

# Let's git it started

Git/GitHub를 통해 친해져보아요!

# Let's git it started

---

Git/GitHub와 Markdown 문법 공부

협업을 하기 위한 첫 걸음

나아가 Git/GitHub를 통해 결과물 만들기



# Contents

01

Let's git it started!

Git/GitHub란?

Git 명령 소개

GitHub 기능 소개

02

Dive into 'Git'

우리 서로 더 알아보기

마크다운 문법이란?

Git 실습하기

03

Feedback/Review

Pull Request 작성하기

리뷰하기

# Chapter1. Let's git it started!

01 Git/GitHub는 무엇일까?

02 Git의 명령 소개

add, commit, push, branch, clone

03 GitHub 기능 소개

pull request, issue, review, merge

# 01 Git/GitHub는 무엇일까?

## 버전 관리



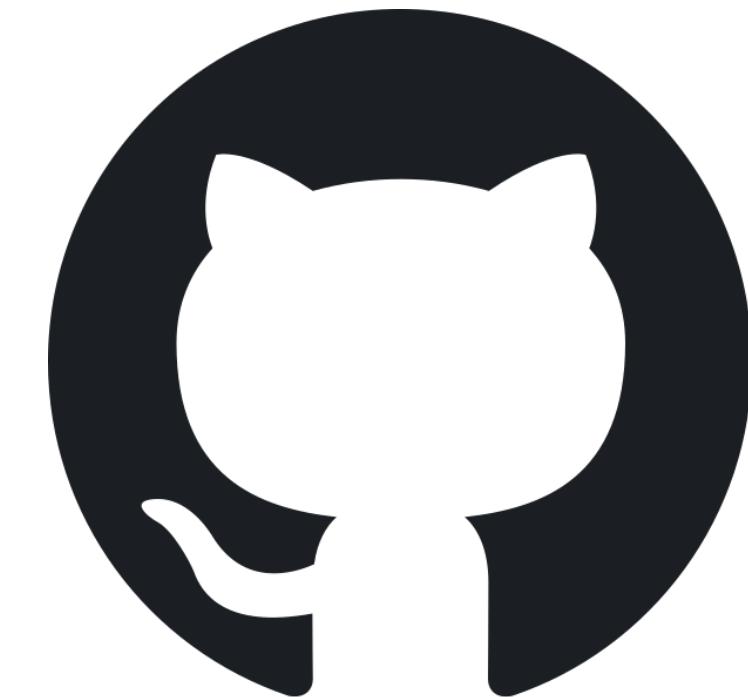
# 01 Git/GitHub는 무엇일까?



- 버전관리 시스템  
(version control system)
- 자유롭게 코드를 더하고 빼고!
- 여러 사람들이 동시에 개발 하기에 편리함

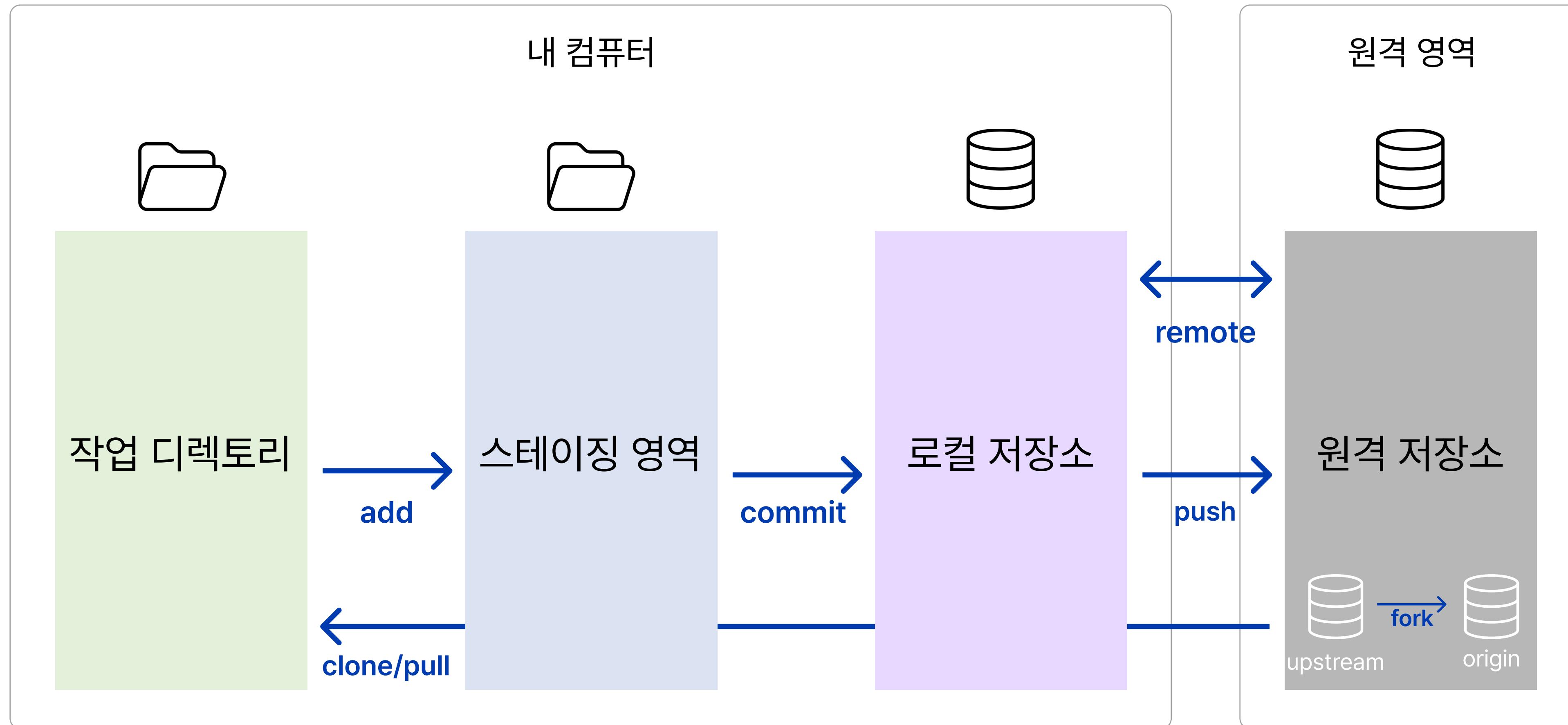
- Git의 데이터를 저장하는 서버
- 커뮤니티 기능이 잘 되어 있음
- 코드를 보면서 의견을 나눔
- 오픈 소스의 발전

# 01 Git/GitHub는 무엇일까?



여러 대의 컴퓨터들과 하나의 메인 서버가 소통하는 관계  
Git GitHub

## 02 Git의 명령어를 알아보자!



## 02 Git의 명령어를 알아보자!

: add, commit, push, branch, clone

1. **add**: 저장을 원하는 파일들을 묶어 올리겠다. / 스테이지에 올린다.

(내가 구현한 기능을 우선 너에게 전달하기 위한 준비를 할게!)

2. **commit**: 저장하는 파일들을 묶어서 save 하는 행동

(내가 구현한 기능들은 말이지... 'horse'야!)

3. **push**: 내 컴퓨터에 저장된 작업 내용을 github에 업로드한다.

(내가 'horse'를 구현했는데 너희들에게 보낼게! 확인 부탁해~)

## 02 Git의 명령어를 알아보자!

: add, commit, push, branch, clone

4. **branch**: 개발 영역을 분담해서 작업할 때, 충돌 예방 방지를 위해 개발 branch를 생성해, 기존에 있던 내용을 가져와 개별적으로 작업

(아, 너가 'horse'를 구현했으니! 나는 'dolphin'을 구현해야겠군!)

5. **clone**: 원격 저장소 내려 받기 (저장소 복제하기)

(horse v1 기반으로 v1.1을 만드려는데 기존에 작성한 코드를 참고해서 해야겠다.)

## 03 GitHub의 기능을 알아보자!

: pull request, review, merge

### 1. **pull request**: GitHub의 강력한 협업 기능

(세상 사람들! 우리 'horse'가 걸을 수 있어요~ 한번 이상 없는지 봐주세요!)

### 2. **review**: pull request 내용을 점검하는 기능, 리뷰를 통한 공동 학습

(오 네! 알겠습니다. 음 'horse'의 이 부분은 좀 더 수정봐야할 것 같아요!)

### 3. **merge**: 각각의 branch를 합치는 과정

(오! 'horse' 가 사용하기 괜찮네요! Merge 해드릴게요! 너 우리의 동료가 되어라!)

# Chapter2. Dive into 'Git'

01 우리 서로 더 알아보기

02 마크다운 문법이란?

03 Git 실습하기

## 01 우리 서로 더 알아보깃

Step1. 자기 소개글을 작성 후 Git을 이용하여 GitHub에 글 올리기

Step2. Pull Request 작성하기

Step3. 서로의 소개글을 읽으면서 궁금하거나,  
공감가는 내용이 있다면 Review를 통해 소통하기

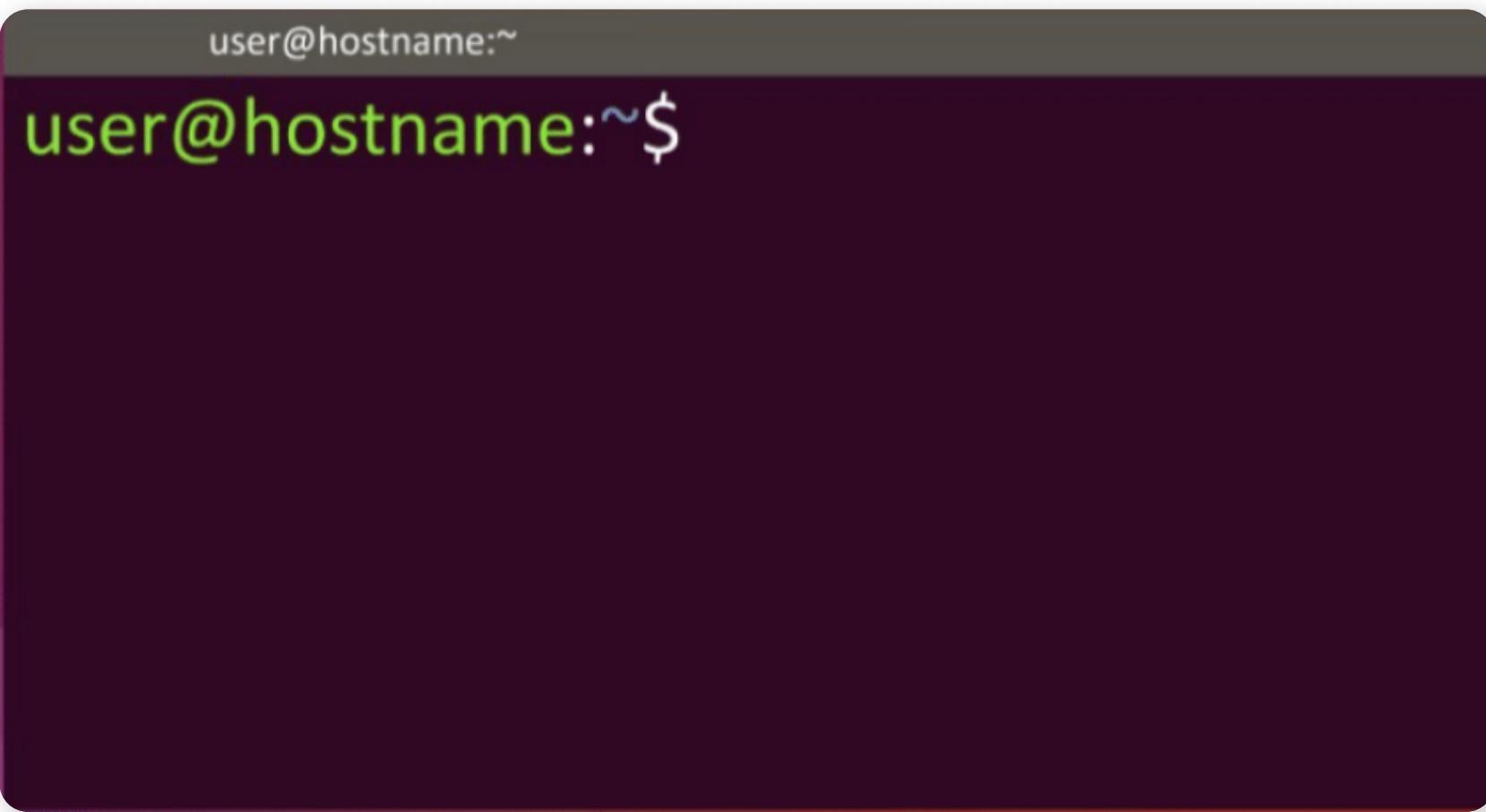
## 02 마크다운 문법이란?

- 2004년 존 그루버에 의해 만들어진 텍스트 기반 마크업 언어
- HTML로 쉽게 변환 가능
- 간단한 구조의 문법
- GitHub의 저장소에 관한 정보를 기록하는 **README.md**로 인해 주목 받고 있음

## 03 Git 실습하기

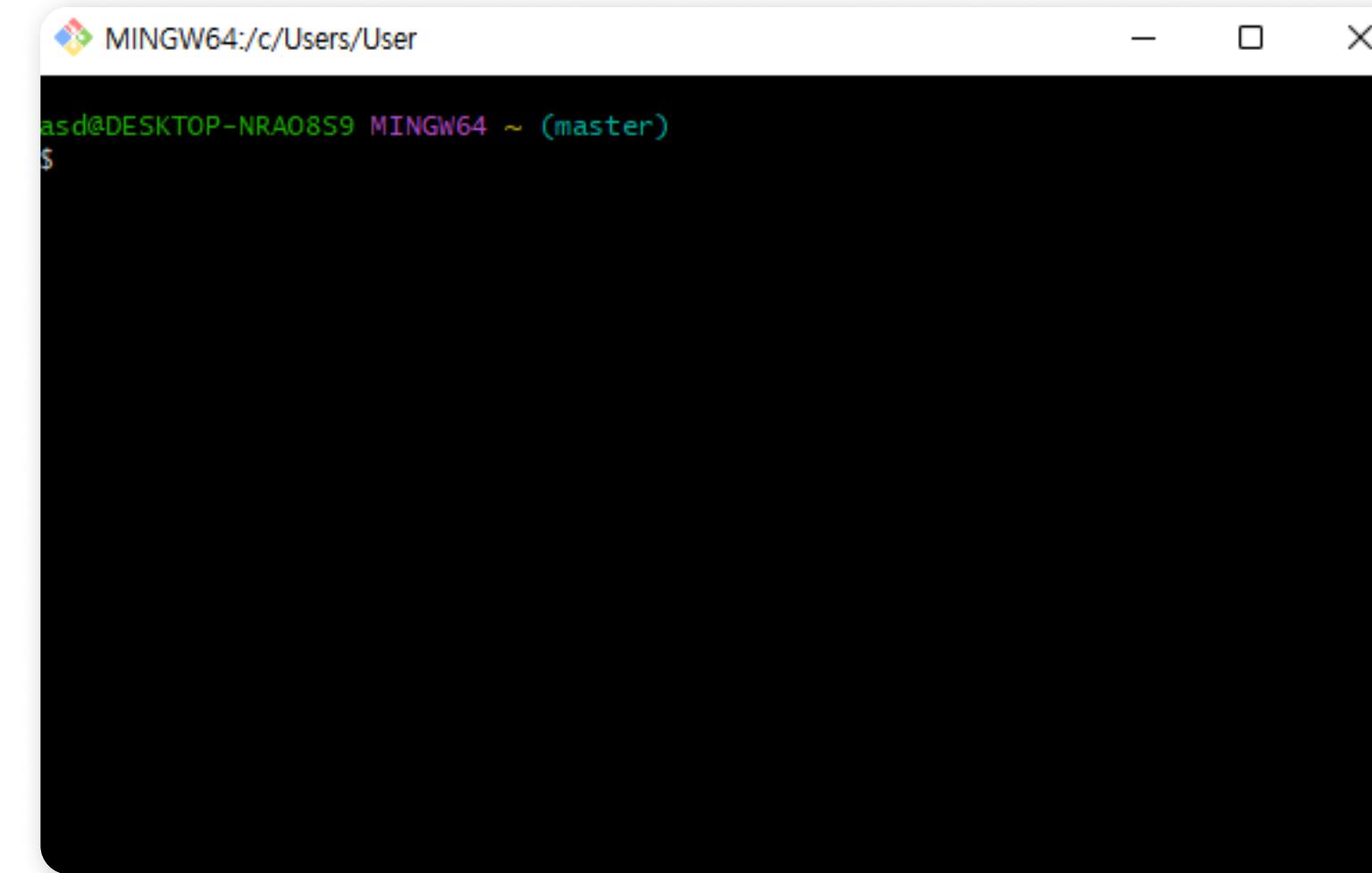
### Step0. 명령어 창 띄우기

UNIX / terminal



```
user@hostname:~  
user@hostname:~$
```

Windows / git bash



```
MINGW64:/c/Users/User  
asd@DESKTOP-NRA08S9 MINGW64 ~ (master)  
$
```

## 03 Git 실습하기

Step1. 작업할 폴더(디렉토리)로 이동하기

**\$ cd (상대 혹은 절대) 경로** cd; change directory

**\$ mkdir 폴더이름** mkdir; make directory

**\$ ls** ls; list

A terminal window titled '~/Desktop/practice' showing a sequence of commands:

- (base) ~ (0.018s)  
cd Desktop/ 원하는 위치로 이동
- base ~/Desktop (0.019s)  
mkdir practice 폴더 생성
- base ~/Desktop (0.016s)  
cd practice 생성한 폴더로 이동
- base ~/Desktop/practice 경로가 바뀐 것을 확인!  
ls
- base ~/Desktop/practice |

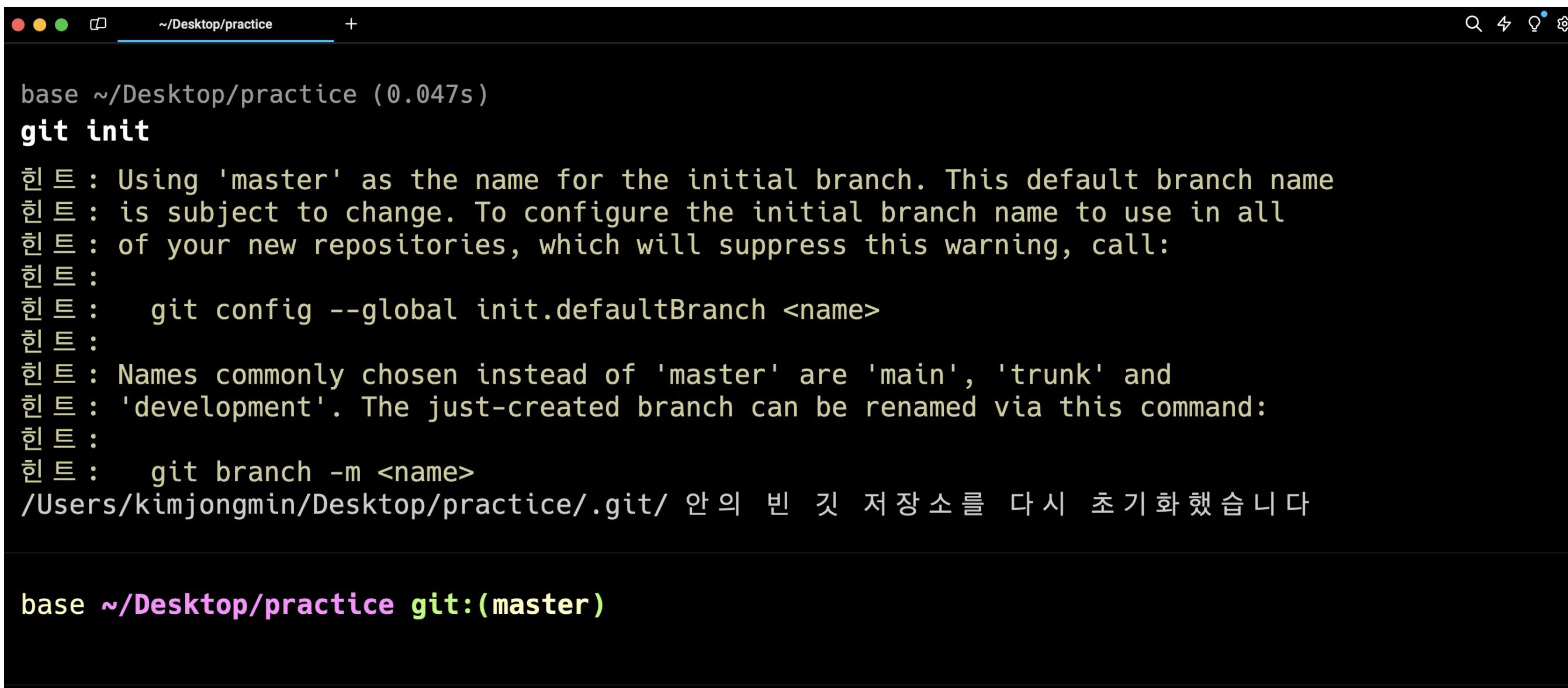
The terminal uses color coding for command output: white text on a black background for standard output, blue boxes for user input, and red boxes for specific annotations.

예시일 뿐이니 원하는 폴더에서 진행해주세요!

## 03 Git 실습하기

### Step2. Git 저장소 생성 및 최초설정

- 작업할 폴더 생성 후 해당 위치에서 `git init`



```
base ~/Desktop/practice (0.047s)
git init
  힌트 : Using 'master' as the name for the initial branch. This default branch name
  힌트 : is subject to change. To configure the initial branch name to use in all
  힌트 : of your new repositories, which will suppress this warning, call:
  힌트 :
  힌트 :   git config --global init.defaultBranch <name>
  힌트 :
  힌트 : Names commonly chosen instead of 'master' are 'main', 'trunk' and
  힌트 : 'development'. The just-created branch can be renamed via this command:
  힌트 :
  힌트 :   git branch -m <name>
/Users/kimjongmin/Desktop/practice/.git/ 안의 빈 깃 저장소를 다시 초기화했습니다

base ~/Desktop/practice git:(master)
```

## 03 Git 실습하기

### Step2. Git 저장소 생성 및 최초설정



Git 사용이 처음인 경우 실시

- 이름과 이메일 주소를 설정하는 이유  
→ commit 할 때 사용자 이름과 이메일이 기록됨

```
base ~/Desktop/practice git:(master) (0.023s)
git config --global user.name "KimJongMin"

base ~/Desktop/practice git:(master) (0.038s)
git config --global user.email "whdals011203@naver.com"

base ~/Desktop/practice git:(master) (0.031s)
git config --global list
error: key does not contain a section: list

base ~/Desktop/practice git:(master) (0.041s)
git config --global --list
user.name=KimJongMin
user.email=whdals011203@naver.com

base ~/Desktop/practice git:(master)
```

제대로 설정이 되었는지 확인

# 03 Git 실습하기

## Step3. 원본 저장소를 내 계정으로 복사하기 - fork

에코노베이션 계정의 레포지토리

JNU-econovation / Let-s-git-it-started

Type / to search

Code Issues Pull requests 3 Zenhub Actions Projects Security Insights Settings

Let-s-git-it-started Public Edit Pins Watch 0 Fork 7 Star 1

main 5 branches 0 tags Go to file Add file Code About

No description, website, or topics provided

baegyeong Merge pull request #54 from baegyeong/main ... d467e2a 1 hour ago 123 commits

.github/ISSUE\_TEMPLATE Update issue templates 4 months ago

1st\_Let\_s\_git\_it\_started(Git... docs: add profile.md 1 hour ago

2nd\_Let\_s\_git\_it\_started(Gi... fix: Let's git it started 디렉토리 구조 최신화 9 hours ago

Glossary\_of\_Terms fix: Let's git it started 디렉토리 구조 최신화 9 hours ago

.gitignore docs: README.md, .gitignore 생성, guide 문서 생성 7 months ago

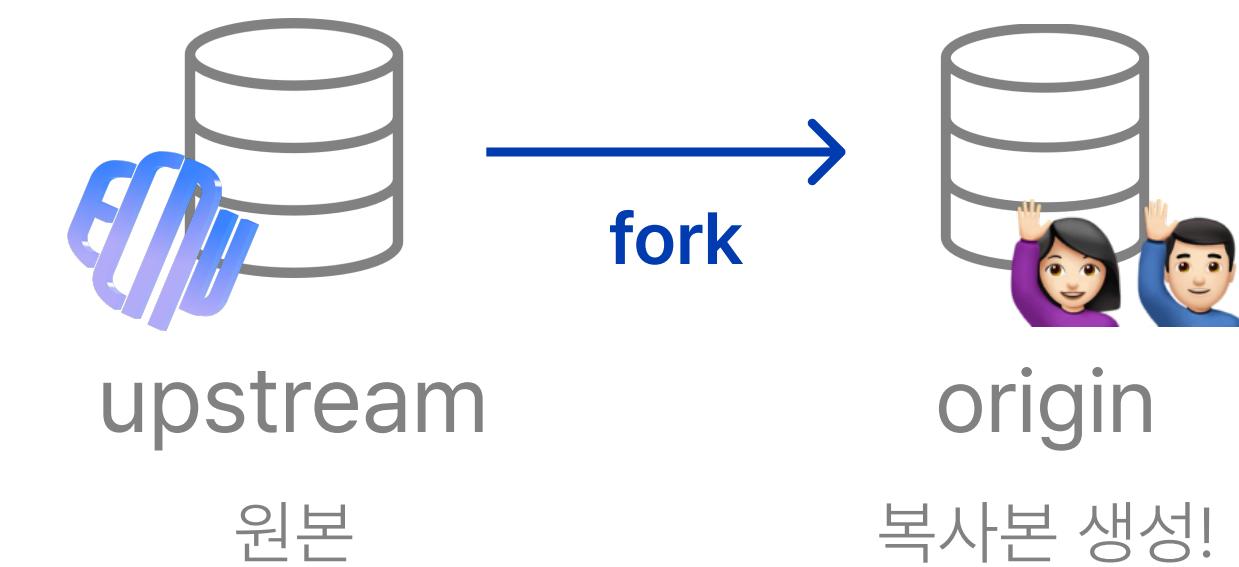
README.md docs: README.md, .gitignore 생성, guide 문서 생성 7 months ago

README.md

Let's qit it started ↗

클릭!

fork는 다른 계정의 원격 저장소를  
내 계정으로 가지고 올 때 사용



# 03 Git 실습하기

## Step3. 원본 저장소를 내 계정으로 복사하기 - fork

**Create a new fork**

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (\*).

Owner *	Repository name *
baegyeong	/ Let-s-git-it-started <small>Let-s-git-it-started is available.</small>

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Copy the main branch only  
Contribute back to JNU-econovation/Let-s-git-it-started by adding your own branch. [Learn more.](#)

(i) You are creating a fork in your personal account.

**클릭!**

**Create fork**

내 계정으로 복사된 것을 확인

baegyeong / Let-s-git-it-started

Let-s-git-it-started (Public)

forked from [JNU-econovation/Let-s-git-it-started](#)

main 1 branch 0 tags

Go to file Add file ▾ Code ▾

This branch is up to date with JNU-econovation/Let-s-git-it-started:main. Contribute Sync fork

baegyeong Merge pull request JNU-econovation#5... d467e2a 2 hours ago 123 commits

.github/ISSUE\_TEMPLATE Update issue templates 4 months ago

1st\_Let\_s\_git\_it\_started(G... docs: add profile.md 3 hours ago

2nd\_Let\_s\_git\_it\_started(... fix: Let's git it started 디렉토리 구조 최신화 10 hours ago

Glossary\_of\_Terms fix: Let's git it started 디렉토리 구조 최신화 10 hours ago

.gitignore docs: README.md, .gitignore 생성, guide 문서 생성 7 months ago

README.md docs: README.md, .gitignore 생성, guide 문서 생성 7 months ago

README.md

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

8 forks

No releases published

Create a new release

Packages

No packages published

Publish your first package

## 03 Git 실습하기

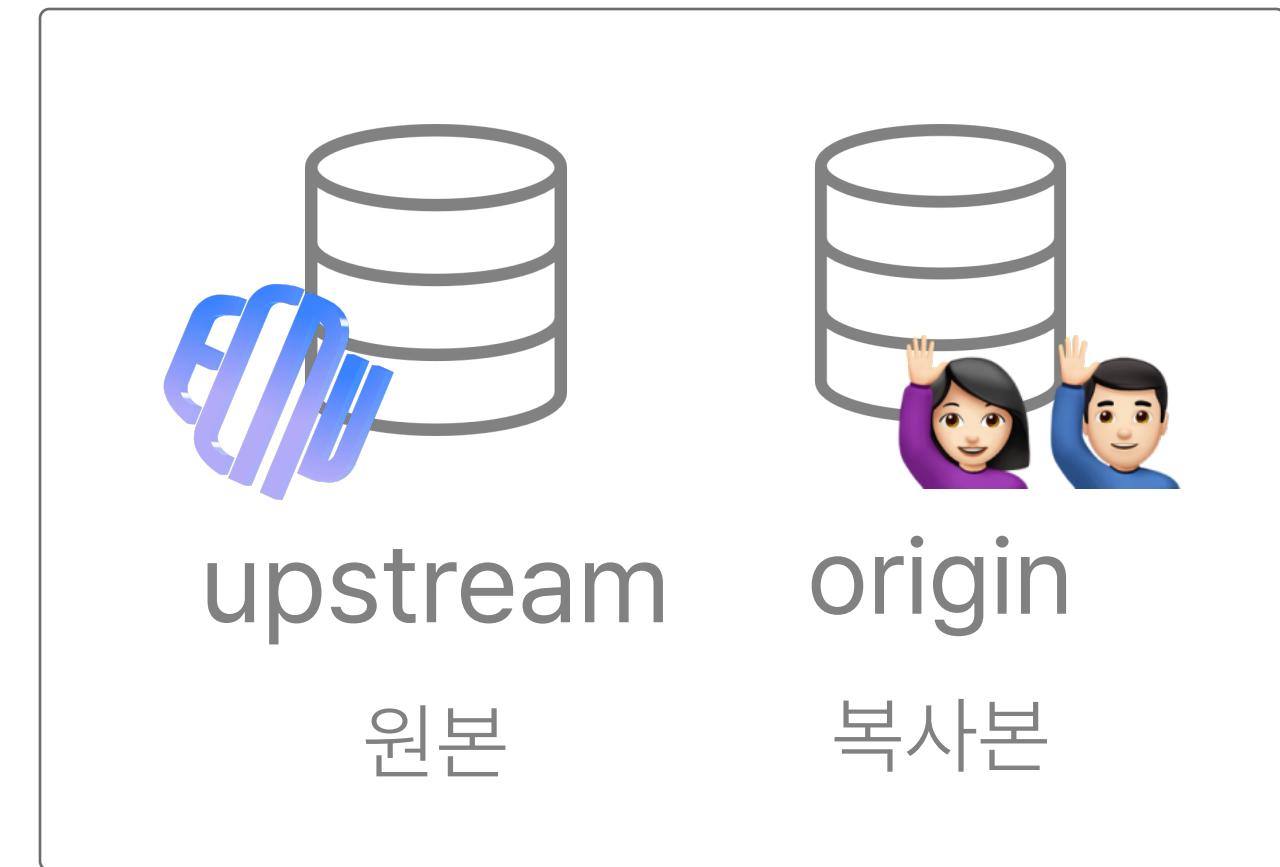
Step4. 자신의 local에 원격 저장소 등록하기

remote  
(github)

**\$ git remote add [저장소 이름] [저장소 주소]**

💡 실습을 위해 두 가지 저장소를 등록합니다.

1. 에코노베이션 레포지토리(원본)
2. 나의 레포지토리 (fork한 복사본)



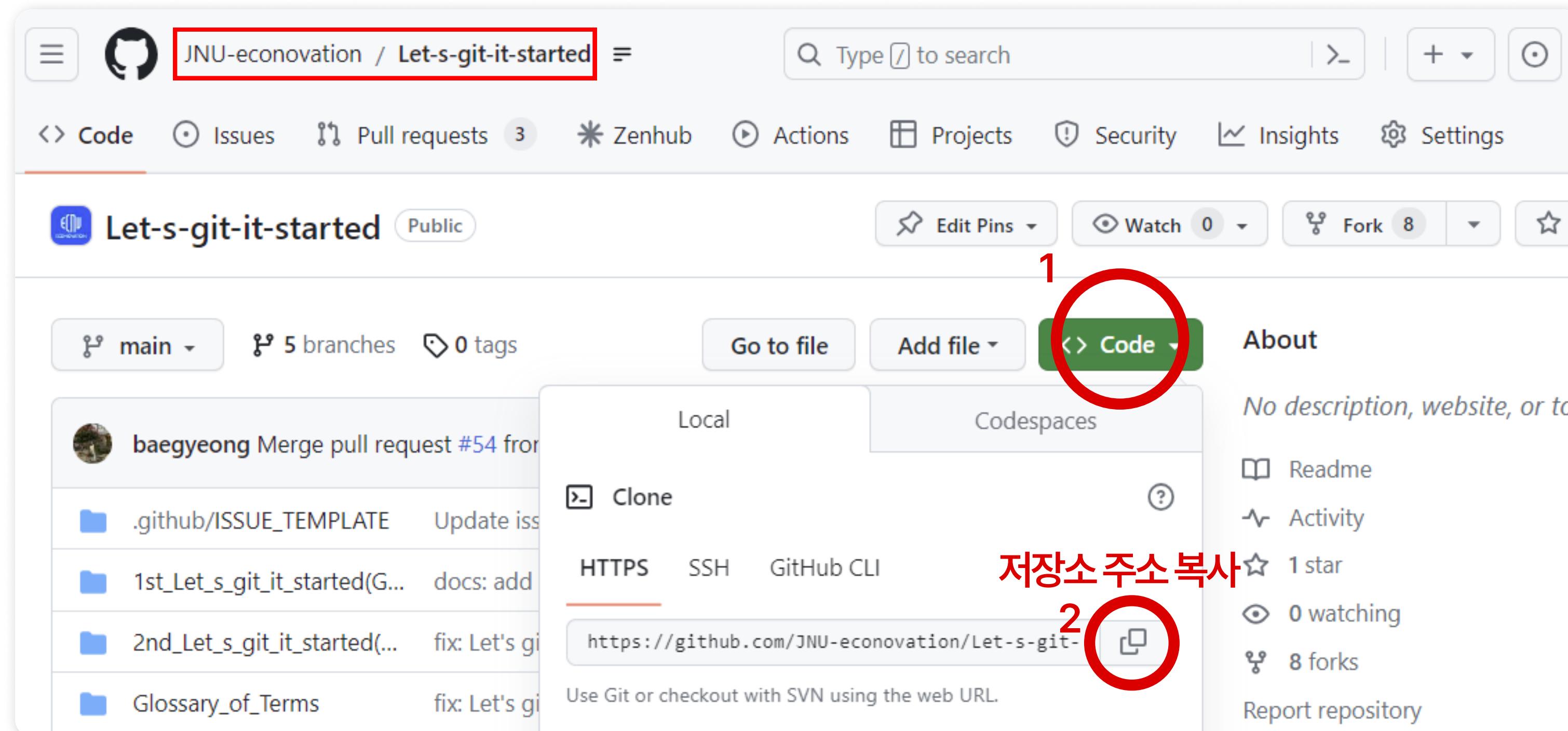
remote, pull

local  
(내 pc)

## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

### Step 4-1. 원본 레포지토리(저장소) 등록하기 - upstream



## 03 Git 실습하기

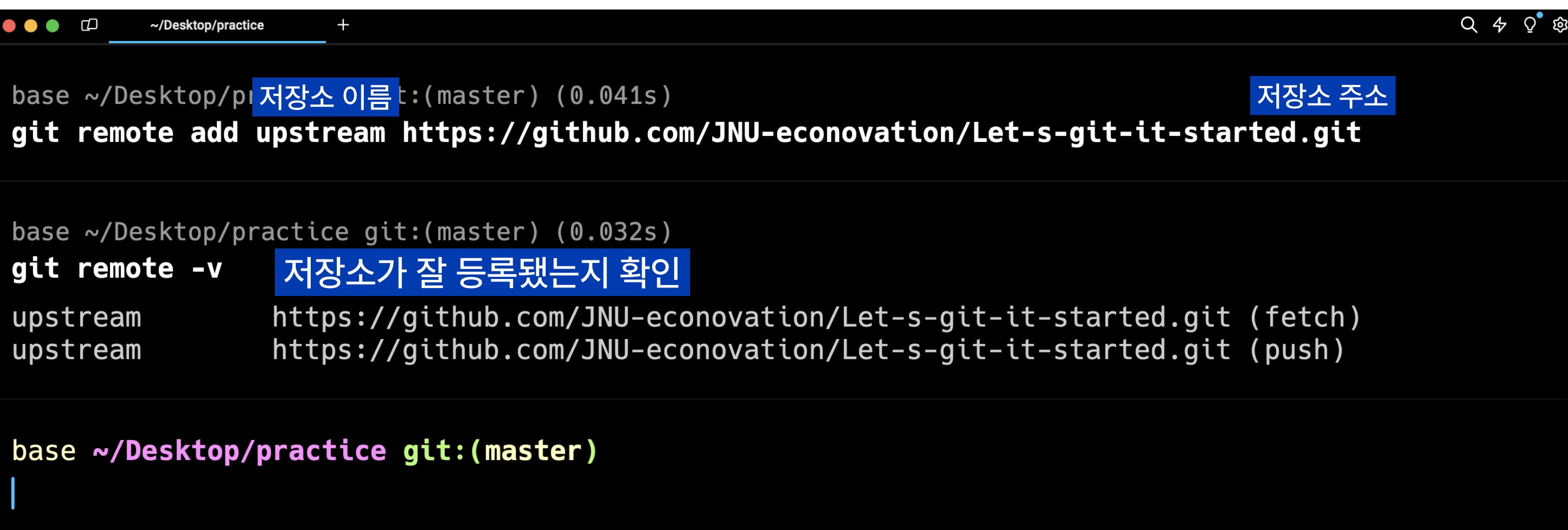
Step4. 자신의 local에 원격 저장소(레포지토리) 등록하기

Step 4-1. 원본 레포지토리(저장소) 등록하기 - **upstream**

**\$ git remote add [저장소 이름] [저장소 주소]**

**\$ git remote -v**

현재 로컬 저장소와 연결된 원격 저장소들의 정보 확인



The screenshot shows a terminal window with the following session:

```
base ~/Desktop/practice git:(master) (0.041s)
git remote add upstream https://github.com/JNU-econovation/Let-s-git-it-started.git

base ~/Desktop/practice git:(master) (0.032s)
git remote -v      저장소가 잘 등록됐는지 확인
upstream           https://github.com/JNU-econovation/Let-s-git-it-started.git (fetch)
upstream           https://github.com/JNU-econovation/Let-s-git-it-started.git (push)

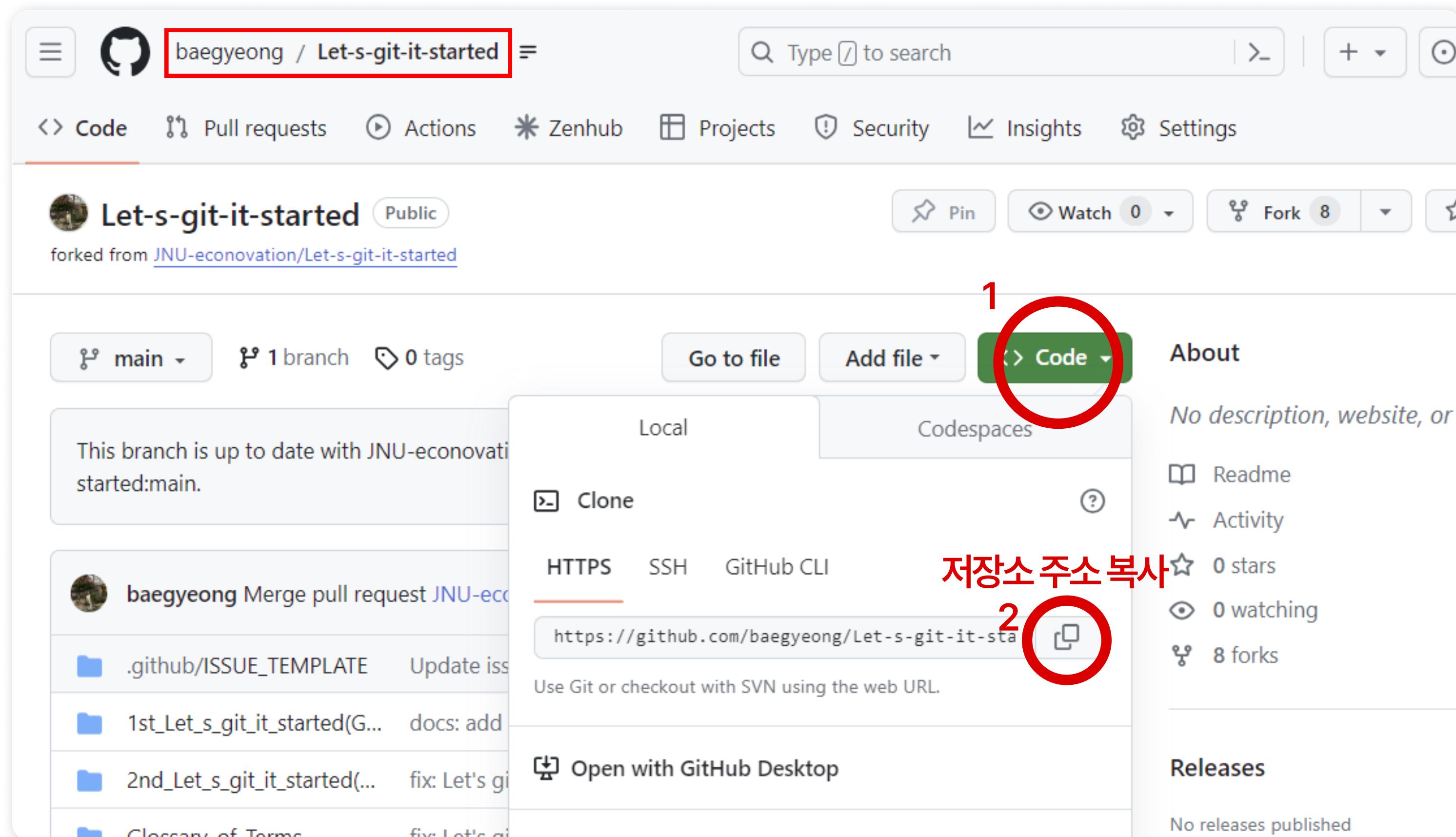
base ~/Desktop/practice git:(master)
```

The terminal window has a dark background and light-colored text. The command `git remote add upstream https://github.com/JNU-econovation/Let-s-git-it-started.git` is run first, followed by `git remote -v`. The output shows the upstream repository has been added and is correctly configured for both fetch and push operations. The prompt at the end indicates the session is still active.

## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

### Step 4-2. fork한 내 레포지토리(저장소) 등록하기 - origin



## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

Step 4-2. fork한 내 레포지토리(저장소) 등록하기 - **origin**

**\$ git remote add [저장소 이름] [저장소 주소]**

**\$ git remote -v**

현재 로컬 저장소와 연결된 원격 저장소들의 정보 확인



The screenshot shows a terminal window with the following session:

```
base ~/Desktop/practice git:(master) (0.043s)      저장소 주소
git remote add origin https://github.com/rlajm1203/Let-s-git-it-started.git

base ~/Desktop/practice git:(master) (0.032s)
git remote -v  저장소가 잘 등록됐는지 확인
origin  https://github.com/rlajm1203/Let-s-git-it-started.git (fetch)
origin  https://github.com/rlajm1203/Let-s-git-it-started.git (push)
upstream    https://github.com/JNU-econovation/Let-s-git-it-started.git (fetch)
upstream    https://github.com/JNU-econovation/Let-s-git-it-started.git (push)

upstream(원본 레포지토리)와 origin(fork한 내 레포지토리)가 잘 등록된 것을 확인

base ~/Desktop/practice git:(master)
```

The terminal window has a dark theme. The command `git remote add` is highlighted in yellow, and the output of `git remote -v` is also highlighted in yellow. The text "저장소 이름" and "저장소 주소" are overlaid on the terminal window to explain the command fields.

## 03 Git 실습하기

Step5. 자신의 local에 원격 저장소 내역을 가져오기

**\$ git pull [저장소 이름] [가져올 브랜치]**



원본 저장소(upstream)의 현재 학기 브랜치(ex. 2024-1)를 가져와야 합니다.

```
base ~/Desktop/practice git:(master) (3.375s)
git pull upstream 2024-1 원격 저장소에 파일 내역 가져오기
remote: Enumerating objects: 661, done.
remote: Counting objects: 100% (192/192), done.
remote: Compressing objects: 100% (104/104), done.
remote: Total 661 (delta 100), reused 111 (delta 86), pack-reused 469
오브젝트를 받는 중: 100% (661/661), 21.02 MiB | 9.55 MiB/s, 완료.
델타를 알아내는 중: 100% (327/327), 완료.
https://github.com/JNU-econovation/Let-s-git-it-started URL에서
 * branch            2024-1      -> FETCH_HEAD
 * [새로운 브랜치]  2024-1      -> upstream/2024-1
현재 폴더의 파일 목록 확인하기(pull을 통해 가져온 내역 확인)
base ~/Desktop/practice git:(master)
|
```

## 03 Git 실습하기

### Step6. 준비해 온 자기소개 글 수정하기

**\$ cp [복사 할 파일] [복사 될 파일]**

💡 자기소개 템플릿 파일인 profile.md를 복제합니다.  
profile.md 파일은 수정하지 마세요!

The screenshot shows a terminal window with the following session:

```
base ~/Desktop/practice git:(master) (0.021s) cd 1st_Let_s_git_it_started\Git_tutorial\ 1st_Let_s_git_it_started 폴더로 이동
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial) git:(master) (0.024s) ls
1차_Let_s_git_it_started_발표자료.pdf 2024-1 README.md exercise.md guide.md profile.md quick_guide.md

base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial) cp profile.md 2024-1/KimJongMin_profile.md profile.md 파일을 [디렉터리]/[이름].md 파일로 복사
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial) git:(master)±1 (0.023s) cd 2024-1/ 2024-1 폴더로 이동
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)±1 (0.024s) ls
KimJongMin_profile.md README.md [이름].md 파일이 생성된 것을 확인
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)±1
```

Annotations highlight specific steps:

- "폴더에 어떤 파일이 있는지 확인" (Check what files are in the folder) points to the output of the `ls` command.
- "profile.md 파일을 [디렉터리]/[이름].md 파일로 복사" (Copy profile.md file to [directory]/[name].md file) points to the `cp profile.md 2024-1/KimJongMin_profile.md` command.
- "2024-1 폴더로 이동" (Move to 2024-1 folder) points to the `cd 2024-1/` command.
- "[이름].md 파일이 생성된 것을 확인" (Check if [name].md file was created) points to the final output showing the new file `[이름].md`.

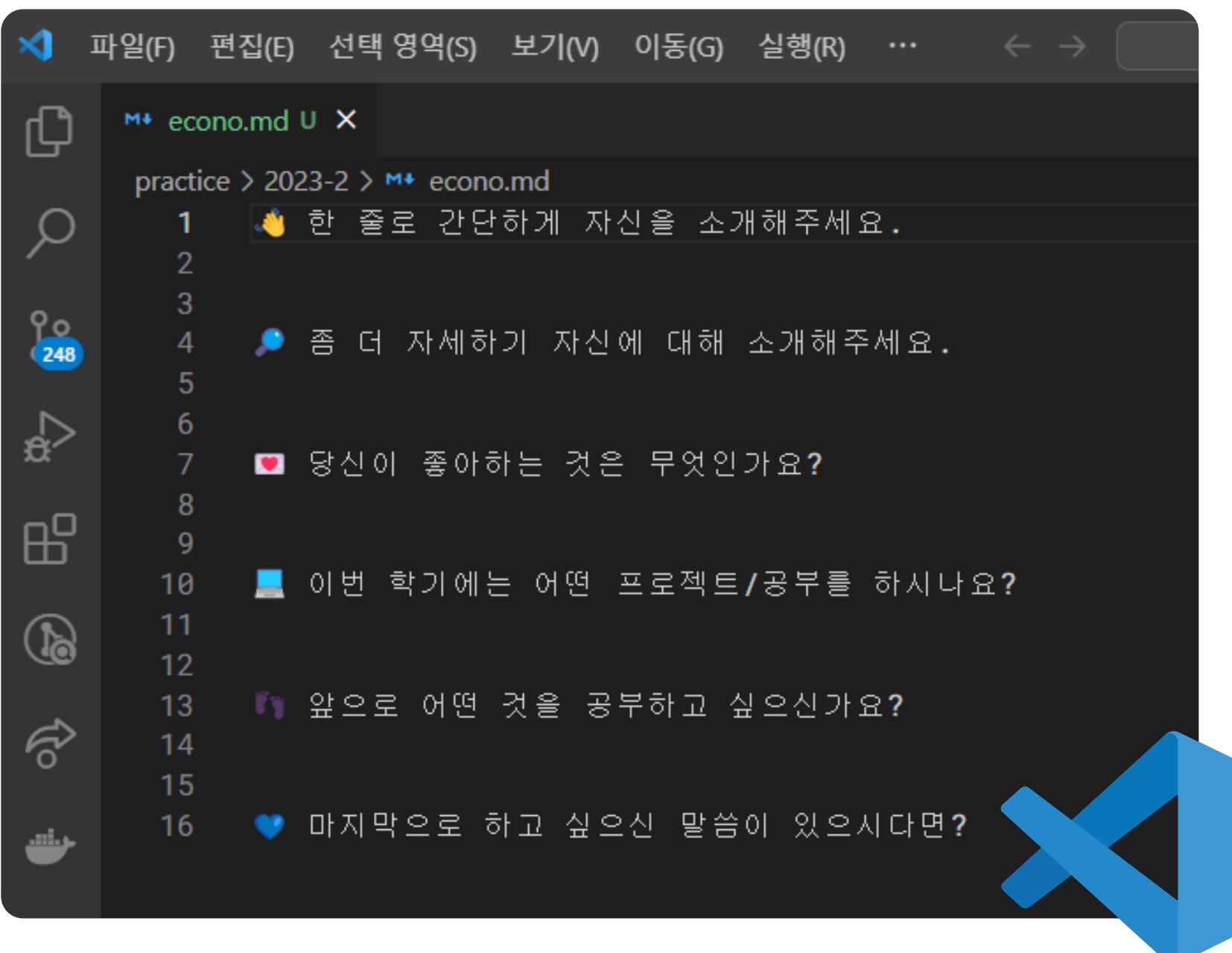
## 03 Git 실습하기

Step5. 준비해 온 자기소개로 글 수정하기  [Markdown 문법 정리](#)

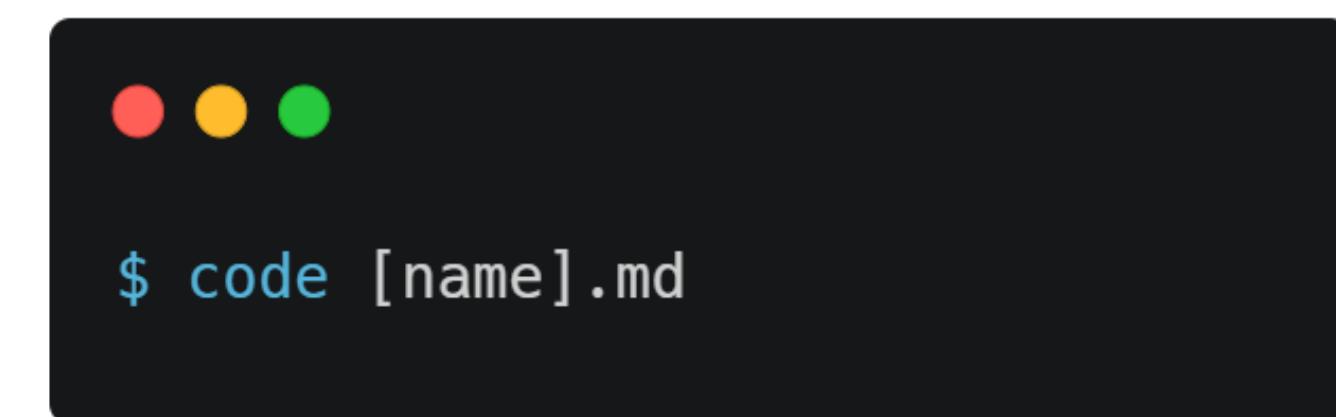


### 글 작성 에디터를 실행하는 다양한 방법

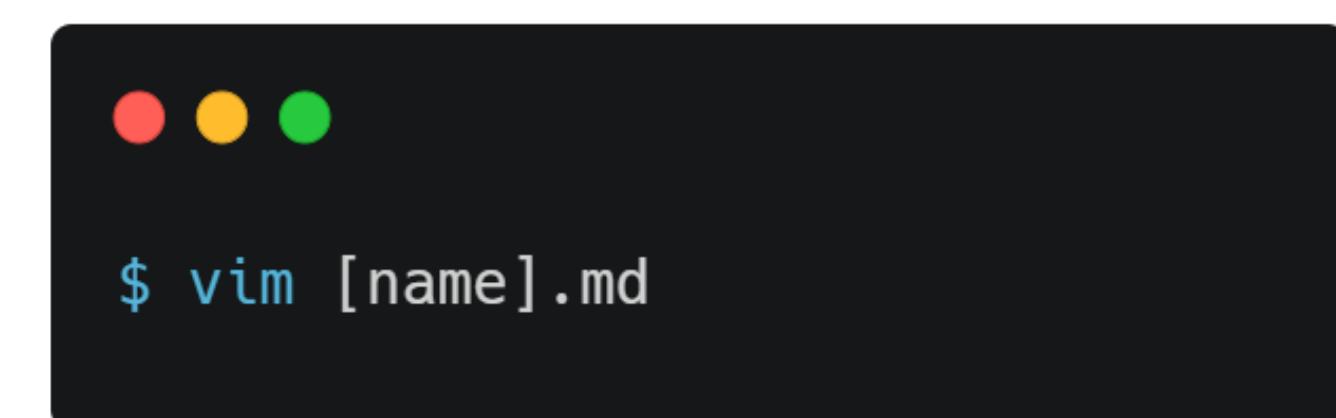
#### 1. Visual Studio Code 직접 실행하기



#### 2. 명령어로 Visual Studio Code 실행시키기



#### 3. 터미널에서 작성하기 - Vim

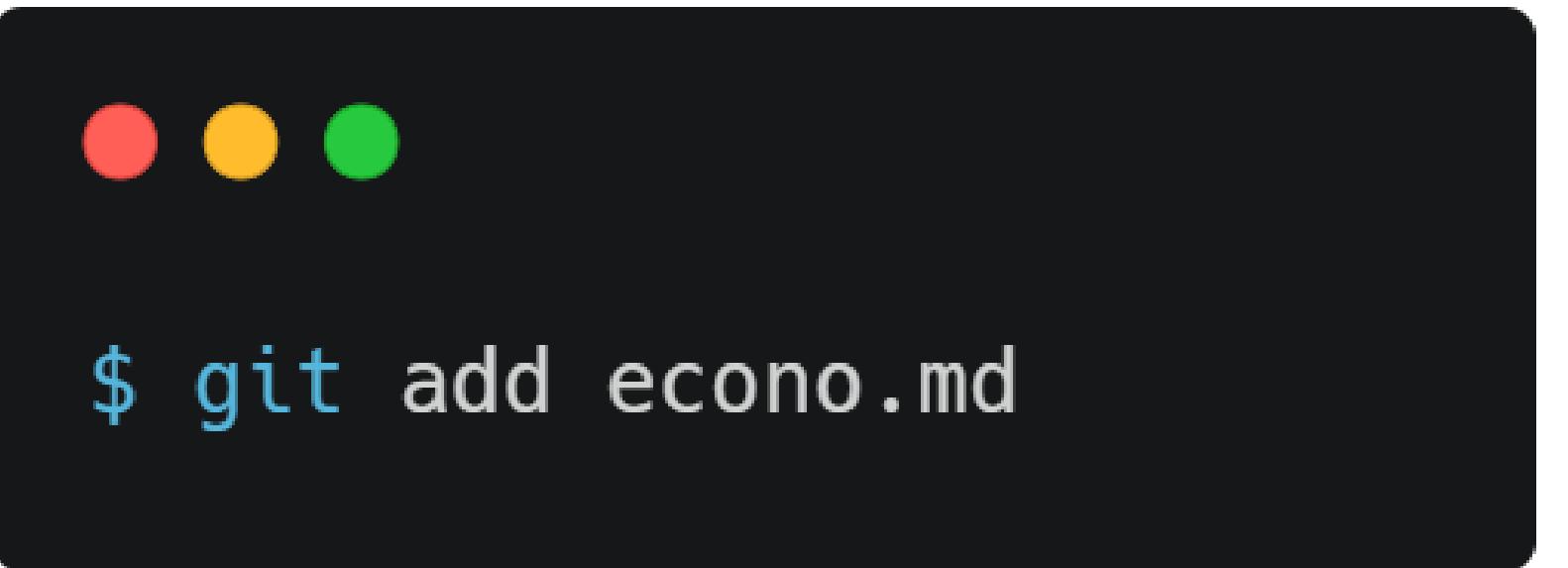


## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**\$ git add filename**

자신이 clone한 local 저장소의 변경사항을 GitHub에 반영하겠다.

A dark gray rectangular box representing a terminal window. In the top-left corner, there are three small colored circles: red, yellow, and green. Below them, the command '\$ git add econo.md' is written in white text.

```
$ git add econo.md
```

- 💡 파일 add 전 후로, '[git status](#)' 명령어를 통해  
현재 작업 중인 저장소의 상태를 확인하는 습관 들이기!

## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

### \$ git add filename

자신이 clone한 local 저장소의 변경사항을 GitHub에 반영하겠다.

The terminal window shows the following sequence of commands:

- Initial state: `base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)±1 (0.027s)`
- `ls` command: **현재 폴더의 파일 목록 확인** (Confirm current folder file list)
- `git add KimJongMin_profile.md` command: **스테이징 영역에 파일 추가하기** (Add file to staging area)
- `git status` command: **저장소 상태 확인** (Check repository status)
- Final state: `base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)±1 (0.025s)`

In the final status output, it shows a new file `KimJongMin_profile.md` has been added.

## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**\$ git commit -m "message 내용"**

변경사항에 대해서 "이러한 내용"이 담긴 메시지를 남기겠다.

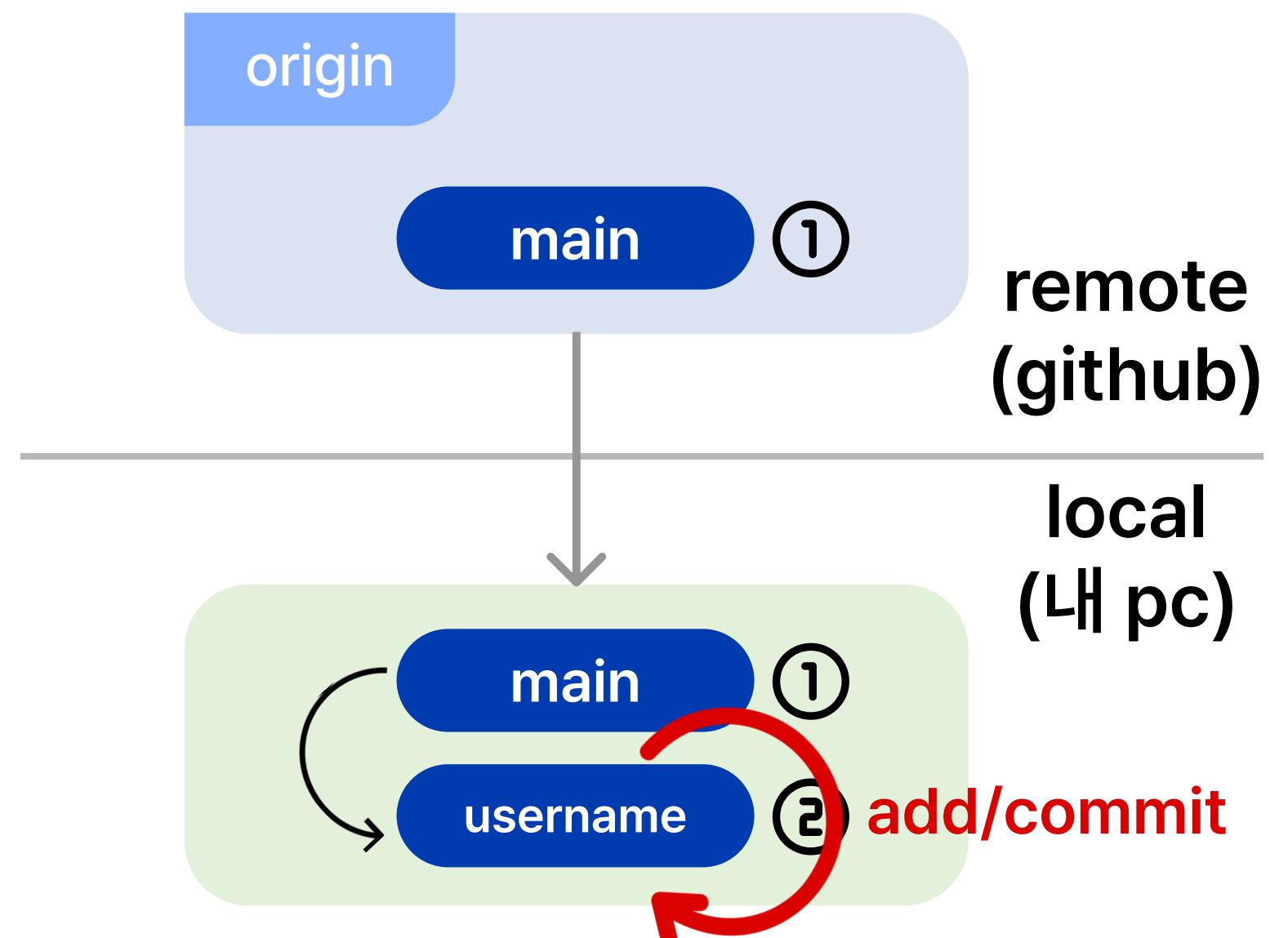
💡 message 내용은 아래 형식을 지켜주세요.



```
$ git commit -m "docs: add [이름].md"
```

원활한 협업을 위해서 commit message를 어떻게 쓰느냐가 정말 중요해요!  
commit 메시지 잘 쓰는 방법이 궁금하시다면 아래의 링크를 참고해주세요.

🔗 커밋 컨벤션



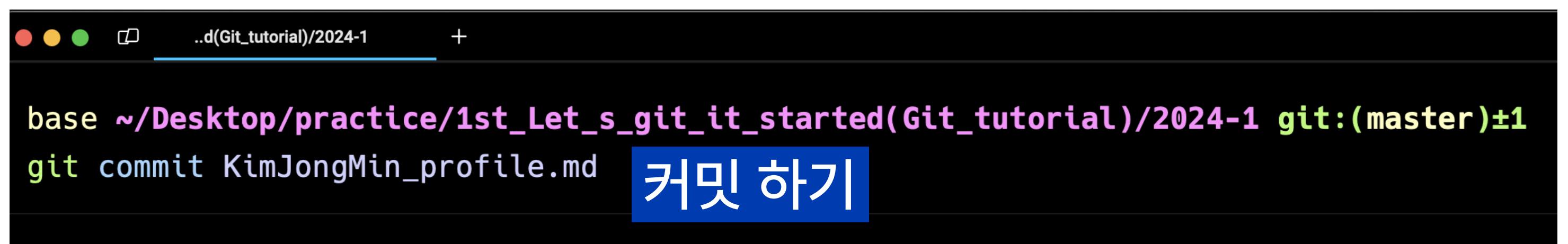
## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**\$ git commit [filename]**

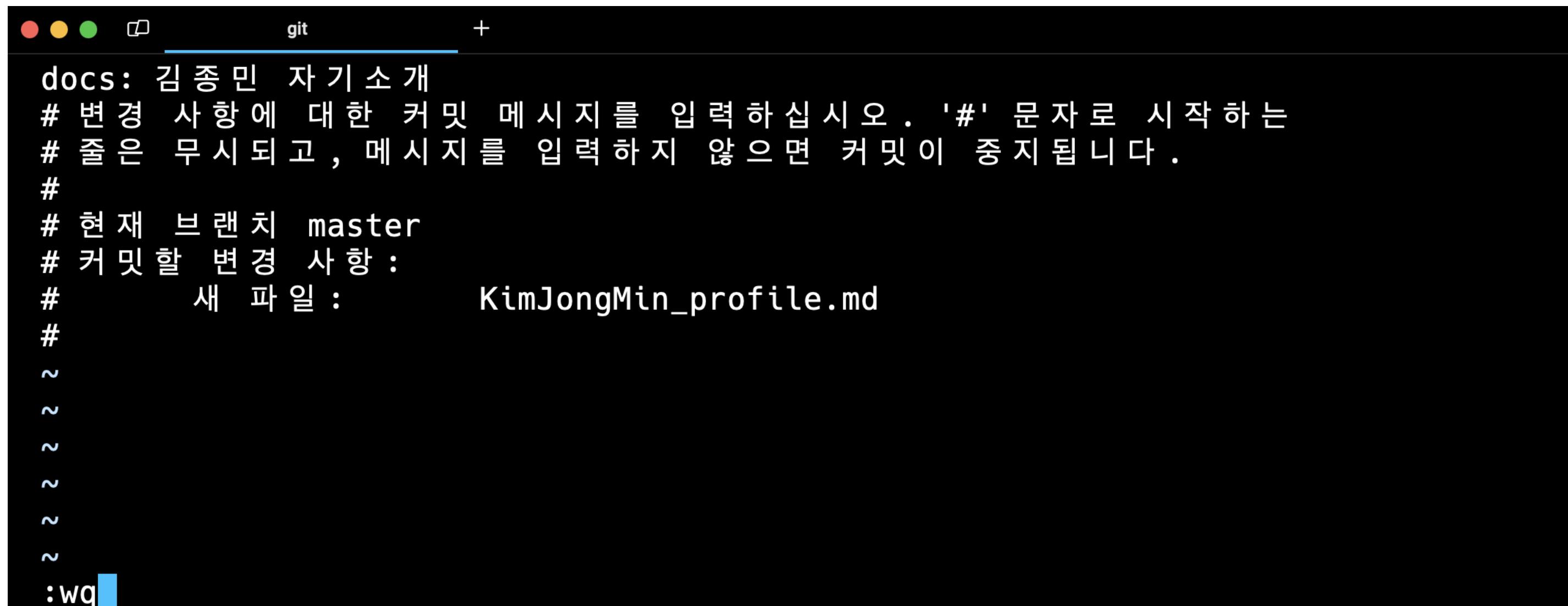
변경사항에 대해서 “이러한 내용”이 담긴 메시지를 남기겠다.

i	입력 모드
메세지 입력	메세지 입력
esc	편집 모드
:wq	저장하고 종료



```
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)±1
git commit KimJongMin_profile.md
```

커밋 하기



```
docs: 김 종 민 자 기 소 개
# 변경 사 항에 대 한 커밋 메 시 지를 입 력 하 십 시 오 . '#' 문 자로 시 작 하 는
# 줄 은 무 시 되 고 , 메 시 지를 입 력 하 지 않 으 면 커밋 이 중 지 됩 니 다 .
#
# 현 재 브 랜 치 master
# 커밋 할 변 경 사 항 :
# 새 파 일 : KimJongMin_profile.md
#
~
```

:wq

스테이징 영역 -> 로컬 저장소 이동  
스테이징 영역은 비워지게 됨

## 03 Git 실습하기

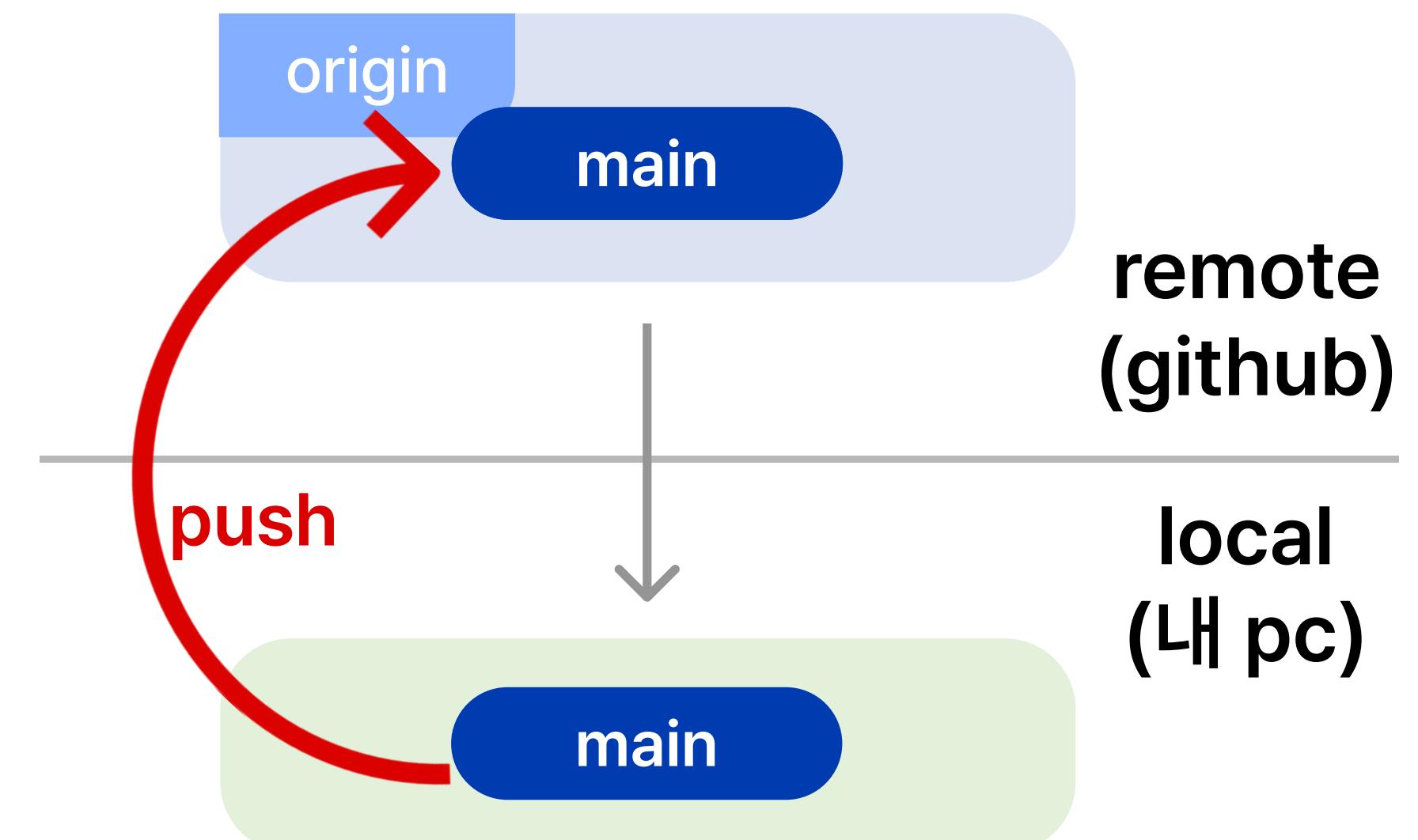
Step6. 그럼 이제! 글을 올려보자

**\$ git push [저장소 이름] [브랜치]**

원격 저장소(remote)의 주소

현재 자신이 위치한 branch 이름

```
● ● ●  
$ git push origin main
```



## 03 Git 실습하기

### Step6. 그럼 이제! 글을 올려보자

**\$ git push [저장소 이름] [브랜치]**

 현재 브랜치가 main라면?

깃허브에서는 'master'로 브랜치가 설정되어 있기 때문에,  
만약 터미널에 반영이 안 됐다면 바꿔줘야 합니다.

#### 1. 현재 로컬 브랜치는 main

```
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(main) (0.024s)
git branch -m master 2. master으로 이름 변경
```

```
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)
```

3. 브랜치명이 변경된 것을 확인!

```
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master) (1.82s)
git push origin master
오브젝트 나열하는 중 : 8, 완료 .
오브젝트 개수 세는 중 : 100% (8/8), 완료 .
Delta compression using up to 8 threads
오브젝트 압축하는 중 : 100% (5/5), 완료 .
오브젝트 쓰는 중 : 100% (5/5), 706 bytes | 706.00 KiB/s, 완료 .
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/rlajm1203/Let-s-git-it-started.git
  2855100..9b4c353 master -> master
```

```
base ~/Desktop/practice/1st_Let_s_git_it_started(Git_tutorial)/2024-1 git:(master)
```

# Chapter3. Feedback/Review

01 Pull Request 작성하기

02 다른 사람 소개 글 읽고 리뷰하기

# 01 Pull Request 작성하기

The screenshot shows the GitHub interface for the repository `baegyeong / Let-s-git-it-started`. The top navigation bar includes links for Code, pull request, Actions, Zenhub, Projects, Security, Insights, and more. A red circle highlights the 'pull request' tab. Below it, a search bar filters for open pull requests. A green button labeled 'New pull request' is circled in red. The main content area displays a 'Welcome to pull requests!' message and instructions for creating a pull request.

1. pull request 클릭

2. New pull request 클릭

The screenshot shows the GitHub interface for the repository `JNU-econovation / Let-s-git-it-started`. It displays a comparison between branches. A red circle highlights the dropdown menu for the base branch, which is currently set to 'main'. Below the dropdown, a message states 'Able to merge. These branches can be automatically merged.' A green button labeled 'Create pull request' is circled in red.

Comparing changes

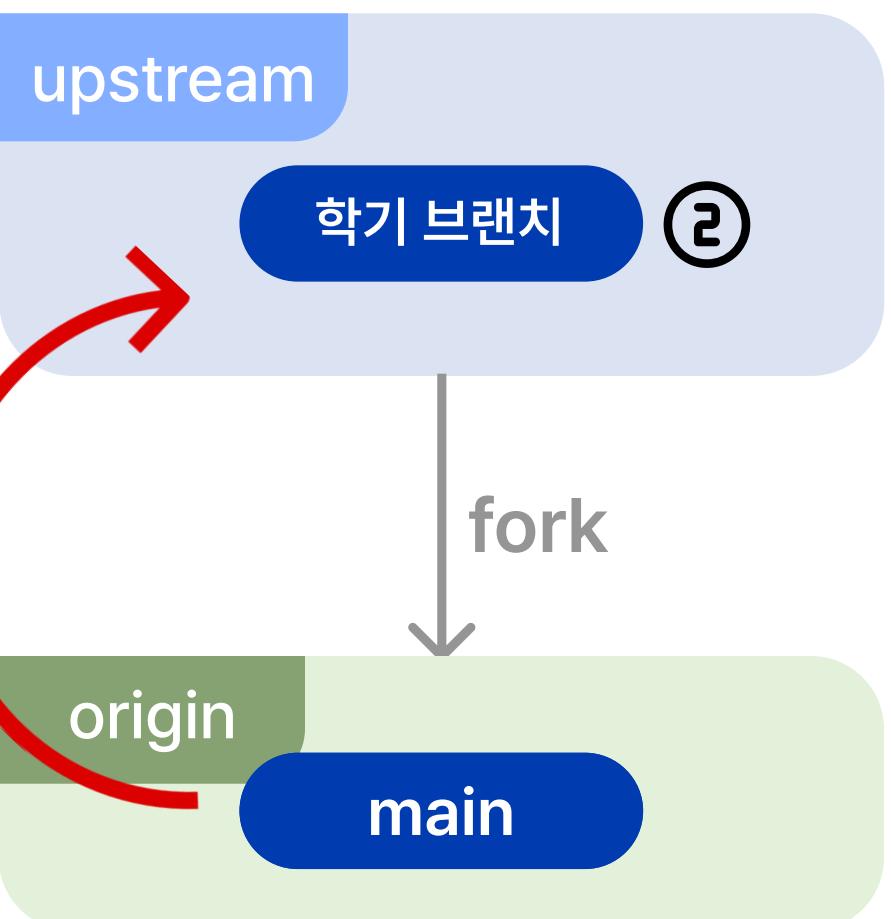
base repository: JNU-econovation/Let-s-git-it-st... base: 2024-1 ... head repository: baegyeong/Let-s-git-it-started compare: main

Able to merge. These branches can be automatically merged.

Create pull request

3. 해당 학기 브랜치로 변경

4. 클릭



# 01 Pull Request 작성하기

The screenshot shows the GitHub 'Open a pull request' interface for the repository 'JNU-econovation / Let-s-git-it-started'. The interface is divided into three main sections:

- 1. pr을 보내는 브랜치 확인**  
내 레포 'master' → 원본 레포 '해당 학기 브랜치'
- 2. 제목과 내용 작성**  
제목: 이름(아이디) - 자기소개  
내용: 자유롭게 구성
- 3. 라벨 지정**  
(기수, mbti, 자기소개)

**1. pr을 보내는 브랜치 확인**: The top navigation bar shows the base repository as 'JNU-econovation/Let-s-git-it-st...' and the head repository as 'baegyeong/Let-s-git-it-started'. The 'base' dropdown is set to '2024-1-' and the 'compare' dropdown is set to 'main'. A red box highlights these fields.

**2. 제목과 내용 작성**: The title field contains '에코노(econo) - 자기소개' and the body text area contains '안녕하세요 에코노입니다! 잘 부탁드려요~'. A red box highlights the title and body input areas.

**3. 라벨 지정**: The 'Labels' section shows two assigned labels: '25기' and '자기소개'. A red box highlights this section.

## 02 다른 사람의 소개글 읽고 리뷰하기

The screenshot shows a GitHub repository page for 'JNU-econovation / 2022-1-I-AM-GROUND'. The 'Pull requests' tab is selected, displaying 9 open pull requests. A search bar at the top filters results for 'is:pr is:open'. The pull requests are listed in descending order of creation date:

- #11: 21기 이윤성 자기소개입니다. (ENFP) [자기소개]. Opened on May 10, 2022 by 2yunseong.
- #10: 23기 나인혜 자기소개입니다. Opened on Apr 29, 2022 by nainhye • Approved.
- #9: 23기 조배경 자기소개입니다. (ISFJ) [자기소개]. Opened on Apr 29, 2022 by baegyeong • Approved.
- #8: 23기 김지민 자기소개입니다. (ENFP) [자기소개]. Opened on Apr 29, 2022 by jminkkk • Approved.
- #7: 최명선(myungsun) 자기소개 (ESFJ) [자기소개]. Opened on Apr 29, 2022 by myungsun7782 • Approved.
- #5: 23기 이도연 자기소개입니다. (enfi) [자기소개]. Opened on Apr 29, 2022 by capDoYeonLee • Approved.
- #4: 23기 이규민 자기소개입니다. [자기소개]. Opened on Apr 29, 2022 by Klomachenko • Approved.

## 02 다른 사람의 소개글 읽고 리뷰하기

이진혁(LJH098) - 자기소개 #115

**Open** LJH098 wants to merge 1 commit into JNU-econovation:2024-1 from LJH098:2024-1

Conversation 0 Commits 1 Checks 0 Files changed 1

LJH098 commented 15 minutes ago  
잘부탁해요~~

docs: add LeeJinHyuk.md

LJH098 added 25기 자기소개 ENTP labels 15 minutes ago

LJH098 self-assigned this 15 minutes ago

a919f83

**클릭!**

Reviewers: No reviews—at least 6 approving reviews are required.  
Still in progress? Convert to draft

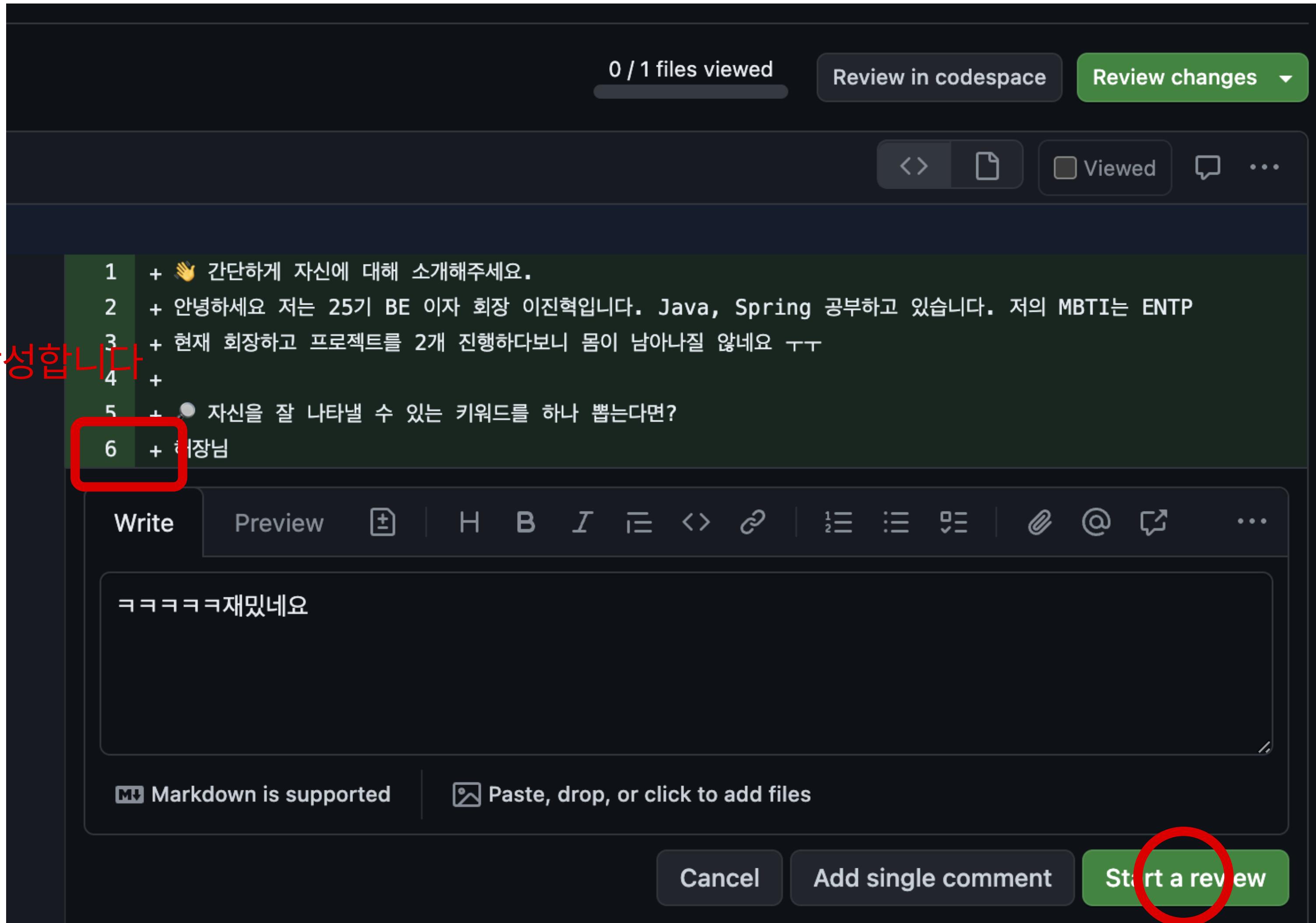
Assignees: LJH098

Labels: 25기 ENTP 자기소개

Projects:

The screenshot shows a GitHub pull request page for a user named LJH098. The title is "이진혁(LJH098) - 자기소개 #115". The status is "Open" and it's merging "LJH098:2024-1" into "JNU-econovation:2024-1". Below the status, there are four buttons: "Conversation 0", "Commits 1", "Checks 0", and "Files changed 1". The "Files changed" button is highlighted with a red box. To the right of the buttons, there is a large red text "클릭!" (Click!). The main content area shows a comment from LJH098: "잘부탁해요~~". Below the comment, there are three log entries: "docs: add LeeJinHyuk.md", "LJH098 added 25기 자기소개 ENTP labels 15 minutes ago", and "LJH098 self-assigned this 15 minutes ago". On the right side, there are sections for "Reviewers" (no reviews required), "Assignees" (LJH098), and "Labels" (25기, ENTP, 자기소개). At the bottom right, there is a "Projects" section.

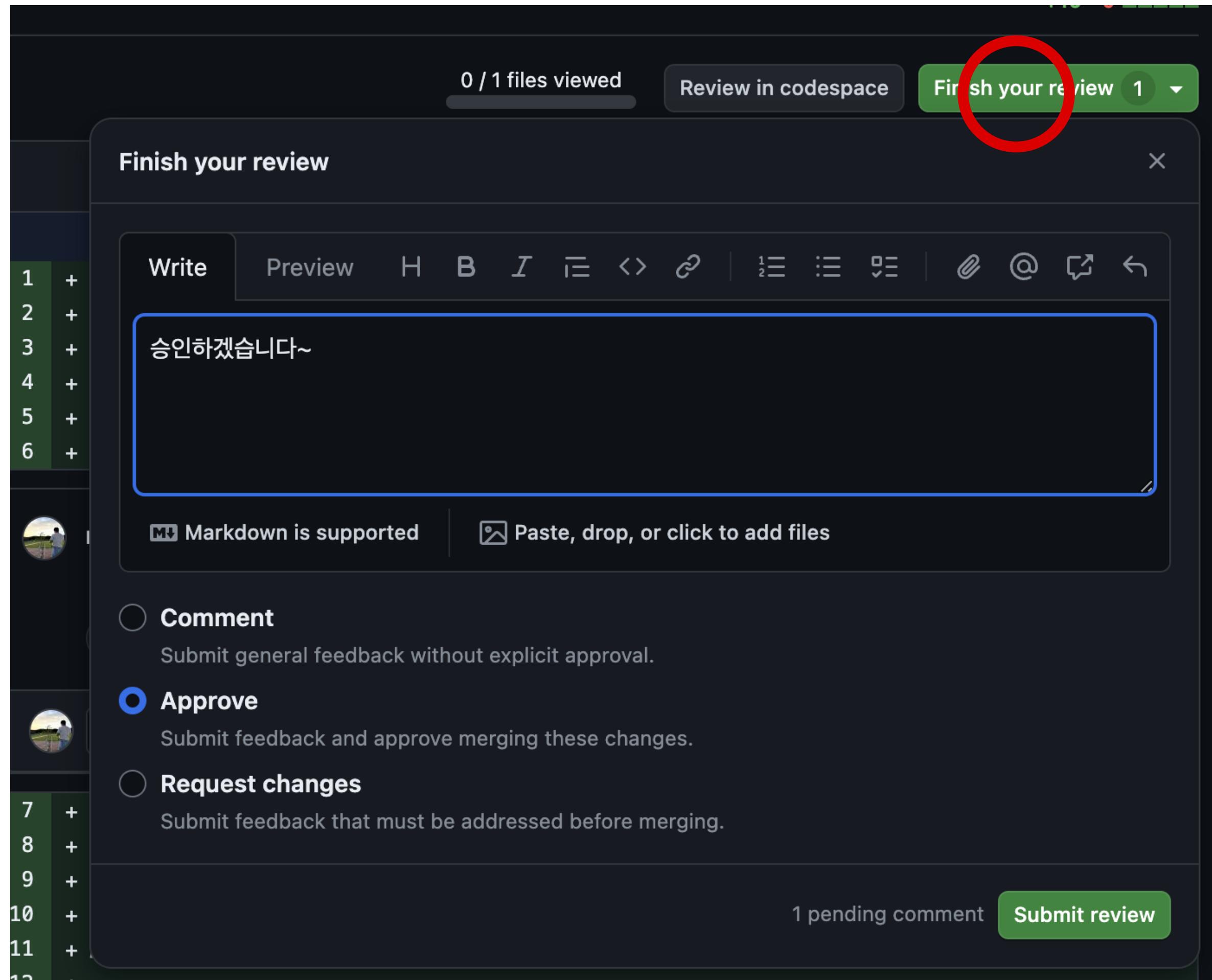
## 02 다른 사람의 소개글 읽고 리뷰하기



2. 리뷰를 작성하셨다면 클릭!

## 02 다른 사람의 소개글 읽고 리뷰하기

모든 리뷰를 완료했다면 클릭!



## 02 다른 사람의 소개글 읽고 리뷰하기

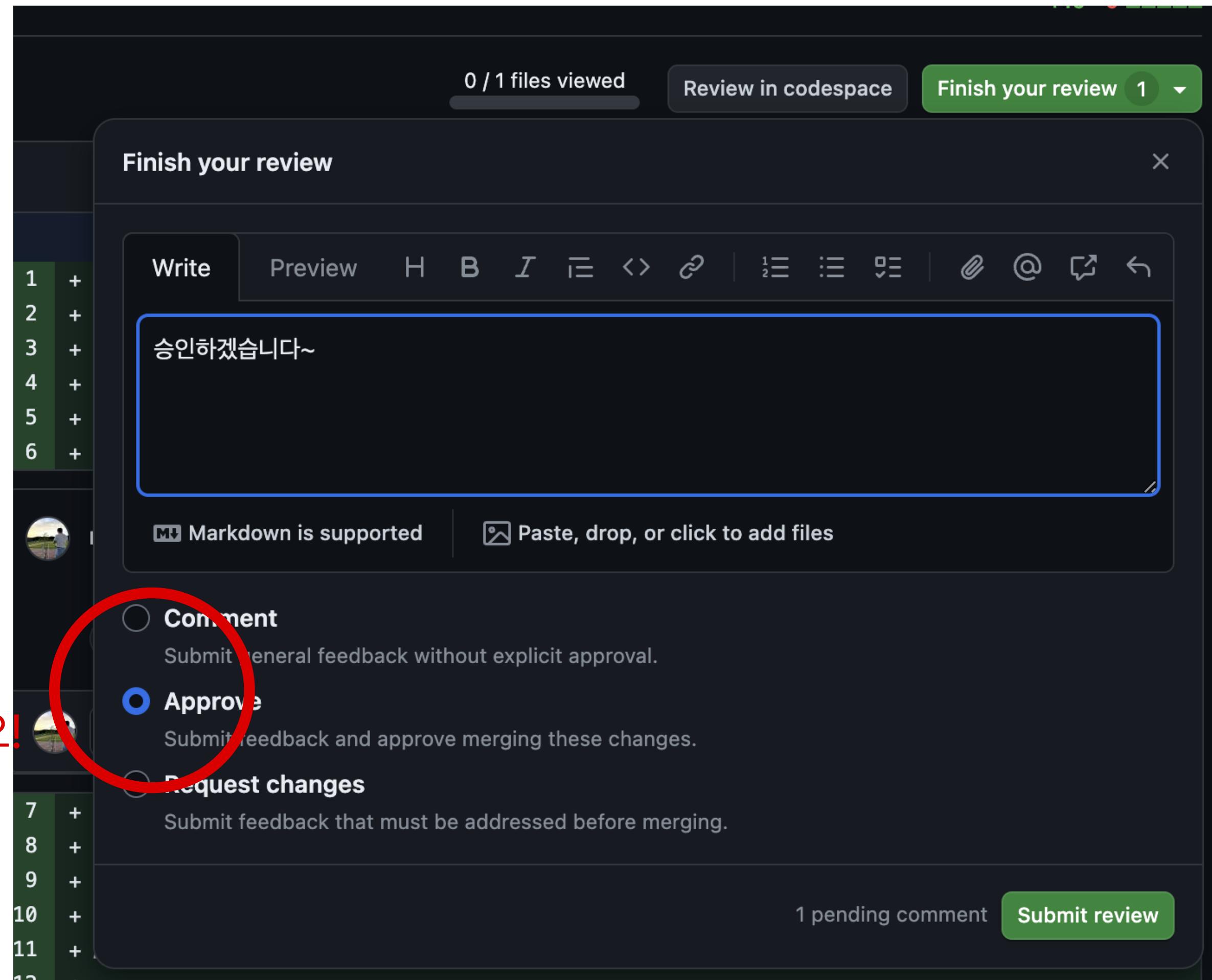
### Comment

명시적으로 Approve를  
제외하고, 순수하게 피드백하기

PR에 별 문제가 없다면  
Approve를 선택해주세요!

### Request changes

변경 사항을 반영하고 싶지 않을 경우



## 참고 링크

- Let's git it started [Git Repository](#)
- 이론 가이드
- 실습 가이드
- Quick Guide

# **End of Document**

# Let's git it started

Git/GitHub를 통해 친해져보아요!

# Let's git it started

---

Git/GitHub와 Markdown 문법 공부

협업을 하기 위한 첫 걸음

나아가 Git/GitHub를 통해 결과물 만들기



# Contents

01

Let's git it started!

Git/GitHub란?

Git 명령 소개

GitHub 기능 소개

02

Dive into 'Git'

우리 서로 더 알아보기

마크다운 문법이란?

Git 실습하기

03

Feedback/Review

Pull Request 작성하기

리뷰하기

# Chapter1. Let's git it started!

01 Git/GitHub는 무엇일까?

02 Git의 명령 소개

add, commit, push, branch, clone

03 GitHub 기능 소개

pull request, issue, review, merge

# 01 Git/GitHub는 무엇일까?

## 버전 관리



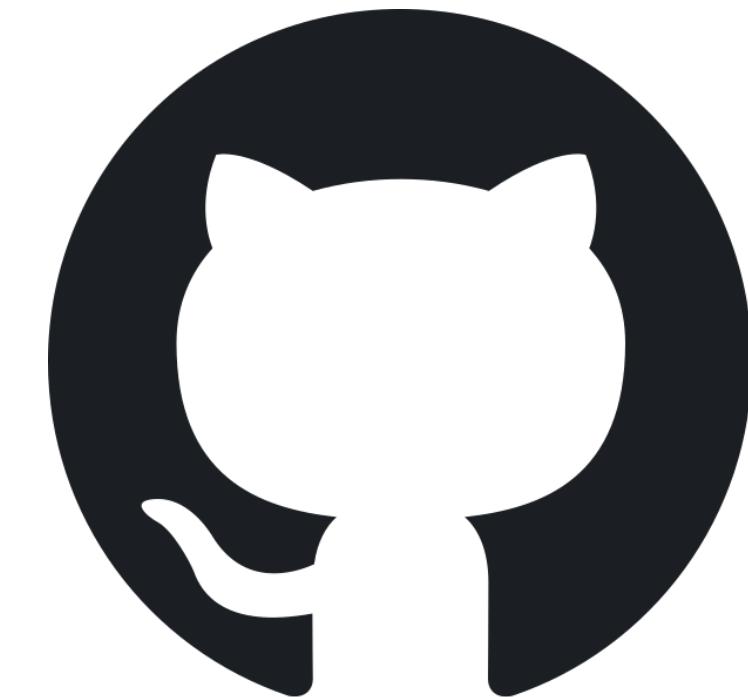
# 01 Git/GitHub는 무엇일까?



- 버전관리 시스템  
(version control system)
- 자유롭게 코드를 더하고 빼고!
- 여러 사람들이 동시에 개발 하기에 편리함

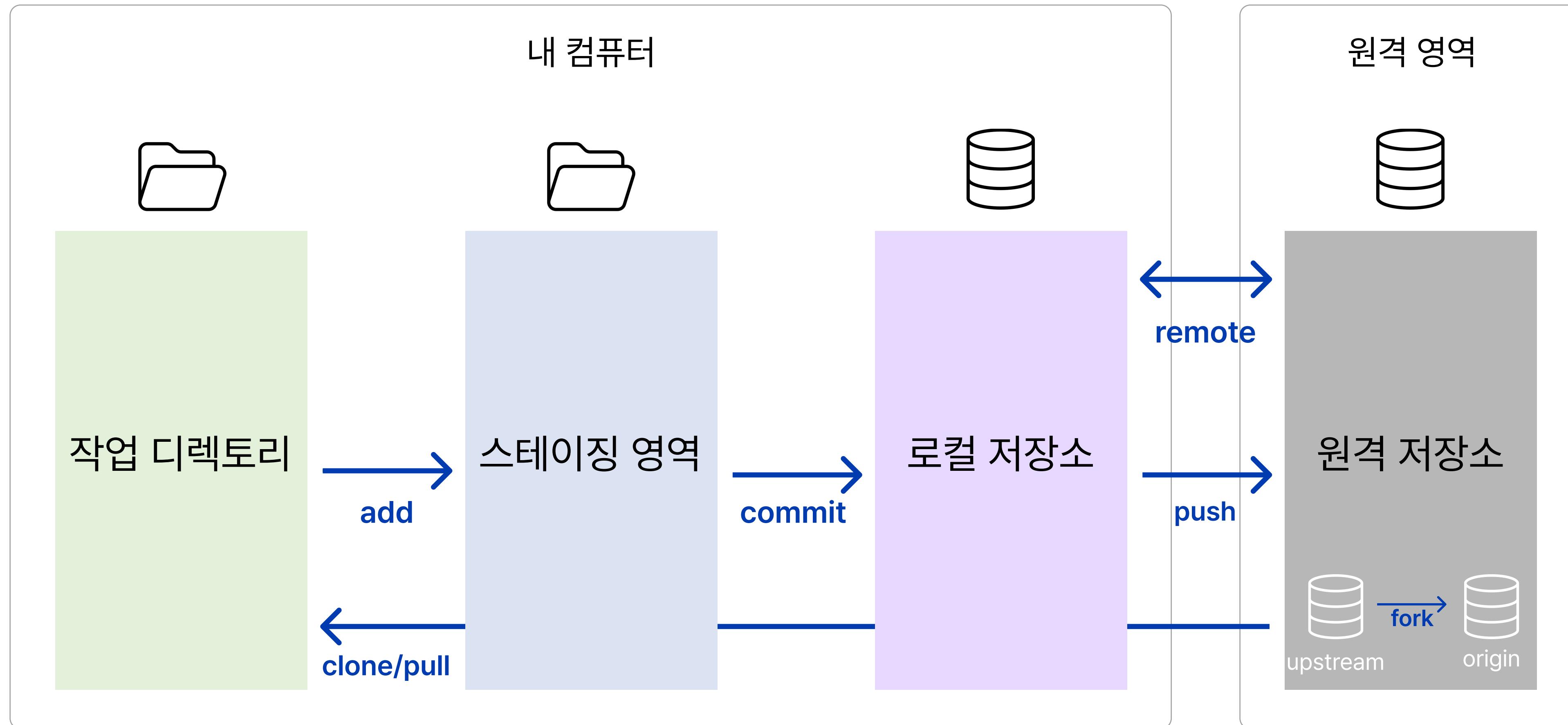
- Git의 데이터를 저장하는 서버
- 커뮤니티 기능이 잘 되어 있음
- 코드를 보면서 의견을 나눔
- 오픈 소스의 발전

# 01 Git/GitHub는 무엇일까?



여러 대의 컴퓨터들과 하나의 메인 서버가 소통하는 관계  
Git GitHub

## 02 Git의 명령어를 알아보자!



## 02 Git의 명령어를 알아보자!

: add, commit, push, branch, clone

1. **add**: 저장을 원하는 파일들을 묶어 올리겠다. / 스테이지에 올린다.

(내가 구현한 기능을 우선 너에게 전달하기 위한 준비를 할게!)

2. **commit**: 저장하는 파일들을 묶어서 save 하는 행동

(내가 구현한 기능들은 말이지... 'horse'야!)

3. **push**: 내 컴퓨터에 저장된 작업 내용을 github에 업로드한다.

(내가 'horse'를 구현했는데 너희들에게 보낼게! 확인 부탁해~)

## 02 Git의 명령어를 알아보자!

: add, commit, push, branch, clone

4. **branch**: 개발 영역을 분담해서 작업할 때, 충돌 예방 방지를 위해 개발 branch를 생성해, 기존에 있던 내용을 가져와 개별적으로 작업

(아, 너가 'horse'를 구현했으니! 나는 'dolphin'을 구현해야겠군!)

5. **clone**: 원격 저장소 내려 받기 (저장소 복제하기)

(horse v1 기반으로 v1.1을 만드려는데 기존에 작성한 코드를 참고해서 해야겠다.)

## 03 GitHub의 기능을 알아보자!

: pull request, review, merge

### 1. **pull request**: GitHub의 강력한 협업 기능

(세상 사람들! 우리 'horse'가 걸을 수 있어요~ 한번 이상 없는지 봐주세요!)

### 2. **review**: pull request 내용을 점검하는 기능, 리뷰를 통한 공동 학습

(오 네! 알겠습니다. 음 'horse'의 이 부분은 좀 더 수정봐야할 것 같아요!)

### 3. **merge**: 각각의 branch를 합치는 과정

(오! 'horse' 가 사용하기 괜찮네요! Merge 해드릴게요! 너 우리의 동료가 되어라!)

# Chapter2. Dive into 'Git'

01 우리 서로 더 알아보기

02 마크다운 문법이란?

03 Git 실습하기

## 01 우리 서로 더 알아보깃

Step1. 자기 소개글을 작성 후 Git을 이용하여 GitHub에 글 올리기

Step2. Pull Request 작성하기

Step3. 서로의 소개글을 읽으면서 궁금하거나,  
공감가는 내용이 있다면 Review를 통해 소통하기

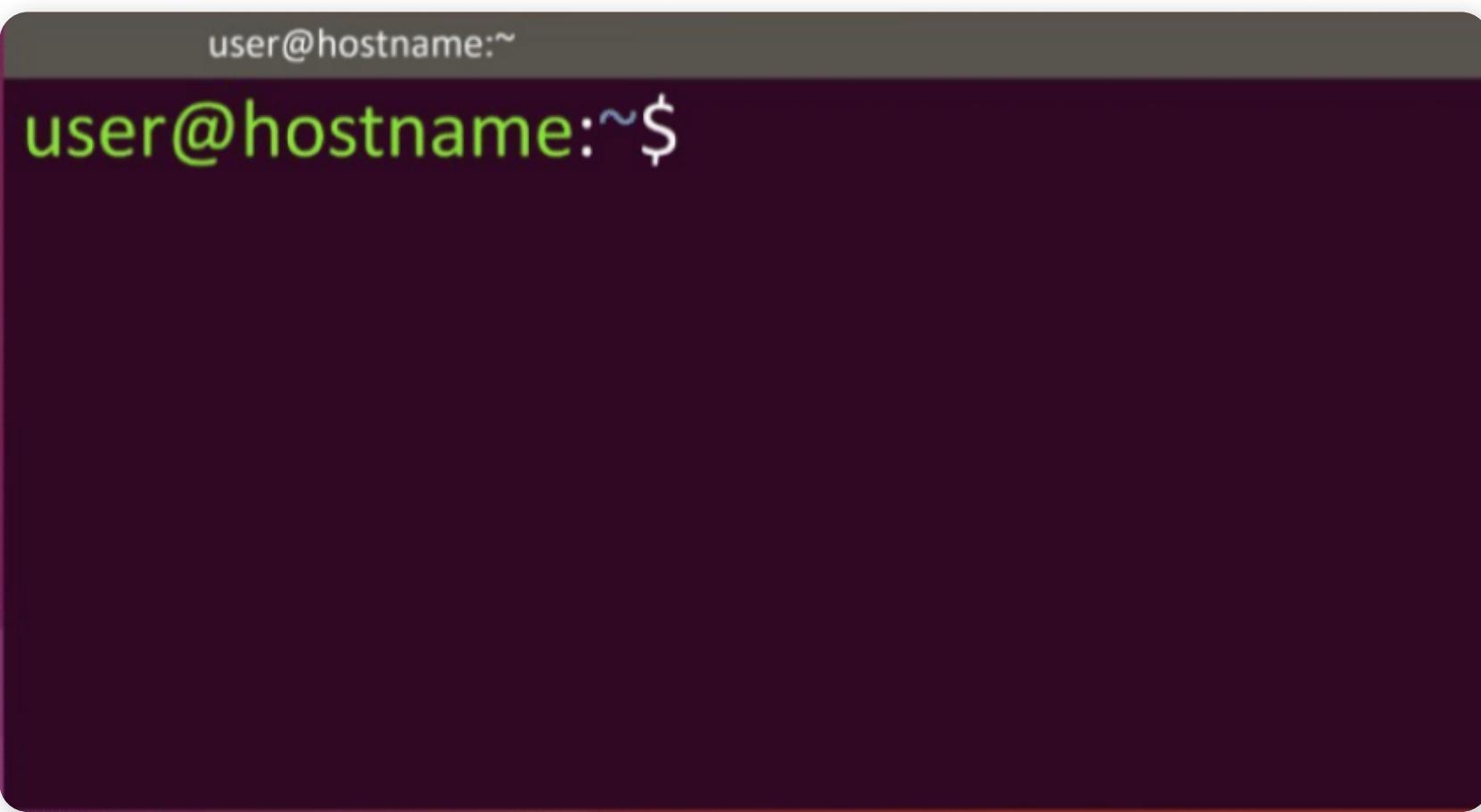
## 02 마크다운 문법이란?

- 2004년 존 그루버에 의해 만들어진 텍스트 기반 마크업 언어
- HTML로 쉽게 변환 가능
- 간단한 구조의 문법
- GitHub의 저장소에 관한 정보를 기록하는 **README.md**로 인해 주목 받고 있음

## 03 Git 실습하기

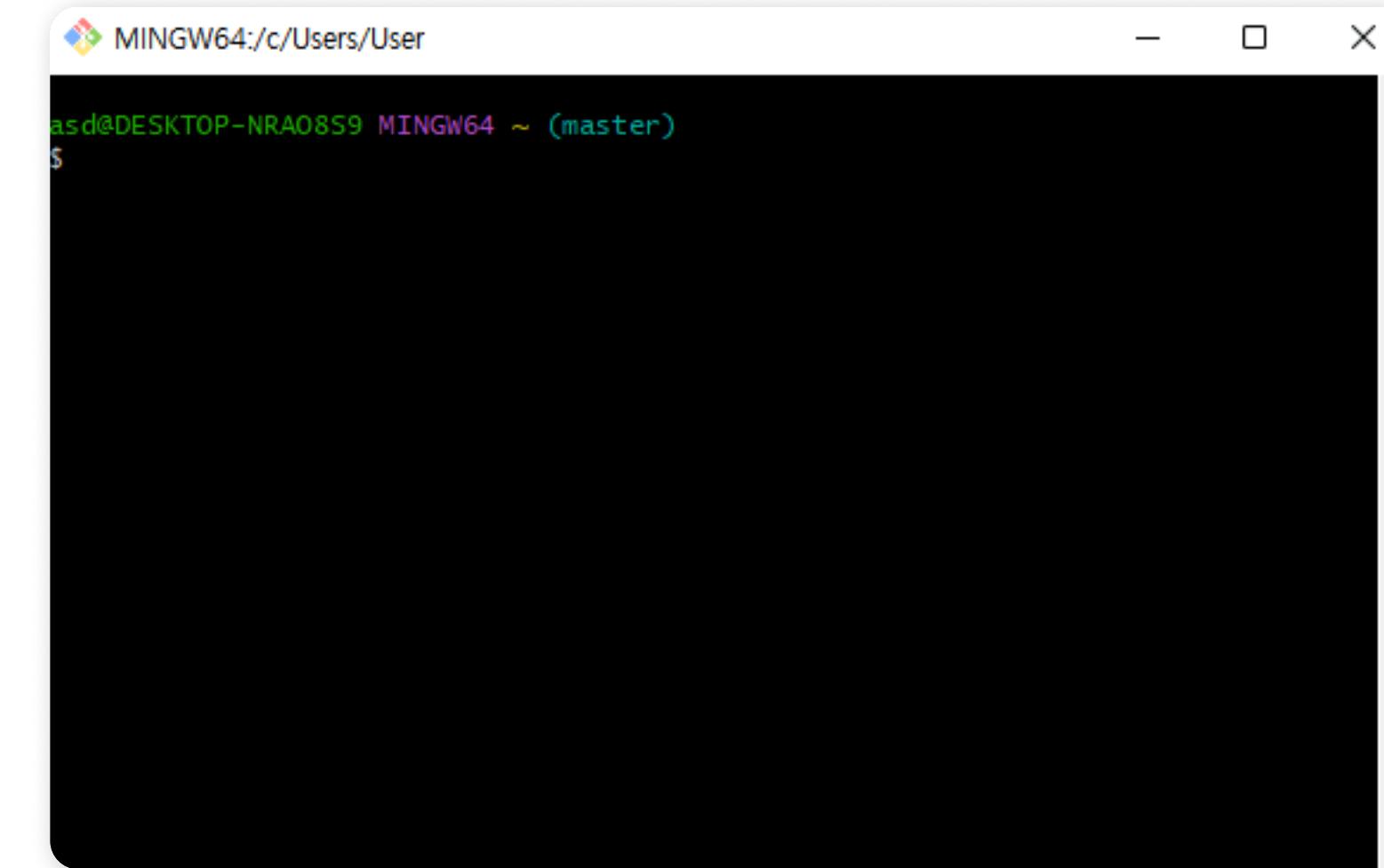
### Step0. 명령어 창 띄우기

UNIX / terminal



```
user@hostname:~  
user@hostname:~$
```

Windows / git bash



```
MINGW64:/c/Users/User  
asd@DESKTOP-NRA08S9 MINGW64 ~ (master)  
$
```

## 03 Git 실습하기

Step1. 작업할 폴더(디렉토리)로 이동하기

**\$ cd (상대 혹은 절대) 경로** cd; change directory

**\$ mkdir 폴더이름** mkdir; make directory

**\$ ls** ls; list

```
asd@DESKTOP-NRA08S9 MINGW64 ~ (master)
$ cd desktop 원하는 위치로 이동

asd@DESKTOP-NRA08S9 MINGW64 ~/desktop (master)
$ mkdir practice 폴더 생성

asd@DESKTOP-NRA08S9 MINGW64 ~/desktop (master)
$ cd practice 생성한 폴더로 이동

asd@DESKTOP-NRA08S9 MINGW64 ~/desktop/practice 경로가 바뀐 것을 확인!
$ ls

asd@DESKTOP-NRA08S9 MINGW64 ~/desktop/practice (master)
$
```

예시일 뿐이니 원하는 폴더에서 진행해주세요!

## 03 Git 실습하기

### Step2. Git 저장소 생성 및 최초설정

- 작업할 폴더 생성 후 해당 위치에서 `git init`

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/practice/.git/
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$
```

## 03 Git 실습하기

### Step2. Git 저장소 생성 및 최초설정



Git 사용이 처음인 경우 실시



```
$ git config --global user.name "Your name"  
$ git config --global user.email "Your email"  
$ git config --global --list 제대로 설정이 되었는지 확인
```

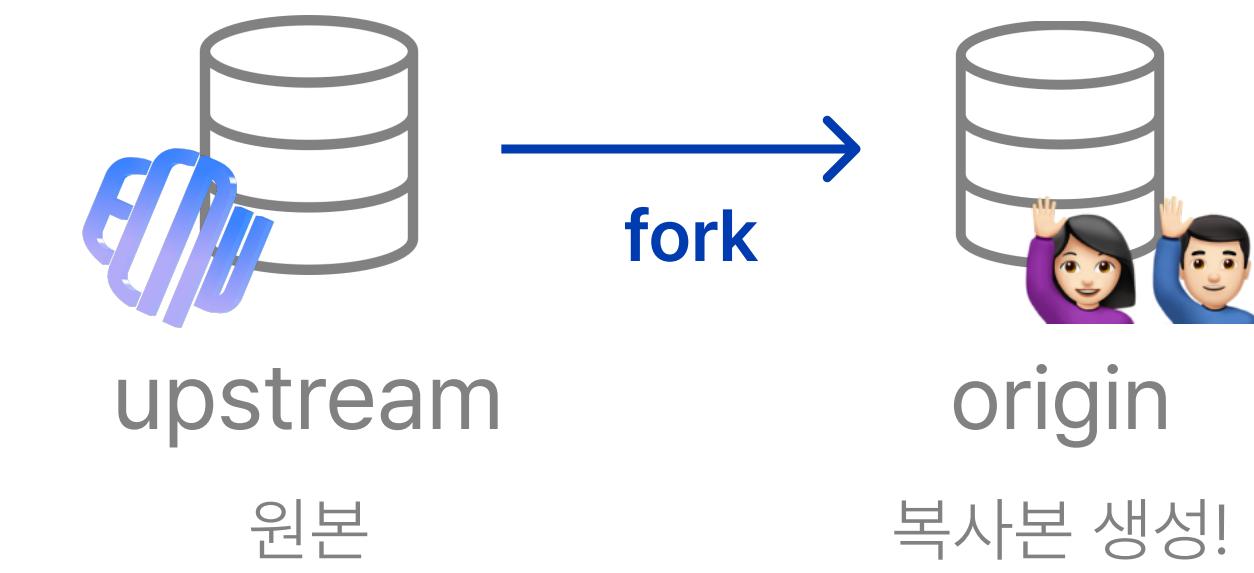
- 이름과 이메일 주소를 설정하는 이유  
→ commit 할 때 사용자 이름과 이메일이 기록됨

## 03 Git 실습하기

### Step3. 원본 저장소를 내 계정으로 복사하기 - fork

The screenshot shows a GitHub repository page for 'Let-s-git-it-started'. At the top, the repository name is displayed with a red box around it. Below the header, there are navigation links for Code, Issues, Pull requests, Zenhub, Actions, Projects, Security, Insights, and Settings. A red circle with the Korean text '클릭!' (Click!) points to the 'Fork' button in the top right corner of the header bar, which has a count of 7 forks. The main content area shows the repository's structure with a 'main' branch, 5 branches, and 0 tags. A list of recent commits is shown, including a merge pull request from 'baegyeong' and several commits from the author. On the right side, there is an 'About' section with a 'Readme' file, activity stats (1 star, 0 watching), and a 'fork' count of 7. There are also sections for 'Releases' (no releases) and 'Packages'.

fork는 다른 계정의 원격 저장소를  
내 계정으로 가지고 올 때 사용



# 03 Git 실습하기

## Step3. 원본 저장소를 내 계정으로 복사하기 - fork

**Create a new fork**

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (\*).

Owner *	Repository name *
baegyeong	/ Let-s-git-it-started <small>Let-s-git-it-started is available.</small>

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Copy the main branch only  
Contribute back to JNU-econovation/Let-s-git-it-started by adding your own branch. [Learn more.](#)

(i) You are creating a fork in your personal account.

**클릭!**

**Create fork**

내 계정으로 복사된 것을 확인

baegyeong / Let-s-git-it-started

Let-s-git-it-started (Public)

forked from [JNU-econovation/Let-s-git-it-started](#)

main 1 branch 0 tags

Go to file Add file ▾ Code ▾

This branch is up to date with JNU-econovation/Let-s-git-it-started:main.

baegyeong Merge pull request JNU-econovation#5... d467e2a 2 hours ago 123 commits

.github/ISSUE\_TEMPLATE Update issue templates 4 months ago

1st\_Let\_s\_git\_it\_started(G... docs: add profile.md 3 hours ago

2nd\_Let\_s\_git\_it\_started(... fix: Let's git it started 디렉토리 구조 최신화 10 hours ago

Glossary\_of\_Terms fix: Let's git it started 디렉토리 구조 최신화 10 hours ago

.gitignore docs: README.md, .gitignore 생성, guide 문서 생성 7 months ago

README.md docs: README.md, .gitignore 생성, guide 문서 생성 7 months ago

README.md

No description, website, or topics provided.

Readme

Activity

0 stars

0 watching

8 forks

No releases published

Create a new release

Packages

No packages published

Publish your first package

## 03 Git 실습하기

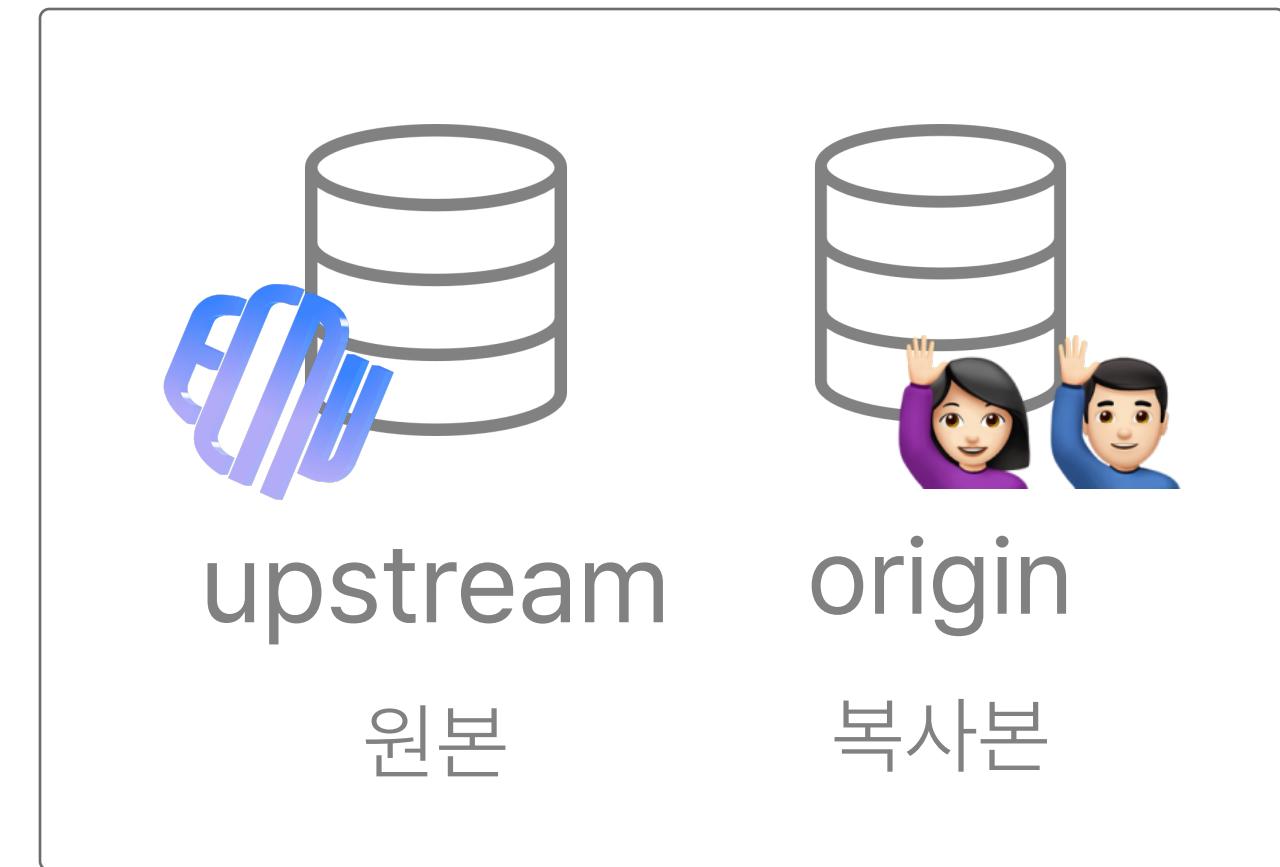
Step4. 자신의 local에 원격 저장소 등록하기

remote  
(github)

**\$ git remote add [저장소 이름] [저장소 주소]**

💡 실습을 위해 두 가지 저장소를 등록합니다.

1. 에코노베이션 레포지토리(원본)
2. 나의 레포지토리 (fork한 복사본)



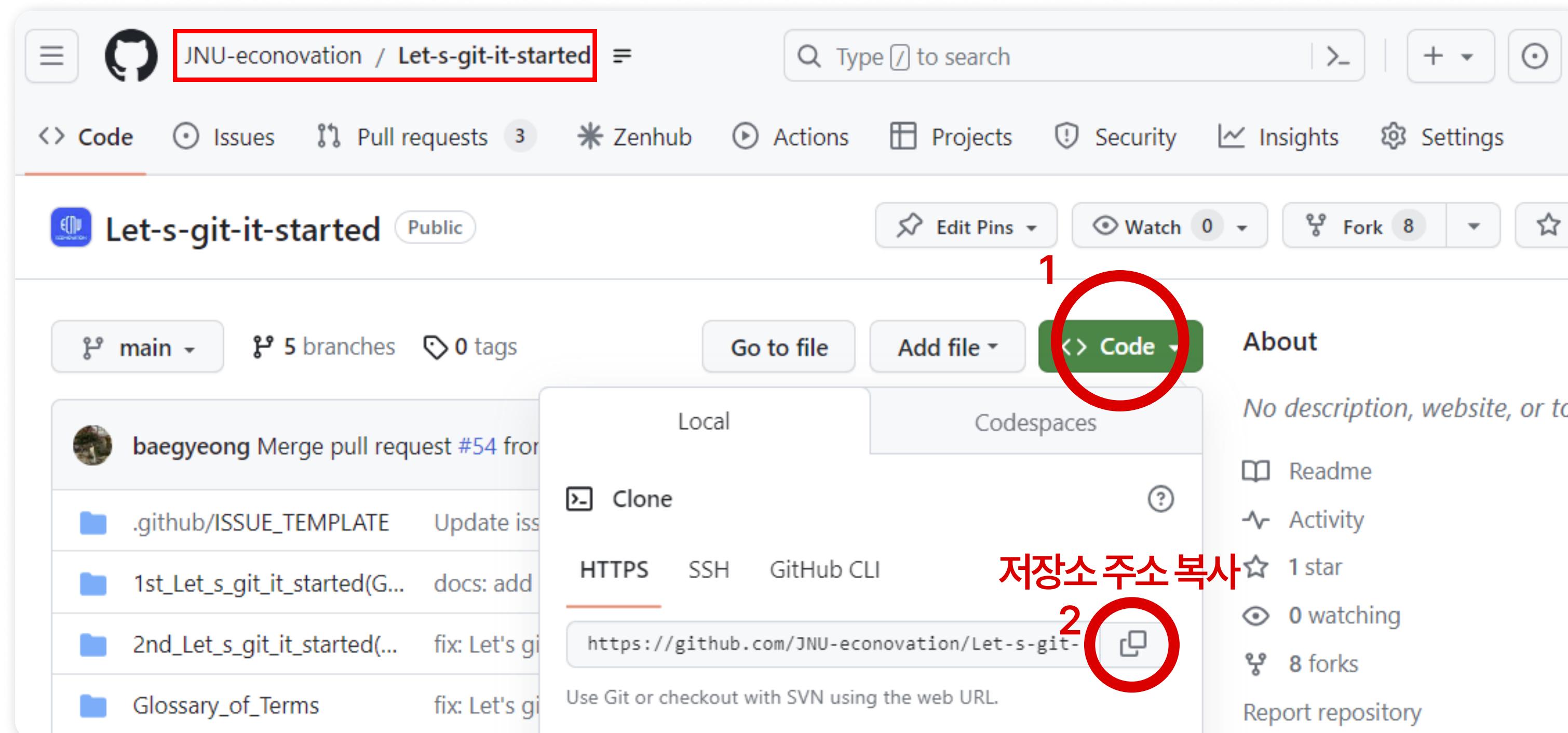
remote, pull

local  
(내 pc)

## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

### Step 4-1. 원본 레포지토리(저장소) 등록하기 - upstream



## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소(레포지토리) 등록하기

Step 4-1. 원본 레포지토리(저장소) 등록하기 - **upstream**

**\$ git remote add [저장소 이름] [저장소 주소]**

**\$ git remote -v**

현재 로컬 저장소와 연결된 원격 저장소들의 정보 확인

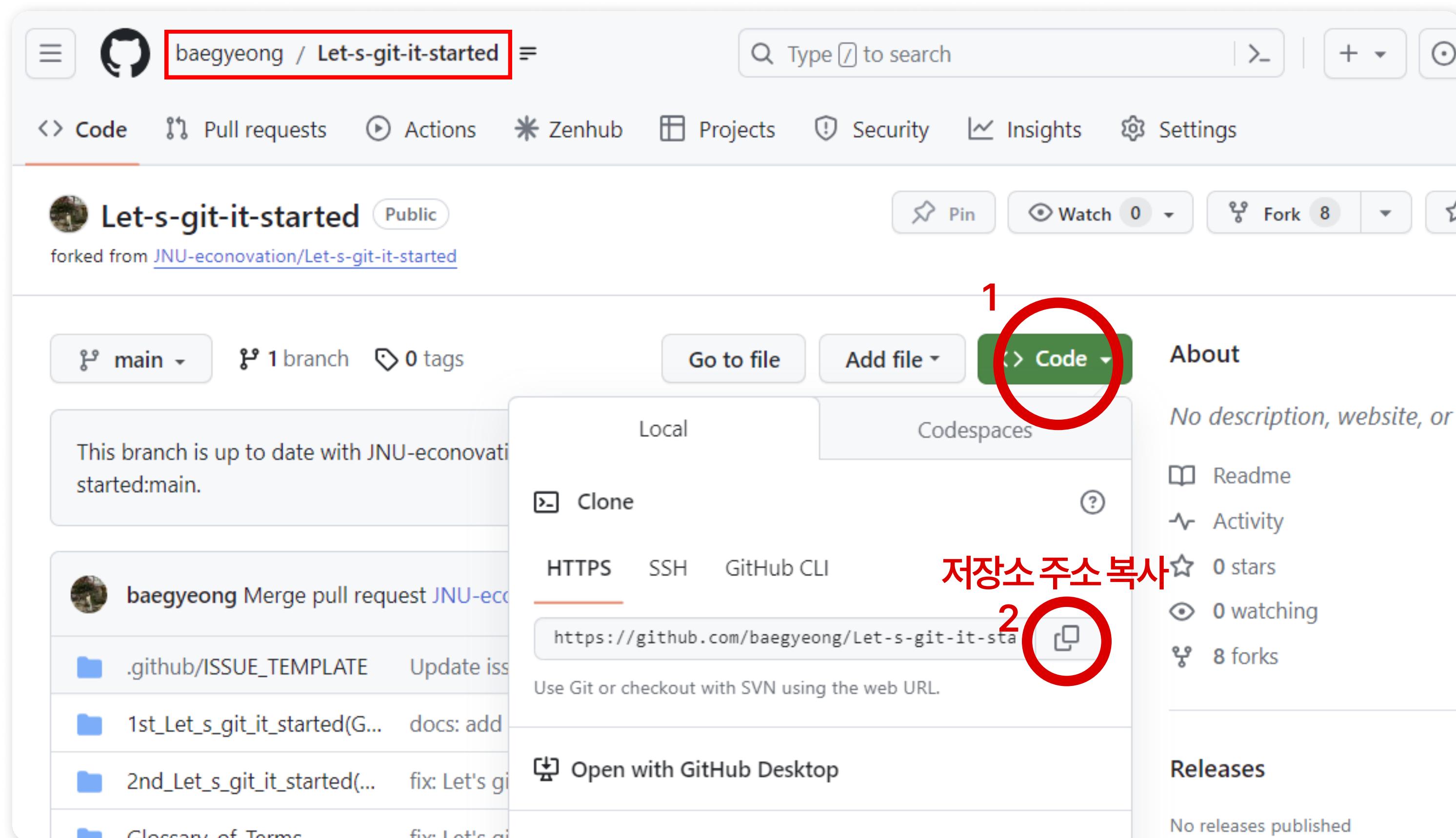
```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ git remote add upstream https://github.com/JNU-econovation/Let-s-git-it-started.git

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ git remote -v 저장소가 잘 등록됐는지 확인
upstream      https://github.com/JNU-econovation/Let-s-git-it-started.git (fetch)
upstream      https://github.com/JNU-econovation/Let-s-git-it-started.git (push)
```

## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

### Step 4-2. fork한 내 레포지토리(저장소) 등록하기 - origin



## 03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

Step 4-2. fork한 내 레포지토리(저장소) 등록하기 - **origin**

**\$ git remote add [저장소 이름] [저장소 주소]**

**\$ git remote -v**

현재 로컬 저장소와 연결된 원격 저장소들의 정보 확인

```
asd@DESKTOP-NRAO8:~$ git remote add origin https://github.com/baegyeong/Let-s-git-it-started.git
asd@DESKTOP-NRAO8:~$ git remote -v
origin  https://github.com/baegyeong/Let-s-git-it-started.git (fetch)
origin  https://github.com/baegyeong/Let-s-git-it-started.git (push)
upstream  https://github.com/JNU-econovation/Let-s-git-it-started.git (fetch)
upstream  https://github.com/JNU-econovation/Let-s-git-it-started.git (push)
```

upstream(원본 레포지토리)와 origin(fork한 내 레포지토리)가 잘 등록된 것을 확인

## 03 Git 실습하기

Step5. 자신의 local에 원격 저장소 내역을 가져오기

**\$ git pull [저장소 이름] [가져올 브랜치]**



원본 저장소(upstream)의 현재 학기 브랜치(ex. 2023-2)를 가져와야 합니다.

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ git pull upstream 2023-2 원격 저장소에 파일 내역 가져오기
remote: Enumerating objects: 486, done.
remote: Counting objects: 100% (226/226), done.
remote: Compressing objects: 100% (155/155), done.
remote: Total 486 (delta 116), reused 113 (delta 64), pack-reused 260
Receiving objects: 100% (486/486), 2.06 MiB | 10.34 MiB/s, done.
Resolving deltas: 100% (235/235), done.
From https://github.com/JNU-econovation/Let-s-git-it-started
 * branch            2023-2      -> FETCH_HEAD
 * [new branch]      2023-2      -> upstream/2023-2

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ ls 현재 폴더의 파일 목록 확인하기(pull을 통해 가져온 내역 확인)
'1st_Let_s_git_it_started(Git_tutorial)'/  '2nd_Let_s_git_it_started(Git_Flow)'/  README.md
2023-2/                                     Glossary_of_Terms/
```

## 03 Git 실습하기

### Step6. 준비해 온 자기소개 글 수정하기

**\$ cp [복사 할 파일] [복사 될 파일]**

- 💡 자기소개 템플릿 파일인 profile.md를 복제합니다.  
profile.md 파일은 수정하지 마세요!

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ cd 2023-2/ 2023-2 폴더로 이동

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ ls 2023-2 폴더에 어떤 파일이 있는지 확인
profile.md

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ cp profile.md econo.md profile.md 파일을 [이름].md 파일로 복사

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ ls [이름].md 파일이 생성된 것을 확인
econo.md profile.md
```

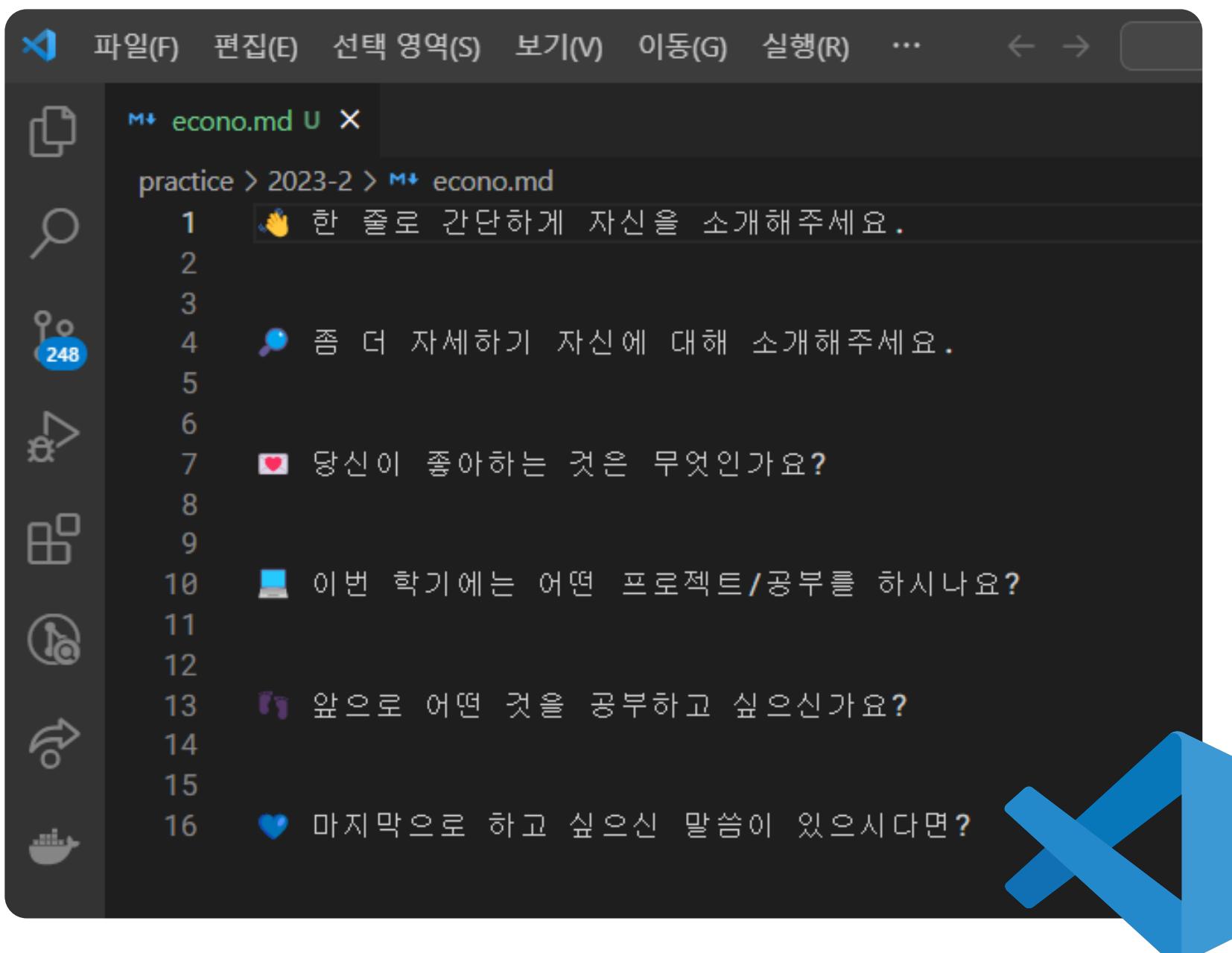
## 03 Git 실습하기

Step5. 준비해 온 자기소개로 글 수정하기  [Markdown 문법 정리](#)

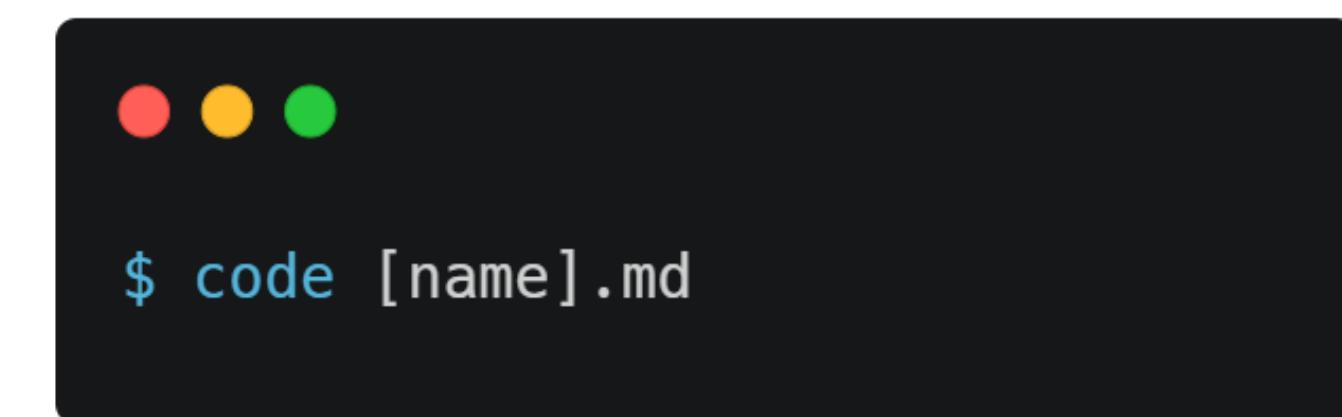


### 글 작성 에디터를 실행하는 다양한 방법

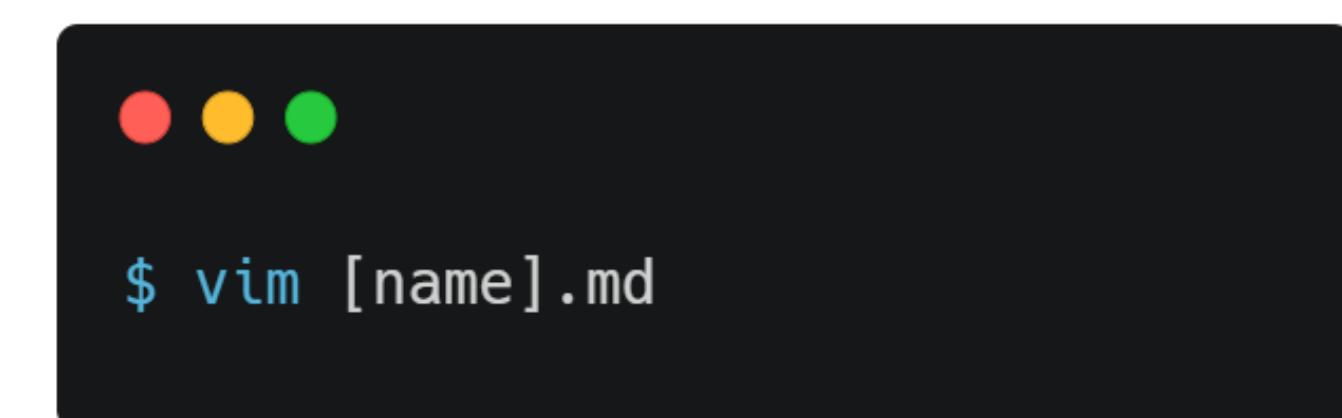
#### 1. Visual Studio Code 직접 실행하기



#### 2. 명령어로 Visual Studio Code 실행시키기



#### 3. 터미널에서 작성하기 - Vim

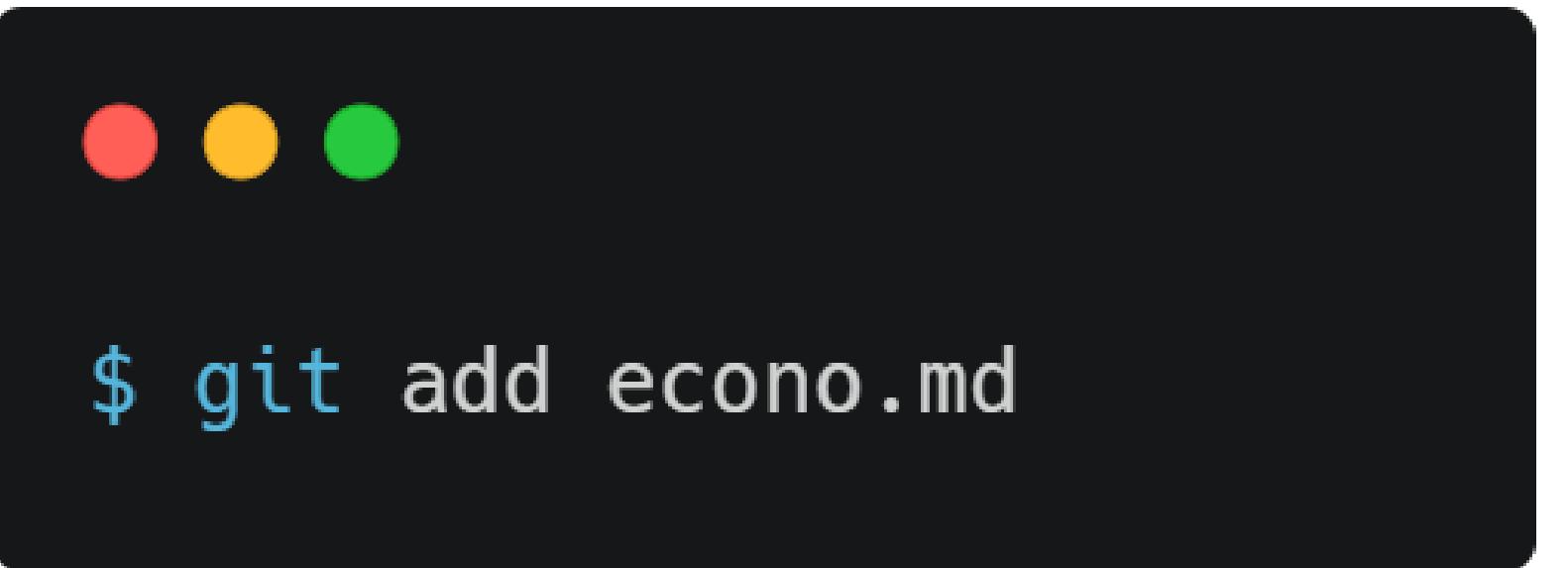


## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**\$ git add filename**

자신이 clone한 local 저장소의 변경사항을 GitHub에 반영하겠다.

A dark gray rectangular box representing a terminal window. In the top-left corner, there are three small colored circles: red, yellow, and green. Below them, the command '\$ git add econo.md' is written in white text.

```
$ git add econo.md
```

- 💡 파일 add 전 후로, '[git status](#)' 명령어를 통해  
현재 작업 중인 저장소의 상태를 확인하는 습관 들이기!

## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

### \$ git add filename

자신이 clone한 local 저장소의 변경사항을 GitHub에 반영하겠다.

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ ls
econo.md profile.md 현재 폴더의 파일 목록 확인

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git add econo.md 스테이징 영역에 파일 추가하기

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git status 저장소 상태 확인
on branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   econo.md 파일이 잘 추가 된 것을 확인!
```

## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**\$ git commit -m "message 내용"**

변경사항에 대해서 "이러한 내용"이 담긴 메시지를 남기겠다.

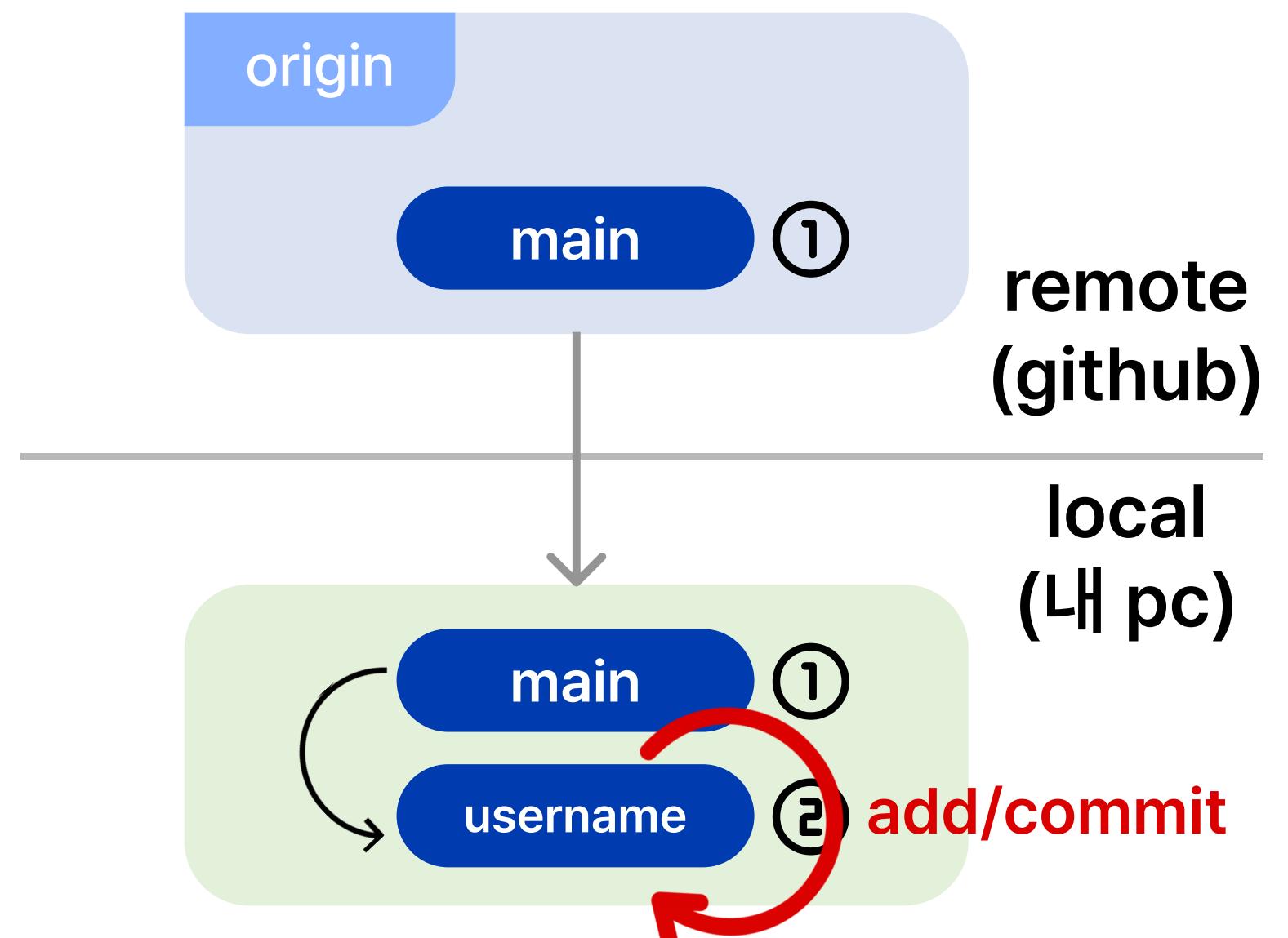
💡 message 내용은 아래 형식을 지켜주세요.



```
$ git commit -m "docs: add [이름].md"
```

원활한 협업을 위해서 commit message를 어떻게 쓰느냐가 정말 중요해요!  
commit 메시지 잘 쓰는 방법이 궁금하시다면 아래의 링크를 참고해주세요.

🔗 커밋 컨벤션



## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**\$ git commit -m "message 내용"**

변경사항에 대해서 "이러한 내용"이 담긴 메시지를 남기겠다.

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git commit -m "docs: add econo.md" 커밋 메시지 작성
[master 89ee9a6] docs: add econo.md
 1 file changed, 16 insertions(+)
 create mode 100644 2023-2/econo.md
```

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

스테이징 영역 -> 로컬 저장소 이동  
스테이징 영역은 비워지게 됨

## 03 Git 실습하기

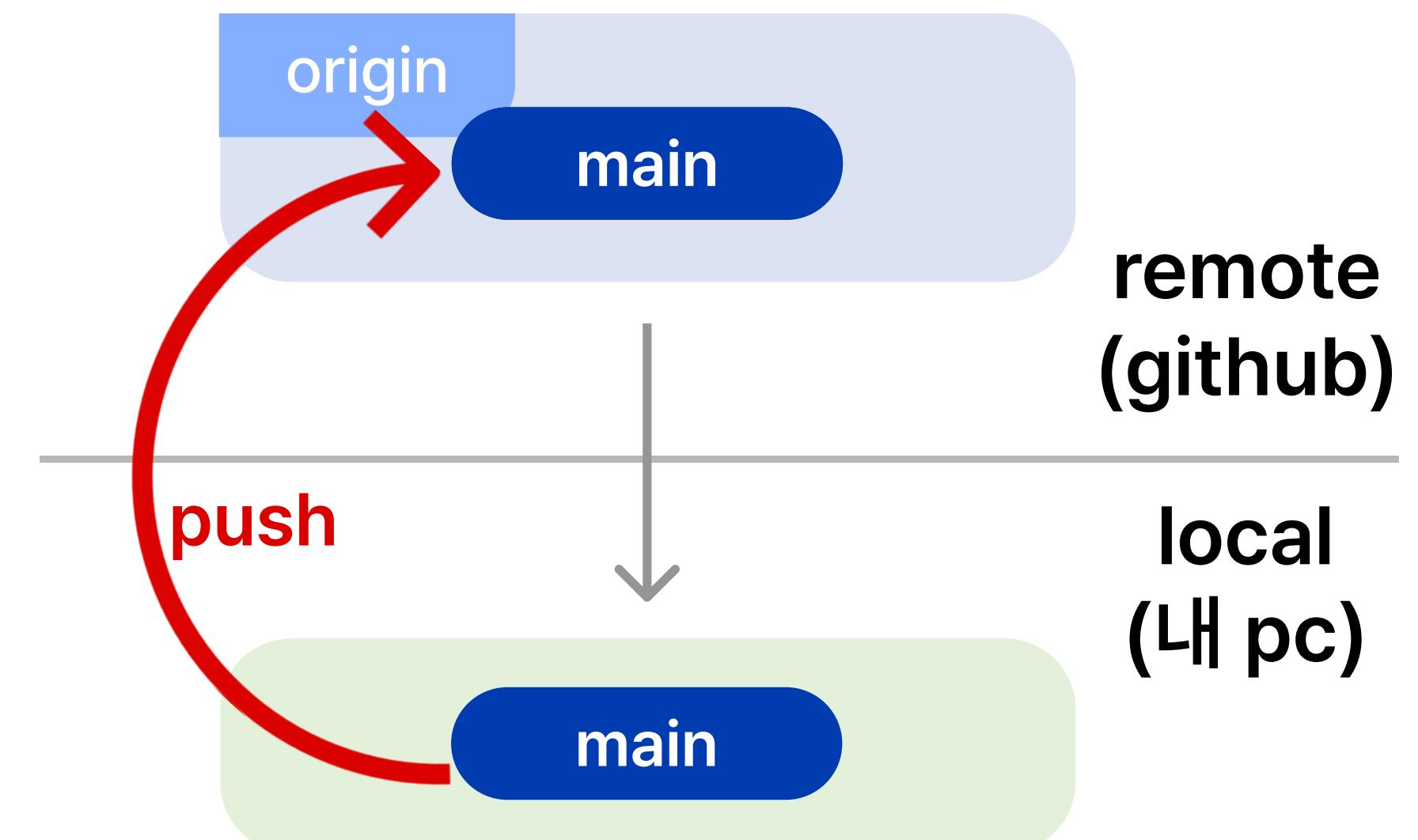
Step6. 그럼 이제! 글을 올려보자

**\$ git push [저장소 이름] [브랜치]**

원격 저장소(remote)의 주소

현재 자신이 위치한 branch 이름

```
● ● ●  
$ git push origin main
```



## 03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

**git push origin main --force**

**\$ git push [저장소 이름] [브랜치]**

💡 현재 브랜치가 master라면?

깃허브에서는 'main'으로 브랜치가 설정되어 있기 때문에,  
만약 터미널에 반영이 안 됐다면 바꿔줘야 합니다.

1. 현재 로컬 브랜치는 master

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
```

\$ git branch -M main 2. main으로 이름 변경

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (main)
```

3. 브랜치명이 변경된 것을 확인!

```
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (main)
$ git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 1.63 KiB | 1.63 MiB/s, done.
Total 10 (delta 4), reused 6 (delta 2), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/baegyeong/Let-s-git-it-started.git
d467e2a..89ee9a6  main -> main

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (main)
```

# Chapter3. Feedback/Review

01 Pull Request 작성하기

02 다른 사람 소개 글 읽고 리뷰하기

# 01 Pull Request 작성하기

The screenshot shows the GitHub interface for the repository `baegyeong / Let-s-git-it-started`. The top navigation bar includes links for Code, pull request, Actions, Zenhub, Projects, Security, Insights, and more. A red circle highlights the 'pull request' tab. Below it, a search bar filters for open pull requests. A green button labeled 'New pull request' is circled in red. The main content area displays a 'Welcome to pull requests!' message and instructions for creating a pull request.

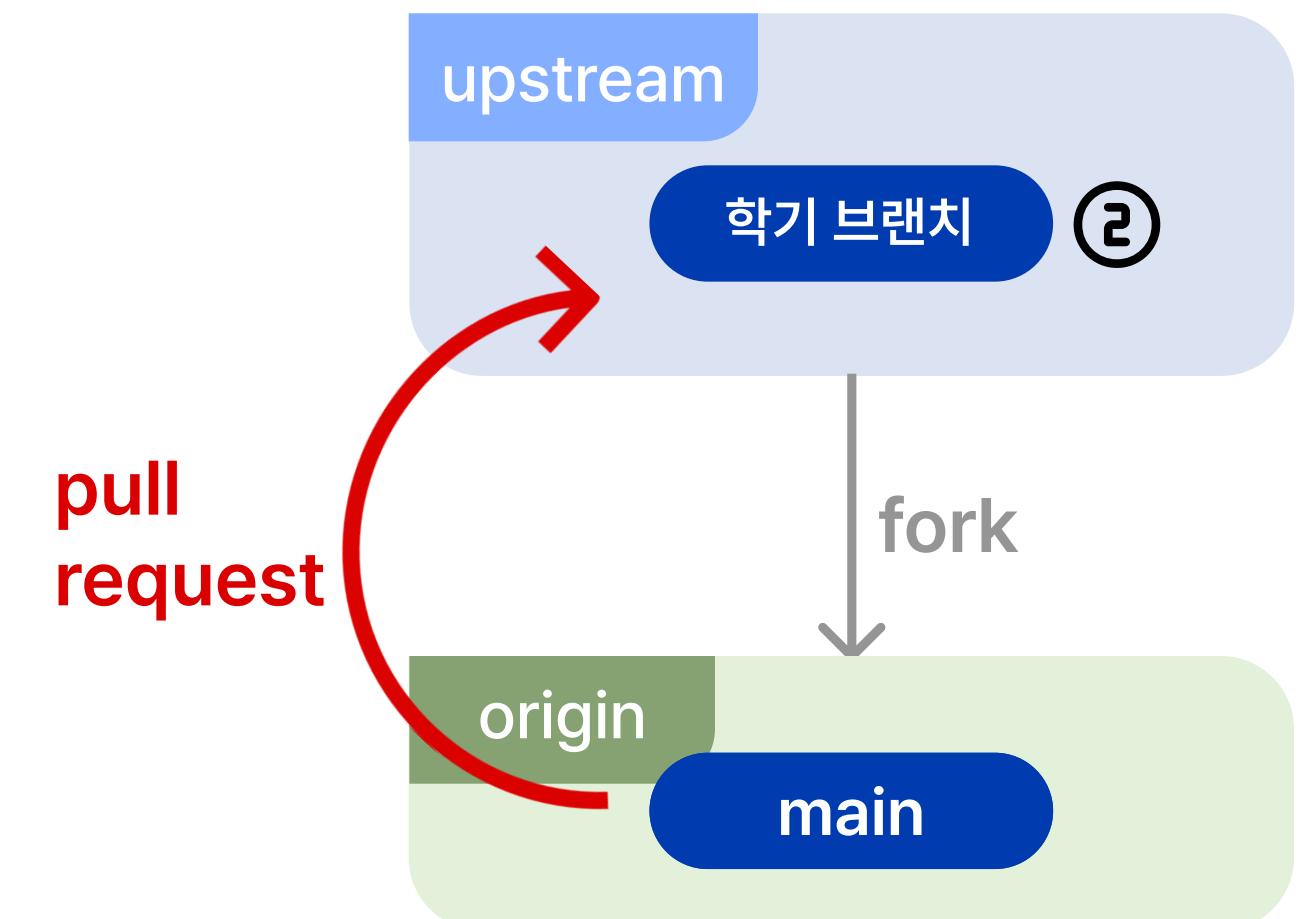
1. pull request 클릭

2. New pull request 클릭

The screenshot shows the GitHub interface for the repository `JNU-econovation / Let-s-git-it-started`. It displays a comparison between branches. A red circle highlights the dropdown menu for the base branch, which is set to '2023-2'. Below it, a message states 'Able to merge. These branches can be automatically merged.' A red circle highlights the 'base' dropdown. The right side of the screen shows a summary of the pull request changes, including 1 commit, 1 file changed, and 1 contributor. A red circle highlights the 'Create pull request' button.

3. 해당학기 브랜치로 변경

4. 클릭



# 01 Pull Request 작성하기

The screenshot shows the GitHub 'Open a pull request' interface for the repository 'JNU-econovation / Let-s-git-it-started'. The interface is divided into three main sections:

- 1. pr을 보내는 브랜치 확인**  
내 레포 'main' -> 원본 레포 '해당 학기 브랜치'
- 2. 제목과 내용 작성**  
제목: 이름(아이디) - 자기소개  
내용: 자유롭게 구성
- 3. 라벨 지정**  
(기수, mbti, 자기소개)

**1. pr을 보내는 브랜치 확인**: The top navigation bar shows the base repository as 'JNU-econovation/Let-s-git-it-st...' and the base branch as '2023-2'. The dropdown for the head repository is set to 'baegyeong/Let-s-git-it-started' and the compare branch is 'main'. A red box highlights this section.

**2. 제목과 내용 작성**: The main form area contains a title '에코노(econo) - 자기소개' and a rich text editor with the content '안녕하세요 에코노입니다! 잘 부탁드려요~'. A red box highlights this section.

**3. 라벨 지정**: The right sidebar shows the 'Labels' section with two selected labels: '25기' and '자기소개'. A red box highlights this section.

## 02 다른 사람의 소개글 읽고 리뷰하기

The screenshot shows a GitHub repository page for 'JNU-econovation / 2022-1-I-AM-GROUND'. The 'Pull requests' tab is selected, displaying 9 open pull requests. The search bar at the top contains the query 'is:pr is:open'. The pull requests are listed in descending order of creation date:

- #11: 21기 이윤성 자기소개입니다. (ENFP) [자기소개]. Opened on May 10, 2022 by 2yunseong.
- #10: 23기 나인혜 자기소개입니다. Opened on Apr 29, 2022 by nainhye • Approved.
- #9: 23기 조배경 자기소개입니다. (ISFJ) [자기소개]. Opened on Apr 29, 2022 by baegyeong • Approved.
- #8: 23기 김지민 자기소개입니다. (ENFP) [자기소개]. Opened on Apr 29, 2022 by jminkkk • Approved.
- #7: 최명선(myungsun) 자기소개 (ESFJ) [자기소개]. Opened on Apr 29, 2022 by myungsun7782 • Approved.
- #5: 23기 이도연 자기소개입니다. (enfi) [자기소개]. Opened on Apr 29, 2022 by capDoYeonLee • Approved.
- #4: 23기 이규민 자기소개입니다. [자기소개]. Opened on Apr 29, 2022 by Klomachenko • Approved.

## 02 다른 사람의 소개글 읽고 리뷰하기

JNU-econovation / 2022-1-I-AM-GROUND Public

Code Issues Pull requests 9 Actions Projects Wiki Security Insights Settings

### 최명선(myungsun) 자기소개 #7

Open myungsun7782 wants to merge 1 commit into master from myungsun

Conversation 11 Commits 1 Checks 0 Files changed 1

myungsun7782 commented on Apr 29, 2022

안녕하세요! 에코노베이션 22기 최명선입니다.

Add: myungsun.md 9b6b9e8 Still in progress

myungsun7782 added **자기소개** **ESFJ** labels on Apr 29, 2022

Reviewers  
mywnajs  
2yunseo  
saint683

Assignees  
No one—assigned

클릭!

## 02 다른 사람의 소개글 읽고 리뷰하기

Open 최명선(myungsun) 자기소개 #7  
myungsun7782 wants to merge 1 commit into master from myungsun

saint6839 approved these changes on Apr 30, 2022 [View reviewed changes](#)

myungsun.md

```
6 +  
7 + 안녕하세요:) 저는 작년에 소프트웨어공학과로 편입을 하고 3학년 2학기에 에코노베이션에 들어오게된 최명선입니다.  
8 +  
9 + 제 MBTI는 ESFJ입니다. 예전부터 MBTI TEST를 반복해서 해봐도 ESFJ가 똑같이 나오더라고요. 저는 사람을 정말 좋아하고,  
친해지고 싶은 사람들에게는 항상 먼저 다가가는 성격입니다. 그래서 23기 분들에게도 제가 먼저 다가가서 친해지고 싶습니다!
```

saint6839 on Apr 29, 2022 [Member](#) ...  
명선 universe에서 헤어나올수 있는 사람은 없지..  
2yunseong on May 10, 2022 [Member](#) ...  
○ㅈ○ㅈ

Write Preview

명선님은 3대 500 아닌가요??

Attach files by dragging & dropping, selecting or pasting them.

Comment

Resolve conversation

1. 리뷰를 작성합니다

2. 리뷰를 작성하셨다면 클릭!

## 02 다른 사람의 소개글 읽고 리뷰하기

모든 리뷰를 완료했다면 클릭!

The screenshot shows a GitHub pull request interface for a file named `myungsun.md`. The file contains a自我介绍 (self-introduction) in Korean. A red circle highlights the green "Finish your review" button in the top right corner of the header bar.

**File Content:**

```
@@ -0,0 +1,36 @@
1 + 🌟 한 줄로 간단하게 자신을 소개해주세요.
2 +
3 + 안녕하세요! 저는 현재 소프트웨어공학과 4학년에 재학중인 22기 최명선입니다.
4 +
5 + 💬 좀 더 자세하기 자신에 대해 소개해주세요.
6 +
7 + 안녕하세요: ) 저는 작년에 소프트웨어공학과로 편입을 하고 3학년 2학기에 에코노베이션에 들어오게된 최명선입니다.
8 +
9 + 제 MBTI는 ESFJ입니다. 예전부터 MBTI TEST를 반복해서 해봐도 ESFJ가 똑같이 나오더라고요. 저는 사람을 정말 좋아하고, 친해지고 싶은 사람들에게는 항상 먼저 다가가는 성격입니다. 그래서 23기 분들에게도 제가 먼저 다가가서 친해지고 싶습니다!
```

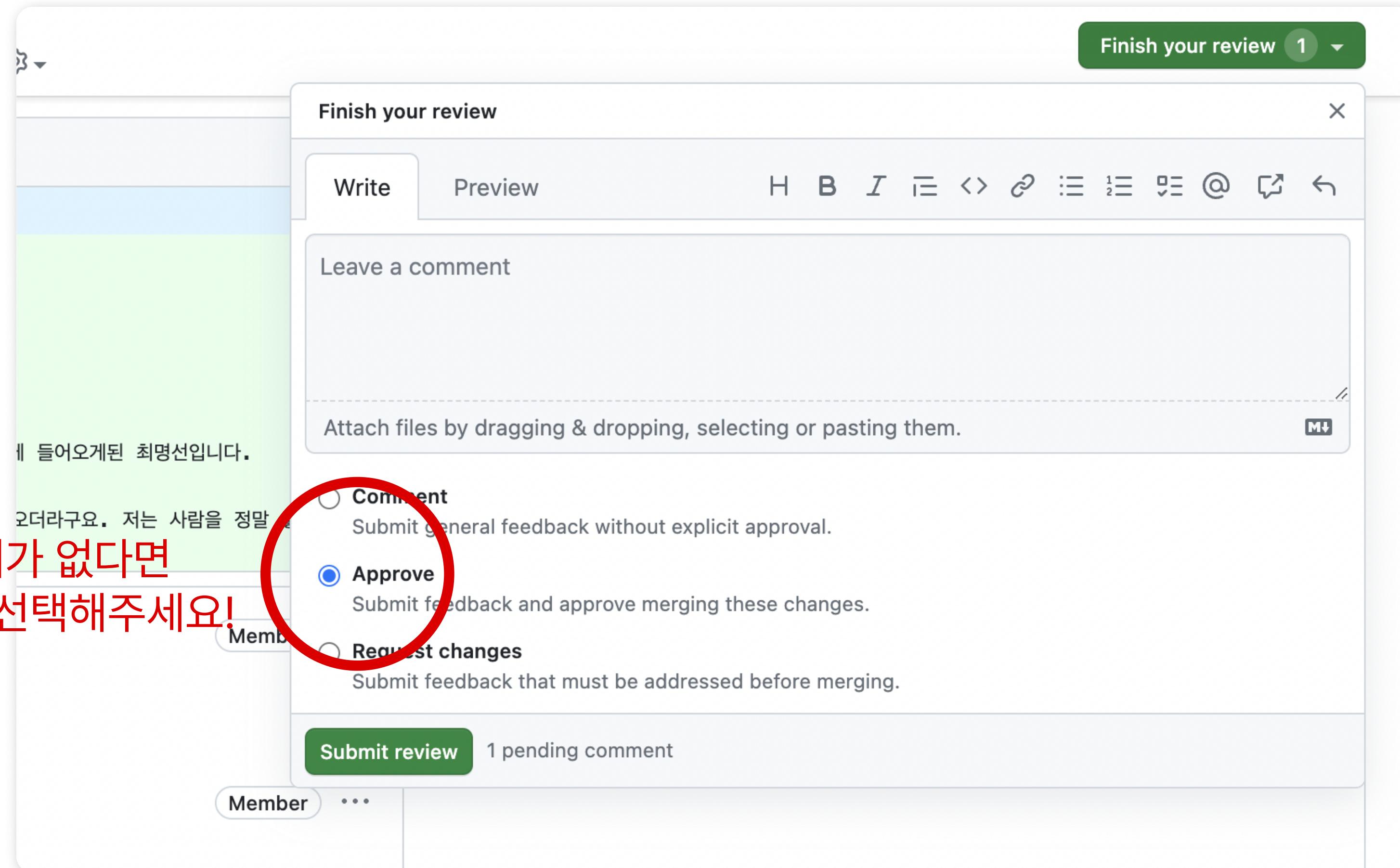
**Review Conversation:**

- saint6839** on Apr 29, 2022: 명선 universe에서 헤어나올수 있는 사람은 없지.. Member
- 2yunseong** on May 10, 2022: ○ㅈ○ㅈ Member
- capDoYeonLee** [Pending]: 명선님 잘생겼어요~ Member
- Reply...** Member

**Buttons:**

- Resolve conversation

## 02 다른 사람의 소개글 읽고 리뷰하기



PR에 별 문제가 없다면  
Approve를 선택해주세요!

## 참고 링크

- Let's git it started [Git Repository](#)
- 이론 가이드
- 실습 가이드
- Quick Guide

# **End of Document**