

Git / Github

입문하기

161018

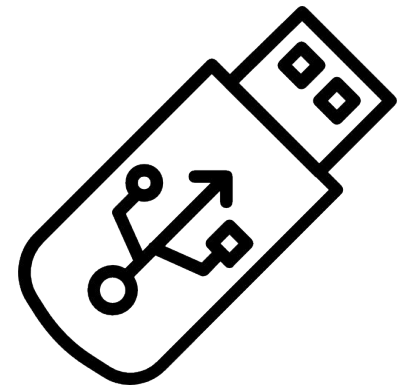
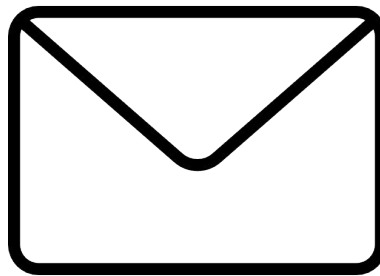
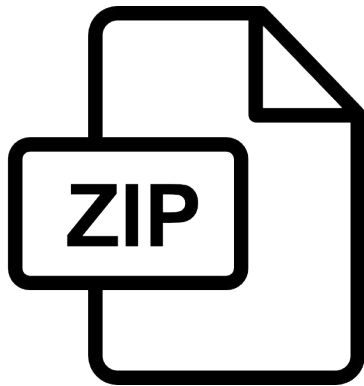
enuri91@gmail.com

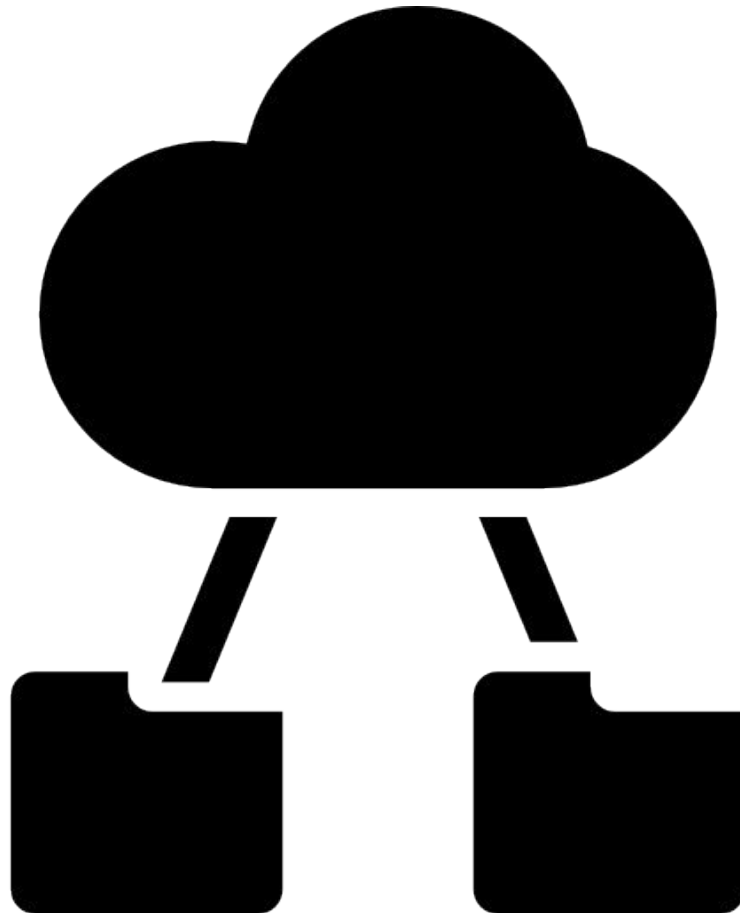
한홍근

Git / Github

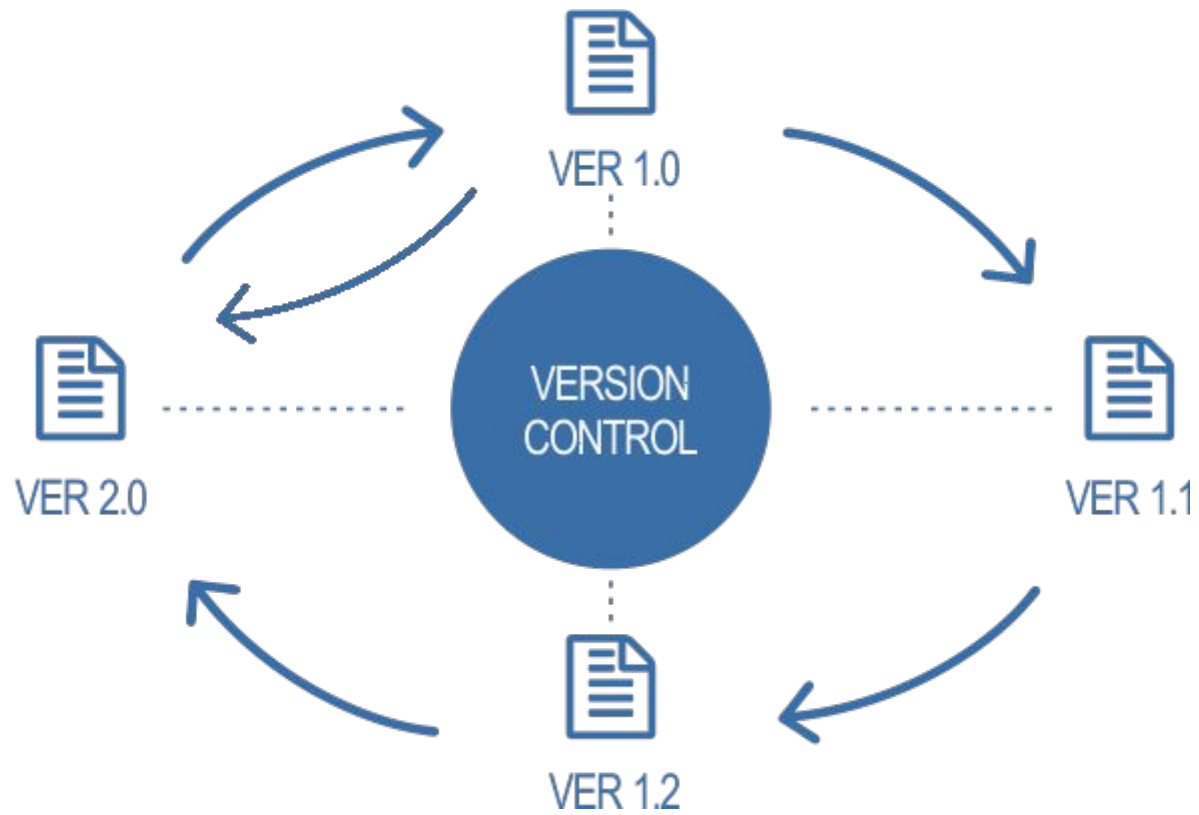




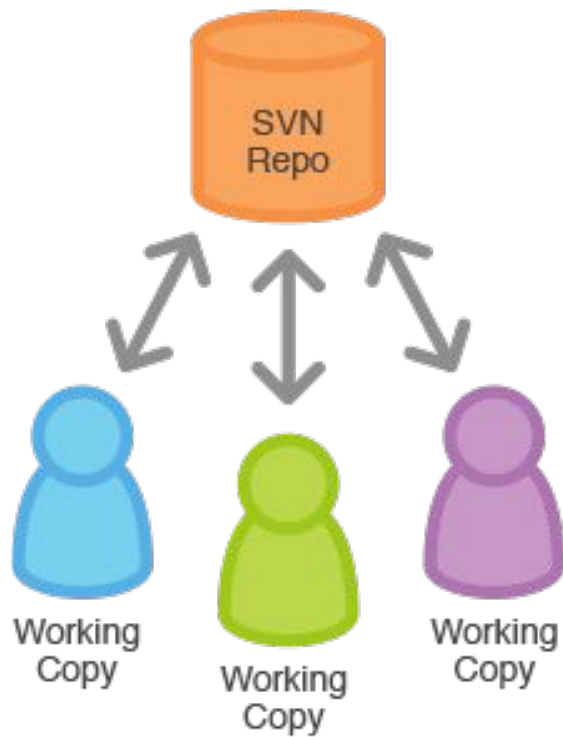




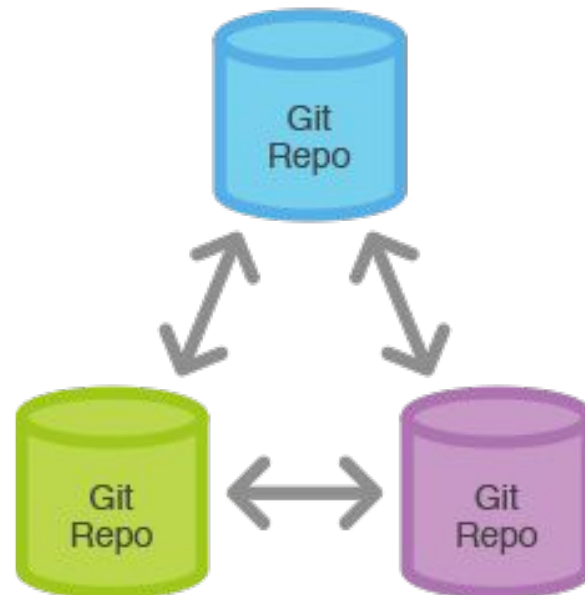


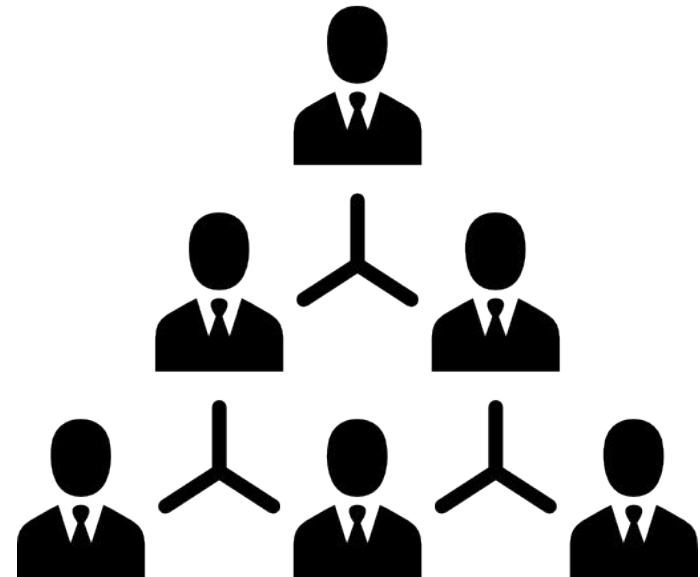


Central-Repo-to-Working-Copy
Collaboration



Repo-to-Repo
Collaboration







- 버전 관리 가능
-
- 분산 작업 가능
-
- 빠른 처리 속도
-
- 오픈 소스 프로젝트



SOURCEFORGE



GitLab



Bitbucket

실습 준비

Git bash

win/osx

<https://git-scm.com/>

linux

`sudo apt-get install git -y`

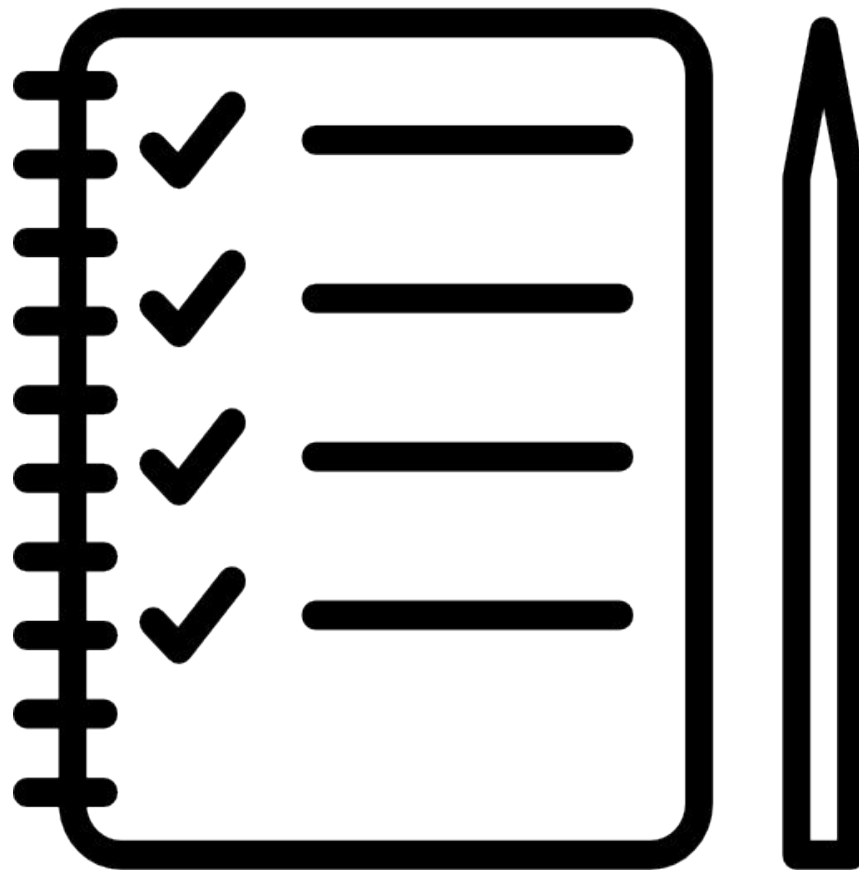
Github

<https://github.com/>

- 이메일 인증 필수

source code

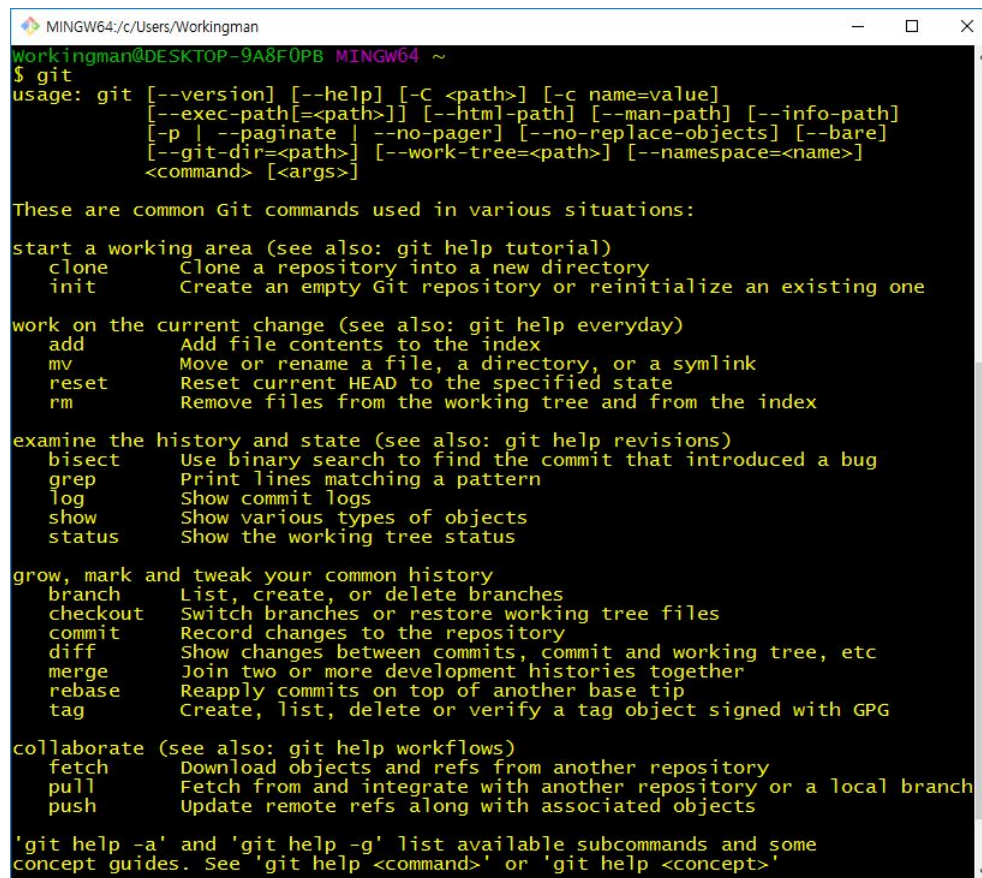
[download link](#)



실습 준비

OSX / linux : 터미널에서 git 입력

윈도우 : git bash 실행



```
MINGW64/c/Users/Workingman
workingman@DESKTOP-9A8F0PB MINGW64 ~
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
       [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index

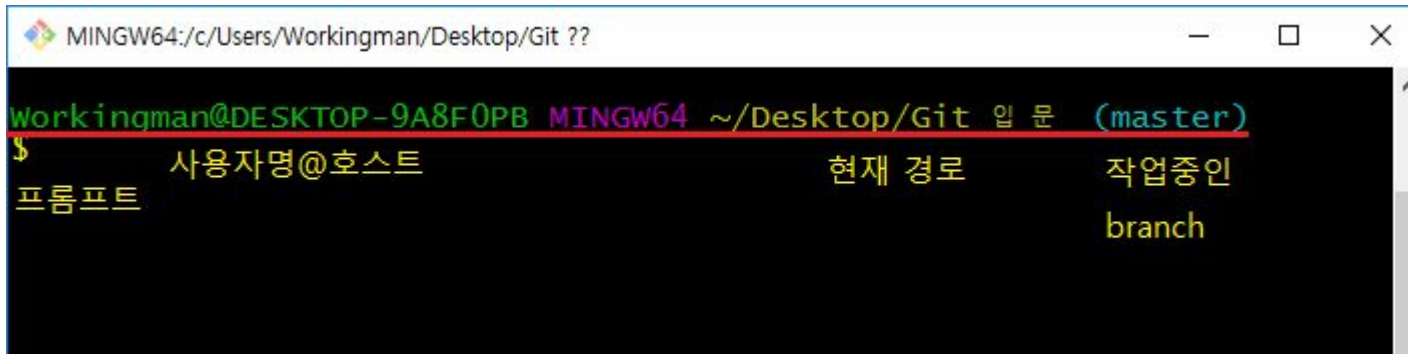
examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  checkout   Switch branches or restore working tree files
  commit     Record changes to the repository
  diff       Show changes between commits, commit and working tree, etc
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
```

Git Bash



```
MINGW64:/c/Users/Workingman/Desktop/Git ??  
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/Git (master)  
$
```

사용자명@호스트 현재 경로 작업중인
프롬프트 branch

\$ cd <경로>

- <경로>로 이동하는 명령어

** 꺾쇠 (< , >)는 입력하지 않습니다

예 > cd ~

홈 디렉토리로 이동

Git Bash

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~  
$ mkdir ~/Desktop/git_begin
```

mkdir - 새 폴더(디렉토리) 생성

ex > mkdir <디렉토리>

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~  
$ ls  
advanced_git/  
AppData/  
'Application Data'  
Contacts/  
Cookies@
```

ls - 목록 출력

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~  
$ cd ~/Desktop/git_begin/
```

cd - 작업 경로 이동

ex > cd <경로>

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin  
$ pwd  
/c/Users/Workingman/Desktop/git_begin
```

pwd - 현재 작업 경로 출력

Git Bash

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin  
$ cp ~/Desktop/git_source/commit1/source_code.c ~/Desktop/git_begin/
```

cp - 파일 복사

ex > cp <원본 경로> <복사본을 생성할 경로>

이후에 파일 덮어쓰기 작업은 GUI에서 옮기면 됩니다. (cp 기능을 설명하기 위한 예제)

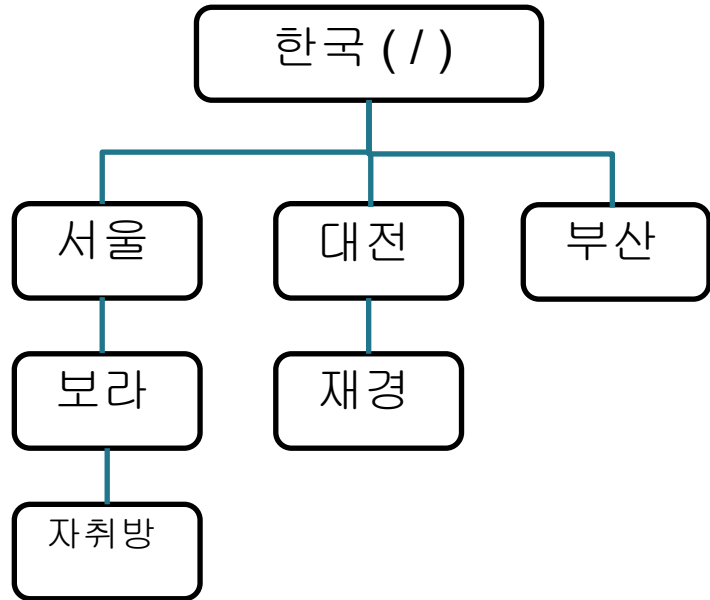
```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin  
$ cat source_code.c  
#include <stdio.h>  
  
int main(void) {  
    printf("Hello World!\n");  
    return 0;  
}
```

cat - 파일 미리보기

ex > cat <파일명>

절대경로 / 상대경로

절대경로 : 최상위 디렉토리(/)를 기준
파일, 폴더의 고유 경로



상대 경로 : 현재 위치를 기준으로 설정되는
'상대적인' 경로

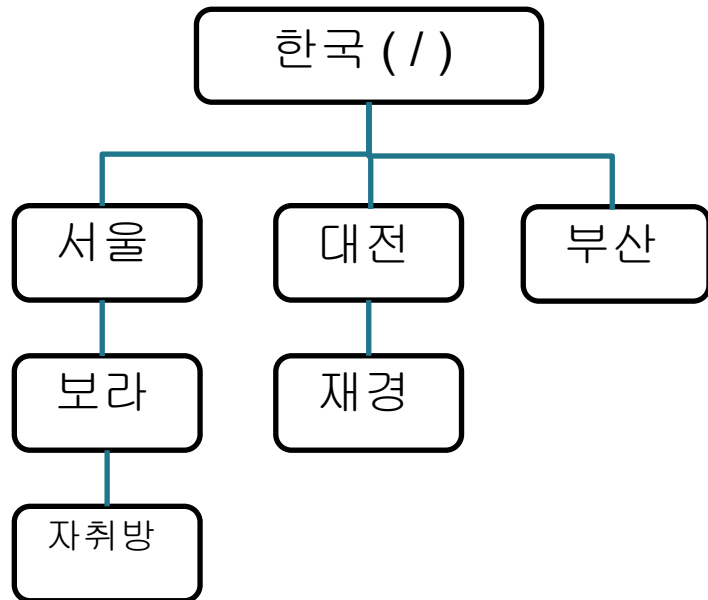
서울에서 보라가 사는 동네를 찾아가자

절 > /서울/보라

상 > 보라

일반작업은 홈 디렉토리(~)에서 이뤄짐

절대경로 / 상대경로



현재 디렉토리 > . (마침표 1개)

상위 디렉토리 > .. (마침표 2개)

홈 디렉토리 > ~

#1 한국에서 보라의 자취방을 찾아가자

현재경로 > 한국(홈 디렉토리)

절 : `cd /서울/보라/자취방`

상 : `cd ./서울/보라/자취방`

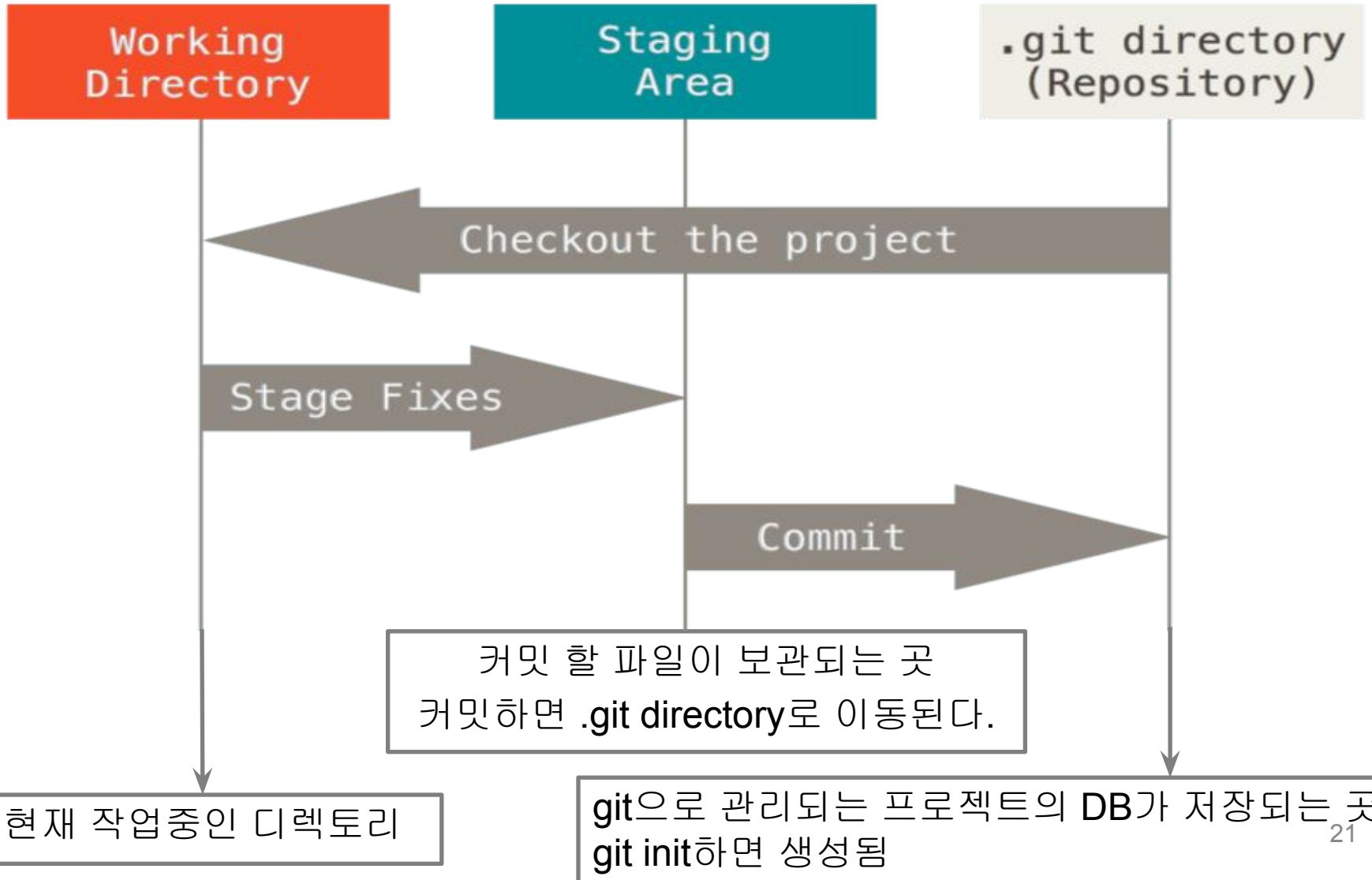
#2 재경이네 동네에서 부산으로 찾아가자

현재경로 > 재경이네 동네

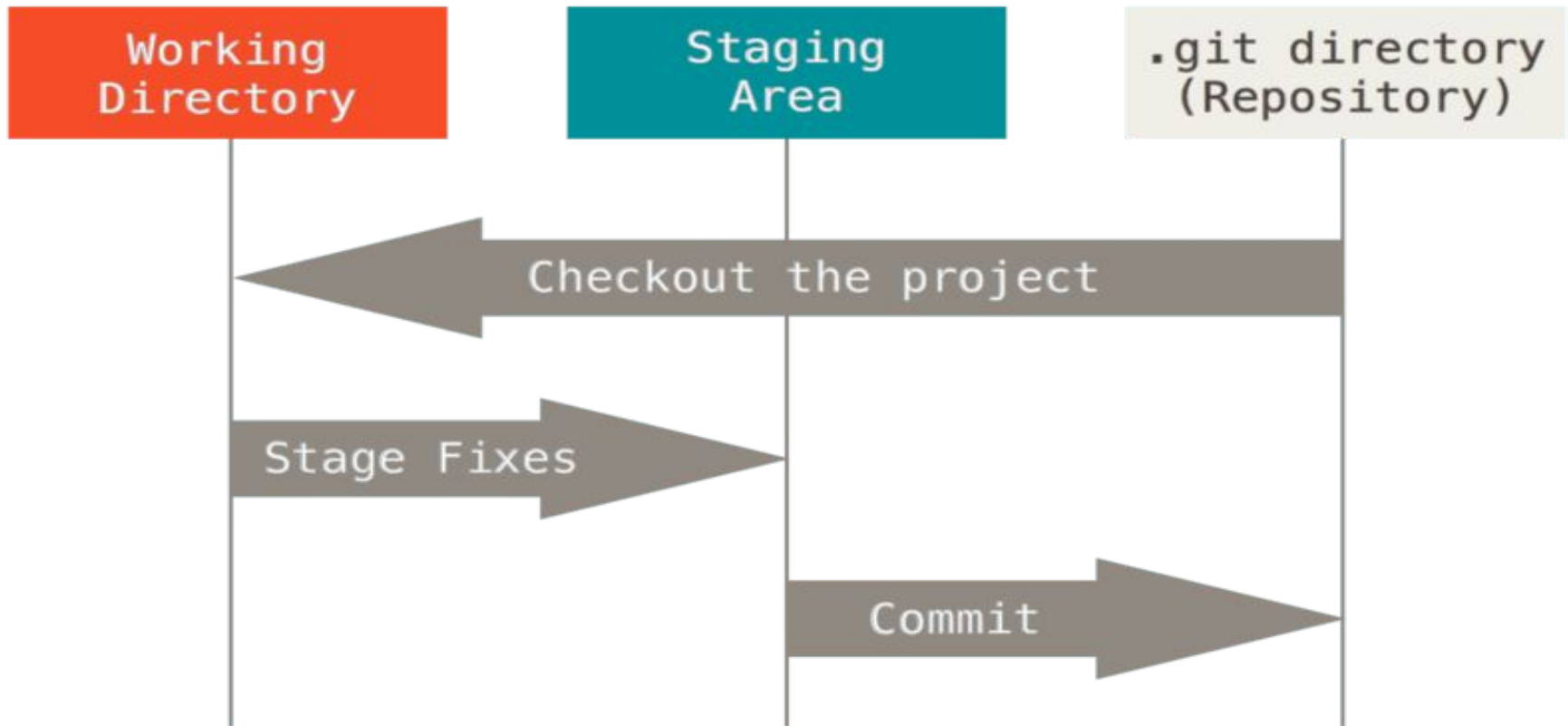
절 : `cd /부산`

상 : `cd ../../부산`

Git의 3 영역



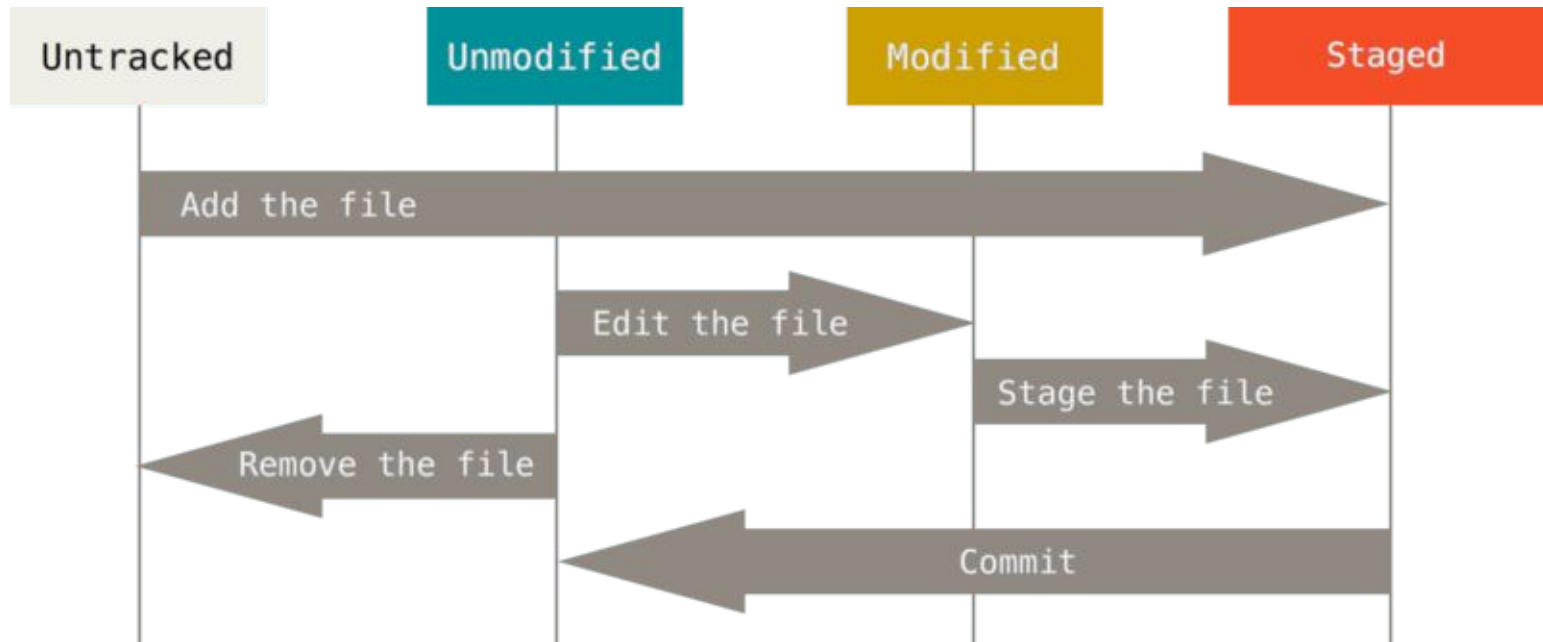
Git의 3 영역



작업 흐름

Working Directory -> Staging Area -> .git directory

Git의 라이프사이클



Untracked - 파일 생성 직후 이력이 git으로 추적되고 있지 않은 상태

Unmodified - git이 추적하고 있으면서, 수정이력이 발생하지 않은 상태

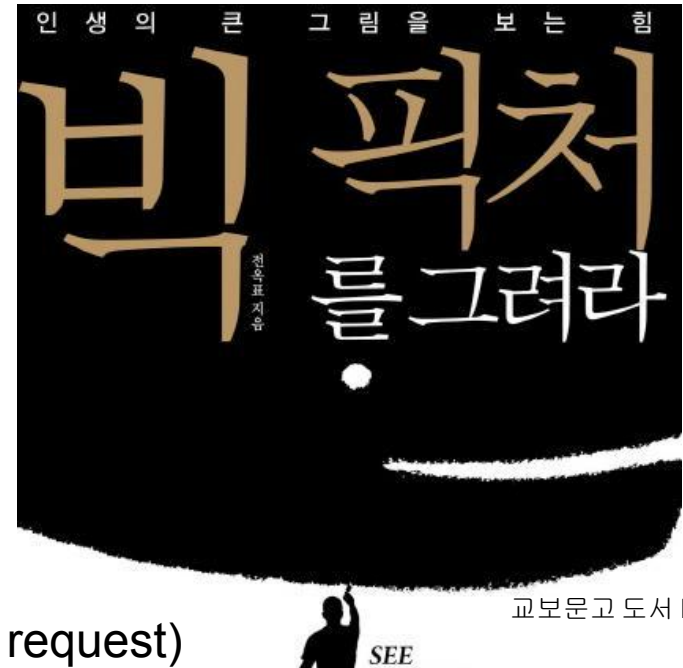
Modified - git이 추적하고 있으면서, 수정이력이 발생한 상태

Staged - 스테이징 영역에 추가된 상태

실습 흐름

구구단(gugu_code.c)과 코드 설명을 도와줄 문서(README.md)를 만들자

- #1. 첫 커밋 기록 - source_code
- #2. 반복 실습 - gugu_code
- #3. 커밋 메시지 수정(amend)
- #4. 작업/기능 추가(branch)
- #5. 문서 작성 - README.md
- #6. 충돌 해결
- #7. Github에 올리자(push)
- #8. 친구의 코드를 가져오자(fork)
- #9. 코드를 추가해서 친구 프로젝트에 합치자(pull request)



교보문고 도서 DB

자주 써 봐야하는 git 명령어

git status - git 상태 확인

git log - 커밋 전체 로그 확인

git shortlog - 커밋 로그 중 메시지만 확인

git diff - 소스코드 변경이력 확인

init - git 시작

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin
$ git init
Initialized empty Git repository in C:/Users/Workingman/Desktop/git_begin/.git/
```

git init

"현재 디렉토리를 git으로 관리 시작하겠다"

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ ls -al
total 9
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 ./
drwxr-xr-x 1 Workingman 197121  0 10월  12 10:50 ../
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 .git/
-rw-r--r-- 1 Workingman 197121 77 10월  12 10:52 source_code.c
```

ls -al

.git이 생성된 것을 확인가능

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ ls -al .git/
total 11
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 ./
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 ../
-rw-r--r-- 1 Workingman 197121 130 10월  12 13:32 config
-rw-r--r-- 1 Workingman 197121  73 10월  12 13:32 description
-rw-r--r-- 1 Workingman 197121  23 10월  12 13:32 HEAD
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 hooks/
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 info/
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 objects/
drwxr-xr-x 1 Workingman 197121  0 10월  12 13:32 refs/
```

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`

commit #1

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git status
On branch master

Initial commit

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

    source_code.c

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

git status

현재 git의 상태를 확인가능하다.

-> Untracked files에 source_code.c가 있다

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git add source_code.c
warning: LF will be replaced by CRLF in source_code.c.
The file will have its original line endings in your working directory.
```

git add <파일 or 디렉토리>

- git status로 다시 상태를 확인해보자

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`

commit #1

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git status
On branch master

Initial commit

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

    new file:   source_code.c
```

`git status`로 다시 상태를 확인해보자

-> Untracked files가 아닌 changes to be committed로 옮겨간다

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git commit -m "first commit"
[master (root-commit) 7eedd42] first commit
warning: LF will be replaced by CRLF in source_code.c.
The file will have its original line endings in your working directory.
1 file changed, 6 insertions(+)
create mode 100644 source_code.c
```

`git commit -m "커밋 메시지"`

- `commit`으로 하나의 역사(기록)을 남긴다
- 이 때, 기록에 대한 메시지를 같이 남긴다

commit #1

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`

```
workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git status
On branch master
nothing to commit, working tree clean
```

또 다시 `git status`

- 아무것도 없다!

```
workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git log
commit 7eedd42ac703679395cbacd58592c35c9fa23f55
Author: yckumon <yckumon@naver.com>
Date:   Wed Oct 12 13:45:21 2016 +0900

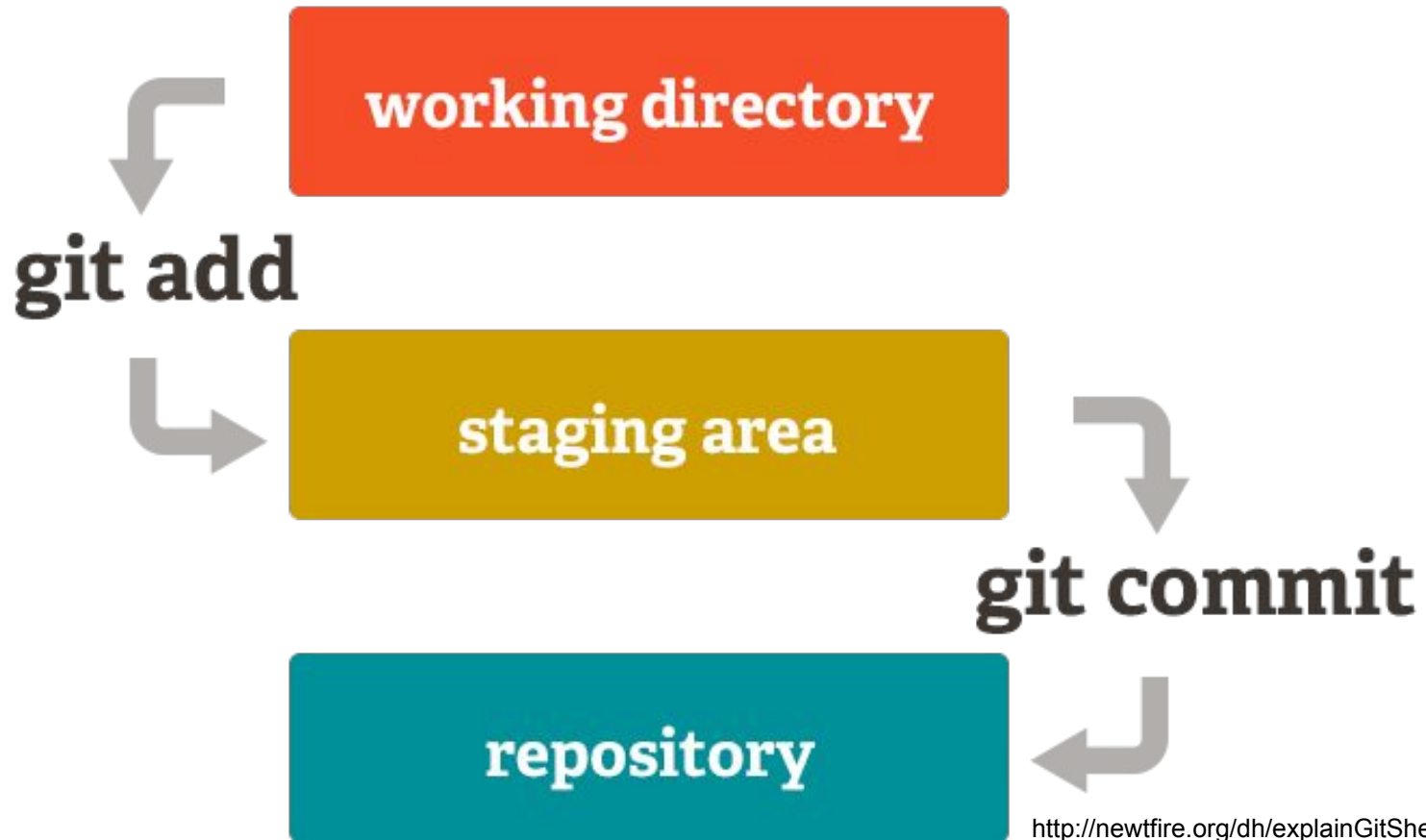
    first commit
```

`git log`

- `commit` 기록을 다시 보자
- 첫 커밋에 입력했던 커밋 메시지가 보인다

돌아보기 #1

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`



<http://newtfire.org/dh/explainGitShell.html>

Working Directory에 `source_code.c` 추가 → `git add` → Staging Area로 이동
→ `git commit` → `.git directory`로 이동 → 하나의 역사(기록) 생성

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`

commit #2, 3 - 복습

commit #2

commit2 폴더의 파일을 실습 폴더로 복사해서(덮어쓰기) 두번째 커밋을 만들기

[커밋 메시지 내용] : "gugu_code #1 "

commit #3

commit3 폴더의 파일을 실습 폴더로 복사해서(덮어쓰기) 세번째 커밋을 만들기

[커밋 메시지 내용] : "gugu_code #2 : 구구단 "

commit #2, 3 - 복습

명령어에 필요한 옵션을 모르겠다면?

git <명령어> --help

```
workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git shortlog
yckumon (3):
    first commit
    gugu_code #1
    gugu_code #2 : 구 구 단
```

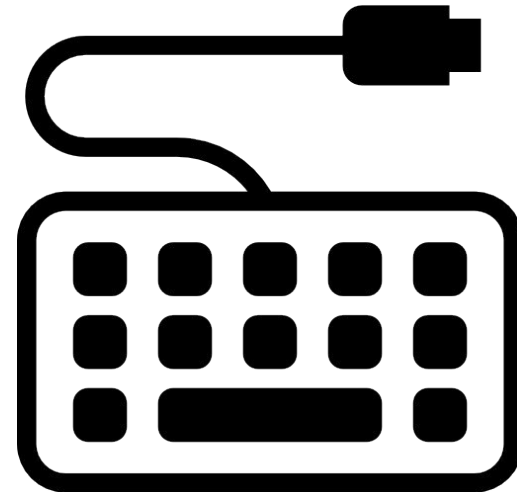
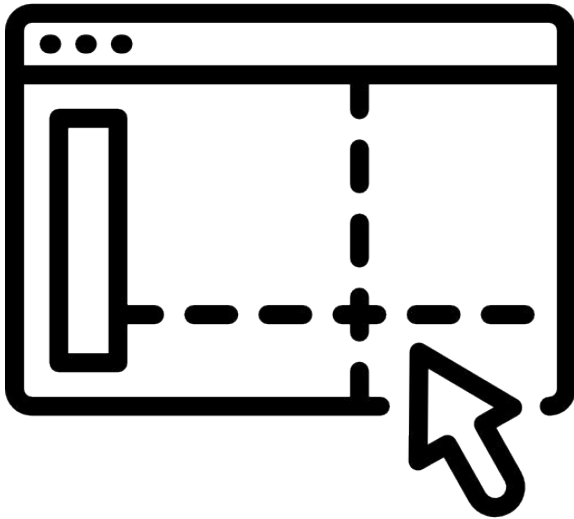
git shortlog

- first commit
- gugu_code #1
- gugu_code #2 : 구구단

세 개의 커밋 로그가 나와야한다

vi - text editor

명령어에 필요한 옵션을 모르겠다면?
`git <명령어> --help`



vi - text editor

명령어에 필요한 옵션을 모르겠다면?
git <명령어> --help



i	커서 위치부터 삽입
[#]d	#글자 삭제 (default : 1)
dd	커서가 있는 1줄 삭제
ESC	글자 입력 끝낼때
:wq	저장하고 나가기

집에서 읽어보세요

vim :

https://wiki.kldp.org/KoreanDoc/html/Vim_Guide-KLDP/Vim_Guide-KLDP.html

vi 모드 : <http://yagi815.tistory.com/146>

커밋 메시지를 수정하자

명령어에 필요한 옵션을 모르겠다면?

`git <명령어> --help`

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git shortlog
yckumon (3):
    first commit
    gugu_code #1
    gugu_code #2 : 구 구 단
```

한글은 인코딩 문제가 발생할 수 있다!

→ 영어로 바꾸자

→ `git commit --amend`

```
gugu_code #2 : 구 구 단

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Oct 12 14:03:23 2016 +0900
#
# On branch master
# Changes to be committed:
#   modified:   gugu_code.c
#
# Changes not staged for commit:
#   modified:   source_code.c
#
```

"구구단"을 "upto nine by nine"으로 바꾸고 저장하자

커밋 메시지를 수정하자

명령어에 필요한 옵션을 모르겠다면?

`git <명령어> --help`

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git shortlog
yckumon (3):
    first commit
    gugu_code #1
    gugu_code #2 : 구 구 단
```

git shortlog

→ "구구단"에서 :upto nine by nine"으로 변경 확인

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git shortlog
yckumon (3):
    first commit
    gugu_code #1
    gugu_code #2 : upto nine by nine
```

diff - 변경 이력 확인

명령어에 필요한 옵션을 모르겠다면?

`git <명령어> --help`

수 백, 수 천라인의 코드 중 '한 줄'이 수정되었다면?

#1. source_code.c 파일 열기 → `vi source_code.c`

#2. `printf("Hello World!\n");` ← 삭제

#3. `printf("Bye World!\n");` ← 추가

#4. 저장(:wq) 후 `git diff` 입력하고 확인해보기

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git diff
diff --git a/source_code.c b/source_code.c
index 411b990..ce96bbd 100644
--- a/source_code.c
+++ b/source_code.c
@@ -1,6 +1,6 @@
#include <stdio.h>

int main(void) {
-    printf("Hello World!\n");
+    printf("Bye World!\n");
    return 0;
}
```

diff - 변경 이력 확인_commit#4

변경 이력을 메시지로 남기자

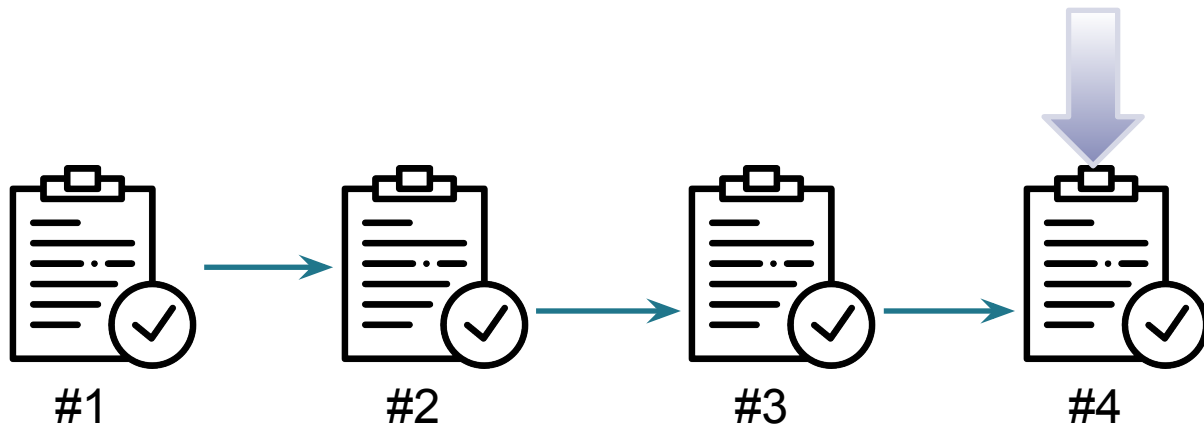
```
git add source_code.c
```

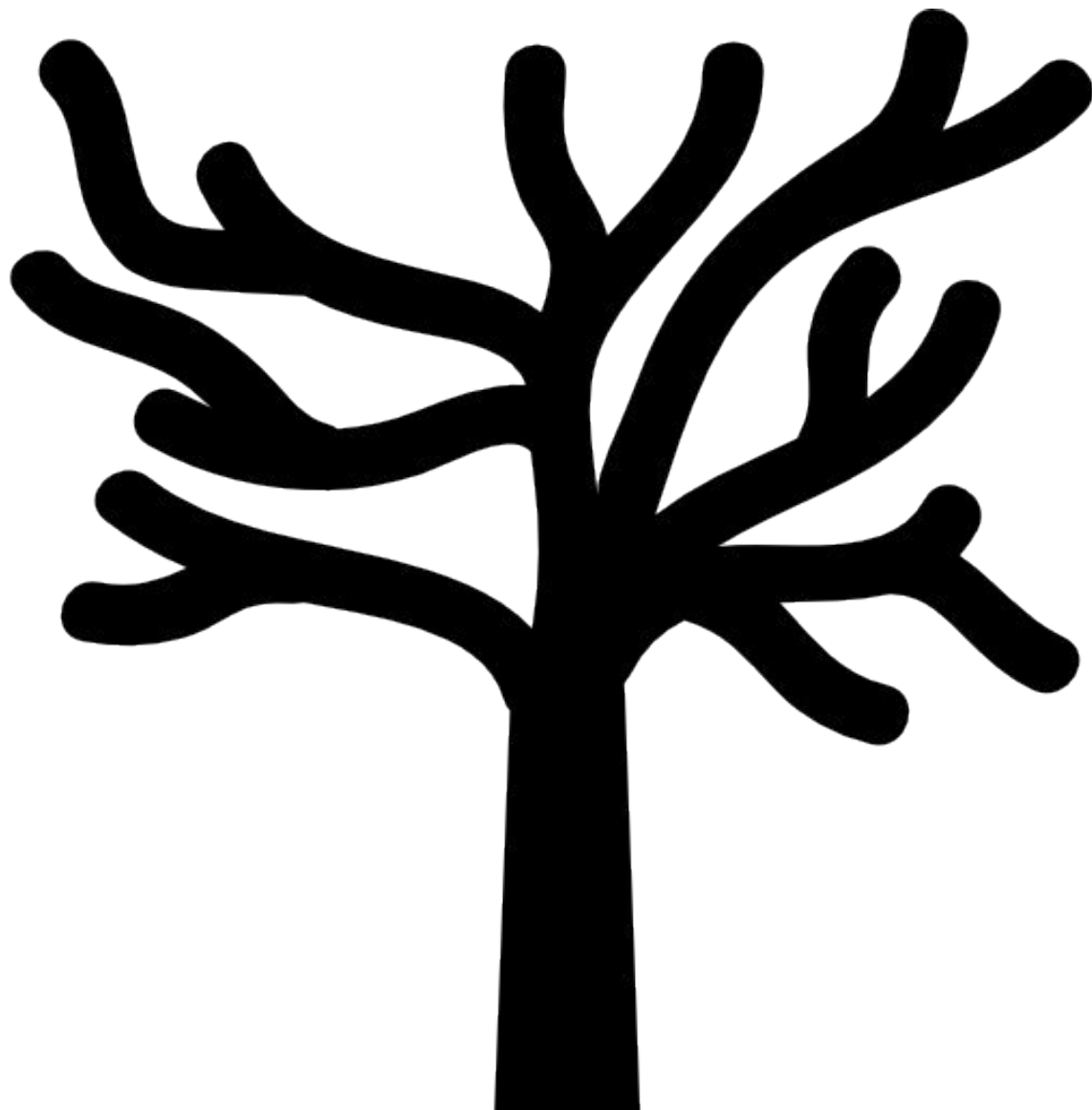
```
git commit -m "change comment from hello to bye"
```

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git shortlog
yckumon (4):
    first commit
    gugu_code #1
    gugu_code #2 : upto nine by nine
    change comment from hello to bye
```

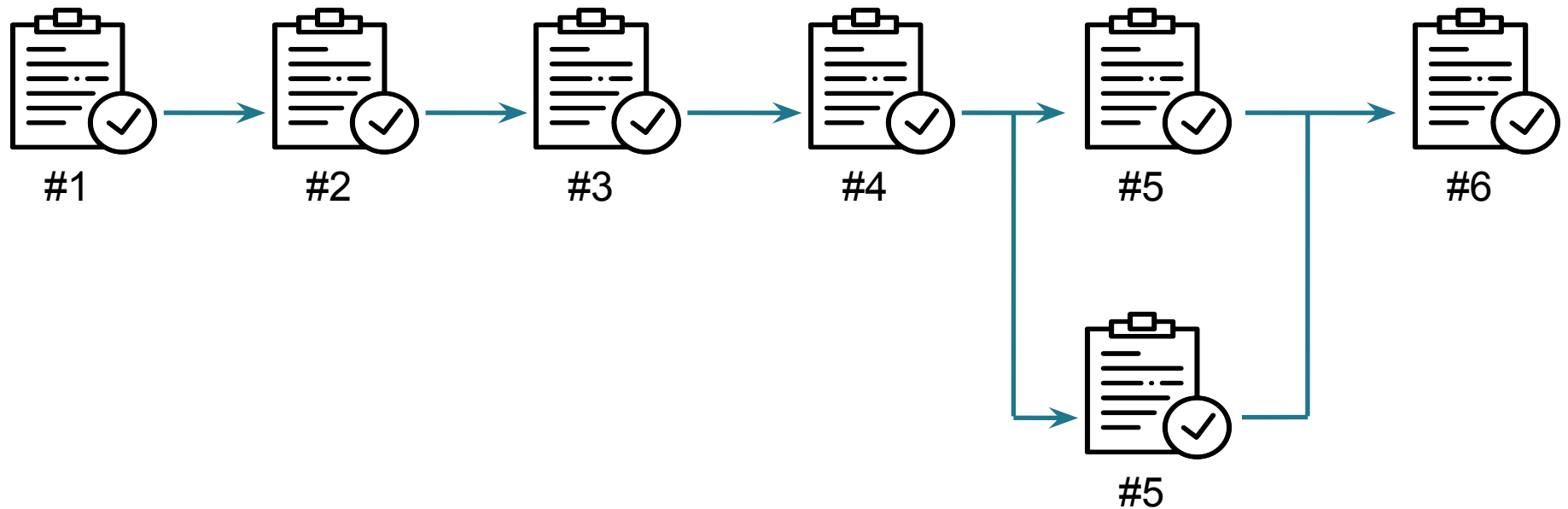
돌아보기 #1

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git shortlog
yckumon (4):
  first commit
  gugu_code #1
  gugu_code #2 : upto nine by nine
  change comment from hello to bye
```

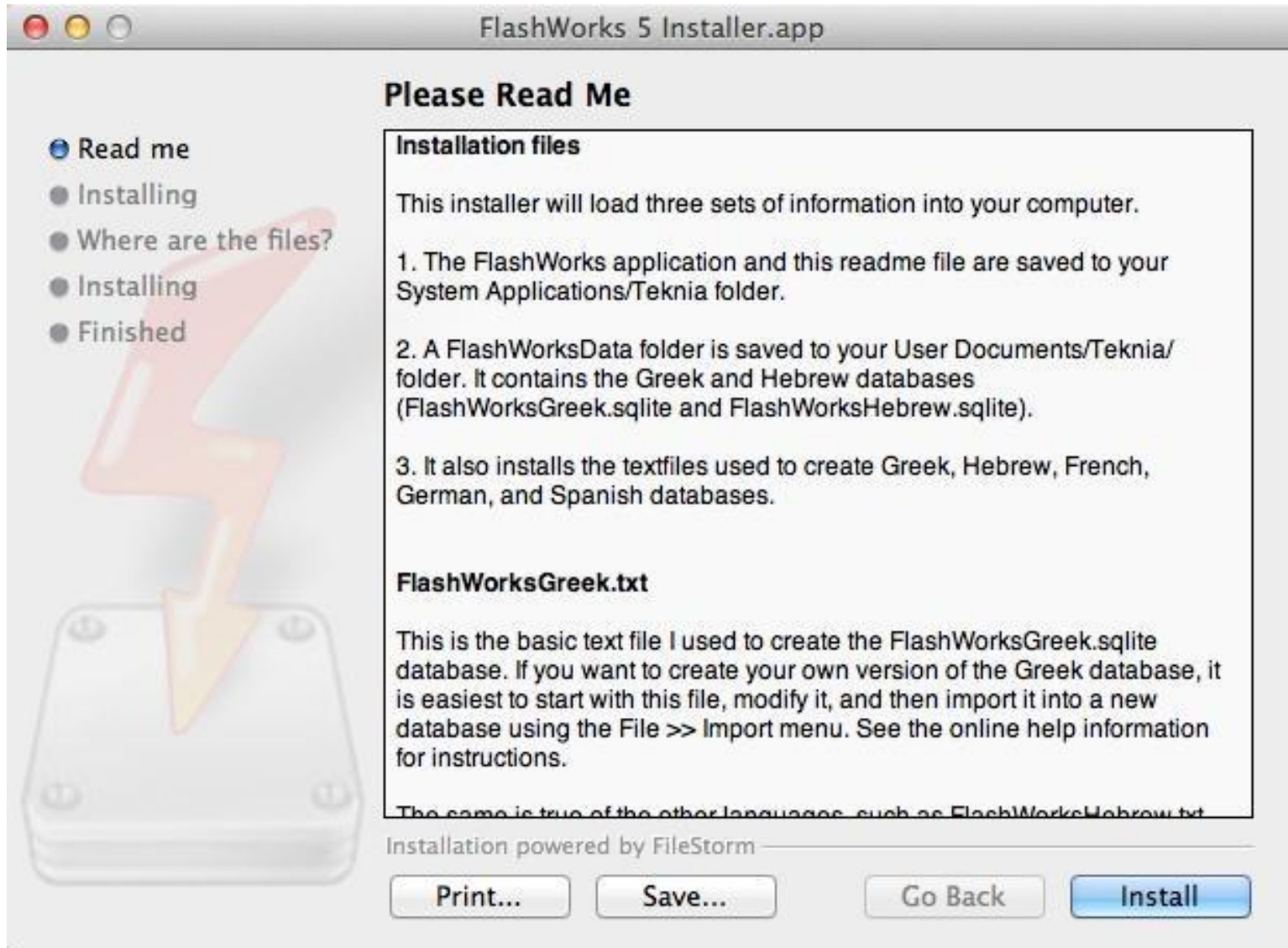




branch - 다른 작업을 해보자



branch - README 파일을 만들자



branch - README 파일을 만들자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git checkout -b write_readme
Switched to a new branch 'write_readme'
```

git checkout

다른 branch로 이동할 때 사용

ex > git checkout -b write_readme

'-b' 옵션을 넣어주면 새 브랜치를 만들고, 그 브랜치로 이동함

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (write_readme)
$ git branch
  master
* write_readme
```

git branch

- 현재 작업중인 브랜치, 생성되어 있는 branch 확인가능

branch - README 파일을 만들자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (write_readme)
$ ls
gugu_code.c  README.md  source_code.c
```

commit5 폴더의 파일들을 옮겨오자

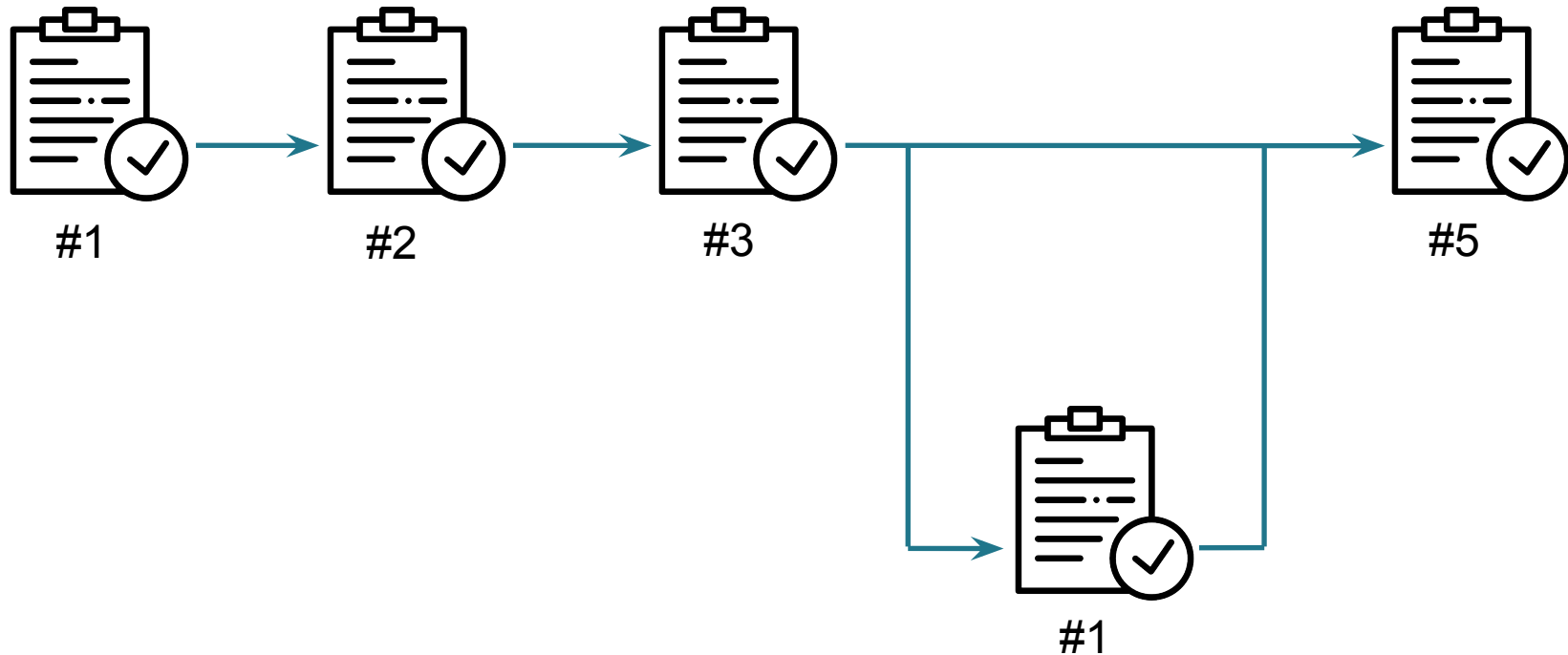
```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (write_readme)
$ git add .

Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (write_readme)
$ git commit -m "add README.md"
[write_readme 76f2ead] add README.md
1 file changed, 15 insertions(+)
create mode 100644 README.md
```

`git add .`

`git commit -m "add README.md"`

merge - 작업 합치기



merge 할 때는 **merge**가 되어야 할 브랜치(master)에서 명령을 실행한다

merge - 작업 합치기

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git merge write_readme --no-ff
Merge made by the 'recursive' strategy.
 README.md | 15 ++++++
 1 file changed, 15 insertions(+)
 create mode 100644 README.md
```

git merge <branch name>

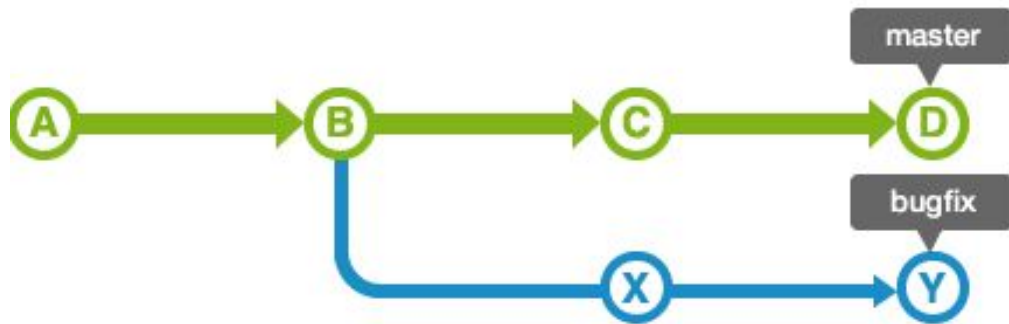
ex > git merge write_readme --no-ff

[--no-ff] 옵션은 fast forward merge의 자동 수행을 막는 명령

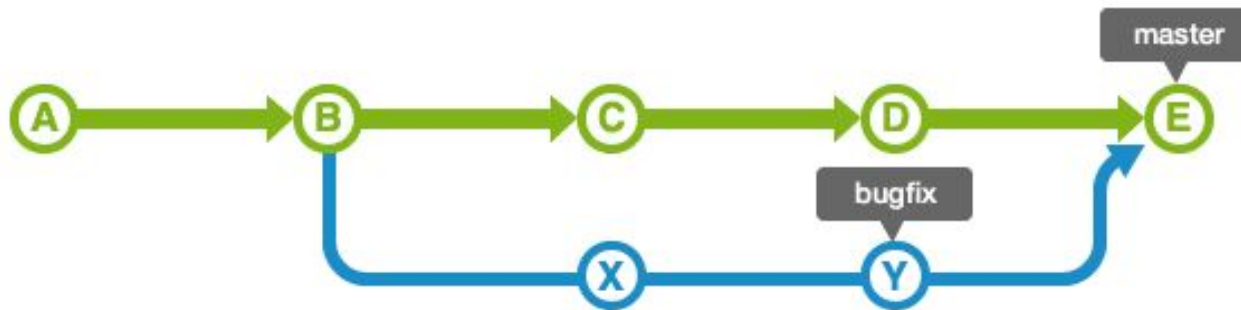
--no-ff를 하지 않아도 상관없다. 하지만 지금은 일단 --no-ff를 쓴다

```
Merge branch 'write_readme'
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
```

merge - 작업 합치기 case #1



bugfix 브랜치를 만든 뒤 master 브랜치에서 commit 발생



merge - 작업 합치기 case #2



bugfix 브랜치를 만든 뒤 master 브랜치에서 추가 commit 발생 x



fast forward 적용



fast forward 미적용

merge - 작업 합치기

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git log --pretty=oneline --graph
*   ca9a3a55f4e0abc75ec663ac80cb95e6180cb37c Merge branch 'write_readme'
/*  \
/*   * b84095d8f6bd719a1842b3f1c4e0544d8581418a add README.md
/*   *
/*   * 003a04c940b6ce84c47e7b24a71b29b5c7e0d8ba change comment from hello to bye
/*   * 9fb457888edc13063d24aef0ded1bbfe6b1b8a81 gugu_code #2 : upto nine by nine
/*   * c4f8b4279eab3ffa9eb17047208cc9fe1316f0e7 gugu_code #1
/*   * 7eedd42ac703679395cbacd58592c35c9fa23f55 first commit
```

git log --pretty=oneline --graph

지금까지 커밋 로그, 머지 기록을 CLI환경에서 그래프와 함께 보여줌

github로 나아가자

The screenshot shows a web browser window with the GitHub website. The user is logged in as 'hong geun'. The page displays the profile of a user named 'RedBean' (GreatBean), who joined on 22 Jun 2016. The profile includes a bio, a bio link, and an 'Edit profile' button. The repository list shows two repositories: 'kkeun-ma-rit-gi' and 'Surround360'. The 'New' button is highlighted with a red box.

hong geun

Your Repositories

GitHub, Inc. [US] | <https://github.com/GreatBean?tab=repositories>

Search GitHub

Pull requests Issues Gist

Overview Repositories 5 Stars 1 Followers 3 Following 6

Search repositories...

Type: All Language: All

New

RedBean
GreatBean
Add a bio
Edit profile
Joined on 22 Jun 2016

kkeun-ma-rit-gi
forked from imyaman/kkeun-ma-rit-gi
pull request 해보고 싶은 분들을 위한 놀이공간

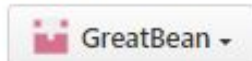
Surround360
forked from facebook/Surround360
Surround360 is Facebook's open source hardware and software for capturing stereoscopic 3D 360 video for VR. The repo contains hardware designs, as well as software for camera control and rendering.

github로 나아가자

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

Great repository names are short and memorable. Need inspiration? How about **congenial-broccoli**.

Description (optional)



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 let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾

Add a license: None ▾



Create repository

github로 나아가자

GreatBean / git_begin

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

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

Set up in Desktop

HTTPS SSH

https://github.com/GreatBean/git_begin.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

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

...or push an existing repository from the command line

```
git remote add origin https://github.com/GreatBean/git_begin.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

github로 나아가자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git remote add origin https://github.com/GreatBean/git_begin.git

Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git remote -v
origin https://test-git-training@github.com/GreatBean/git_begin.git (fetch)
origin https://test-git-training@github.com/GreatBean/git_begin.git (push)
```

git remote add <alias> <github repository url>

→git remote에 github url을 origin이라는 alias로 add하겠다

git remote -v

등록된 remote 확인

git push origin master

→origin에 master의 커밋들을 push하겠다

→github 계정 로그인 필요

github로 나아가자

GreatBean / git_begin

Watch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Pulse

Graphs

No description or website provided.

6 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

GreatBean committed with test-git-training Merge branch 'write_readme'

Latest commit 81847b6 an hour ago

README.md	add README.md	an hour ago
gugu_code.c	gugu_code #2 : upto nine by nine	3 hours ago
source_code.c	change comment from hello to bye	3 hours ago

README.md













github로 나아가자

GreatBean / [git_begin](#) Watch 1 Star 0 Fork 0

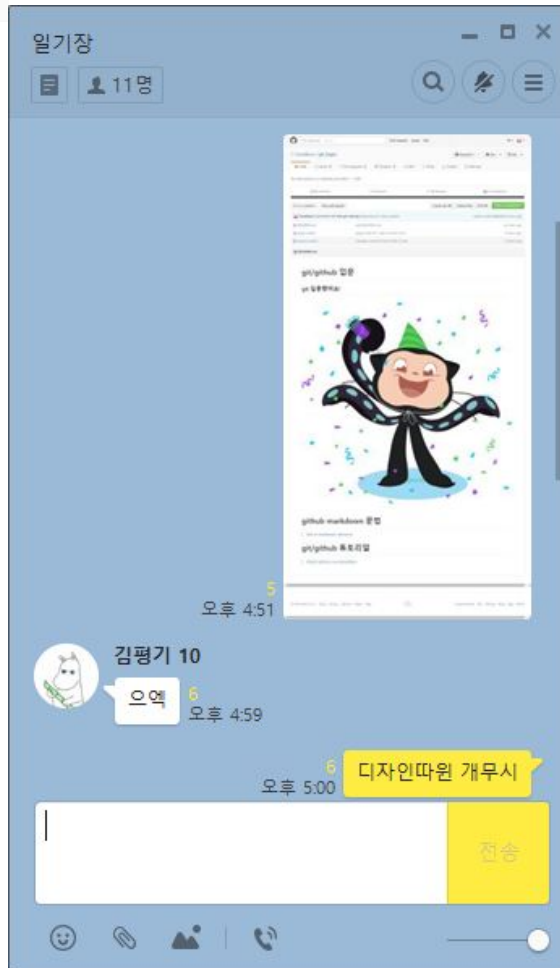
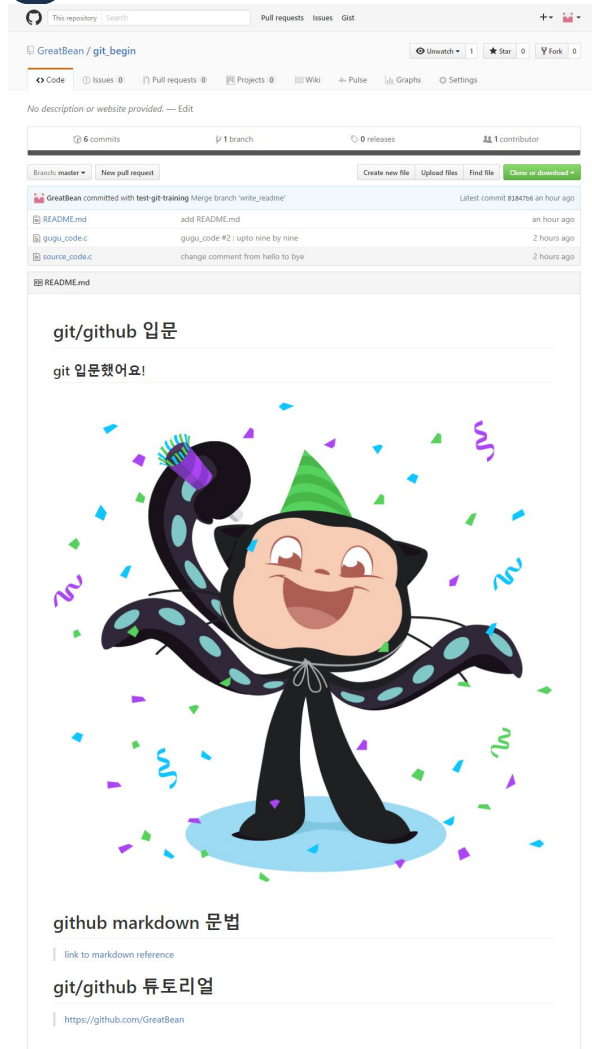
[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#)

Branch: master ▾


Commits on Oct 12, 2016

 Merge branch 'write_readme' GreatBean committed with test-git-training an hour ago	 81847b6 Code
 add README.md GreatBean committed with test-git-training an hour ago	 8e11150 Code
 change comment from hello to bye GreatBean committed with test-git-training 3 hours ago	 1f22e47 Code
 gugu_code #2 : upto nine by nine GreatBean committed with test-git-training 3 hours ago	 a6b75b4 Code
 gugu_code #1 GreatBean committed with test-git-training 3 hours ago	 4f016c0 Code
 first commit GreatBean committed with test-git-training 3 hours ago	 d9cdf72 Code

github로 나아가자



README를 수정하자

```
1 # git/github 입문
2 ## git 입문 했 어 요 !
3 
4 (https://octodex.github.com/images/welcometocat.png)
5
6
7
8 # github markdown 문법
9 >[link to markdown reference](https://guides.github.com/features/markdown/)
10
11
12
13
14 # git/github 튜토리얼
15 >[https://github.com/GreatBean](https://github.com/GreatBean)
~
~
~
```

3,4번 라인 - 옥토캣 이미지를 보이게 하는 코드

README를 수정하자 : modify_image

```
workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git checkout -b modify_image
Switched to a new branch 'modify_image'
```

modify_image 브랜치로 체크아웃

```
1 # git/github 입문
2 ## git 일문했 어요 !
3 
4
5
6 # github markdown 문법
7 >[link to markdown reference](https://guides.github.com/features/mastering-m
  arkdwn/)
8
9
10
11
12 # git/github 튜토리얼
13 >[https://github.com/GreatBean](https://github.com/GreatBean)
~
~
```

3,4번 라인의 코드를 지우고

```

```

으로 수정 후 저장

README를 수정하자 : modify_image

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (modify_image)
$ git commit -am "modify image size to 200"
[modify_image 257ccd9] modify image size to 200
1 file changed, 1 insertion(+), 3 deletions(-)
```

`git commit -am "modify size to 200"`

`git commit -am "메시지"`는 add, commit을 한번에 해준다

README를 수정하자_master

```
1 # git/github 입문
2 ## git 입문했 어 요 !
3 
4
5
6 # github markdown 문 법
7 >[link to markdown reference](https://guides.github.com/features/mastering-m
  arkdwn/)
8
9
10
11
12 # git/github 튜 토 리 얼
13 >[https://github.com/GreatBean](https://github.com/GreatBean)
~
~
```

3,4번 라인의 코드를 지우고

으로 수정

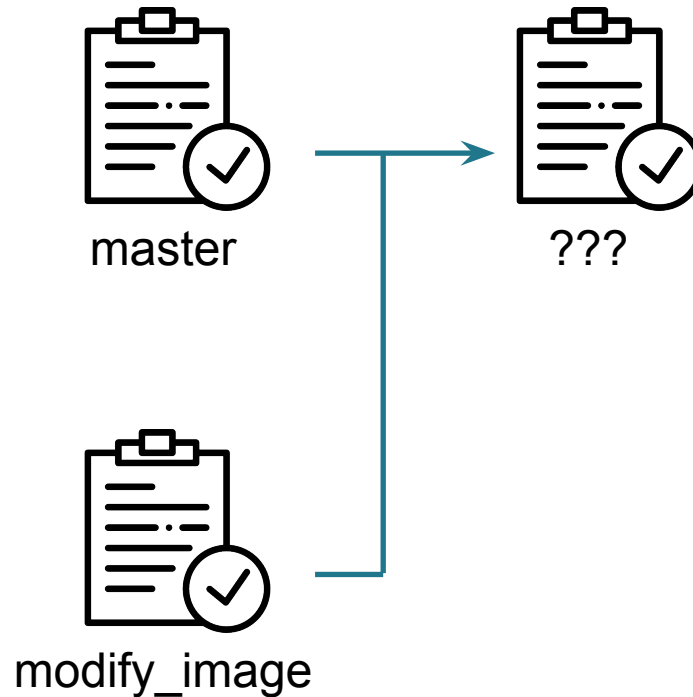
README를 수정하자_master

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git commit -am "modify image size to 300"
[master 1015997] modify image size to 300
1 file changed, 1 insertion(+), 3 deletions(-)
```

`git commit -am "modify size to 300"`

`git commit -am "메시지"`는 add, commit을 한번에 해준다

README를 수정하자_충돌 해결



master는 300, modify_image는 200 똑같은 파일인데 merge???

Let's do it~!!!

README를 수정하자 충돌 해결

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git merge modify_image
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master|MERGING)
$ |
```

CONFLICT (content) : Merge conflict in README.md

Automatic merge failed; fix conflicts and then commit the result.

충돌이 발생했다!!! 그리고 master|MERGING으로 상태변화가 일어났다
일단, README.md 파일을 열어보자

vi README.md

```
# git/github 입문
## git 입문 했 어 요 !
<<<<<<< HEAD

=====

>>>>>>> modify_image
```

README를 수정하자_충돌 해결

```
# git/github 입문
## git 입문했어요!
<<<<<<< HEAD

=====

>>>>>>> modify image
```

HEAD는 현재 git이 가리키고있는 작업 루트

README를 수정하자_충돌 해결

컴퓨터가 내린 결론

"똑같은 내용인데 어쩌라는거야? 난 모르겠고 니가 결정해!"

README를 수정하자_충돌 해결

```
# git/github 입문
## git 입문했어요!
<<<<<< HEAD

=====

>>>>>> modify_image
```

둘 중 적용시키고 싶은 내용만 남기자

```
# git/github 입문
## git 입문했어요!

```

저장하고 다시 add, commit

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master|MERGING)
$ git commit -am "fix image to 200 final"
[master c0d2c84] fix image to 200 final
```

README를 수정하자_충돌 해결


```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git log --pretty=oneline --graph
*      c0d2c8457705d5501e4f91df85eb7406227b14f0 fix image to 200 final
|
| \
|  * 257ccd943cf0ac6e48aa1db10108d87d4a7e5993 modify image size to 200
|  * 1015997e9cd8df3a897bcdd8c6763defaee5c807 modify image size to 300
|  /
| *   81847b61415243d810361d7a6cf16dcc1f978458 Merge branch 'write_readme'
|   /
|  * 8e1115081e3d9c4bf70f7a0ec6155450950b2147 add README.md
| /
* 1f22e47fa2dad5388b63bd0dcd7b5014adea0940 change comment from hello to bye
* a6b75b4a6f681a69766adf051e5a9223fd151377 gugu_code #2 : upto nine by nine
* 4f016c068f22e5a676831cbb388336d690607e1e gugu_code #1
* d9cdf72031d36780cb4c802c614dc67db546254a first commit
```

아름다운 그래프가 완성됐다...

제대로 수정되었는지 확인하자

git push origin master

README를 수정하자_충돌 해결

 This repository Search Pull requests Issues Gist + -

GreatBean / git_begin Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)

No description or website provided. — Edit

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download


GreatBean fix image to 200 final Latest commit c0d2c84 3 minutes ago

README.md	modify image size to 200	32 minutes ago
gugu_code.c	gugu_code #2 : upto nine by nine	4 hours ago
source_code.c	change comment from hello to bye	4 hours ago

README.md

git/github 입문

git 입문했어요!



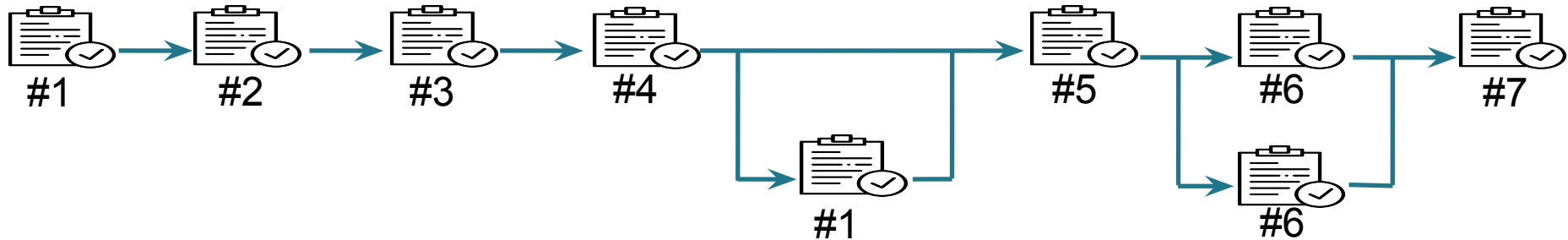
github markdown 문법

[link to markdown reference](#)

git/github 튜토리얼

<https://github.com/GreatBean>

돌아보기 #2



```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_begin (master)
$ git log --pretty=oneline --graph
* c0d2c8457705d5501e4f91df85eb7406227b14f0 fix image to 200 final
/* 257ccd943cf0ac6e48aa1db10108d87d4a7e5993 modify image size to 200
* | 1015997e9cd8df3a897bcdd8c6763defaee5c807 modify image size to 300
/* 81847b61415243d810361d7a6cf16dcc1f978458 Merge branch 'write_readme'
/* 8e1115081e3d9c4bf70f7a0ec6155450950b2147 add README.md
/* 1f22e47fa2dad5388b63bd0dcd7b5014adea0940 change comment from hello to bye
* a6b75b4a6f681a69766adf051e5a9223fd151377 gugu_code #2 : upto nine by nine
* 4f016c068f22e5a676831cbb388336d690607e1e gugu_code #1
* d9cdf72031d36780cb4c802c614dc67db546254a first commit
```

github 미니게임

The screenshot shows the GitHub interface for the repository 'imyaman / kkeun-ma-rit-gi'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there's a 'Watch' button and a navigation bar with links for 'Code', 'Issues' (highlighted with 3 issues), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Pulse', and 'Graphs'. The main heading is '놀이에 참여하는 분들에게 브랜치 만들도록 하기 #21'. Below it, a green 'Open' button indicates the issue status, followed by the text 'imyaman opened this issue on 14 Mar 2015 · 0 comments'. A comment by 'imyaman' from 14 Mar 2015 is shown, with a profile picture and a 'Owner' label. The comment text is in Korean and lists five steps for creating and managing branches. At the bottom of the comment, there is a smiley face emoji and the number '1'.

This repository Search

Pull requests Issues Gist

imyaman / kkeun-ma-rit-gi Watch

<> Code Issues 3 Pull requests 0 Projects 0 Wiki Pulse Graphs

놀이에 참여하는 분들에게 브랜치 만들도록 하기 #21

Open imyaman opened this issue on 14 Mar 2015 · 0 comments

imyaman commented on 14 Mar 2015 Owner +😊

앞으로 새 단어를 추가하실 때에는 각자 dev 브랜치를 만들어주세요.

1. imyaman/kkeun-ma-rit-gi 의 최신 내용으로 업데이트합니다.
2. dev 브랜치를 만듭니다.
3. dev 브랜치에서 새 단어를 추가합니다.
4. dev 브랜치를 master 브랜치에 merge 합니다.
5. pull request를 생성합니다.

😊 1

<https://github.com/imyaman/kkeun-ma-rit-gi>

github 미니게임



imyaman / kkeun-ma-rit-gi

<> Code 3 Issues 0 Pull requests

Branch: master kkeun-ma-rit-gi / wordlist.txt

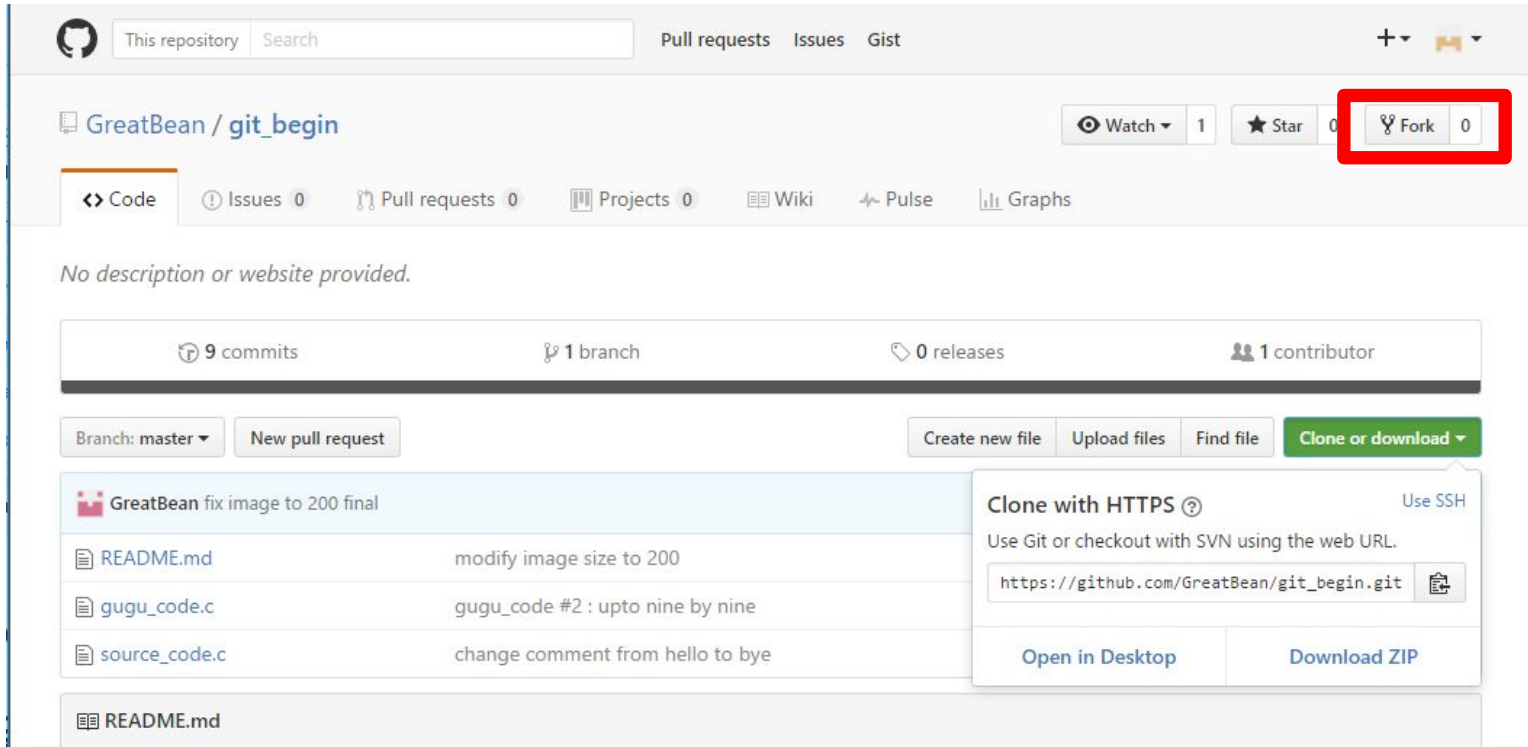
HaeBong add yellow after money

26 contributors

77 lines (76 sloc) | 511 Bytes

```
1 github
2 blob
3 blank
4 brazil
5 legal
6 linear
7 roar
8 rails
9 scopes
10 smile
```


fork - 친구의 코드를 가져오자



친구의 github repository로 가서 'fork'버튼을 누른다

fork - 친구의 코드를 가져오자

The screenshot shows a GitHub repository page for 'test-git-training / git_begin', which is a fork of 'GreatBean/git_begin'. The repository name is highlighted with a red box. The page includes a search bar, navigation tabs for Pull requests, Issues, and Gist, and a header with Watch (0), Star (0), and Fork (1) buttons. Below the header, there are tabs for Code, Pull requests (0), Projects (0), Wiki, Pulse, Graphs, and Settings. The main content area shows 'No description or website provided.' and a summary of the repository: 9 commits, 1 branch, 0 releases, and 1 contributor. A 'New pull request' button is visible. The commit history shows a recent commit by 'GreatBean' titled 'fix image to 200 final' with the latest commit hash 'c0d2c84' 17 hours ago. Below the commit history, there is a table of files:

File	Commit Message	Time
README.md	modify image size to 200	17 hours ago
gugu_code.c	gugu_code #2 : upto nine by nine	21 hours ago
source_code.c	change comment from hello to bye	21 hours ago

자신의 github 페이지로 돌아오면 똑같지만 똑같지 않은 페이지가 보인다

fork - 친구의 코드를 가져오자

The screenshot shows the GitHub interface for a repository named 'test-git-training / git_begin', which is a fork of 'GreatBean/git_begin'. The repository has 9 commits, 1 branch, 0 releases, and 1 contributor. A red box highlights the 'Clone or download' button. Below the repository information, there is a table of files and their commit history.

File	Commit Message	Time
GreatBean	fix image to 200 final	Latest commit c0d2c84 17 hours ago
README.md	modify image size to 200	17 hours ago
gugu_code.c	gugu_code #2 : upto nine by nine	21 hours ago
source_code.c	change comment from hello to bye	21 hours ago

Clone or download 버튼을 누르자

fork - 친구의 코드를 가져오자

The screenshot shows a GitHub repository page for 'test-git-training / git_begin', which is a fork of 'GreatBean/git_begin'. The repository has 9 commits, 1 branch, 0 releases, and 1 contributor. A modal dialog titled 'Clone with HTTPS' is open, displaying the URL 'https://github.com/test-git-training/git_b' and a 'Copy to clipboard' button. The dialog also includes options for 'Open in Desktop' and 'Download ZIP'.

test-git-training / git_begin
forked from GreatBean/git_begin

Watch 0 Star 0 Fork 1

Code Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is even with GreatBean:master.

GreatBean fix image to 200 final

README.md	modify image size to 200
gugu_code.c	gugu_code #2 : upto nine by nine
source_code.c	change comment from hello to bye

21 hours ago

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL

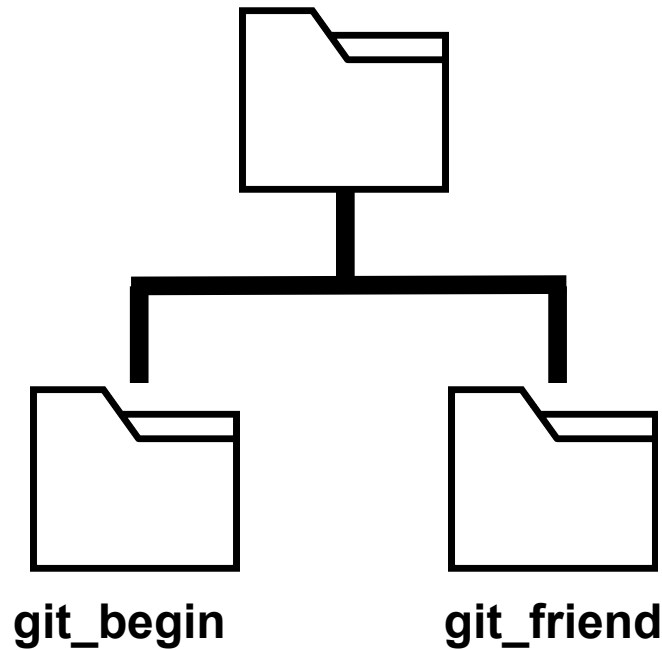
https://github.com/test-git-training/git_b

Copy to clipboard

Open in Desktop Download ZIP

.git 주소를 복사하자

fork - 친구의 코드를 가져오자



git_begin이 아닌 새 폴더를 만든다

이 때, **.git**이 없는 폴더(**git init**을 하지 않은 폴더)여야 한다

fork - 친구의 코드를 가져오자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop  
$ mkdir git_friend; cd git_friend;  
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_friend  
$ |
```

`mkdir git_friend; cd git_friend;`

코드를 가져오기 위한 폴더를 만들고, 이동한다

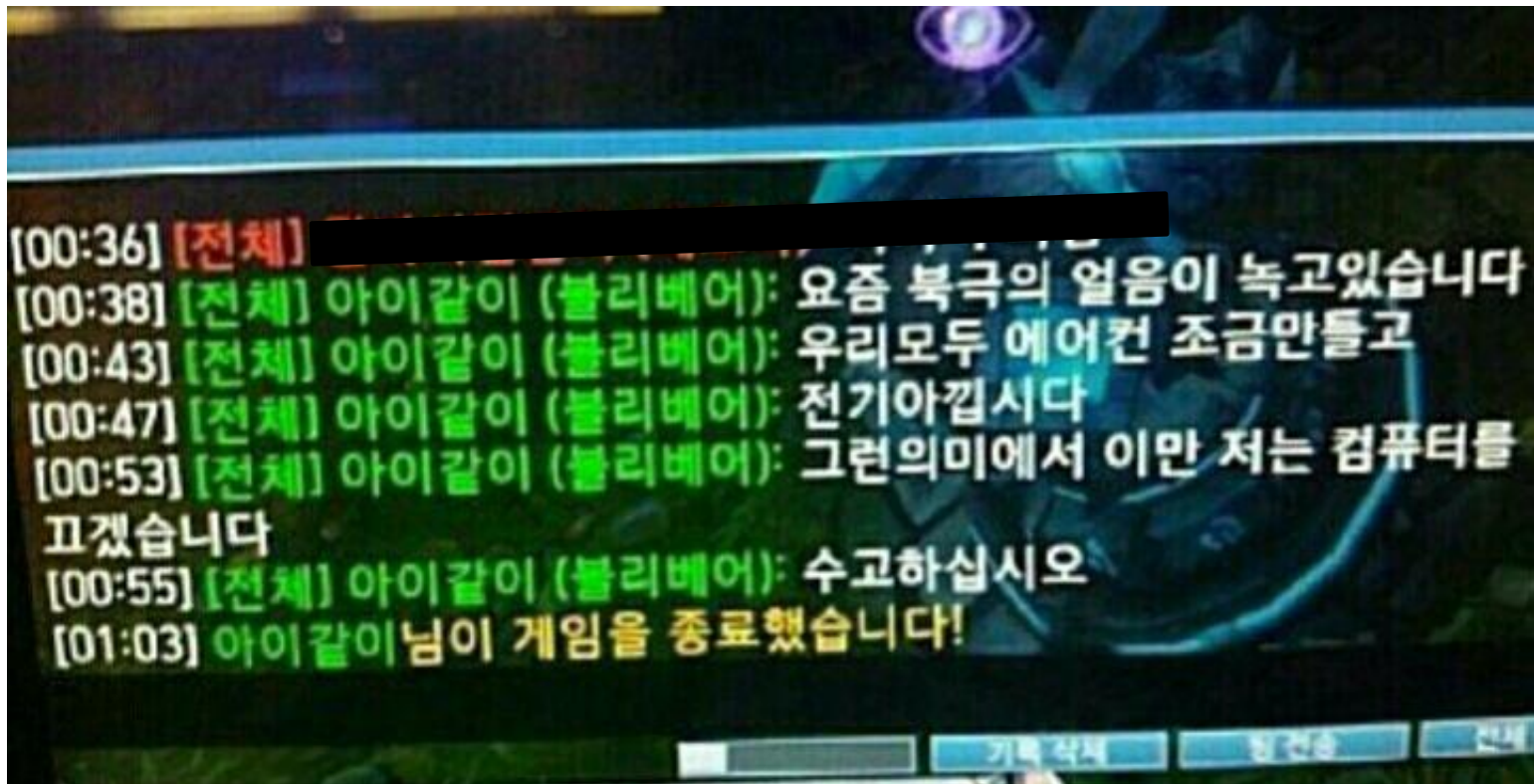
```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_friend  
$ git clone github.com-yckumon:test-git-training/git_begin.git  
Cloning into 'git_begin'...  
remote: Counting objects: 23, done.  
remote: Compressing objects: 100% (18/18), done.  
remote: Total 23 (delta 4), reused 23 (delta 4), pack-reused 0  
Receiving objects: 100% (23/23), done.  
Resolving deltas: 100% (4/4), done.  
Checking connectivity... done.
```

git init을 하지 말고

`git clone <복사해뒀던 .git 주소>를 입력한다`

실습 자료 작성을 위해 1PC, 2계정 접근에 따라 입력 방식이 조금 다릅니다
실습하시는 분들은 설명글에 적힌대로 따라하시면 됩니다

왜 fork → clone 할까?



fork - 친구의 코드를 가져오자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_friend/git_begin (master)
$ ls -al | grep .
total 11
drwxr-xr-x 1 Workingman 197121 0 10월 13 11:32 ./
drwxr-xr-x 1 Workingman 197121 0 10월 13 11:32 ../
drwxr-xr-x 1 Workingman 197121 0 10월 13 11:32 .git/
-rw-r--r-- 1 Workingman 197121 202 10월 13 11:32 gugu_code.c
-rw-r--r-- 1 Workingman 197121 339 10월 13 11:32 README.md
-rw-r--r-- 1 Workingman 197121 81 10월 13 11:32 source_code.c
```

clone 한 폴더의 파일 목록

→ git init을 안했지만 .git이 있다!

새로운 작업을 위해 브랜치를 만들고 이동하자

git checkout -b <nickname>

ex > git checkout -b yckumon

내가 만든 코드를 넣자(코드를 만드는 과정이라고 생각하기)

echo "<소스코드>" > friend.txt

ex> echo "Dear my friend" > friend.txt

fork - 친구의 코드를 가져오자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_friend/git_begin (yckumon)
$ echo "Dear my friend" > friend.txt

Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_friend/git_begin (yckumon)
$ cat friend.txt
Dear my friend
```

제대로 생성된게 확인 되면 add, commit하자

```
Workingman@DESKTOP-9A8F0PB MINGW64 ~/Desktop/git_friend/git_begin (yckumon)
$ git push origin yckumon
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 369 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com-yckumon:test-git-training/git_begin.git
* [new branch]      yckumon -> yckumon
```

자신의 github로 push

이때, git push origin <nickname> (**master아님**)

PR - 내 마음을 받아줘

test-git-training / git_begin
forked from GreatBean/git_begin

Watch 0 Star 0 Fork 1

Code Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

9 commits 2 branches 0 releases 1 contributor

Your recently pushed branches:

88 yckumon (2 minutes ago) Compare & pull request

Branch: master New pull request

Switch branches/tags

Find or create a branch...

Branches Tags

✓ master

ychumon

Create new file Upload files Find file Clone or download

Pull request Compare

	Latest commit c0d2c84 17 hours ago
image size to 200	18 hours ago
code #2 : upto nine by nine	22 hours ago
comment from hello to bye	21 hours ago

<nickname>으로 된 브랜치로 push된 것을 확인가능
해당 브랜치를 클릭해서 이동하자

PR - 내 마음을 받아줘

Your recently pushed branches:

yckumon (4 minutes ago)

Compare & pull request

Branch: yckumon ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is 1 commit ahead of GreatBean:master.

Pull request Compare

test-git-training insert my code & pull request test

Latest commit ae44fb4 7 minutes ago


README.md	modify image size to 200	18 hours ago
friend.txt	insert my code & pull request test	7 minutes ago
gugu_code.c	gugu_code #2 : upto nine by nine	22 hours ago
source_code.c	change comment from hello to bye	22 hours ago

pull request(PR)을 보내자!


PR - 내 마음을 받아줘

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base fork: GreatBean/git_begin ▾ base: master ▾ ... head fork: test-git-training/git_begin ▾ compare: yckumon ▾

✓ **Able to merge.** These branches can be automatically merged.



insert my code & pull request test

Write

Preview

AA ▾ B i “ <> 🔗 ⋮ ⋮ ✓ ↩ @ ★

friend.txt를 추가했습니다.

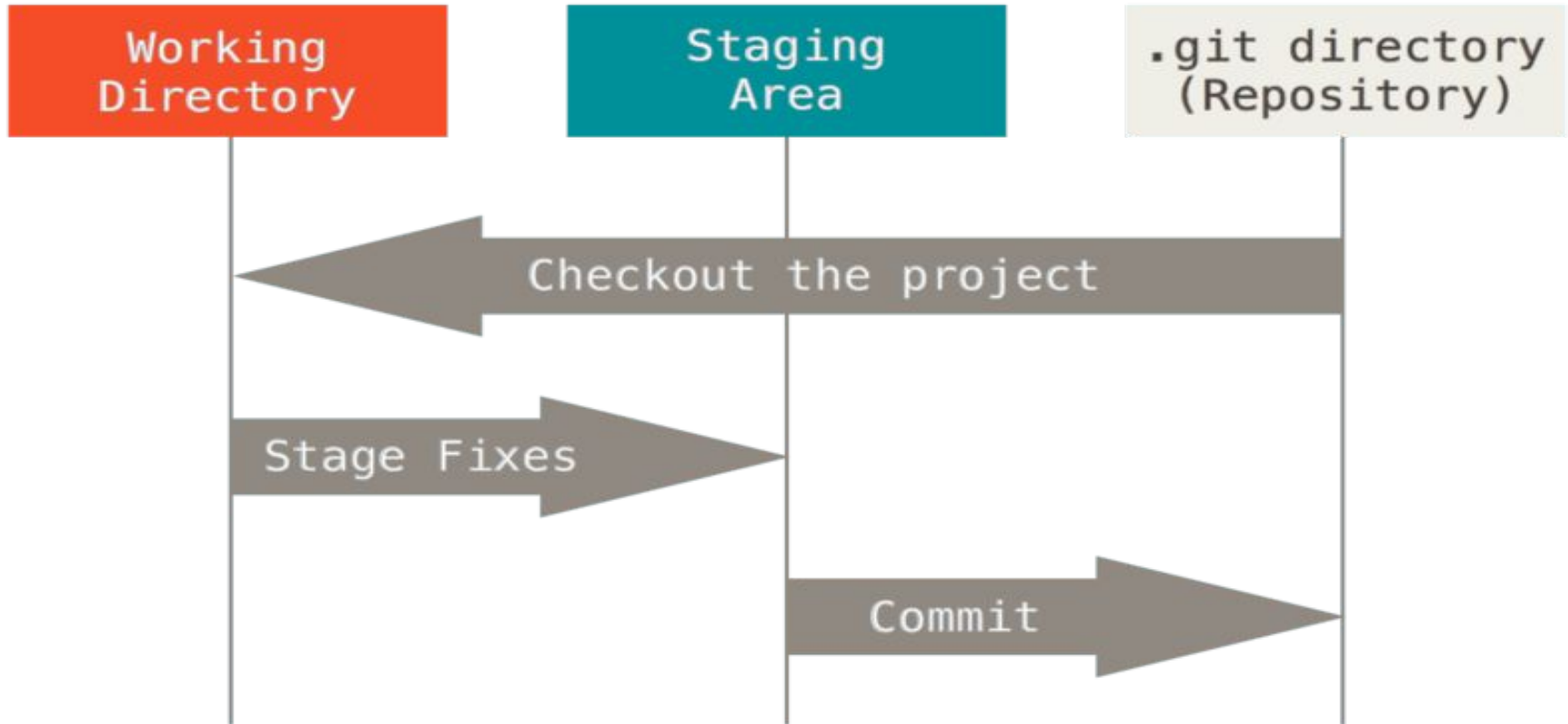
제발 PR 받아주세요 ㅠㅠ

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

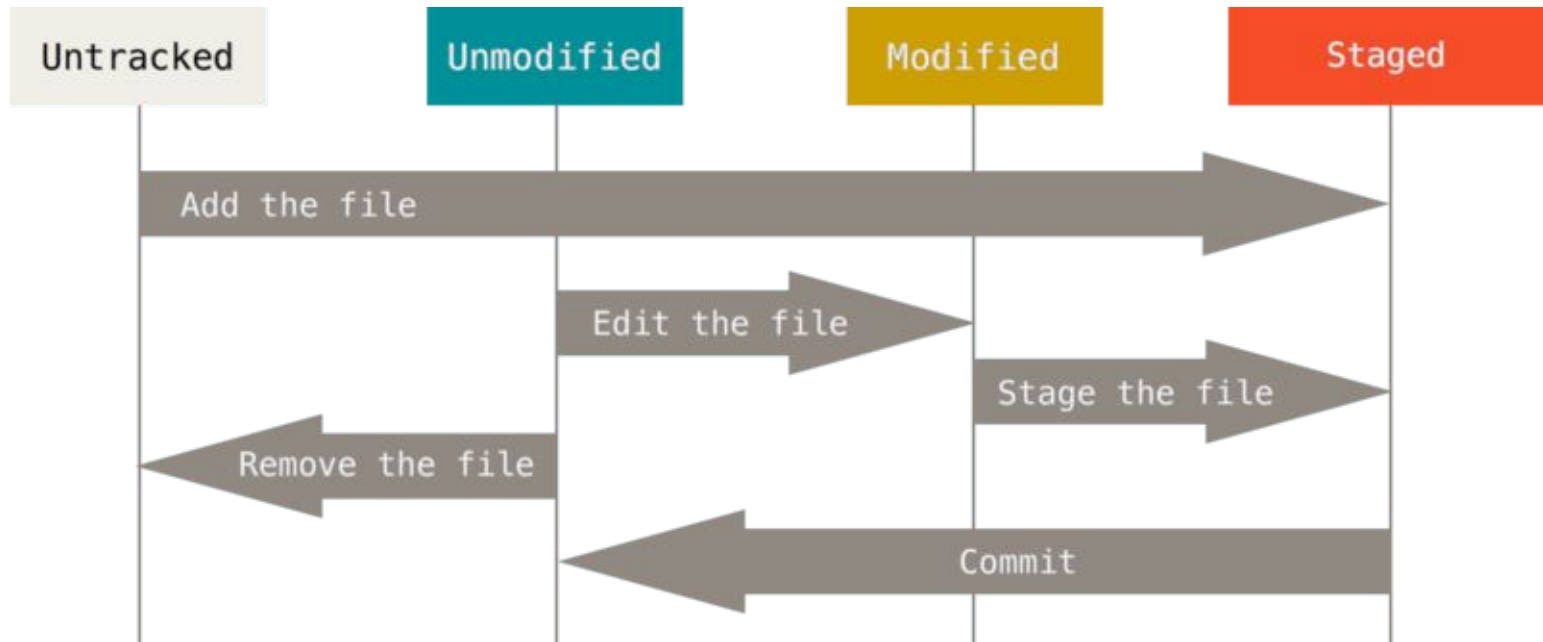
지금은...? Git의 3 영역



작업 흐름

Working Directory -> Staging Area -> .git directory

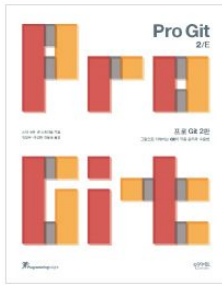
지금은...? Git의 라이프사이클



더 공부하려면

e-book

pro-git <https://git-scm.com/book/ko/v2>



MD의선택 이벤트 무료배송

프로 Git 그림으로 이해하는 Git의 작동 원리와 사용법 Programming Insight | 2판

스캇 사콘, 벤 스트라움 지음 | 박창우, 이성환, 최용재 옮김 | 인사이트 | 2016년 03월 30일 출간

이 책의 개정정보: 2013년 출간

★★★★★ 리뷰 1개 [리뷰쓰기](#)

정가: 26,000원

판매가: **23,400원** [10%↓ 2,600원 할인]

이 상품

· 개발자

생활코딩 git 강의

<https://opentutorials.org/course/1492>

소셜 코딩으로 이끄는 GitHub 실천기술

Git과 GitHub를 직접 따라하며 배운다

Full Stack
개발자를 위한
Git과 GitHub



버전 관리/브랜치/Fork/merge
Issue/버그 관리/Markdown
소셜 코딩/Jenkins/GitHub Flow
Git Flow/hub 명령어/코드 리뷰

오재호 지음 | 인사이트 | 2016년 03월 30일 출간



