

비개발자도 내 마음대로! 바이브코딩 생존 가이드

코딩 지식 제로, 영어 문법 제로!
카카오톡을 할 줄 안다면 누구나 나만의 학교 도구를 만들 수 있습니다

오늘의 학습 여정

1부

문턱이 낮아진 코딩 세계와
바이브코딩의 본질

2부

AI 언어학 마스터:
대화의 규칙과 한계 극복

3부

망하지 않는 지름길:
교사용 PRD 작성법

4부

개발자의 방 구경하기:
안티그래비티와 개발 환경

5부

프로젝트를 살리는 디버깅과 철저
한 학교 보안

6부

우렁각시 자동 배포: Git 연동과
Vercel 정복

바이브코딩의 어원



개념

- 엄격한 영문법이나 소스코드가 아니라, 내가 원하는 '의도와 느낌(Vibe)'을 한글로 전달하여 프로그램을 만드는 방식



교실 비유

- 새로 부임한 기간제 선생님에게 '컴퓨터실 갈 때 애들 줄 맞춰서 데려가 주시고, 조용히 시켜주세요'라고 대화하듯 편하게 지시하는 것과 같습니다



시각적 변환

- 한글 말풍선이 컴퓨터 코드로 자연스럽게 변환되는 과정

문턱이 낮아진 코딩: 선생님은 기획자입니다



과거의 코딩

- 난해한 영어 명령어, 세미콜론 하나만 틀려도 터지는 프로그램
- 기술자의 영역



현재의 코딩

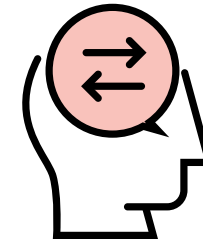
- 아이디어와 교육적 철학만 있으면 AI가 손과 발이 되어 대신 타이핑
- 기획자의 영역



선생님의 역할

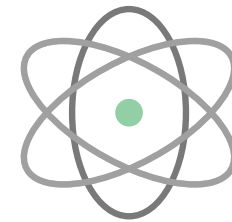
- 선생님은 이제 코딩 기술자가 아니라, AI 비서를 부리는 '교실의 디렉터(감독)'입니다

바이트코딩, 10분 만에 망하는 이유



증상 1

- 대화가 길어질수록 첫 기능이 망가짐
- 초기 설정 무시



증상 2

- 코드가 누더기가 되어 작동을 멈춤
- 프로그램 충돌



증상 3

- AI가 내 말을 무시하고 똑같은 에러만 반복함
- 무한 루프

문제는 AI가 아니라 '이것'이 없기 때문



무작정 '만들어줘'라고
외치는 개발 방식의 문제점

- AI 성능의 문제가 아닙니다
- 명확한 계획 없이 즉흥적으로
요구하면 프로젝트가 무너집니다
- 대화가 길어질수록 초기 기능이
망가지는 이유는 설계도 부재 때문
- 체계적인 접근 없이는 10분 만에
포기하게 됩니다

생존을 위한 핵심 열쇠 2가지

- 무엇을 만들지 명확히 정의하는
'설계도(PRD)'
- 고장 난 곳을 실시간으로 감시하는
'테스트 환경(Harness)'
- 이 두 가지가 있어야
AI 코딩이 성공합니다
- 기초 공사 없이 모래 위에 집을
짓는 것과 같은 실수를 방지합니다



AI 코딩의 비밀병기: 하네스(Harness)란?



하네스의 개념

- 특정 기능을 자동 검증
- 테스트 묶음인 안전벨트
- 번지점프용 보호 수트처럼 고정

교실 비유로 이해하기

- 문제 풀이 자동 채점기 설치
- 선생님 대신 자동 검증
- 교무실 업무 자동화와 유사

AI 수정 시

안전장치 역할

- 수정 전 필수 점검
- 기존 기능 보호 장치 적용
- 코드 변경 시 이전 기능 유지

프로그램 보호 원리

- 수정마다 기능 정상 자동 확인
- 프로그램 누더기 방지
- 안정적 개발 환경 구축

교무실 비유로 이해하는 '테스트 하네스'



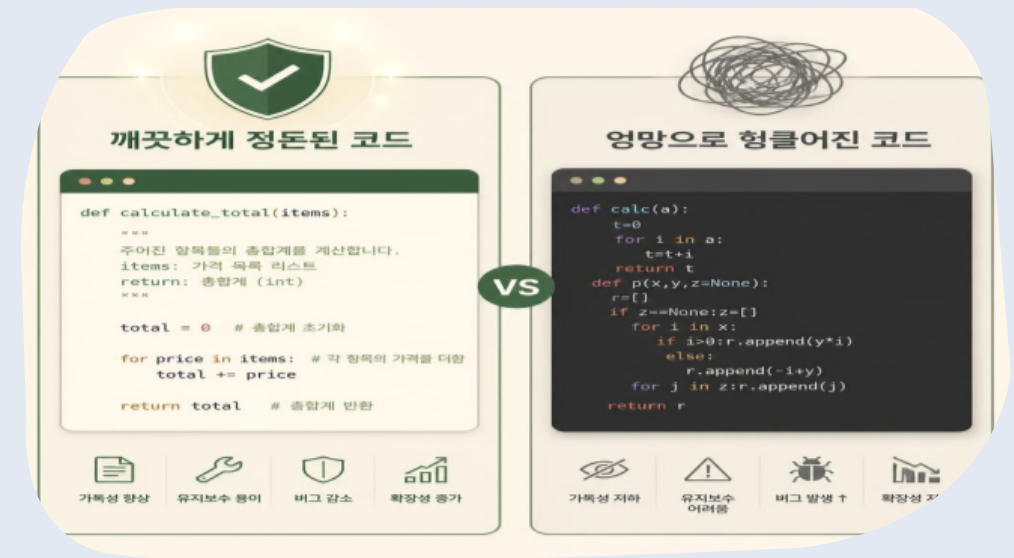
자동 채점기 시스템

- 학생들에게 수학 문제를 풀게 한 뒤 자동 채점기를 설치
- 선생님이 매번 채점하지 않아도 자동으로 정답 확인
- 교무실 업무 자동화와 동일한 개념



코드 수정 시 자동 검증

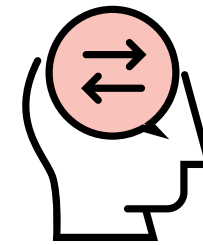
- 코드가 수정될 때마다 '1번 기능 정상! 2번 기능 정상!' 자동 외침
- 기존 기능이 망가지지 않았는지 실시간 점검
- AI에게 새 작업 지시 전 안전 확인



누더기 방지 원리

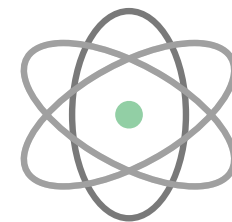
- 프로그램이 누더기가 되는 것을 원천 차단
- 초기 기능이 대화 중 망가지는 문제 해결
- 안정적이고 지속 가능한 개발 환경 제공

프롬프트(Prompt): AI 비서의 업무 지시서



프롬프트의 개념

- AI에 입력하는 지시어·질문·맥락
- AI와의 모든 대화 입력
- 명령이자 요청서



교실 비유

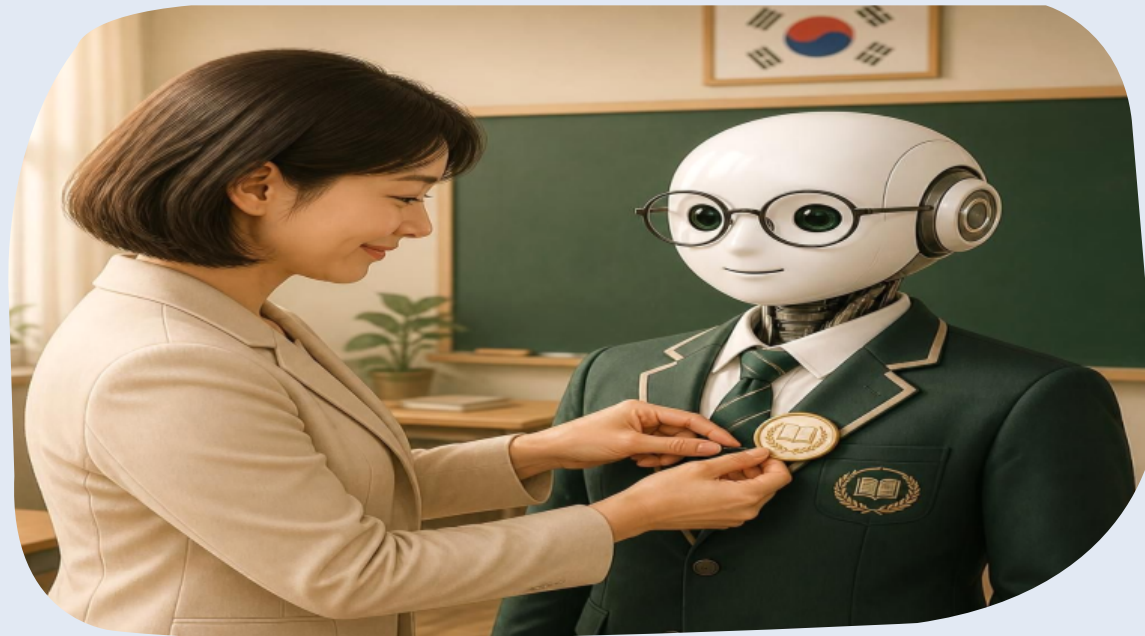
- 새 교생에 주는 주간 안내서
- 업무를 구체적으로 전달하는 문서
- 명확할수록 수행 품질 ↑



결과물 퀄리티 향상 비결

- 구체적일수록 결과물 퀄리티 ↑
- 애매한 지시는 엉뚱한 결과
- 체크리스트형 프롬프트가 최적

Rules (시스템 프롬프트)



AI에게 영구적인 역할 부여

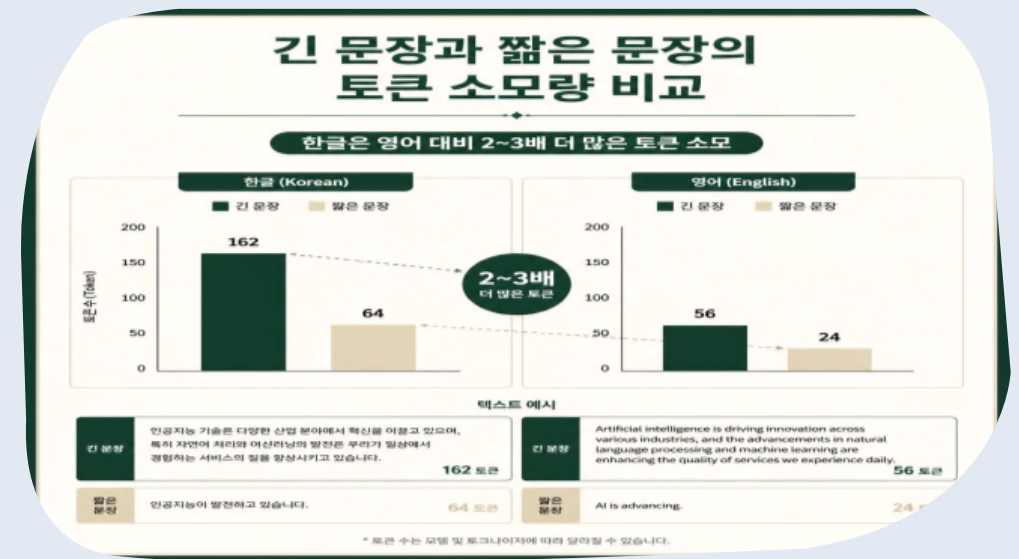
- AI에게 영구적인 성격, 역할, 절대 규칙을 부여하는 최상위 지시서
- 대화 전체에 걸쳐 유지되는 AI의 '정체성' 설정
- 모든 답변의 기준이 되는 핵심 규칙
- 한 번 설정하면 계속 적용되는 영구 지침

학교 현장 활용법

- '너는 20년 경력의 베테랑 정보 교사이자, 친절한 수석교사야'
- '바이브코딩이 익숙하지 않은 선생님들을 위해 어려운 용어는 교무실 비유를 들어 설명해줘'
- 이렇게 규칙을 심어두면 AI가 교사 관점으로 답변
- 전문 용어를 학교 환경에 맞게 쉽게 풀어서 설명



토큰(Token): AI가 글자를 읽는 화폐 단위



토큰의 개념

- AI가 텍스트를 인식하고 처리하는 최소 단위
- 영어는 단어 단위, 한글은 음절/형태소 단위로 쪼개짐
- AI가 글자를 읽을 때 사용하는 '화폐'

교실 비유

- 인쇄기를 돌릴 때마다 소모되는 '토너 가루'
- 복사기에서 사용하는 'A4 용지 한 장'의 비용
- 교무실에서 프린트할 때 용지 소모되는 것과 동일

토큰 소모 원리

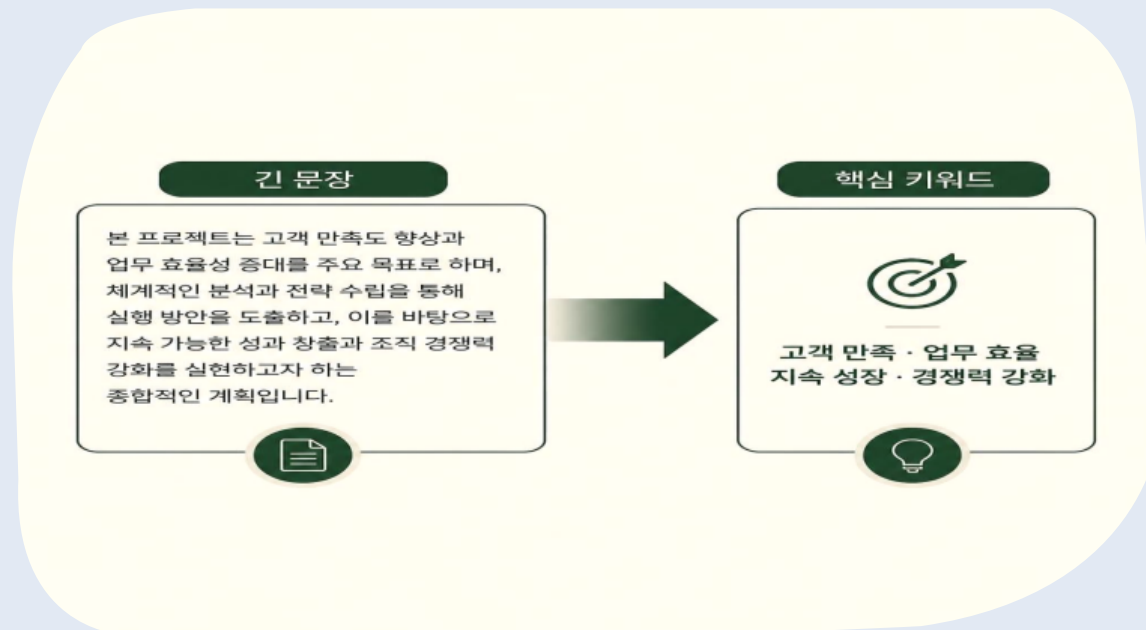
- 글자가 길어질수록 토큰 화폐가 소모됨
- 한글은 영어보다 토큰 소모량이 2~3배 많음
- 비용과 속도를 아끼려면 간결하게 작성

가성비 좋은 학교 맞춤형 토큰 절약법



한글 토큰 소모의 현실

- 한글은 영어보다 토큰 소모량이 2~3배 많습니다
- 쓸데없는 미사여구를 줄여야 비용과 속도를 아낄 수 있습니다
- 글자가 길어질수록 토큰 화폐가 더 많이 소모됩니다
- 핵심 키워드 중심으로 간결하게 작성하는 것이 중요합니다



프롬프트 다이어트 실전 예시

- 나쁜 예: '부탁 좀 드려도 될까요?
아주 예쁜 초록색 초등학교 칠판
느낌으로 배경을 정성스럽게 칠해
주시면 감사하겠습니다.'
- 좋은 예: '배경 디자인: 눈이 편안한
초등 교실 초록색 칠판 스타일.'
- 긴 문장을 핵심 키워드로 압축하면
토큰 절약 효과가 큼니다

컨텍스트(Context): 대화의 맥락과 기억력



개념

- AI가 현재 답변을 만들기 위해 머릿속에 띄워놓은 모든 맥락 정보
- 이전 대화 내용, 소스코드, 시스템 규칙 등을 포함합니다

작동 원리

- 과거 대화 기록들이 사슬처럼 연결되어 있습니다
- AI는 이 맥락을 바탕으로 일관된 답변을 생성합니다

교실 비유

- 부장 선생님과 회의할 때 지난주 대화 내용을 기억하는 것과 같습니다
- 과거 대화를 기억해야 오늘 회의가 매끄럽게 이어집니다

중요성

- 맥락이 유지되어야 AI가 선생님의 의도를 정확히 이해합니다
- 맥락이 끊기면 엉뚱한 답변이 나올 수 있습니다

컨텍스트 윈도우(Context Window)

AI가 한 번에 머릿속에 담을 수 있는 기억력의 총용량

컨텍스트 윈도우는 AI의 '기억력 한계'를 의미합니다.

마치 책상 위에 펼쳐놓을 수 있는 서류의 양이 정해져 있는 것처럼,
AI도 한 번에 처리할 수 있는 정보량이 제한되어 있습니다.

기억 용량 초과 시 발생하는 문제점

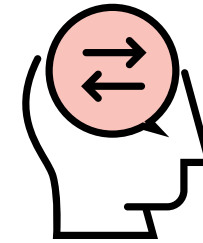
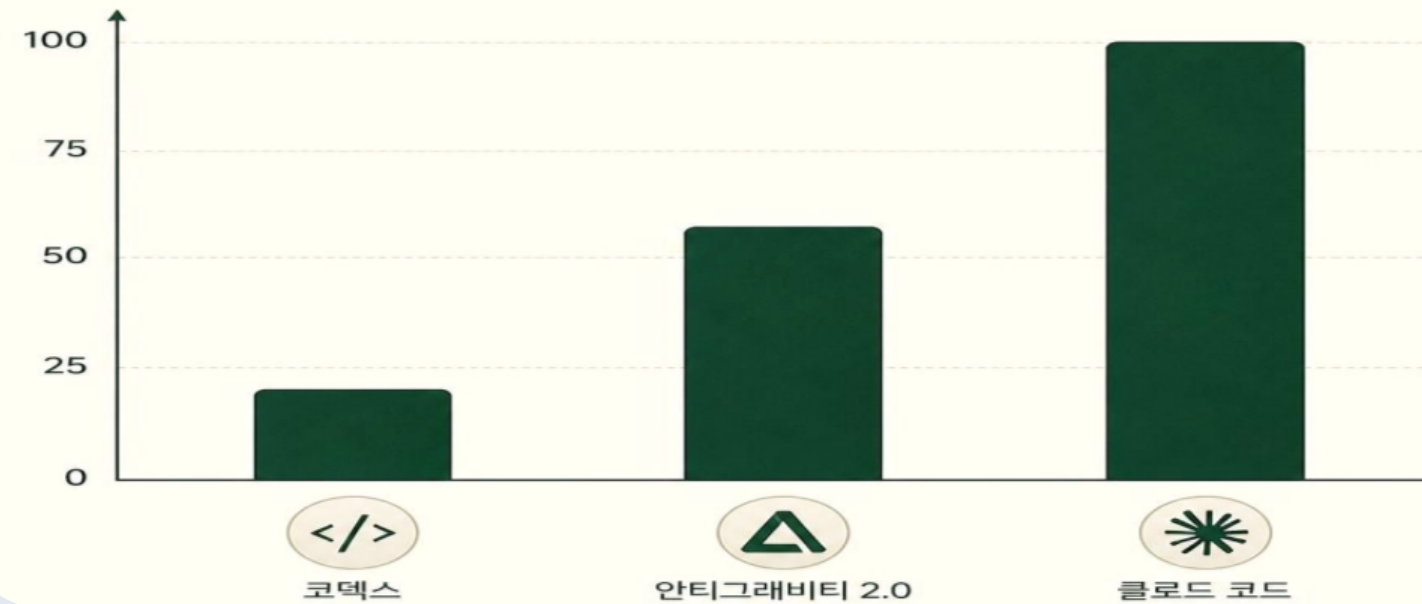
- 기억 용량을 초과하면 AI는 대화 초반에 선생님이 내린 명령이나 규칙을 까맣게 잊어버립니다
- 엉뚱한 코드를 짜기 시작하거나 이전에 합의한 내용을 무시하게 됩니다
- 작은 책상에 책이 넘치는 것처럼, 정보가 넘치면 중요한 내용이 밀려나갑니다



기억력 왕좌의 게임: AI 코딩 도구 3종 비교

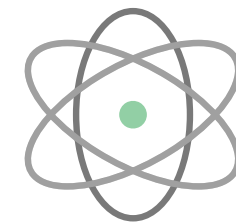
세 가지 AI 도구의 기억력 용량 비교

기억력 용량
(상대적 크기)



안티그래비티 2.0

- 대화 맥락 유지에 최적화
- 교실 도구 개발 시 대화가 중간에 끊기지 않는 안정적인 방 크기 제공



클로드 코드

- 매우 넓은 컨텍스트 윈도우
- 프로젝트 전체 코드를 통째로 집어 넣고 분석하는 데 유리



코덱스

- 초기 코딩 모델로 기억 용량이 상대적으로 작음
- 긴 대화 시 앞 내용을 쉽게 잊어버림

할루시네이션(Hallucination): AI의 거짓말



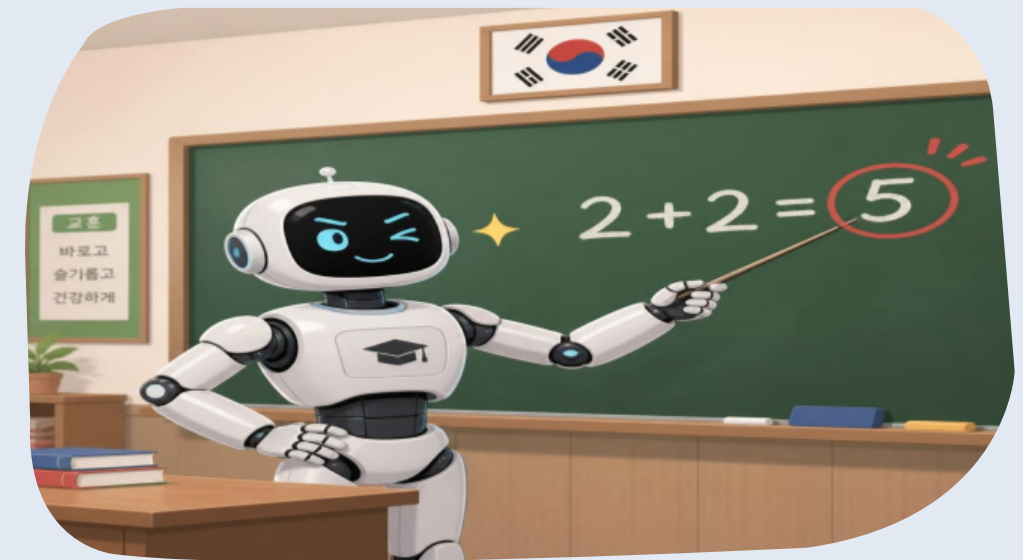
개념

- AI가 모르는 정보나 에러를 마치 진짜인 것처럼 그럴듯하고 당당하게 거짓말로 꾸며내는 환각 현상입니다
- 확신에 찬 어조로 잘못된 정보를 제공하기 때문에 주의가 필요합니다



교실 비유

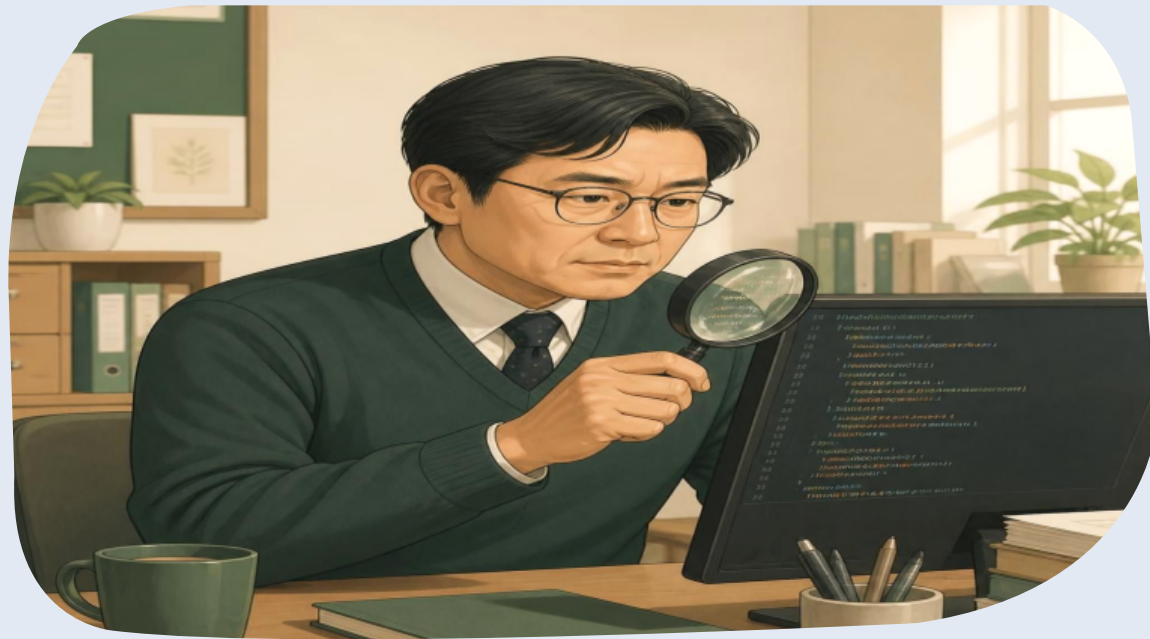
- 숙제 안 한 학생의 교묘한 변명과 같음
- 거짓을 사실처럼 믿게 만드는 현상



시각적 표현

- 피노키오 코처럼 길어진 로봇의 코
- 당당하게 잘못된 공식을 가리키는 로봇의 모습
- AI의 환각 현상을 시각적으로 표현한 일러스트

할루시네이션을 제어하는 교사의 검증 기술



기본 원칙

- AI가 짠 코드가 무조건 정답이라고 믿으면 안 됩니다
- 항상 의심하고 검증하는 자세가 필요합니다
- 특히 중요한 기능이나 학생 정보를 다루는 코드는 반드시 확인해야 합니다
- 선생님이 최종 검수자이자 책임자입니다

방어 프롬프트 전략

- '확실하지 않은 코드가 있거나, 특정 라이브러리가 존재하지 않는다면 추측해서 코딩하지 말고 반드시 에러 가능성을 나에게 먼저 알려줘.'
- 이렇게 미리 방어 프롬프트를 설정하면 AI의 거짓말을 예방할 수 있습니다
- 불확실한 부분은 솔직하게 말하도록 지시하는 것이 핵심입니다



RAG (검색 증강 생성)



개념

- AI 내장 지식에만 의존하지 않음
- 외부 문서를 먼저 검색
- 검색 결과 바탕으로 답변 생성
- PDF, 교육과정 지침서 등 활용

학교 활용 예시

- 2022 개정 교육과정 문서 업로드
- 학교 규정집 PDF 첨부
- 교육청 지침서 참고
- 정확한 정보 기반 답변 생성

교실 비유

- 암기해서 시험 보지 않음
- 교과서와 교육청 지침서 펼쳐둠
- 오픈북 테스트 방식
- 참고 자료 보며 정확하게 답변

효과

- AI의 거짓말(할루시네이션) 예방
- 신뢰할 수 있는 답변 제공
- 학교 문서 기반 정확성 확보
- 교육 현장 맞춤형 결과물

멀티모달: 눈과 귀가 트인 AI



개념

- 텍스트 문자만 이해하는 것이 아님
- 이미지, 음성, 영상 등 동시 처리
- 다양한 형태의 데이터 이해
- 복합적인 입력 방식 지원

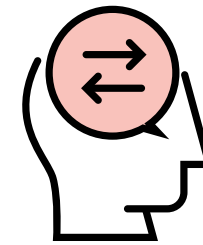
학교 활용법

- 손으로 그린 교실 자리 배치도 촬영
- 스마트폰으로 찍어 업로드
- '이 구조랑 똑같이 디자인해줘' 지시
- 그림이 웹페이지로 변환됨

실전 예시

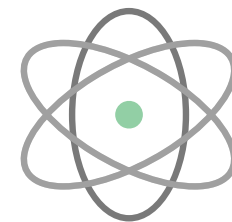
- 학급 게시판 스케치 사진 업로드
- 시간표 손글씨 이미지 인식
- 음성으로 프로그램 수정 지시
- 교실 환경 사진 기반 디자인 생성

1부 정리: 문턱이 낮아진 코딩 세계



바이브코딩의 본질

- 의도·느낌을 한글로 전달
- 엄격한 영문법 불필요
- 대화하듯 편하게 지시
- 프로그램 자동 생성



선생님의 새로운 역할

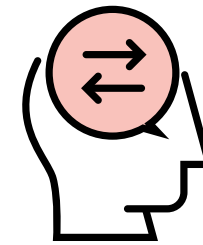
- 코딩 기술자 아님
- AI 비서를 운용하는 기획자
- 교실의 디렉터 역할
- 아이디어와 교육철학이 핵심



안전장치의 필요성

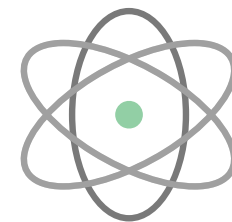
- 하네스로 기능 자동 검증
- 기존 기능 보호는 필수
- 프로그램 누더기 방지
- 안전하게 수정 진행

2부 정리: AI 언어학 마스터



프롬프트 작성법

- AI 비서 업무 지시서
- 구체적·명확하게 작성
- 결과물 퀄리티 향상



토큰 절약 전략

- 핵심 키워드 중심 문장
- 불필요한 수식 제거
- 토큰·비용·속도 절약



컨텍스트 윈도우 관리

- AI의 총기억 용량
- 용량 초과 시 명령 망각
- 규칙 유지로 안정화

2부 정리: AI의 한계와 극복 방법



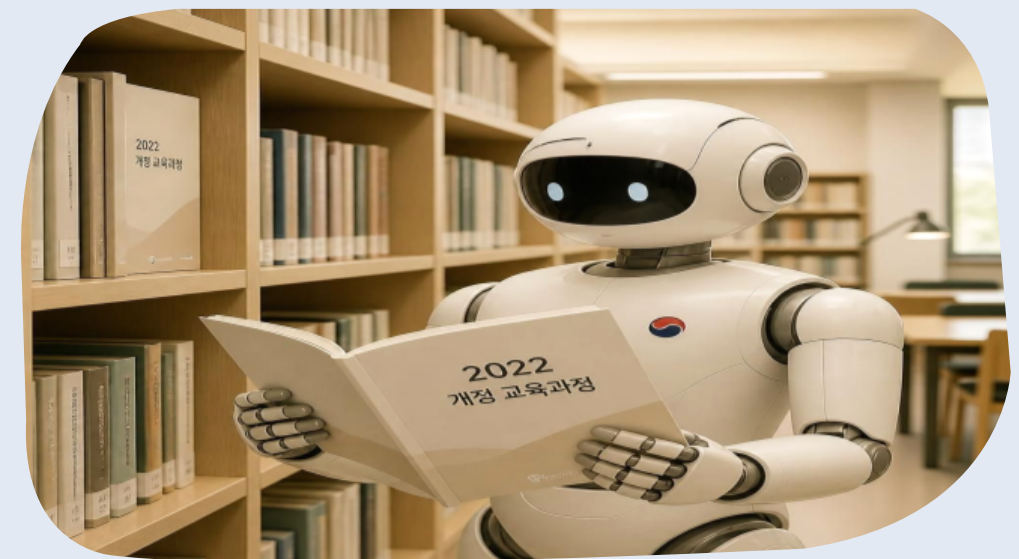
할루시네이션 현상

- AI가 모르는 정보를 꾸며냄
- 그럴듯하게 거짓말 생성
- 당당하게 잘못된 답변 제공
- 숙제 안 해온 학생의 변명과 동일



방어 프롬프트 기법

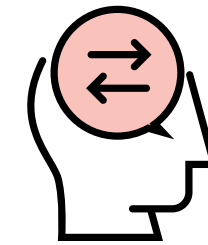
- '확실하지 않으면 추측 금지'
- '에러 가능성 먼저 알려줘'
- 돋보기로 코드 꼼꼼히 검사
- AI 답변 무조건 신뢰 금지



RAG 기술 활용

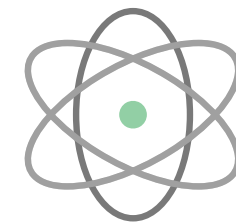
- 외부 문서 먼저 검색 후 답변
- 교과서와 지침서 기반 생성
- 오픈북 테스트 방식 적용
- 거짓말 원천 예방 효과

안티그래비티 2.0의 강점



컨텍스트 윈도우 최적화

- 대화 맥락 유지 특화
- 넓은 기억 용량 제공
- 긴 대화도 명령 기억



안정적인 대화 유지

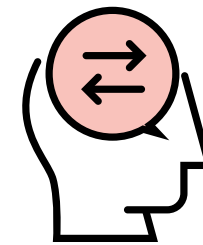
- 대화 중단 없이 진행
- 프로젝트 흐름 전체 파악
- 초반 규칙 끝까지 유지



멀티모달 지원

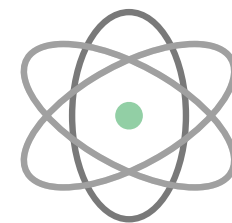
- 이미지·음성 등 입력 지원
- 손그림 업로드 가능
- 복합 지시도 처리 가능

교사용 바이트코딩 성공 체크리스트



명확한 설계도 작성

- PRD로 무엇을 만들지 명확히 정의
- 기능 목록과 우선순위 문서화
- AI에게 구체적인 목표 제시



테스트 하네스 활용

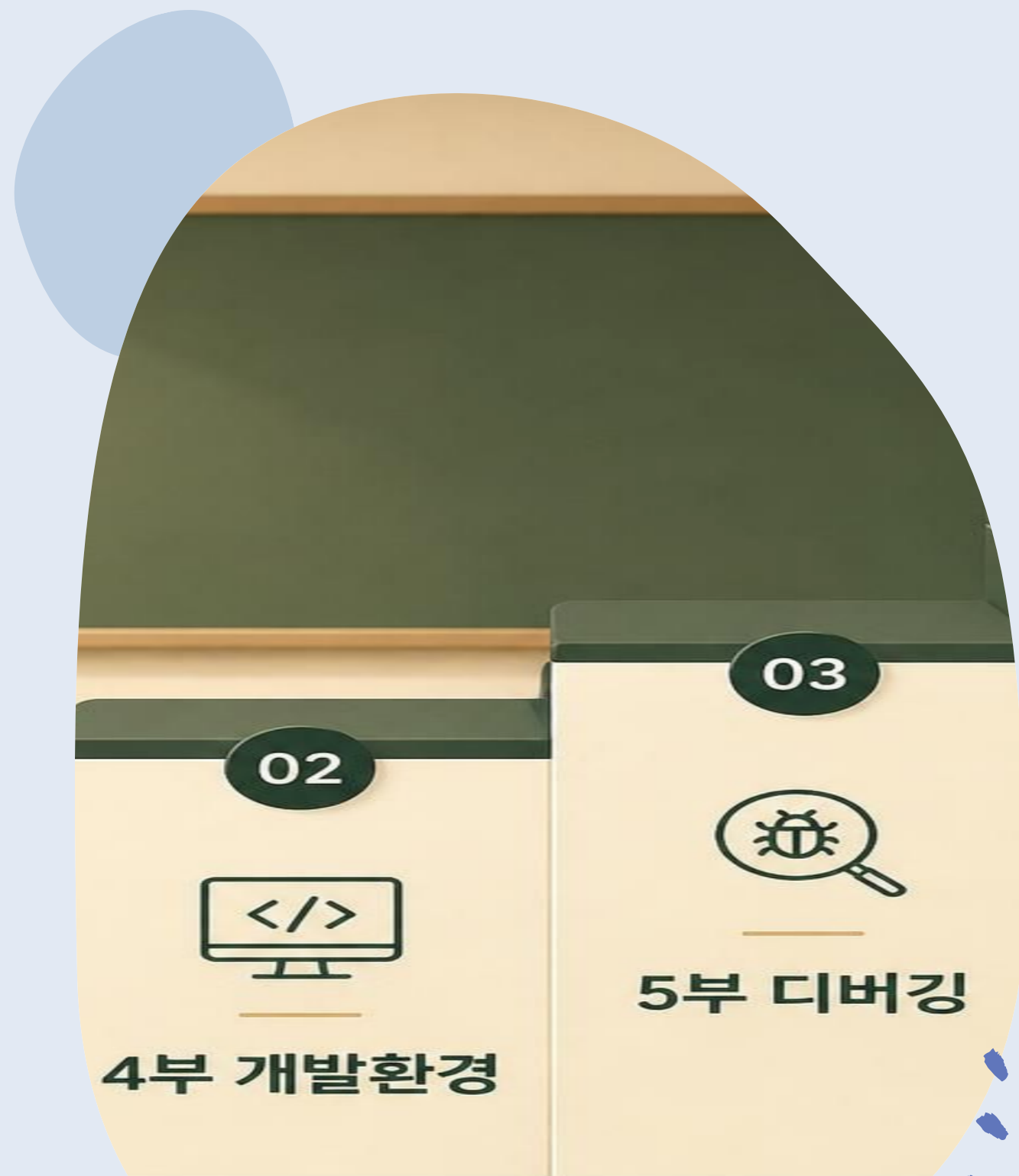
- 기존 기능 보호하며 수정 진행
- 자동 검증으로 누더기 코드 방지
- 1번 기능 정상! 2번 기능 정상! 확인



방어 프롬프트 설정

- 확실하지 않으면 추측 금지 지시
- 에러 가능성 먼저 알려달라고 요청
- AI의 할루시네이션 원천 차단

다음 단계 예고: 3부~6부



3부: 망하지 않는 지름길

- 교사용 PRD 작성법 실습
- 요구사항 명세서 템플릿 제공
- 학교 도구 설계도 완성하기
- 명확한 기능 정의 연습

5부: 프로젝트를 살리는 디버깅

- 디버깅 기술과 오류 해결법
- 철저한 학교 보안 설정 방법
- 학생 정보 보호 체크리스트
- 안전한 프로젝트 운영 가이드

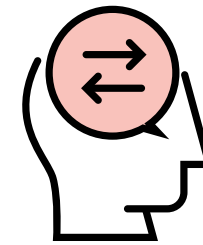
4부: 개발자의 방 구경하기

- 안티그래비티 2.0 개발 환경 탐색
- 코딩 도구의 구조와 작동 원리
- 프로젝트 파일 구조 이해하기
- 실전 개발 환경 체험

6부: 우렁각시 자동 배포

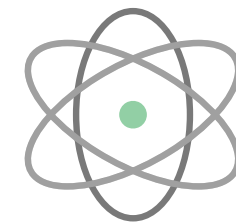
- Git 연동과 버전 관리 기초
- Vercel 자동 배포 설정 방법
- 클릭 한 번으로 업데이트 완료
- 유지보수 자동화 시스템 구축

실습 준비 사항



안티그래비티 2.0

- 계정 생성 및 로그인 완료
- 대화형 AI 코딩 환경 접속
- 프로젝트 생성 준비 완료



Vercel 계정

- Vercel 계정 생성하기
- GitHub 연동 권장 (자동 배포용)
- 무료 플랜으로 시작 가능



학교 도구 아이디어

- 출석부, 성적 관리, 학급 게시판 등
- 내가 만들고 싶은 도구 1개 선정
- 구체적인 기능 목록 작성

교사 커뮤니티와 지원



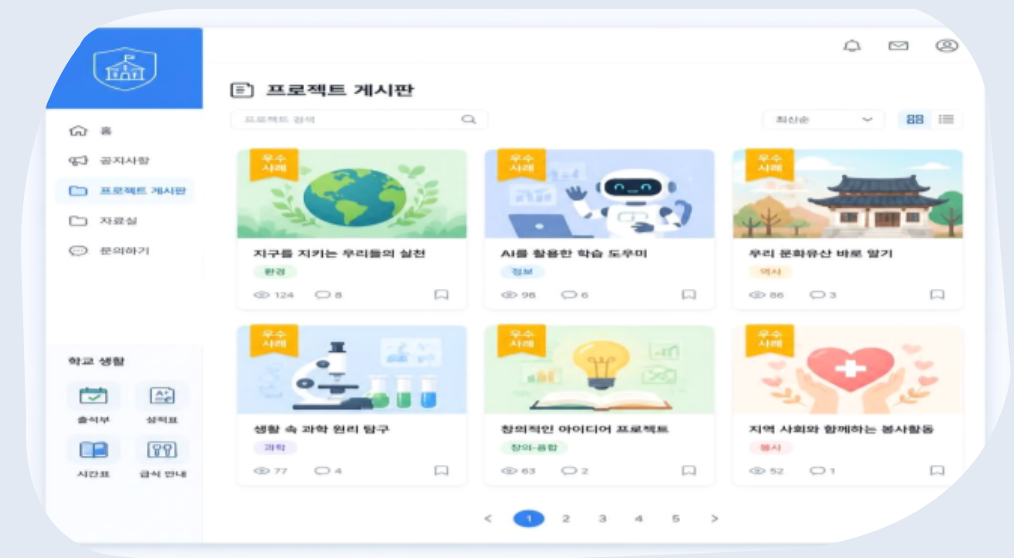
DoRms 커뮤니티

- 교직원 바이브코딩 결과물 공유 플랫폼
- dorms.school 접속
- 전국 선생님들의 프로젝트 구경
- 코드 공유 및 협업 가능



소통 카톡방

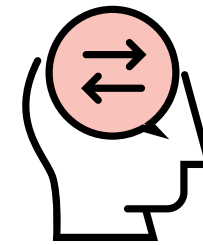
- 실시간 질의응답 지원
- 막히는 부분 즉시 해결
- 선배 교사들의 노하우 공유
- 함께 성장하는 학습 공동체



우수 사례 공유 게시판

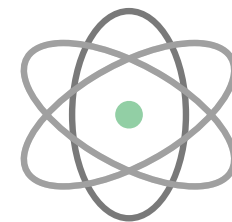
- 다른 선생님들의 프로젝트 참고
- 성공 사례 벤치마킹
- 실전 팁과 노하우 습득
- 아이디어 영감 얻기

바이트코딩으로 교실 혁신 시작하기



진입 장벽 제로

- 코딩 지식 제로, 영어 문법 제로!
- 카카오톡 할 줄 안다면 누구나 가능
- 한글로 대화하듯 프로그램 제작



역할의 재정의

- AI는 손과 발 역할 담당
- 선생님은 교실의 디렉터(감독)
- 기획자로서 아이디어와 철학 제시



실전 도구 활용

- 안티그래비티 2.0으로 개발
- Vercel로 자동 배포 완성
- 나만의 학교 도구 만들기 시작

질의응답 및 다음 시간 안내

오늘 배운 바이트코딩의 본질과 AI 언어학 마스터 내용 중 궁금하신 점이나 이해가 어려운 부분이 있으시면 편하게 질문해 주세요. 프롬프트, 토큰, 컨텍스트, 할루시네이션 등 생소한 용어들도 다시 한번 설명드리겠습니다.

- 다음 시간: 3부 PRD 작성 실습
- 설계도 작성 방법을 단계별 학습
- 실습: 학교 도구 아이디어 1개 구체화
- 필요한 기능·사용자·작동 방식 미리 준비
- 출석부·성적관리·게시판 등 실제 도구 가능
- 다음 시간에 PRD 문서로 완성 실습 진행
- AI와 효과적 소통법을 함께 익힙시다