**깃허브**

- 코딩 외 실무적인 문제를 해결함

- 포트폴리오를 올릴 장소

- 소스코드를 깃허브로 공유

1. 내 소스코드를 저장

2. 소스코드 공유

3. 협업하는 공간

git 설치

git add . 어떤 파일을 추가 해줄지 찾아주겠다

.(점) 모든 파일을 추가해주겠다 싫으면 index.html 하면됨

git status: 너에게서 새로운 파일이 있다는걸 알려줌

<히스토리 만들기> : git commit -m "first commit"

git commit: 히스토리를 만드는 것.

<연결고리 보기>: git remote add origin

<연결고리 확인>: git remote -v로 연결고리 확인

<보내기> git push origin master

backlog.com/git-tutorial/kr/stepup/stepup1\_1.html

init: 현재 디렉토리에다가 내가 작업을 진행하겠다는 것을 깃에게 알려줌

브랜치: 개발자들이 동시에 다양한 작업을 할 수 있게 만들어주는 기능

깃의 저장소를 초기화했다고 나옴

생성된 정보는 .git에 저장이 됨

git init: 프로젝트 버전관리를 하겠다고 깃에게 알려줌

파일을 만들어서 버전관리 해보기

파일: vim(어디서나 쓸 수 있는데 쓰기 어려움)]

f1.txt를 만들고 숫자 1을 쓸 것.

vim f1.txt

n, r이라도 아무것도 입력이 안될 것 -> 입력모드가 아니기 때문.

이럴 때는 알파벳 i를 누르면 insert가 되면서 입력할 수 있는 상태가 됨

입력 히주고 esc를 누르면 입력 불가능. 다시 입력하려면 i를 눌러줘야함

콜론: 작성한 내용을 저장하고 싶을 때+프로그램 종료하고 싶을 때

w: write q: quit -> :wq 해주기

cat f1.txt: f1.txt 읽기

status: git status: f1.txt라는 파일이 untrackted 추적되고 있지 않다. 버전관리가 도고 있는 디렉토리에 존재하지만 버전관기를 시작하라 할 때까지 얘를 무시함

이럴 때는 git add f1.txt를 해주면 버전관리 시작함

깃이 f1.txt를 새 파일이라고 인식함

새로 추가할 파일은 add 명령어를 통해 해줘야됨.

버전 = 의미있는 변화

ex) 로직을 만들다 만 상태 -> 작업하다가 나가고 만 상태(단위.작업이 있으면 그게 완결된 상태가 버전임)

git commit: 파일 변경 이유 등의 버전 메세지 작성

f1이 새로운 버전이 됨

cp f1.txt f2.txt -> f1.txt 복사하기

깃이 왜 add를 포함하고 있나?

많은 작업들을 담은 거대한 버전을 하나 만들 때(commit의 시기를 놓쳤을 때) add를 통해서 commit 할 수 있음