

# Day 7 - Git, Github

## Git

- 버전관리 시스템의 종류
  - 버전관리
    - 여러 파일을 하나의 버전으로 묶어 관리하는 것
  - 버전관리 시스템의 종류
    - 1. 클라이언트 - 서버 모델
      - a. 하나의 중앙 서버로 여러 클라이언트들이 각자 필요한 것만 가져와서 작업을 하고 다시 중앙 서버로 보내서 통합하는 방식
      - b. CVS, SVN
    - 2. 분산 모델
      - a. 하나의 중앙 서버가 존재하지만, 여러 클라이언트들은 각자의 컴퓨터 저장소에 전체 사본을 가지고 작업하는 방식
      - b. GIT

## Git의 장점

1. 동시에 작업하는 사람들과 소스코드를 주고받을 필요가 없음
2. 같은 파일을 여러명이 동시에 병렬 개발이 가능
3. 변동 과정을 체계적으로 관리할 수 있고, 언제든지 지난 시점의 버전으로 되돌릴 수 있다.
4. 인터넷이 연결되지 않은 곳에서도 개발을 진행 할 수 있고, 중앙 서버의 데이터가 유실되어도 다시 복구할 수 있다.

## Git 설치

1. 구글에서 “git 설치” 검색
2. 왼쪽 메뉴에서 Downloads를 클릭  
<https://git-scm.com/downloads/win>
3. x64 setup
4. 다운후 표준설치

## Git 명령어

1. 프로젝트 디렉토리 내에 로컬 저장소(local repository)를 생성
  - a. git init
    - i. .git 폴더 생성됨
2. git 버전 관리할 파일을 선택 (stage에 파일을 올림)
  - a. git add [[main.py](#)]
3. 하나의 버전으로 만들기
  - a. git commit -m [[main.py](#)를 커밋]

만약 아래와 같이 메시지가 나온다면:

Author identity unknown

\*\*\* Please tell me who you are.

Run

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

to set your account's default identity.

Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got  
'Administrator@DESKTOP-HUJUVSS.(none)')

→ →

```
git config --global user.email "gjlee99211@gmail.com"
```

```
git config --global user.name "Gyujin Lee"
```

#### 4. 여러 파일을 함께 선택 (stage에 현재 디렉토리에 있는 변화된 모든 파일)

- a. `git add .`

#### 5. 선택된 모든 파일을 커밋

- a. `git commit -m "현재 디렉토리에 있는 변화된 모든 파일을 커밋"`

#### 6. 로그 확인

- a. `git log`

```
C:\Gyujin\GitTest>git log
commit f1a86819ffb816eecd4ff42421598e7090afb0e4 (HEAD -> master)
Author: Gyujin Lee <gjlee99211@gmail.com>
Date: Fri May 30 20:54:09 2025 +0900

현재 디렉토리에 있는 변화된 모든 파일을 커밋

commit 24151ef78badd2269baab6a6f0a70ae389567856
Author: 오지환 <ovitwisdom@gmail.com>
Date: Fri May 30 20:43:04 2025 +0900

main.py를 커밋

C:\Gyujin\GitTest>
```

## 깃허브(Git Hub)

1. <https://github.com/>
2. git으로 버전 관리한 코드를 업로드 할 수 있는 클라우드 서비스
3. 단순히 저장만 하는 것이 아니라 유저들과 함께 코드를 공유하고 온라인으로 하나의 프로젝트를 개발할 수 있게 해주는 사이트
4. 깃허브 가입 후 **repository** 만들기
  - a. create repository 버튼을 클릭하여 생성
5. 로컬 저장소에서 **github** 원격 저장소를 설정
  - a. git remote add origin [https://github.com/JinLeeGG/Git\\_practice.git](https://github.com/JinLeeGG/Git_practice.git)
6. 생성한 커밋(버전)들 **github**로 전송(푸시)
  - a. git push -u origin master

ssh-keygen -t rsa -b 4096 -C "[gjlee99211@gmail.com](mailto:gjlee99211@gmail.com)"