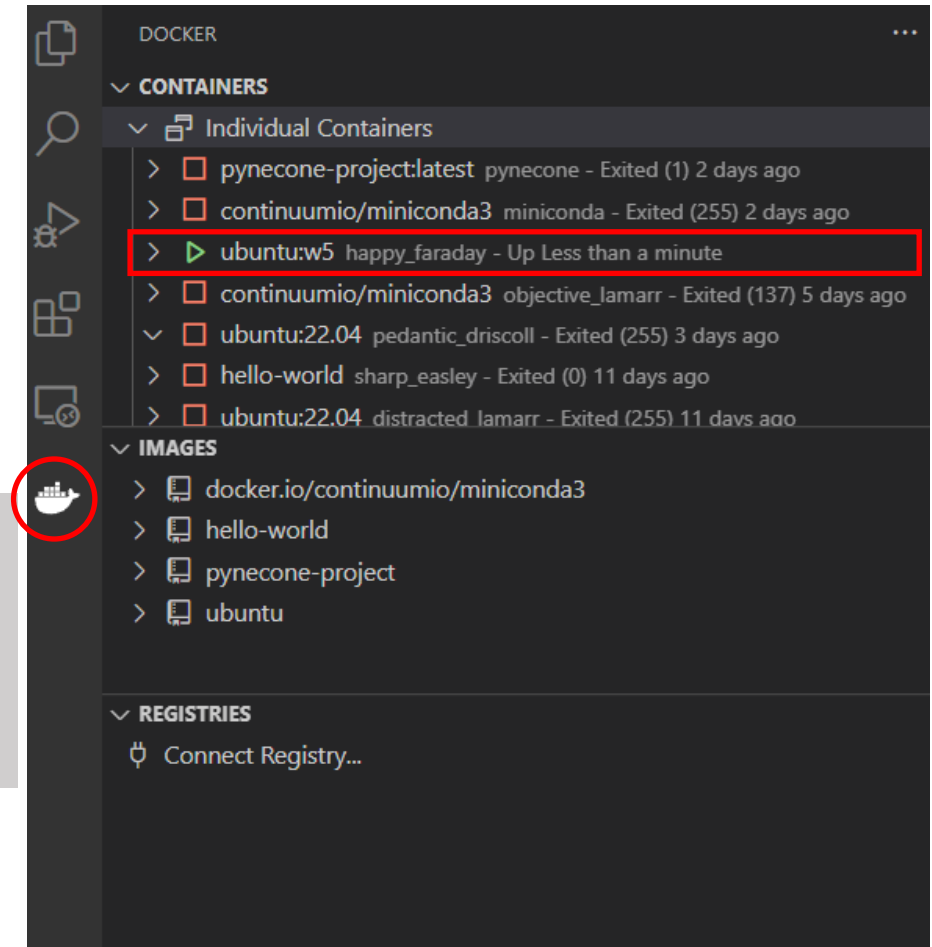
- 지난 강의(w5, 21쪽)에 사용한 컨테이너 실행
- `docker ps -a` 명령으로 실행할 컨테이너 확인
- `docker start`로 실행
- 컨테이너를 실행 한 후 VS Code에서 고래 아이콘 클릭
- 실행 중인 컨테이너가 녹색 삼각형으로 표시

### WSL에서 도커 컨테이너 실행

```
sudo service docker start
```

```
docker ps -a -> 실행할 컨테이너 확인
```

```
docker start <container name>
```

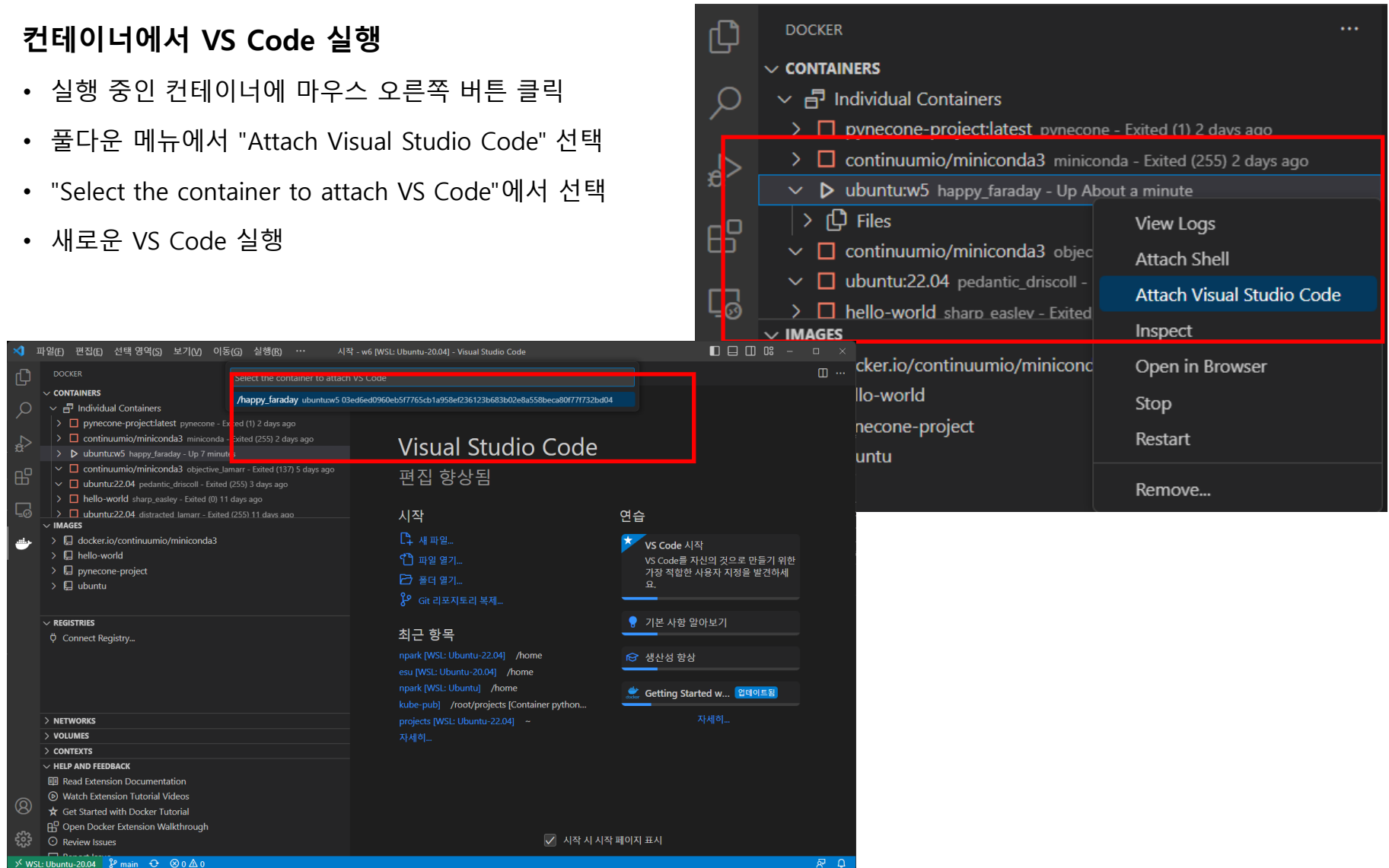


# W6 VS Code : 사용하기

## 도커 컨테이너 연결

### 컨테이너에서 VS Code 실행

- 실행 중인 컨테이너에 마우스 오른쪽 버튼 클릭
- 풀다운 메뉴에서 "Attach Visual Studio Code" 선택
- "Select the container to attach VS Code"에서 선택
- 새로운 VS Code 실행



# W6 VS Code : 사용하기

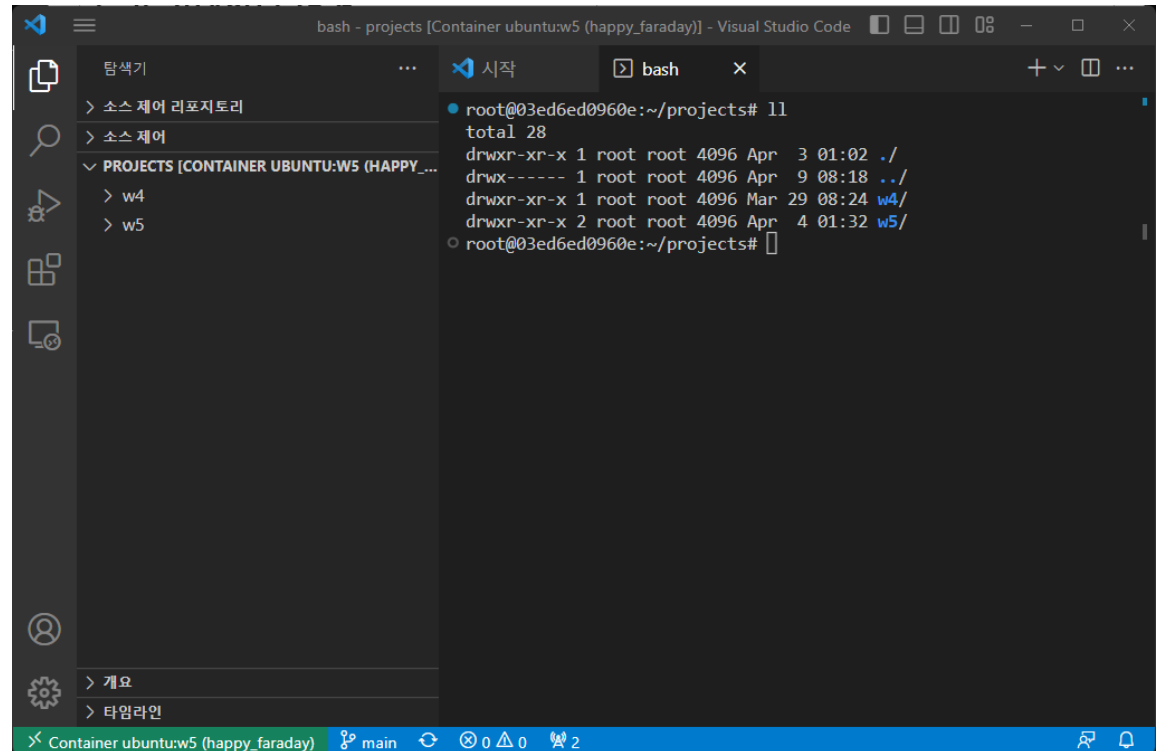
## 도커 컨테이너 연결

### 폴더 열기

- 폴더 열기(Ctrl + O)
- /root/projects 선택

### VS Code Terminal 열기

- 터미널>새 터미널 또는 Ctrl + ` (백틱)
- 드래그하여 위로 이동



## Python 사용

### github 저장소 만들기

- Repository name : w6
- Public : 체크(공개 저장소)
- Add a README file : 체크
- Add .gitignore : Python
- Choose a license : MIT License

### 컨테이너에서 클론

- 컨테이너의 ~/projects에서 클론

### 컨테이너에서 github 저장소 클론

```
cd ~/projects  
git clone <git_ssh_url>
```



# W6 VS Code : 사용하기

## Python 사용

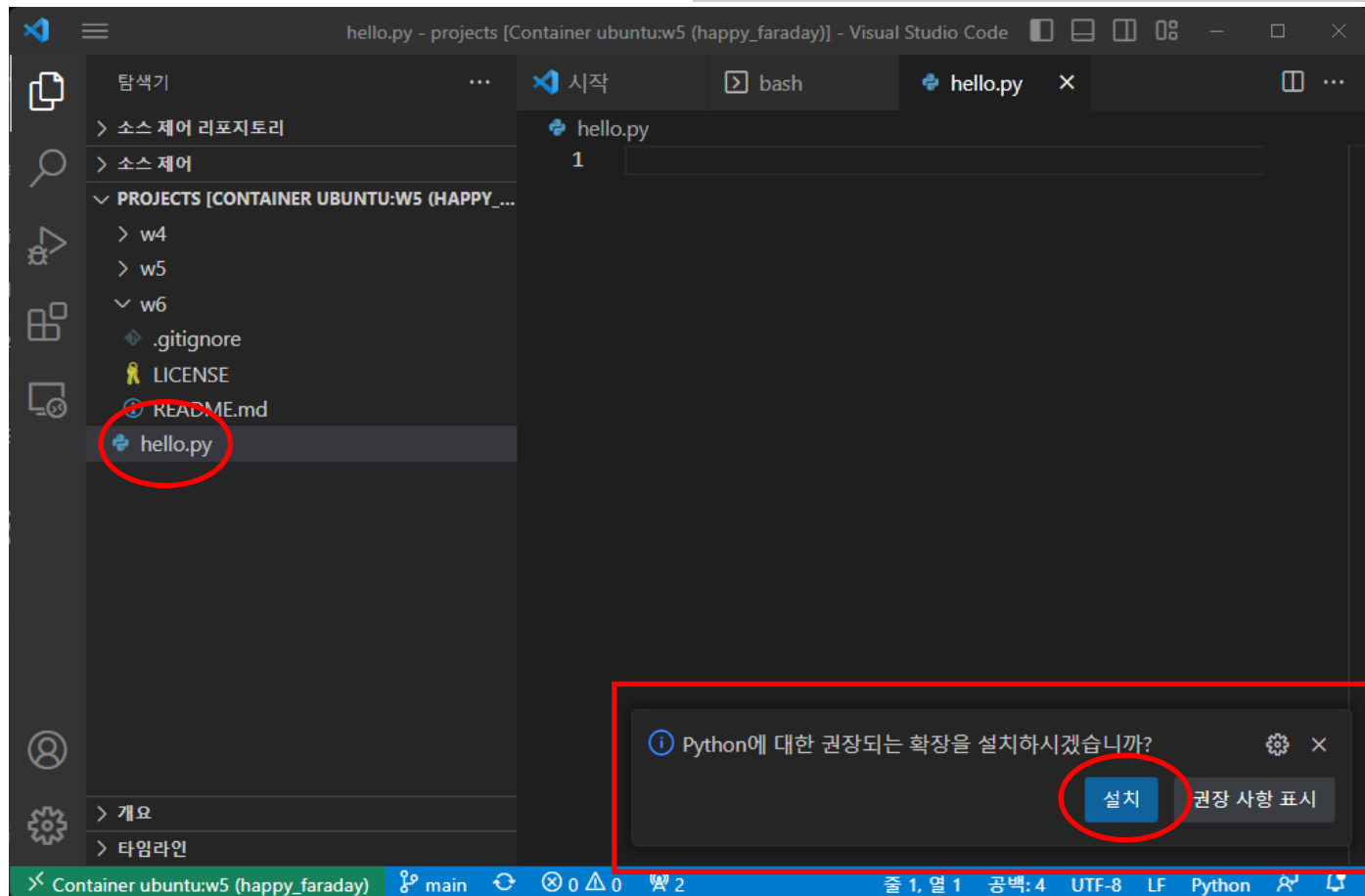
### VS Code에서 Python 사용

- Python 지원 플러그인 설치 필요
- python 파일을 만들면 설치 권장

### VS Code w6 폴더에서 파이썬 파일 추가

```
~/projects/w6/hello.py
```

-> VS Code에서 권장 설치

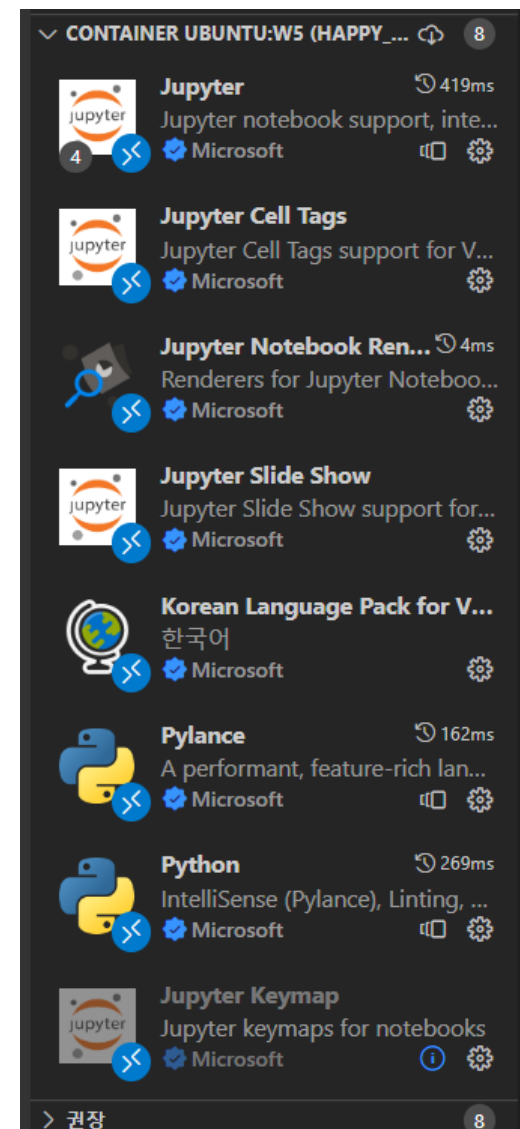
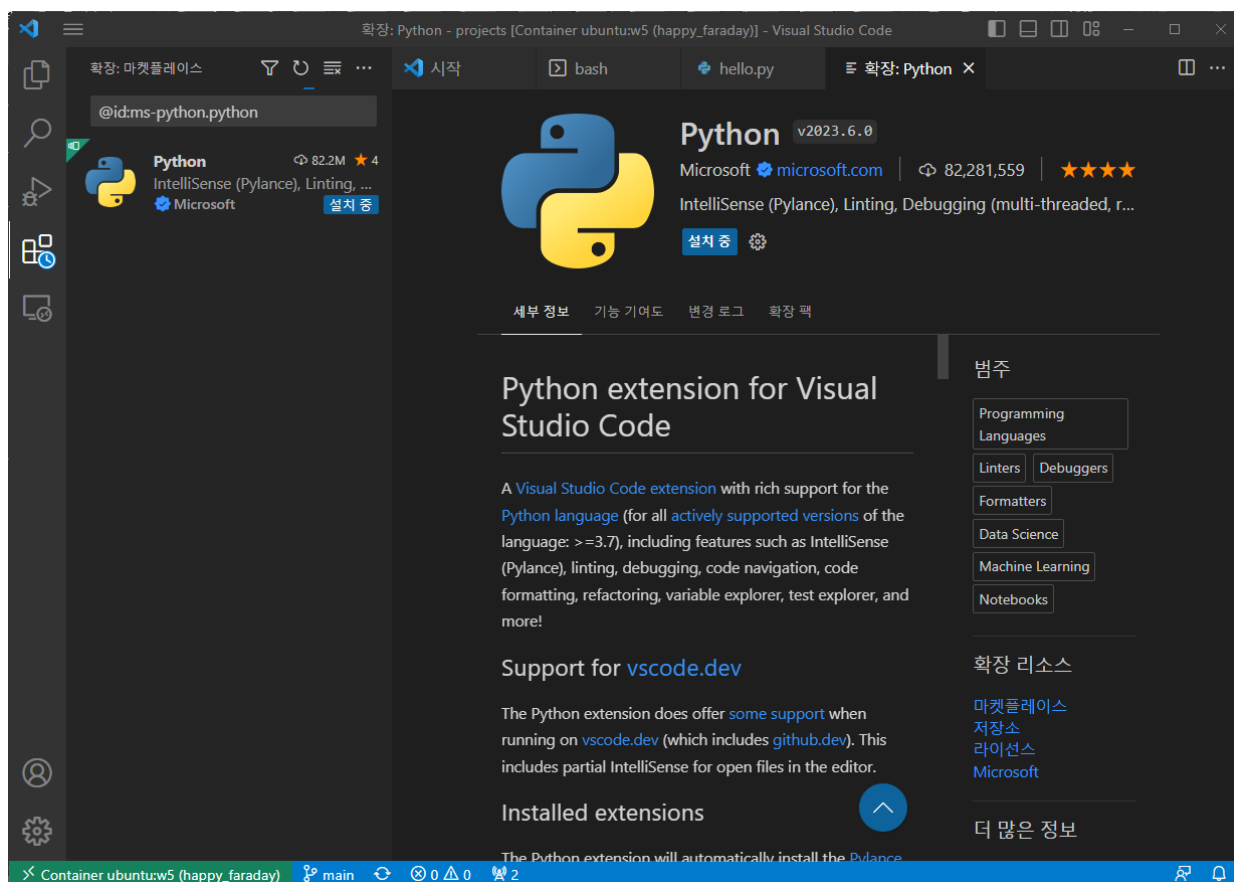


# W6 VS Code : 사용하기

## Python 사용

### VS Code에서 Python 사용

- 권장 설치를 하면 파이썬에 필요한 많은 플러그인들이 설치됨



# W6 VS Code : 사용하기

## Python 사용

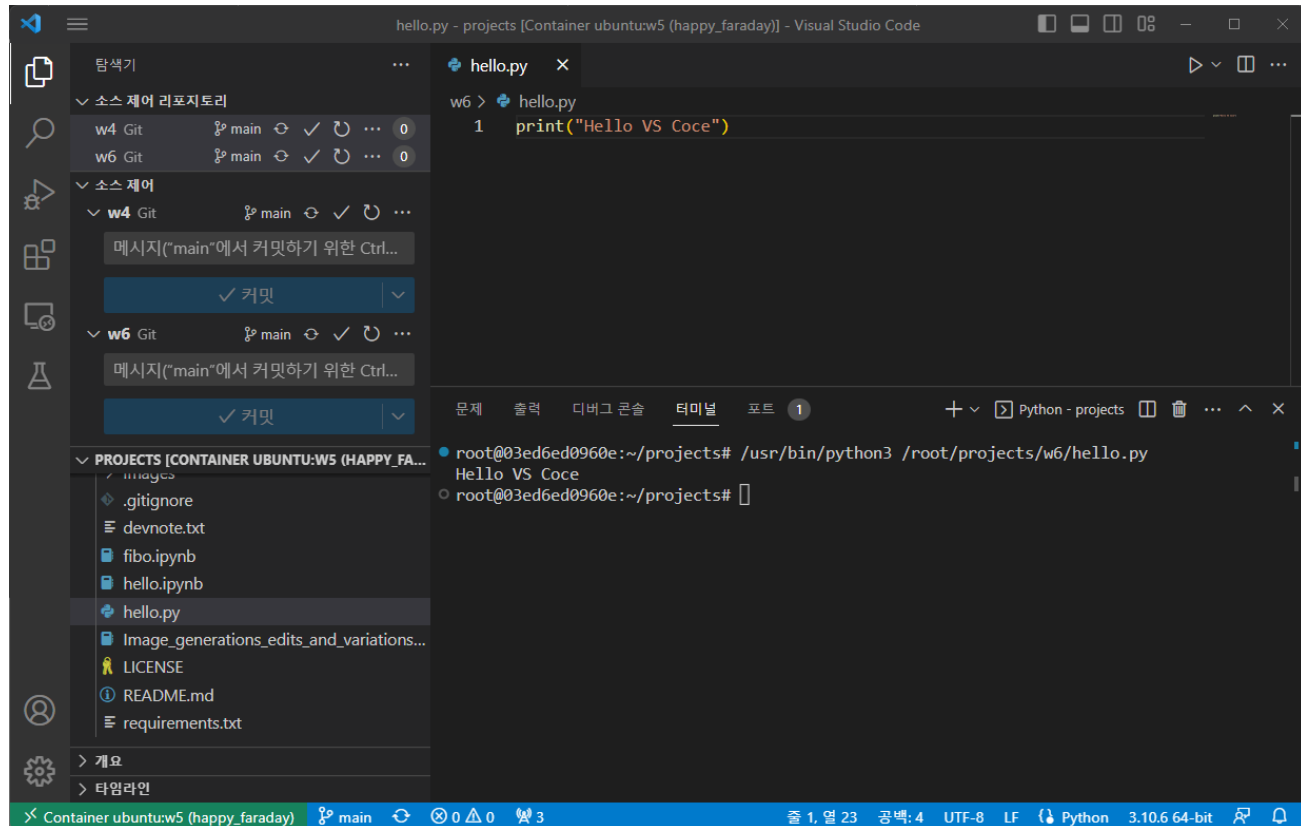
### 코딩

- VS Code 편집기로 hello.py 파일을 열고 코드 내용 입력 -> 저장

### 실행

- 일반 파이썬 실행하듯 터미널에서 실행

python3 hello.py



# W6 VS Code : 사용하기

## Python 사용

### Jupyter Notebook 파일 실행

- hello.ipynb 파일 생성
- 실행 -> ipykernel와 pip 설치 메시지 나타남
- ipykernel을 설치하기 위해서는 pip 필요
- 먼저 pip 설치 : apt install pip
- ipykernel 설치 메시지에서 "설치" 버튼 클릭

### Jupyter Notebook 파일 실행

- Notebook Cell에 코드 입력  

```
print("Hello Jupyter")
```
- VS Code에서 hello.ipynb 파일 실행 버튼 클릭  
(Shift + Enter)

### VS Code에서 notebook file 만들고 실행

```
~/projects/w6/hello.ipynb
```

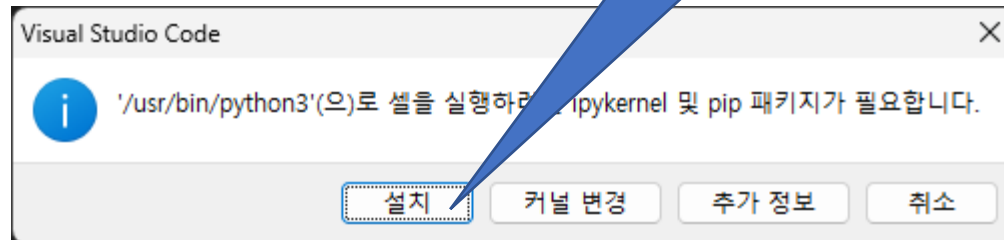
```
-> ipykernel 설치 메시지
```

### 컨테이너 터미널에서 pip 설치

```
apt update
```

```
apt install pip
```

pip를 먼저 설치하고  
"설치"버튼 클릭

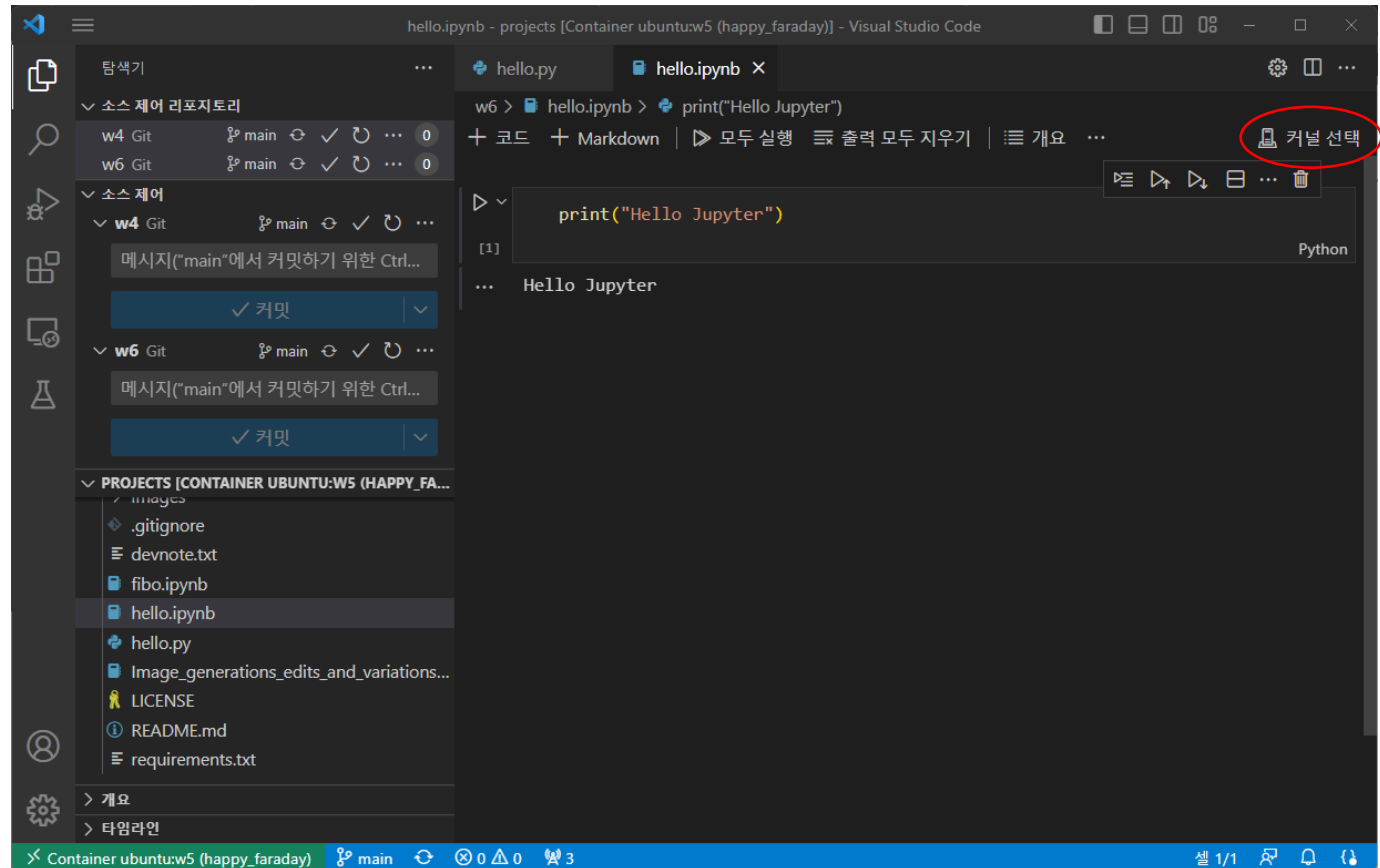


# W6 VS Code : 사용하기

## Python 사용

### 커널 선택

- jupyter notebook 파일 실행을 위해서는 커널 지정 필요
- 오른쪽 위에 "커널 선택" 버튼 있음
  - > 적합한 커널 선택



# W6 VS Code : 사용하기

## github 사용

### VS Code에서 github 사용

- VS Code에는 git 기능이 포함되어 있음
- git 저장소 및 변동 사항 자동 감지

### git 사용

- 변경 사항 스테이징
  - git add
- 커밋
  - git commit
- 푸시
  - git push

### VS Code에서 git 사용

스테이징

변경사항의 "+" 버튼 클릭

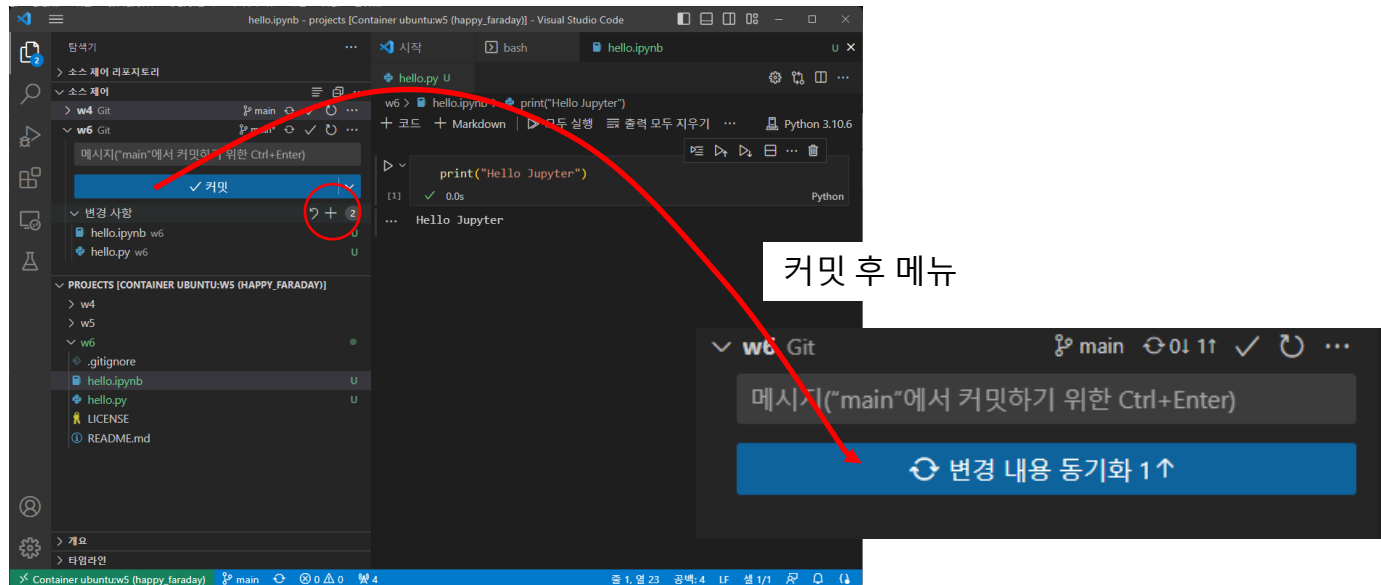
커밋

메시지 입력

"커밋" 버튼 클릭

푸시

"변경 내용 동기화" 버튼 클릭



## 세째 시간

7교시

15:00

### ◆ VS Code 사용준비

- VS Code 소개
- VS Code 설치
- WSL에서 VS Code 사용

8교시

16:00

### ◆ VS Code 사용하기

- VS Code 도커 컨테이너 연결
- VS Code에서 Python 사용
- VS Code에서 github 사용

9교시

17:00

### ◆ 예제 실행

- OpenAI 사용하기
- DALL-E 사용 예제
- ChatGPT 사용 예제

## OpenAI 사용하기

### OpenAI

- 2015년 google의 인공지능 독점을 방지하기 위해 비영리 기관으로 시작
- <https://openai.com/>
- 창업 공동 의장
  - 일론 리브 머스크(Elon Reeve Musk) : 테슬라, 스페이스엑스
  - 새뮤얼 H. 올트먼(Samuel H. Altman) : 와이콤비네이터
- 공동설립자, 후원자
  - 레이드 호프먼 : 링크드인
  - 피터 딜 : 페이팔
  - 그레그 브로크먼 : 스트라이프
  - 아마존 웹 서비스
  - 인포시스 : 인도의 IT 컨설팅 회사
- 2019년 마이크로소프트 투자
- GPT1 : 2018년
- GPT3 : 2020년
- ChatGPT : 2022년(GPT3.5 모델 사용)



## OpenAI 사용하기

### OpenAI 사용

- OpenAI의 API를 사용하기 위해서는 API Key가 필요하며 API Key를 만들기 위해 계정이 필요
- 계정 생성
  - <https://beta.openai.com/signup>
- 키 생성
  - <https://platform.openai.com/account/api-keys>
  - key는 다시 읽을 수 없으므로 잘 보관해야 함
- 참고 자료
  - <https://github.com/openai/openai-cookbook>

## OpenAI 사용하기

### DALL-E 예제 실행

- DALL-E는 OpenAI의 이미지 생성 모델
- 주어진 문장(Prompt)에 맞는 이미지를 생성
- 예제 소스 URL(openai-cookbook)
  - [https://github.com/openai/openai-cookbook/blob/main/examples/dalle/Image\\_generations\\_edits\\_and\\_variations\\_with\\_DALL-E.ipynb](https://github.com/openai/openai-cookbook/blob/main/examples/dalle/Image_generations_edits_and_variations_with_DALL-E.ipynb)
- 필요 패키지 설치
  - openai
  - pillow
- 소스 수정
  - API Key 수정 : 자신이 생성한 키 사용
- VS Code에서 실행
- 커널 선택

### 컨테이너에서 필요 패키지 설치

```
pip install openai  
pip install pillow
```

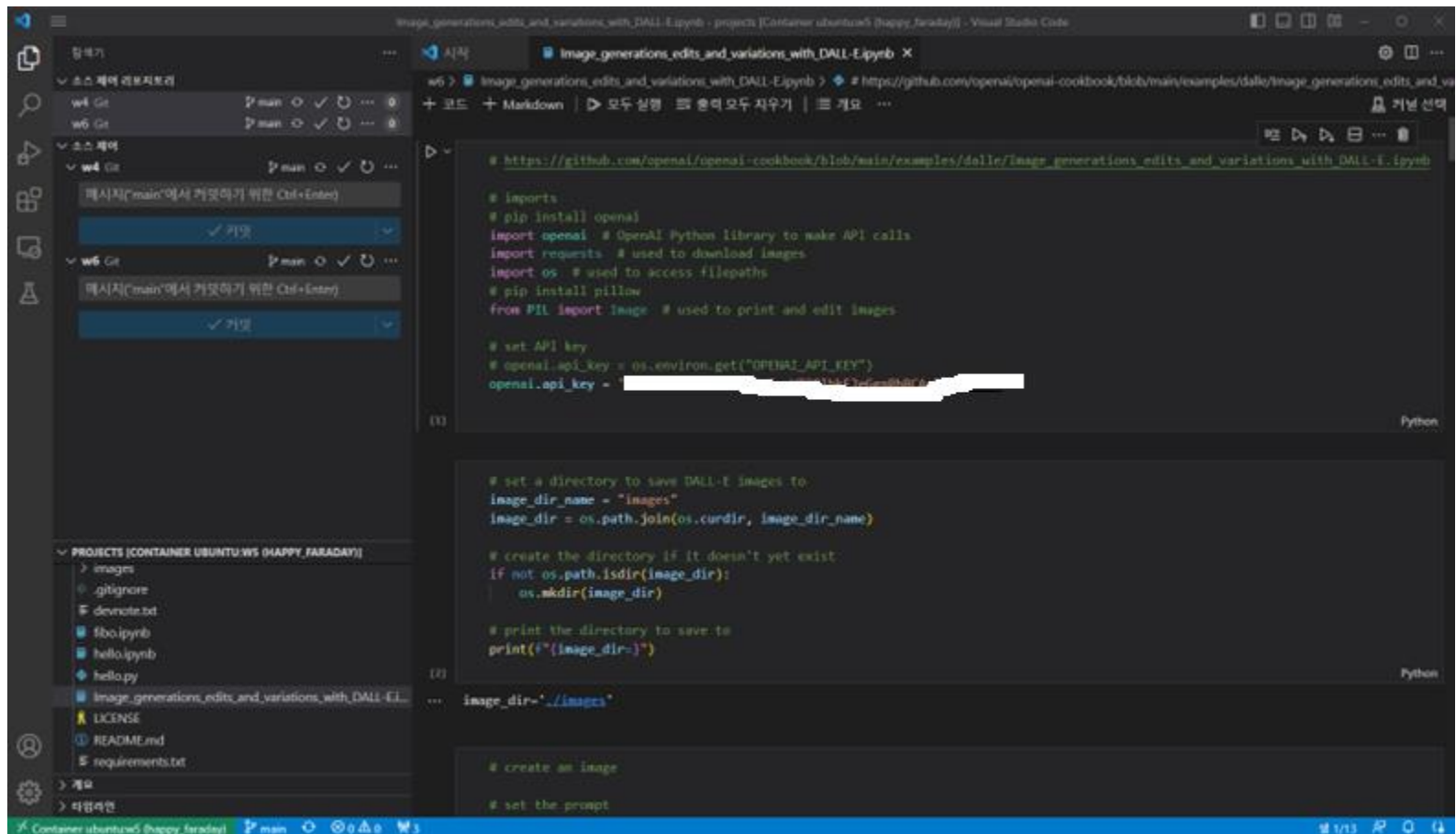
# W6 VS Code : 예제 실행

## OpenAI 사용하기

### DALL-E 예제 실행

- 강의 자료 : notebook source

[https://github.com/nparkcourage/2023-kau-0504/blob/main/w6/Image\\_generations\\_edits\\_and\\_variations\\_with\\_DALL-E.ipynb](https://github.com/nparkcourage/2023-kau-0504/blob/main/w6/Image_generations_edits_and_variations_with_DALL-E.ipynb)



The screenshot shows the Visual Studio Code interface with a Python script titled 'Image\_generations\_edits\_and\_variations\_with\_DALL-E.ipynb'. The script is written in Python and includes comments in Korean. It sets up the OpenAI API key, imports necessary libraries (openai, requests, os, PIL), and defines a directory for saving images. The script is currently at the 'set the prompt' step.

```
# https://github.com/openai/openai-cookbook/blob/main/examples/dalle/image_generations_edits_and_variations_with_DALL-E.ipynb

# imports
# pip install openai
import openai # OpenAI Python library to make API calls
import requests # used to download images
import os # used to access filepaths
# pip install pillow
from PIL import Image # used to print and edit images

# set API key
# openai.api_key = os.environ.get("OPENAI_API_KEY")
openai.api_key = "sk-Cyru-4b8drg"

# set a directory to save DALL-E images to
image_dir_name = "images"
image_dir = os.path.join(os.getcwd(), image_dir_name)

# create the directory if it doesn't yet exist
if not os.path.isdir(image_dir):
    os.mkdir(image_dir)

# print the directory to save to
print(f"{image_dir}")

image_dir = "../images"

# create an image
# set the prompt
```

## OpenAI 사용하기

### ChatGPT 예제 실행

- ChatGPT는 GPT Model을 이용하는 대화형 AI 모델
- 주어진 질문에 답변
- 예제
  - Flask를 이용한 python web service
  - 저장소 : <https://github.com/logankilpatrick/ChatGPT-Simple>
- 필요 패키지 설치 : requirements.txt 파일에 저장
  - openai
  - python-dotenv
  - flask
  - pip install -r requirements.txt 명령으로 설치 가능
- 소스 일부 수정
  - API Key 수정 : 자신이 생성한 키 사용
  - 서비스 포트 수정 : 9090
    - > 끝 줄 수정 : `app.run(debug=True, host='0.0.0.0', port=9090)`
- 컨테이너에서 서버 실행
- 브라우저 접속 : localhost:9090

### 컨테이너에서 실행

```
cd ~/projects
git clone https://github.com/logankilpatrick/ChatGPT-Simple
cd ChatGPT-Simple
pip install -r requirements.txt
python server.py
```

## OpenAI 사용하기

### ChatGPT 예제 실행

- 터미널에서 실행

```
python3 server.py
```

The screenshot shows the Visual Studio Code interface with a project named 'w6' (ChatGPT-Simple) open. The file explorer on the left shows the project structure, including 'templates', '.gitignore', 'LICENSE', 'README.md', 'requirements.txt', and 'server.py'. The main editor displays the 'server.py' file, which contains a Flask application using the OpenAI API. The code defines a 'get\_response()' function that takes a message and returns a completion from the OpenAI model 'gpt-3.5-turbo'. The application is configured to run on all addresses (0.0.0.0) on port 9090.

```
w6 > ChatGPT-Simple > server.py > ...
22 def get_response():
23     message = request.args.get("message")
24     completion = openai.ChatCompletion.create(
25         # You can switch this to `gpt-4` if you have access to that model.
26         model="gpt-3.5-turbo",
27         messages=[{"role": "user", "content": message}],
28     )
29     response = completion["choices"][0]["message"]["content"]
30     return response
31
32
33 if __name__ == "__main__":
34     # app.run(debug=True)
35     app.run(debug=True, host='0.0.0.0', port=9090)
36
```

The terminal output shows the command 'python3 server.py' being executed, resulting in the following messages:

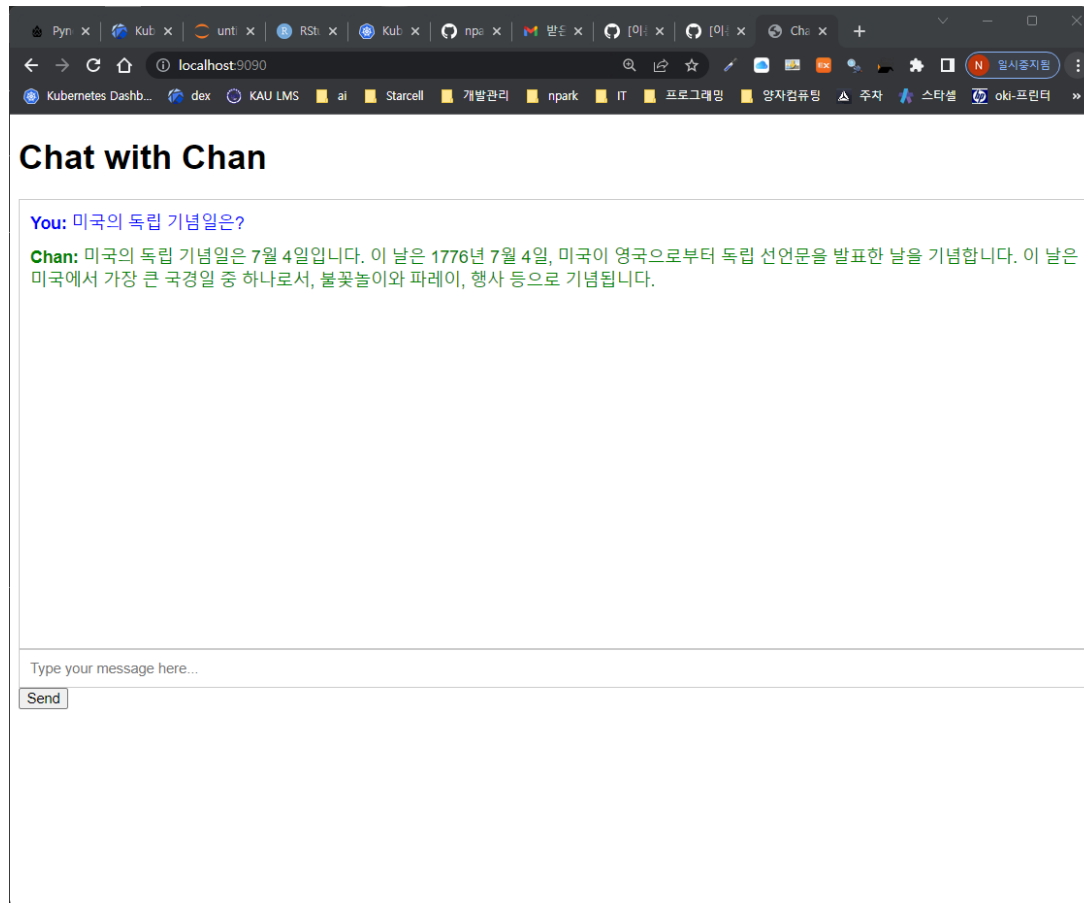
```
root@03ed6ed0960e:~/projects/w6/ChatGPT-Simple# python3 server.py
* Serving Flask app 'server'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:9090
* Running on http://172.17.0.2:9090
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 563-668-325
```

## OpenAI 사용하기

### ChatGPT 예제 실행

- 브라우저 접속

URL : localhost:9090



## 과제

### 1. 다음 시간까지 팀 구성하여 명단 제출(3점)

- 1팀당 5명
- 팀 이름
- 팀장과 팀원 명단
- 팀장만 제출하면 됨
- 팀장이 제출한 팀원 명단에 포함되어 있으면 3점
- 팀장은 +1점 가산점
- 팀 구성이 안되는 학생은 조교에게 문의

### 2. github 저장소의 Issues에 글쓰기(각2점)

- <https://github.com/nparkcourage/2023-kau-0504/issues>
- [이름][학번]-긍정의견  
6주차까지 배운 기술적인 내용 중 유익했던 것 하나 이상 쓰기
- [이름][학번]-부정의견  
6주차까지 배운 기술적인 내용 중 아쉽거나 부족하거나 불필요하거나 개선이 필요한 것 하나 이상 쓰기

### 제출

- 다음 주 월요일(4월17일) 자정(24시)까지(KST;한국 표준시)
- 1번)은 팀장만 제출
- 2번) 자기가 쓴 이슈 글의 URL을 보고서에 써서 LMS에 제출
- LMS 제출은 LMS의 공지 확인하고 공지내용에 맞게 제출  
(조교에게 문의)