

- [팀 개발을 위한 Git, GitHub 시작하기](#)
- [처음 시작하는 FastAPI](#)

2주 과정 주요 내용

| 주차 | 목표 | 핵심 기술 | 최종 평가 방식 |
|-----|------------------------|----------------------------------|-----------------------------------|
| 1주차 | 개인 개발 역량 강화 및 개발 환경 구축 | Git 기본, FastAPI 기초, 개인 프로젝트 | 개인 프로젝트 시연 및 코드 리뷰 |
| 2주차 | 팀 협업 프로세스 및 실무 능력 체득 | Git 협업 워크플로우, FastAPI 심화, 팀 프로젝트 | 팀 프로젝트 발표 (작동하는 프로그램) 및 협업 과정 세미나 |

1 주차

| 날짜 | 주제 | 상세 내용 | 비고 |
|-----|---------------------|---|--|
| 1일차 | Git 기본기 다지기 | <ul style="list-style-type: none"> - 버전 관리(VCS)의 필요성 이해 - Git 설치 및 환경 설정 - 로컬 저장소: <code>init</code>, <code>add</code>, <code>commit</code>, <code>log</code> - 원격 저장소: <code>clone</code>, <code>remote</code>, <code>push</code>, <code>pull</code> | 각 명령어의 역할과 사용법을 설명할 수 있는가? |
| 2일차 | FastAPI 첫걸음 & 브랜치 | <ul style="list-style-type: none"> - FastAPI 및 <code>uvicorn</code> 설치 - "Hello, World!" API 엔드포인트 만들기 - Git 브랜치 전략의 중요성 - <code>branch</code>, <code>checkout</code>, <code>merge</code> (개인 작업 흐름) | API가 정상적으로 실행되는가? Commit 히스토리 리뷰: 브랜치를 목적에 맞게 사용했는가? |
| 3일차 | FastAPI 핵심: 데이터 입출력 | <ul style="list-style-type: none"> - Path, Query 파라미터 - Pydantic을 이용한 Request Body 및 Response Model 정의 - [과제] 개인 프로젝트 (e.g., 나만의 맛집 리스트 API) 시작 | Pydantic 모델을 적절하게 정의했는가? 작동하는 프로그램: POST 요청으로 데이터를 생성할 수 있는가? |
| 4일차 | API 기능 확장 | <ul style="list-style-type: none"> - <code>PUT</code>, <code>DELETE</code>를 이용한 데이터 수정/삭제 - <code>HTTPException</code>을 이용한 기 | 작동하는 프로그램: 특정 데이터를 조회, 수정, 삭제할 수 있는가? (CRUD 완성) |

| 날짜 | 주제 | 상세 내용 | 비고 |
|-----|----------------|---|---|
| | | 본 에러 처리 - [과제] 개인 프로젝트에 수정/삭제 기능 추가 | |
| 5일차 | 1주차 회고 및 협업 준비 | - 각자 자신의 개인 프로젝트 코드와 작동 방식 시연 - 1주차 학습 내용 Q&A - Git 협업 워크플로우 소개 (Pull Request 개념) | 개인 프로젝트 시연 (세미나): 본인이 만든 API의 구조와 코드를 설명하는가? 코드 리뷰: 코드 가독성, 변수명 등 개선점 피드백 |

2 주차

2주차의 목표는 실제 현업과 가장 유사한 환경에서 두 개발자가 소통하며 하나의 결과물을 만들어내는 경험을 하는 것입니다.

| 날짜 | 주제 | 상세 내용 | 평가 방법 |
|-------|---------------------|--|--|
| 6일차 | 팀 프로젝트 기획 및 설정 | - [프로젝트] 간단한 블로그 API, 설문조사 API 등 주제 선정 - 기능 명세 확정 및 역할 분담 - GitHub Issue에 할 일 (Task) 등록 - 공유 Repository 설정 및 GitHub Flow 협업 전략 합의 | 문서 기반 평가: 프로젝트 기획 및 이슈 등록 내용이 명확하고 체계적인가? |
| 7.5일차 | 협업 사이클 1: PR과 코드 리뷰 | - 각자 맡은 기능 브랜치에서 개발 시작 - Pull Request(PR) 생성 및 동료에게 코드 리뷰 요청 - 코드 리뷰 피드백 반영 및 Merge - <i>이 과정을 계속 반복</i> | PR 퀄리티: PR 제목과 내용이 명확한가? 코드 리뷰 퀄리티: 동료에게 건설적인 피드백을 제공하는가? |
| 8일차 | 협업 사이클 2: 충돌 해결 | - 의도적으로 Merge Conflict(충돌) 상황 만들기 - 충돌의 원인 파악 및 해결 과정 학습 - FastAPI 심화: Depends 를 이용한 의존성 주입 | 문제 해결 능력: 충돌 상황에서 동료와 소통하며 문제를 해결하는가? |
| 9일차 | 프로젝트 마무리 및 발표 준비 | - 남은 기능 구현 및 버그 수정 - FastAPI 자동 API 문서 | API 문서 완성도: 자동 생성된 문서가 명확하고 이해하기 쉬운가? |

| 날짜 | 주제 | 상세 내용 | 평가 방법 |
|------|------------|---|--|
| | | (/docs) 정리 - 최종 발표 및 시연 준비 | |
| 10일차 | 최종 발표 및 회고 | <ul style="list-style-type: none"> - 팀 프로젝트 최종 시연 및 발표 - 프로젝트 구조, 사용한 기술, 협업 과정 공유 - 프로젝트 회고: KPT(Keep, Problem, Try) 방식 | <p>작동하는 프로그램 (최종 평가)</p> <p>문서 기반 세미나 (최종 평가)</p> <p>협업 과정 회고: 2주간의 경험을 통해 무엇을 배우고 느꼈는지 공유</p> |