

## 버전관리시스템(VCS)

: 파일의 변화를 시간에 따라 기록한 뒤, 특정 시점의 버전을 사용할 수 있게 하는 시스템  
원하는 시점으로 복구하여 프로젝트에서 파일의 잘못된 수정이나 분실에 대처할 수 있음

## 로컬 버전관리 시스템(RCS)

: 파일이 변경되는 부분(Patch set)을 기록  
기록된 Patch Set을 이용해 파일을 특정 시점으로 되돌림



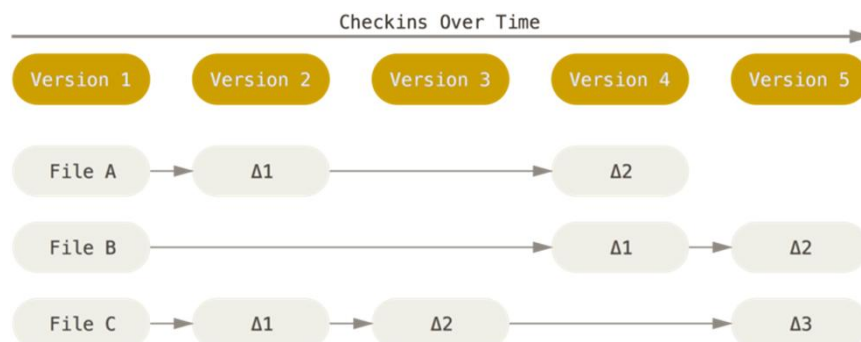
## 중앙집중식 버전관리 시스템(CVCS)

: 다른 개발자와 함께 작업하기 위한 버전관리 시스템  
중앙 데이터베이스 서버를 통해 관리하므로 협업이나 관리가 편함  
중앙 서버에 문제가 발생할 경우, 프로젝트의 모든 변경사항을 잃을 수 있음



## 분산 버전 관리 시스템(DVCS)

: 클라이언트에 저장소와 히스토리를 전부 복제하는 버전관리 시스템  
각 개발자가 중앙 서버에 접속하지 않고 코드 작업을 할 수 있음  
서버에 문제가 생길 경우, 클라이언트에 Clone된 데이터로 서버 복원 가능



## Git

프로젝트 등에서 소스코드 관리를 위한 분산버전 관리 시스템

변경되지 않은 파일을 새로 저장하는 것이 아니라 그 파일에 대한 링크만을 저장

빠른 수행속도에 중점, 데이터 무결성, 분산, 비선형 워크플로를 지원

로컬에서 대부분의 명령을 실행, 오프라인이나 VPN 연결 없이 프로젝트 진행가능

Git에서는 파일을 3가지 상태로 관리

- Committed : 데이터가 로컬 데이터베이스에 저장된 상태
- Modified : 수정된 파일이 로컬 데이터베이스에 저장되지 않은 상태
- Staged : 현재 수정한 파일을 저장할 것으로 표시된 상태

## Git repository 관리

Github <https://github.com/>

Git을 지원하는 웹 호스팅 서비스

Git 호스팅 사이트 중 가장 많은 사용자를 보유

많은 사람이 참여하는 오픈소스 프로젝트에 적합

유료 계정으로 비공개 Private 저장소 사용 가능

1GB로 용량이 제한되어 있음

Gitlab <https://about.gitlab.com/>

Git의 원격 저장소 기능과 이슈 트래커 기능등을 제공

웹 기반의 데브옵스 시스템

자체 호스팅을 통해 코드 비공개 유지

Private 저장소 무료로 사용 가능

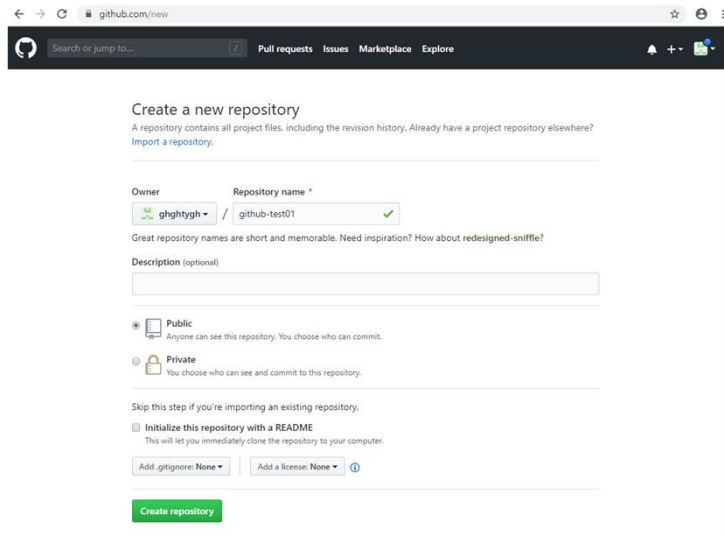
프로젝트 관리를 위한 CI 시스템 제공

리포지토리의 Push Pull 동작이 Github에 비해 느림

인터페이스 페이지 전환에 시간이 걸림

## GitHub

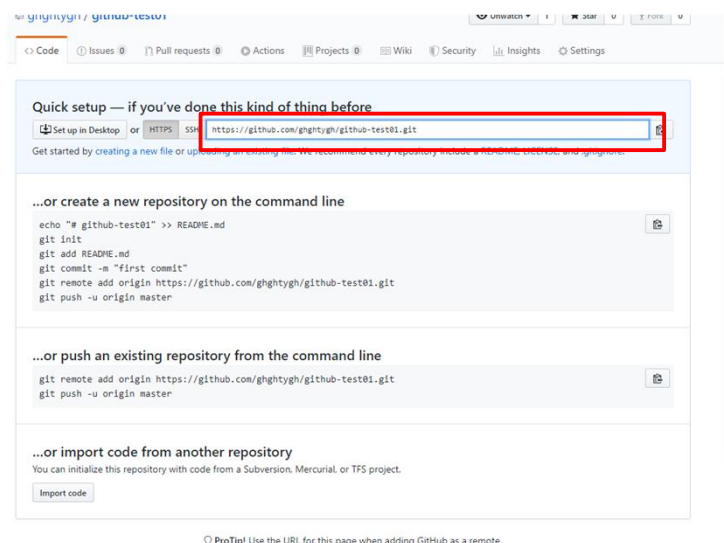
### 원격 저장소 생성



Github.com에 접속하여 로그인한 뒤,

우측 상단 + 버튼을 통해

새로운 저장소 생성



네모 칸의 주소를 통해 원격 저장소의 주소를 복사

## GitHub

### 로컬 저장소와 원격 저장소 연결



원하는 디렉토리에서

마우스 우클릭을 통해 Git Bash Here 실행

```
MINGW64/c/Users/vinea/Desktop/github-test01
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01
$ git init
Initialized empty Git repository in C:/Users/vinea/Desktop/github-test01/.git/
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01 (master)
$ git clone https://github.com/ghgtygh/github-test01.git
Cloning into 'github-test01'...
warning: You appear to have cloned an empty repository.
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01 (master)
$ |
```

git init

git 시작

git clone [원격저장소 주소] 로컬저장소에  
원격저장소 주소 복사

### 로컬 저장소 변경 내용 - 원격 저장소에 반영

```
MINGW64/c/Users/vinea/Desktop/github-test01/github-test01
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.MD
nothing added to commit but untracked files present (use "git add" to track)
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ git add .
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   README.MD
```

git status

변경 내용 확인

git add . 모든 파일 Tracked 전환(등록)  
git add [파일이름] 해당 파일 Tracked 전환

```
MINGW64/c/Users/vinea/Desktop/github-test01/github-test01
No commits yet
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   README.MD
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ git commit -m 'add readme'
[master (root-commit) 39d9812] add readme
1 file changed, 1 insertion(+)
 create mode 100644 README.MD
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 215 bytes | 215.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ghgtygh/github-test01.git
 * [new branch] master -> master
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ |
```

git commit -m ['메시지내용'] 변경 내용  
로컬 저장소에 저장

git push origin master 원격 저장소에 반영  
git push origin [브랜치명] 해당 branch에 반영

## GitHub

### 원격 저장소 변경내용 - 로컬 저장소에 반영

```
MINGW64/c:/Users/vinea/Desktop/github-test01/github-test01
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/ghghygh/github-test01
* branch      master      -> FETCH_HEAD
   39d9812..00ec4dd master  -> origin/master
Updating 39d9812..00ec4dd
Fast-forward
 README.MD | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)

vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/github-test01/github-test01 (master)
$ |
```

git pull origin master

원격저장소의 소스를  
불러와 로컬 저장소에 병합

git pull origin [브랜치명]

### 브랜치(Branch) 생성

```
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (master)
$ git branch b1

vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (master)
$ git branch
  b1
* master
```

git branch [브랜치이름]    branch 생성

git branch

생성된 branch 확인

```
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (master)
$ git checkout b1
Switched to branch 'b1'

vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (b1)
$ |
```

git checkout [브랜치이름]    branch 전환

### 브랜치(Branch) 병합

```
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (b1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (master)
$ git merge b1
Updating ee38c82..234f265
Fast-forward
 b1.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 b1.txt
```

git checkout master

master로 전환

git merge [브랜치이름]

branch 병합

### 브랜치(Branch) 삭제

```
vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (master)
$ git branch -d b1
Deleted branch b1 (was 234f265).

vinea@DESKTOP-HH12F4D MINGW64 ~/Desktop/git_lab/github-test01 (master)
$ git branch
* master
```

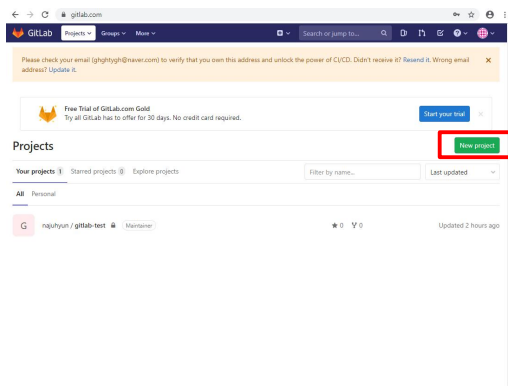
git branch -b [브랜치이름] 병합 끝난 branch 삭제

git branch

삭제되었는지 확인

## GitLab

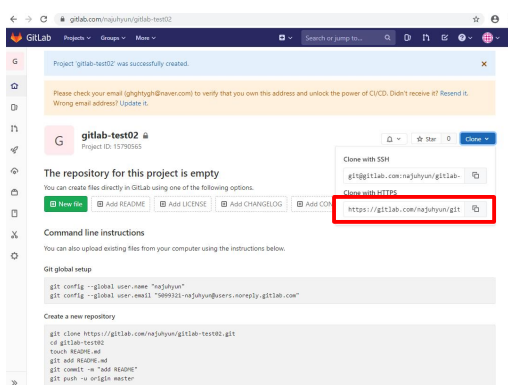
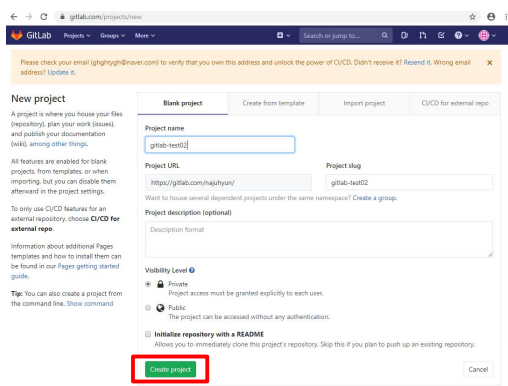
### 새로운 프로젝트 생성



<http://gitlab.com>에 접속하여 로그인한 뒤,

우측 New project 버튼을 통해

새로운 프로젝트 생성



네모 칸의 주소를 통해 원격 저장소의 주소를 복사

나머지 동작은 GitHub와 동일