

# Git 2

---

코드 관리 도구 & 협업 도구

## 협업 도구로서의 git

---

git으로 관리되는 프로젝트 == git으로 관리되는 폴더

## Git 원격 관련 명령어

---

### 1. git remote

현재 관리되고 있는 원격 저장소의 정보를 출력

- `git remote -v`: 주소까지 상세 정보를 출력(verbose mode)

### 2. git remote add [원격저장소의 이름] [원격저장소의 주소]

새로운 원격 저장소 정보를 추가

- `git remote add origin https://github.com/github유저네임/저장소의이름`
- `git remote remove [원격저장소의 이름]`: 해당 원격저장소 정보를 삭제

### 3. git push [원격저장소의 이름] [브랜치의 이름]

원격 저장소에 코드를 업로드(밀어넣기)

- `git push origin master`
- `git push -u origin master`: 업스트림(upstream) 설정

### 4. git clone [원격저장소의 주소] (폴더명)

원격 저장소의 코드를 다운로드

## Git Branch 명령어

---

## 1. git branch

- 현재 생성되어 있는 branch들의 목록을 출력

## 2. git branch [브랜치 이름]

- 새로운 branch 생성

## 3. git merge [브랜치 이름]

- branch를 병합 (현재 속한 브랜치에서 인자로 주어진 브랜치를 합병)\
- `git merge test (master)` : master 브랜치가 test를 병합함

## 4. git branch -d [브랜치 이름]

(거의 모든, 주요 branch를 제외한) branch는 일회용이다. 병합된 브랜치는 항상 삭제한다.

- `-d` : 삭제(delete)
- `git branch -d test` : test라는 브랜치를 삭제

## 번외 질문

### commit 은 얼마나 빈번히 해야하나요?

내 PJT는 내맘대로, 다른 사람들과 함께 하는 합의된 룰대로

1. 파일마다 1번씩
2. 코드 1줄마다 1번씩
3. 코드 10줄마다 1번씩
4. 내맘대로
5. 선임 맘대로
6. 회사 룰대로 -> logically separable functional commit (논리적으로 분리 가능한 기능 중심)

```
def hello(
```

### push는 얼마나 빈번히 해야하나요?

필요할 때에 (보수적)

(1) 다른 호스트/로컬 컴퓨터로 옮겨야할 때에 (2) 코드를 공유하거나

1. 내맘대로
2. 선임 맘대로
3. 회사 룰대로
4. 1커밋 1푸쉬

## 5. 10커밋 1푸쉬

### **Github의 프로젝트를 다운로드 받는 방법**

1. Zip 파일로 다운로드
2. `git clone`