

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 알아보겟
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



Report.pptx



Report_수정.pptx



Report_수정_최종.pptx

...



최종_최종of최종_final은최종.pptx

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

Chapter1. Let's git it started!

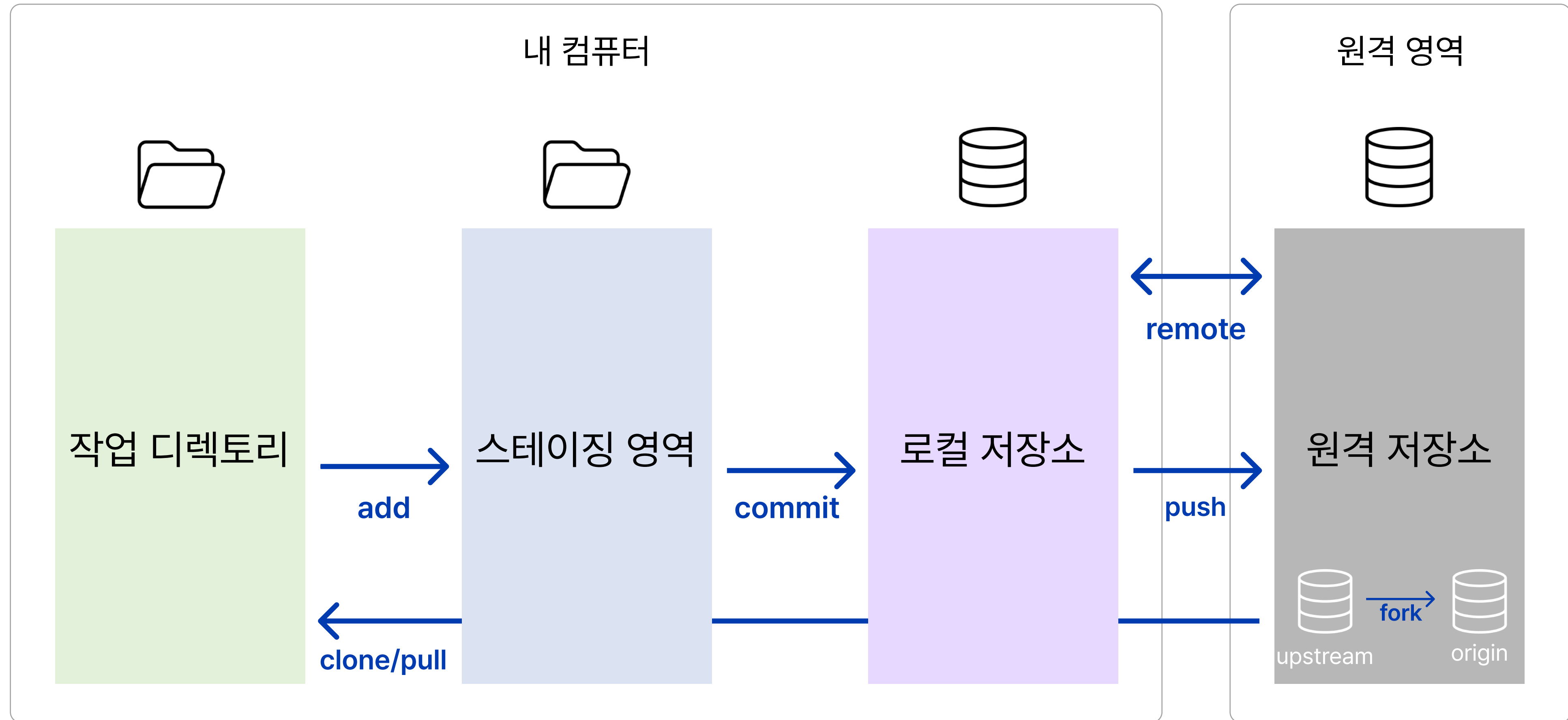
01 Git/GitHub는 무엇일까?



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**: 저장할 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 넵! 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 알아보킷

02 마크다운 문법이란?

03 Git 실습하기

01 우리 서로 더 알아보기

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

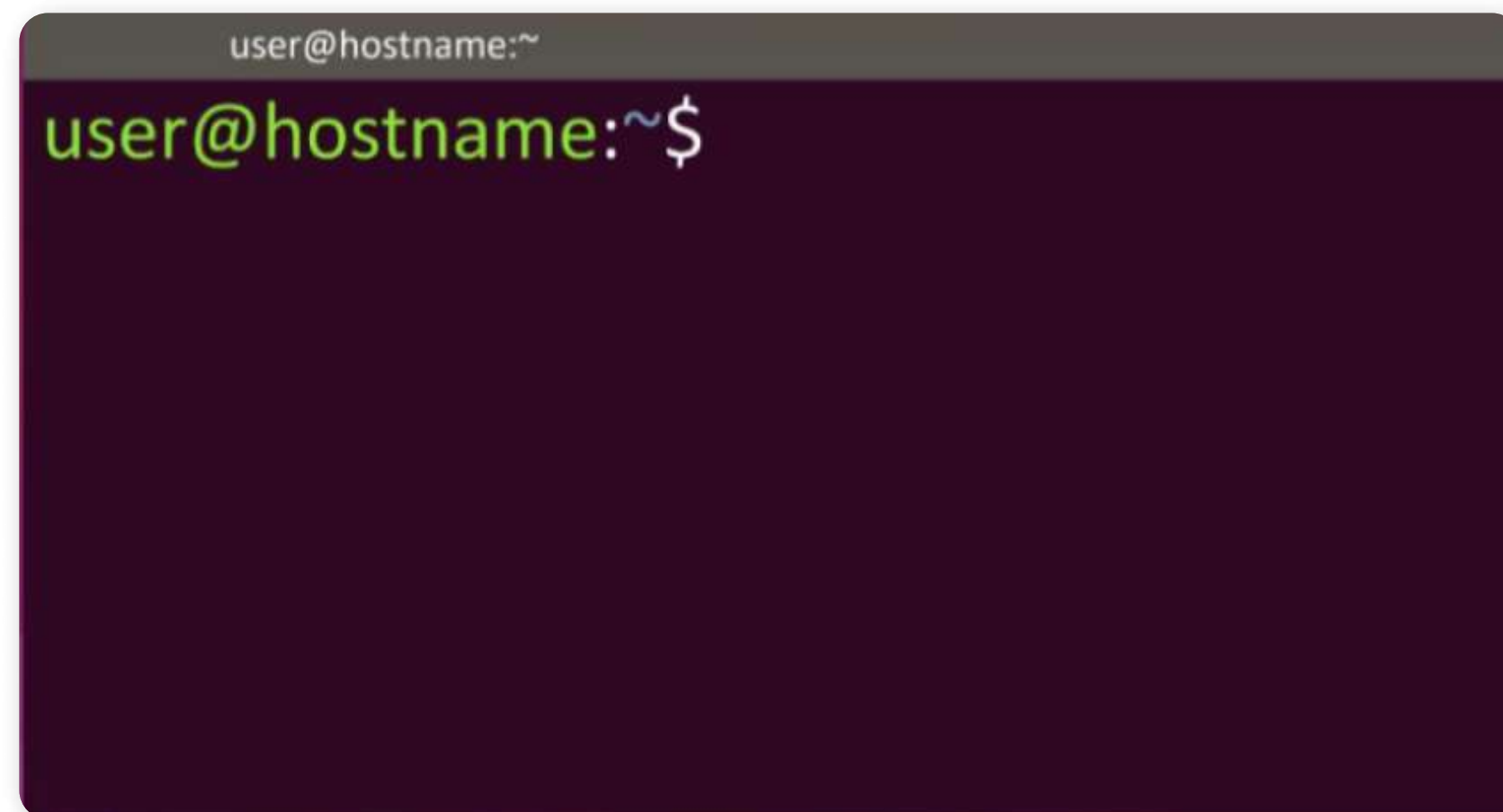
02 마크다운 문법이란?

- 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

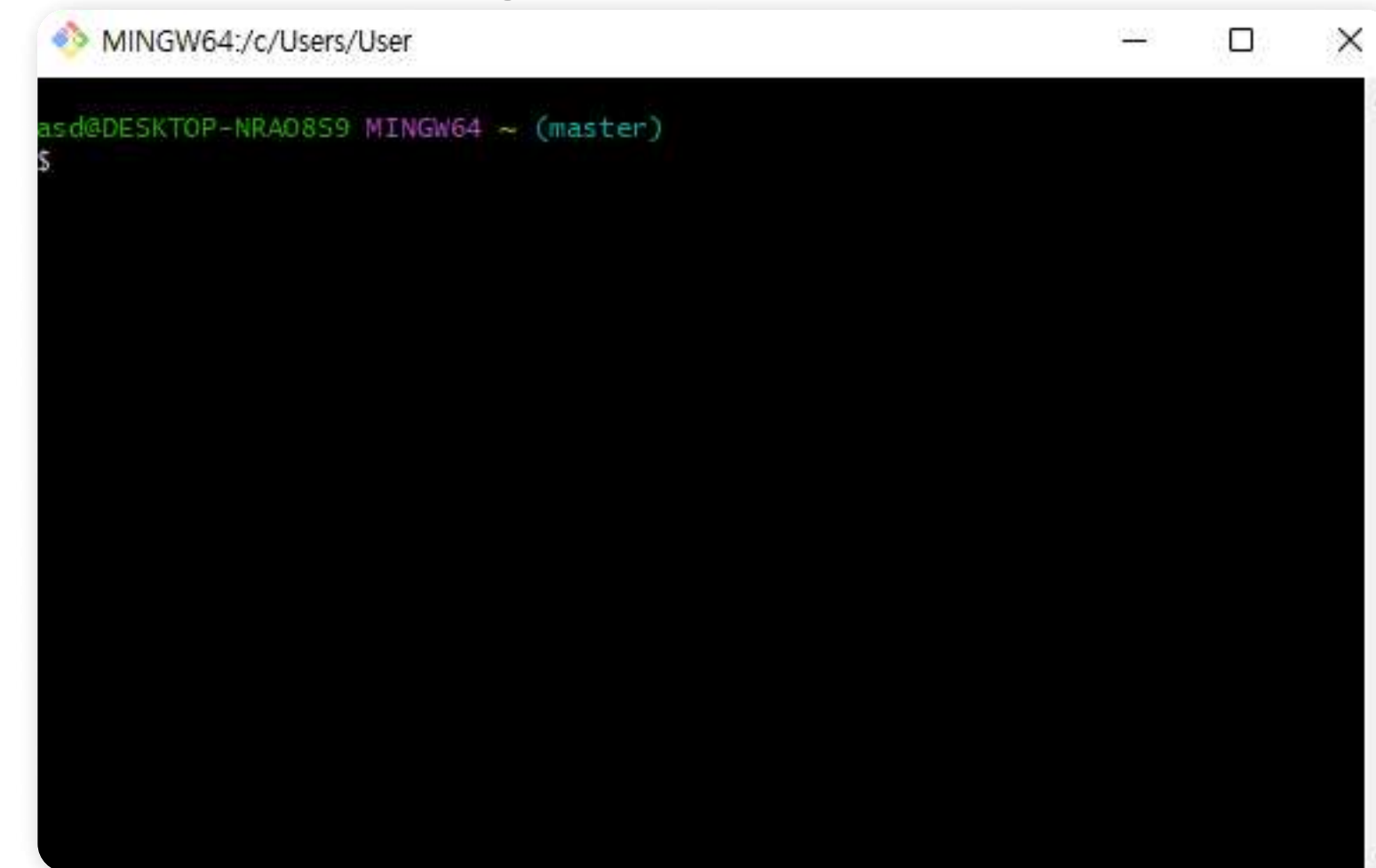
03 Git 실습하기

Step0. 명령어 창 띄우기

UNIX / terminal



Windows / git bash



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-NRA08S9 MINGW64 ~/Desktop/practice (master)
$ git init
Initialized empty Git repository in C:/Users/User/Desktop/practice/.git/

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

03 Git 실습하기

Step2. Git 저장소 생성 및 최초설정

💡 Git 사용이 처음인 경우 실시

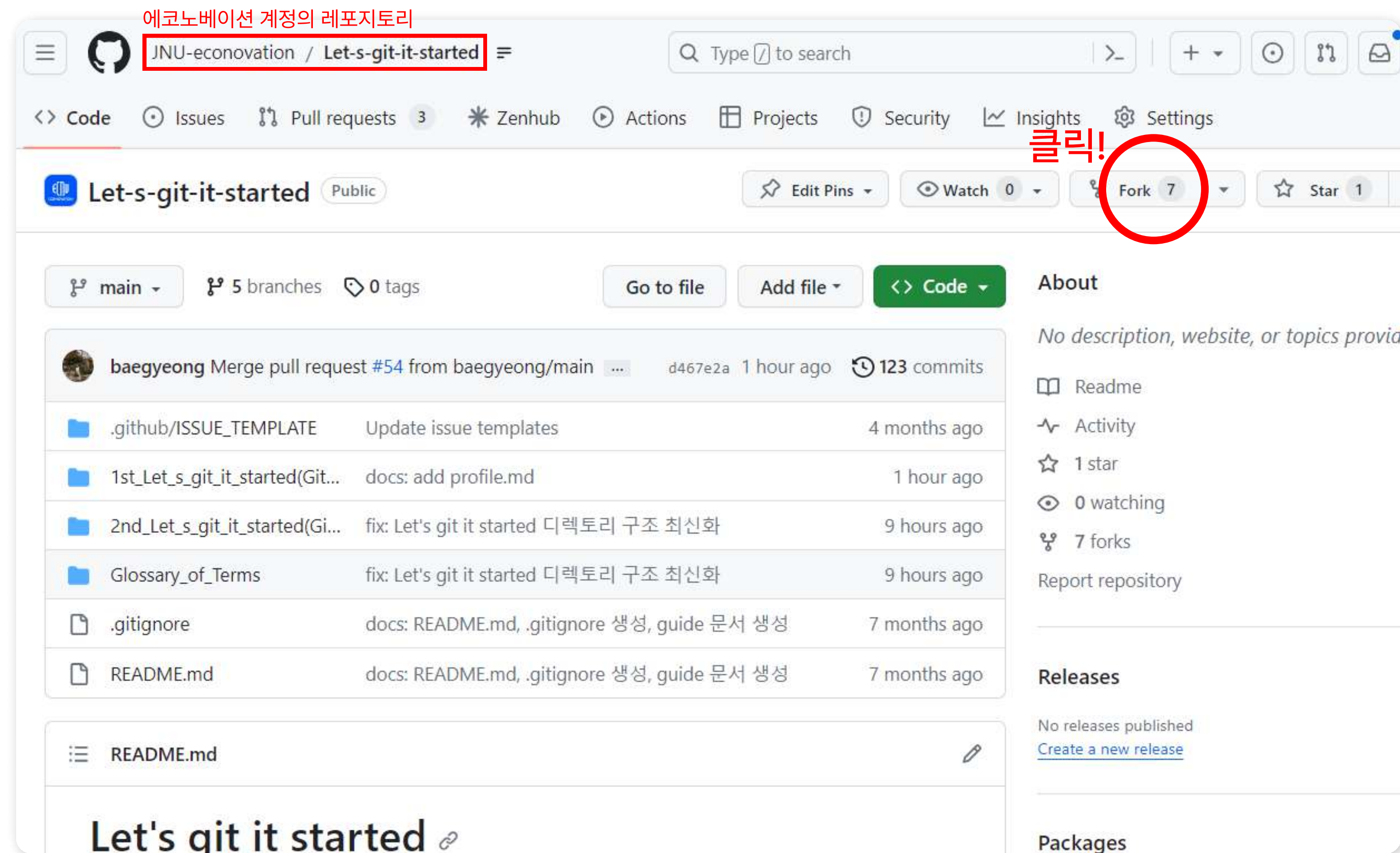


```
$ git config --global user.name "Your name"
$ git config --global user.email "Your email"
$ git config --global --list 제대로 설정이 되었는지 확인
```

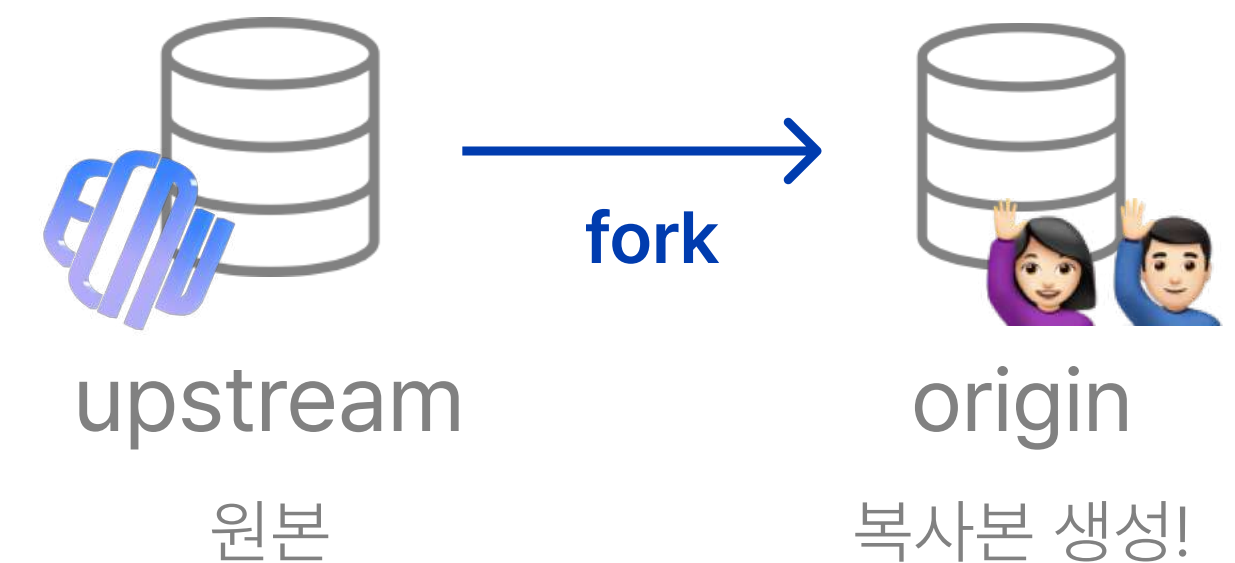
- 이름과 이메일 주소를 설정하는 이유
→ commit 할 때 사용자 이름과 이메일이 기록됨

03 Git 실습하기

Step3. 원본 저장소를 내 계정으로 복사하기 - fork



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

Let-s-git-it-started is available.

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.](#)

You are creating a fork in your personal account.

클릭!

Create fork

baegyeong / Let-s-git-it-started

Type to search

Code Pull requests Actions Zenhub Projects Security Insights Settings

Let-s-git-it-started

Public

Pin Watch 0 Fork 8 Star 0

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

main 1 branch 0 tags

Go to file Add file Code

About

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

Activity

0 stars

0 watching

8 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

ECONOVATION

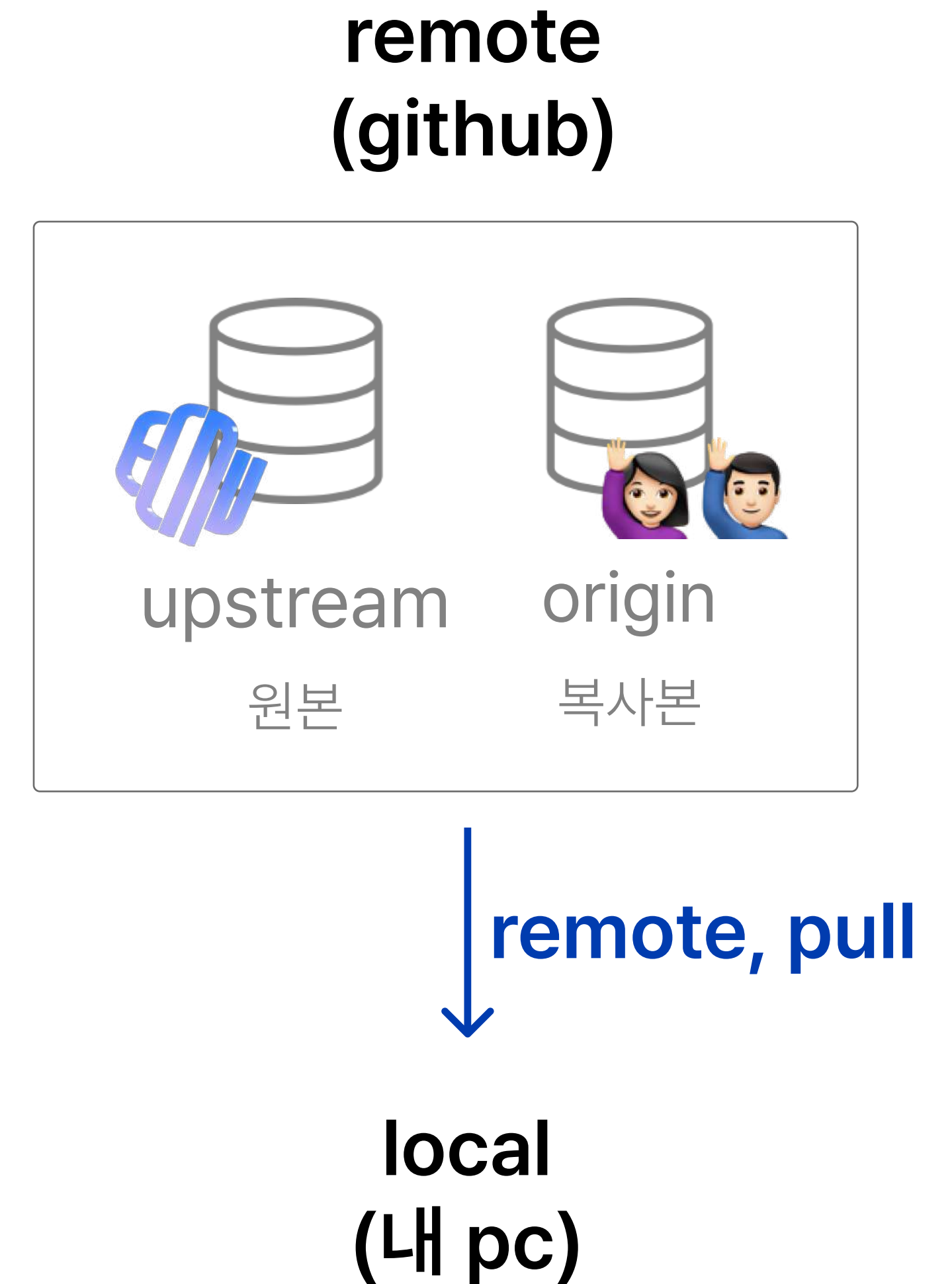
03 Git 실습하기

Step4. 자신의 local에 원격 저장소 등록하기

\$ git remote add [저장소 이름] [저장소 주소]

💡 실습을 위해 두 가지 저장소를 등록합니다.

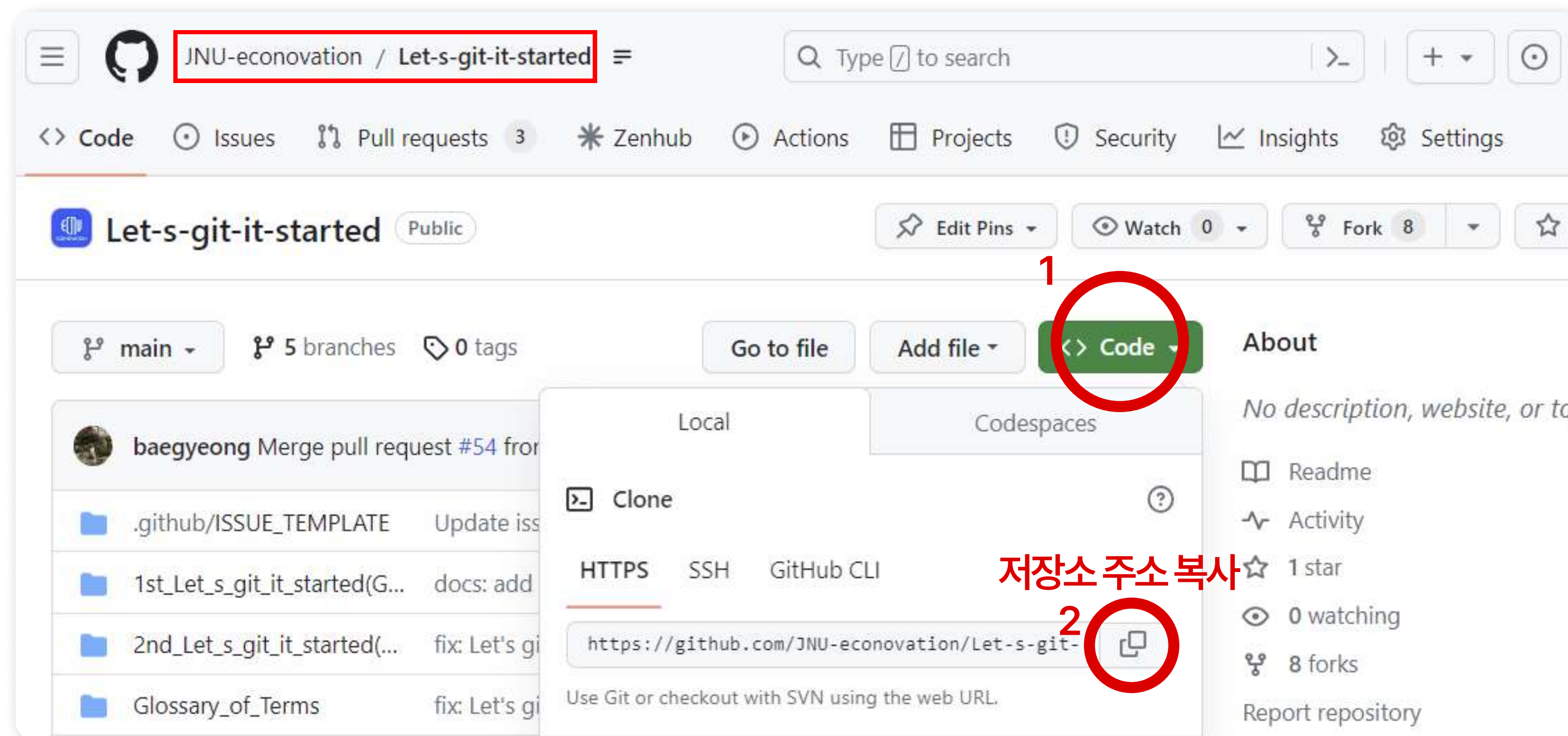
1. 에코노베이션 레포지토리(원본)
2. 나의 레포지토리 (fork한 복사본)



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

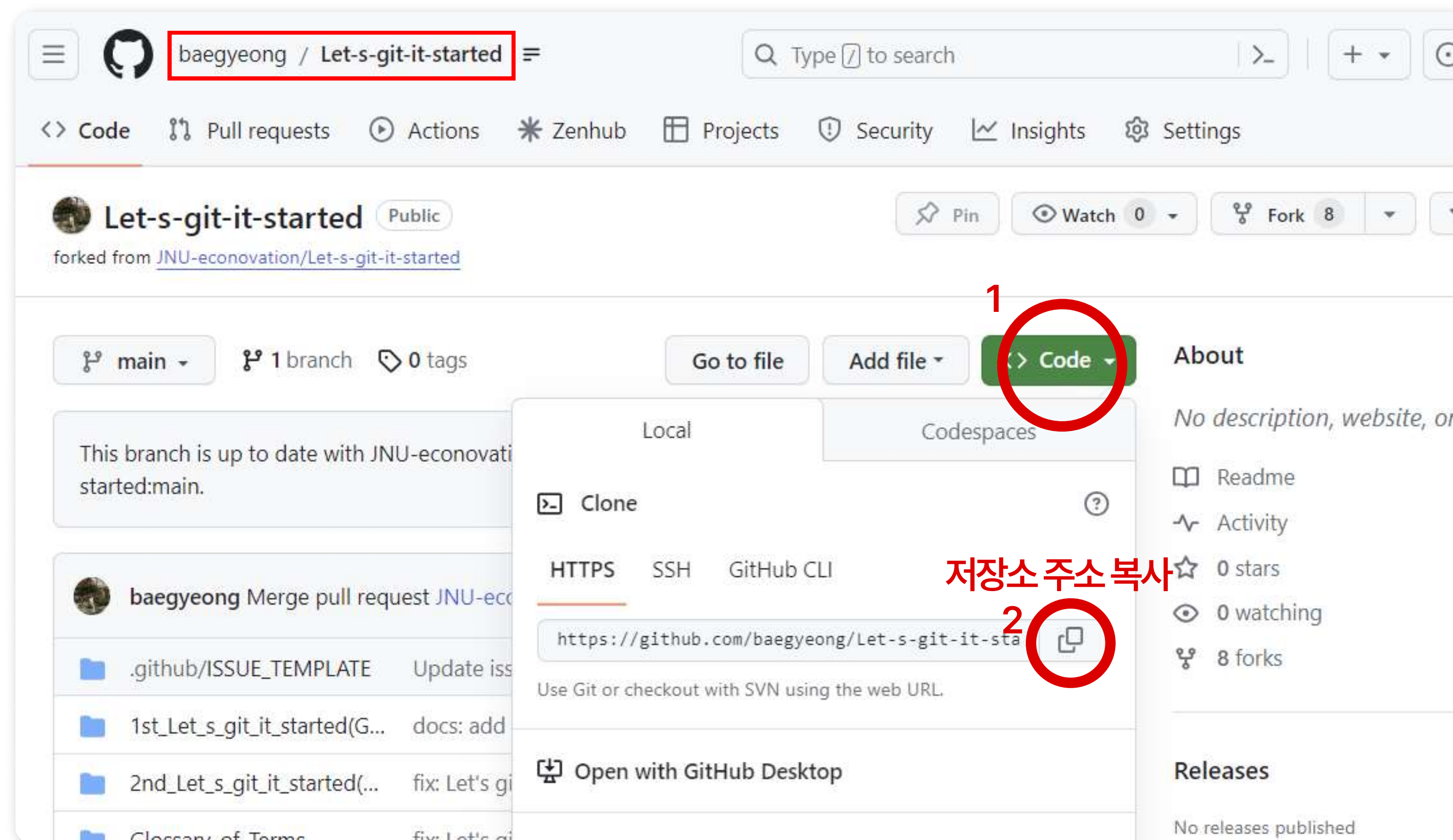
```
asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice (master)
$ git remote add upstream https://github.com/JNU-econovation/Let-s-git-it-started.git

asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice (master)
$ git remote -v
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-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ git remote add origin https://github.com/baegyeong/Let-s-git-it-started.git

asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice (master)
$ 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-NRA08S9 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-NRA08S9 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-NRA08S9 MINGW64 ~/Desktop/practice (master)
$ cd 2023-2/ 2023-2 폴더로 이동

asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ ls 2023-2 폴더에 어떤 파일이 있는지 확인
profile.md

asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ cp profile.md econo.md profile.md 파일을 [이름].md 파일로 복사

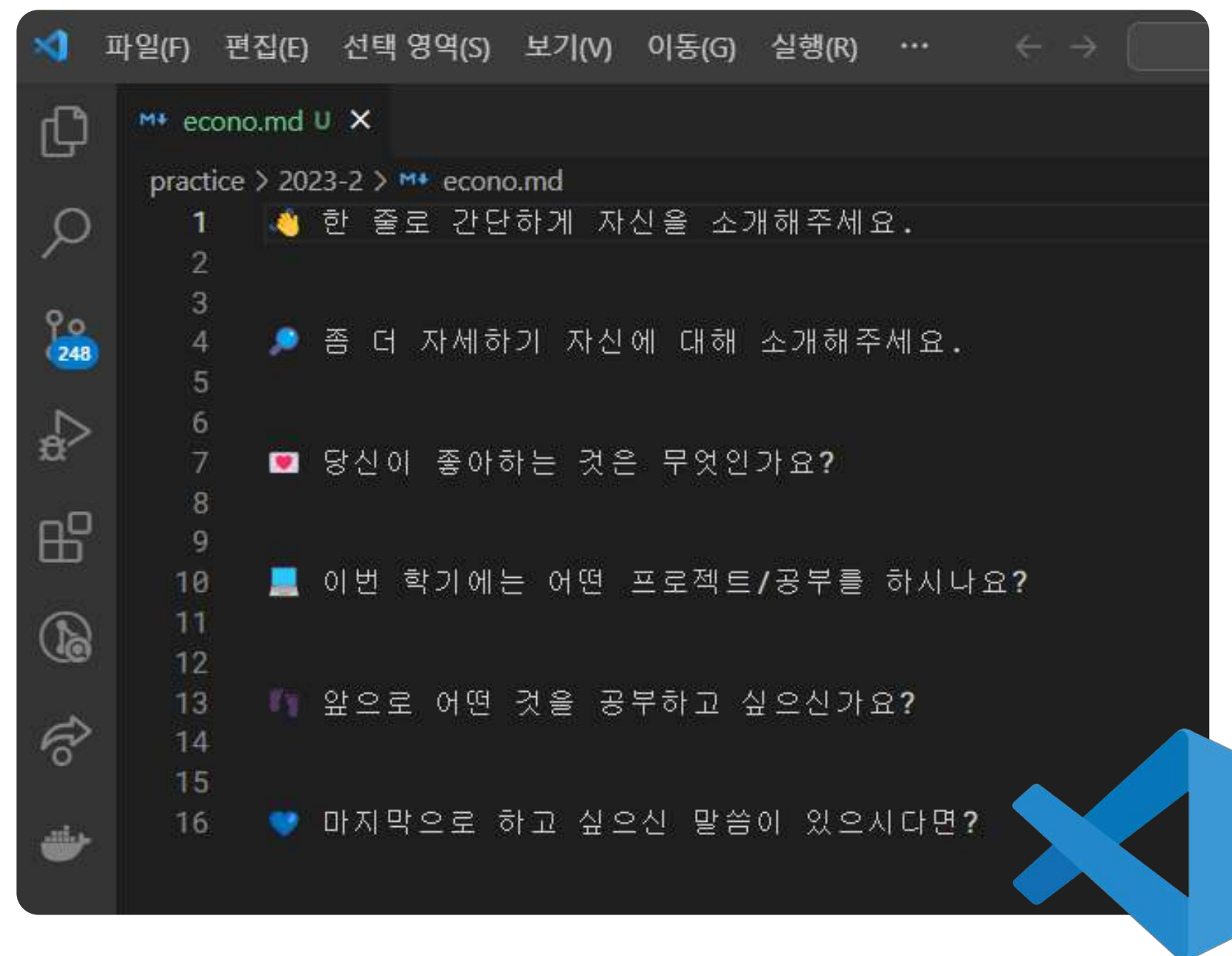
asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ ls [이름].md 파일이 생성된 것을 확인
econo.md profile.md
```

03 Git 실습하기

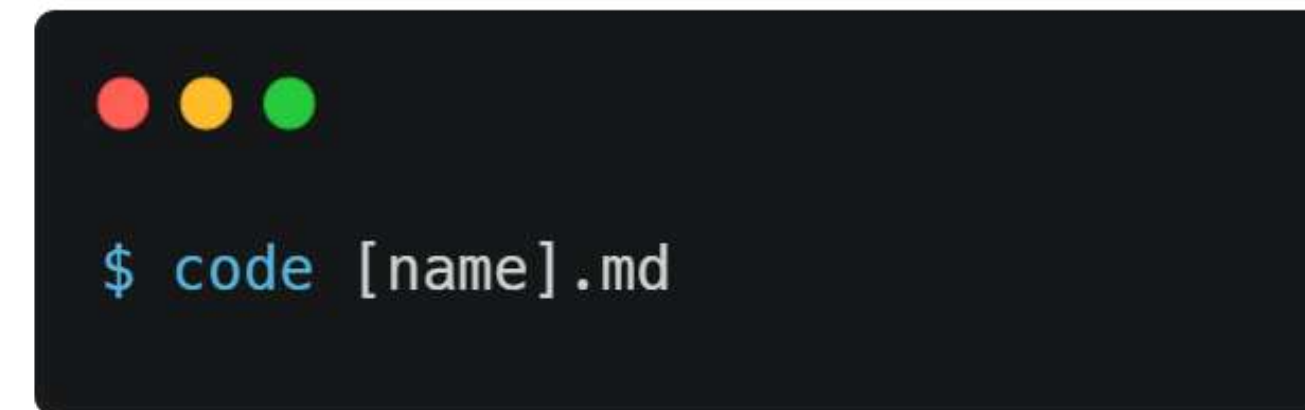
Step5. 준비해 온 자기소개로 글 수정하기 [🔗 Markdown 문법 정리](#)

💡 글 작성 에디터를 실행하는 다양한 방법

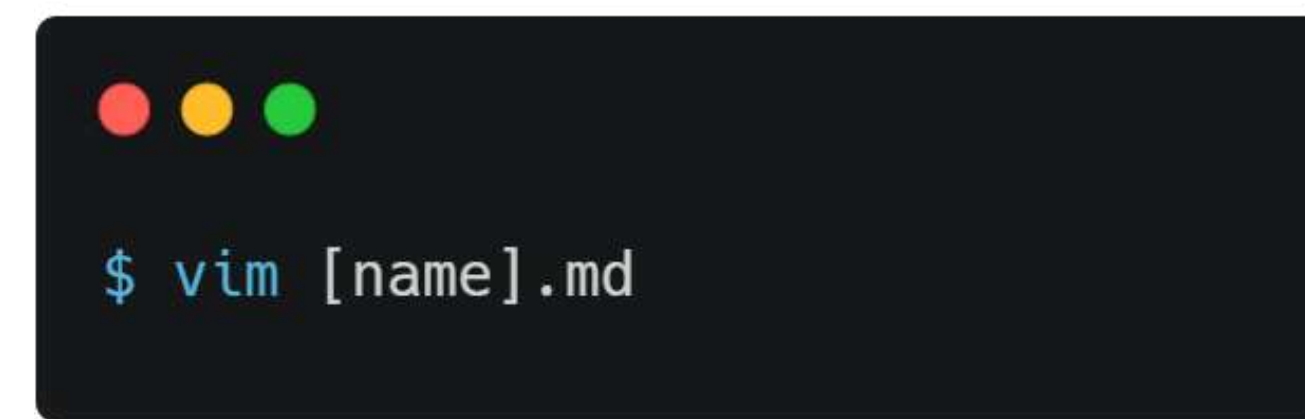
1. Visual Studio Code 직접 실행하기



2. 명령어로 Visual Studio Code 실행시키기



3. 터미널에서 작성하기 - Vim

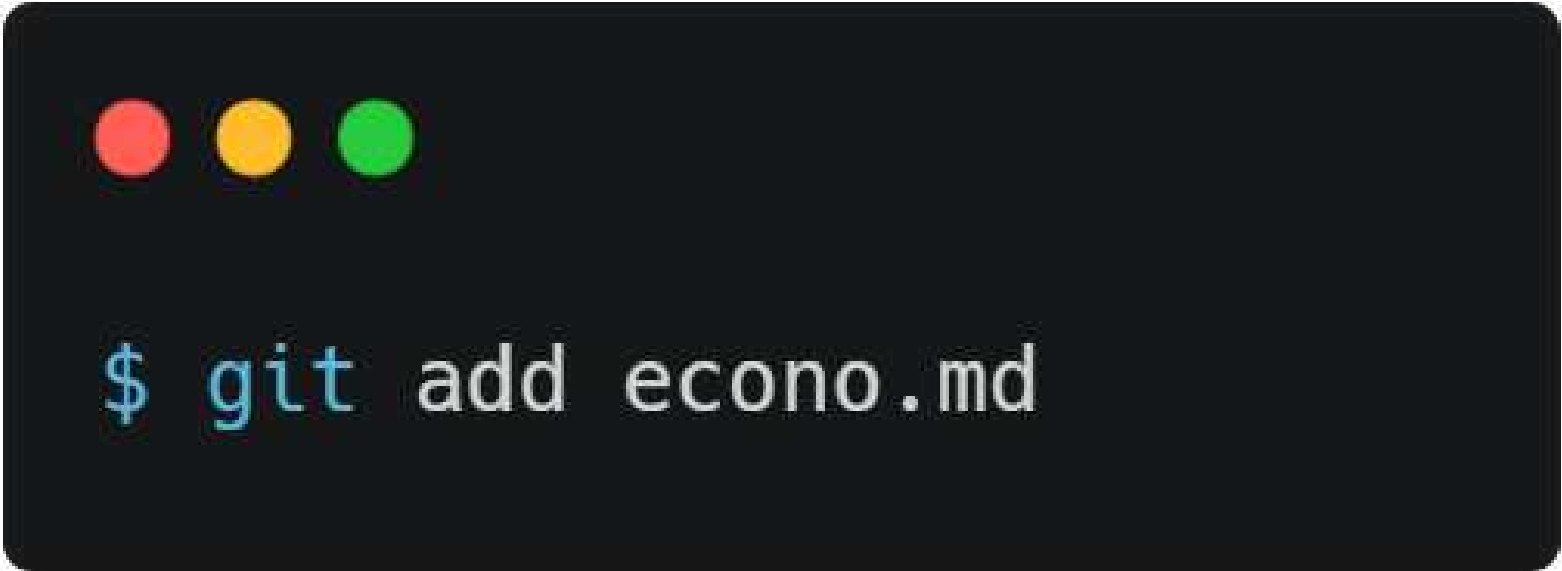


03 Git 실습하기

Step6. 그럼 이제! 글을 올려보자

\$ git add filename

자신이 clone한 local 저장소의 변경사항을 GitHub에 반영하겠다.



```
$ 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-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ ls
econo.md  profile.md
현재 폴더의 파일 목록 확인

asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git add econo.md
스테이징 영역에 파일 추가하기

asd@DESKTOP-NRA08S9 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"

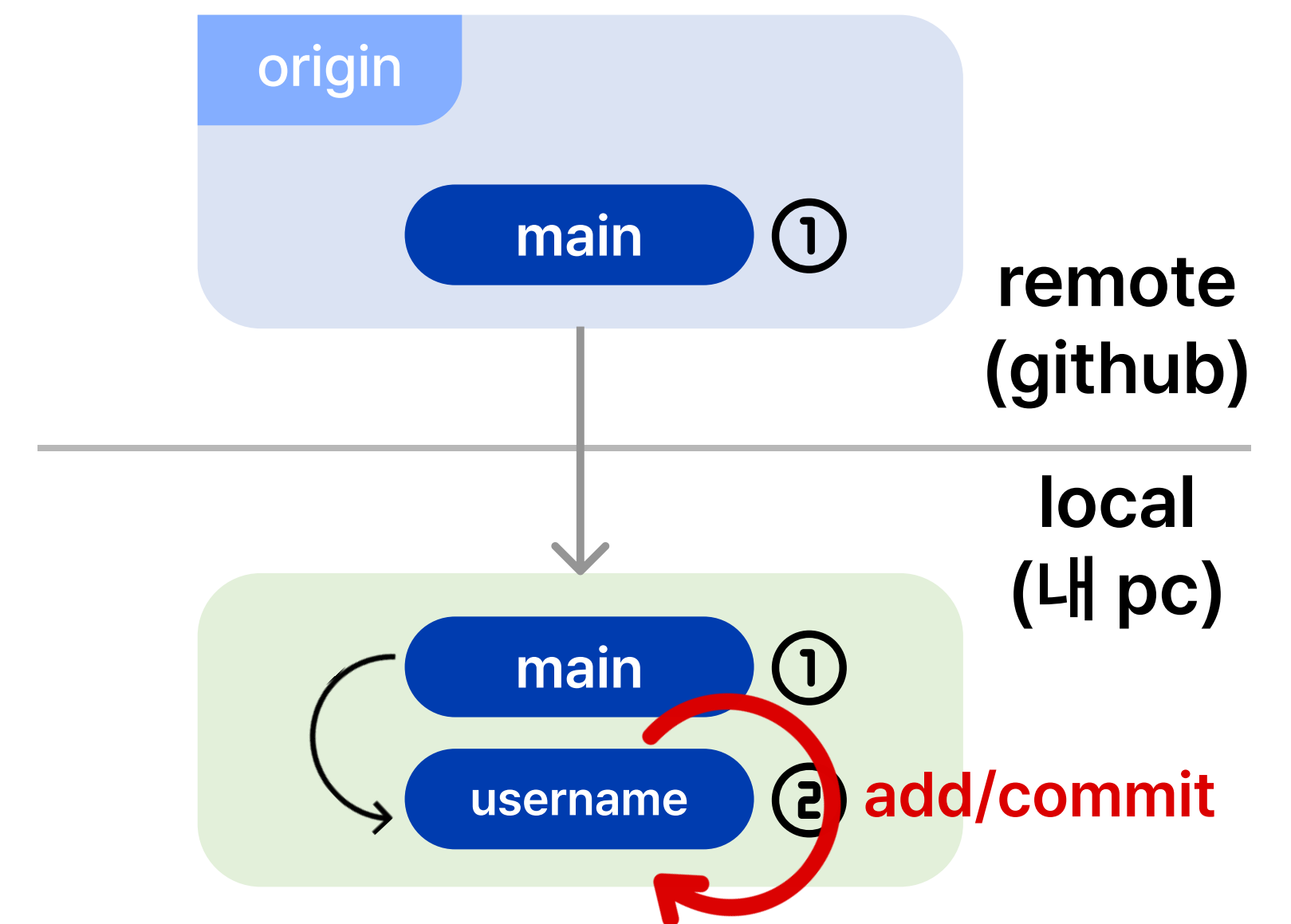
변경사항에 대해서 "이러한 내용"이 담긴 메시지를 남기겠다.

💡 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-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git commit -m "docs: add econo.md"
[master 89ee9a6] docs: add econo.md
1 file changed, 16 insertions(+)
create mode 100644 2023-2/econo.md

asd@DESKTOP-NRA08S9 MINGW64 ~/Desktop/practice/2023-2 (master)
$ git status
On branch master
nothing to commit, working tree clean
```

커밋 메시지 작성

스테이징 영역 -> 로컬 저장소 이동
스테이징 영역은 비워지게 됨

03 Git 실습하기

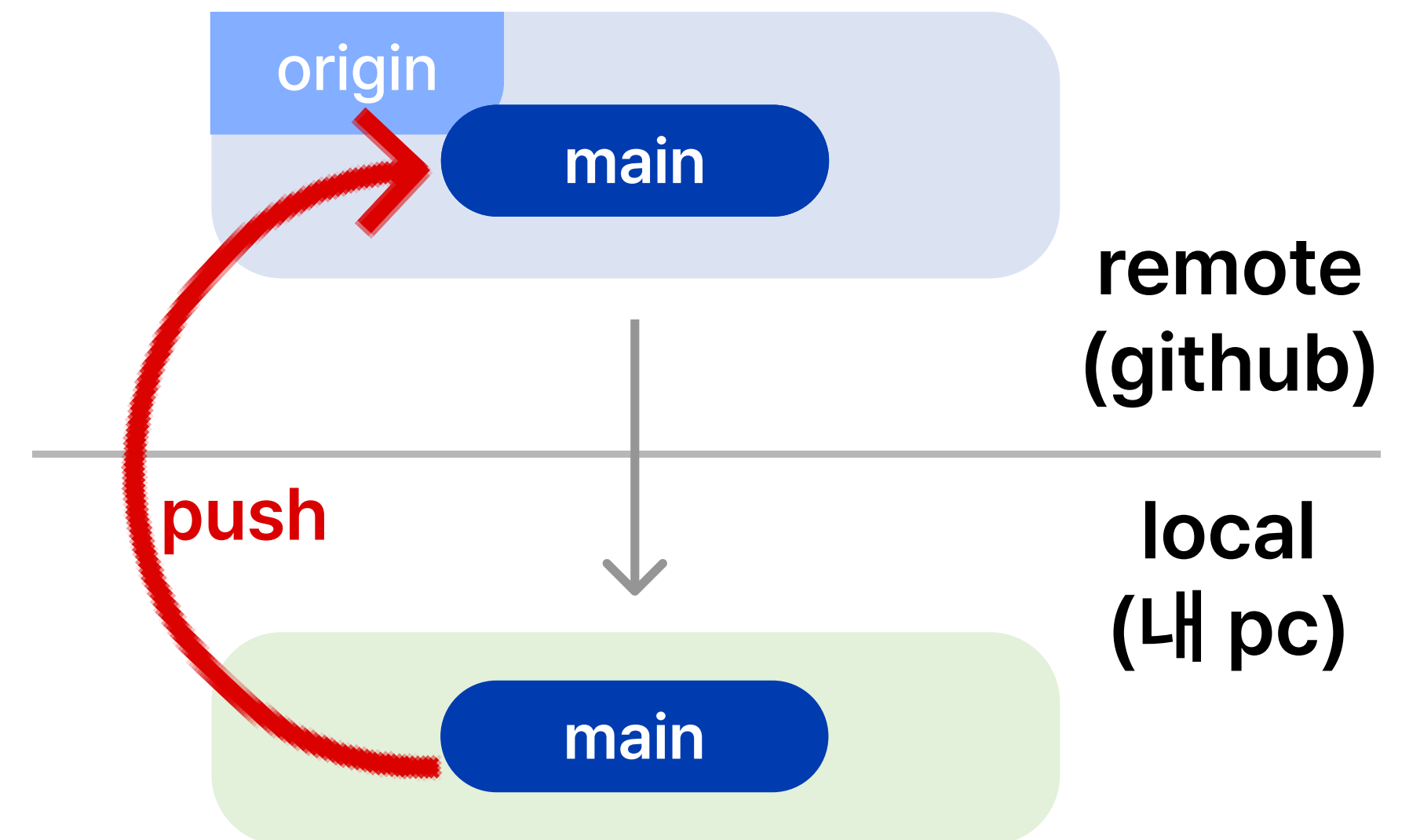
Step6. 그럼 이제! 글을 올려보자

\$ 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 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

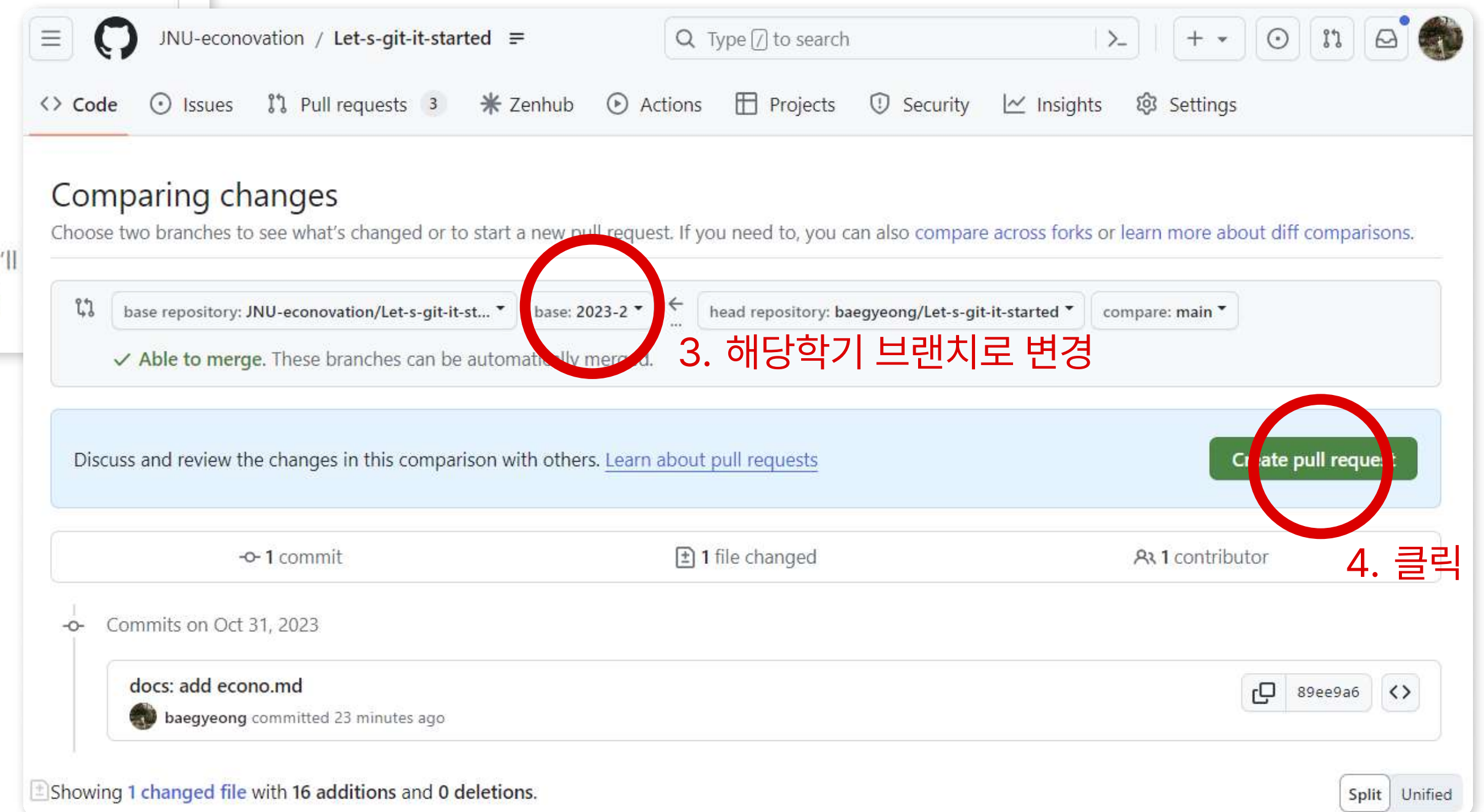
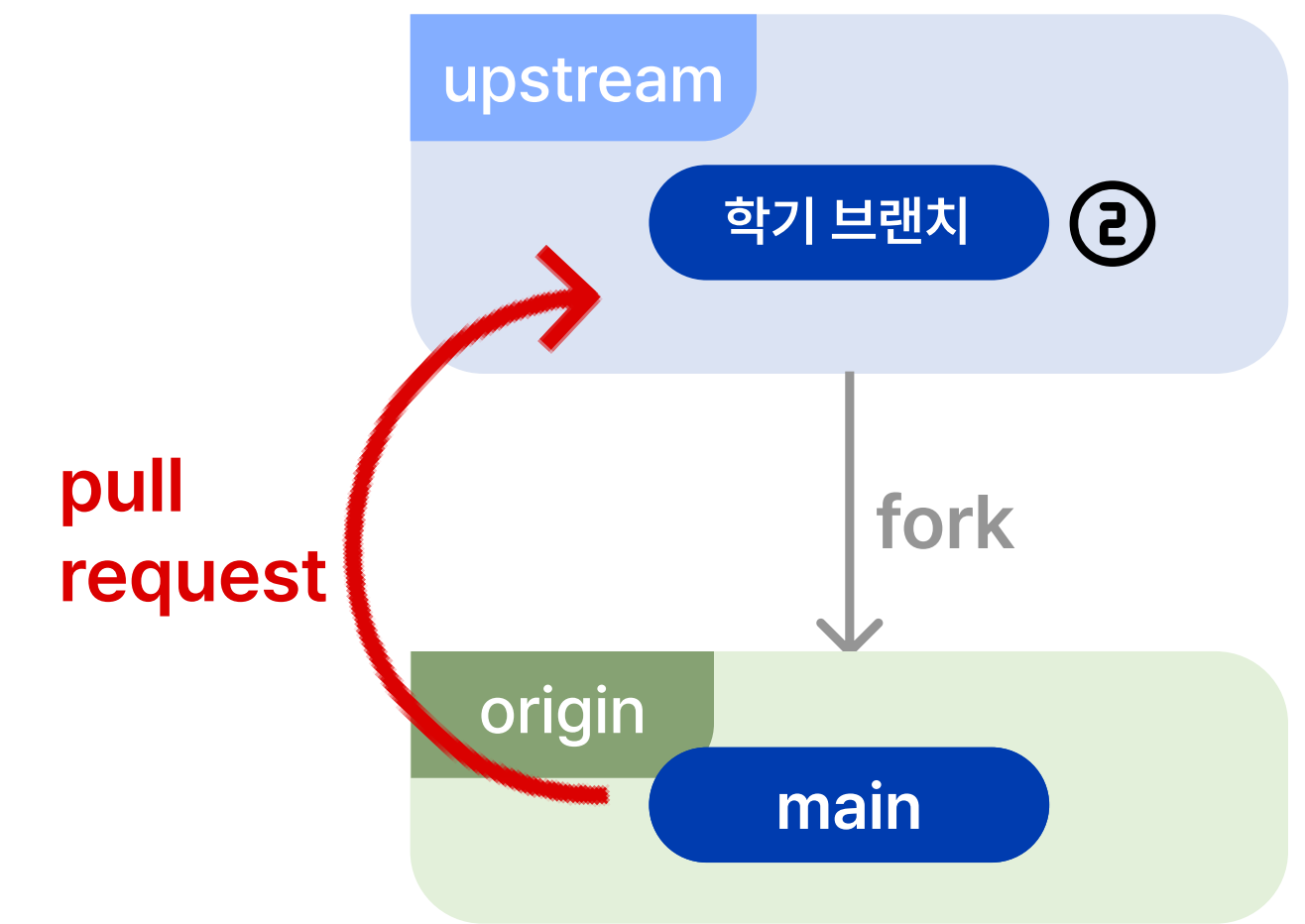
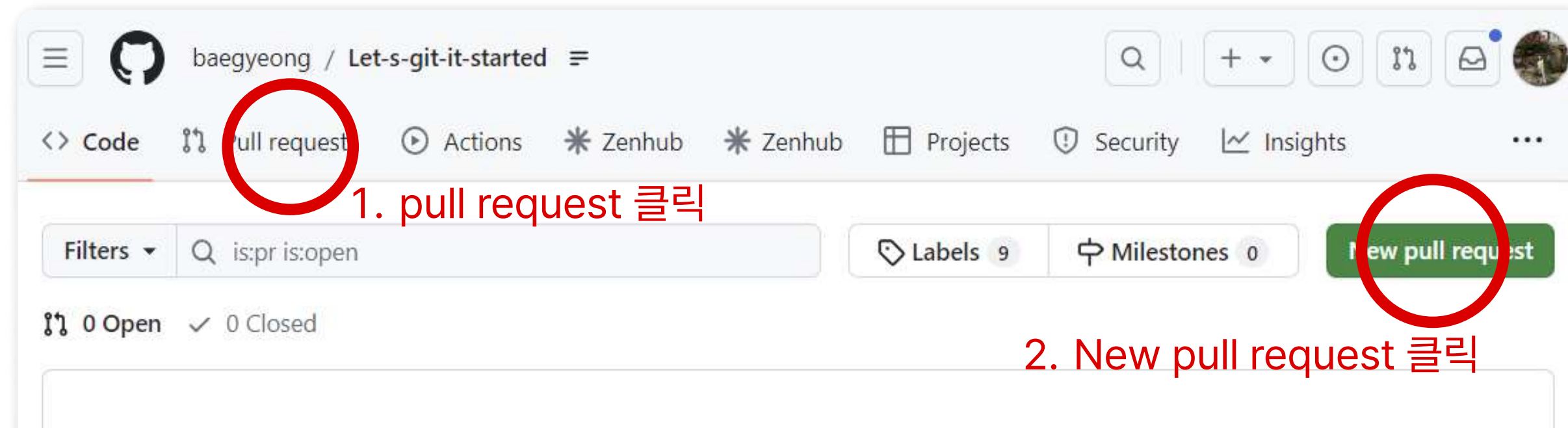
asd@DESKTOP-NRAO8S9 MINGW64 ~/Desktop/practice/2023-2 (main)
```

Chapter3. Feedback/Review

01 Pull Request 작성하기

02 다른 사람 소개 글 읽고 리뷰하기

01 Pull Request 작성하기



01 Pull Request 작성하기

2. 제목과 내용 작성
제목: 이름(아이디) - 자기소개
내용: 자유롭게 구성

base repository: JNU-econovation/Let-s-git-it-st... base: 2023-2 head repository: baegyeong/Let-s-git-it-started compare: main

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

에코노(econo) - 자기소개

Write Preview H B I ≡ <> 🔗 ≡ ≡ ≡ @ ↗ ↶ ↷

안녕하세요 에코노입니다! 잘 부탁드립니다~

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

☒ Allow edits by maintainers ? Create pull request

Pipeline
No Workspace yet - Create One

Reviewers
No reviews

Assignees
No one—assign yourself

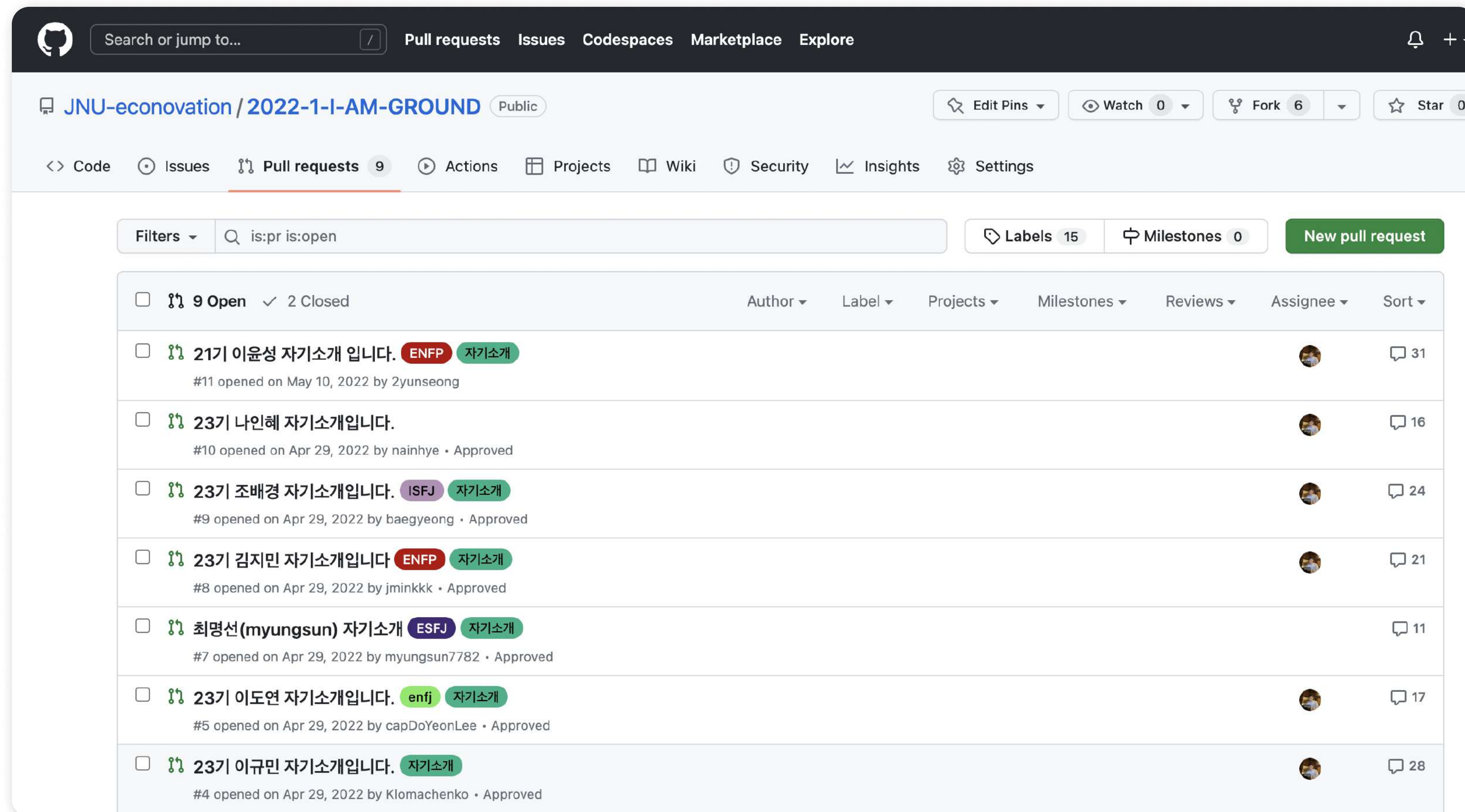
Labels
25기 자기소개

Projects
None yet

1. pr을 보내는 브랜치 확인
내 레포 'main' -> 원본 레포 '해당 학기 브랜치'

3. 라벨 지정
(기수, mbti, 자기소개)

02 다른 사람의 소개글 읽고 리뷰하기



The screenshot shows the GitHub interface for the repository `JNU-econovation / 2022-1-I-AM-GROUND`. The `Pull requests` tab is selected, displaying a list of 9 open pull requests. The interface includes a search bar, navigation links (Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings), and filters for the pull requests.

Filters: `is:pr is:open`

Labels: 15 **Milestones:** 0 **New pull request**

<input type="checkbox"/>	9 Open ✓ 2 Closed	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>	21기 이윤성 자기소개입니다. ENFP 자기소개 #11 opened on May 10, 2022 by 2yunseong							31
<input type="checkbox"/>	23기 나인혜 자기소개입니다. #10 opened on Apr 29, 2022 by nainhye • Approved							16
<input type="checkbox"/>	23기 조배경 자기소개입니다. ISFJ 자기소개 #9 opened on Apr 29, 2022 by baegyeong • Approved							24
<input type="checkbox"/>	23기 김지민 자기소개입니다 ENFP 자기소개 #8 opened on Apr 29, 2022 by jminkkk • Approved							21
<input type="checkbox"/>	최명선(myungsun) 자기소개 ESFJ 자기소개 #7 opened on Apr 29, 2022 by myungsun7782 • Approved							11
<input type="checkbox"/>	23기 이도연 자기소개입니다. enfj 자기소개 #5 opened on Apr 29, 2022 by capDoYeonLee • Approved							17
<input type="checkbox"/>	23기 이규민 자기소개입니다. 자기소개 #4 opened on Apr 29, 2022 by Klomachenko • Approved							28

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

myungsun7782 added 자기소개 ESFJ labels on Apr 29, 2022

Reviewers

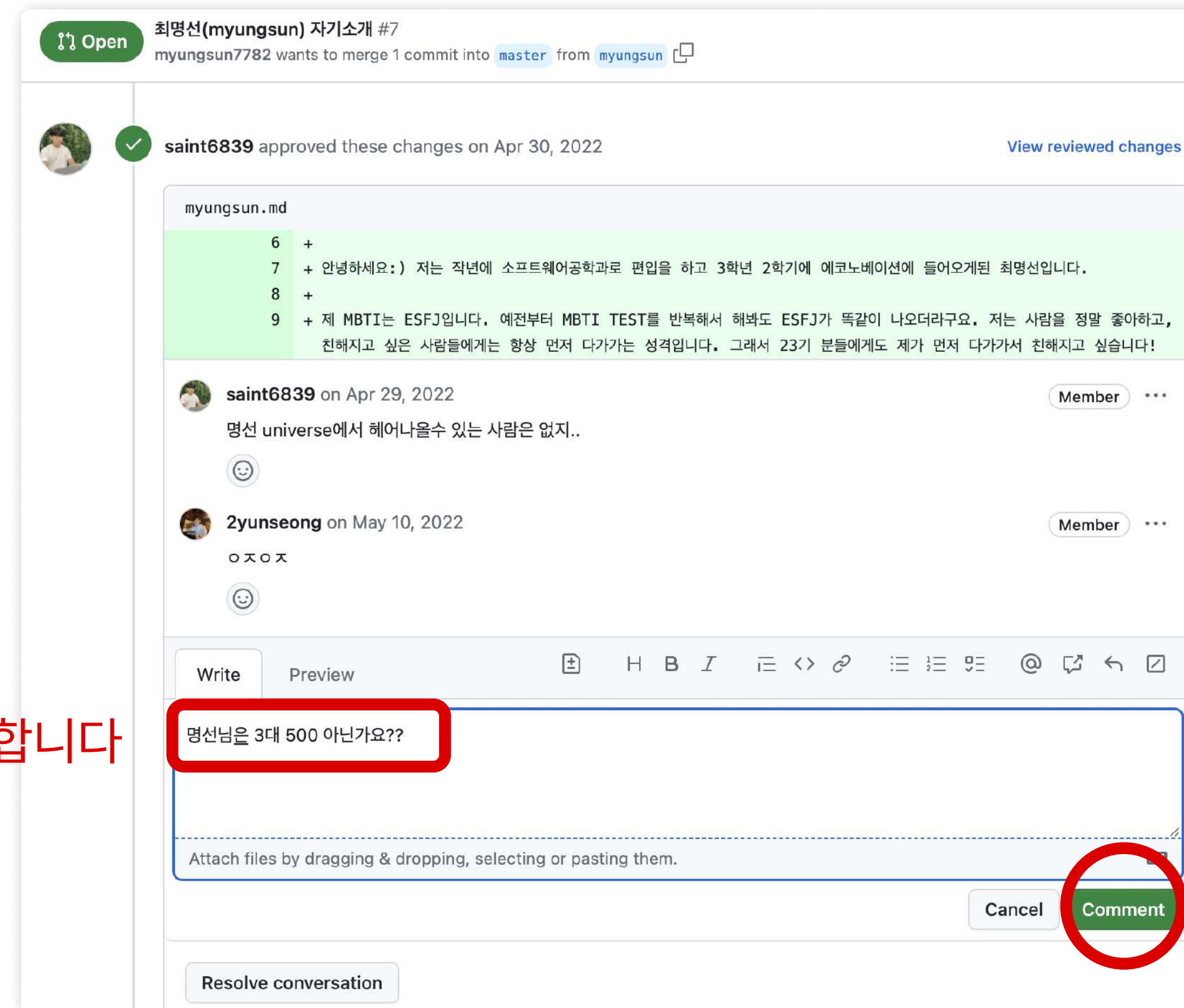
- mywnajs
- 2yunseo
- saint683

Assignees

No one—assigned

02 다른 사람의 소개글 읽고 리뷰하기

1. 리뷰를 작성합니다



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 헤어날수 있는 사람은 없지..
😊

2yunseong on May 10, 2022 Member ...
ㅇㅈㅇㅈ
😊

Write Preview

명선님은 3대 500 아닌가요??

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

Cancel **Comment**

Resolve conversation

2. 리뷰를 작성하셨다면 클릭!

02 다른 사람의 소개글 읽고 리뷰하기

모든 리뷰를 완료했다면 클릭!

The screenshot shows a GitHub pull request review interface. At the top, there's a header with 'Open', '최명선(myungsun) 자기소개 #7', and navigation links like 'Changes from all commits', 'File filter', 'Conversations', and 'Jump to'. A green button 'Finish your review' with a dropdown arrow is circled in red. Below the header, the file '36 myungsun.md' is shown. The file content is a Korean self-introduction. Below the file content, there are three reviews from other users: 'saint6839', '2yunseong', and 'capDoYeonLee'. Each review includes a user profile picture, name, date, and a 'Member' badge. The reviews contain text and smiley face emojis. At the bottom, there's a 'Reply...' input field and a 'Resolve conversation' button.

02 다른 사람의 소개글 읽고 리뷰하기

PR에 별 문제가 없다면
Approve를 선택해주세요!

Finish your review 1

Finish your review

Write Preview

Leave a comment

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

☐ Comment
Submit general feedback without explicit approval.

☒ **Approve**
Submit feedback and approve merging these changes.

☐ Request changes
Submit feedback that must be addressed before merging.

Submit review 1 pending comment

참고 링크

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

End of Document