github 유료 , 매우 대중적

gitlab 은 무료 노과금 --- 둘중하나 사용

원격 저장소와 연결

1. 이미 존재하는 저장소를 원격 저장소에 연결하는 경우
   1. git remote add origin [깃허브주소]
      1. 원격저장소를 여러 개 쓸수 있으니 origin이라는 별명을 붙인다.
   2. git push –set-upstream origin master
      1. 기본 push의 원격저장소를 지정해주는 과정
2. 원격저장소를 복제해서 지역저장소를 만드는 경우
   1. git clone [주소]
      1. 이미 존재하는 원격저장소 내용을 복사해서 git init없이 명령어 친 장소에 저장소를 만들어줌
   2. git clone [addr] [folder name]
      1. 명시한 폴더를 만들면서 그곳에 복제

push 명령어

원래는 git push origin master # 현재 브랜치를 오리진 리모트 서버의 마스터 브랜치로 푸쉬 해줘 라는 뜻이다.  
이걸 set upstream 옵션줘서 자동으로 바꾸어 git push 처럼 사용하는것이다.

협업

* 동시에 작업할 때, git push가 거절 당하면 remote의 내용이 바뀐경우일수 있다.
  + git pull로 일단 바낀 내용을 가져온다.
  + if 충돌
    - 도구 쓰고 싶으면 config에 있는 mergetool같은 명령어 사용
    - 아니면 수동으로 하나하나 수정
  + 충돌을 해결하고 add 와 commit을 해서 merge완료
  + 이제 git push

pull 과 fetch의 차이점

* git pull = git fetch + git merge FETCH\_HEAD
* git fetch을 하면 병합없이 remote의 앞선 기록을 가져온다(git log로 확인)
  + 그다음에 앞에 있는 커밋해쉬(ex: origin/master)와 merge을 하면 된다.
    - git merge FETCH\_HEAD로 해도됨: 최근의 업데이트한 버전과 merge을 한다.
  + 또는 git pull 해도 되고(git fetch 두번 하고 merge하는 것이니)
* git fetch는 git pull을 더더욱 세심하게 하거나 쓰기권한이 없는경우 fetch을 통해 local에서 병합하고 잘돌아가는지 확인할수 있다.

git rebase : merge랑 결과는 비슷한데 과정이 다르고 버전관리모습이 깔끔

* git rabase br: 해당 br을 조상으로 해서 이어 붙인다.
  + 즉, 1-2-3-4(origin/master)
  + |-5-6-7(HEAD-master) 일경우 git rebase origin/master
  + -> 1-2-3-4-5’-6’-7’(HEAD-master) 의 결과가 나온다.
  + |-origin/master // 중요한건 이대로 push해도 깃의 버전은 건들지 않았기 때문에 push해도 된다는점이다.