

Git

[1 GitHub 가입하기](#)

[2 Git 접속 SSH Key 만들기](#)

[3 Repositories 만들기](#)

[4 로컬 저장소 관리](#)

[4.1 로컬 저장소 만들기](#)

[기본용어](#)

[옵션 설정](#)

[기본 CLI 명령어](#)

[원격저장소 관리](#)

[원격저장소에 커밋 올리기](#)


[원격저장소에서 파일 가져오기](#)

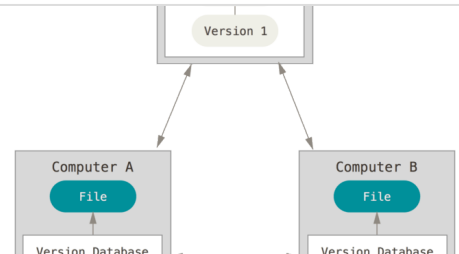
[원격 저장소 삭제](#)

참고 사이트

Git - 버전 관리란?

"버전 관리"는 무엇이고 우리는 왜 이것을 알아야 할까? 버전 관리 시스템은 파일 변화를 시간에 따라 기록했다가 나중에 특정 시점의 버전을 다시 꺼내올 수 있는 시스템이다. 이 책에서는 버전 관리하는 예제


 <https://git-scm.com/book/ko/v2/%EC%8B%9C%EC%9E%91%ED%95%98%EA%B8%B0-%EB%B2%84%EC%A0%84-%EA%B4%80%EB%A6%AC%EB%9E%80%3F>

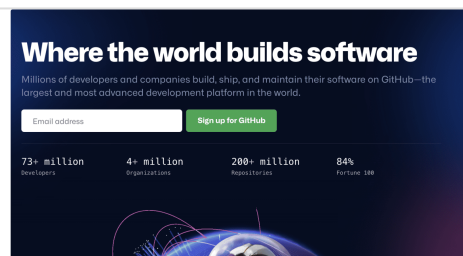


1 GitHub 가입하기

GitHub 계정 생성하는 방법

GitHub는 원격 Git 저장소를 호스팅해주고, 다른 개발자와 협업할 수 있는 다양한 기능을 제공해주는 서비스입니다. 현재는 개발자라면 누구나 사용하는 서비스 중 하나입니다. 이 글에서는 깃허브 계정 생성하


 <https://www.lainyzine.com/ko/article/how-to-create-github-account/>



2 Git 접속 SSH Key 만들기

GitHub 접속 용 SSH 키 만드는 방법

GitHub 를 사용하려면 SSH 키를 등록해야 합니다. Git과 GitHub를 처음 사용하면 SSH키를 왜 만들어야 하는지, 어떻게 만들어야 하는지, 그리고 따라서 키를 만들어봤는데 잘 만들어진 건지 확인하는 데 어려

 <https://www.lainyzine.com/ko/article/creating-ssh-key-for-github/>

SSH keys / Add new

Title

Key

Begin with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', or 'ssh-ed25519'

Add SSH key

3 Repositories 만들기

4 로컬 저장소 관리

4.1 로컬 저장소 만들기

기본용어

워킹트리 (Working tree)	워크트리, 워킹 디렉토리, 작업 디렉토리, 작업 폴더 모두 같은 뜻으로 사용 사용자가 파일과 하위 폴더를 만들고 작업 결과물을 저장하는 곳
로컬저장소 (Local Repository)	Git init 명령으로 생성되는 [.git] 폴더가 로컬저장소이다. 커밋, 커밋을 구성하는 객체, 스테이지가 모두 이 폴더에 저장된다.
원격저장소 (Remote Repository)	로컬저장소를 업로드하는 곳을 원격저장소라고 한다.
Git 저장소	Git 명령으로 관리할 수 있는 폴더 전체를 일반적으로 Git 프로젝트 혹은 Git 저장소라고 한다.

git init	현재 폴더에 Git 저장소를 생성한다.
git status	Git 워킹트리의 상태를 보는 명령으로 워킹트리가 아닌 폴더에서 실행하면 오류가 발생
git status -s	git status 명령보다 짧게 요약해서 상태를 보여주는 명령으로 변경된 파일이 많을

때 유용하다.

옵션 설정

```
# 지정한 전역 옵션의 내용을 살펴본다.
git config --global <옵션명>

# 지정한 전역 옵션의 값을 새로 설정한다.
git config --global <옵션명> <새로운 값>

# 지정한 전역 옵션을 삭제한다.
git config --global --unset <옵션명>

# 지정한 지역 옵션의 내용을 살펴본다.
git config --local <옵션명>

# 지정한 지역 옵션의 값을 새로 설정한다.
git config --local <옵션명> <새로운 값>

# 지정한 지역 옵션을 삭제한다.
git config --local --unset <옵션명>

# 지정한 시스템 옵션의 내용을 살펴본다.
git config --system <옵션명>

# 지정한 시스템 옵션의 값을 새로 설정한다.
git config --system <옵션명> <새로운 값>

# 지정한 시스템 옵션을 삭제한다.
git config --system --unset <옵션명>

# 현재 프로젝트의 모든 옵션을 검색한다.
git config --list
```

```
git config --global user.email # 현재 이메일 확인
git config --global user.name # 현재 사용자 확인

# 기본 에디터 확인
git config core.editor
git config --global core.editor
git config --system core.editor
```

기본 CLI 명령어

```

# 파일을 스테이지에 추가
git add 파일1 파일2 ...

# 스테이지에 있는 파일들을 커밋
git commit

# add 명령을 생략하고 바로 커밋하고 싶을 때
git commit -a

# 현재 브랜치에서 새로 생성한 커밋들을 원격저장소에 업로드한다.
git push [-u] [원격저장소 별명] [브랜치 이름]

# 원격저장소의 변경사항을 워킹트리에 반영
git pull

# 원격저장소의 브랜치와 커밋들을 로컬저장소와 동기화 한다.
git fetch [원격저장소 별명] [브랜치 이름]

# 지정한 브랜치의 커밋들을 현재 브랜치 및 워킹트리에 반영한다.
git merge 브랜치 이름

# 계정 정보 삭제
git config --global --unset user.name "계정명"
git config --global --unset user.email "이메일 주소"

```

```

$ mkdir <Directory>
$ cd <Directory>
$ git init

# 계정 정보 입력
# 현재 저장소의 사용자만 등록할때는 --global 생략
$ git config --global user.email "이메일 주소"
$ git config --global user.name "계정명"

# 파일 추가
$ git add <파일명> # 디렉토리내 전체 파일 선택은 git add .

# 커밋하기
$ git commit -m "메시지"

```

원격저장소 관리

원격저장소에 커밋 올리기

```
$ git remote add origin <Git 주소 코드 복사>
$ git push origin main
```

원격저장소에서 파일 가져오기

```
$ git remote -v # 원격저장소 확인
$ git pull origin main
```

원격 저장소 삭제

```
$ git remote -v # 원격저장소 확인
$ git remote rm 저장소 이름 # 원격저장소 삭제
```

```
# git 저장소 초기화
$ git init

$ ls -al
total 12
drwxrwxr-x  3 ubuntu ubuntu 4096 Jun  6 21:35 .
drwxr-xr-x 17 ubuntu ubuntu 4096 Jun  6 21:35 ..
drwxrwxr-x  7 ubuntu ubuntu 4096 Jun  6 21:35 .git

# 계정 정보 입력
# 현재 저장소의 사용자만 등록할때는 --global 생략
$ git config --global user.email "이메일 주소"
$ git config --global user.name "계정명"

# git 계정 정보 확인
$ cat ~/.gitconfig
```

git 접속용 SSH Key 만들기

```
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

$ ~/.ssh/config
Host github.com
```

```
IdentityFile ~/.ssh/"<개인키 파일 이름>"
User git

# Git 접속 테스트
$ echo "# git_test" >> README.md
$ git init
$ git add README.md
$ git commit -m "first commit"
$ git branch -M main
$ git remote add origin git@github.com:"<레포지토리 이름>"/git_test.git
$ git push -u origin main
```

```
! [rejected]          main -> main (fetch first)

# 해결방법
$ git push origin +main
```