**Sandwich 2일차**

**1. Slack**

- 대화에 대해 댓글로 엮이는 방식

- 서로의 대화에 대한 피드백 잘하기!(프사 바꾸기)

- Channels에서 class와 notice에 가입

**2. Git**

- Source code의 버전 관리(여러 프로그램이 있지만 95%로 점유율 1등)

- 리누스 토발즈

- Repository : 어떠한 폴더를 지정하면 거기서 버전관리를 시작

- mkdir 폴더명 : 폴더 생성

- cd directory명 : 이동

- git init : repository할 폴더 안에서진행. 이전 사항들과 다른 것을 저장해줌)

- touch 파일명(sandwich.txt) : 파일 생성

- vim(또는 cat) 파일명(sandwich.txt) : 편집

- Stage : 변경사항 저장해 줄 것을 등록(Commit 대상으로 등록)

- Commit : 체크포인트 생성(끊고 저장) – Git 버전 관리의 핵심

- git add 파일명 : stage에 등록

- git add –A : 모두 다 stage에 등록(보통 이렇게 사용)

- git commit –m “메모메모” : commit, commit명을 메모메모에

- git log : commit 한 list를 보여줌

- branch : Repository 안의 repository 🡺 협업이 편리해짐!

Branch를 새로 만드는 행위는, 그 repository를 하나 복사해서 새롭게 만들고, 그런

새로운 환경에서 작업하는 것.

- git branch : 현재 branch 확인(처음에는 다 master)

- git checkout –b apple : apple이라는 branch 생성 및 이동

- git checkout apple : apple이라는 branch로 이동

- git merge apple : 현재 있는 branch 안에서 apple이라는 branch 합침

🡺 version이 높은 것으로 합쳐짐

version이 높은 것은 commit의 개수로 따짐(같으면 둘의 내용 합침)

**3. 학회의 방향성**

- 10개의 주제를 스스로 공부할 수 있을 만큼 알아가는 것

- 공모전(소프트웨어 경진대회)

- 기업 탐방(Naver, Microsoft 등)

- Tech blog – 공용의 것(누군가 날마다 정리해서 올림)과 개인의 것

🡺 20개 정도의 크고 작은 프로젝트의 체계화+저장 / 지식의 공유

- 숙제 : 프로젝트 1개 / 주 1회(평균 6시간 소요)

🡺 진행해 나가면서 조정!

- 피드백, 질문을 피하지 말기

🡺 수업 후 질문은 단체 톡으로만 가능.(원한다면 익명 가능)

- 개발 환경 : Code(Microsoft)로 결정

**4. 1일차 숙제**

- Git 다운로드 (git-scm.org)

- Git hub ID 만들기

- Source Tree 프로그램 다운로드

- VS Code 프로그램 다운로드

- 인터넷에 Git가 뭔지 찾아보고 알아보기(예습)