**간단한 사용법**

1. 터미널(윈도우에서는 cmd인가요?) 에서 원하는 폴더로 들어간다.
2. 최초 설정(이미 되어있을 수도 있음)

git config --global user.name "Wanbae Park"

git config --global user.email [wbpark91@gmail.com](mailto:wbpark91@gmail.com)

1. git init을 입력(해당 폴더가 로컬 저장소가 됨)
2. git clone <https://github.com/team7216/FE531> : 웹 저장소가 로컬 저장소로 복사됨
3. git add “filename.py” 파일을 commit하기 위한 stage에 올림
4. git commit: 파일을 commit

Commit: git이 파일 변경내용을 저장하게 됨. 한 번 commit하면 나중에 오류가 나도 무조건 되돌릴 수 있음.

1. git branch {branch이름} : 새로운 branch를 만듬

각 branch는 독립적임. 즉, master branch에서 작업하는 파일은 다른 branch에 영향을 주지 않음. 한 개의 파일로 두세가지의 작업을 동시에 할 수 있음. 따라서 오류가 나도 쉽게 되돌릴 수 있음.

1. git remote –v : 웹 저장소와 연결되어있는 상태를 확인

origin https://github.com/team7216/FE531.git (fetch)

origin https://github.com/team7216/FE531.git (push)

origin은 웹 repository의 별칭과 같음

1. git push origin {branch 이름}: 현재 로컬 저장소에 commit된 파일을 웹 저장소에 업로드 (push 권한이 부여되어있어야 가능)
2. git pull origin {branch 이름}: 웹 저장소에 있는 파일을 로컬 저장소로 불러와 병합

(혹은 git fetch origin {branch 이름}, git merge를 사용할 수 있음)

1. git diff {branch 이름} {branch 이름} : 두 branch 사이의 차이를 보여줌

ex: git diff master origin/master : 로컬 저장소의 master branch와 웹 저장소의 master 브랜치 간의 차이

12. git checkout branch이름 브랜치 바꾸는 거

13. –a all

14. –m 블라블라 comment 그냥 바로 쓰는 거

**주의사항**

1. 로컬 브랜치와 리모트 브랜치의 이름은 같게 하는 것이 좋음. (그래야 에러 없이 편하게 push가 가능함)
2. 처음 push 할 때 git push **–u** origin [branch name]으로 push하는 것이 좋음 (git push –u : upstream을 지정. 한 번 지정되면 “git push(또는 pull”라는 명령어만으로도 자동적으로 지정된 리모트 브랜치로 push(pull)이 됨