

2024-04-03 1. GIT

! 현행 문서: 1. GIT

- 1 교재
- 2 Definition of Done
 - 2.1 시작하기
 - 2.2 Git의 기초
 - 2.3 Git 브랜치
 - 2.4 분산 환경에서의 Git
 - 2.5 Git 도구

1 교재

Pro Git 2nd Edition

GitHub- 리포지토리 포크

GitHub- 프로젝트에 기여

트렁크 기반 개발

2 Definition of Done

• 필수 : 2.1 시작하기, 2.2 Git의 기초, 2.3 Git 브랜치

2.1 시작하기

- 교재
 - 시작하기
- 버전 관리를 왜 사용하는지 알고 있다.
- Git의 개발 목표를 알고 있다.
- Git CLI를 사용할 수 있는 환경을 갖추었다. ([Dev Environment Settings #4](#) 참고)
- Git의 글로벌 설정을 아래와 같이 수행하였다.
 - `user.name`: GitHub의 Username
 - `user.email`: GitHub에 등록한 Email
 - `core.editor`: vim(또는 원하는 텍스트 에디터)

2.2 Git의 기초

- 교재
 - Git의 기초
- `git clone` 명령어를 사용할 수 있다.
- `git status` 명령어를 사용할 수 있다.
- 변경된 파일을 Stage 할 수 있다.
- Staged 파일을 Unstage 할 수 있다.
- `.gitignore` 파일의 역할을 이해한다.
- Staged와 Unstaged의 변경 내용을 볼 수 있다.
- 커밋을 만들 수 있다.
- 커밋 히스토리를 조회할 수 있다.
- 커밋 메시지를 변경할 수 있다.
- 리모트 저장소를 확인하고 추가나 변경, 삭제할 수 있다.
- 태그를 만들고 확인할 수 있다.
- 작성한 태그를 리모트 저장소에 공유할 수 있다.
- 태그를 Checkout 할 수 있다.

2.3 Git 브랜치

- 교재
 - Git 브랜치
- 브랜치가 무엇인지 이해할 수 있다.
- 브랜치를 이동할 수 있다.
- 서로 다른 브랜치를 Merge 할 수 있다.
- 브랜치 Merge 시 충돌을 해결할 수 있다.
- 브랜치를 조회하거나 삭제할 수 있다.
- Long-Running(Long-Live) 브랜치와 Topic 브랜치의 차이를 이해할 수 있다.
- 리모트 저장소의 브랜치를 조회할 수 있다.
- 브랜치를 리모트 저장소에 Push 할 수 있다.
- 현재 브랜치가 리모트 저장소의 브랜치를 추적하게 할 수 있다.

- 리모트 저장소의 브랜치를 삭제할 수 있다.
- 현재 브랜치에 변경을 주지 않고 리모트 저장소의 최신 브랜치를 받을 수 있다.
- 추적 중인 리모트 저장소의 최신 브랜치를 현재 브랜치에 반영할 수 있다.
- Rebase가 무엇인지 이해할 수 있다.
- 이미 공개 저장소에 Push 한 커밋을 Rebase 하면 안 되는지 말할 수 있다.
- Rebase와 Merge의 차이를 말할 수 있다.

2.4 분산 환경에서의 Git

- 교재
 - [분산 환경에서의 Git](#)
- Git Workflow가 무엇인지 알 수 있다.
- Tim Pope가 작성한 커밋 메시지의 템플릿을 확인했다.
- GitHub을 통해 리포지토리를 Fork할 수 있다.
- GitHub을 통해 Pull Request를 작성하여 프로젝트에 기여할 수 있다.
- 트렁크 기반 개발 방식을 이해한다.

2.5 Git 도구

- 교재
 - [Git 도구](#)
- reflog를 확인할 수 있다.
- 범위로 커밋을 가리키는 방법을 알고 있다.
- stash를 사용할 수 있다.
- git rebase interactive 모드를 이용할 수 있다.
- reset을 할 수 있다.
- 충돌 상황에서 our와 their의 차이를 이해할 수 있다.