

04

Git의 다양한 기법

Git

.gitignore

- Git 의 root 디렉토리에 위치하는 설정 파일
 - Git Repository 나 Staging Area 에 추가되지 말아야 하는 폴더 혹은 파일을 정의하는 파일
 - .gitignore 에 설정한 파일은 Staging Area 에 올라가지 않음으로 Tracking 되지 않는다.
 - git status 로도 보이지 않는다.
-
- [폴더 이름]/ : 폴더 전체 무시
 - *.[확장자] : 특정 확장자 파일 무시
 - [폴더 이름]/[파일 이름] : 특정 폴더의 특정 파일 무시
-
- 이미 Staging Area 혹은 Repos 에 추가된 파일을 .gitignore 로 설정하려면 먼저 파일을 제거해야 한다.

Tag

- commit 은 해시코드로 식별된다.
- 개발하면서 수 많은 commit 을 한다.
- 태그는 특정 위치의 commit 에 별칭을 추가하는 개념이다.
- Lightweight 태그 : 태그 이름만 추가. 일반적으로 버전명을 많이 이용한다.
- Annotated 태그 : 태그 이름이외에 메시지 추가 가능
- git tag <name>
- 새로운 Lightweight 태그 추가

```
kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/test2 (main)
$ git tag v1.0.0

kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/test2 (main)
$ git log --oneline
1c2146c (HEAD -> main, tag: v1.0.0) 2
1485b0d 1
```

Tag

- `git tag -a <name>`
- 새로운 Annotated 태그 추가
- `git tag`
- 태그 목록 출력
- `git tag -n`
- 태그 목록과 메시지 출력

```
kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/test8 (main)
$ git tag -n
fourTag          four tag message
twoTag           two commit
```

Tag

- `git tag -d <tag>`
- 태그 삭제

- `git tag <tag> <hash>`
- 특정 commit 에 태그 추가

- `git reset --hard <tag>`
- 태그를 이용한 reset

config

- **Git 의 설정 파일**

- system, global, local 로 구분
- system -> global -> local 순으로 인지

- **/etc/gitconfig**

- System 영역의 설정 파일이다.
- 수정하려면 시스템 관리자 권한이 필요
- 모든 사용자와 모든 저장소에 적용되는 설정

- **~/.gitconfig**

- 특정 사용자의 모든 저장소

- **.git/config**

- 특정 저장소

config

- `git config --local -l`
- `git config --global -l`
- `git config --system -l`
- 설정 정보 조회
- `git config --local user.name "kkang"`
- 설정 정보 추가

A screenshot of a terminal window titled 'MINGW64:/c/study-git/test8'. The terminal displays the output of the 'git config' command, showing configuration for the core, remote, and user sections. The output is as follows:

```
[core]
  repositoryformatversion = 0
  filemode = false
  bare = false
  logallrefupdates = true
  symlinks = false
  ignorecase = true
[remote "test1"]
  url = https://github.com/kkang104test/test1.git
  fetch = +refs/heads/*:refs/remotes/test1/*
[user]
  name = kkang104test
  email = kkang104@naver.com
```

config

- `core.autocrlf`
- 플랫폼마다 줄바꿈을 표현하는 문자열이 다름
- LF(Line-Feed, Mac Linux Unix, `\n`)
- CR(Carriage-Return, 초기 Mac, `\r`)
- CRLF(Carriage-Return+Line-Feed, Windows, `\r\n`)
- 실제 코드적으로는 변경되지 않았지만 변경된 것으로 인지될 가능성
- merge 시에 문제의 소지
- Git 이 줄바꿈을 어떻게 처리해야 하는지에 대한 설정

config

- `core.autocrlf=false`
 - 줄바꿈 자동 변경기능 끄
- `core.autocrlf=true`
 - text file 을 Git Database 에 추가할때 CRLF->LF 로 변경
 - Git Database 의 내용을 text file 로 가져올때 LF->CRLF 로 변경
 - Windows 권장
- `core.autocrlf=input`
 - 체크인할 때 CRLF->LF
 - 체크아웃은 변경없이 그대로 적용
 - Linux, Mac 권장

config

- user.name
- user.email
- commit 정보에 포함
- email 검증을 하지 않는다.
- GitHub 등 remote 저장소를 사용할때 신뢰성 문제 발생 가능성

```
kkang@LAPTOP-TD944GIF MINGW64 /c/study-git/test7 (main)
$ git log
commit be8d32ee66a786655426227181f3775e2e2572db (HEAD -> main)
Author: kkang <kkang104@naver.com>
Date: Thu Jan 26 09:35:59 2023 +0900
```

commit message pattern

- commit message 의 subject line 은 항상 아래의 문장을 완성할 수 있게 지정할 것을 권장
- If applied, this commit <subject line>

[type](optional scope): [subject]

[optional body]

[optional footer]

commit message pattern

type

- feat : 새로운 기능 추가
- docs : 문서 수정
- refactor : 코드 리팩토링
- test : 테스트 코드, 리팩토링 테스트 코드 추가(프로덕션 코드 변경 없는 경우)
- chore : 빌드 업무 수정, 패키지 매니저 수정(프로덕션 코드 변경 없는 경우)
- design : css 등 UI 디자인 변경
- ci : CI 관련 변경
- comment : 주석 추가 및 변경
- revert : 되돌리기
- rename : 파일 혹은 폴더명을 수정하거나 옮기는 작업
- remove : 파일을 삭제하는 작업
- !BREAKING CHANGE : 외부에 영향을 미치는 api 변경
- !HOTFIX : 급한 오류를 해결하는 경우
- fix : 버그 수정
- style : 코드 formatting, 세미콜론 누락, 코드 변경 없는 경우
- build : 관련 변경 사항 빌드
- perf : 성능을 향상시키는 코드 변경

commit message pattern

scope

- optional
- 변경된 부분을 표기
- 함수가 변경되었다면 함수명
- 함수가 추가되었다면 클래스명

subject

- 첫 글자는 대문자
- 마지막에 .(마침표)를 찍지 않으며 영문 기준 최대 50자 이내 작성
- Subject 는 명령문의 형태로 작성(동사 원형 사용)

body

- 각 줄은 최대 72자 이내로 작성
- 어떻게 변경했는지보다, 무엇을 변경했고, 왜 변경했는지를 설명

commit message pattern

footer

- 선택사항
- 관련된 이슈를 언급
- Closes(종료), Fixes(수정), Resolves(해결), Ref(참고), Related to(관련) 키워드 이용

권장사항

- Subject 와 body 는 공백 라인으로 구분
- Subject 는 50자 이내
- Subject 의 첫 글자는 대문자로 작성
- Subject 의 마지막에 마침표 사용하지 않는다.
- Subject 는 명령조(동사 원형)
- Body 는 72자 마다 줄 바꿈
- Body 는 "어떻게" 보다 "무엇을", "왜" 에 초점을 맞추어 작성



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare