

Github

소프트웨어 형상관리란 소프트웨어의 변경사항을 체계적으로 추적하고 통제하는 것을 말 합니다. Git은 오늘날 가장 많이 사용되는 형상관리 시스템으로 Git을 웹 사이트 형태로 서비스하는 곳이 [github](https://github.com) 입니다.

#01. Git 클라이언트 설치

1) Windows

<https://git-scm.com/>에서 소프트웨어를 내려받아 설치 진행

2) Mac

터미널상에서 아래의 명령어를 수행

```
$ xcode-select
```

3) Git 클라이언트 환경 설정

명령프롬프트(혹은 터미널)에서 아래 명령어 입력

```
$ git config --global user.name "본인이름"  
$ git config --global user.email "자신의 이메일 주소"
```

예시

```
$ git config --global user.name "leekh"  
$ git config --global user.email "leekh4232@kakao.com"
```

#02. Github 계정 설정

1) 사이트 가입

<https://github.com>에서 회원가입 진행

2) SSH 인증서 생성

명령 프롬프트(혹은 터미널)에서 아래 명령어 수행

```
$ ssh-keygen
```

사용 예시

```
C:\Users\leekh>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\leekh/.ssh/id_rsa):  # <-- 저장경로 지정 (기본경로 사용을 위해 그냥엔터)
C:\Users\leekh/.ssh/id_rsa already exists.
Overwrite (y/n)? y      # <-- 이미 인증서 파일이 존재할 경우 덮어쓸지 여부 확인
Enter passphrase (empty for no passphrase):  # <-- 인증서 비밀번호 (지정하지 않고 그냥 엔터)
Enter same passphrase again:  # <-- 비밀번호 확인 (그냥 엔터)
Your identification has been saved in C:\Users\leekh/.ssh/id_rsa.
Your public key has been saved in C:\Users\leekh/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:uopLy6C2vGyhTeYtTTiWf9Rp6HFRHg3hmyLApNAinQY leekh@DESKTOP-7E34813
The key's randomart image is:
+---[RSA 3072]-----+
|E+ ..      o+      |
|o.=+      .o .    |
|.o. o      o..     |
|      . . .o      |
|  o  .oSoo      |
|.B . +o=.      |
|o*o* o.+      |
|=*+o+ o.      |
|+=Bo.o.      |
+-----[SHA256]-----+
```

인증서 확인

사용자 홈 디렉토리내의 `.ssh` 폴더 안에 `id_rsa` 파일과 `id_rsa.pub` 파일이 생성된다.

OS	위치
Windows	C:\Users\계정이름\.ssh
Mac	/Users/계정이름/.ssh

맥의 경우는 해당 디렉토리에 접근하기 위해 터미널에서 명령어 입력

```
$ open ~/.ssh
```

파일	설명
id_rsa	서버키 (내 컴퓨터에 보관해야 하는 파일. 파일 경로 고정)
id_rsa.pub	공개키 (github에 등록해야 하는 파일)

컴퓨터에서 git 관련 명령어 실행시 서버키와 공개키 두 파일을 비교하여 한 쌍임이 확인되면 사용자 인증을 생략한다.

github 연동시 **https** 방식이 아닌 **ssh** 방식을 사용할 수 있기 때문에 파일 전송 속도가 더 빠르다.

3) Github에 공개키 등록하기

1. 공개키(**id_rsa.pub**) 파일을 텍스트 편집기로 열고 내용을 복사한다.
2. Github의 계정 설정 페이지 (<https://github.com/settings/profile>) 에 접속하여 **SSH and GPG keys** 메뉴로 이동
3. 우측 상단의 **New SSH Key**를 클릭
 1. Title은 임의로 설정
 2. Key type은 기본값 유지
 3. Key 항목에 공개키의 내용을 적용
 4. **Add SSH key** 버튼을 눌러서 저장

내가 사용하는 컴퓨터에 저장된 파일과 Github에 등록된 내용을 비교하여 인증이 처리 되므로 두 곳중 한 곳에서 키가 삭제되면 인증이 불가능해 진다.

#03. 저장소 생성

1) Github에 저장소 생성하기

<https://github.com/new> 페이지에서 저장소 생성

우측 상단의 **+**버튼 클릭 후 **New Repository** 선택

2) Git 저장소를 내 컴퓨터에 복제하기

새로 작업을 시작하는 경우

Github으로부터 새로 만들어진 빈 저장소를 복제한다.

```
$ git clone git@github.com:사용자아이디/저장소이름
```

사용예시

```
C:\Users\leekh>git clone git@github.com:leekh4232/MegaitDataAnalysis
Cloning into 'MegaitDataAnalysis'...
The authenticity of host 'github.com (20.200.245.247)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
```

```
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

이후 복제한 저장소를 VSCode의 **폴더 열기** 메뉴로 열고 작업 진행

기존의 작업 폴더를 저장소에 연결하는 경우

명령프롬프트(혹은 터미널)상에서 작업 폴더로 이동

git 초기화

작업 폴더 안에 **.git** 폴더를 숨김상태로 생성한다.

```
$ git init
```

branch 설정

이 명령은 아무런 결과가 표시되지 않는다.

```
$ git branch -M main
```

원격 저장소 연결

```
$ git remote add origin git@github.com:사용자이름/저장소이름
```

저장소의 현재 상태 내려받기

```
$ git pull origin main
```

커밋

변경 사항을 업로드 대기 상태로 전환

```
$ git add -A
$ git commit -m "작업내용에 대한 간단한 코멘트"
```

push

커밋 목록을 github에 전송

```
$ git push origin main
```

#04. Github 활용

내 컴퓨터에 저장된 파일을 git에 push한 이후에는 git에 등록된 파일이 원본

내 컴퓨터의 파일을 삭제하더라도 상관 없다.

Commit은 변경 내역을 업로드 대기 상태로 등록하는 과정일 뿐, 실제로 Git에 파일이 전송되지는 않는다.

Push를 수행하면 Commit 내역이 한번에 일괄 업로드 된다.

Clone 받은 후 변경 사항만 내려 받기 위해서는 Pull을 수행하면 된다.