

# Git

---

버전을 통해 코드를 관리 하는 도구

## SCM & VCS

---

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

## Git 명령어

---

Git은 폴더 단위로 프로젝트/코드를 관리

### 1. `git init`

initialize, initiate

- git으로 코드 관리를 시작
  1. `(master)` 라는 표시가 프롬프트에 표시됨
  2. `.git` 폴더가 생성됨 (맥, 윈도우 공통)

### 2. `git status`

가장 중요한 명령어

- git의 상태를 출력
  1. `git init` 직후

```
On branch master
```

```
No commits yet
```

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

- No commits yet: 아직 commit이 없다 (버전 == 스냅샷 == 특정상태 == 저장)
- nothing to commit: commit 할게 없다.
- 2. `a.txt` 파일 추가 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)
```

- untracked files: 추적되지 않는 파일이 있어요. (어떤 파일)
- nothing added to commit but untracked files present: 커밋할 파일은 없지만, 추적되지 않는 파일은 다.

3. `git add a.txt` 이후

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

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

```
new file:   a.txt
```

- changes to be committed: 커밋될 준비가 된 파일이 있음

4. `git commit -m 'first commit'` 이후

```
On branch master
```

```
nothing to commit, working tree clean
```

- nothing to commit: 커밋할 게 없음
- working tree(directory) clean: 작업 폴더가 깔끔

### 3. `git add` [파일명]

git이 스냅샷을 찍기 위해 추적하는 리스트에 파일 추가

### 4. `git commit -m '[커밋 메시지]'`

git을 통해 스냅샷을 찍음 (== 버전을 만듦, 현재 상태를 저장)

- `-m` : message 줄임말

1. 언제 찍었는지
2. 누가 찍었는지: 이름, 이메일
3. 메시지
4. commit hash

## 5. `git log`

git으로 지금까지 저장된 커밋들의 로그를 출력

- `git log --oneline` : 한줄로 커밋을 출력
- `git log --[숫자]` : 입력된 숫자만큼 역순으로 출력

## 6. `git restore --staged [파일명]`

staging area에 추가된 파일을 복원시키는 것

---

## `git config` 설정

설치 후 1번만

- `git config --global user.name '사용자 이름'`
- `git config --global user.email '사용자 이메일'`