

AI 활용 프로그래밍 Week 1

프로그래밍 언어 구조와 LLM 협업의 시작

The diagram illustrates a workflow. On the left, a code editor window titled 'CODE.py' contains Python code. The code defines a function 'greet' that takes a name as input and returns a greeting message using an f-string. It also includes a print statement calling the function with the argument "Alice". In the center, a vertical stack of logic components represents the execution flow. From top to bottom, it shows: a red circle (input), a hatched rectangle (function call), a blue circle (parameter), another hatched rectangle (return value), a blue circle (output), and finally a white circle at the bottom. On the right, a 'CHAT WITH AI' interface shows a conversation between a user and an AI. The user asks to explain the function, and the AI responds that it's a Python function that greets a user. The user then asks how to optimize it, and the AI suggests using type hinting. At the bottom right of the AI interface is a button labeled 'Ask AI about your code...' with a play icon.

```
CODE.py
01
02
03 def greet(name):
04     """This function greets the user."""
05     return f"Hello, {name}!"
06
07
08 print(greet("Alice"))
09
10
```

CHAT WITH AI

User: Explain this function.

AI: This Python function takes a name as input and returns a greeting message using an f-string.

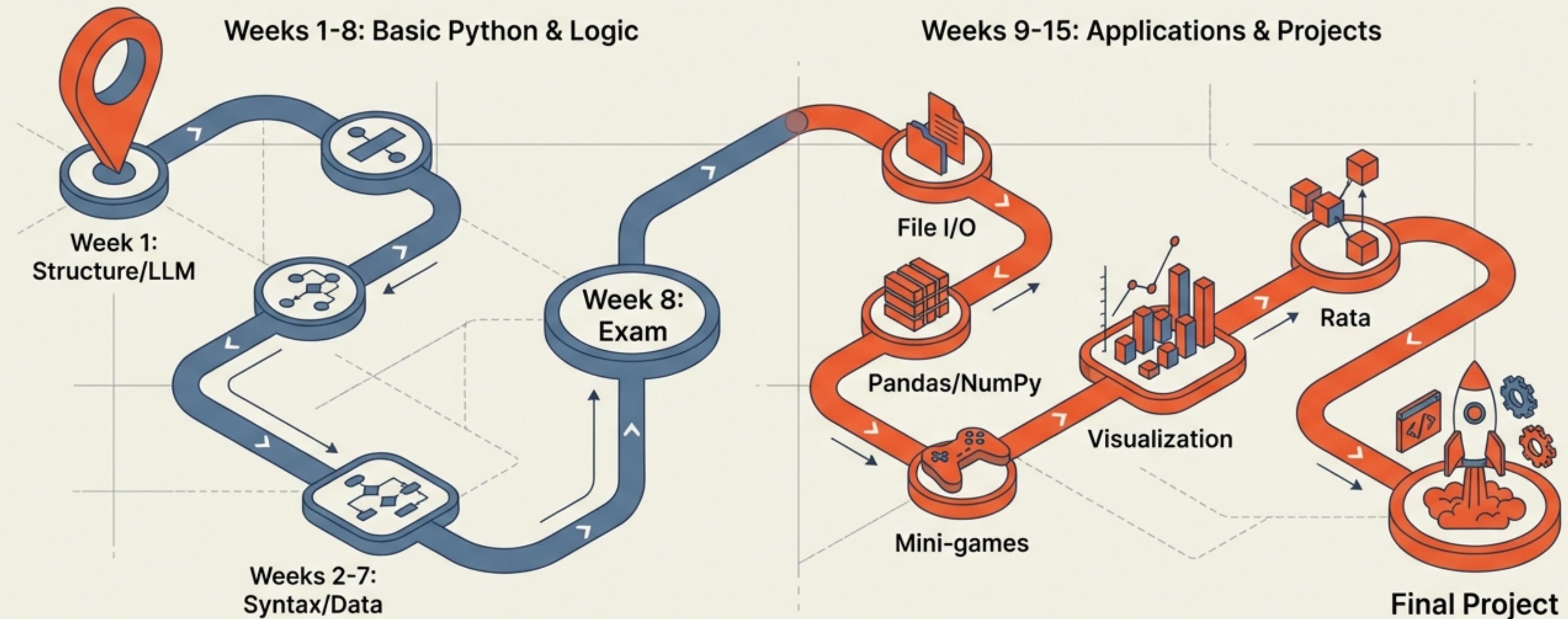
User: How can I optimize it?

AI: You could consider using type hinting and

Ask AI about your code...

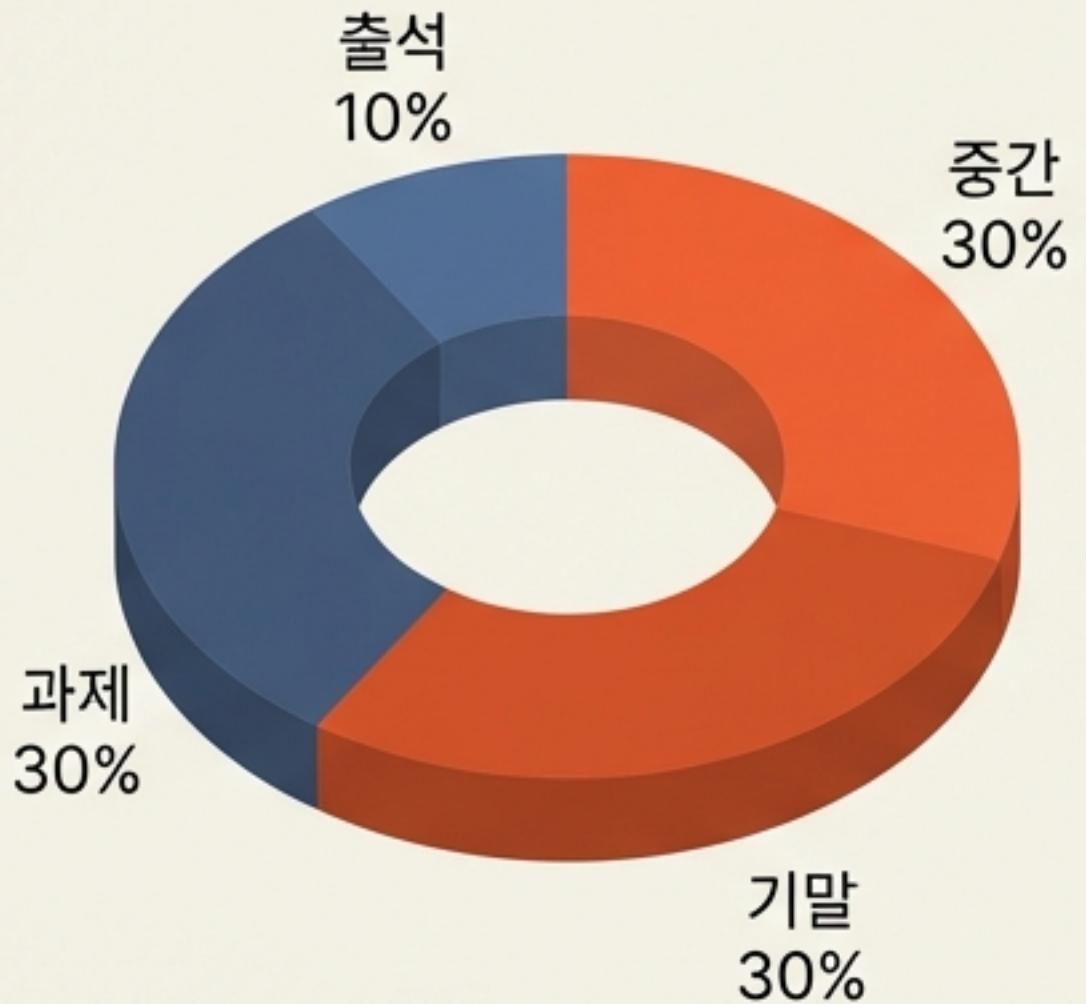
Fundamental Logic + AI Workflow

15주 학습 로드맵: 기초에서 프로젝트까지



강의 운영 및 평가 기준

Evaluation (평가)



Attendance Rule (출석)



출석률 3/4 미만 = F 처리

Resources (자료)

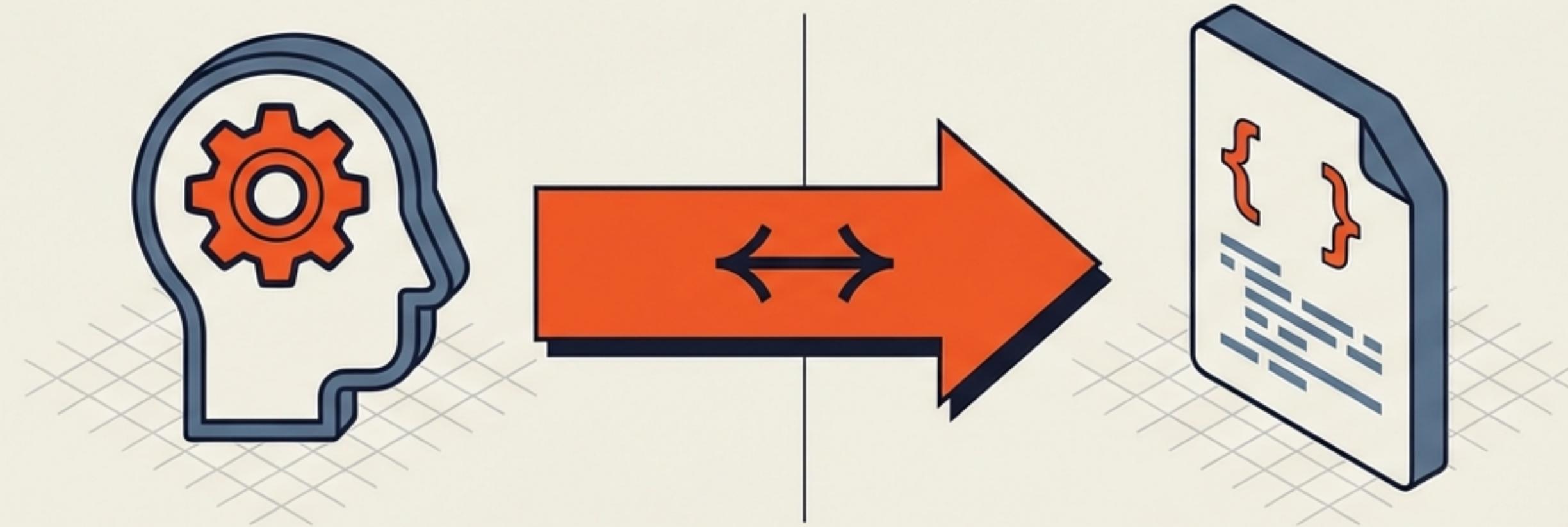


강의자료 및 코드 저장소

A screenshot of a GitHub repository page titled "At programming". The repository has 14 stars, 13 forks, and 13 issues. The codebase consists of 1 file with 11 lines of code. The repository was created on April 1, 2017, and last updated on April 1, 2017. The repository URL is <https://github.com/At-programming/At-programming>.

프로그래밍이란?

사람의 문제 해결 절차(Algorithm)를 컴퓨터가 실행 가능한 규칙으로 바꾸는 것.



- 문법 (Syntax):

어떻게 쓰는가 (Grammar/Rules)

- 의미 (Semantics):

무슨 뜻인가 (Meaning/Intent)

언어는 사람과 컴퓨터 사이의 약속입니다.

번역의 두 가지 방식: 컴파일 vs 인터프리트

Compile (C/C++, Java)



- 실행 전 변환
- 빠른 속도
- 문법 오류 미리 발견

Interpret (Python, JS)



- 한 줄씩 실행
- 즉시 피드백
- 런타임 에러 가능성

파이썬의 실행 흐름



.py 파일

결과 출력

실행 (Run)

Learning
Loop

→ 관찰 (Observe)

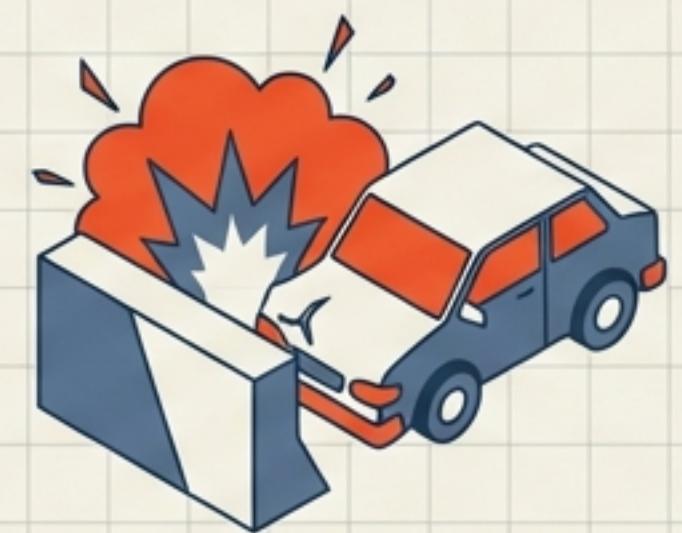
수정 (Fix)

프로그래머가 만나는 세 가지 오류



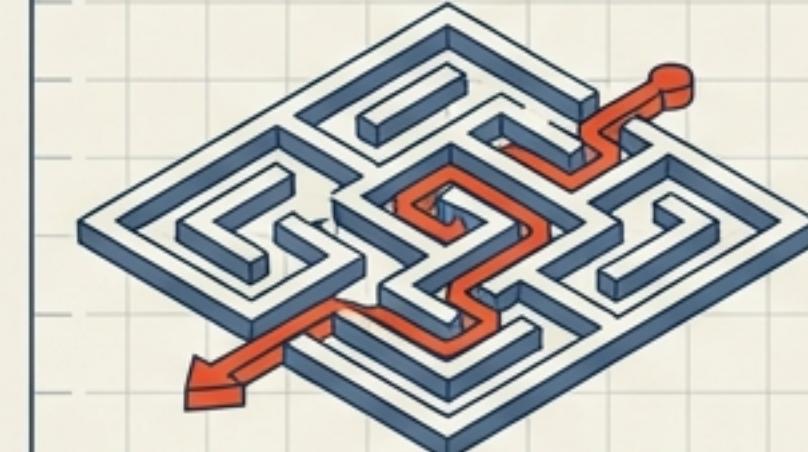
Syntax Error
(문법 오류)

실행 자체가 불가.
예: 괄호 누락.



Runtime Error
(런타임 오류)

실행 중 발생 (Crash).
예: 0으로 나누기.



Logic Error
(논리 오류)

실행은 되나 결과가 틀림.
예: 잘못된 수식.

디버깅 기본 루틴

증상 재현
(Reproduce)

입력/환경 고정

가설 및 수정

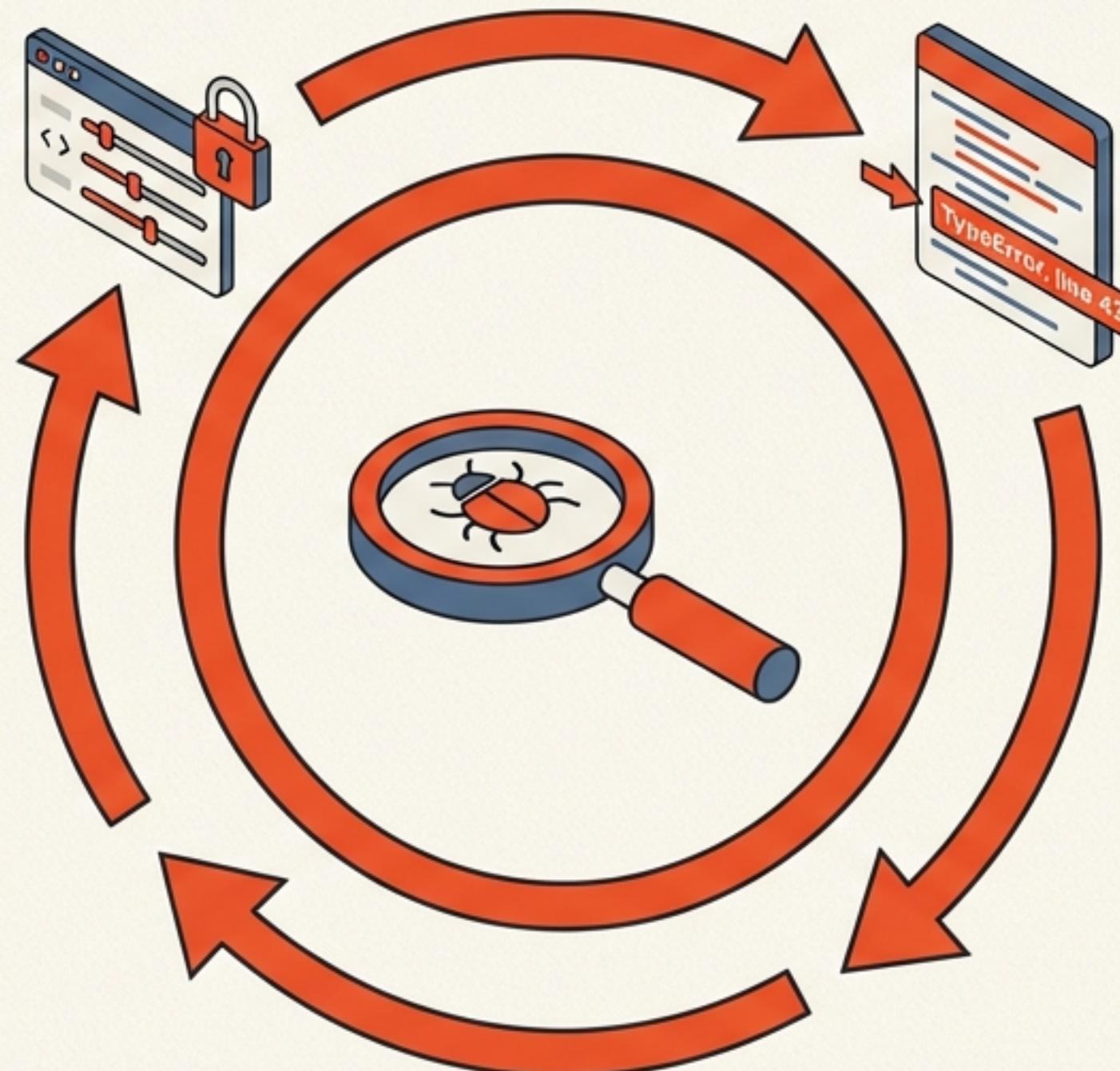
한 번에 하나씩 변경

Traceback 읽기

에러 종류와 줄 번호 확인

최소 재현 코드 (MRE)

문제를 가장 작은 형태로 줄이기



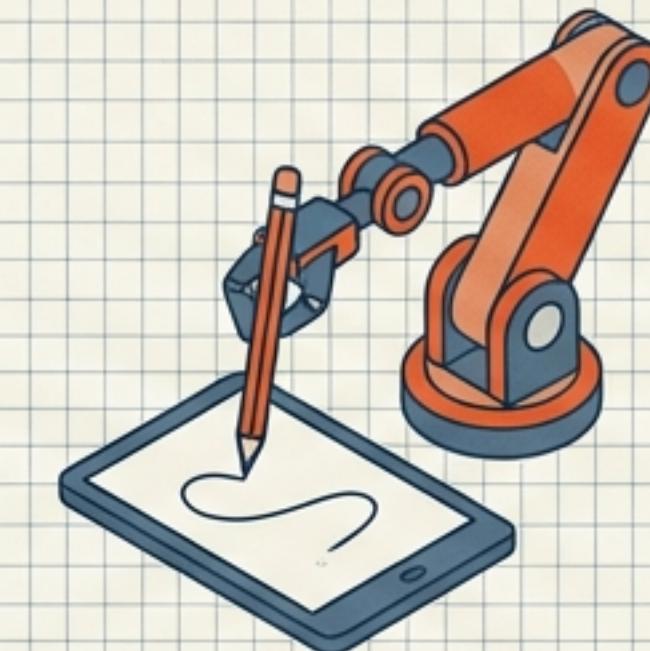
With AI 코딩: 역할 분담의 원칙

Human (Lead)



문제 정의
요구사항 정리
최종 판단 & 윤리

AI (Assistant)



초안 코드 생성
개념 설명
오류 원인 제안
리팩터링



LLM의 본질과 한계

확률적으로 다음 단어를 예측하는 모델



Caution

Hallucination (환각): 그럴듯한 오답을 확신 있게 말함.
AI 답변은 '초안'이며, 검증(Verification)이 필수.

완벽한 답변을 위한 프롬프트 템플릿

Recipe Card

1. 역할 (Role)

너는 친절한 파이썬 튜터야

2. 목표 (Goal)

이 코드를 초보자가 이해하게 설명해줘

3. 맥락 (Context)

나는 이제 막 배우기 시작했어

4. 제약 (Constraints) 핵심만 5줄로 요약해

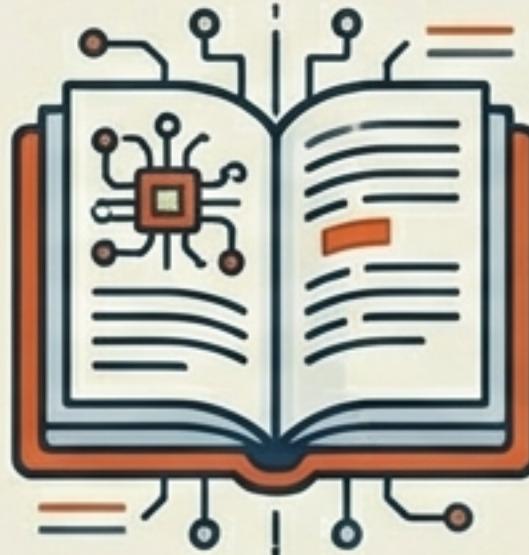
5. 형식 (Format)

리스트 형태로

6. 코드 (Code)

[Code snippet here]

실전 AI 활용 워크플로우



코드 설명 (Explanation)

이 코드를 줄별로
해석해줘.



디버깅 (Debugging)

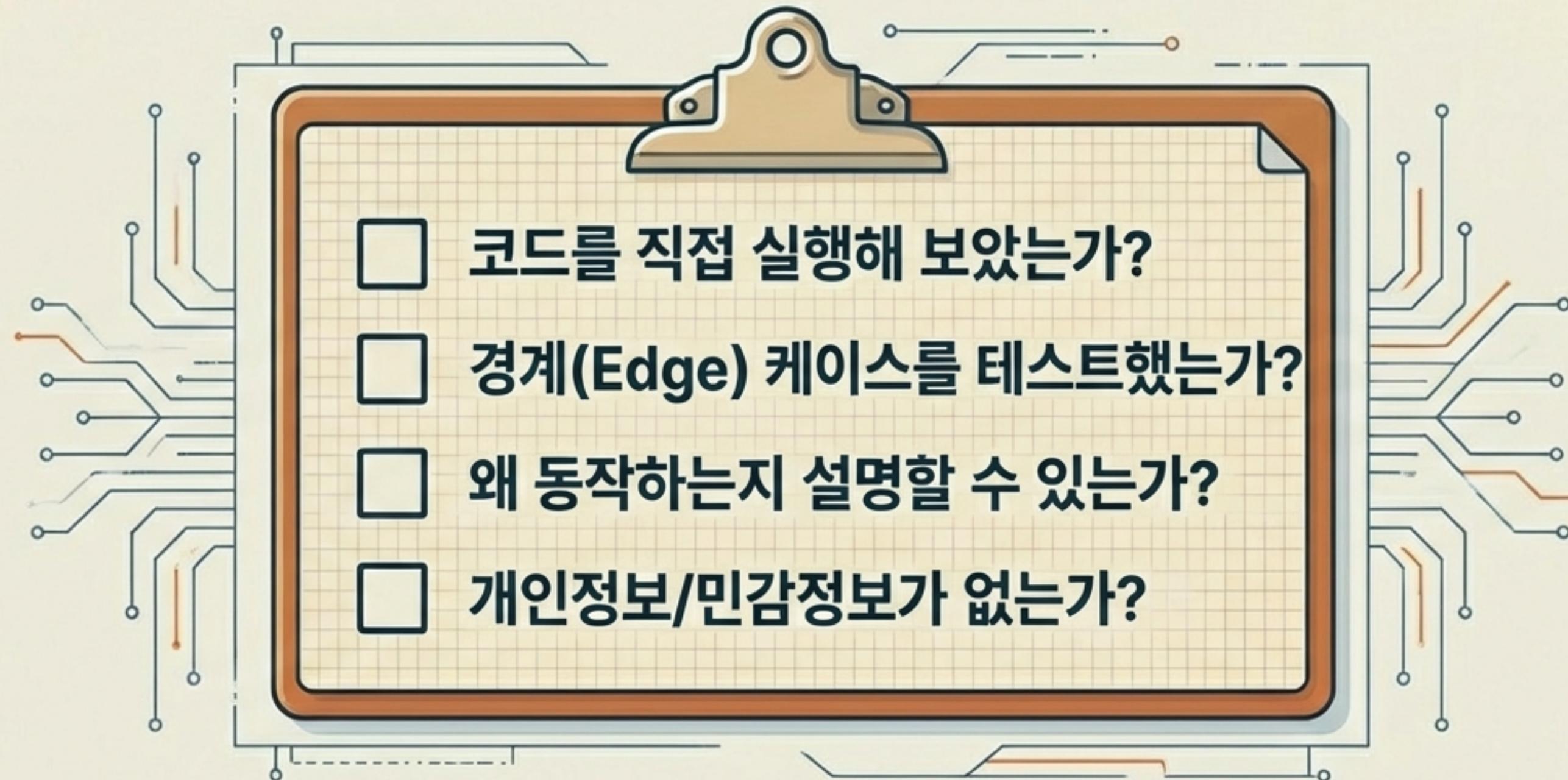
에러 메시지가 이건데
원인이 뭐야?



리팩터링 (Refactoring)

변수명을 개선하고
주석을 달아줘.

AI 답변 검증 체크리스트



“검증하지 않은 AI 답변은 참고 의견일 뿐입니다.”

핵심 개념 미니 퀴즈

Q: 문법 오류(Syntax Error)는 언제 발견되는가?

**A: 실행 전
(Before Run)**



Q: LLM 답변을 대하는 태도는?

**A: 참고하고 검증
(Reference & Verify)**



Q: 디버깅의 첫 단계는?

A: 증상 재현 (Reproduce)



Q: 논리 오류(Logic Error)의 특징은?

A: 실행되지만 결과가 틀림.



요약 및 다음 주 과제

This Week's Mission

1. Programming = Syntax + Semantics + Execution.
2. Assignment: Write 3 Prompts (Explain, Debug, Refactor). Verify results.
3. Next Week: Python Environment Setup & Hello World.

