

3. Git 설정하기

SSH 키 접속하기

SSH 키 생성

```
-- SSH키 생성
$ cd ~/.ssh
$ ssh-keygen -t ed25519 -C "your_email@example.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/lainyzine/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/lainyzine/.ssh/id_ed25519.
Your public key has been saved in /Users/lainyzine/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:MRR2EWPjezAxwlrxCgfgjq43CJjtZVl8Heo5YU/912I your_email@example.com
The key's randomart image is:
+--[ED25519 256]--+
|      ..*oXo      |
|      . +o*.=      |
|      ...o+o=0     |
|      . +o=+o+.    |
|.o  .o +S=.  ..  .|
|+ .o+   + ..  E o|
|.o.o.   .   . o |
|..+           |
|.o..           |
+-----[SHA256]-----+

$ ls
id_ed25519  id_ed25519.pub
```

공개키 GitHub에 등록하기

1. 로그인 후 **오른쪽 상단의 프로필**을 클릭하고, **Settings** 메뉴로 이동
2. 오른쪽 사이드 바에서 **SSH and GPG keys** 메뉴를 선택
3. 오른쪽 상단의 **New SSH key** 클릭
4. SSH 키 등록창 표시

SSH keys / Add new

Title

Key type

Authentication Key ↕

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

- Title: 적당한 이름 기입
- Key: id_ed25519.pub 의 내용을 복사

SSH 접속 설정

```
$ vim config
Host github.com
  IdentityFile ~/.ssh/id_ed25519
  User git
:wq

$ chmod 400 config
$ ssh -T git@github.com
...
```

GitHub 계정 정보 입력

```
-- 계정 정보 입력
$ git config --global user.email "your_email@example.com"
$ git config --global user.name "your_ID"

-- 계정 정보 확인
$ git config --global user.email
your_email@example.com
```

```

$ git config --global user.name
your_ID

-- 원격 저장소 등록
$ git remote add origin git@github.com:your_ID/리포지토리.git

-- 원격 저장소 확인
$ git remote -v
origin  git@github.com:your_ID/리포지토리.git (fetch)
origin  git@github.com:your_ID/리포지토리.git (push)

```

Git 로컬 저장소 관리

로컬 저장소 생성 및 버전 만들기

```

$ git init
$ ls -al
drwxrwxr-x  3 ubuntu ubuntu 4096 Nov 16 06:45 .
drwxr-x--- 11 ubuntu ubuntu 4096 Nov 17 03:26 ..
drwxrwxr-x  8 ubuntu ubuntu 4096 Nov 16 07:07 .git

-- 현재 작업 디렉토리의 상태를 알려주는 명령
$ git status
On branch main -- 현재 기본 브랜치

nothing to commit, working tree clean

-- 새로운 파일 생성
$ echo "Hello~ " > hello.txt

$ git status
On branch main

No commit yet -- 현재 아무런 커밋이 없다.
Untracked files: -- 깃이 기존에 변경 사항을 추적하지 않은 대상이다.
    (use "git add <file>..." to include in what will be committed)
        hello.txt

nothing added to commit but untracked files present (use "git add" to track)

$ git add hello.txt
$ git status
On branch main
Changes to be committed: -- 성공적으로 스테이지에 추가 되었다.
    (use "git restore --staged <file>..." to unstage)
        new file:   hello.txt

$ git commit -m "first commit" -- 커밋하기

```

```

[main af67d10] first commit
1 file changed, 1 insertion(+)
create mode 100644 hello.txt

$ git log
commit af67d109fd99eea73b941b4a2e51a0 (HEAD -> main) -- 커밋 해시
Author: your_ID <your_ID@example.com> -- 만든 사람
Date: Thu Nov 17 04:32:35 2022 +0000 -- 만든 날짜

    first commit
(END)

$ vim hello.txt
Hello~ World!! -- 추가

:wq

$ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   hello.txt -- hello.txt 가 수정 되었다고 표시

no changes added to commit (use "git add" and/or "git commit -a")

$ git commit -am "second commit" -- git add와 커밋을 동시에 진행
[main 340a9c1] second commit
1 file changed, 1 insertion(+), 1 deletion(-)

$ git log
commit 340a9c1218f4a73d9f271dbe132e7a7 (HEAD -> main)
Author: <your_ID@example.com>
Date: Thu Nov 17 04:38:47 2022 +0000

    second commit

commit af67d109fd99eea73b941b4a2e51a0
Author: <your_ID@example.com>
Date: Thu Nov 17 04:32:35 2022 +0000

    first commit

```

git log 옵션

```

0
$ git log --oneline -- 커밋 목록을 한줄로 표시
340a9c1 (HEAD -> main) second commit
af67d10 first commit

$ git log -p -- 커밋이 무엇을 변경했는지 상세하게 표시
commit 340a9c1218f4a73d9f271dbe132e7a7 (HEAD -> main)

```

```
Author: your_ID <your_ID@example.com>
Date:   Thu Nov 17 04:38:47 2022 +0000
```

second commit

```
diff --git a/hello.txt b/hello.txt
index 6d223d2..3f98a14 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,1 @@
-Hello~
+Hello~ World!!
```

```
commit af67d109fd99eea73b941b4a2e51a0
Author: your_ID <your_ID@example.com>
Date:   Thu Nov 17 04:32:35 2022 +0000
```

first commit

```
diff --git a/hello.txt b/hello.txt
new file mode 100644
index 0000000..6d223d2
--- /dev/null
+++ b/hello.txt
@@ -0,0 +1 @@
+Hello~
```