0515 - 섹션5 Github 협업

2023년 5월 15일 월요일

오후 10:31

* + README.md파일 만들면 브랜치가 생성됨
    1. git init
    2. 파일생성 (Readme.md)
    3. add.
    4. comit -> 브랜치생성
  + main: 배포 브랜치
  + dev: 개발 브랜치
  + git merge --no-off (merge할 브랜치명) : fast-forward 머지에서 로그를 남김
  + git rebase -i HEAD~2 : 2개의 로그를 리베이스
  + git tag (태그명) : 태그를 다는 명령
  + git tag -n : 태그를 볼 수 있음(태그명, 어디에 달렸는지 확인 가능, git tag는 태그명만 나옴)
  + git push origin main : 깃허브에 푸쉬(태그 업로드 X)
  + 태그도 같이 업로드 : git push --tags origin main

main에 올릴 때 꼭! --tags 옵션 설정해 줘야 태그 업로드됨

* + 메인은 제품이 완료되었을 때 푸시
  + dev도 푸시해두는 게 좋다

* git push --all : 모든 브랜치 푸시(최초 푸시에서 이렇게 하면 편함)

브랜치가 어디에 있던 상관없이 다 올라감

* 레퍼지토리의 settings > branches에서 rules을 추가함으로 보안강화,

Require a pull request before merging 이 레퍼지토리에 승인된 사람만 push할 수 있도록 할 수 있음

* PR요청 : push/request 요청, 팀원이 새로운 브랜치를 만들어서 올리고 merge요청

* Webhooks : 실시간 모니터링하는 웹 페이지를 만들 수 있음(레퍼지토리 세팅에서)
* git push --delete origin join\_topic : (팀원이 자신이 만든) 브랜치 제거(그렇지만 굳이 로컬의 브랜치까지 삭제할 필요 없음)
* rebase후에는 push가 되지 않는다 => 커밋상태가 달라서

git에서 pull하고 하라고 하지만 그러면 rebase한 것 원래대로 돌아가서 안됨! 이 때는 강제로 push

강제 푸시 명령어 : git push -f origin login\_topic

* rebase는 자신의 브랜치에서만 해야한다!!!
* git push --tags origin main : 태그를 함께 메인 브랜치에 푸시