

# Git

프로젝트(코드) 관리 도구

## SCM & VCS

- SCM(Source Code Management): (소스)코드 관리 도구
- VCS(Version Control System): 버전(형상) 관리

Git (버전을 통해) 코드 관리 도구

## Git 명령어

### git init

git으로 코드 관리를 시작(initiate)

- (중요) git은 폴더를 기준으로 프로젝트(코드) 관리
  1. .git 폴더 생성
  2. git으로 프로젝트 관리 시작
  3. (master) 프롬프터가 생성

### git status

git의 상태를 출력

- 생성 직후

```
On branch master
```

```
No commits yet
```

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

- a.txt 파일 생성 후

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

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

```
a.txt
```

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

## `git add` [파일/폴더명]

저장을 위한 준비

## `git commit -m '커밋 메시지'`

준비된 파일/폴더의 버전을 생성(현재 상태 스냅샷, 현재 상태 저장)

- 버전: 날짜, 누가 기록, **커밋 메세지**, 커밋 해쉬(hash - 임의의 숫자)

## `git log`

현재까지의 버전 히스토리 출력

- `git log --oneline`: 한줄로 출력

## `git diff` [파일명]

이전 버전과의 차이를 출력

---

## `git remote`

원격저장소의 정보를 출력

- `git remote -v`: 상세한 원격저장소 정보 출력

## `git remote add` [원격저장소이름] [원격저장소주소]

- 일반적으로 첫번째 원격저장소의 이름은 `origin` (원본)
- `git remote add origin [주소]`
- github 원격 저장소 생성 후 받는 URL

## `git push` [원격저장소이름] [브랜치이름]

- 일반적으로 첫번째 원격저장소의 이름은 `origin` (원본)
- 일반적으로 기본 브랜치의 이름은 `master`
- `git push origin master`