

# Github 기본 설명서

Semin Hyun

2021.04.20



# 1. Git 과 GitHub의 차이

---

- Git
  - 버전관리 및 협업 도구
- GitHub
  - 서비스식 Git 저장소

## 2. Git이란?

---

### ■ Git이 만들어진 이유

- Git이 없었을 때는 분산버전 관리 시스템이라는 BitKeeper라는 시스템을 사용
- 리눅스를 사용하던 개발자가 BitKeeper를 해킹하여 문제가 발생
- 리눅스 커널에서 더 이상 무료로 BitKeeper를 사용하지 못함.
- 리눅스 오픈 소스를 만든 리눅스 토발즈가 개발

### ■ Git을 쓰는 이유

- 버전관리 기능
  - 기본의 압축파일 시스템의 문제점
  - 가벼운 메모리 차지
- 협업 기능
  - 협업 시 같은 파일을 수정했을 경우 내용을 합치려면 일일이 다 확인해야 하는 문제점을 해결

### 3. GitBash 설치

---

- <https://gitforwindows.org/>
  - Windows OS 사용자
  - 해당 링크 접속 후 다운로드
- Mac OS User
  - <https://git-scm.com/downloads>에 들어가 MAC용 Git을 설치

## 4. New Repository 생성

---

- 사전 작업
  - <https://github.com/>
  - Github에 접속하여 회원가입 후 로그인 진행하기
- Repository 생성
  - github에 나만의 공간을 만든다고 생각하면 됨.

## 5. Root Folder 연결

---

- 기본 폴더
  - Web 상의 Git과 연결될 Local Folder
    - 1) Local 원하는 위치에 폴더 생성
    - 2) Command창에 아래 명령어를 하나씩 입력 (Root Folder를 연결하는 명령어)
      - ① **git init**
      - ② **git remote add origin repository address**(본인의 repository 주소를 입력)

## 6. Push & Pull

---

- **Push**

- Git repository로 자료를 upload하는 것.

- **Pull**

- Push와 반대로 가져오는 것.

- 기본 명령어

- **git pull origin master** : 아까 지정해 놓은 Repository에서 지정 repository에 모든 자료를 한 번에 끌어오는 명령어
- **git status** : 현재 나의 local folder와 git과의 싱크 상태를 체크해주는 명령어
- **git add .** : local folder와 git repository 사이에 어떤 변경된 파일이 있을 때 그 모든 파일들을 tracking 하고 적재하는 명령어
- **git add {file\_name}** : 특정 파일을 적재
- **git commit -m "message"** : 인식할 수 있는 commit을 달아주는 명령어
- **git push origin +master** : 실제로 Git에 Push를 요청하는 명령어

# 7. git commit

---

## ▪ git commit

- Commit message에 Title과 Contents 입력을 위해 vim으로 이동
- git commit -m “제목”
- 만약 첫 commit을 경우 커맨드 창에 아래 명령어 입력
  - git config --global user.email “이메일”
  - git config --global user.name “계정이름”
- commit 규칙
  - 제목과 본문을 빈행으로 분리
  - 제목 행을 50자로 제한하고, 제목 행의 첫 글자는 대문자
  - 제목 행 끝에 마침표(.)를 넣지 않음
  - 제목 행에 명령문 사용



Gachon  
Cognitive Computing Lab

