

Git

Source Code Management(SCM) Tool: 코드 관리 도구

Version Control System(VCS): 버전 관리 시스템

버전을 통해 코드를 관리하는 도구

버전 관리를 하는 이유

1. 언제든지 과거로 돌아갈 수 있음
2. 수정 이력을 확인할 수 있음

Git 명령어

0. 폴더 생성

git 저장소로 만들 폴더 생성

1. `git init`

git 프로젝트(저장소)를 시작(폴더 단위)

```
Initialized empty Git repository in C:/Users/student/git_basic/.git/
```

1. `(master)`
2. `.git` 폴더 생성

(중요) git 프로젝트를 종료

- `.git` 폴더 삭제

2. `git status`

git의 현재 상태를 출력

- 최초상태

```
On branch master: master라는 브랜치에 있음
```

```
No commits yet: 아직 commit이 없음
```

```
nothing to commit (create/copy files and use "git add" to track)  
: commit할 게 없음
```

- 파일/폴더 추가시 (`touch a.txt`)

...

Untracked files: (추적되지 않은 파일)

(use "git add <file>..." to include in what will be committed)

a.txt (빨강)

nothing added to commit but untracked files present (use "git add" to track)

: commit하기 위해 추가된 파일이 없음, 그러나 추적되지 않은 파일은 있음

- add 이후(`git add a.txt`)

...

Changes to be committed:

commit할 변화들

(use "git rm --cached <file>..." to unstage)

new file: a.txt (초록)

- commit 이후(`git commit -m "first commit"`)

On branch master

nothing to commit, working tree clean

3. `git add` [파일명]

commit하기 위한 stage에 파일 추가

4. `git commit -m "커밋메시지"`

스냅샷 & 버전 생성

- `git commit --amend`: (주의) 최종 커밋메시지 변경

commit의 구성 요소

- committer(author): commit을 찍은 사람
- commit datetime(date): commit을 찍은 시간
- commit message: commit에 대한 내용 (필수)

5. `git config`

- `git config --global user.email "이메일"`: 사용자 이메일 설정
- `git config --global user.name "이름"`: 사용자 이름 설정
- `git config --global user.email`: 사용자 이메일 출력
- `git config --global user.name`: 사용자 이름 출력

6. `git log`

commit 목록(log) 출력

- `git log --oneline`: 한줄 출력
- `git log -1`: 출력 개수 한정

7. `git restore --staged` [파일명]

unstage: stage 상태 해제

8. `git restore` [파일명]

최종 커밋 시점으로 파일 상태를 복원(restore)

9. `git diff` [파일명]

변경 내역 출력

10. `git remote`

원격 저장소 정보 출력

- `git remote -v`: 상세 출력(verbose)
-

11. `git remote add` [이름] [주소]

원격 저장소 정보 추가

- 이름: 관례/관습(Convention) `origin` (원본)
- 주소: Github에서 복사 가능

12. `git push` [이름] [브랜치]

원격 저장소에 코드 업로드

- `git push origin master`

```
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 668 bytes | 668.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/hphk-john/git_review.git
* [new branch]      master -> master
```

13. `git clone` [주소]

원격 저장소 다운로드(복제)

14. `git pull` [이름] [브랜치]

- `git pull origin master`

Git Branch

15. `git branch`

모든 branch를 출력

16. `git branch` [브랜치명]

새로운 branch를 생성

17. `git checkout` [브랜치명]

branch 이동

18. `git merge` [브랜치명](목적)

현재 branch(S)에서 대상 branch(O)를 병합

(참고) CLI 명령어

- `echo` [문자열]
 - command line 출력
- `echo` [문자열] > [파일명]
 - 특정 파일에 출력(기록)
- `cat` [파일명]

- 파일의 내용을 출력