**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. Git hub & Source Tree**

- new repository (online의 repository)

1. FTP같은 것으로 direct로 수정

2. 그대로 local로 포크해온 것을 이용해 수정한 뒤 online에 merge

- Source Tree : git를 더 편리하게 이용할 수 있게 도와줌!(완전 편함)

- 새 저장소-url로 fork해서 받아오기

- 변경사항 -> 체크가 stage 등록과 같음 -> commit

- push : 원격으로(온라인 github로) 업데이트

- pull : 원격에서 가져오기

- patch : 내 것으로 merge해서 만들기 전에 내용들을 가져와보는 것. 간보기 작업.

**4. 2일차 과제**

- 과제는 2일 후 자정까지(23일 자정)

- Git, Github, SourceTree 사용법 올리기(최소 commit 5번)