

# USING GITHUB

Homepage : <https://drv.tw/~sj6219@hotmail.com/od/Doc4Code/>

Email: [sj6219@hotmail.com](mailto:sj6219@hotmail.com)

2021/04/17

## SETUP

- 1 [GitHub](#) 에 계정 생성
- 2 <https://www.syntevo.com/smartgit/> 를 설치  
프로그램 처음 실행시 non-commercial 를 선택해야 한다.
- 3 SSH Key 생성
  - 3.1 cmd 창에서 다음을 실행한다.  
  
**> ssh-keygen**  
  
passphrase 를 공백으로 하면 편리하다.
- 4 GitHub 에 SSH Key 에 등록
  - 4.1 GitHub > Settings > SSH and GPG keys > New SSH key

## SSH keys / Add new

Title

Key

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

Add SSH key

파일 %HOMEPATH%\W.ssh\id\_rsa.pub 내용을 key 에 복사

5 GitHub> Repositories > New

새로운 Repo 를 생성한다.

6 GitHub> Repositories > Clone or download

생성된 Repo 의 주소 확인한다. SSH 를 사용하는 게 암호입력을 안 해도 되서 편하다.

## BASIC

```
$ git clone <repository>
```

```
$ git log [--graph] [HEAD]
```

```
$ git log [--graph] [HEAD]
```

```
$ git status
```

```
$ git fetch [origin [[+]master[:master]]]
```

```
$ git fetch = $ git fetch <origin>
```

```
$ git pull [origin [[+]master[:master]]]
```

```
$ git pull = $ git fetch; git merge <origin/master>
```

```
$ git push [origin [[+]master[:master]]]
```

```
$ git push = $ git push <origin> :
```

## BRANCH

```
$ git branch [-a] [-vv]
```

```
$ git branch <newbranch>
```

```
$ git branch -d <branchname>
```

```
$ git checkout <branchname>
```

```
$ git push -u <origin> <branchname>[:<branchname>]
```

```
$ git push <origin> :<branchname>
```

## MERGE

```
$ git merge <commit>
```

---

```
      A—B—C topic  
      /  
D—E—F—G master
```

```
$ git checkout master
```

```
$ git merge topic
```

```
      /———A—B—C topic  
      /      \  
D—E—F—G—————H master
```

## CONFLICT CONTINUE

```
$ git commit -a -m "commit message"
```

## CONFLICT ABORT

```
$ git merge --abort
```

## UNDO COMMIT

```
$ git reset --hard HEAD^
```

## REBASE

```
$ git rebase [--onto <newbase>] <upstream> [<branch>]
```

```
      A—B—C topic
     /
D—E—F—G master
```

```
$ git checkout topic
```

```
$ git rebase master
```

```
      A'—B'—C' topic
     /
D—E—F—G master
```

```
      A—B—C  topic
     /
D—E—F—G—H next
   /
I—J—K—L—M  master
```

```
$ git rebase --onto master next topic
```

```
$ git rebase --onto master topic~3 topic
```

```
      D—E—F—G—H  next
     /
    /
   /
  /
I—J—K—L—M  master
      /
     A'—B'—C'  topic
```

## CONFLICT CONTINUE

```
$ git add --all
```

```
$ git rebase --continue
```

## CONFLICT ABORT

```
$ git rebase --abort
```

## REWRITING HISTORY

```
$ git rebase -i master
```

```
pick    f7f3f6d changed my name a bit
```

```
squash 310154e updated README formatting and added blame
```

```
squash a5f4a0d added cat-file
```

## SUBMODULE

```
$ git submodule add https://github.com/sj6219/sub_project.git sub_project
```

```
$ git commit -m "sub merge"
```

```
$ git clone https://github.com/sj6219/super_project.git
```

```
$ cd super_project
```

```
$ git -c diff.mnemonicprefix=false -c core.quotepath=false --no-optional-locks submodule update  
--init
```

```
$ git pull
```

```
$ git submodule update
```