Week 2.
파이썬 환경 구축과 Hello World

설치·설정·실행까지 한 번에
(With AI 활용)

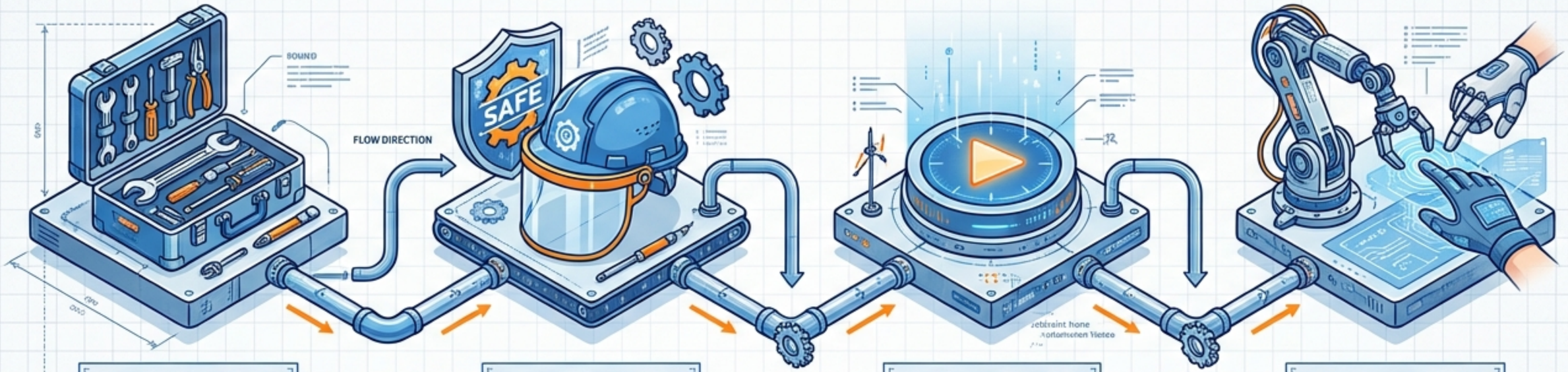
Pretendard

저장소: https://github.com/MNU-AI-Programming/AI-programming

AI활용프로그래밍

NotebookLM

오늘의 학습 목표 (The Goal)



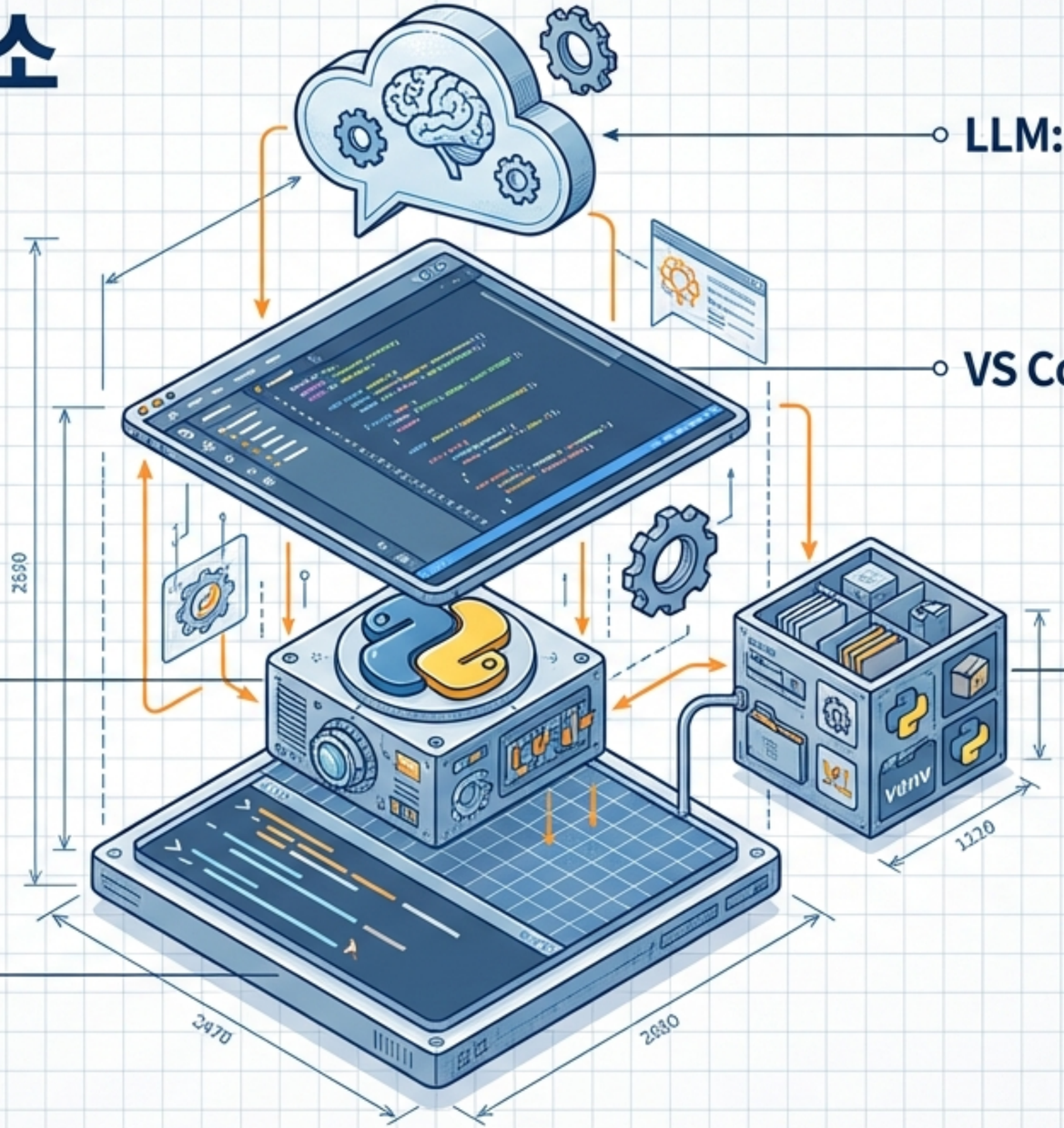
1. 개발환경 구성 → 2. 안전한 작업장 → 3. 첫 프로토타입 → 4. AI 파트너

FLOW DIRECTION · MILESTONE LINKAGE · SYSTEM INTEGRATION

"설치 → 실행 → 확인 → 정리까지
스스로 할 수 있게 만들기"

NotebookLM

개발환경 구성 요소 (The Toolkit)



LLM: 24/7 멘토

VS Code: 작업대 (Workbench)

Python: 엔진

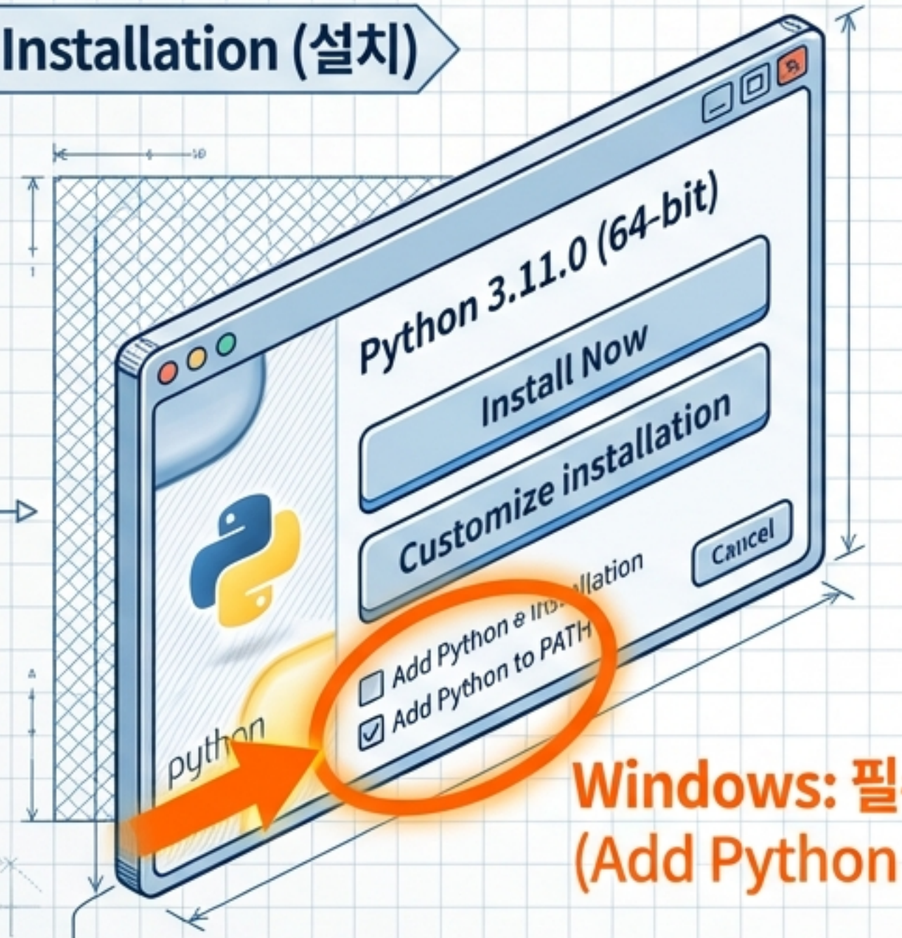
Pip/Venv: 창고 및 안전장치

Terminal: 제어판

NotebookLM

Python 설치와 기초 공사

Installation (설치)



Windows: 필수 체크!
(Add Python to PATH)

macOS/Linux: 공식 설치 또는 Homebrew 사용

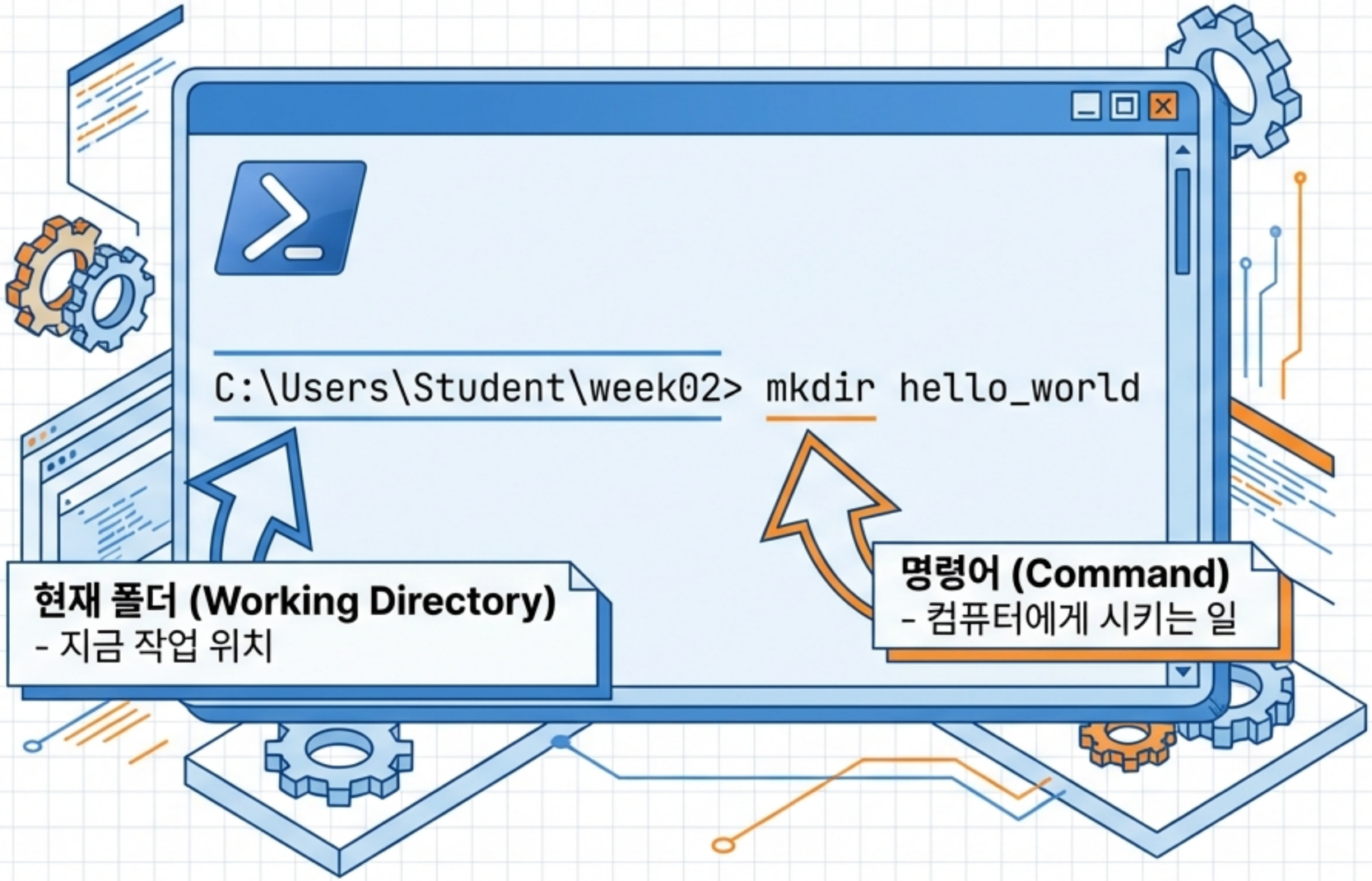
Verification (설치 확인)

설치 확인 (The Handshake)

```
$ python --version
Python 3.11.0
$
```

NotebookLM

터미널(Terminal): 컴퓨터와 대화하는 창



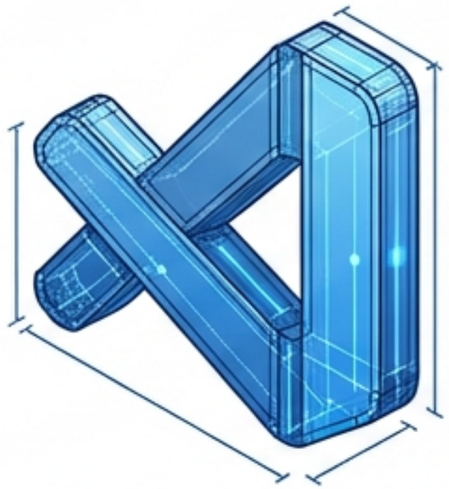
기초 명령어

- cd [folder]: 폴더 이동
- ls / dir: 목록 보기
- mkdir [name]: 새 폴더 만들기

NotebookLM

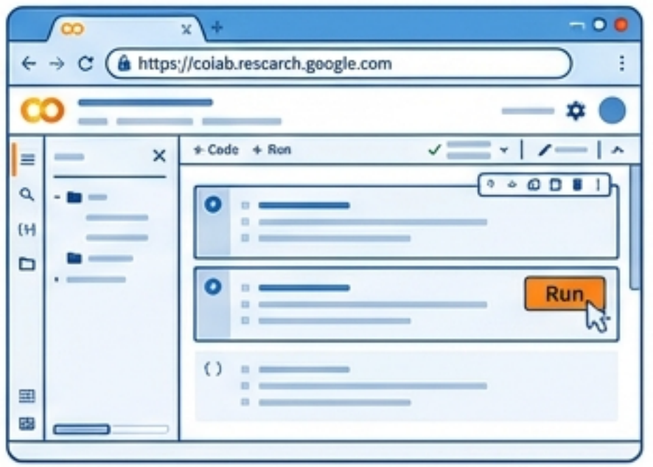
코드 에디터 선택하기

Option 1: VS Code



- 현업 표준, 강력한 기능
- Python Extension 설치 필수
- 습관: 저장(Save) → 실행(Run)

Option 2: Google Colab

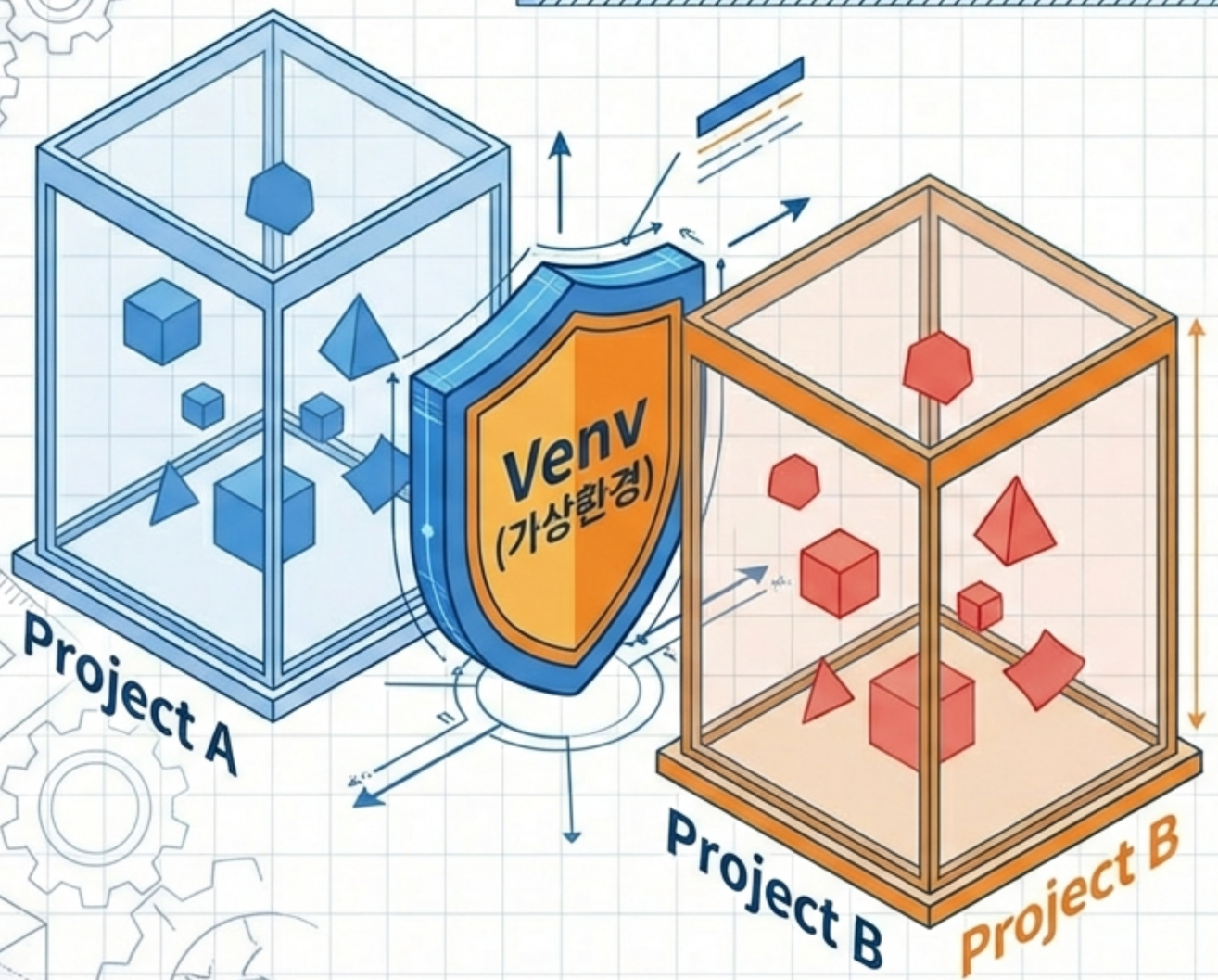


- 설치 없이 웹에서 바로 실행
- 구글 계정 필요
- .ipynb 자동 생성

Tip: week02/ 폴더를 생성하여 파일을 정리하세요.

NotebookLM

가상환경(venv) 왜 복제



가상환경(venv)은 왜 필요한가?

- **의존성 격리:** 프로젝트마다 필요한 도구가 다름
- **충돌 방지:** A와 B가 섞이지 않도록 차단
- **협업 용이:** `requirements.txt`로 환경 복제

NotebookLM

venv 생성 및 활성화

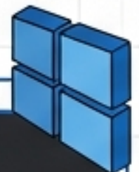
Section 1: 생성 (Create)

```
python -m venv .venv
```

Section 2: 활성화 (Activate)

Windows

```
.venv\Scripts\activate
```



macOS/Linux

```
source .venv/bin/activate
```

Visual Cue Section

Before:

```
C:\Project>
```

After (Highlighted in Orange):

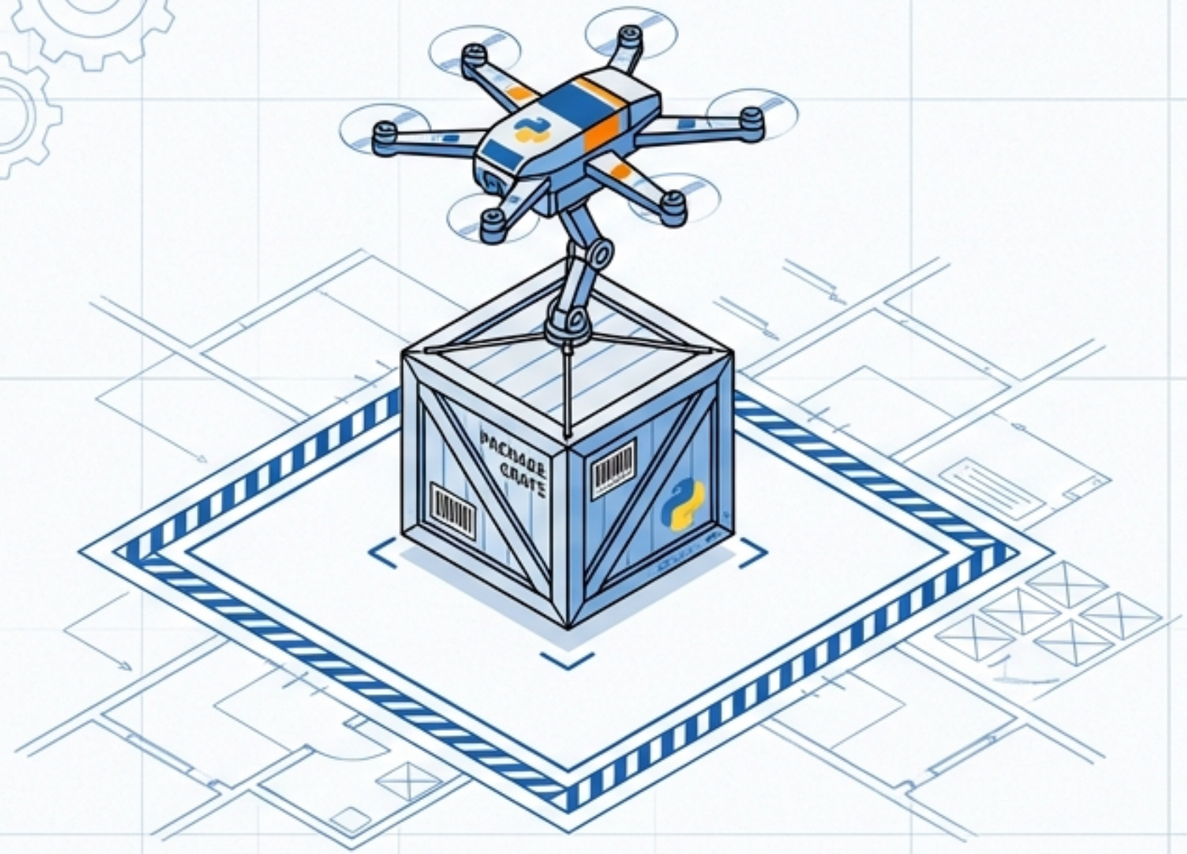
```
(.venv) C:\Project>
```

성공 신호: 프롬프트 앞 (.venv)

끄기: deactivate

NotebookLM

pip: 파이썬 패키지 관리자



LABEL 1
(IMPORT)

설치 (Install) Pretendard

```
pip install [package]
>_
```

JetBrains Mono

LABEL 2
(INVENTORY CHECK)

목록 확인 (List) Pretendard

```
pip list
>_
```

JetBrains Mono

LABEL 3
(MANIFEST)

목록 저장 (Freeze) Pretendard

```
pip freeze > requirements.txt
>_
```

JetBrains Mono

Blueprint Paper

Noto Sans KR

⚠ WARNING TAPE 주의: 반드시 가상환경(.venv)이 켜진 상태에서 실행하세요!

Blueprint

NotebookLM

Hello World: 첫 스크립트 실행

Drafting Paper
White

Insight: 가장 작은 성공 경험을 만든 뒤 확장한다.

```
python hello.py
```

1. Terminal: `python hello.py` (Main Method)

2. VS Code: 'Run' Play Button (UI)

3. REPL: Interactive Mode

NotebookLM

입출력 업그레이드 (Input & Output)

Before & After

```
print("Hello")
```

User Input - 데이터 받기

```
name = input("이름을 입력하세요: ")
print(f"Hello, {name}!")
```

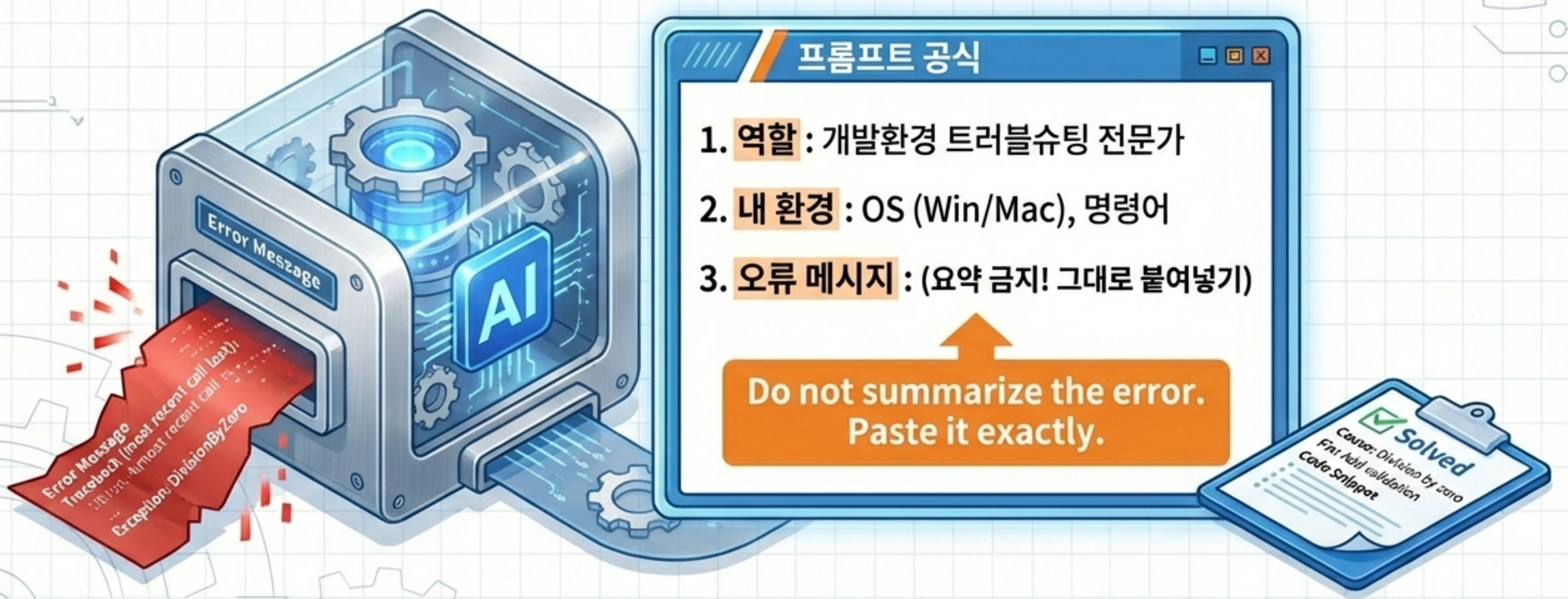
f-string - 변수 삽입

```
print("A", "B", sep="-", end="")
```

Formatting Options (sep, end)

NotebookLM

LLM 활용 1: 에러가 발생했을 때



NotebookLM

LLM 활용 2: 코드 설명과 주석

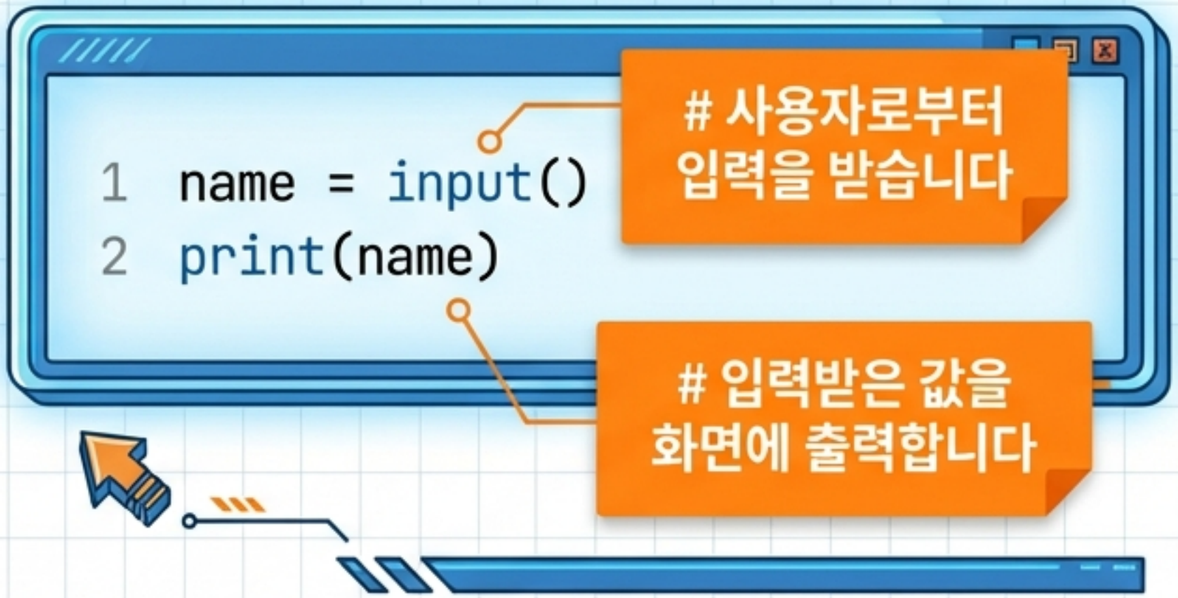
The Request

이 코드를 줄별로 설명하고,
초보자용 주석을 달아줘.

Code Snippet

```
name = input()
print(name)
```

The AI Output



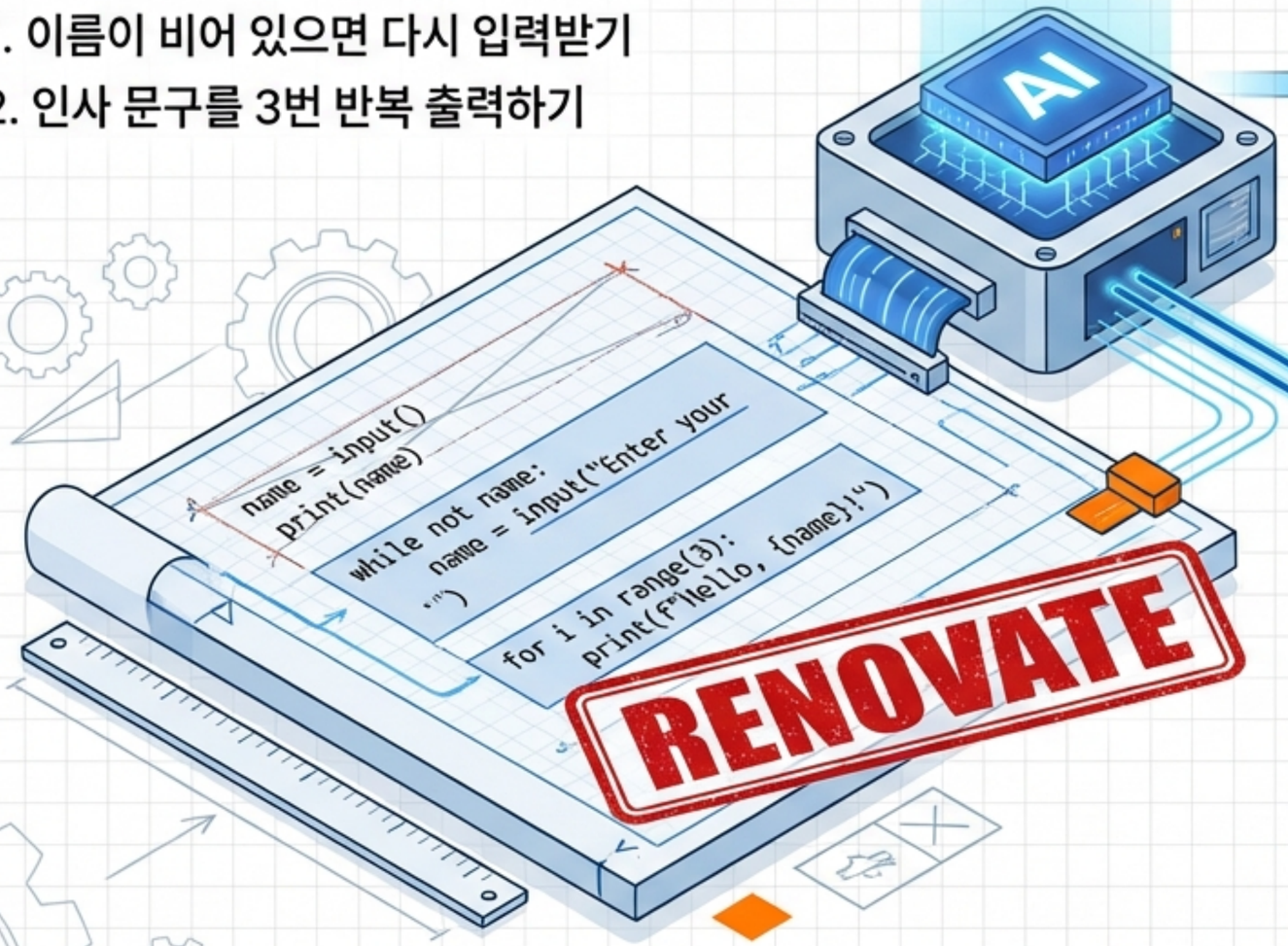
Tip: '출력 결과(로직)는 바꾸지 말 것'이라고 제약을 거세요.

NotebookLM

LLM 활용 3: 기능 추가 요청

The Request

1. 이름이 비어 있으면 다시 입력받기
2. 인사 문구를 3번 반복 출력하기



위 기능을 추가해줘.
단, 반드시 실행 예시를 보여줘.

The AI Output

```
name = input()
print(name)
```

```
while not name:
    name = input("Enter your name: ")
```

```
for i in range(3):
    print(f"Hello, {name}!")
```

Tip: '실행 예시'를 요청하면, 수정된 코드의 작동 방식을 빠르게 검증할 수 있습니다.

NotebookLM

미션 완료 및 다음 단계

Next Week: 변수(Variables), 자료형(Data Types), 연산자

"그냥 되게 하기(*Getting it to work*)가 아니라 왜 되는지 설명하기(*Understanding why*)!"

NotebookLM