

# 바이브코딩(Vibe Coding) 실습

---

## 1. 개발 환경 준비

### 1.1 Google Antigravity 설치

1. 다운로드: [Google Antigravity 공식 사이트](#) 에서 운영체제에 맞는 설치 파일을 내려 받아 설치한다.
2. 로그인: 구글 계정으로 로그인 한다. (프리뷰 기간 동안 대부분의 기능 무료 체험)

### 1.2 에이전트 브라우저 설정

1. 에이전트 매니저 실행: 상단 메뉴에서 'Open Agent Manager'를 클릭한다.
2. 브라우저 확장 프로그램 설치: 에이전트가 브라우저를 제어할 수 있도록 안내에 따라 Chrome 전용 확장 프로그램을 설치한다.
3. 권한 승인: 에이전트가 브라우저에서 버튼을 클릭하거나 폼을 입력할 수 있도록 'Setup' 버튼을 눌러 권한을 활성화한다.

### 1.3 워크스페이스 및 모드 설정

1. 샌드박스 폴더 생성 및 워크스페이스에 연결
  - 빈 폴더를 하나 만든다. 예) `$USER_HOME/git/exam01`
  - 방법1) **Open Agent Manager** 선택 > **Open Workspace** 선택 > **Open New Workspace** 선택 > **exam01** 폴더 선택
  - 방법2) **Open Folder** 선택 > **exam01** 폴더 선택
2. 작업 모드 선택
  - Planning Mode: 복잡한 기능을 구현할 때 추천. AI가 먼저 계획서를 쓰고 승인을 요청한다.
  - Fast Mode: 단순한 디자인 수정이나 빠른 프로토타이핑에 적합
3. 모델 선택
  - Gemini 3 Pro (High):
    - 추론 깊이: 최상 (Deep Thinking)
    - 응답 속도: 느림
    - 추천 용도: 아키텍처 설계, 복잡한 로직 구현
  - Gemini 3 Pro (Low):
    - 추론 깊이: 상 (Balanced)
    - 응답 속도: 보통
    - 추천 용도: 일반적인 코딩 및 기능 추가
  - Gemini 3 Pro (Flash):
    - 추론 깊이: 중상 (Fast Agent)
    - 응답 속도: 매우 빠름
    - 추천 용도: 이브 코딩, UI/UX 수정, 빠른 반복 테스트
4. 로컬 실행 환경 준비
  - 언어별 도구 설치: 프로젝트 성격에 따라 Node.js(20 버전 이상 추천)나 Python, JDK 등을 미리 설치해 둔다.

## 실습

### 실습 1 - Todo List 웹앱 만들기

## 기술 명세 미지정 방식

다음 사항에 따라 간단한 Todo List 웹앱을 만들어줘.

- 기능
  - 할 일 목록 보기
  - 할 일 추가
  - 할 일 완료 체크 표시
  - 할 일 삭제

## 기술 명세 지정 방식

다음 사항에 따라 간단한 Todo List 웹앱을 만들어줘.

- 기능
  - 할 일 목록 보기
  - 할 일 추가
  - 할 일 완료 체크 표시
  - 할 일 삭제
- 기술 스택
  - 프론트엔드: Next.js(ver. 16+) + React + TypeScript + Tailwind CSS

## 프로젝트 명세서 생성하기

현재 프로젝트를 분석하여 Todo List 웹앱 개발을 위한 프로젝트 명세서를 작성해줘.

## 프로젝트 다시 생성하기

- 새 프로젝트 폴더를 만든다. 예) `exam03`
- 프로젝트 명세서를 가지고 어떻게 바이브코딩을 해야 하는지 묻는다.

PROJECT\_SPEC.md 파일을 가지고 바이브코딩을 하려고 한다. 어떻게 명령을 내려야 하는가?

- 바이브코딩 예:
  - 1단계: 계획 수립 요청(Planning)

먼저 PROJECT\_SPEC.md를 분석해서 어떤 순서로 개발할지 개발 계획(implementation\_plan.md)을 세워줘.

- 2단계: 순차적 구현 진행 (Execution)

계획이 마음에 들어. 이제 하나씩 코드를 작성해줘.

- 3단계: 검증 및 피드백 (Verification)

다 된 것 같아. 실제로 잘 동작하는지 확인하고, 디자인이 더 세련되게 수정해줘.

## 빈 프로젝트에서 처음부터 만들기

- 프로젝트 생성에 필요한 파일의 템플릿을 요청한다.

새로 프로젝트를 만들려고 한다. 무엇을 알려줘야 하는가? 그 내용을 어떤 문서에 저장해 줘야 하는가?

- 프로젝트의 주요 기능과 주요 기술 스택을 알려주고 프로젝트 명세서를 작성해 달라고 요청한다.

템플릿 파일(`PROJECT_SPEC_TEMPLATE.md`)을 참고하여 다음 사항을 가지고 프로젝트 명세서 파일을 작성해줘.

- 기능
  - 할 일 목록 보기
  - 할 일 추가
  - 할 일 완료 체크 표시
  - 할 일 삭제
- 기술 스택
  - 프론트엔드: `Next.js(ver. 16+)` + `React` + `TypeScript` + `Tailwind CSS`

- 프로젝트 명세서를 가지고 바이브코딩을 명령한다.

`PROJECT_SPEC.md` 파일에 있는 내용대로 프로젝트를 생성해줘.

## 실습 2 - 외부 API 사용하여 날씨 조회 웹앱 만들기

### 날씨 조회 앱 프로젝트 명세서 작성하기

다음 REST API를 활용하여 날씨 조회 웹앱을 만들고 싶다.

다음 내용을 가지고 프로젝트 명세서 파일을 작성해줘.

- 기술
  - 프론트엔드: `Next.js(버전 16 이상)` + `React` + `TypeScript` + `Tailwind CSS`
  - REST API 통신: `Axios`
- 날씨 조회 API: `https://apihub.kma.go.kr/api/typ01/url/fct_afs_dl.php?reg=11B20604&tmfc=&disp=1&help=1&authKey=발급받은인증번호`
- 날씨 조회 API 응답 결과  
`#START7777`  
`#-----`

-----  
# 단기에보 육상 조회 [입력인수형태] [예] ?  
reg=&tmfc1=2013121018&tmfc2=2013121106&disp=0&help=1  
#-----

-----  
# 1. REG\_ID : 예보구역코드  
# 2. TM\_FC : 발표시각(년월일시분,KST)  
# 3. TM\_EF : 발효시각(년월일시분,KST)  
# 4. MOD : 구간 (A01(24시간),A02(12시간))  
# 5. NE : 발효번호  
# 6. STN : 발표관서  
# 7. C : 발표코드  
# 8. MAN\_ID : 예보관ID  
# 9. MAN\_FC : 예보관명  
# 10. W1 : 풍향1(16방위)  
# 11. T : 풍향경향(1:-, 2:후)  
# 12. W2 : 풍향2(16방위)  
# 13. TA : 기온  
# 14. ST : 강수확률(%)  
# 15. SKY : 하늘상태코드 (DB01(맑음),DB02(구름조금),DB03(구름많음),DB04(흐림))  
# 16. PREP : 강수유무코드 (0(없음),1(비),2(비/눈),3(눈),4(눈/비(~'19.6.4.),소나기('19.6.4~)))  
# 17. WF : 예보  
#-----

-----  
# REG\_ID TM\_FC TM\_EF MOD NE STN C MAN\_ID MAN\_FC W1  
T W2 TA ST SKY PREP WF  
11B20604,202509031700,202509031200,A02,0,109,2,imc\*\*\*\*\*,임충  
환,E,1,SE,-99,60,DB04,4,흐리고 가끔 소나기,=  
11B20604,202509031700,202509040000,A02,1,109,2,imc\*\*\*\*\*,임충  
환,NE,1,E,23,70,DB04,1,흐리고 비,=  
11B20604,202509031700,202509041200,A02,2,109,2,imc\*\*\*\*\*,임충  
환,NE,1,E,28,60,DB04,1,흐리고 한때 비,=  
11B20604,202509031700,202509050000,A02,3,109,2,imc\*\*\*\*\*,임충  
환,NE,1,E,22,20,DB03,0,구름많음,=  
11B20604,202509031700,202509051200,A02,4,109,2,imc\*\*\*\*\*,임충  
환,SW,1,W,32,20,DB03,0,구름많음,=  
11B20604,202509031700,202509060000,A02,5,109,2,imc\*\*\*\*\*,임충  
환,SE,1,S,24,60,DB04,1,흐리고 한때 비,=  
11B20604,202509031700,202509061200,A02,6,109,2,imc\*\*\*\*\*,임충  
환,SW,1,W,30,70,DB04,1,흐리고 비,=  
11B20604,202509031700,202509070000,A02,7,109,2,imc\*\*\*\*\*,임충  
환,NW,1,N,22,60,DB04,1,흐리고 한때 비,=  
11B20604,202509031700,202509071200,A02,8,109,2,imc\*\*\*\*\*,임충  
환,W,1,NW,29,30,DB04,0,흐림,=  
#7777END

화면에 출력할 내용은 예보구역코드(REG\_ID), 발효시각(TM\_EF), 기온(TA), 강수확률(ST), 하늘상태(SKY), 강수유무(PREP), 발표시각(TM\_FC) 이다.

#### [프롬프트]

다음 REST API는 예보구역코드 목록을 출력한다.

이 API 응답 결과를 참고하여 날씨 조회하는 화면에서 REG\_ID 대신에 REG\_NAME을 출력해줘.

- API: [https://apihub.kma.go.kr/api/typ01/url/fct\\_shrt\\_reg.php?](https://apihub.kma.go.kr/api/typ01/url/fct_shrt_reg.php?tmfc=0&authKey=발급받은인증번호)

tmfc=0&authKey=발급받은인증번호

- API 응답 결과 일부 예:

#START7777

#-----

# 단기에보구역 조회

#-----

# 1. REG\_ID : 예보구역코드

# 2. TM\_ST : 시작시각(년월일시분,KST)

# 3. TM\_ED : 종료시각(년월일시분,KST)

# 4. REG\_SP : 특성 (A:육상광역,B:육상국지,C:도시,D:산악,E:고속도로,H:해상광역,I:해상국지,J:연안바다,K:해수욕장,L:연안항로,M:먼항로,P:산악)

# 5. REG\_NAME : 예보구역명

#-----

#	REG_ID	TM_ST	TM_ED	REG_SP	REG_NAME
11000000	199001010000	210012310000	A	육상	
11A00000	201501221100	210012310000	A	서해5도	
11A00100	199001010000	210012310000	B	서해5도	
11A00101	201610131800	210012310000	C	백령도	
11A00102	199001010000	210012310000	C	연평도	
11A00103	199001010000	210012310000	C	소청도	

#### 프로젝트 생성하기

PROJECT\_SPEC.md 파일을 참고하여 날씨 조회 웹앱 프로젝트를 만들어줘.

#### 인코딩 변환하기(필요할 시)

예보구역코드 REST API의 응답 결과는 EUC-KR 로 인코딩 되어 있다. 이를 UTF-8로 변환해줘.

#### 실습 3 - 쇼핑몰 웹 애플리케이션 만들기

##### 쇼핑몰 백엔드 웹 애플리케이션 만들기 (SpringBoot 기반)

다음 사항에 따라 간단한 쇼핑몰 백엔드 웹 애플리케이션을 만들어줘.

- 기능

- 상품 목록 조회
- 상품 상세 조회
- 상품 추가

- 상품 수정
- 상품 삭제
- 기술 스택
  - 백엔드: Spring Boot + Java + JPA + H2 Database
  - API: RESTful API
  - 빌드 도구: Gradle

## Swagger 추가하기

프로젝트에 Swagger 를 추가해줘.

## 쇼핑몰 프론트엔드 웹 애플리케이션 만들기 (Next.js 기반)

다음 사항에 따라 간단한 쇼핑몰 프론트엔드 웹 애플리케이션을 만들어줘.

- 기능
  - 상품 목록 보기
  - 상품 상세 보기
  - 상품 추가
  - 상품 수정
  - 상품 삭제
- 기술 스택
  - 프론트엔드: Next.js (버전 16 이상) + React + TypeScript + Tailwind CSS
  - REST API 통신: Axios
  - REST API 스웨거 문서: <http://localhost:8080/v3/api-docs>