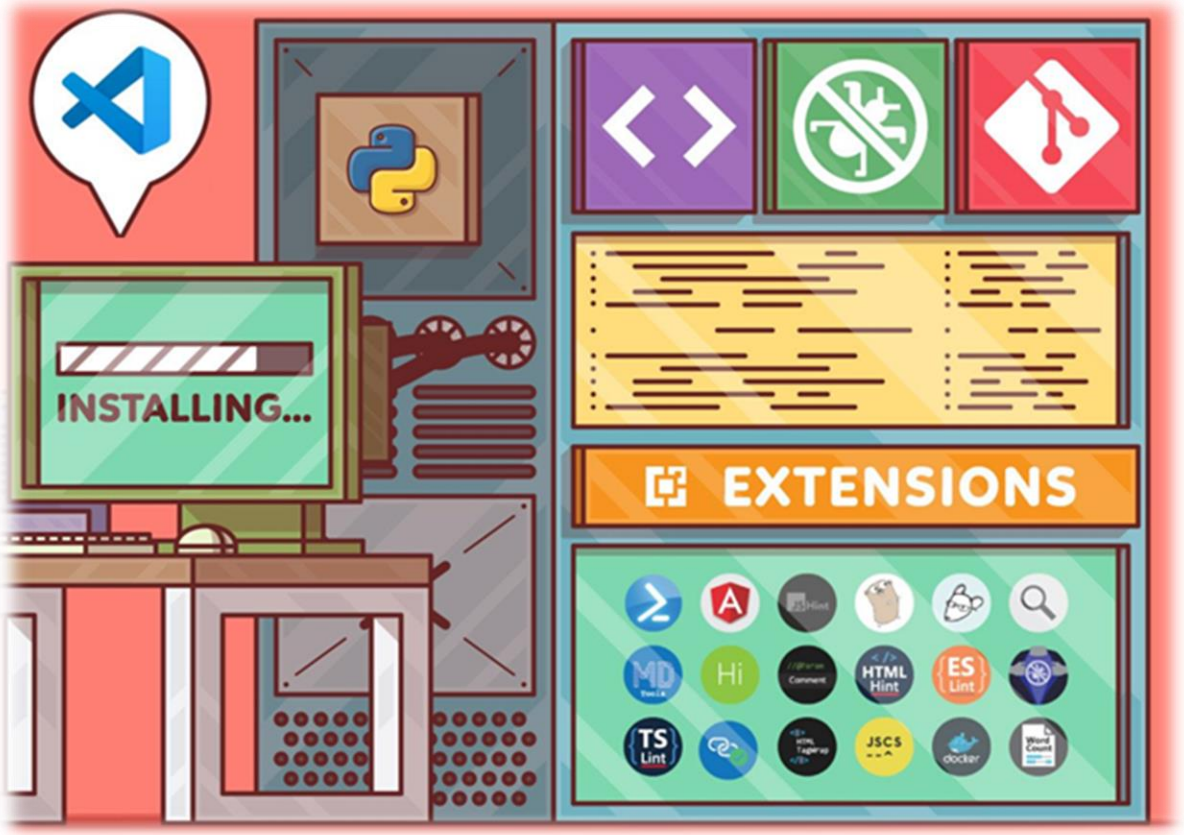
개발환경 구성



박경규

개발환경

PC

Visual Studio Code

Cursor(AI Code Editor)

Sublime Text

Jupyter Notebook

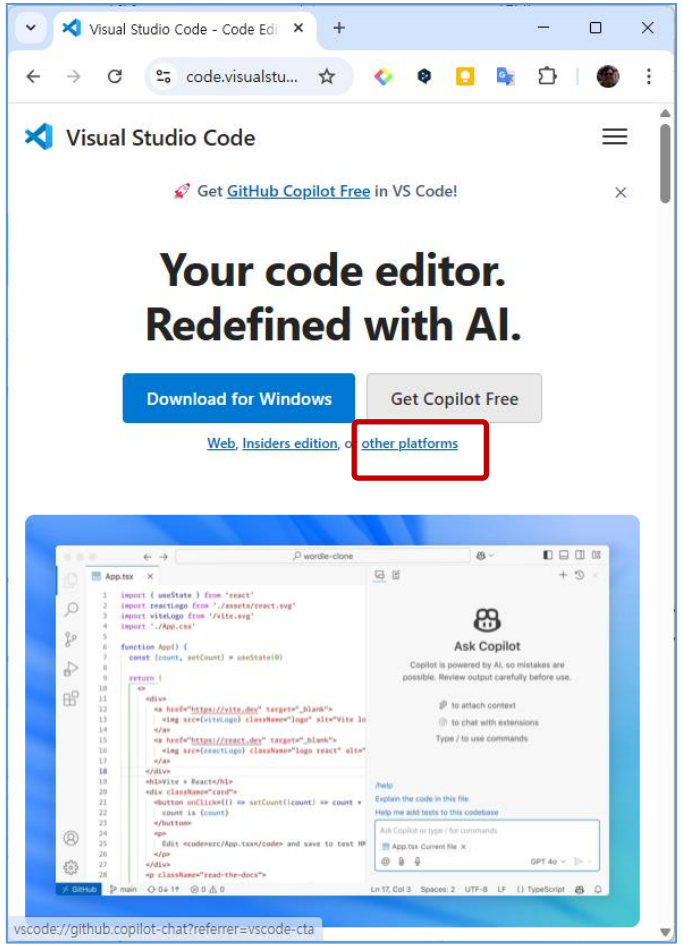
Jupyter Lab

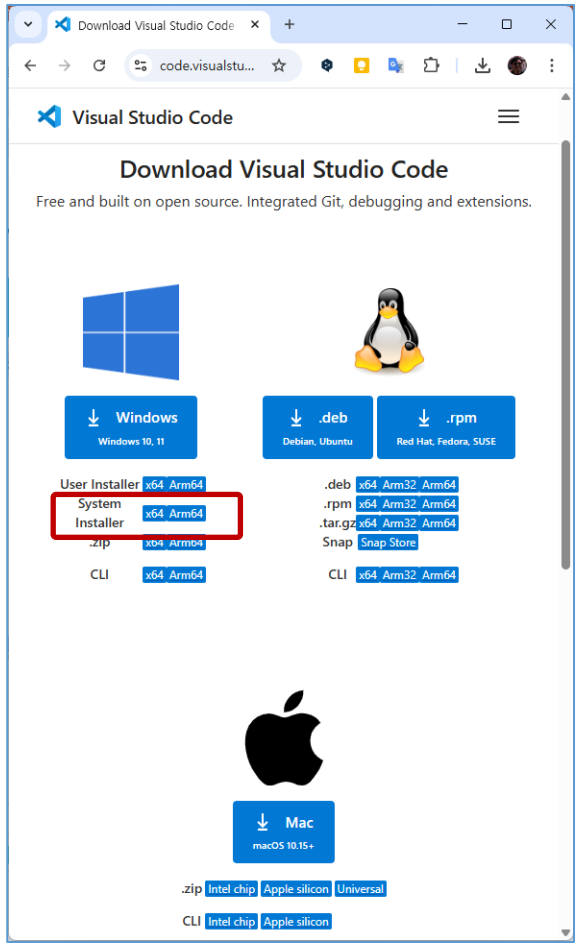
웹

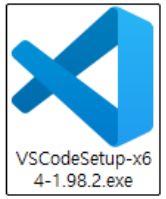
https://colab.research.google.com/

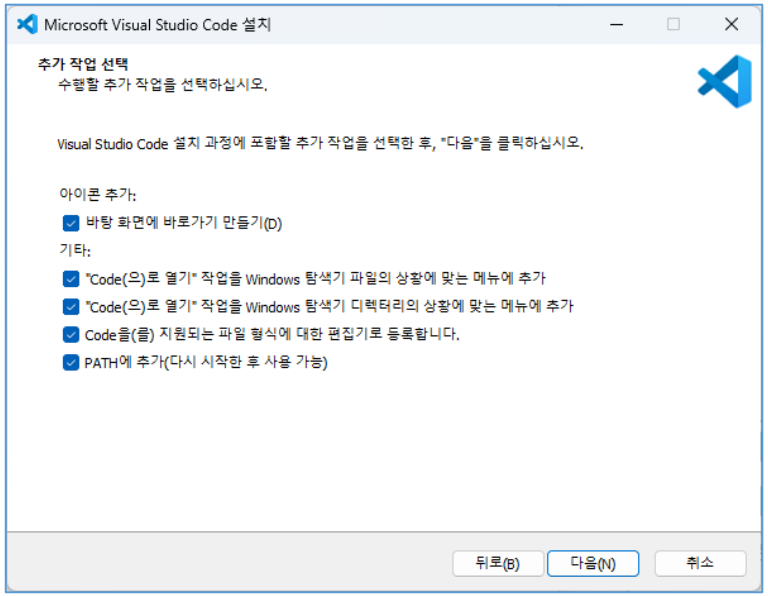
VS Code 설치 – Windows

■ 설치 프로그램 https://code.visualstudio.com/



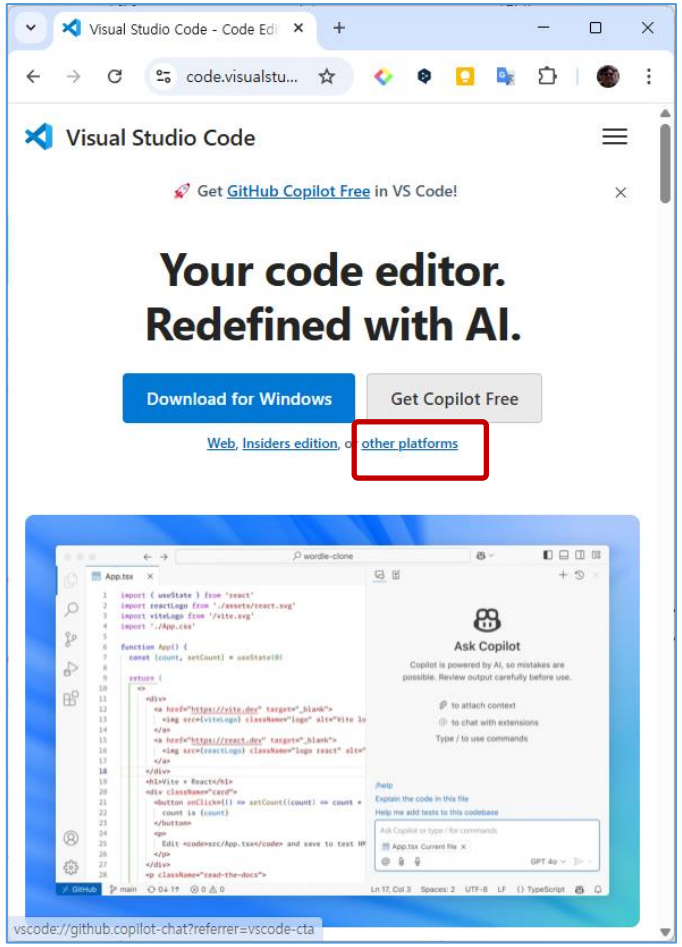


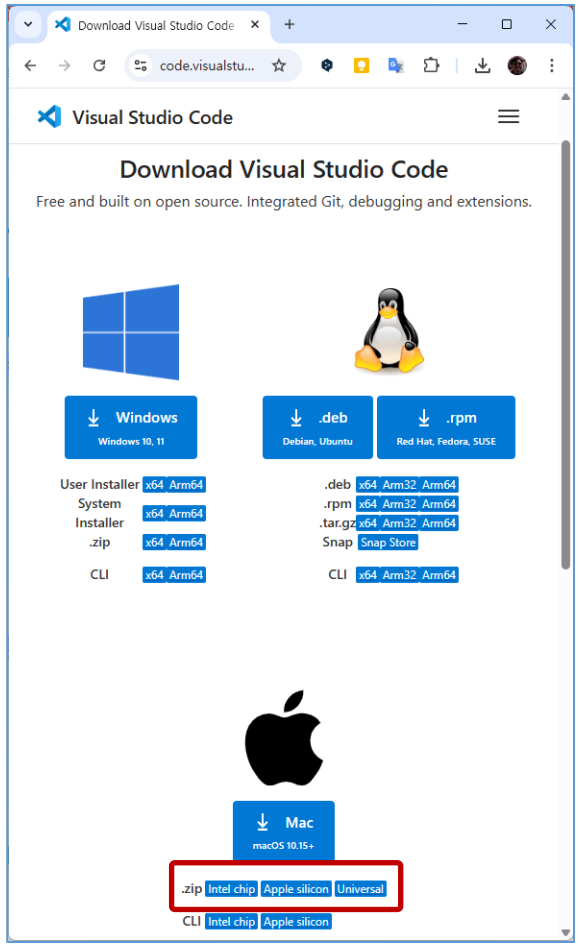




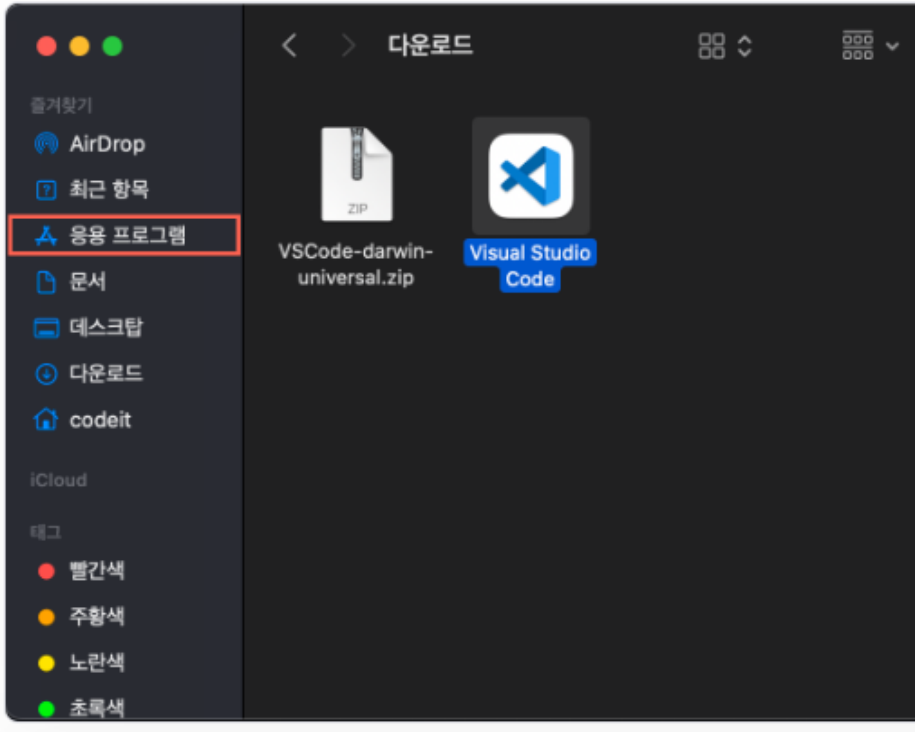
VS Code 설치 – macOS

■ 설치 프로그램 https://code.visualstudio.com/

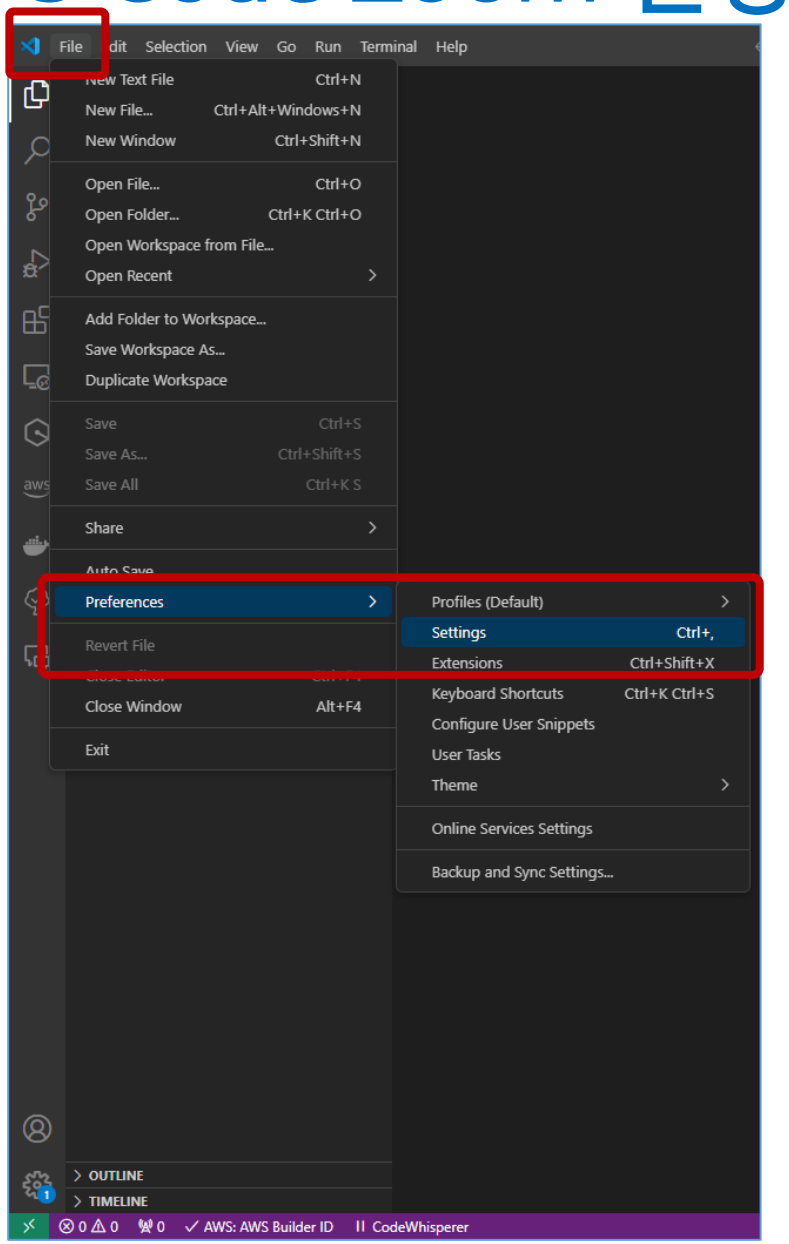


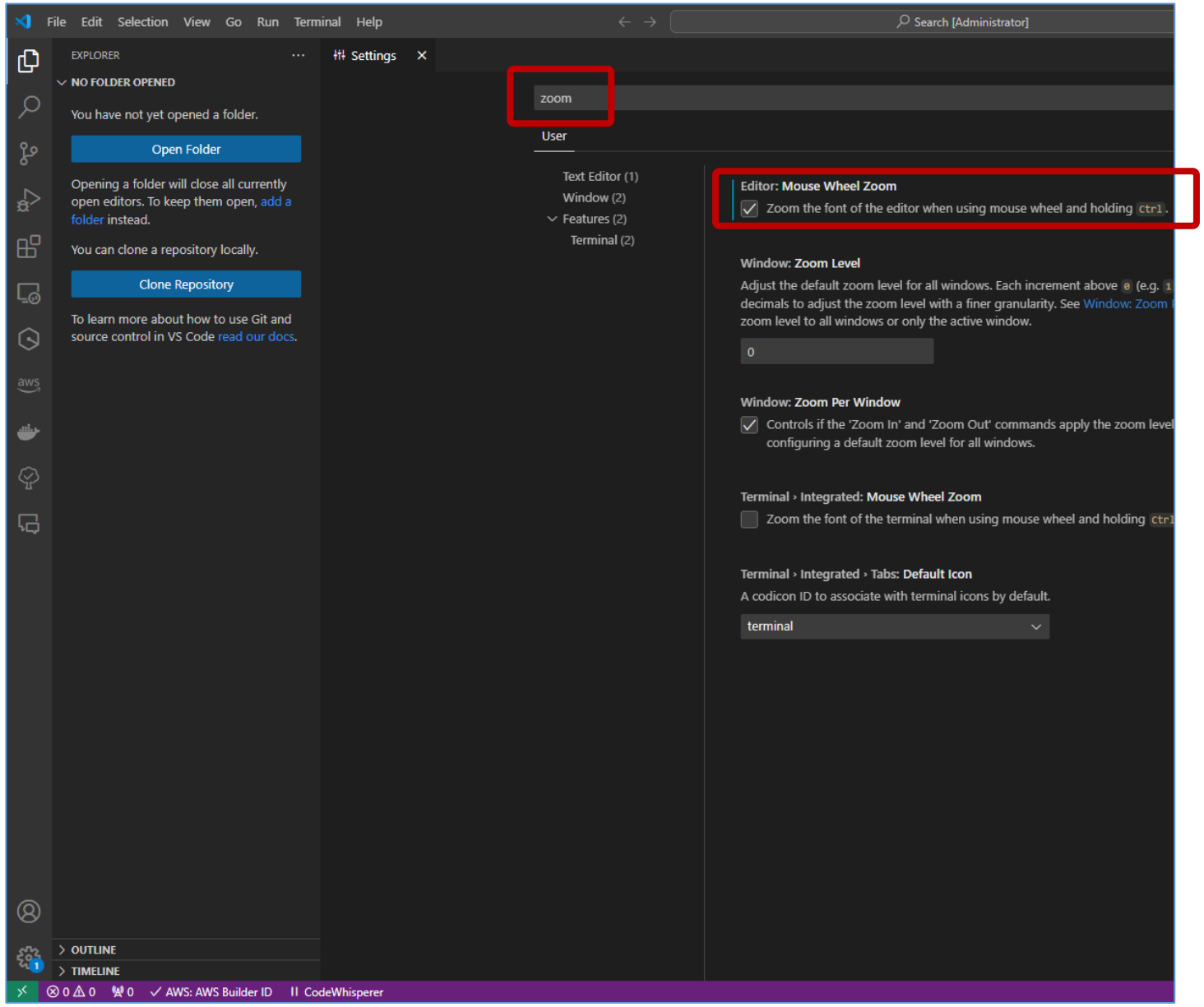


Visual Studio Code를
응용 프로그램(Applications)
폴더로 옮겨 주세요.

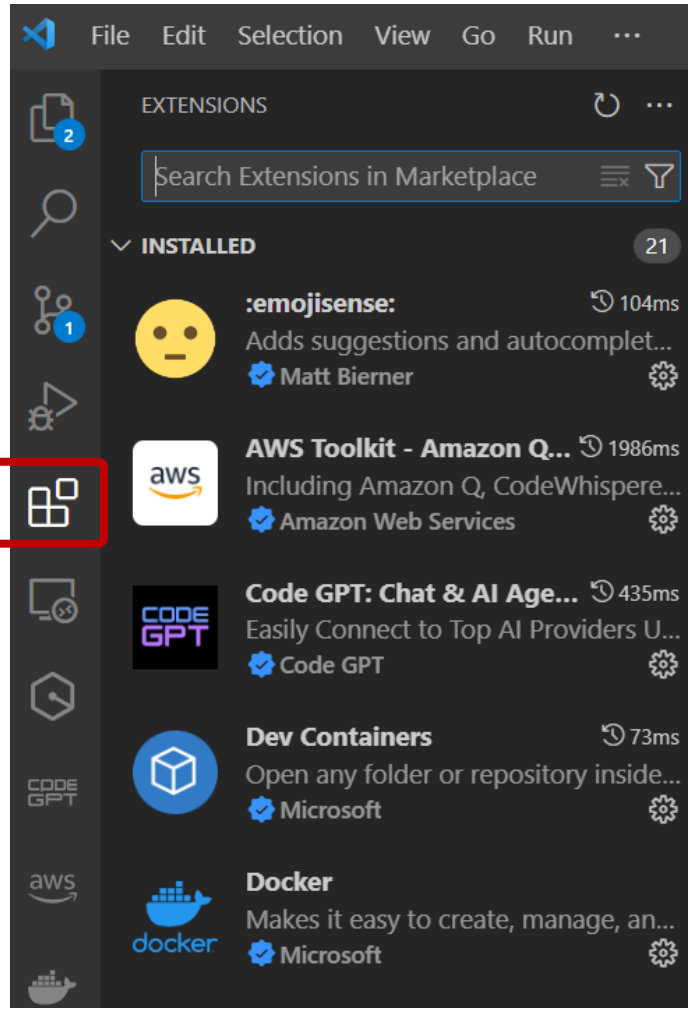


VS Code Zoom 설정





VS Code Extension 설치



- **Python** : 파이썬에 대한 풍부한 지원 제공, IntelliSense(Pylance), Linting, 디버깅, 코드 탐색 등의 기능을 제공
- **Jupyter** : Jupyter 노트북 지원
- **Black Formatter** : Python 파일에 대한 포맷팅 지원 제공
- **vscode-icons** : Visual Studio Code용 아이콘
- **TODO Highlight** : 코드 내에서 TODO, FIXME 및 기타 주석을 강조 표시
- **Todo Tree** : TODO, FIXME와 같은 주석 태그를 빠르게 검색하고 활동 표시줄의 트리 보기에 표시
- **Path Intellisense** : 파일 이름 자동 완성
- **Live Preview** : 웹페이지 미리 보기
- **REST Client** : REST 클라이언트

VS Code 단축키 및 코딩 지원 기능

- **단축키**

터미널 : Ctrl + `

파일 찾기 : Ctrl + P

행 삭제 : Ctrl + X

행 복사 : Ctrl + C

행 붙여넣기 : Ctrl + V

위에 행 복사 추가 : Shift + Alt + Down

아래에 행 복사 추가 : Shift + Alt + Up

행을 아래로 이동 : Alt + Down

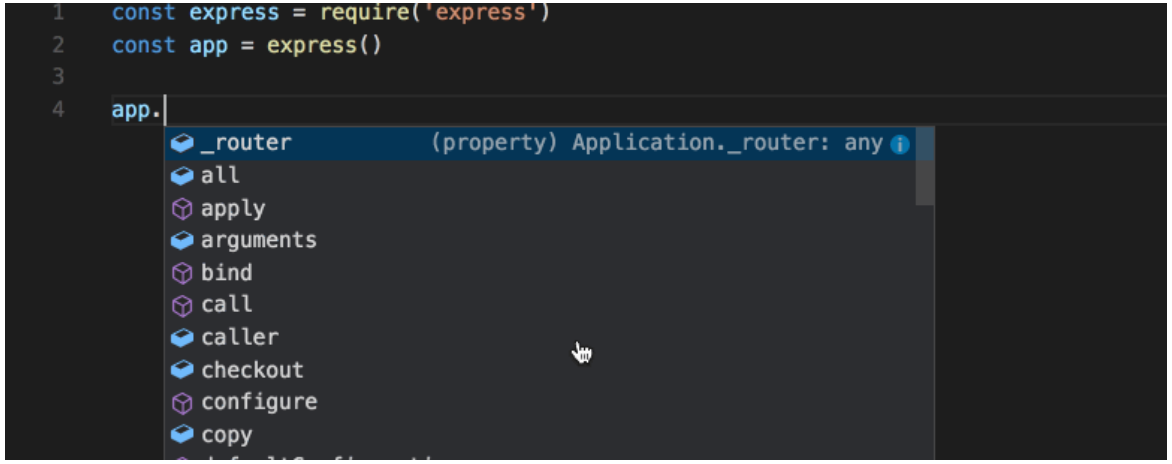
행을 위로 이동 : Alt + Up

화면 크기를 조정 : Ctrl + '+' 또는 Ctrl + '-'

Pytion 인터프리터 선택 : Ctrl + Shift+P (Windows)

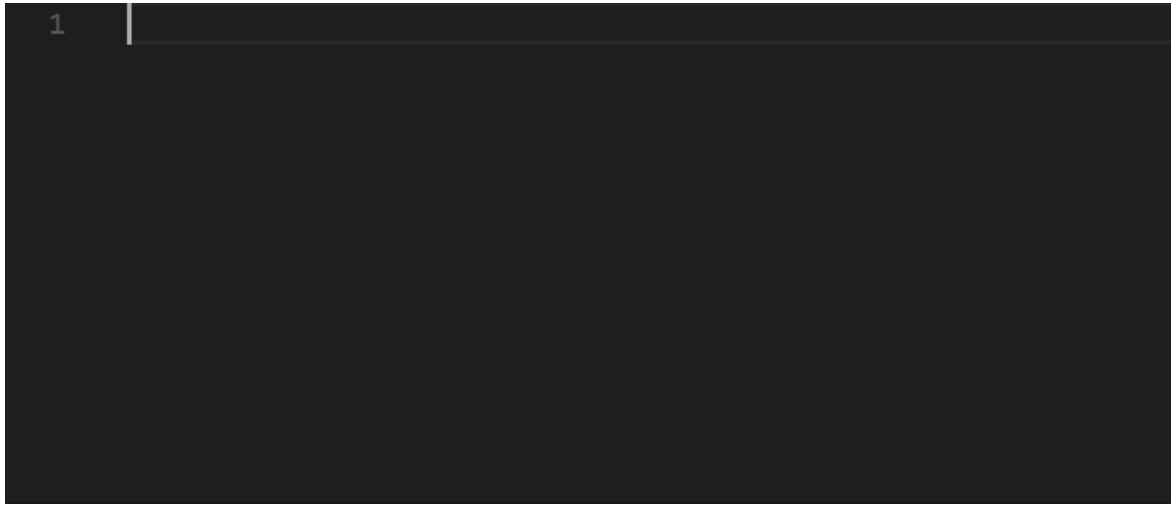
⌘ + ⇧ + P (macOS)

- **인텔리센스(IntelliSense) : 코드 완성**



https://code.visualstudio.com/docs/editor/intellisense

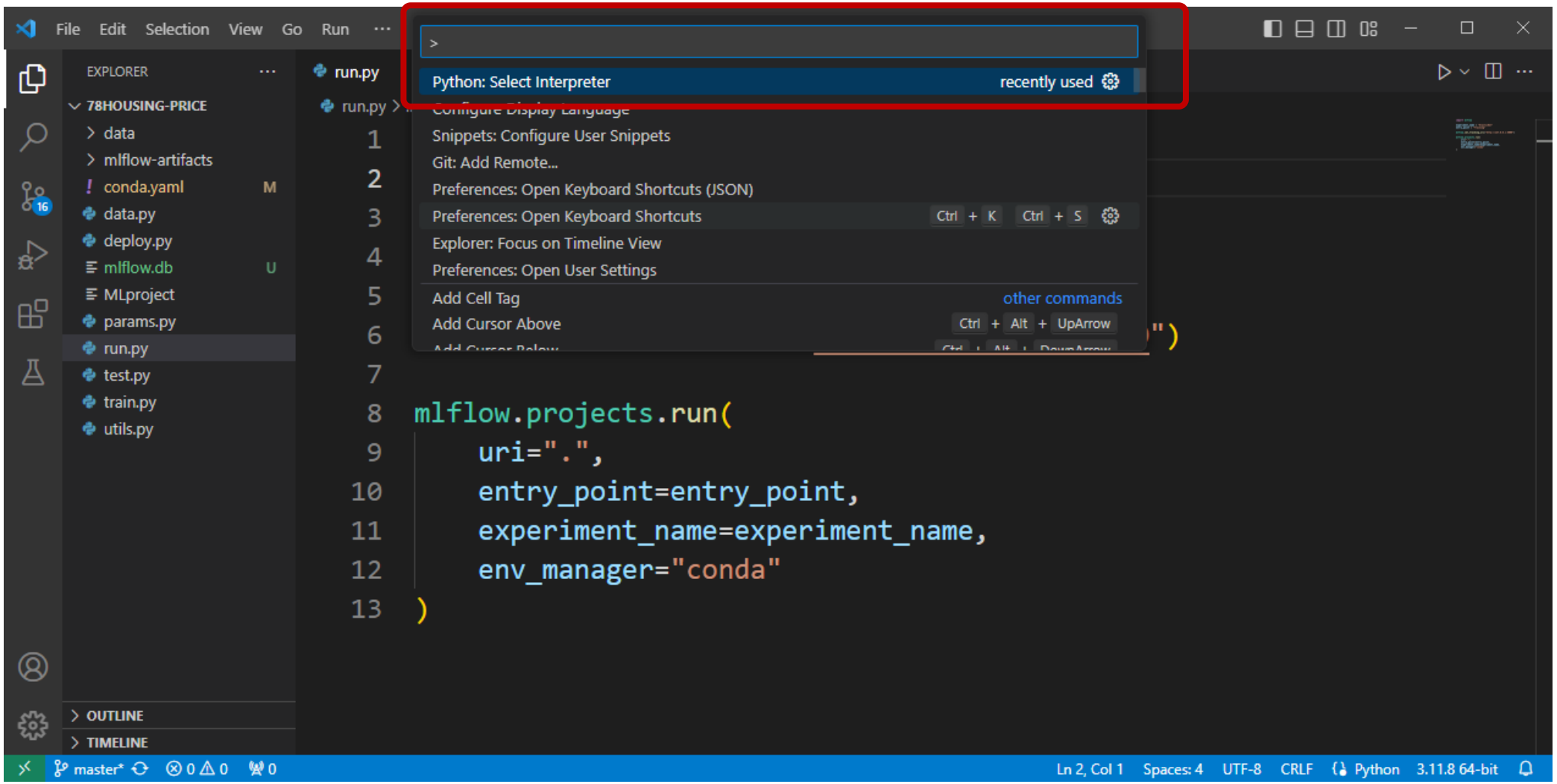
- **코드 스니펫 : 반복되는 코드 패턴을 입력하기 쉽게 해주는 템플릿**



https://code.visualstudio.com/docs/editor/userdefinedsnippets

VS Code : Python 선택

- **Windows : Ctrl + Shift + P**
  Python : Select Interpreter
- **macOS : ⌘ + ⇧ + P**
  Python : Select Interpreter
  Shell Command : Install 'code' command in PATH



NodeJS 설치

NodeJS

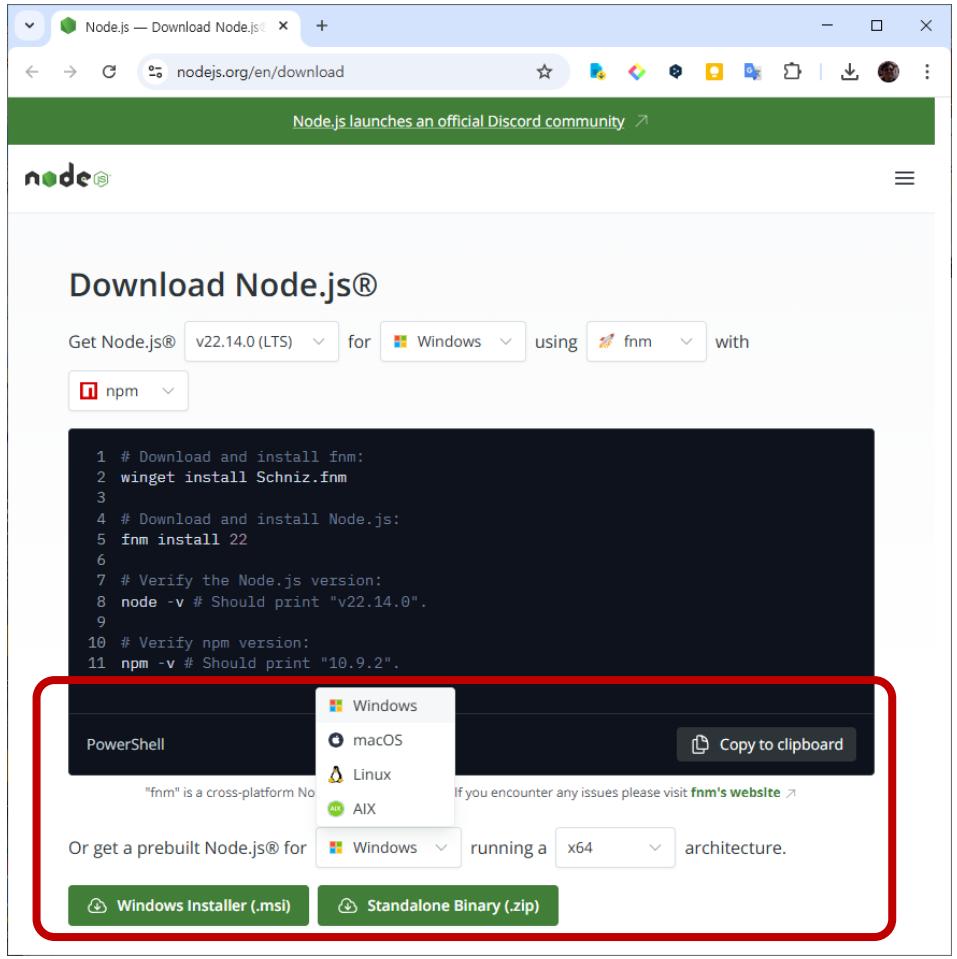
Run JavaScript Everywhere.

Node.js® is a free, open-source, cross-platform JavaScript run-time environment—that lets developers write command line tools and server-side scripts outside of a browser.

Node.js®는 무료, 오픈 소스, 크로스 플랫폼 JavaScript 런타임 환경입니다. 개발자가 브라우저 외부에서 command line 도구와 server-side scripts를 작성할 수 있도록 합니다.

NodeJS 설치

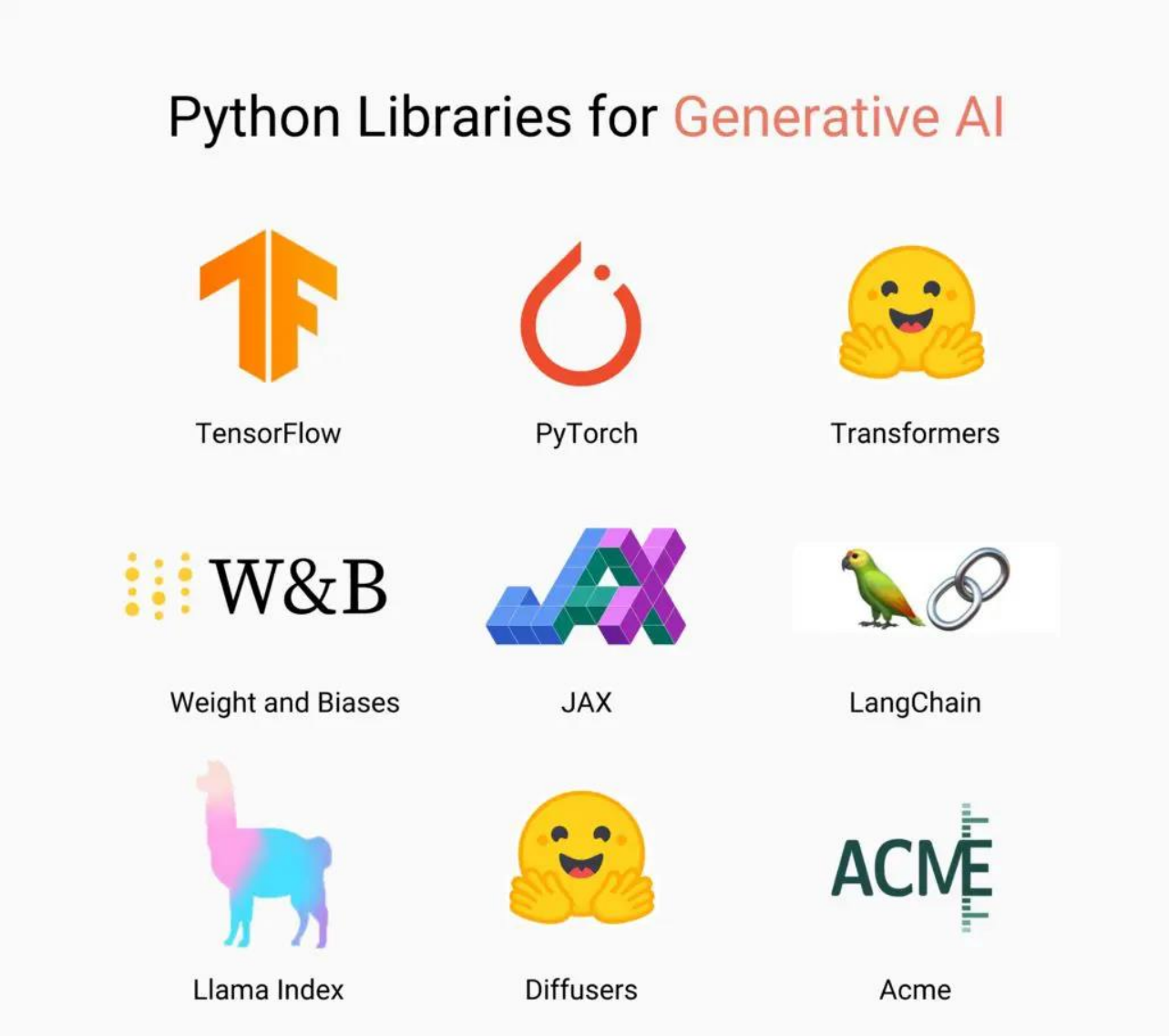
https://nodejs.org/en/download

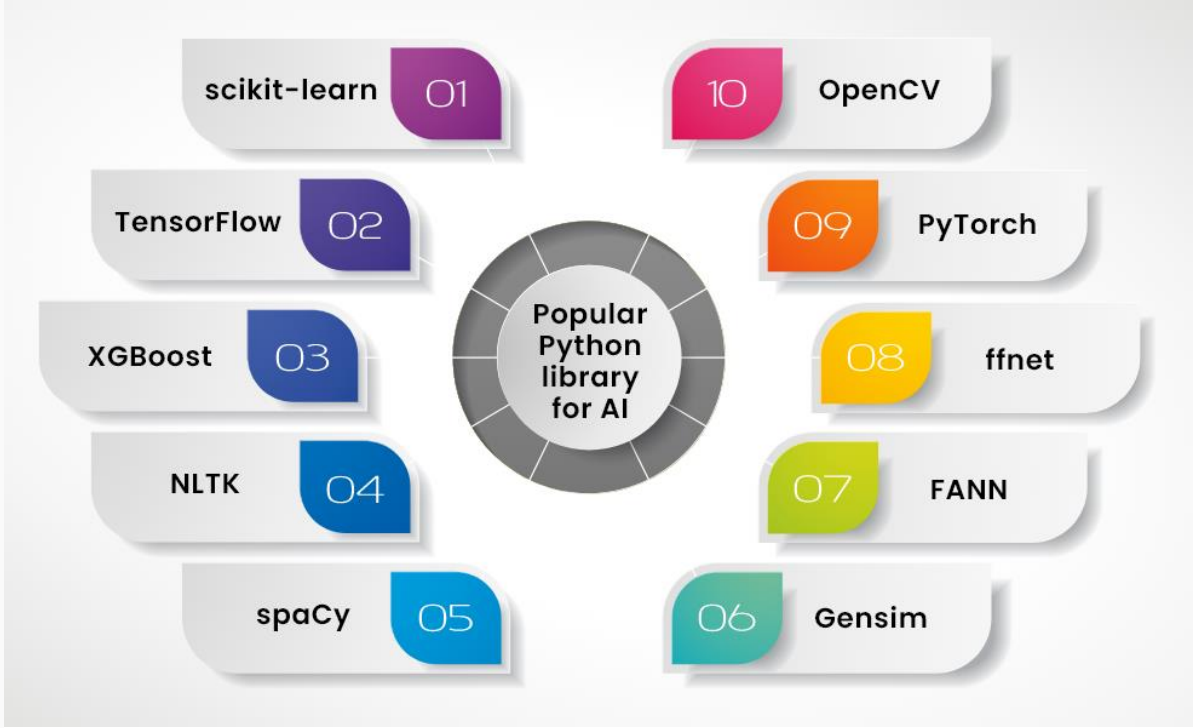


Node 패키지 설치 방법

npm install 패키지명

Python(파이썬)





https://datasciencedojo.com/blog/python-libraries-for-generative-ai/

Python 설치

■ 파이썬 다운로드

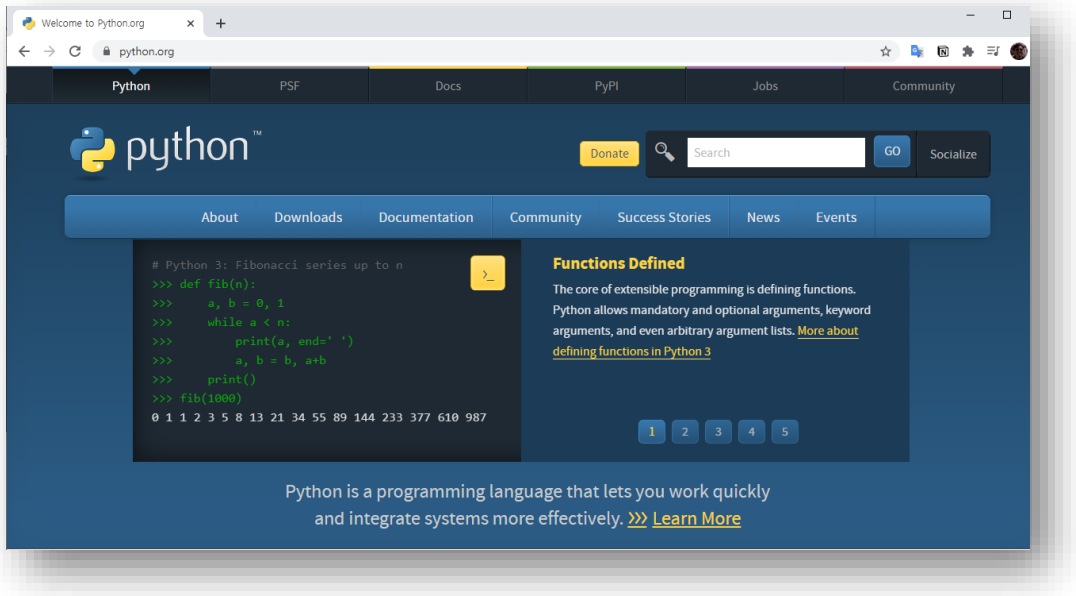
https://www.python.org/downloads/windows/

- Python 3.12.9 - Feb. 4, 2025
  - Download Windows installer (64-bit)
  - Download Windows installer (32-bit)
  - Download Windows installer (ARM64)
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (32-bit)
  - Download Windows embeddable package (ARM64)

https://www.python.org/downloads/macos/

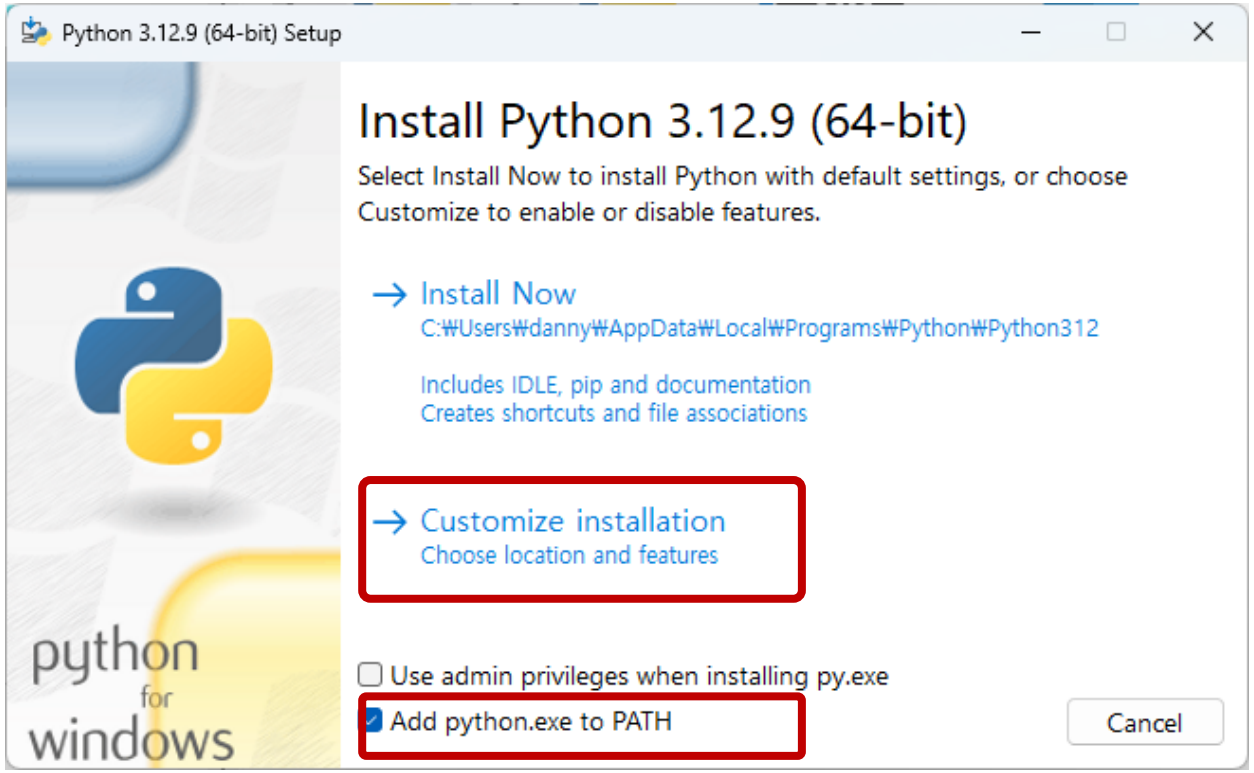
Stable Releases

- Python 3.12.9 - Feb. 4, 2025
  - Download macOS 64-bit universal2 installer



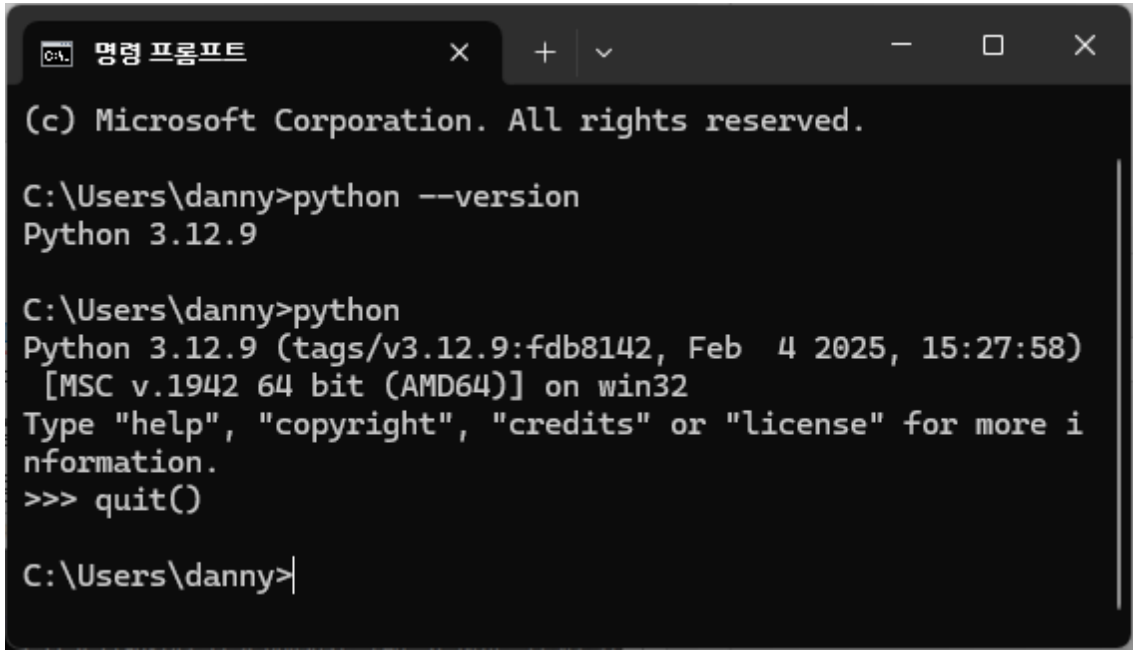
Python 설치

■ 파이썬 설치



■ 파이썬 실행

- 버전 확인 : python --version
- 실행 : python
- 종료 : quit()



Python 가상환경 설치

프로젝트별로 독립된 파이썬 실행 환경을 사용할 수 있는 가상 환경(Virtual Environment) 구성을 권장합니다.

- 가상환경 생성 : `python -m venv venv`

- 가상환경 실행
  - Windows : `venv\Scripts\activate.bat`
  - macOS/Linux : `source venv/bin/activate`

- 파이썬 패키지 설치 : `pip install jupyterlab notebook openai`
  - Jupyter Lab 실행 : `jupyter lab`
  - Jupyter Notebook 실행 : `jupyter notebook`

- 패키지 목록파일 만들기
  ```
  pip freeze > requirements.txt
  ```

- 패키지 목록파일로 패키지 설치 하는 방법
  ```
  pip install -r requirements.txt
  ```

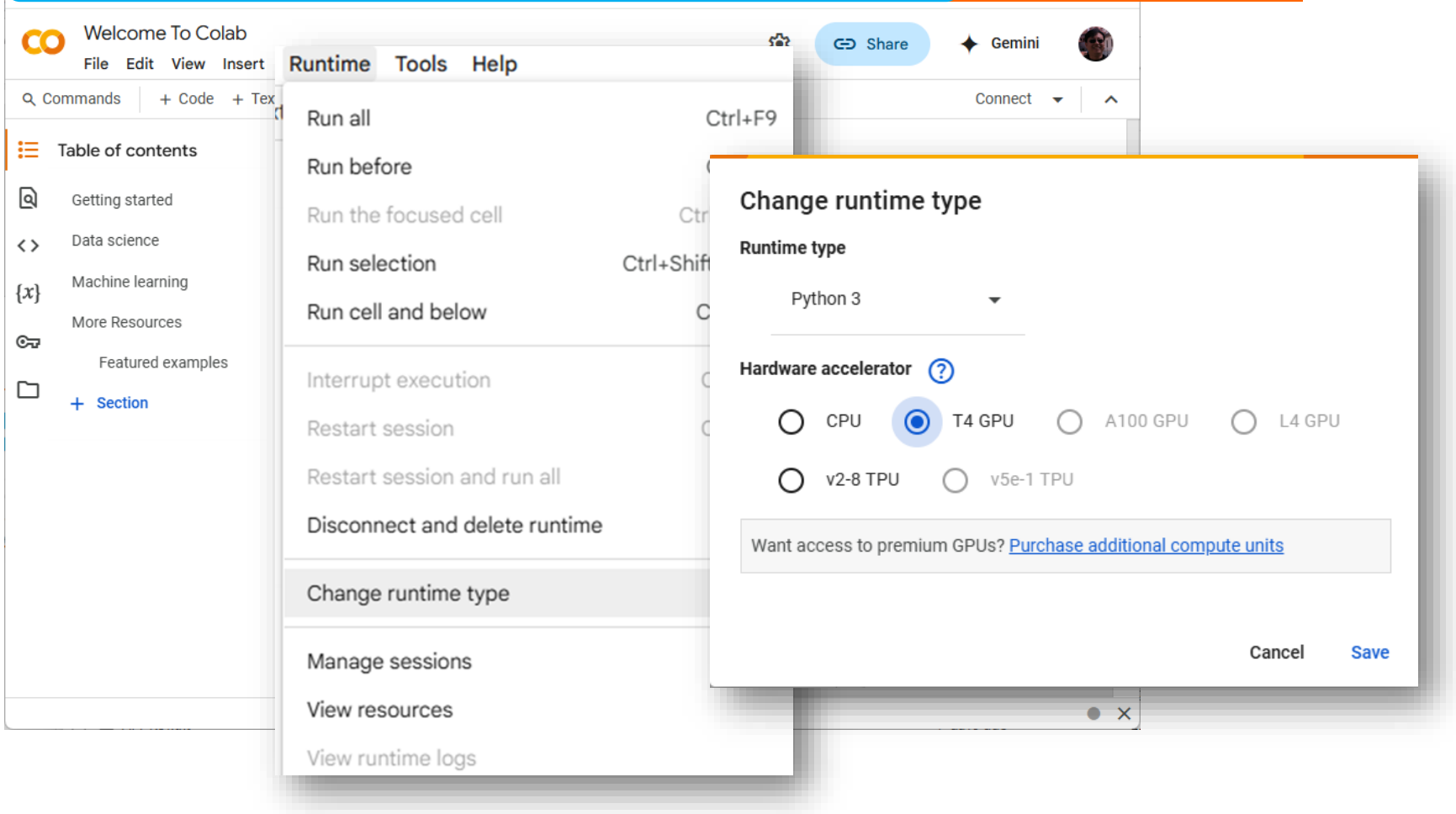
- 파이썬 패키지 삭제 : `pip uninstall 패키지명`

Colab(코랩)

개발툴 설치없이 웹상에서 파이썬 프로그램을 할수 있는 환경으로 딥러닝에 필요한 GPU를 사용할 수 있습니다.

https://colab.research.google.com

구글 계정 필요

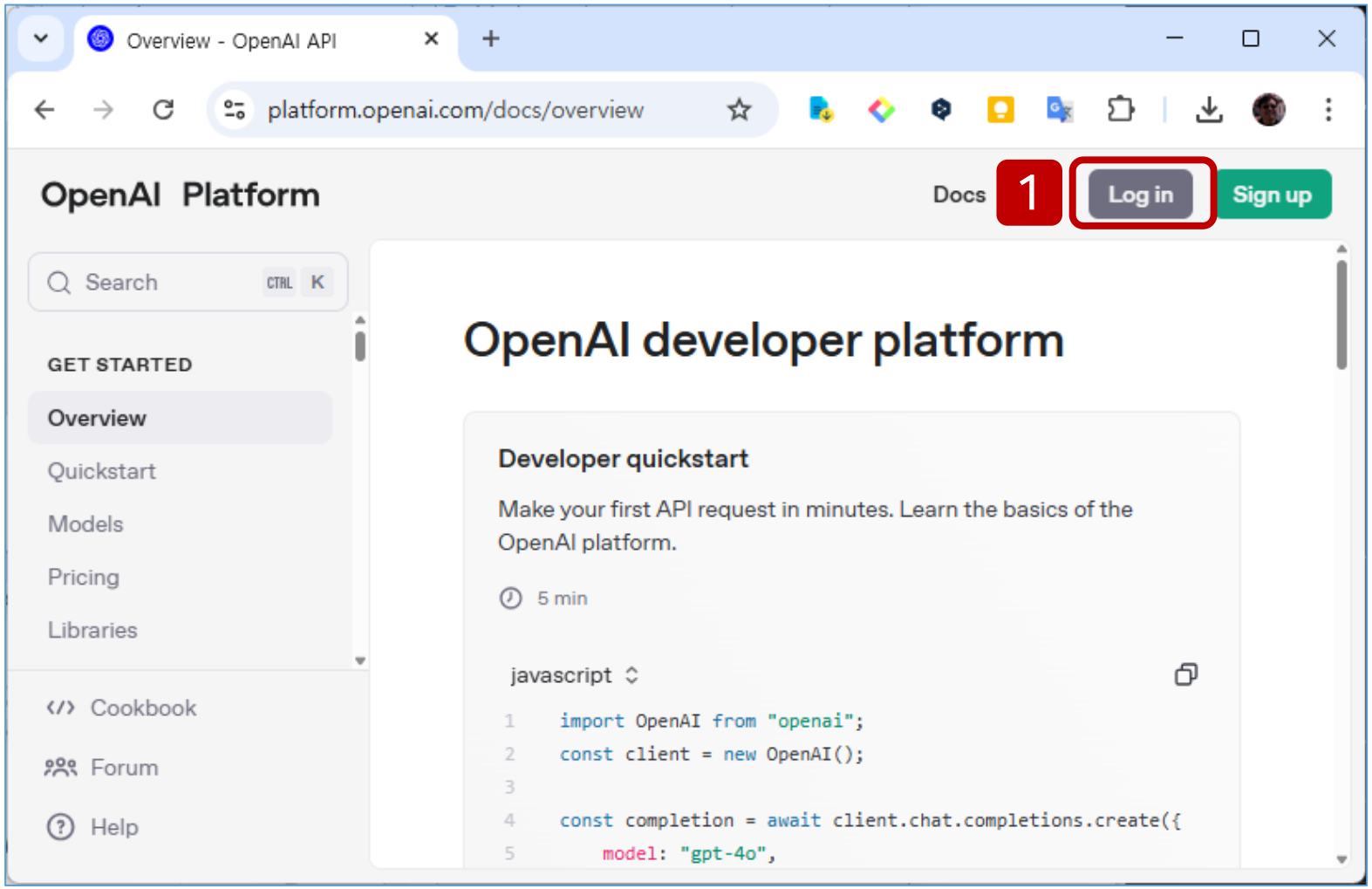


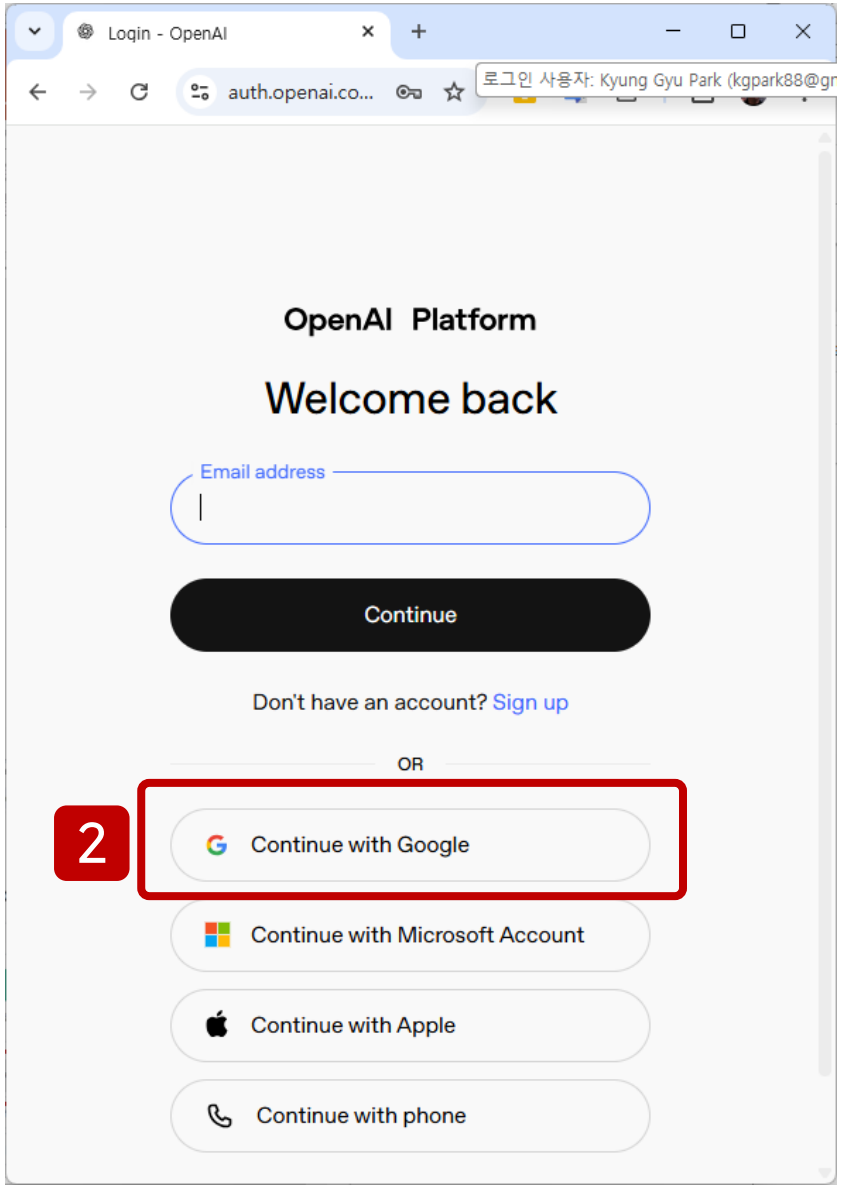
고성능GPU(Graphics Processing Unit)

OpenAI API 사용

https://platform.openai.com/

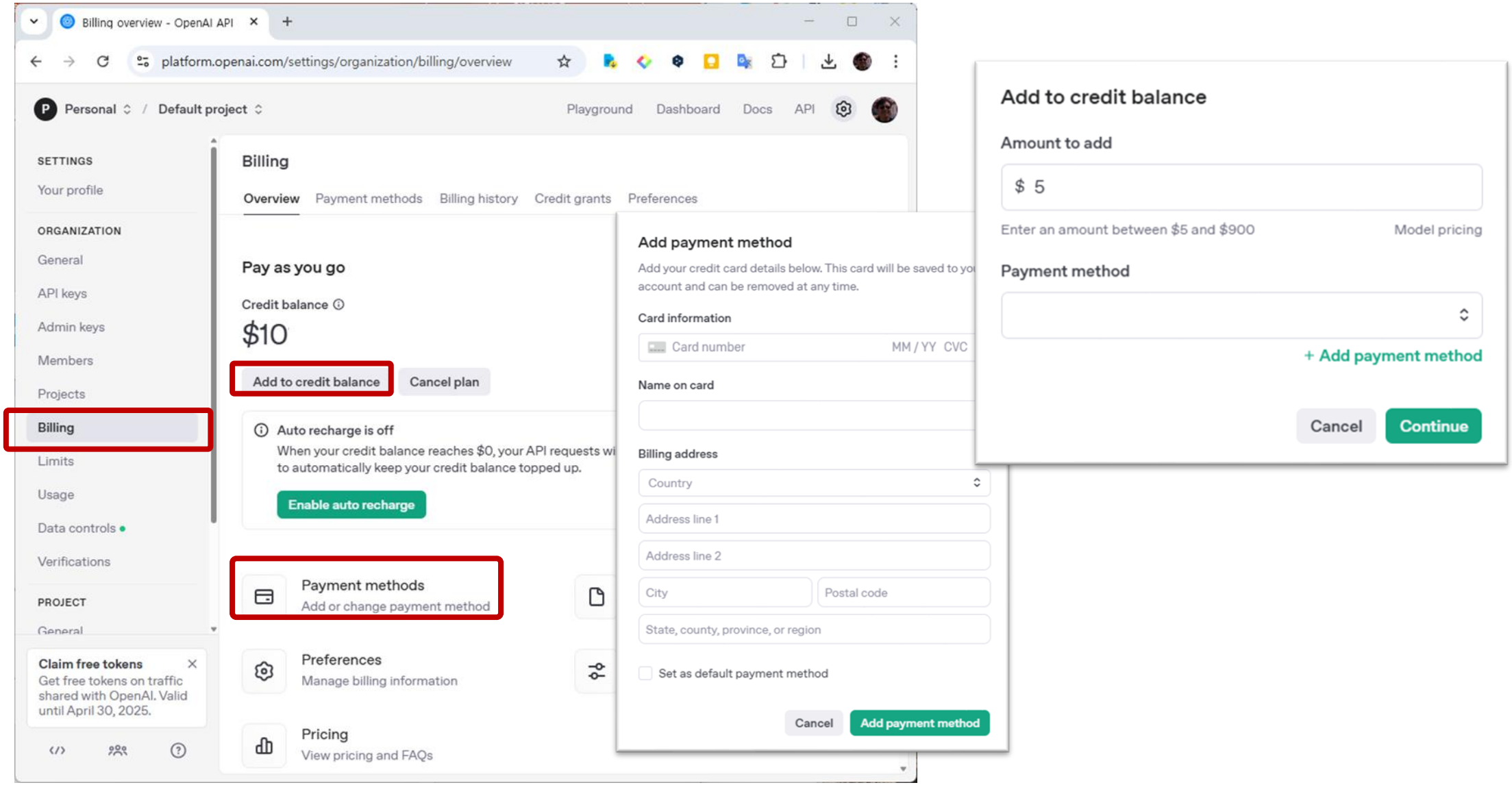
SIGN UP



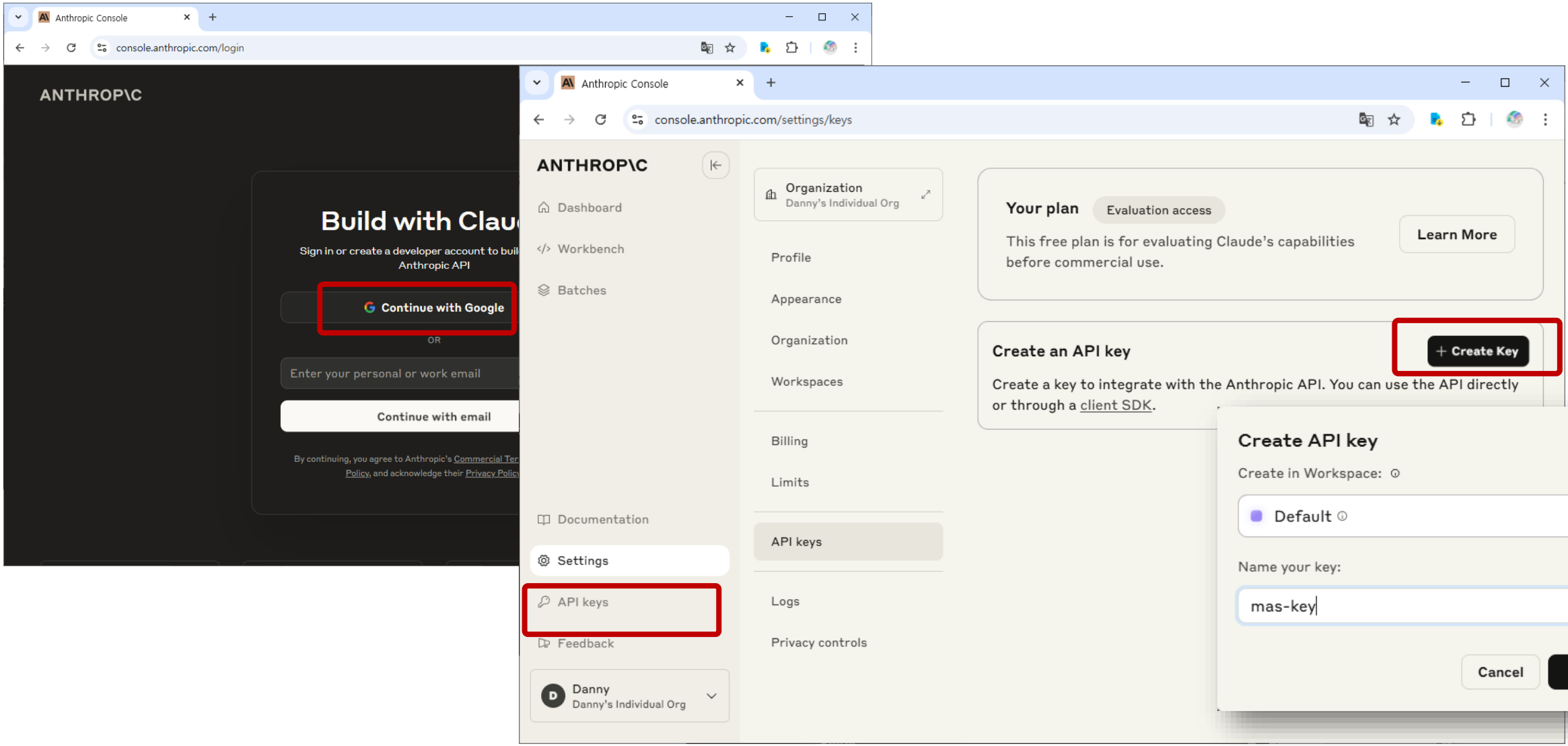


OpenAI API 구매

https://platform.openai.com/settings/organization/billing/overview



Anthropic API 사용 https://console.anthropic.com/



Python 기초

■ 변수 할당(Variable Assignment)

```
x = 2
y = 3
z = x + y

x = 'hello'
x = "hello"
X
[Out] 'hello'
```

Single Quotation
작은 따옴표

Double
Quotation
쌍 따옴표

■ 출력

```
print(x)
[Out] 'hello'
```

■ 리스트(List)

```
[1, 2, 3]
['a', 'b', 'c']
my_list = [1, 2, 'apple', True]
my_list.append(100)
my_list[0]
my_list[:-1]
my_list[-1]
```

Bracket
대괄호

■ 딕셔너리(Dictionary)

```
d = {'key1':'item1','key2':'item2'}
d['key1']
[Out] 'item1'
```

Brace
중괄호

실습

colab

python_essence.ipynb

cursor_practice.py

Thank you