

# Git / GitHub

2014년 12월 29일  
경희대학교 컴퓨터공학과  
한상윤

# 목차

- Git / GitHub 소개
- Git 설치 및 설정
- Git 사용법
- add와 commit
- GitHub(원격저장소) 연결

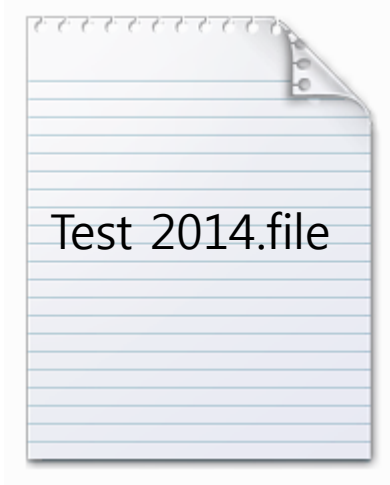
# Git이란?



- 프로젝트(소스코드, 리소스 등)를 관리하기 위한 분산 버전 관리 시스템(DVCS)
- 네트워크에 접근하거나 중앙 서버에 의존하지 않는 완전한 형태의 저장소
- 리누즈 토발즈가 리눅스 커널 개발에 이용하려고 최초 개발 및 사용

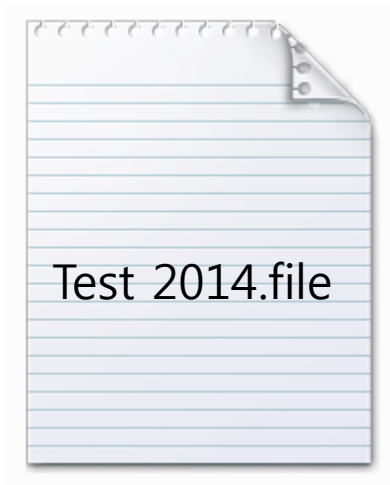


# 문서 or 소스코드 등의 수정 내역을 기록



```
#include <iostream>
using namespace std;

int main()
{
    cout << "test 2014" << endl;
    return 0;
}
```

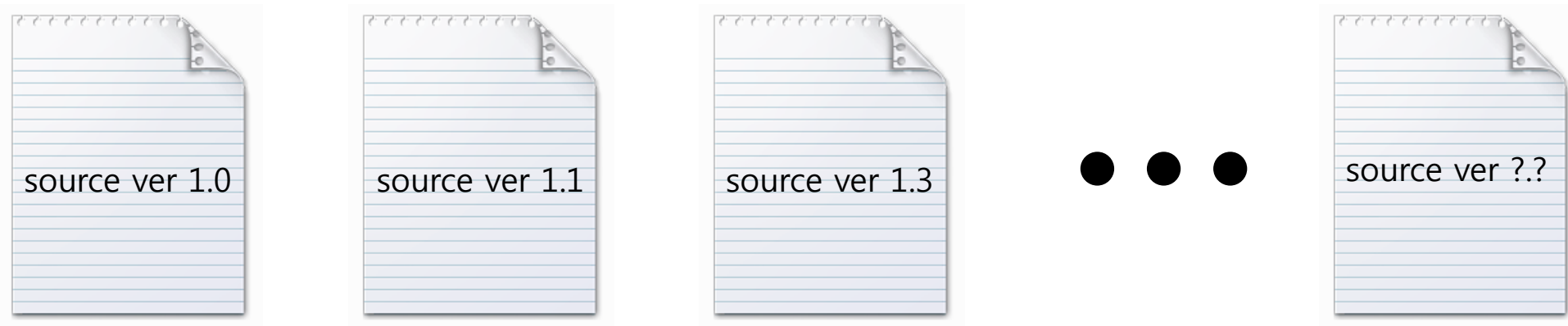


```
#include <iostream>
using namespace std;

int main()
{
    cout << "test 2015" << endl;
    return 0;
}
```

서로 다른 버전

수정할 때마다 다른 이름의 파일로 저장한다면?



수정하는 사람이 여러 명이라면?

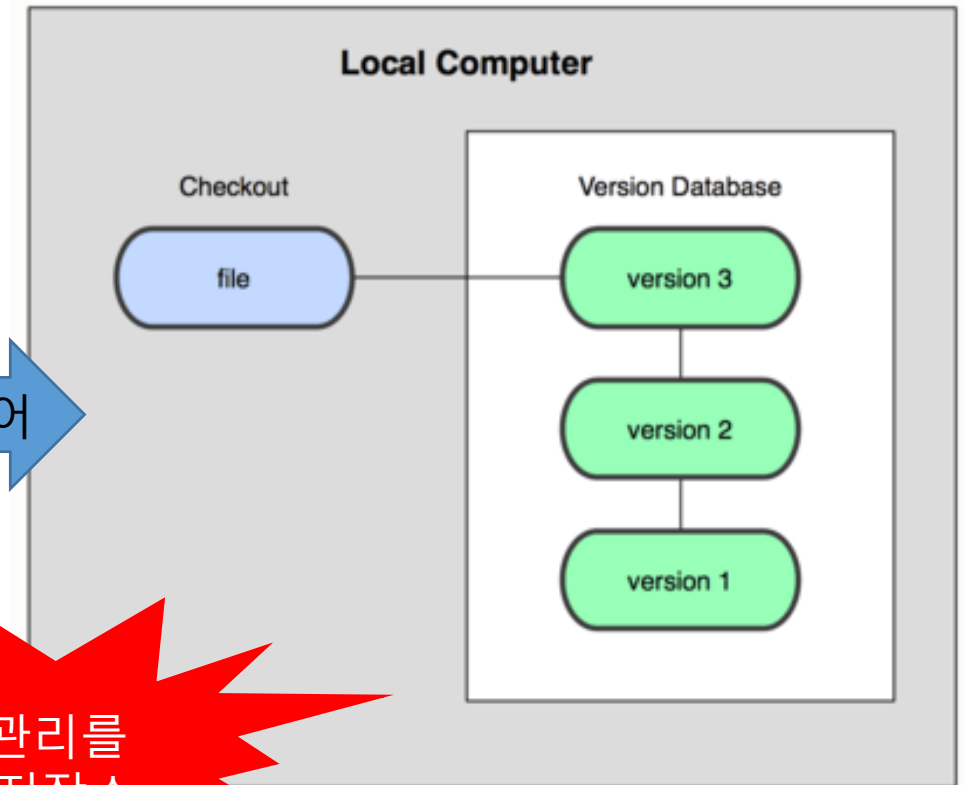
## 무식한 버전 관리의 폐해



# 버전 관리 시스템

ReUnited_Chord4Mame-0815 2032.zip	2013-08-15 오후...
ReUnited_Chord4Mame-0816 0200.zip	2013-08-16 오전...
ReUnited_Chord4Mame-0816 1416.zip	2013-08-17 오후...
ReUnited_Chord4Mame-0817 1710.zip	2013-08-17 오후...
ReUnited_Chord4Mame-0817 2248.zip	2013-08-18 오전...
ReUnited_Chord4Mame-0818 0100.zip	2013-08-18 오전...
ReUnited_Chord4Mame-0818 0131.zip	2013-08-18 오전...
ReUnited_Chord4Mame-0818 1732.zip	2013-08-18 오후...
ReUnited_Chord4Mame-0818 1900.zip	2013-08-18 오후...
ReUnited_Chord4Mame-0820 1641.zip	2013-08-20 오후...
ReUnited_Chord4Mame-0820 1817.zip	2013-08-20 오후...
ReUnited_Chord4Mame-0820 1931.zip	2013-08-21 오후...
ReUnited_Chord4Mame-0822 2200.zip	2013-08-22 오후...
ReUnited_Chord4Mame-0824 0000.zip	2013-08-24 오후...
ReUnited_Chord4Mame-0824 1543.zip	2013-08-24 오후...
ReUnited_Chord4Mame-0824 1643.zip	2013-08-24 오후...
ReUnited_Chord4Mame-0824 1943.zip	2013-08-24 오후...
ReUnited_Chord4Mame-0825 0300.zip	2013-08-25 오전...
ReUnited_Chord4Mame-0826 0700.zip	2013-08-26 오후...
ReUnited_Chord4Mame-0827 1454.zip	2013-08-27 오후...
ReUnited_Chord4Mame-0827 1941.zip	2013-08-27 오후...
ReUnited_Chord4Mame-0827 2131.zip	2013-08-27 오후...
ReUnited_Chord4Mame-0828 1216.zip	2013-08-28 오전...
ReUnited_Chord4Mame-0829 1524.zip	2013-08-29 오후...
ReUnited_Chord4Mame-0831 1140.zip	2013-08-31 오후...
ReUnited_Chord4Mame-0901 0501.zip	2013-09-01 오후...
ReUnited_Chord4Mame-0901 1244.zip	2013-09-01 오후...
ReUnited_Chord4Mame-0901 1917.zip	2013-09-02 오후...
ReUnited_Chord4Mame-cracked.zip	2013-09-02 오후...
ReUnited_Chord4Mame-final-1.0.2.zip	2013-09-02 오후...
ReUnited_Chord4Mame-final-1.0.3.zip	2013-09-03 오전...
ReUnited_Chord4Mame-first.zip	2013-08-15 오후...

버전 관리를 위한 소프트웨어



이력관리를  
위한 저장소



# Git이란?



버전 관리 시스템? OK  
그럼 분산은?

- 프로젝트(소스코드, 리소스 등)를 관리하기 위한 분산 **버전 관리 시스템(DVCS)**
- 네트워크에 접근하거나 중앙 서버에 의존하지 않는 완전한 형태의 저장소
- 리누즈 토발즈가 리눅스 커널 개발에 이용하려고 최초 개발 및 사용

# Git이란?

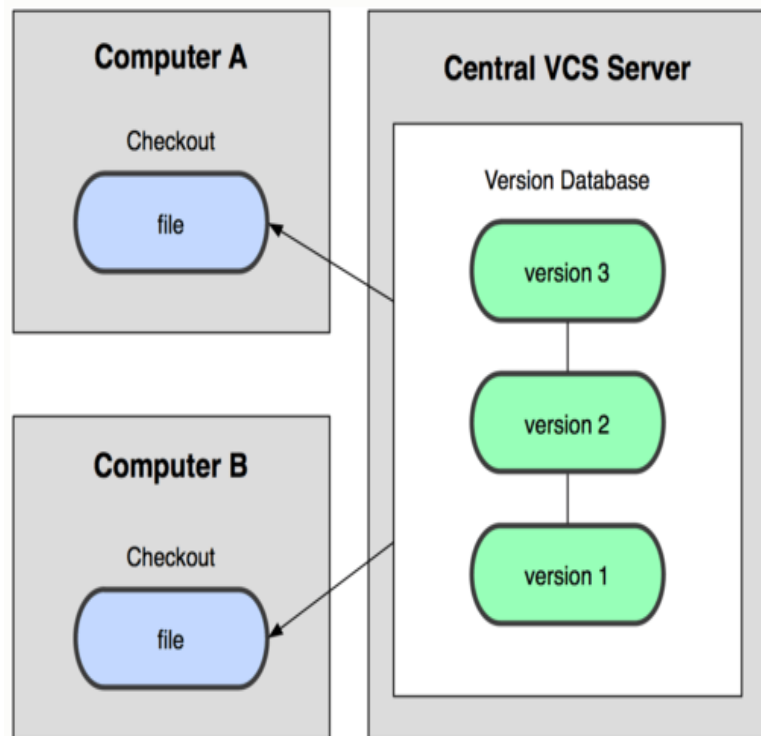


버전 관리 시스템? OK  
그럼 분산은?

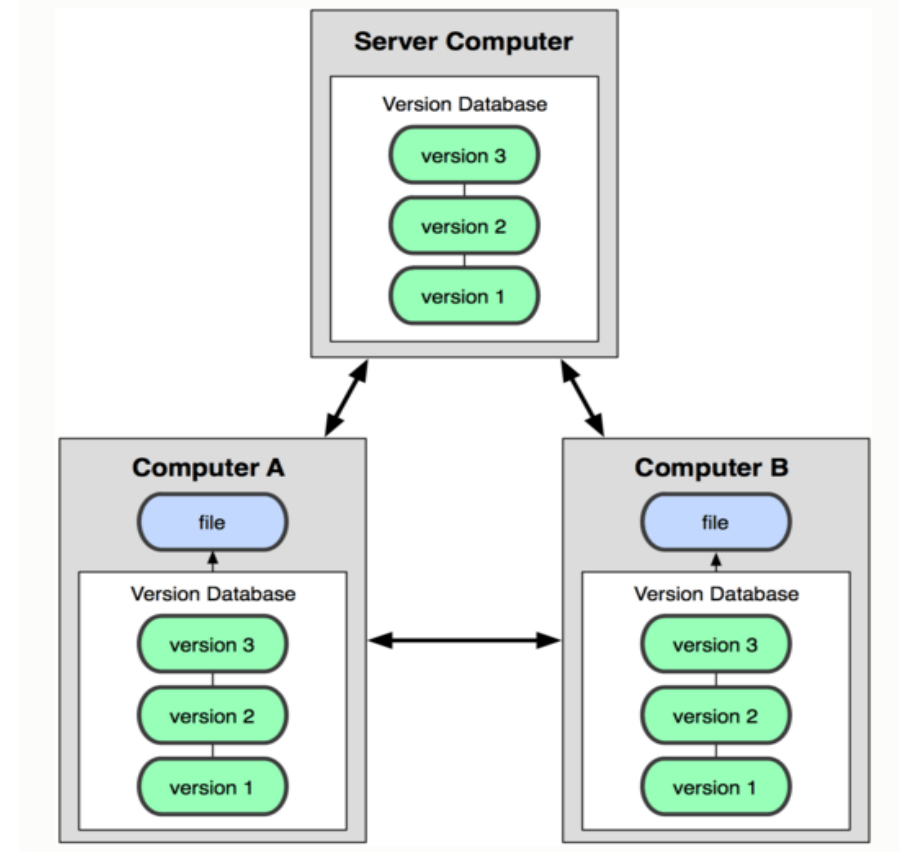
- 프로젝트(소스코드, 리소스 등)를 관리하기 위한 **분산 버전 관리 시스템(DVCS)**
- 네트워크에 접근하거나 중앙 서버에 의존하지 않는 완전한 형태의 저장소
- 리누즈 토발즈가 리눅스 커널 개발에 이용하려고 최초 개발 및 사용

# 분산 버전 관리 시스템

→ 로컬 저장소를 가지며 이를 가지고 작업 가능



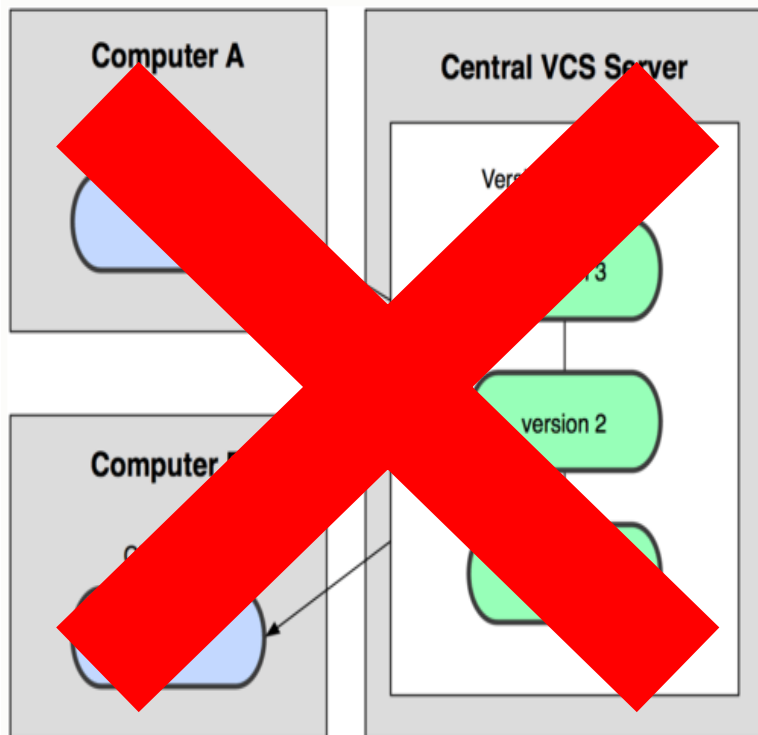
중앙 집중식 버전 관리 시스템



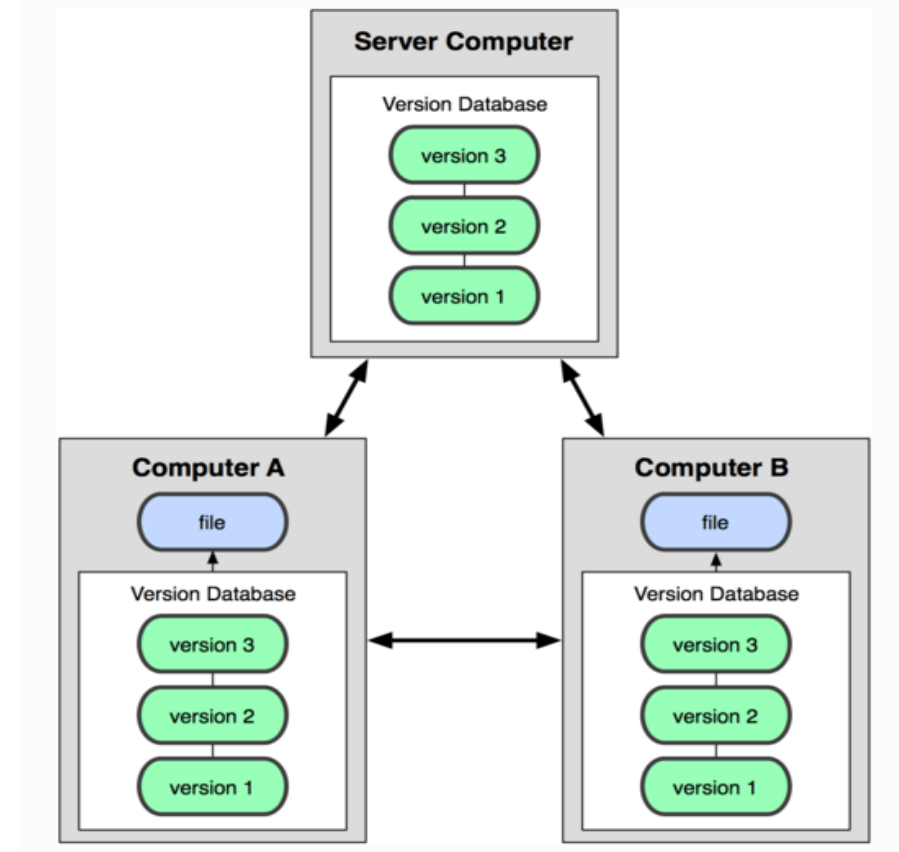
분산 버전 관리 시스템

# 분산 버전 관리 시스템

→ 로컬 저장소를 가지며 이를 가지고 작업 가능



중앙 집중식 버전 관리 시스템



분산 버전 관리 시스템

Git?  
분산 버전 관리 시스템!!  
Ok!!



Github는 뭐지?

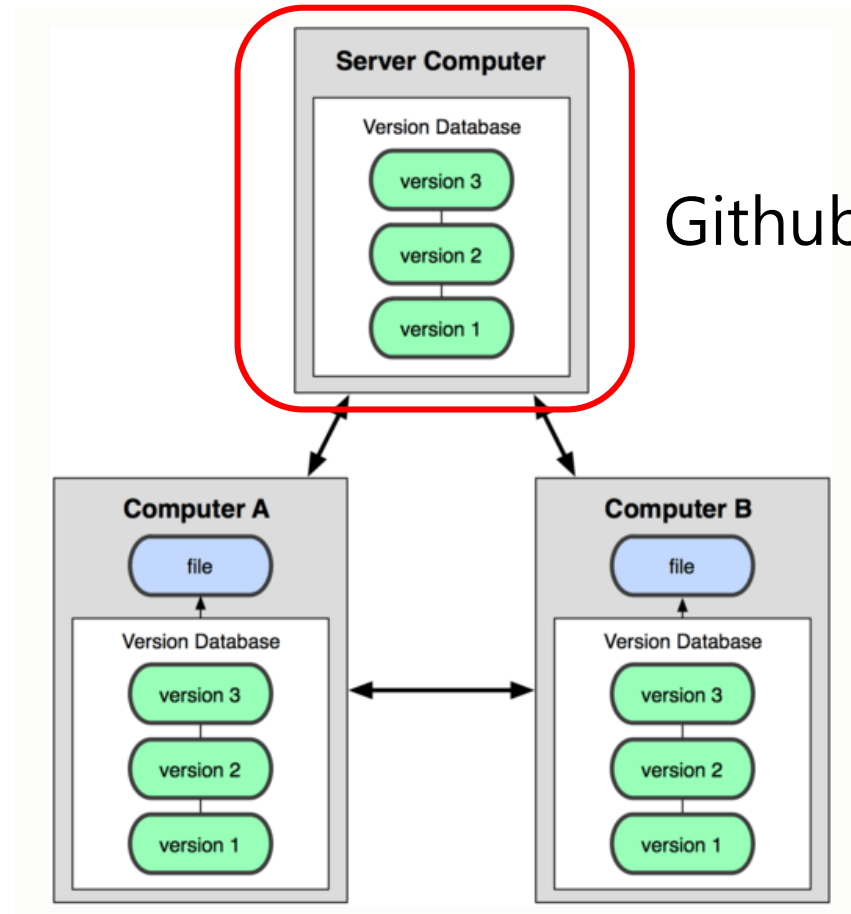


# GitHub



- git을 호스팅해주는 웹서비스
- Git 저장소 서버를 대신 유지 및 관리해주는 서비스
- 오픈소스 프로젝트 = 무료 / private 프로젝트 = 유료

# Git 원격 저장소를 제공



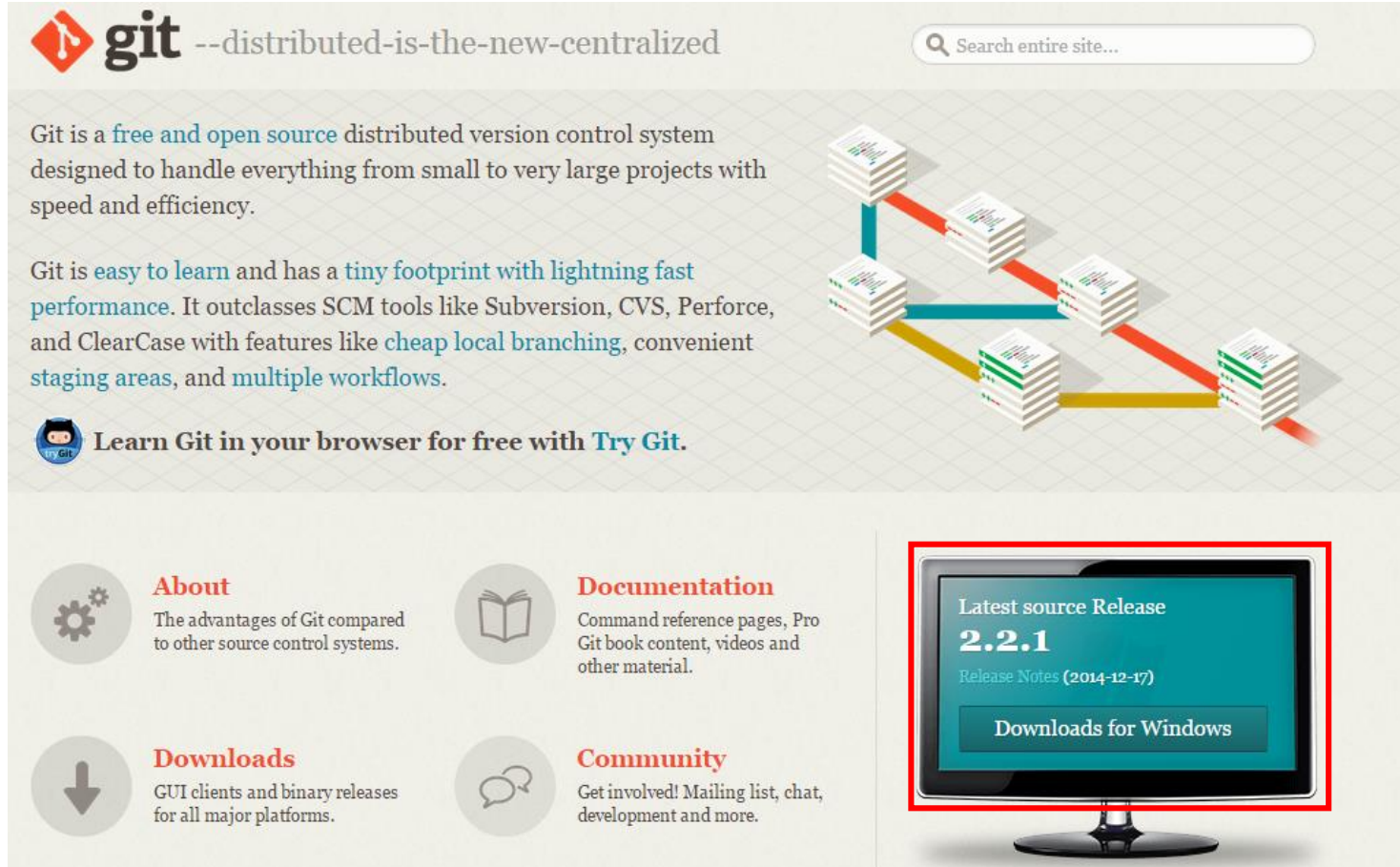
Github의 역할

유사 서비스 :  
bitbucket



# Git 설치 및 설정


# Windows 설치



**git** --distributed-is-the-new-centralized

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint](#) with [lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), convenient [staging areas](#), and [multiple workflows](#).

 Learn Git in your browser for free with [Try Git](#).

**About**  
The advantages of Git compared to other source control systems.

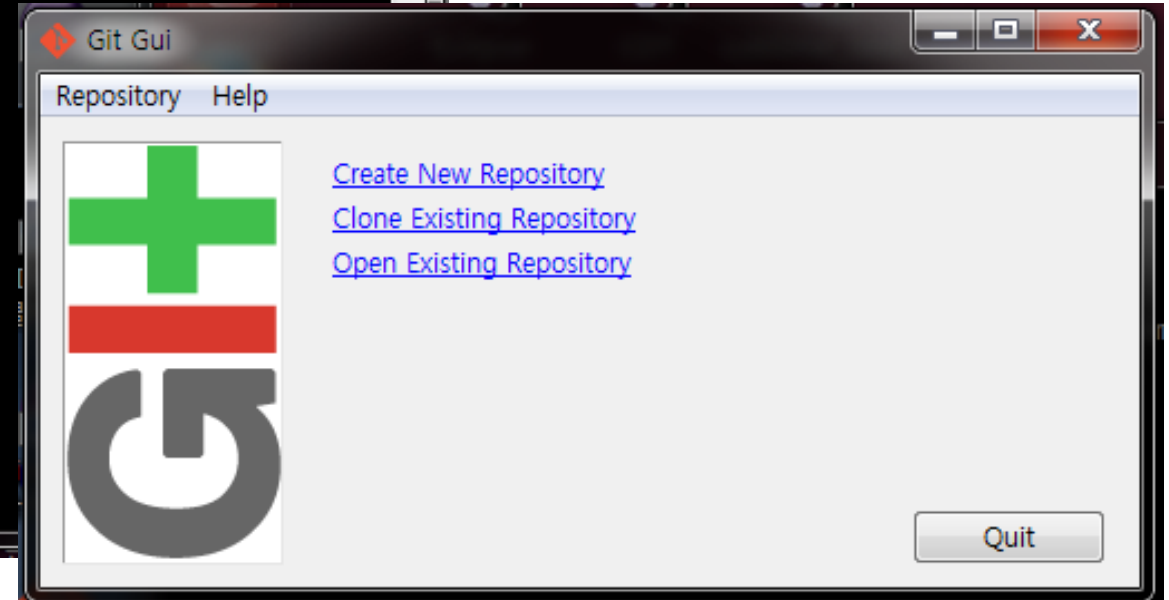
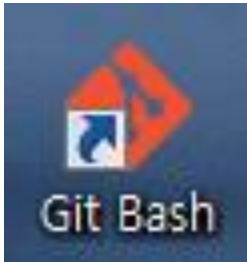
**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Mailing list, chat, development and more.

Latest source Release  
**2.2.1**  
[Release Notes \(2014-12-17\)](#)  
[Downloads for Windows](#)

# Windows 터미널 프로그램



# Linux 설치

1. `sudo apt-get install git-core git-doc`

```
hansangyun@hansangyun:~$ sudo apt-get install git-core git-doc
```

2. `git --version`

```
hansangyun@hansangyun:~$ git --version  
git version 1.9.1
```

3. 설치 완료

# Git 사용

1. 저장소 생성
2. 사용자 정보 입력
3. 변경 적용
4. 프로젝트 상태 확인

# 저장소 생성하기(명령어 : **git init**)

1. mkdir git-test
2. cd git-test
3. git init

```
hansangyun@hansangyun:~/바탕화면$ mkdir git-test
hansangyun@hansangyun:~/바탕화면$ cd git-test
hansangyun@hansangyun:~/바탕화면/git-test$ git init
Initialized empty Git repository in /home/hansangyun/바탕화면/git-test/.git/
```

# 저장소 받아오기(명령어 : **git clone**)

1. 로컬 저장소에서 받아오기  
`git clone /로컬/저장소/경로`

2. 원격 저장소에서 받아오기  
`git clone 사용자명@호스트:/원격/저장소/경로`  
`git clone 저장소 주소(ex: git clone git://github.com/sangyunHan/kairos-xxx)`

```
hansangyun@hansangyun:~/바탕화면$ git clone git://github.com/sangyunHan/kairos-xxx
Cloning into 'kairos-xxx'...
remote: Counting objects: 300, done.
remote: Total 300 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (300/300), 249.51 KiB | 57.00 KiB/s, done.
Resolving deltas: 100% (47/47), done.
Checking connectivity... done.
```

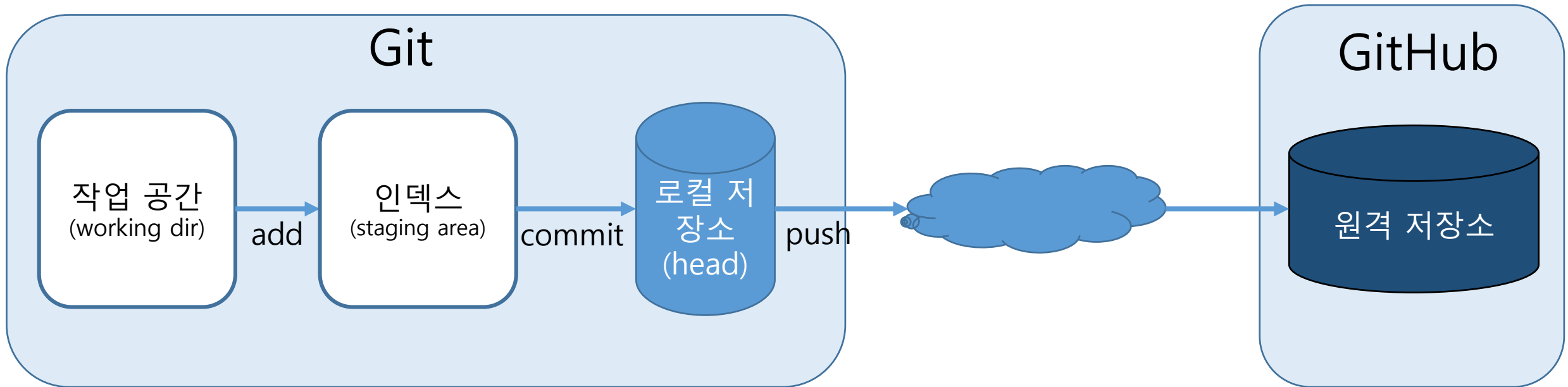
# 커미터 등록

- Git을 사용하는 사용자에게 대한 정보를 git에 등록한다.
- 공동 작업을 할 때 각각의 버전이 누구의 작업인지 구별해주는 역할
- `git config --global user.name "사용자명"`
- `git config --global user.email "사용자 이메일"`

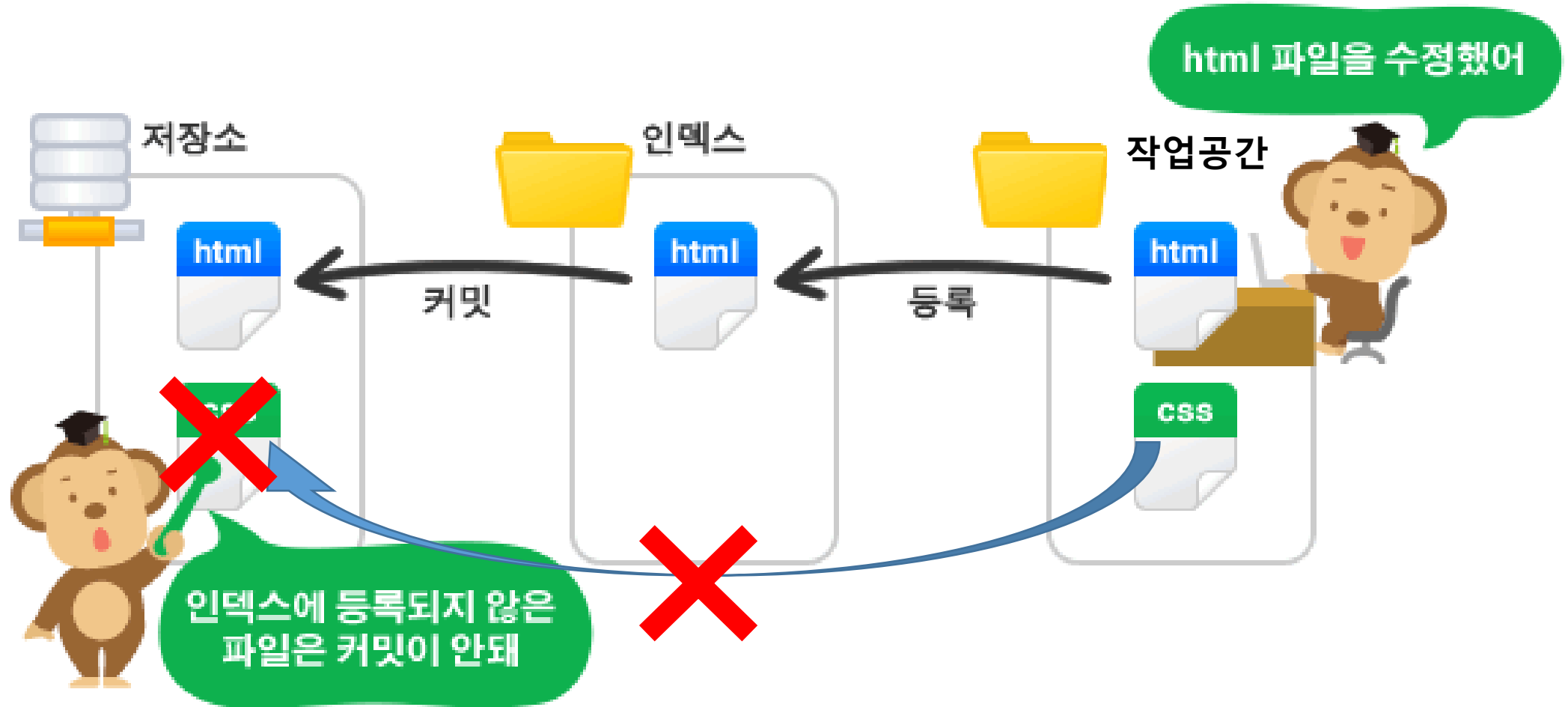
```
hansangyun@hansangyun:~/바탕화면/git-test$ git config --global user.name "sangyunHan"  
hansangyun@hansangyun:~/바탕화면/git-test$ git config --global user.email "sangyun0628@naver.com"
```



# 작업의 흐름



# 인덱스의 중요성!



# add와 commit

git status

git add

git commit

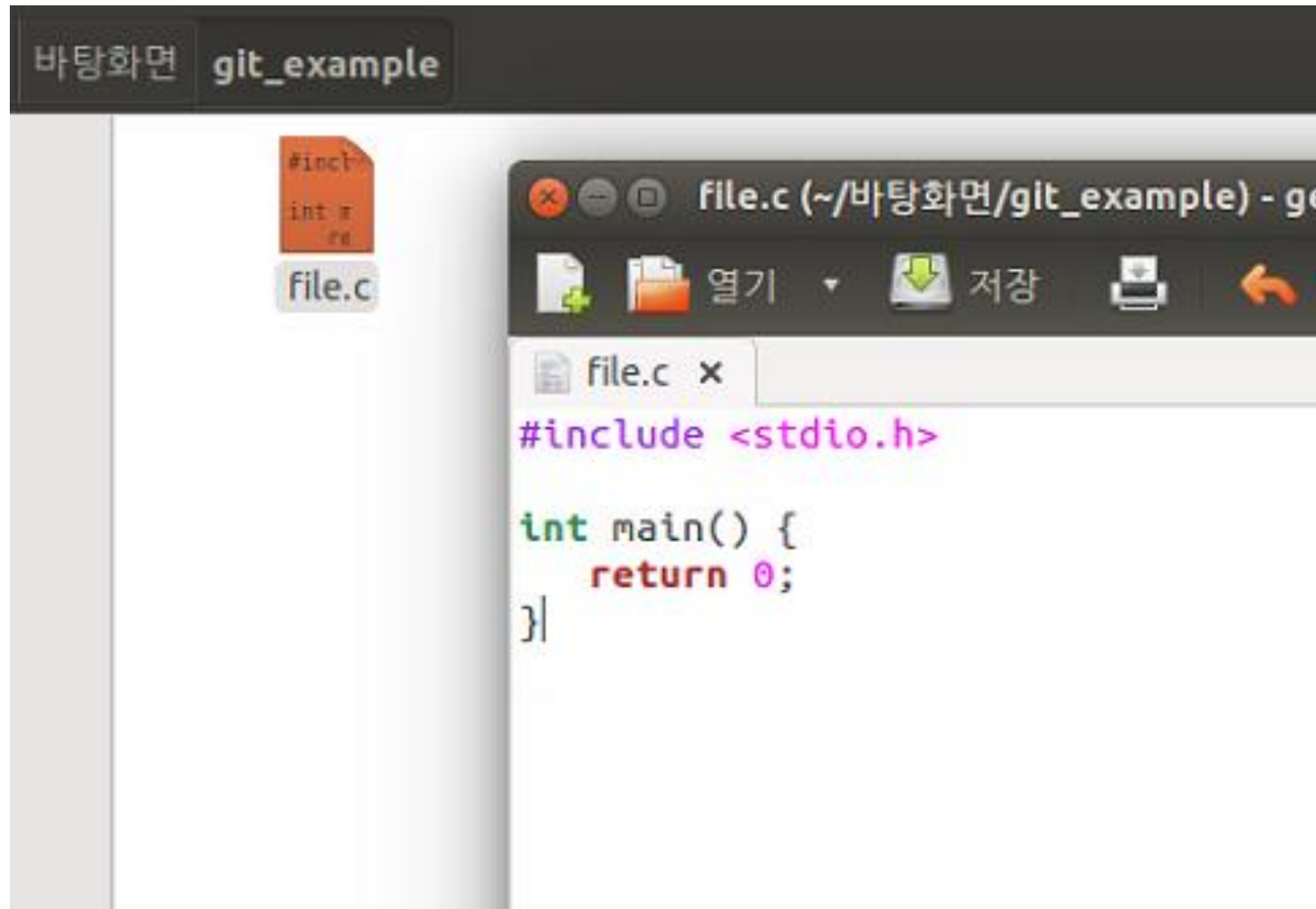
# 상태보기(명령어 : git status)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git init
Initialized empty Git repository in /home/hansangyun/바탕화면/git_example/.git/
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master

Initial commit

nothing to commit (create/copy files and use "git add" to track)
```

# 파일 추가



# 파일 추가 후 상태

```
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.c
        file.c~

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

# 파일 추가 후 상태

```
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    file.c
    file.c~

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

인덱스(staging area)에 등록되지 않아서 버전관리가 되고 있지 않은 파일들

# add : 인덱스에(staging area) 등록

- git add 파일명(ex: git add file.c)

```
hansangyun@hansangyun:~/바탕화면/git_examples$ git add file.c
hansangyun@hansangyun:~/바탕화면/git_examples$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   file.c

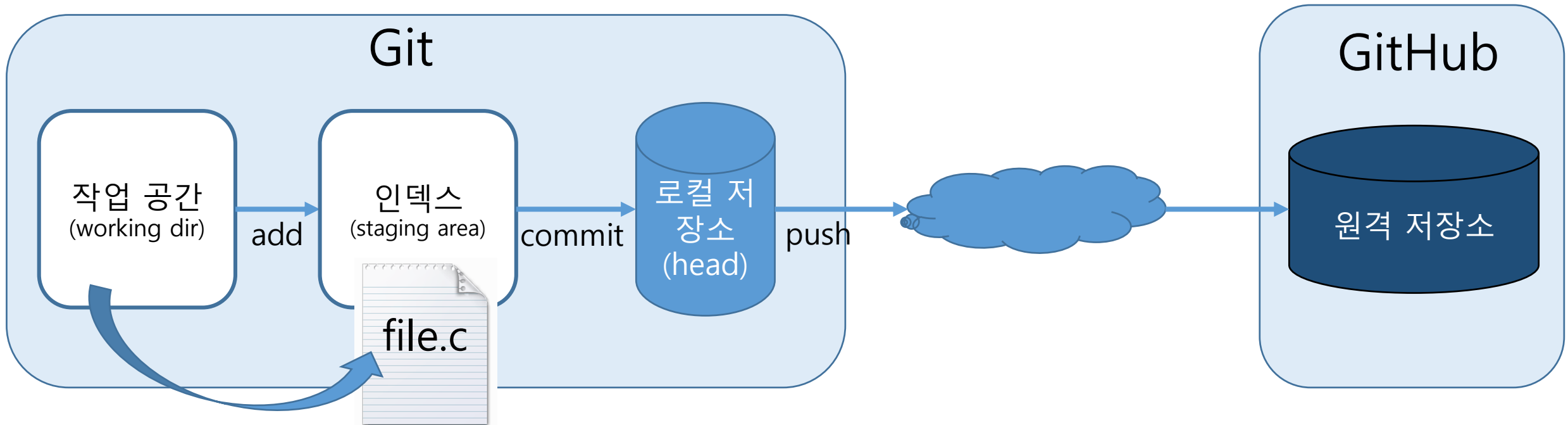
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    file.c~

hansangyun@hansangyun:~/바탕화면/git_examples$
```



# 작업의 흐름



# commit : 저장소에 등록

- `git commit -m "이번 commit에 대한 설명"`

```
hansangyun@hansangyun:~/바탕화면/git_example$ git commit -m "this is first commit"
[master (root-commit) 614bffd] this is first commit
1 file changed, 5 insertions(+)
create mode 100644 file.c
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.c~

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

# commit : 저장소에 등록

- `git commit -m "이번 commit에 대한 설명"`

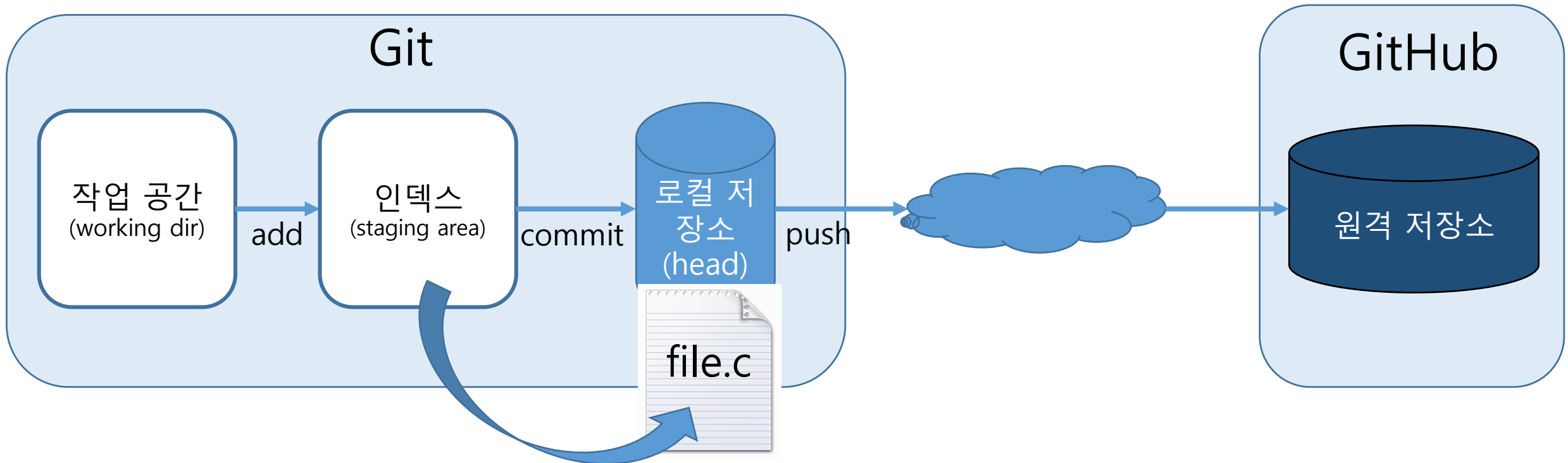
```
hansangyun@hansangyun:~/바탕화면/git_example$ git commit -m "this is first commit"
[master (root-commit) 614bffd] this is first commit
1 file changed, 5 insertions(+)
create mode 100644 file.c
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

file.c~

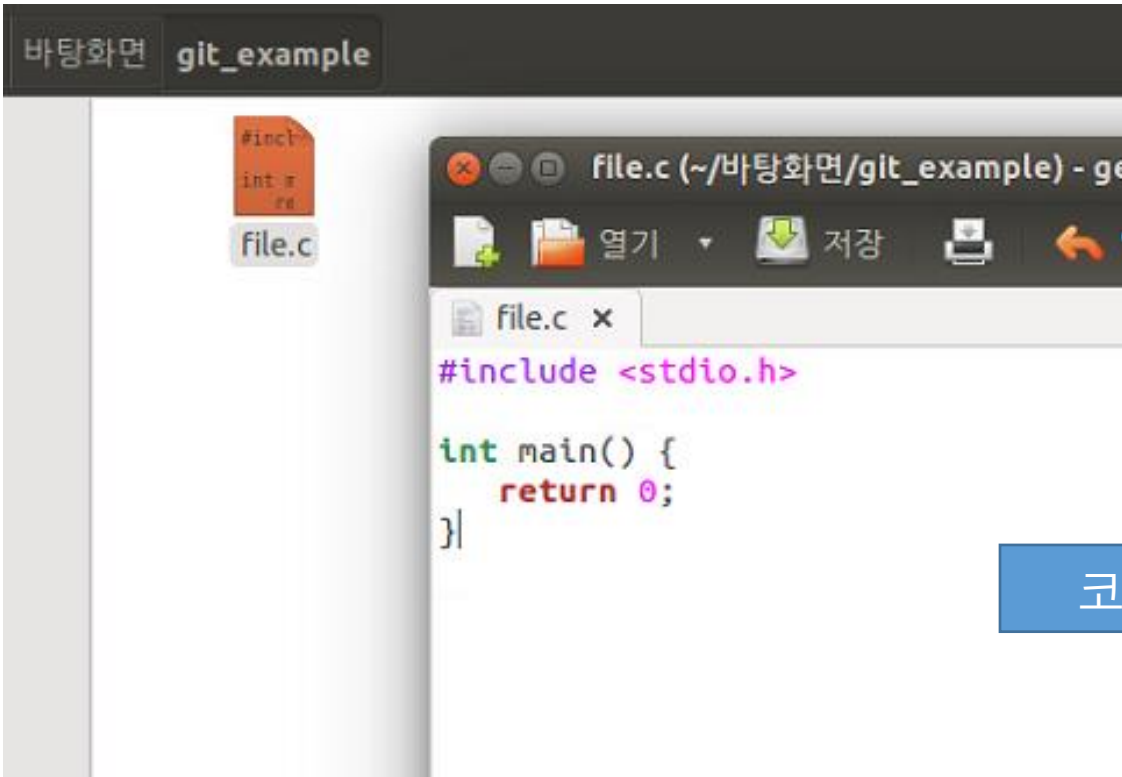
nothing added to commit but untracked files present (use "git add" to track)
```

커밋에 대한 메시지

# 작업의 흐름



# 파일 수정



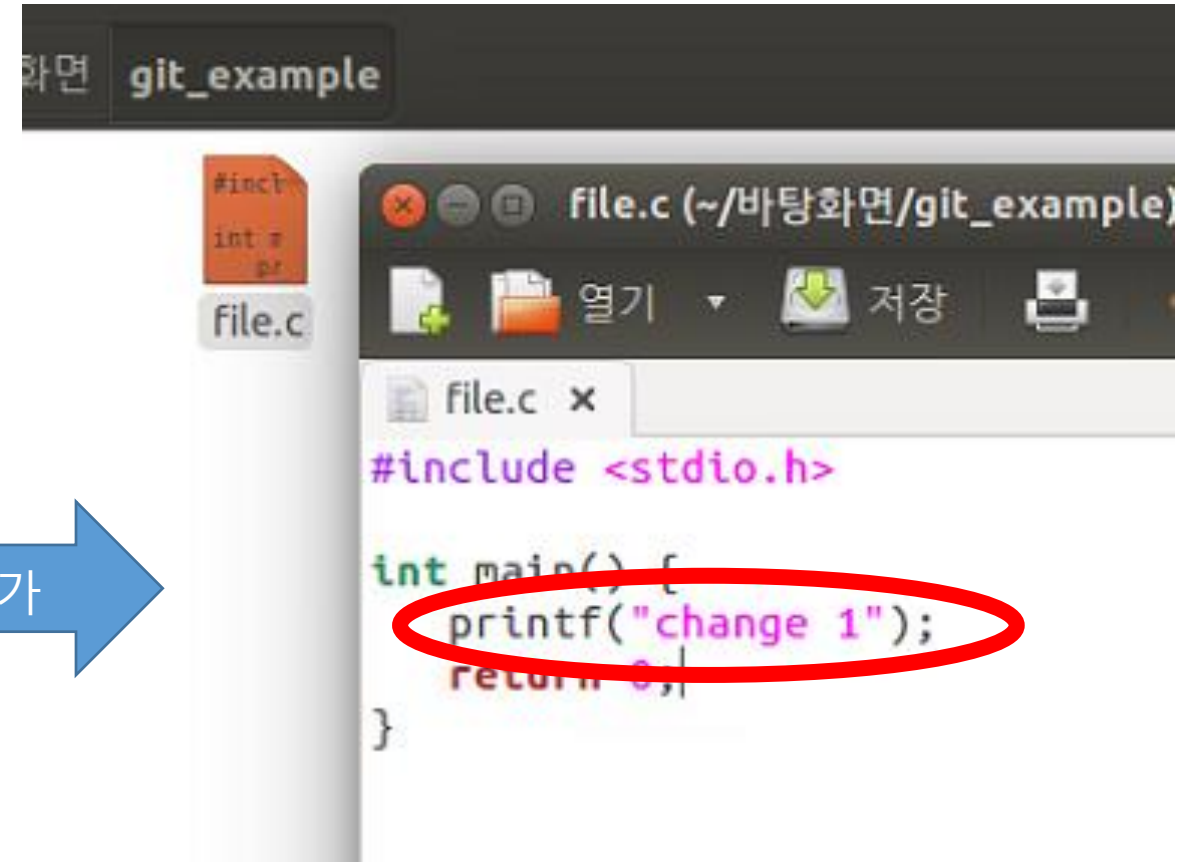
바탕화면 git\_example

file.c

```
#include <stdio.h>

int main() {
    return 0;
}
```

코드 추가



화면 git\_example

file.c

```
#include <stdio.h>

int main() {
    printf("change 1");
    return 0;
}
```



# add 후, status 확인

```
hansangyun@hansangyun:~/바탕화면/git_example$ git add file.c
hansangyun@hansangyun:~/바탕화면/git_example$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   file.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        file.c~
```

# 작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf901f7b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

# 작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/gtc_examples$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf901f7b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋 리비전

각각의 커밋에 대한 고유 ID  
SHA-1 해시값으로 구성



# 작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a2520dd3cbe1ce9292bfad6abf90177b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋 작성자 정보

# 작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date: Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

commit 614bffd7a3530dd3cbe1ce9292bfad6abf90
Author: sangyunHan <sangyun0628@naver.com>
Date: Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋 날짜와 시간

# 작업 이력 확인(명령어 : git log)

```
hansangyun@hansangyun:~/바탕화면/git_example$ git log
commit b3e687450a18671e1e3e64d454f790374800ab4b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:40:50 2014 +0900

    wow this is second commit

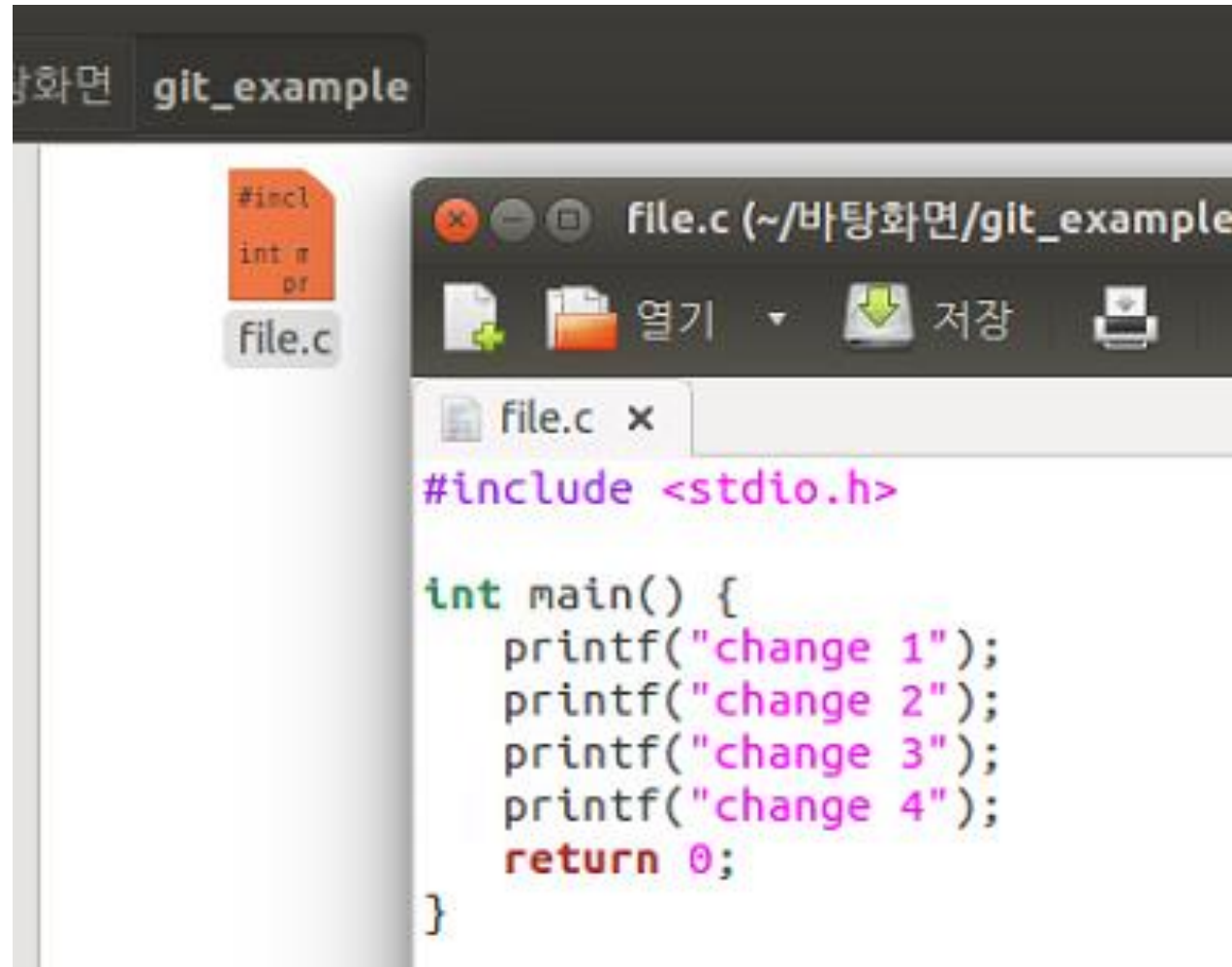
commit 614bffd7a3530dd3cbe1ce9292bfad6abf901f7b
Author: sangyunHan <sangyun0628@naver.com>
Date:   Fri Dec 26 18:23:45 2014 +0900

    this is first commit
```

커밋에 대한 메시지들



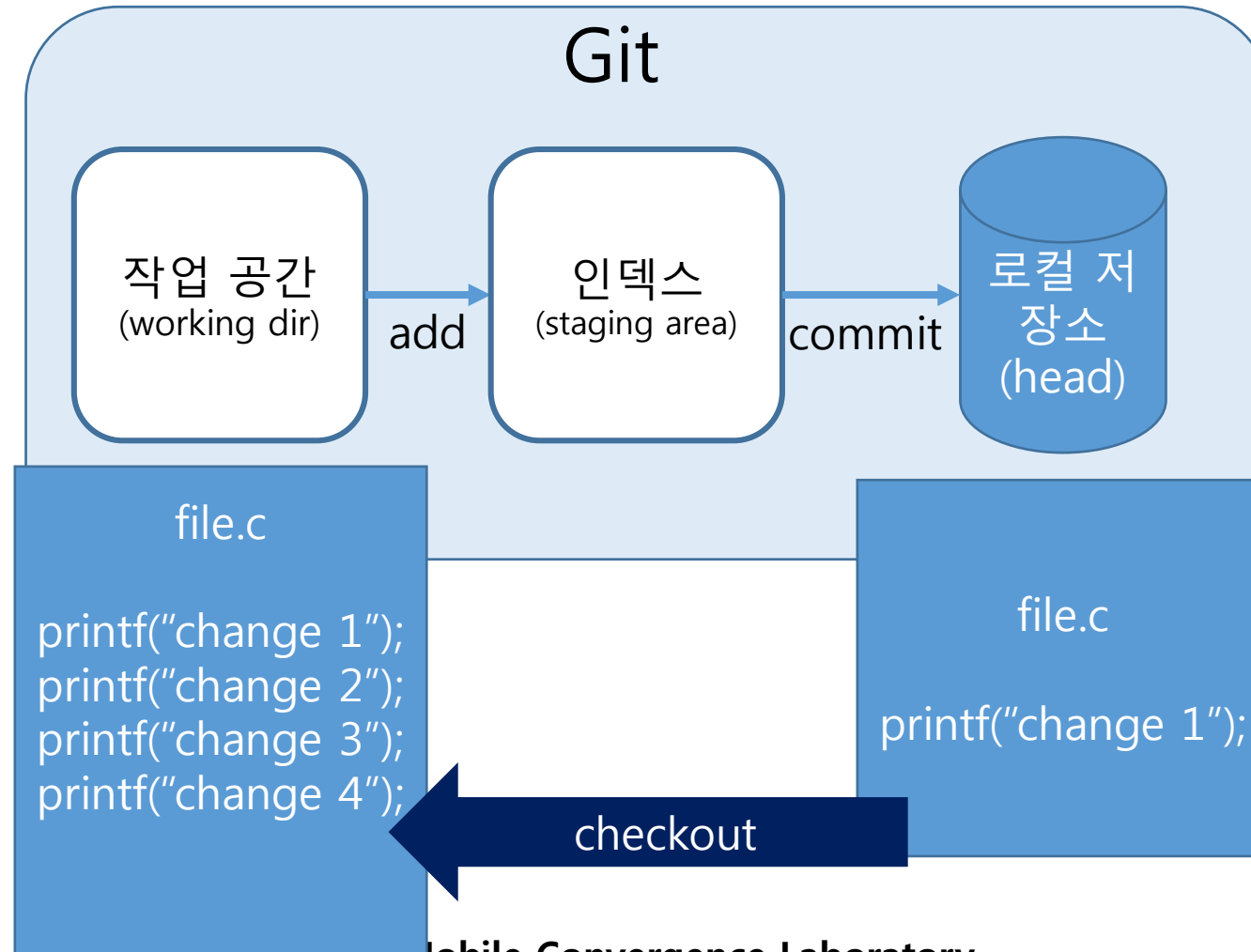
# 변경 취소(이전 commit 내용 가져오기)



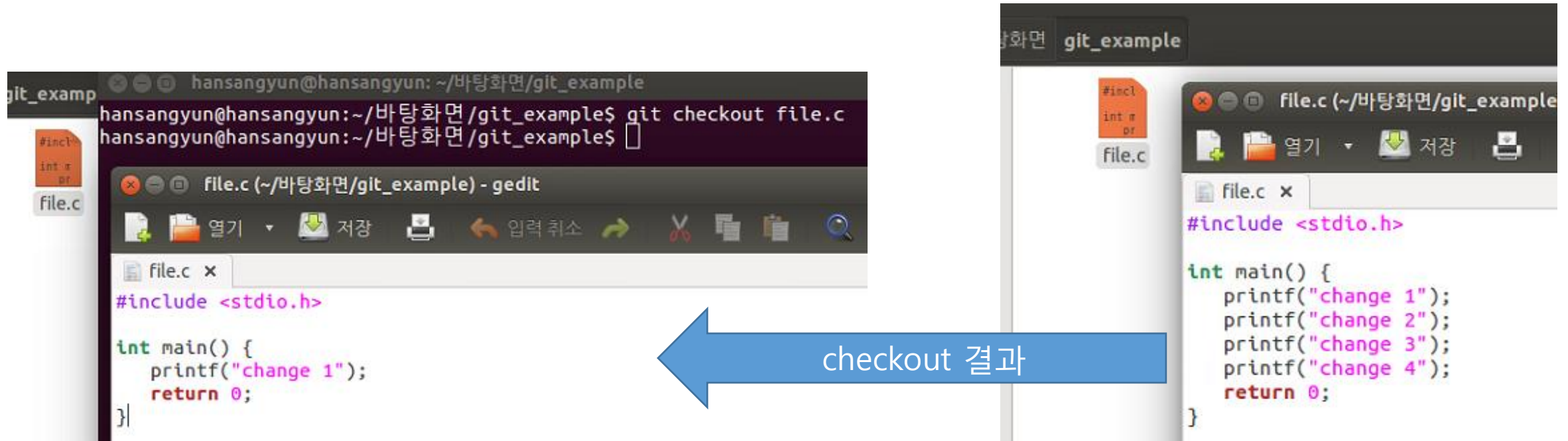
```
#include <stdio.h>

int main() {
    printf("change 1");
    printf("change 2");
    printf("change 3");
    printf("change 4");
    return 0;
}
```

# 작업의 흐름



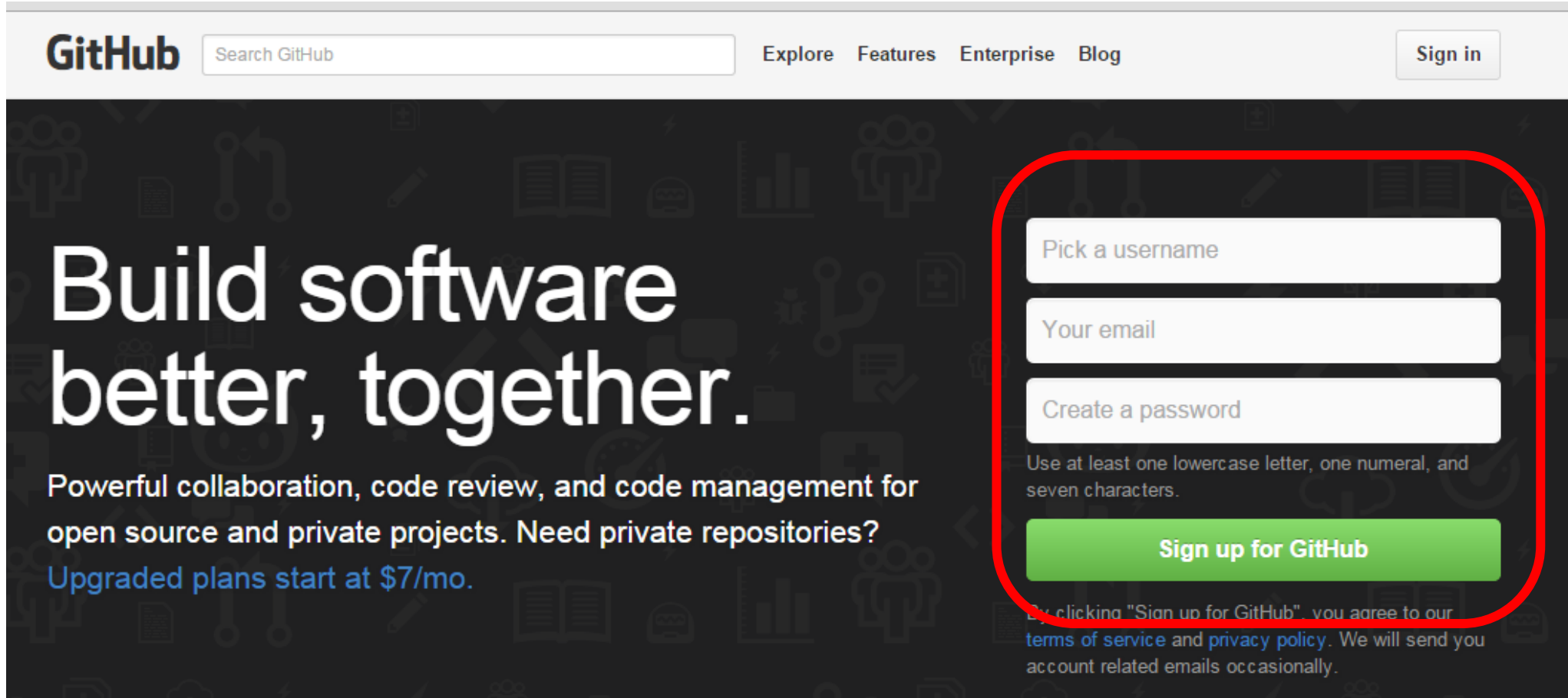
# 변경 취소(명령어 : git checkout 파일명)



# GitHub[원격 저장소] 연결

1. 계정 생성
2. SSH Key 인증
3. Repository 생성

# GitHub 계정 생성



**GitHub** Search GitHub Explore Features Enterprise Blog Sign in

## Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.



# SSH key 발급

- `ssh-keygen -t rsa -C "E-mail"`

```
hansangyun@hansangyun:~/바탕화면/git_example$ ssh-keygen -t rsa -C "sangyun0628@naver.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hansangyun/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hansangyun/.ssh/id_rsa.
Your public key has been saved in /home/hansangyun/.ssh/id_rsa.pub.
The key fingerprint is:
5b:e3:8f:b7:9f:ec:7e:77:3d:7f:ee:fe:5c:f6:03:c7 sangyun0628@naver.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|      S o .
|     + . . E
|    . . o +
|   o.. *O
|  ..O=*^
|
+-----+
```

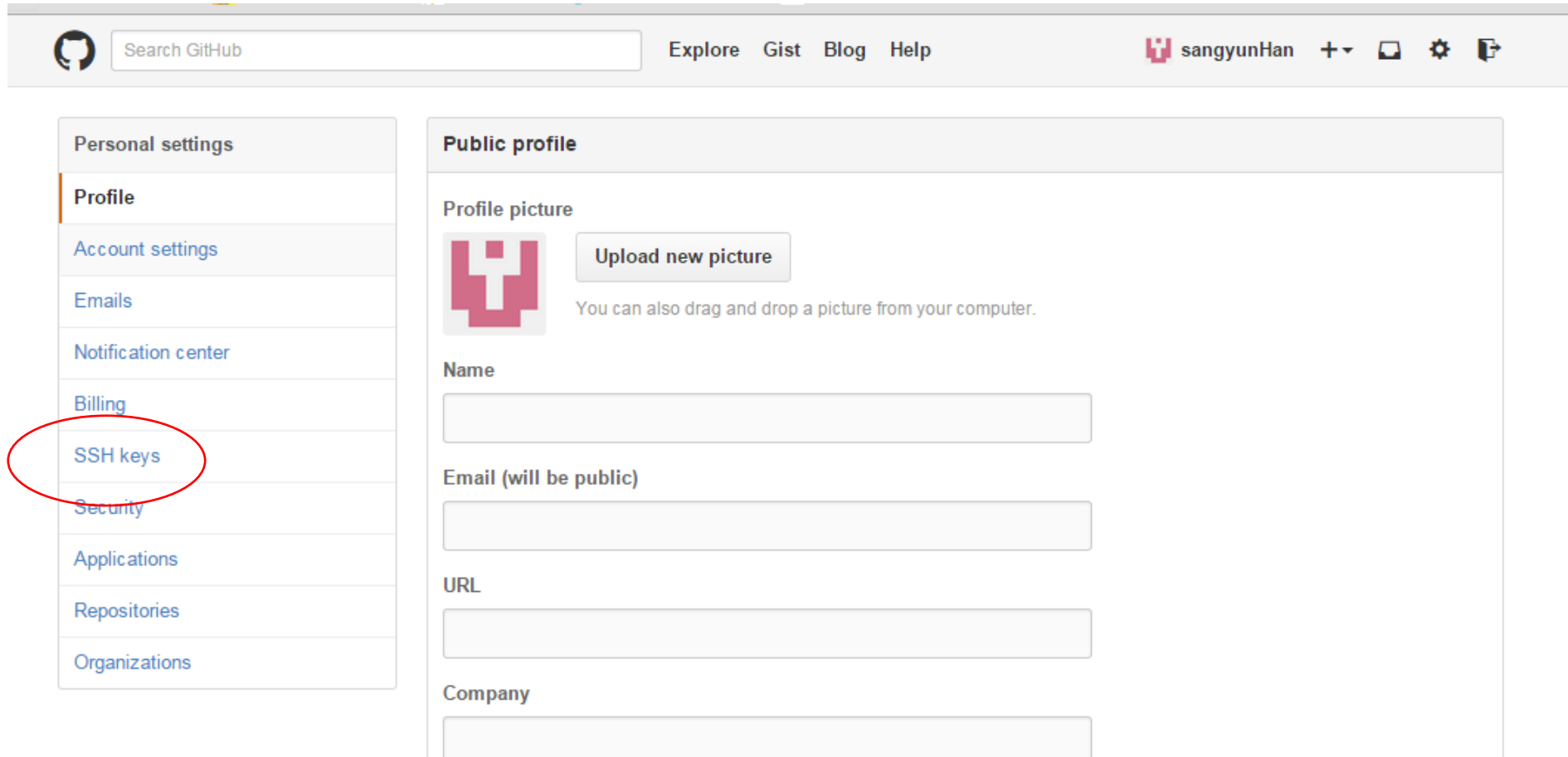
키값이 저장된 곳

# SSH key 발급

- Key가 저장된 경로에서 키값 복사

```
hansangyun@hansangyun:~/바탕화면/git_example$ cd /home/hansangyun/.ssh
hansangyun@hansangyun:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQC84A0jtxsfn9zD6W0/1JdGmbpviGh0mp3NuGzE/wz7
eDoBR1wUd94kvXQayhv6SW6lVeoCo0kjsW7DiXJiWcxrWvDQJcqJuPptq2vLGAW8ES3S5tgSSLKQUJic
fyuEGI9/L88IdQ8Yaqqhjqjfr4v5PhYyDffEKpr0cd0aSU7UU5K6GqCNT0V1AlXE0oMKBJ+NhykAaP2DE
wXxY9zHzvQk+8GhUG9baqp+lVIADUxaC6hEULZtVC8fhy6Ep1XMXuDBZ0hTpDVPeUTOWHfkD4FykDIzr
/VgvDp51v53yadV4LFsXUizJyvXi7dYGY2vbdmGdB54pcDQAzaMQ5Nn7ixVf sangyun0628@naver.c
om
```

# SSH keys 등록



The screenshot shows the GitHub profile settings page for user 'sangyunHan'. The left sidebar contains a list of settings: Personal settings, Profile, Account settings, Emails, Notification center, Billing, SSH keys (highlighted with a red circle), Security, Applications, Repositories, and Organizations. The main content area is titled 'Public profile' and includes a profile picture section with an 'Upload new picture' button, and input fields for Name, Email (will be public), URL, and Company.

Search GitHub

Explore Gist Blog Help

sangyunHan + ▾

Personal settings

Profile

Account settings

Emails

Notification center

Billing

SSH keys

Security

Applications

Repositories

Organizations

Public profile

Profile picture

Upload new picture

You can also drag and drop a picture from your computer.

Name

Email (will be public)

URL

Company

# SSH keys 등록

Emails
Notification center
Billing
<b>SSH keys</b>
Security
Applications
Repositories
Organizations

There are no SSH keys with access to your account.

## Add an SSH Key

Title

git\_test

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQBAQC84A0jtxsfn9zD6W0/1JdGmbpviGh0mp3NuGzE/wz7eDoBR1wUd94kvXQ
ayhv6SW6IVeoCoOkjsW7DiXJiWcxrWvDQJcqJuPptq2vLGAW8ES3S5tgSSIKQUJicfyuEGl9/L88ldQ8Yaqghqjfr4v5Ph
YyDffEKprOcd0aSU7UU5K6GqCNtOV1AIXEOoMKBj+NhykAaP2DEwXxY9zHzvQk+8GhUG9baqp+VIADUxaC6hEUI
ZtVC8fhv6Ep1XMXuDBZ0hTpDVPeUTOWHfkD4FykDIzrVgvDp51v53yadV4LFsXUizJyvXi7dYGY2vbdmGdB54pcDQ
AzaMQ5Nn7ixVf sangyun0628@naver.com
```

Add key

키값 붙여넣기

# SSH keys 등록 완료

## SSH Keys

[Add SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

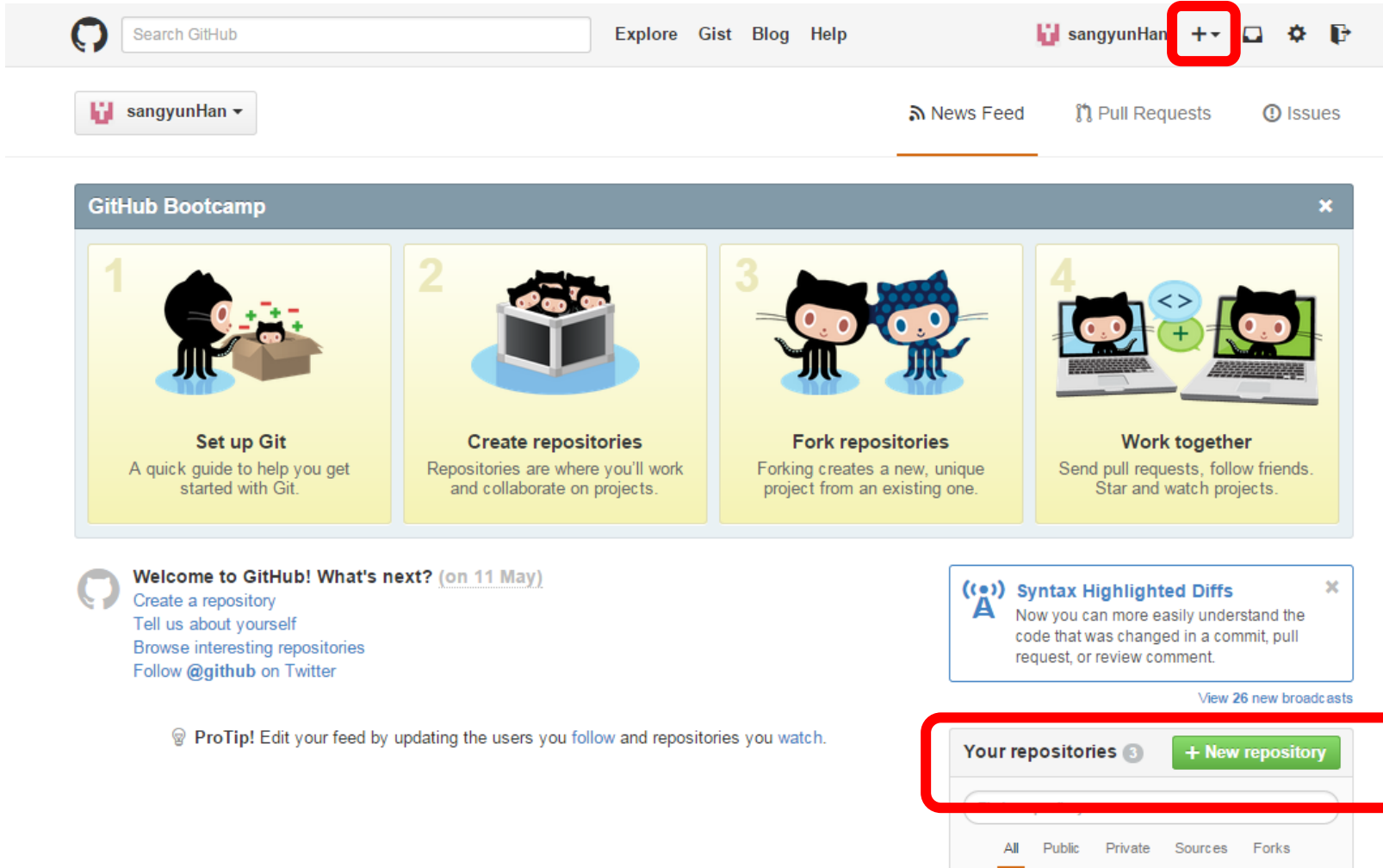
**git\_test**

5b:e3:8f:b7:9f:ec:7e:77:3d:7f:ee:fe:5c:f6:03:c7

Added on 28 Dec 2014 — Never used

[Delete](#)

# GitHub repository 생성



Search GitHub Explore Gist Blog Help sangyunHan +

sangyunHan News Feed Pull Requests Issues

### GitHub Bootcamp

- 1 Set up Git**  
A quick guide to help you get started with Git.
- 2 Create repositories**  
Repositories are where you'll work and collaborate on projects.
- 3 Fork repositories**  
Forking creates a new, unique project from an existing one.
- 4 Work together**  
Send pull requests, follow friends. Star and watch projects.

**Welcome to GitHub! What's next? (on 11 May)**  
[Create a repository](#)  
[Tell us about yourself](#)  
[Browse interesting repositories](#)  
[Follow @github on Twitter](#)

**ProTip!** Edit your feed by updating the users you [follow](#) and repositories you [watch](#).

**Syntax Highlighted Diffs**  
Now you can more easily understand the code that was changed in a commit, pull request, or review comment.  
[View 26 new broadcasts](#)



**Your repositories 3** [+ New repository](#)

All Public Private Sources Forks

# GitHub repository 생성

Owner

Repository name


  sangyunHan ▾ / git\_test ✓


Great repository names are short and memorable. Need inspiration? How about **potential-robot**.

**Description** (optional)

This repository is test for github seminar

---


☒  **Public**  
Anyone can see this repository. You choose who can commit.

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

---

☒ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** ▾

Add a license: **None** ▾ 

Create repository



# GitHub repository 생성

Owner: sangyunHan / Repository name: git\_test ✓

Great repository names are short and memorable. Need inspiration? How about **potential-robot**.

Description (optional): This repository is test for github seminar

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

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

☒ Initialize this repository with a README  
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: None Add a license: None ⓘ

Create repository

해당 repository의  
이름과 간단한 설명



# 프로젝트의 라이선스 선택

Owner: sangyunHan / Repository name: git\_test ✓

Great repository names are short and memorable. Need inspiration? How about **potential-roll**

Description (optional)  
This repository is test for github seminar

☒ **Public**  
Anyone can see this repository. You choose who can commit.

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

☒ **Initialize this repository with a README**  
This will allow you to `git clone` the repository immediately. Skip this step if

Add .gitignore: **None** Add a license: **None** ⓘ

**Create repository**

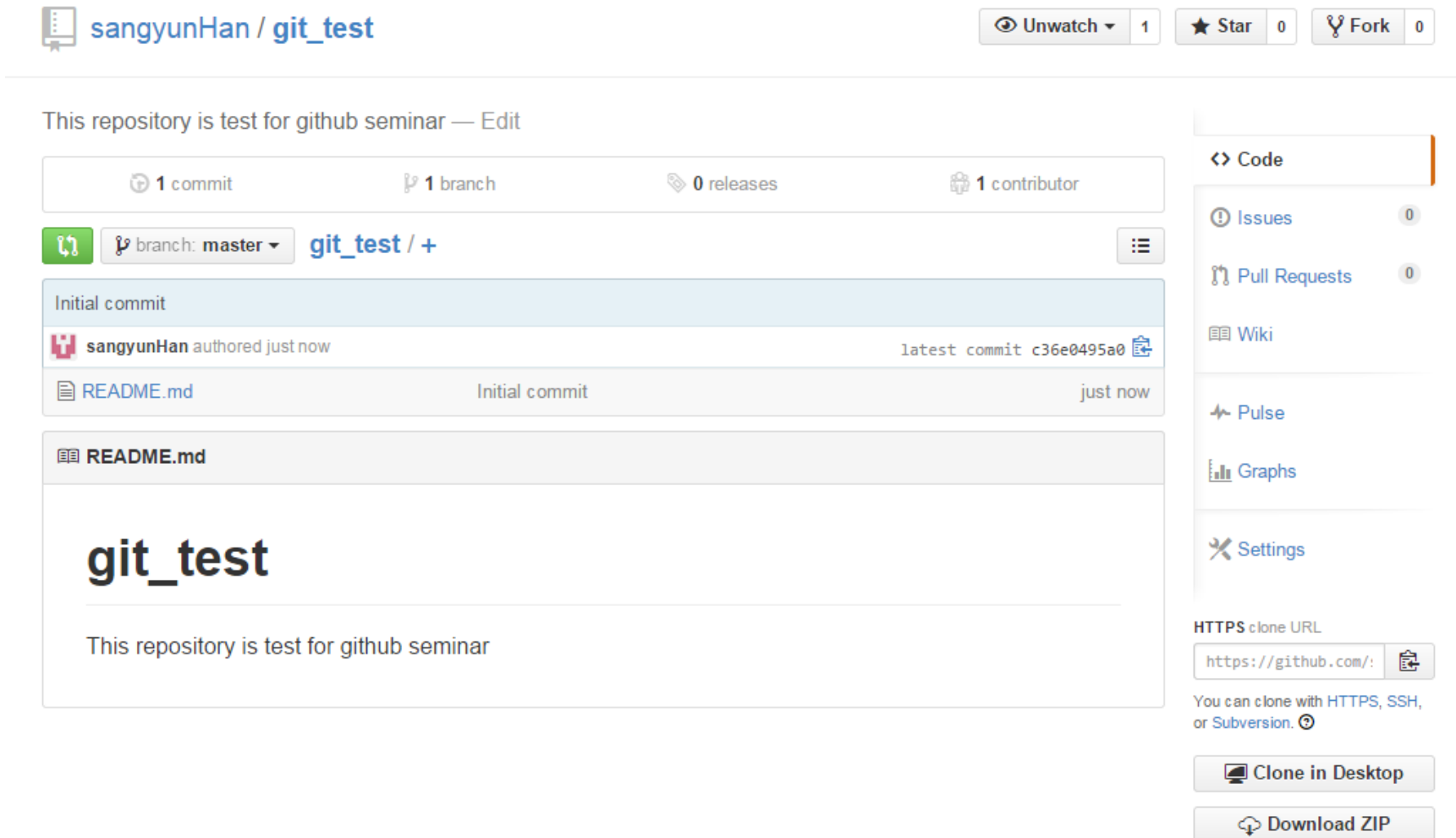
Add a license: **GNU GPL v3.0** ⓘ

**Licenses** ✕

Filter licenses...

- Apache License 2.0
- GNU GPL v2.0
- MIT License
- Artistic License 2.0
- CC0 1.0 Universal
- Eclipse Public License v1.0
- GNU Affero GPL v3.0
- ☒ **GNU GPL v3.0**
- GNU LGPL v2.1
- GNU LGPL v3.0
- ISC license
- Mozilla Public License 2.0**

# GitHub repository 생성 완료



The screenshot shows the GitHub interface for a repository named 'git\_test' created by 'sangyunHan'. At the top, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). Below this, a message states 'This repository is test for github seminar — Edit'. A summary bar indicates '1 commit', '1 branch', '0 releases', and '1 contributor'. The main content area shows an 'Initial commit' by 'sangyunHan' just now, with the latest commit hash 'c36e0495a0'. A 'README.md' file is listed, and its content is displayed below: 'git\_test' followed by 'This repository is test for github seminar'. On the right sidebar, there are links for 'Code', 'Issues' (0), 'Pull Requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. At the bottom right, there are options to 'Clone in Desktop' and 'Download ZIP', along with the 'HTTPS clone URL' 'https://github.com/:'. A note mentions that the repository can also be cloned using 'SSH' or 'Subversion'.

# Q&A