

NOTI : AI 시대의 코딩교육 혁신 프로젝트

초급자를 위한 피드백 중심의 혁신적 코딩교육 시스템





발표자: 김병정 | 소프트웨어학부, AI융합연구원

2025년 10월 30일

발표 목차

- 1 기존 코딩교육의 한계 
- 2 초급자 입장에서의 문제점 
- 3 즉각적이고 정확한 피드백의 중요성 
- 4 실제 예시로 보는 어려움 
- 5 기존 IDE와 AI 활용의 한계 
- 6 혁신 방안 및 기대 효과 
- 7 IT 서비스 발전에 따른 개발툴 복잡성 증가 
- 8 코딩테스트와 프로젝트 평가 방식 비교 
- 9 라이브코딩 서비스 소개 
- 10 Kafka와 Hugging Face 데이터셋 업로드 전략 
- 11 실제 서비스 데모 화면 





기존 코딩교육의 한계

-  **정형화된 커리큘럼과 일방향 강의**
학생의 이해도와 관계없이 진행되는 일방적 지식 전달 방식
-  **실시간 피드백의 부족**
학습 과정에서 즉각적인 피드백이 없어 초급자들이 잘못된 개념을 형성
-  **문제 풀이 위주의 평가**
코드의 품질, 효율성, 가독성보다 단순 결과 도출 여부만 중시
-  **실무와의 괴리**
실제 개발 환경과 동떨어진 예제들로 실무 적응에 어려움 발생






코딩교육 피드백 사이클

-  학습 (동영상/교재)
-  코드 작성 시도
-  문제 발생 (오류/막힘)
-  **피드백 지연 (수업 후 또는 다음 수업)**
-  학습 흐름 단절, 의욕 저하

초급자 입장에서의 문제점

-  **자신의 지식 격차를 인식하지 못함**
무엇을 모르는지조차 알지 못하는 상태로, 도움 요청이 어려움
-  **효과적으로 질문하지 못함**
문제 상황을 정확히 설명하지 못해 적절한 도움을 받지 못함
-  **에러 메시지 이해 부족**
코드 오류 메시지를 해석하지 못해 디버깅 과정에서 좌절
-  **학습 방향성 상실**
무엇부터 배워야 할지 모르는 상태에서 비효율적 학습 경로 선택

초급자의 학습 혼란 사이클

-  코딩 학습 시작
-  에러 발생 / 막힘 현상
-  **문제 원인 파악 불가**
-  무작위적 해결 시도
-  좌절감 / 자신감 하락

즉각적이고 정확한 피드백의 중요성

⚡ 초급자는 빠른 피드백이 생명줄
실시간으로 오류를 식별하고 수정하는 과정이 학습 효율을 극대화

👁️ 작은 실수가 큰 좌절로 이어짐
세미콜론 누락, 괄호 불일치와 같은 사소한 실수도 초급자에게는 큰 장벽

예시

`if (x = 5) { ... }` vs `if (x == 5) { ... }`
할당 연산자와 비교 연산자의 혼동

💡 적시 피드백은 학습 동기 유지의 핵심
문제 해결의 즉각적인 성취감이 지속적인 학습 동기 부여와 직결

피드백 시점에 따른 학습 효과 비교




즉각적인 피드백

- ✓ 오개념 형성 방지
- ✓ 학습 흐름 유지
- ✓ 문제해결 능력 향상
- ✓ 자신감 상승
- ✓ 학습 의욕 지속

지연된 피드백

- ✗ 잘못된 습관 형성
- ✗ 학습 흐름 단절
- ✗ 좌절감 누적
- ✗ 같은 실수 반복
- ✗ 학습 포기 가능성 증가





실제 예시로 보는 어려움

-  **VSCode 환경설정의 어려움**
초보자는 확장 프로그램 선택, 설정 파일 구성, 디버깅 환경 설정 등에서 어려움을 겪습니다. 어떤 요소가 필요한지 모르는 상태에서 진입장벽을 느낍니다.
-  **문제 해결 과정에서의 막힘**
에러 메시지를 이해하지 못하거나, 어느 부분에서 코드가 잘못되었는지 식별하지 못해 디버깅을 포기하는 사례가 많습니다. 신속한 피드백 부재가 학습 흐름을 끊습니다.
-  **AI 도구 사용 시 이해 부족**
Chat GPT나 GitHub Copilot 같은 AI 도구가 제공하는 코드를 그대로 복사해 사용하지만, 그 작동 원리나 구조를 이해하지 못하여 실제 학습이 일어나지 않습니다.

초급자의 학습 시나리오

- 1 문제를 마주함**
"배열에서 중복 요소를 제거하는 함수를 작성하세요."
- 2 접근 방식 탐색**
초급자: "for문을 써야 할 것 같은데... 어떻게 중복을 찾지?"
방향 설정 어려움
- 3 AI에 질문**
"const uniqueArray = [...new Set(array)];라는 한 줄 코드를 받음"
정답 획득 **원리 미이해**
- 4 평가 단계**
"코드는 동작하지만 Spread 연산자, Set 객체의 개념과 원리를 설명하지 못함"
진정한 학습 부재

기존 IDE와 AI 활용의 한계

-  **VSCode/CursorAI의 적합성**
전문 IDE는 이미 자율적 탐색이 가능한 **중급 이상** 학습자에게 적합하나, 초급자에게는 오히려 진입장벽
-  **AI 풀이의 평가 문제점**
AI가 완벽한 정답을 제시하여 학생 스스로의 사고과정과 실력 향상을 평가하기 어려움
-  **정제된 문제의 아이러니**
문제가 잘 정의될수록 AI가 더 정확한 답변을 제공하는 교육적 딜레마 발생
-  **진정한 학습의 어려움**
도구에 의존하게 되면 근본적인 문제해결 능력과 코딩 원리 이해가 어려움

교육 도구 효과 비교

학습 난이도

VSCode: 초기 설정 및 환경 구축이 복잡하여 초보자에게 어려움

AI 도구: 사용은 쉬우나 **문제** 학습 원리 파악이 어려움

평가 가능성

기존 과제: AI가 완벽한 답안을 제공하여 **평가 불가** 상태

정제된 문제: 더 명확한 문제일수록 AI 답변의 정확도가 **상승**

교육적 가치

현재 도구들은 **자기주도적 사고**보다 **결과 도출**에 집중된 한계

혁신 방안 및 기대 효과

-  **문제의식 자체를 평가하는 시스템**
정답 유무가 아닌, 문제 접근 방식과 사고과정을 평가하는 혁신적 방법론
-  **개별 피드백 자동화**
학습자 수준과 진도에 맞춘 맞춤형 피드백으로 성장 곡선 최적화
-  **다이나믹 피드백 시스템**
실제 교육현장의 문제 상황을 실시간 분석하여 적시에 필요한 도움 제공
-  **AI-인간 협력 모델**
인간 교육자의 통찰력과 AI의 분석력을 결합한 신뢰도 높은 교육 환경 구축

AI 코딩교육 혁신 시스템

 학생 코드 제출 및 실시간 분석

 문제 접근방식 패턴 인식

 즉각적인 맞춤형 피드백

 교육자에게 학습 패턴 인사이트 제공

 지속적인 학습 데이터 분석 및 개선

IT 서비스 발전에 따른 개발툴 복잡성 증가

IT 서비스가 발전하면서 개발툴, 프레임워크, 플랫폼, 서비스들이 폭발적으로 늘어나 이해와 선택이 점점 어려워지고 있습니다.

</> IDE / 코드 편집기

VSCode PyCharm IntelliJ Eclipse Xcode

🔑 버전 관리

Git GitHub GitLab Bitbucket

⚙️ 빌드 / CI-CD

Jenkins GitHub Actions GitLab CI CircleCI

☰ 백엔드 프레임워크

Flask Django Express Spring FastAPI

💻 프론트엔드 프레임워크

React Angular Vue.js Svelte

📱 모바일 개발

React Native Flutter Swift Kotlin

🗄️ DB / 스토리지

MySQL PostgreSQL MongoDB Redis
Firebase

☁️ 클라우드 플랫폼

AWS Azure GCP DigitalOcean

🔗 컨테이너 / 가상화

Docker Kubernetes Vagrant Podman

↔️ API / 통신

Postman Swagger GraphQL gRPC

✍️ 테스트 / 품질

Jest Mocha Pytest Selenium Cypress

🛡️ 보안 / 인증

OAuth JWT Keycloak HashiCorp Vault

🧠 데이터 / AI / ML

Pandas TensorFlow PyTorch Hugging Face
LangChain

📄 로깅 / 모니터링

Prometheus Grafana ELK Stack

👥 협업 / 문서화

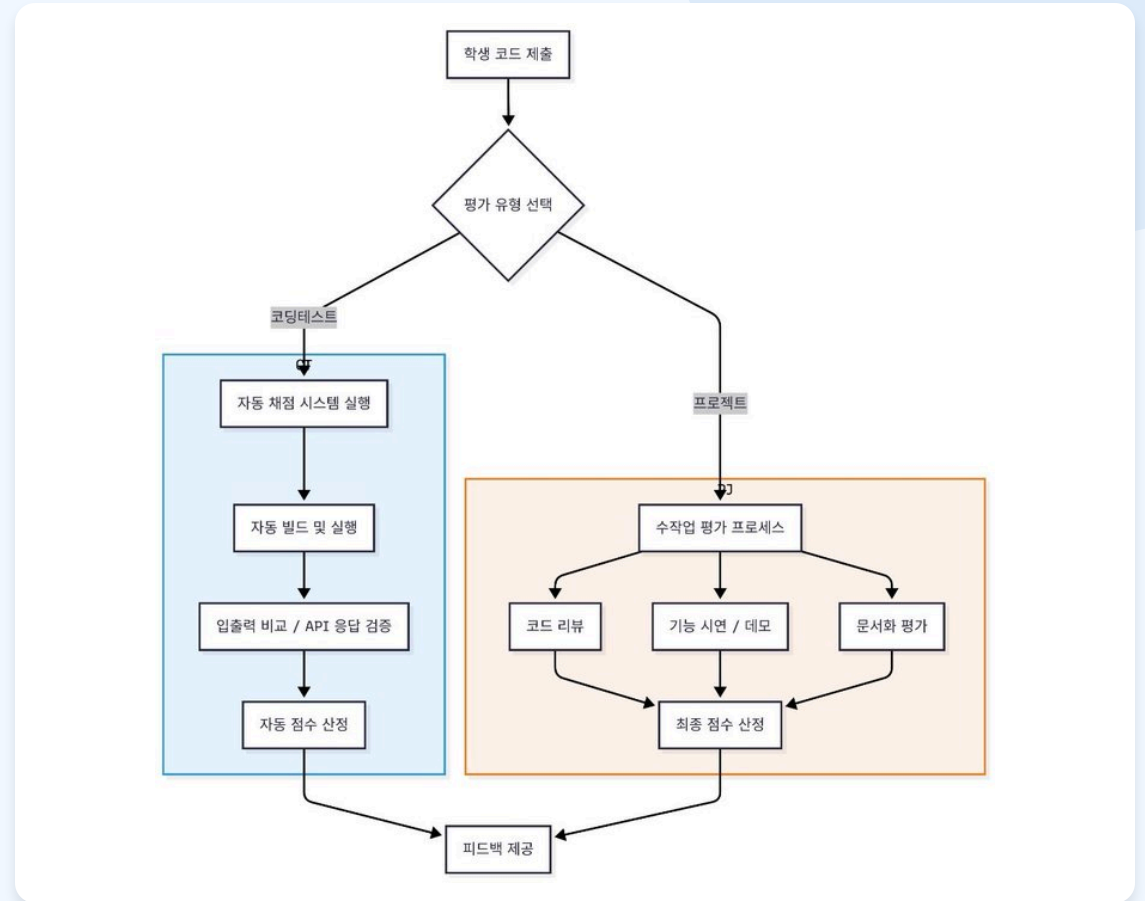
Notion Confluence Slack MS Teams

코딩테스트와 프로젝트 평가 방식 비교

- 코딩테스트 평가 방식**
자동 채점 시스템을 통한 빠른 결과, 정해진 입출력 테스트 케이스 기반 평가
- 프로젝트 평가 방식**
코드 리뷰, 기능 시연, 문서화 품질 등 다각도 평가, 자동화가 어려운 영역
- 평가 유형 선택의 중요성**
학생의 학습 목표와 수준에 따른 적절한 평가 방식 선택이 교육 효과 결정

평가 방식별 특징 비교

- 코딩테스트** 자동화된 즉각 피드백, 객관적 채점
- 프로젝트** 종합적 역량 평가, 창의성과 문제해결 과정 중시



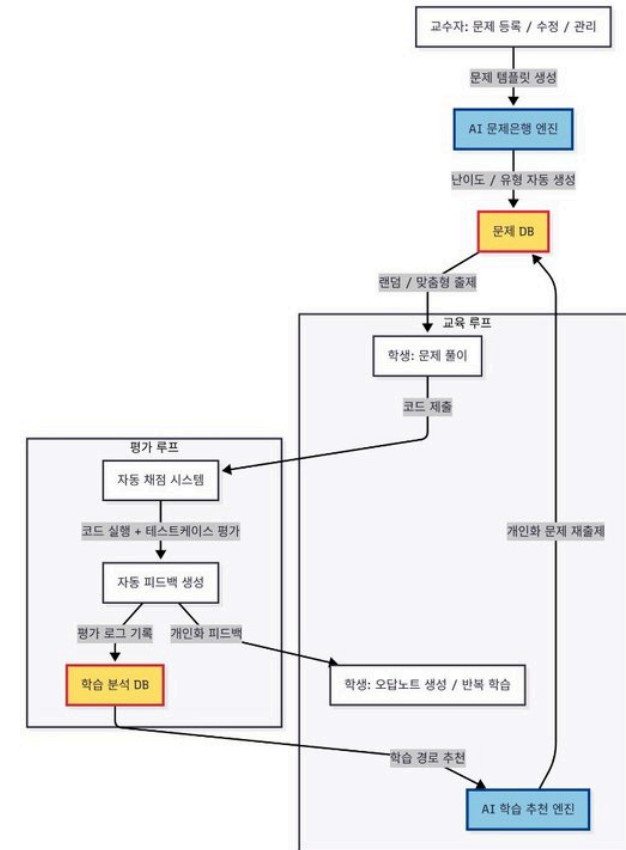
AI 기반 문제은행 및 평가 통합 구조

문제 생성부터 학습 피드백까지의 자동화 흐름

- AI 문제은행 엔진**
교수자의 문제 템플릿을 기반으로 AI가 다양하고 수준별 문제를 자동 생성하는 시스템
- 통합 평가 플랫폼**
학생 코드 제출부터 자동화된 평가, 맞춤형 피드백까지 하나의 흐름으로 연결된 구조
- 학습 분석 데이터베이스**
학생의 코드 제출 내역, 오류 패턴, 학습 진행도를 추적하여 AI 학습 추천 엔진에 활용

시스템 특징

- 완전 자동화 흐름:** 교수자의 문제 템플릿 생성부터 학생의 학습까지 모든 과정이 자동화된 파이프라인으로 연결
- 개인화된 피드백:** 학생별 코드 제출 패턴과 오류 유형에 따른 맞춤형 AI 피드백 제공
- 학습 데이터 추적:** 모든 과정의 데이터가 DB에 저장되어 교육 품질 향상에 기여



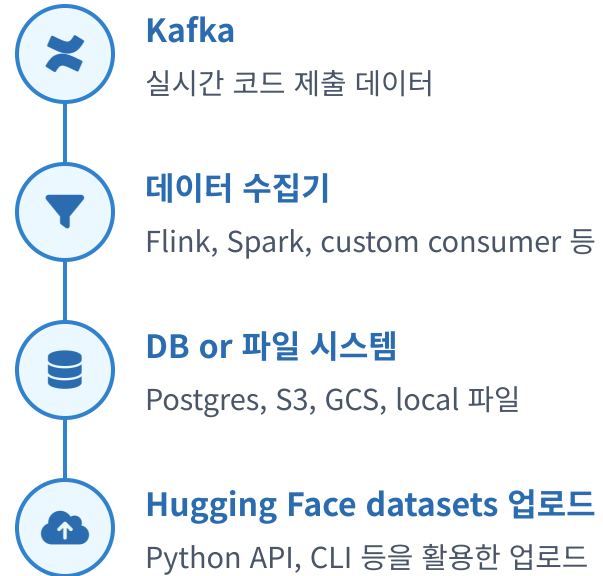
Kafka와 Hugging Face 데이터셋 업로드 전략

- 데이터 파이프라인 구축**
Kafka의 실시간 코드 제출 데이터를 Hugging Face 데이터셋으로 안전하게 변환하기 위한 처리 과정
- 서로 다른 목적성 인식**
 - 👉 Kafka: 단순히 데이터 스트림(흐름)을 처리하고 전달하는 시스템
 - 👉 Hugging Face: 모델, 데이터셋, 코드, space 등을 공유하는 플랫폼

🔑 Kafka + Hugging Face 데이터셋 활용 시 주의

- Kafka 데이터는 실시간성 → Hugging Face 데이터셋은 정적 공유용
- 개인정보, 코드 제출 데이터 → 익명화/가공 필수
- Hugging Face는 대용량 로그 raw upload보다 정제된 학습 데이터셋 공유에 적합

데이터 파이프라인 흐름도



라이브 코딩 + Event 아키텍처 + NPC 기반 피드백 시스템



실시간 라이브 코딩 환경

핵심 1

코드 작성 즉시 분석 및 실행되는 환경으로 코드와 결과를 동시에 확인



Event 기반 아키텍처 적용

핵심 2

코드 에디터의 이벤트가 스트림 브로커를 통해 실시간 처리되는 반응형 구조



NPC 기반 게이미피케이션 피드백

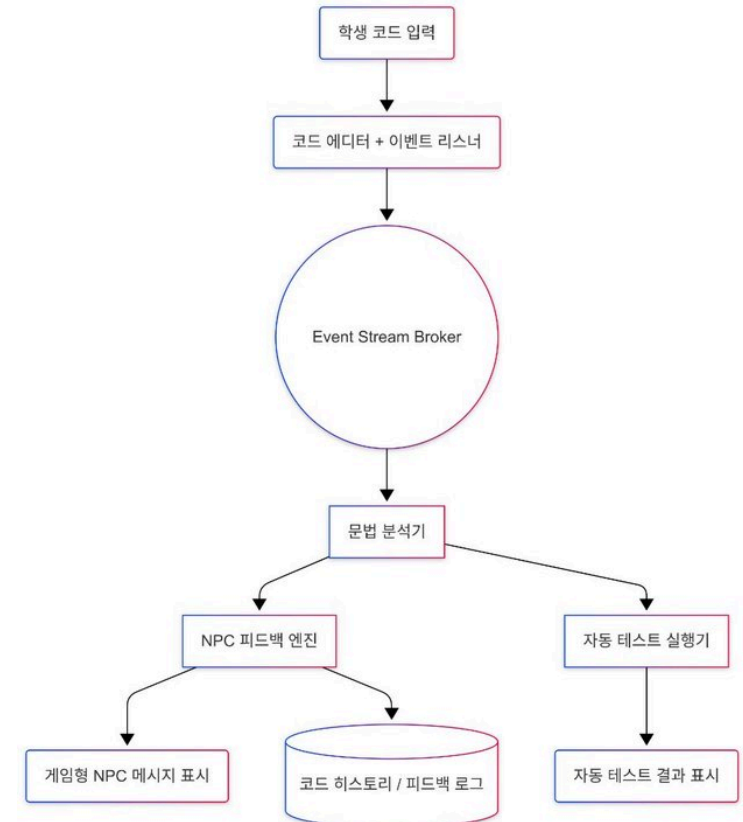
핵심 3

문법 분석기에 연결된 NPC 캐릭터가 친근하고 직관적인 피드백 메시지 제공




통합 시스템 작동 예시

- 1 **라이브 코딩**: 학생이 `for i in range(5)` 입력
- 2 **Event 처리**: 이벤트 브로커가 코드를 SyntaxAnalyzer로 전달, 콜론 누락 탐지
- 3 **NPC 피드백**:
"콜론(:)이 빠졌어! for문 끝엔 콜론이 필요해요~"
- 4 피드백 기반으로 학생이 수정 후 자동 테스트 통과

✔ "모든 테스트 통과! 훌륭해요!"



실제 서비스 데모 화면

-  **파이썬 과학 프로그래밍 기초 UI**
코딩 초급자를 위한 직관적이고 효율적인 사용자 인터페이스로 설계
-  **참여형 학습 환경**
교수자와 학생 간 실시간 코드 공유 및 피드백을 통한 효과적인 학습 진행
-  **실전 문제 중심 학습**
hello world 같은 기초부터 시작해 단계적으로 실력을 쌓아가는 과정 설계

★ 시스템 주요 기능

- 동시 비교** 제출 코드와 정답 코드 동시 확인
- 점수 확인** 실시간 진행 상황과 점수 추적
- 공동 작업** 교수자와 학습자 동시 접속 및 피드백



The screenshot displays a web interface for a Python programming course. The main area is divided into two columns for student work. The left column shows the work of '김병정 교수' (Kim Byung-jung) and '나' (Me), with '파이썬 기초 문제1' (Python Basic Problem 1) selected. The right column shows the work of '나' (Me) and '김병정 교수' (Kim Byung-jung), with '파이썬 기초 문제2' (Python Basic Problem 2) selected. Each problem area includes a description, conditions, code input field, and an '실행하기' (Run) button. A sidebar on the right lists students with their IDs and status indicators. The top navigation bar shows '2주차' (Week 2) and '속도 확인 보드' (Speed Check Board).

감사합니다

질문과 답변 시간입니다

김병정

소프트웨어학부, AI융합연구원



kevin@hallym.ac.kr