

1장 Git 설치

1. Git 설치

<https://git-scm.com/> --> Download --> Git-2.28.0-64-bit.exe

2. Git 커맨드 라인 명령어(CLI: command line Instruction)

- (1) 저장소 생성: git init
- (2) 저장소에 파일 추가: git add <filename>
- (3) 저장소에 변경 내용의 반영: git commit
- (4) 저장소의 상태 확인 :git status

실습1)

C:\Users\User>	mkdir test_git	test_git 폴더 생성
C:\Users\User>	cd test_git	test_git 폴더로 이동
C:\Users\User>test_git>	git init	새로운 repository 만듦
C:\Users\User>test_git>	dir	보이지 않음 (맥은 보임)
C:\Users\User>test_git>	notepad hello.html	메모장에서 아래 작업하시오

```
<html>
<body>
  mast -- Hello World
</body>
</html>
```

실습2)

C:\Users\User>test_git>	git status	상태확인:빨강색은 추가 안된 상태
C:\Users\User>test_git>	git add hello.html	파일추가
C:\Users\User>test_git>	git status	상태확인:녹색은 추가된 상태
C:\Users\User>test_git>	git commit -m "test"	변경된 내용 완료, 메시지 "test"로 남김
C:\Users\User>test_git>	git status	상태확인
C:\Users\User>test_git>	git log	log 확인



실습3)

C:\Users\User>test_git>	git commit -m "branch test1"	
C:\Users\User>test_git>	git checkout master	
C:\Users\User>test_git>	notepad hello.html	이전 내용을 확인
C:\Users\User>test_git>	git checkout test1	
C:\Users\User>test_git>	notepad hello.html	변경 내용을 확인
C:\Users\User>test_git>	git checkout master	
C:\Users\User>test_git>	git merge test1	test1을 master로 병합
C:\Users\User>test_git>	notepad hello.html	변경된 내용으로 변경
C:\Users\User>test_git>	git branch -d test1	test1 브랜치를 삭제

C:\Users\User>test_git>	git branch	
C:\Users\User>test_git>	git branch test2	
C:\Users\User>test_git>	git branch	
C:\Users\User>test_git>	git checkout test2	
C:\Users\User>test_git>	git branch	
C:\Users\User>test_git>	notepad hello.html	
C:\Users\User>test_git>	git add hello.html	
C:\Users\User>test_git>	git commit -m "Edit hello.html on test2"	
C:\Users\User>test_git>	git checkout master	
C:\Users\User>test_git>	notepad hello.html	내용 변경하시오
C:\Users\User>test_git>	git add hello.html	
C:\Users\User>test_git>	git commit -m "Edit hello.html on Master"	
C:\Users\User>test_git>	git merge test2	
C:\Users\User>test_git>	git status	
C:\Users\User>test_git>	notepad hello.html	내용 변경하시오
C:\Users\User>test_git>	git add hello.html	
C:\Users\User>test_git>	git commit -m "Edit hello.html on Master"	
C:\Users\User>test_git>	git merge test2	
C:\Users\User>test_git>	git status	두 상태 모두 변경됨

```

선택 명령 프롬프트
1 file changed, 1 insertion(+)
C:\Users\User>test_git>git checkout master
Switched to branch 'master'
C:\Users\User>test_git>notepad hello.html
C:\Users\User>test_git>git add hello.html
C:\Users\User>test_git>git commit -m "edit hello.html on master 22"
[master 4e59402] edit hello.html on master 22
1 file changed, 1 insertion(+), 1 deletion(-)
C:\Users\User>test_git>git merge test2
Auto-merging hello.html
CONFLICT (content): Merge conflict in hello.html
Automatic merge failed; fix conflicts and then commit the result.
C:\Users\User>test_git>git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   hello.html

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

C:\Users\User>test_git>	notepad hello.html	상단과 하단을 비교하며 수정할것
C:\Users\User>test_git>	git add hello.html	
C:\Users\User>test_git>	git commit -m "Final hello.html"	
C:\Users\User>test_git>	git log	q 로 빠져나옴
C:\Users\User>test_git>	gitk	시각적으로 확인

2장 깃과 깃허브 사용법(크롬브라우저 이용)

1. 깃허브 가입하기(<https://github.com>)

가입시 Username → 깃허브URL 인 “<http://github.com/Username>” 으로 사용된다.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Username

Owner * Repository name *

Great repository names are [HelloWorld is available](#). Need inspiration? How about furry-octo-robot?

Description (optional)

☒ Public
Anyone on the internet can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: [None](#) Add a license: [None](#)

Create repository

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

choimysil / HelloWorld

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH

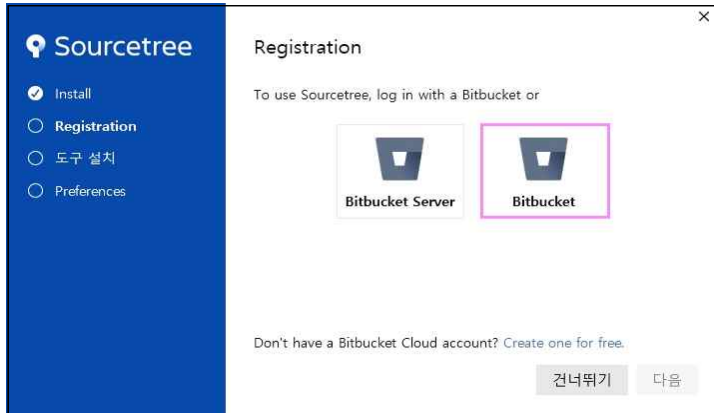
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

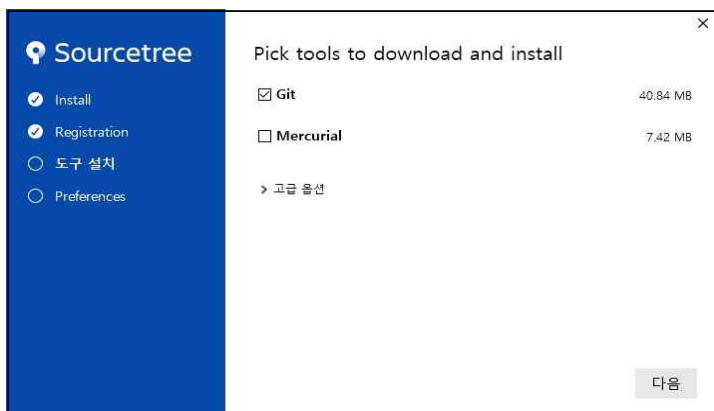
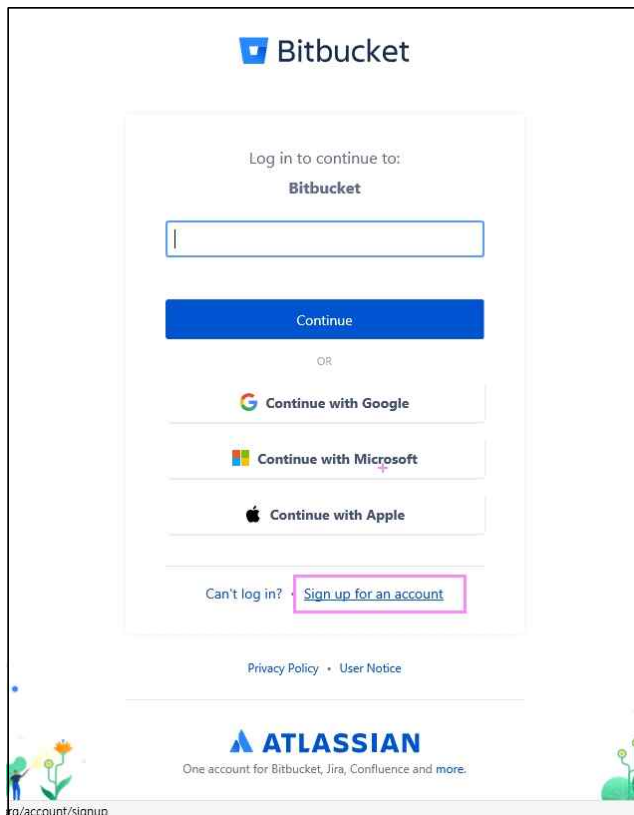
```
echo "# HelloWorld" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/choimysil/HelloWorld.git
git push -u origin master
```

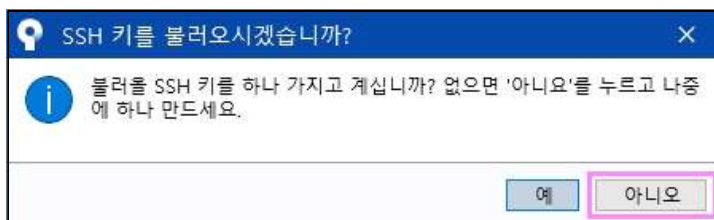
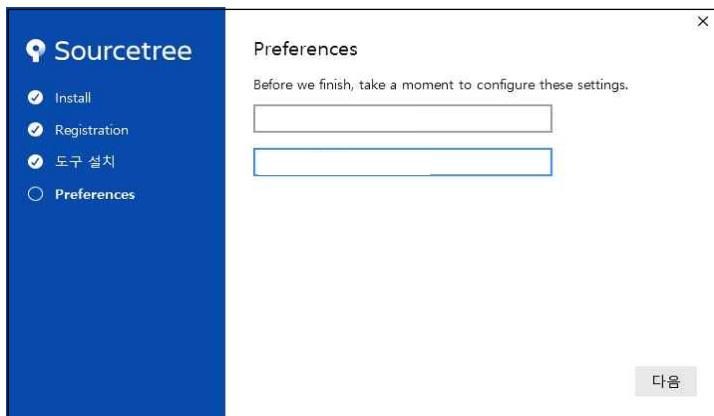
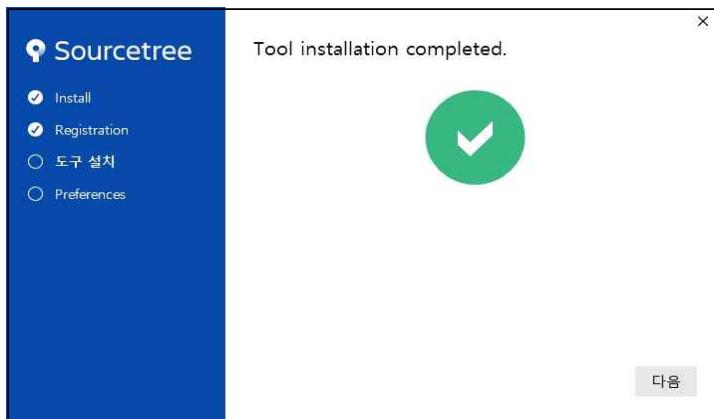
3장 깃(git) :버전 관리 시스템

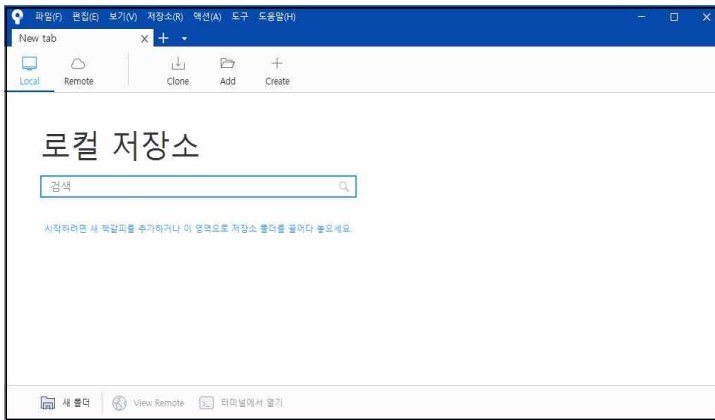
CLI를 대신하기 위한 관리 툴로 Atlassian에서 만든 SourceTree 활용 →
(<https://www.sourcetreeapp.com>) 다운로드후 설치



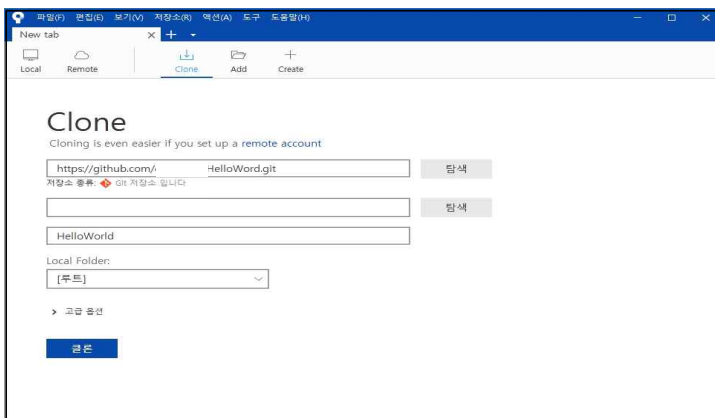
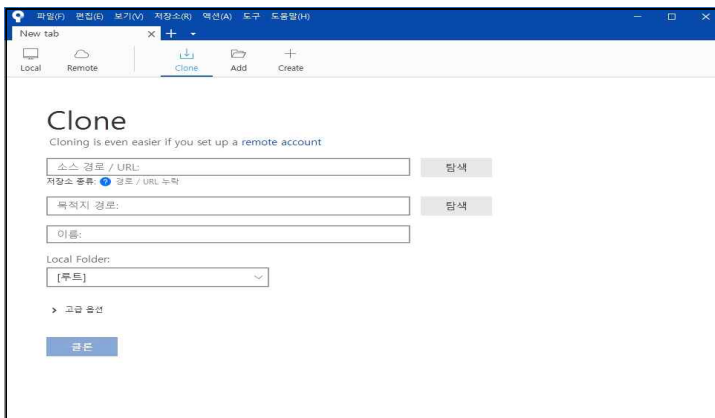
1. 아틀라시안에 가입후 mail인증







2. [파일]-[복제/생성]



3. 동작

- (1) commit (묶어서 저장)
- (2) add (스테이지에 올림)
- (3) push (github 사이트에 올림)

4. 환경설정

