

# Github를 통한 협업의 기본 배우기 with SourceTree

---



¶ Sourcetree



# Github는 무엇일까?

---

소프트웨어 개발 프로젝트를 위한 코드 관리 서비스

# SourceTree는 무엇일까?

---

Git을 더 쉽게 사용하기 위한 툴

# 사용방법 - Fork

The screenshot shows a GitHub repository page for 'sejonaqinterface / Study-Git'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. On the right side of the header are buttons for Pin, Watch (0), Fork (0) (which is highlighted with a red box), Star (0), and a profile icon.

The main repository page displays the following information:

- Code tab is selected.
- Branches: main (selected), 1 branch, 0 tags.
- Commits: jkey20 SourceTree 설치 (84f1f71, 3 hours ago, 3 commits).
- Files: README.md (Update README.md, 5 hours ago) and SourceTree 설치.md (SourceTree 설치, 3 hours ago).
- Code navigation buttons: Go to file, Add file, Code (dropdown).
- About section:
  - Owner: 깃허브 스터디
  - Readme
  - 1 star
  - 0 watching
  - 0 forks
- Releases: No releases published. Create a new release.
- Packages: No packages published. Publish your first package.

# 사용방법 - Fork

A screenshot of a GitHub repository page for 'sejonginterface / Study-Git'. The repository is public and contains 1 branch and 0 tags. A modal window titled 'Fork Study-Git' is open, asking 'Where should we fork Study-Git?'. The 'jkey20' option is selected and highlighted with an orange border. The right sidebar shows repository details: 'About' (깃허브 스터디), 'Code' (Commits, Readme, 2 stars, 0 watching, 0 forks), 'Releases' (No releases published, Create a new release), and 'Packages'.

Search or jump to...

Pull requests Issues Marketplace Explore

sejonginterface / Study-Git Public

Code Issues Pull requests Actions Projects

main 1 branch 0 tags

jkey20 Git bash 설치

Git bash 설치.md Git bash 설치

README.md Update README

SourceTree 설치.md SourceTree 설치

README.md

**Study-Git**

깃허브 스터디

Fork Study-Git

Where should we fork Study-Git?

jkey20

About

깃허브 스터디

Commits

Readme

2 stars

0 watching

0 forks

Code

Commits

Readme

2 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

# 사용방법 - Fork

The screenshot shows a GitHub repository page for a forked repository named `jkey20/Study-Git`. The repository is public and was forked from `sejonginterface/Study-Git`.

**Header:** Search or jump to... / Pull requests Issues Marketplace Explore

**Repository Information:** jkey20 / Study-Git (Public) forked from sejonginterface/Study-Git

**Actions:** Pin Watch 0 Fork 1 Star 0

**Navigation:** Code (selected), Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

**Code Overview:** main branch, 1 branch, 0 tags

This branch is up to date with sejonginterface/Study-Git:main.

**Commits:**

Author	Commit Message	Date	Commits
jkey20	Git bash 설치	3de77d8 5 minutes ago	4 commits
	Git bash 설치.md		5 minutes ago
	Git bash 설치		
	README.md		6 hours ago
	Update README.md		
	SourceTree 설치.md		4 hours ago
	SourceTree 설치		

**README.md Content:**

```
Study-Git
```

**About:**

깃허브 스터디

- Readme
- 0 stars
- 0 watching
- 1 fork

**Releases:**

No releases published

[Create a new release](#)

**Packages:**

No packages published

[Publish your first package](#)

# Fork

---

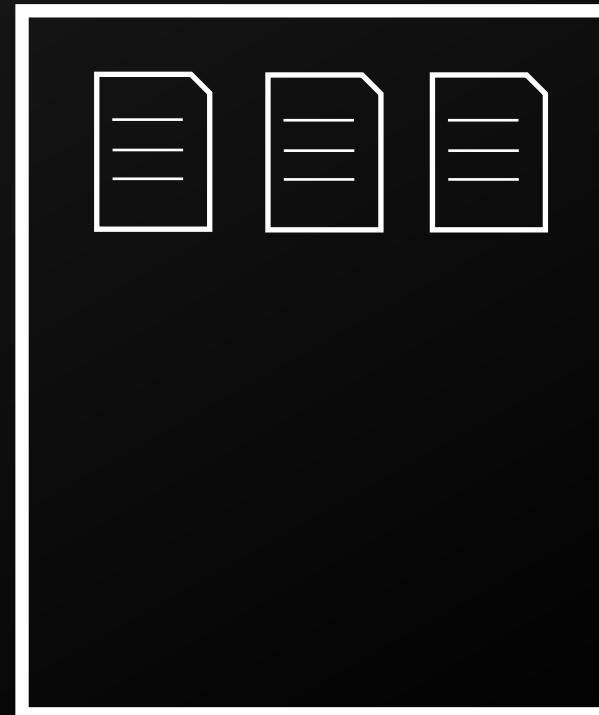
Fork란?

- 협업 repo를 복사해서 개인 repo를 만듬

협업 repository(sejonginterface/Study-Git)



개인 repository([username]/Study-Git)



Fork  
→

# 사용방법 - Clone

The screenshot shows a GitHub repository page for [jkey20 / Study-Git](#). The repository is public and was forked from [sejonginterface/Study-Git](#). The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. On the right side, there are buttons for Pin, Watch (0), Fork (1), and Star (0). Below the navigation, there are tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Code tab is currently selected.

Key elements visible on the page include:

- Branches and Tags:** main (selected), 1 branch, 0 tags.
- Commit History:** A recent commit by jkey20 titled "Git bash 설치" was pushed 11 minutes ago. It includes four commits and a link to the commit details (3de77d8).
- Files:** Git bash 설치.md, README.md, and SourceTree 설치.md.
- README.md Content Preview:** The content of the README.md file is displayed, featuring the title "Study-Git" and the text "기하브 스터디".
- Code Dropdown:** A green button labeled "Code" with a dropdown arrow is highlighted with a red box.
- About Section:** Contains information about the repository, including its name ("깃허브 스터디"), a Readme file, 0 stars, 0 watching, and 1 fork.
- Releases Section:** Shows that no releases have been published and provides a link to "Create a new release".
- Packages Section:** Shows that no packages have been published and provides a link to "Publish your first package".

# 사용방법 - Clone

Searched or jump to... / Pull requests Issues Marketplace Explore

jkey20 / Study-Git Public forked from sejonginterface/Study-Git

Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

This branch is up to date with sejonginterface/Study-Git:main.

jkey20 Git bash 설치

- Git bash 설치.md
- README.md
- SourceTree 설치.md

Git bash 설치

Update README.md

SourceTree 설치

README.md

Study-Git

기하브 스터디

Clone

HTTPS SSH GitHub CLI

<https://github.com/jkey20/Study-Git.git>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Open with Visual Studio

Download ZIP

Pin Watch 0 Fork 1 Star 0

About

깃허브 스터디

- Readme
- 0 stars
- 0 watching
- 1 fork

Releases

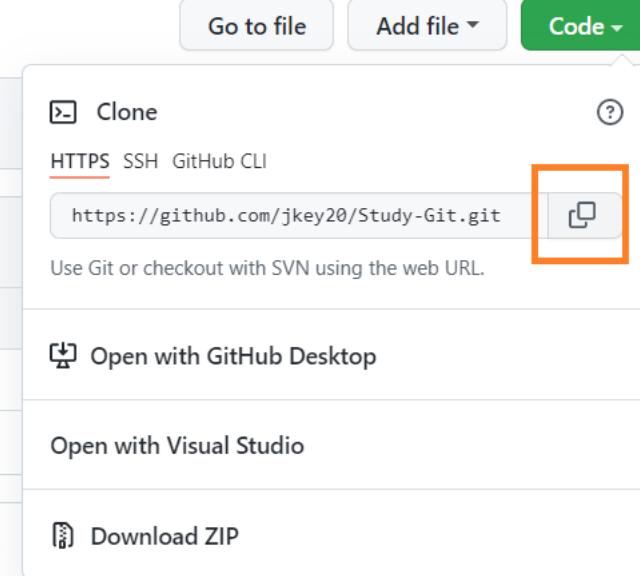
No releases published

Create a new release

Packages

No packages published

Publish your first package



# 사용방법 - Clone

The screenshot shows a software interface with a blue header bar. The menu bar includes '파일(F)', '편집(E)', '보기(V)', '저장소(R)', '액션(A)', '도구', and '도움말(H)'. On the right side of the header are icons for minimizing, maximizing, and closing the window. Below the header is a toolbar with 'New tab' (with a close button), a '+' icon, and three buttons labeled 'Local', 'Remote', and 'Clone'. The 'Clone' button is underlined, indicating it is selected.

The main content area is titled 'Clone' and contains the following text: 'Cloning is even easier if you set up a remote account'. There are three input fields with placeholder text: '소스 경로 / URL:' (highlighted with a red box), '목적지 경로:', and '이름:'. Each input field has a '탐색' (Browse) button to its right. Below these fields is a 'Local Folder:' dropdown menu set to '[루트]'. At the bottom left is a '고급 옵션' (Advanced Options) link, and at the bottom center is a large blue '클론' (Clone) button.

Cloning is even easier if you set up a remote account

소스 경로 / URL:

탐색

저장소 종류: ① 경로 / URL 노란

목적지 경로:

탐색

이름:

Local Folder:

[루트]

고급 옵션

클론

# 사용방법 - Clone

The screenshot shows a web browser window with a blue header bar. The address bar contains the URL `ifgitstudy`. Below the address bar is a toolbar with icons for Local, Remote, Clone (which is highlighted in blue), Add, and Create. The main content area is titled "Clone". It displays the message "Cloning is even easier if you set up a remote account". A URL input field contains `https://github.com/jkey20/Study-Git.git`. Below it, a note says "저장소 종류: Git 저장소입니다". A red box highlights the "목적지 경로:" input field, which currently contains "Study-Git". To the right of this field is a "탐색" (Search) button. Further down, there is a "Local Folder:" dropdown menu with "[루트]" selected. At the bottom left, there is a link "▶ 고급 옵션" and a red error message: "Please correct the following input errors: Destination cannot be empty". A large blue button at the bottom right is labeled "클론" (Clone).

파일(F) 편집(E) 보기(V) 저장소(R) 액션(A) 도구 도움말(H)

ifgitstudy New tab × +

Local Remote **Clone** Add Create

## Clone

Cloning is even easier if you set up a [remote account](#)

`https://github.com/jkey20/Study-Git.git`

저장소 종류: Git 저장소입니다

목적지 경로:

Study-Git

탐색

Local Folder:

[루트]

▶ 고급 옵션

**Please correct the following input errors:**

Destination cannot be empty

**클론**

# 사용방법

The screenshot shows the SourceTree application interface for managing a GitHub repository named 'ifgitstudy'. The main window displays the commit history for the 'main' branch, showing three commits made on March 25, 2022, at 18:25. The commits are: 'Git bash 설치' (commit 3de77d8), 'SourceTree 설치' (commit 84f1f71397), and 'Update README.md' (commit 210202). The commit details show the author as '주이식 <helios2sic@gmail.com>' and the date as '25 3 2022 18:25'. The commit message for 'Git bash 설치' is 'Git bash 설치'. Below the commit history, a file named 'Git bash 설치.md' is open in the editor, showing its content.

파일(F) 편집(E) 보기(V) 저장소(R) 액션(A) 도구 도움말(H)

ifgitstudy gitstudyfork x +

커밋 Pull Push 패치 브랜치 병합 스테시 폐기 태그

깃 플로우... 원격 터미널 탐색기 설정

WORKSPACE

파일 상태 History Search

브랜치 main 태그 원격 스테시

모든 브랜치 원격 브랜치 날짜순으로 정렬

그래프 설명 날짜 작성자

Git bash 설치 25 3 2022 18:25 주이식 <helios2sic@gmail.com>

SourceTree 설치 25 3 2022 14:26 주이식 <helios2sic@gmail.com>

Update README.md 25 3 2022 12:29 210202 <helios2sic@gmail.com>

Initial commit 25 3 2022 12:21 210202 <helios2sic@gmail.com>

파일 상태순 정렬

커밋: 3de77d8872194ecc428e9f68c195d4d7129fdf61 [3de77d8]

상위 항목: 84f1f71397

작성자: 주이식 <helios2sic@gmail.com>

날짜: 2022년 3월 25일 금요일 오후 6:25:48

커밋한 사람: 주이식

Git bash 설치

Git bash 설치.md

파일 내용

```
1 + ## 1. 다운로드
2 + 이란?

- 코드의 변화를 시간순으로 기록하는 것

# 사용방법 - Commit

The screenshot shows the SourceTree application interface for managing a GitHub repository named 'gitstudyfork'. The 'History' tab is selected in the left sidebar. A commit from 'main' branch on March 25, 2022, at 18:25 is highlighted, titled 'Git bash 설치'. The commit message details the installation of Git bash, including updating the README.md and performing an initial commit.

| 작성자                           | 날짜              | 설명               |
|-------------------------------|-----------------|------------------|
| 주이식 <helios2sic@gmail.com>    | 25 3 2022 18:25 | Git bash 설치      |
| 주이식 <helios2sic@gmail.com>    | 25 3 2022 14:26 | SourceTree 설치    |
| 210202 <helios2sic@gmail.com> | 25 3 2022 12:29 | Update README.md |
| 210202 <helios2sic@gmail.com> | 25 3 2022 12:21 | Initial commit   |

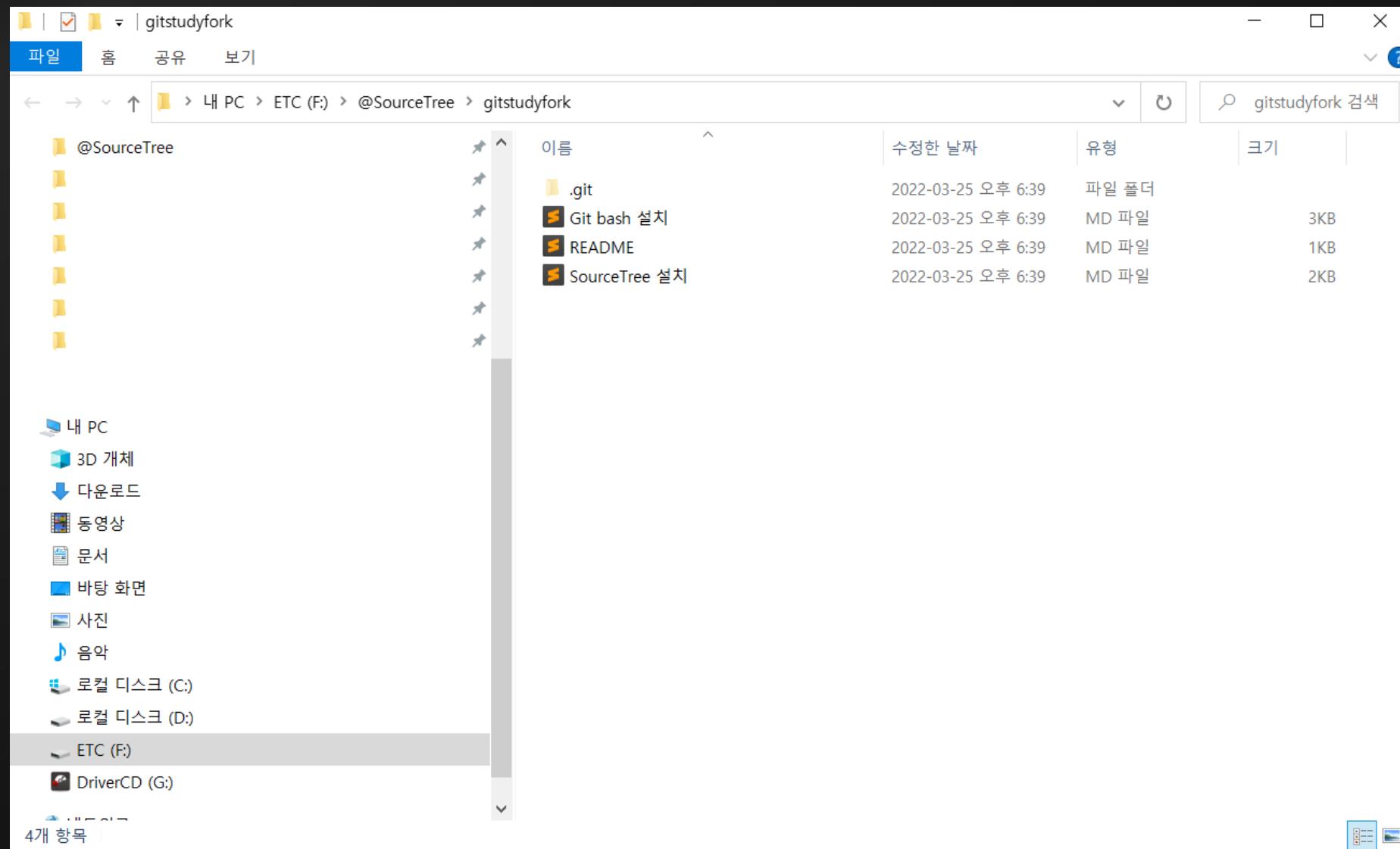
In the bottom right corner, a file editor window is open for 'Git bash 설치.md'. The content of the file is as follows:

```
파일 내용
1 + ## 1. 다운로드
2 + ! [image] (https://user-images.githubusercontent.com/32587845/160089436-4d15d6c9-2935-49e5-8b8
3 +
4 + ## 2. 설치
5 + ! [image] (https://user-images.githubusercontent.com/32587845/160090996-583580cc-aab4-480a-b47
6 +
7 + ## 3. 경로지정
8 + ! [image] (https://user-images.githubusercontent.com/32587845/160091045-e8def3b0-e712-4a2b-a69
9 +
10 + ## 4. 설치 옵션
11 + next 누르기
12 + ! [image] (https://user-images.githubusercontent.com/32587845/160091170-4040d343-0181-4919-898
13 +
14 + ## 5. 시작폴더 경로
15 + next 누르기
```

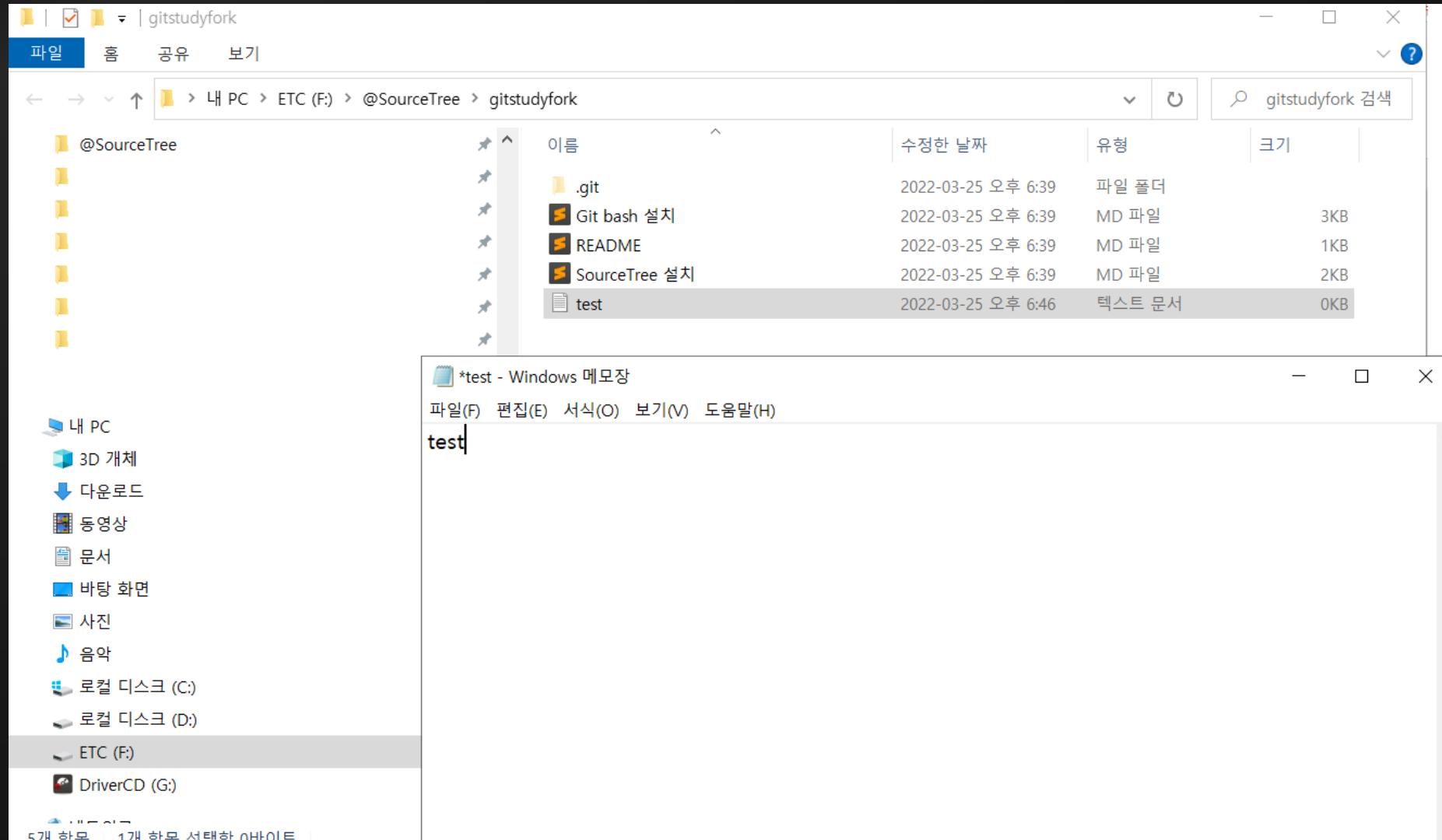
# 사용방법 - Commit

The screenshot shows the GitHub desktop application interface. The top navigation bar includes '파일(F)', '편집(E)', '보기(V)', '저장소(R)', '액션(A)', '도구', and '도움말(H)'. The repository path 'ifgitstudydy / gitstudyfork' is displayed in the title bar. The toolbar contains icons for '커밋' (Commit), 'Pull', 'Push', '패치' (Patch), '브랜치' (Branch), '병합' (Merge), '스태시' (Stash), '폐기' (Delete), '태그' (Tag), '깃 플로우...' (Git Flow...), '원격' (Remotes), '터미널' (Terminal), and '탐색기' (Search), with '탐색기' highlighted by a red box. The left sidebar shows 'WORKSPACE' with '파일 상태' selected, followed by 'History' and 'Search'. Under '브랜치', 'main' is checked. Other options include '태그', '원격', and '스태시'. The main area displays a message '커밋할 내용 없음' (No commit content) with a button '익스플로러에서 열기' (Open in Explorer). The bottom status bar shows the user '주이식 <helios2sic@gmail.com>' and a '커밋 옵션...' (Commit Options...) button.

# 사용방법 - Commit

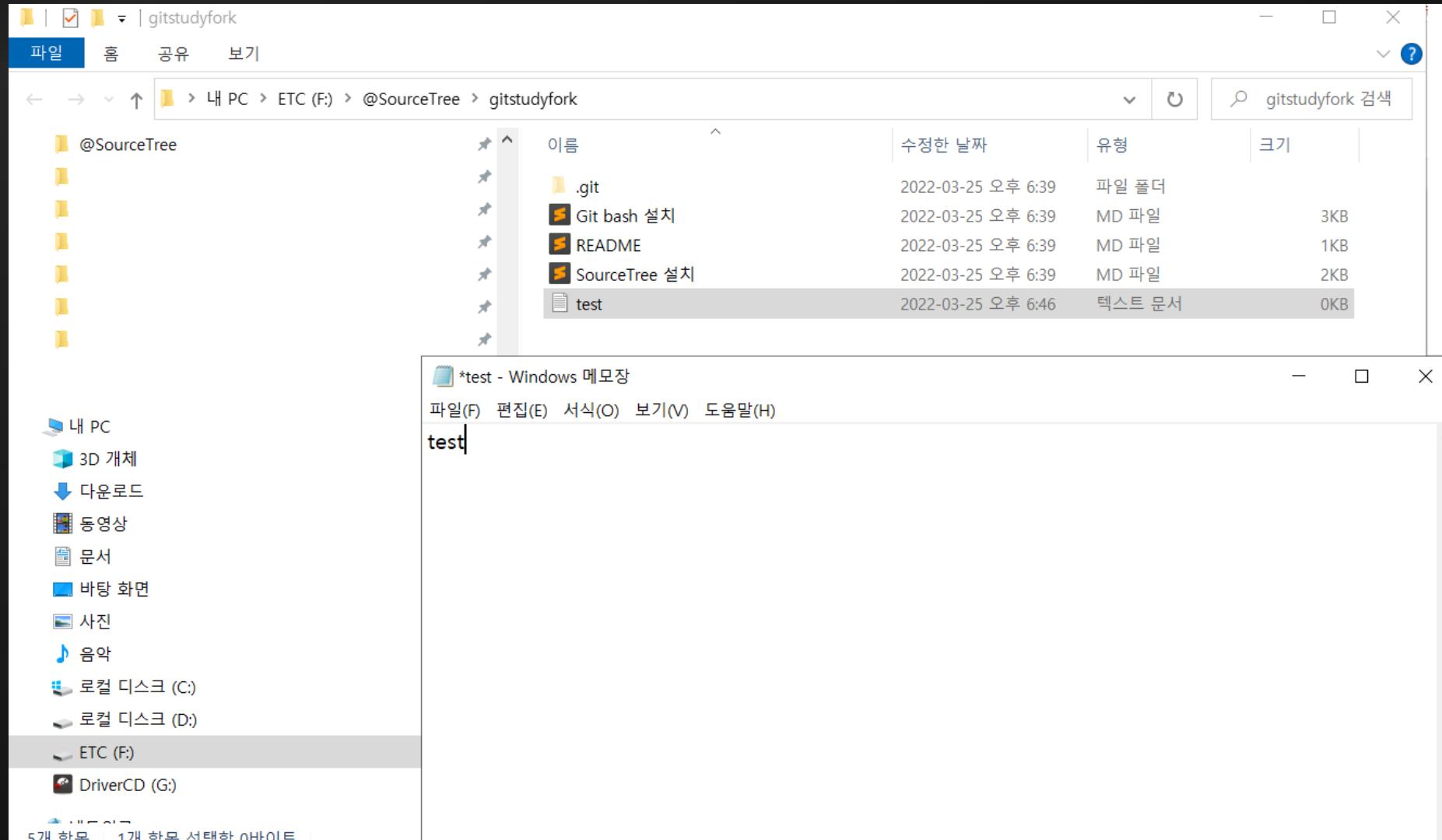


# 사용방법 - Commit



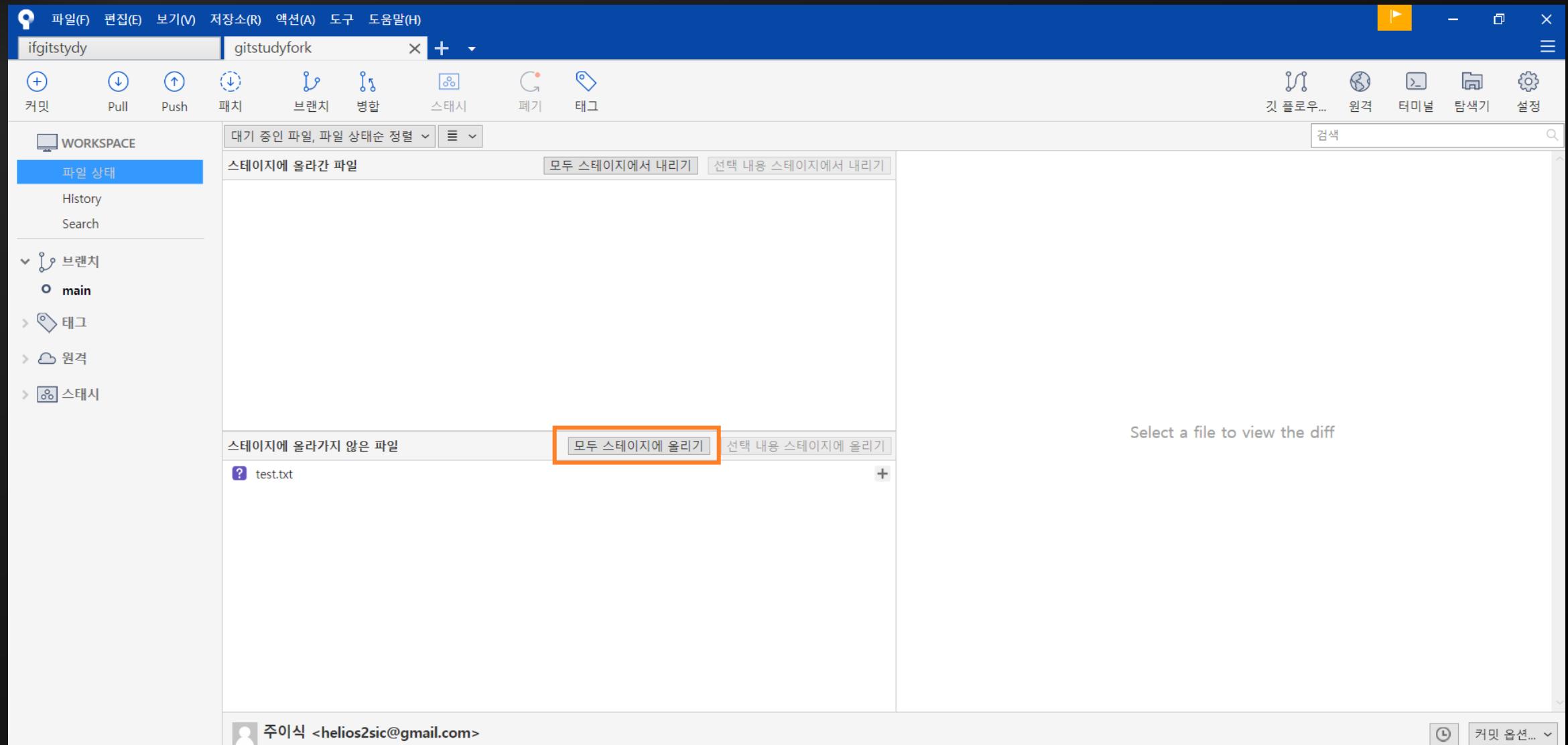
폴더 우클릭 -> 새로 만들기 -> 텍스트 문서 -> test로 이름변경 -> 내용 test로 수정

# 사용방법 - Commit



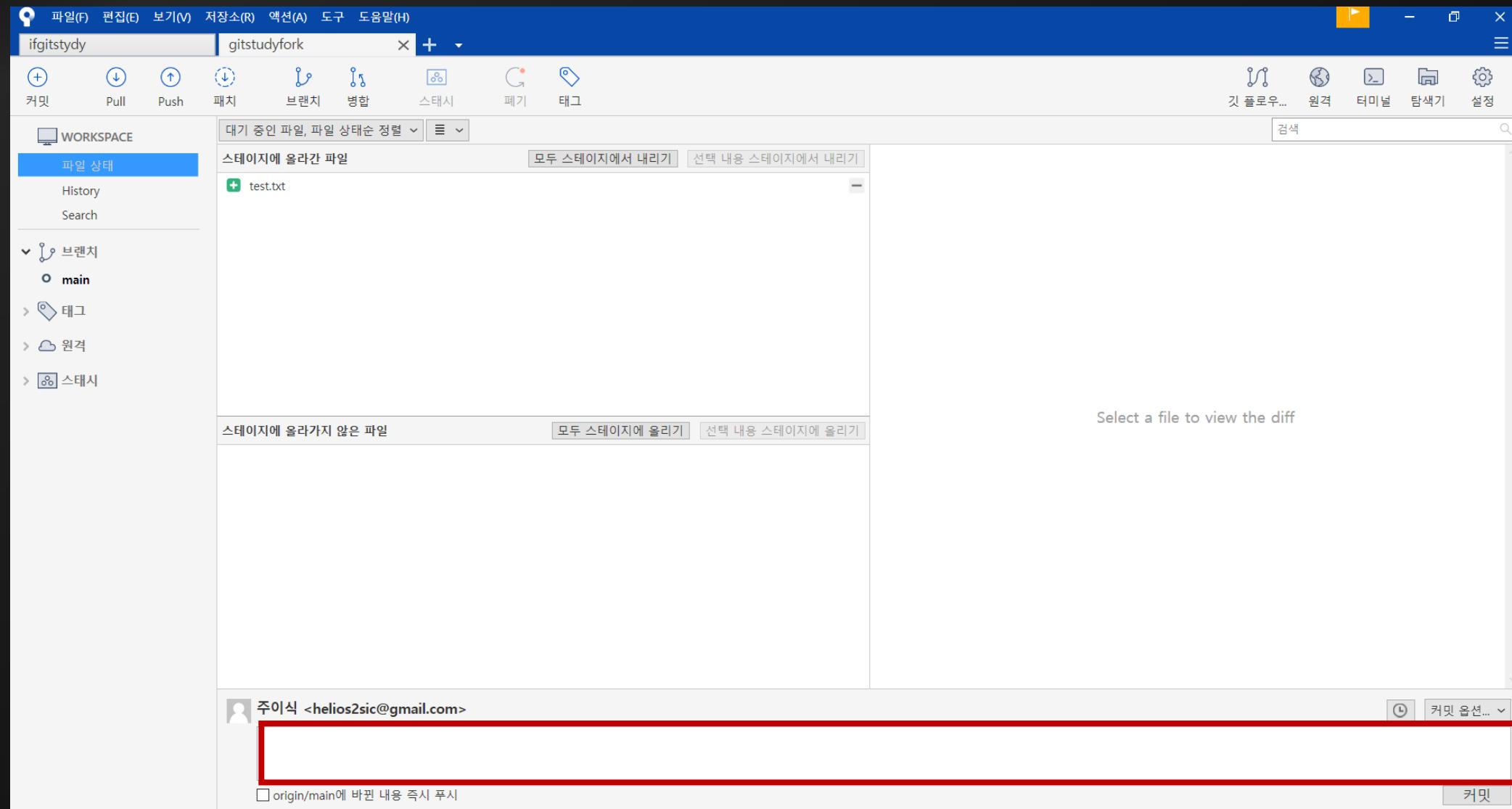
폴더 우클릭 -> 새로 만들기 -> 텍스트 문서 -> test로 이름변경 -> 내용 test로 수정

# 사용방법 - Commit



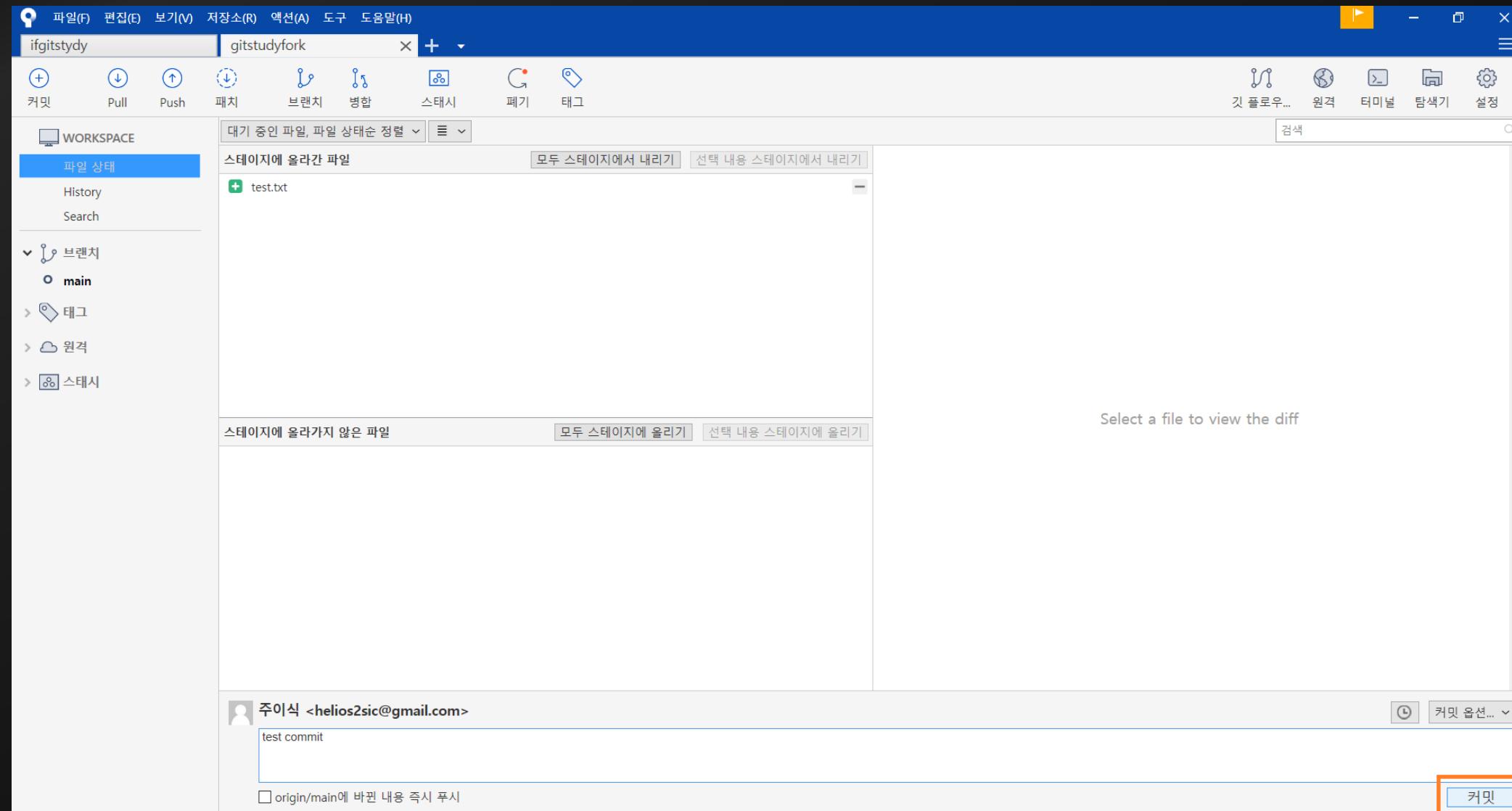
SourceTree로 가면 다음과 같이 바뀜 -> 모두 스테이지에 올리기 클릭

# 사용방법 - Commit



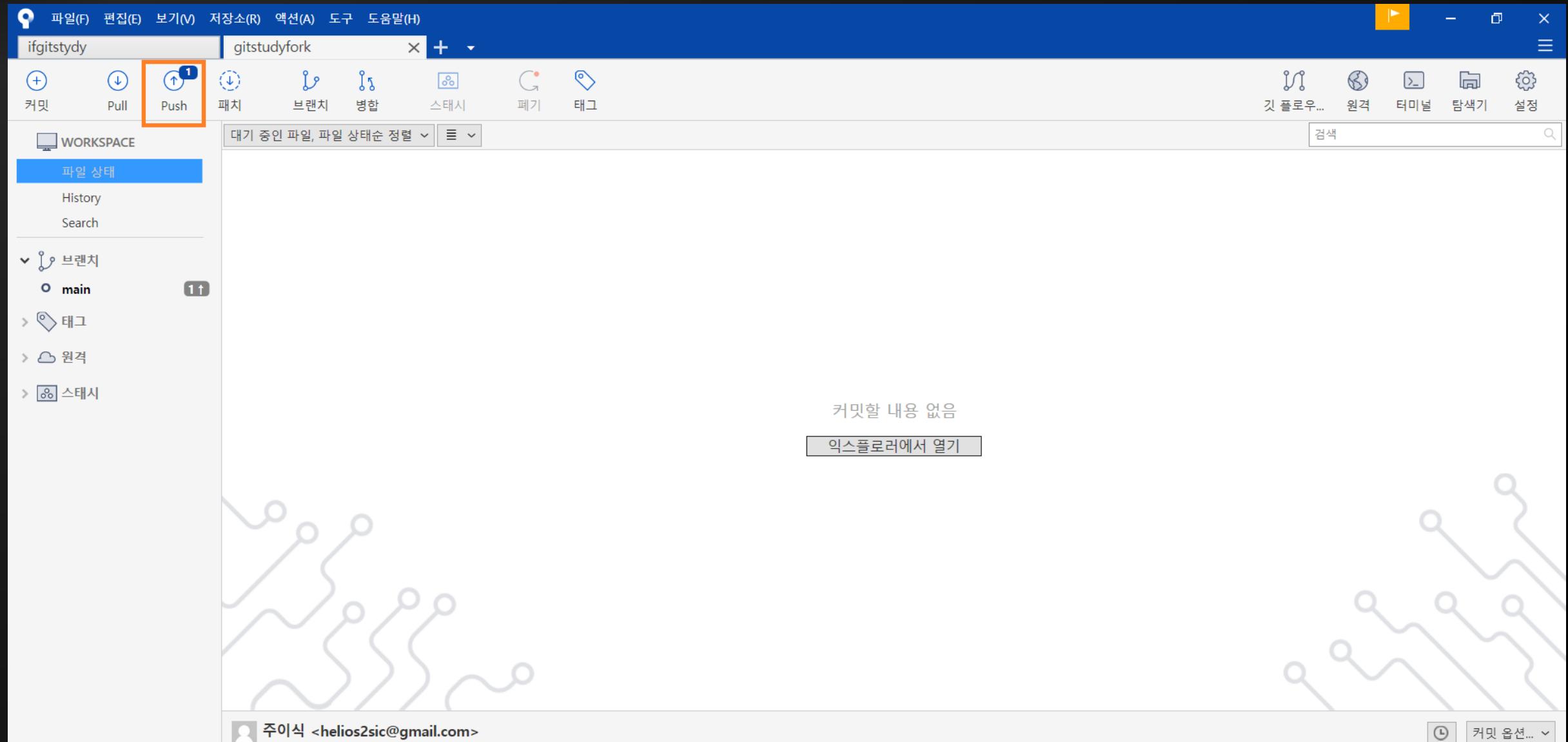
빨간곳에 커밋 메시지를 쓸 수 있음

# 사용방법 - Commit



다음과 같이 커밋 메시지를 적고 우측 하단에 커밋을 누르면 커밋완료

# 사용방법 - Push



커밋이 완료되면 다음과 같이 push 우측 상단에 숫자가 생김

# Push

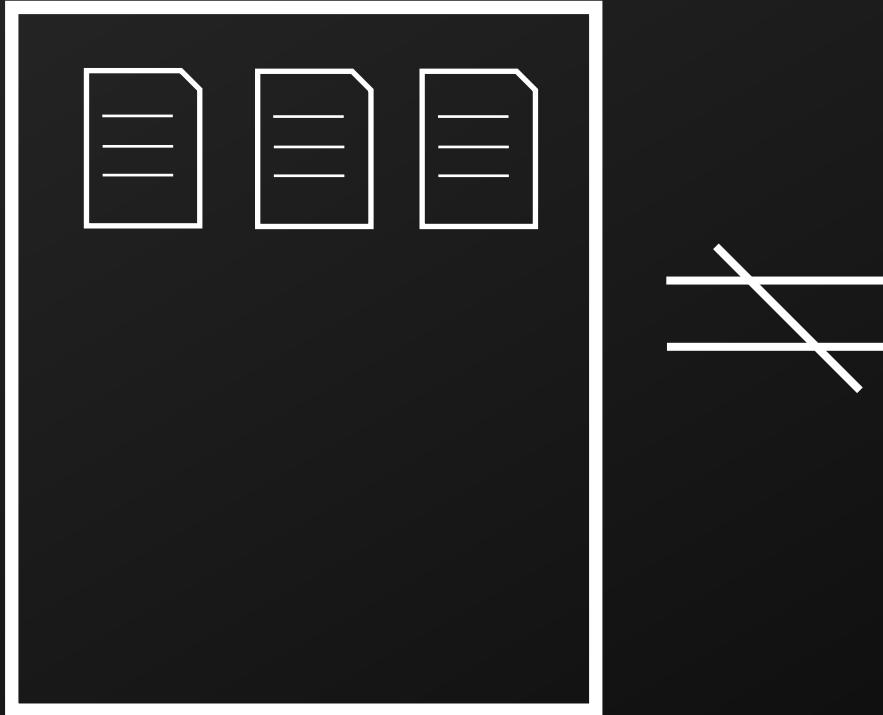
---

푸시(Push)란?

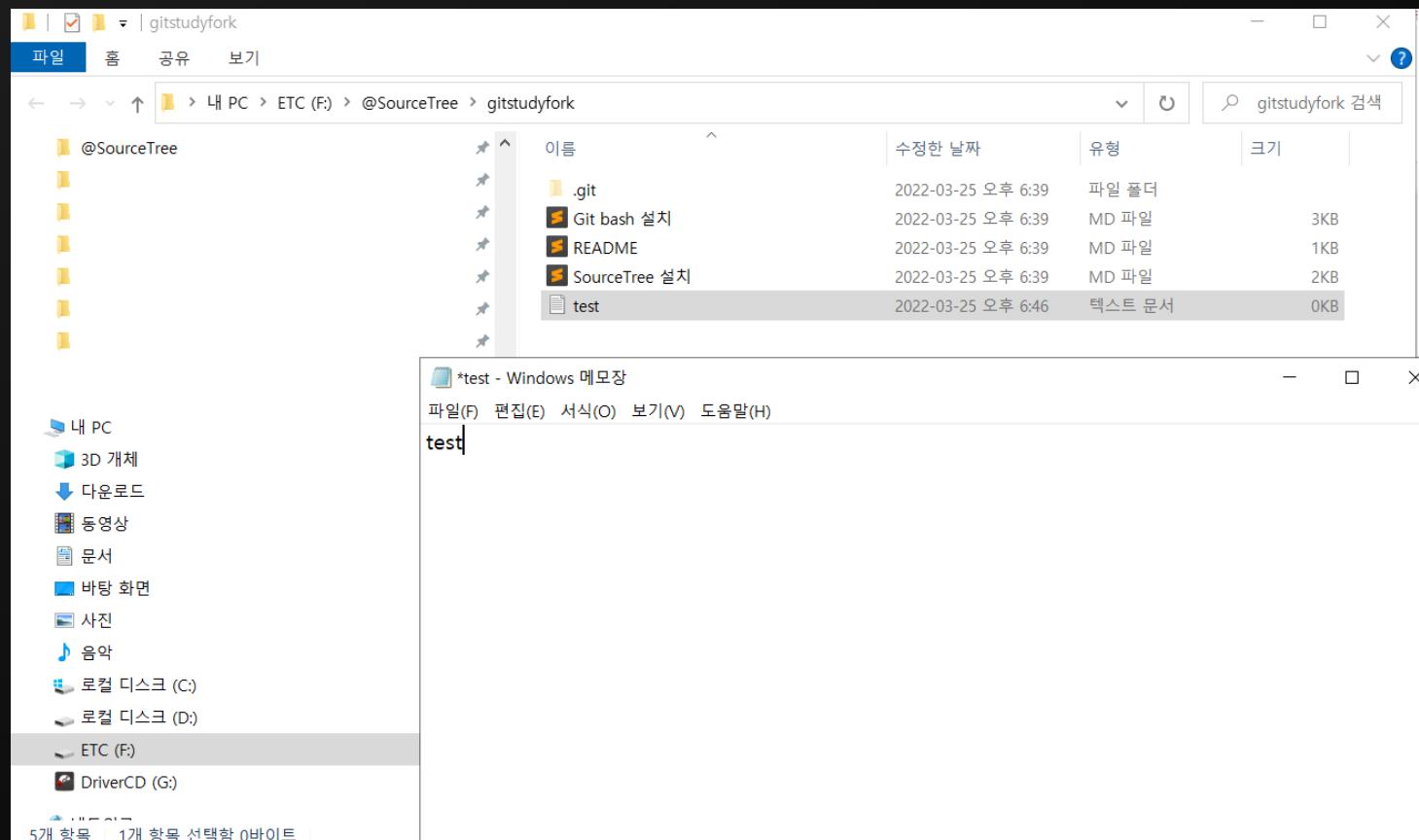
- 저장소로 파일을 업로드 하는 것

# Push

개인 repository([username]/Study-Git)



로컬 repo

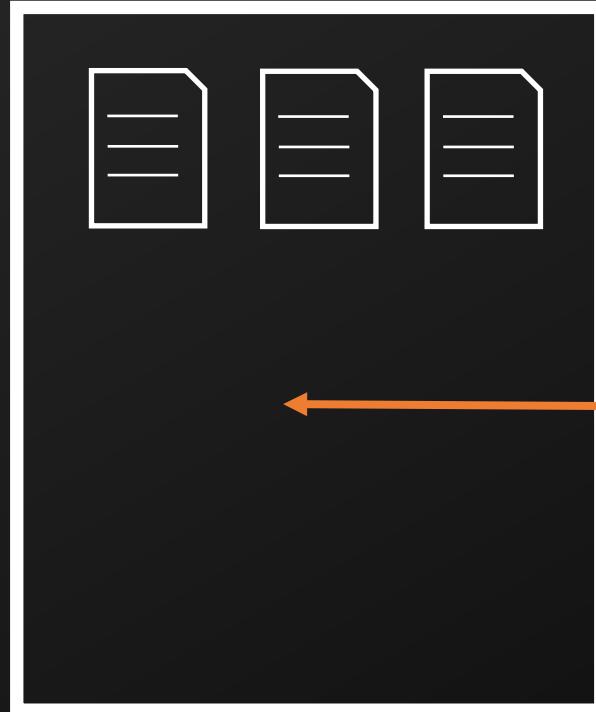


로컬 repo에서 test란 파일을 만들면 개인 repo에는 적용 안되어있음  
따라서 개인 repo에 파일 업로드해서 변경사항을 업데이트 시킴

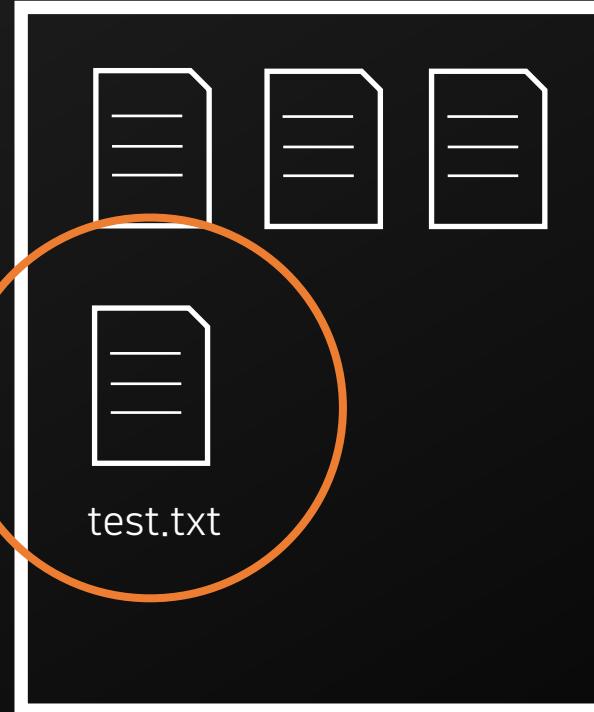
# Push

---

개인 repository([username]/Study-Git)



로컬 repo

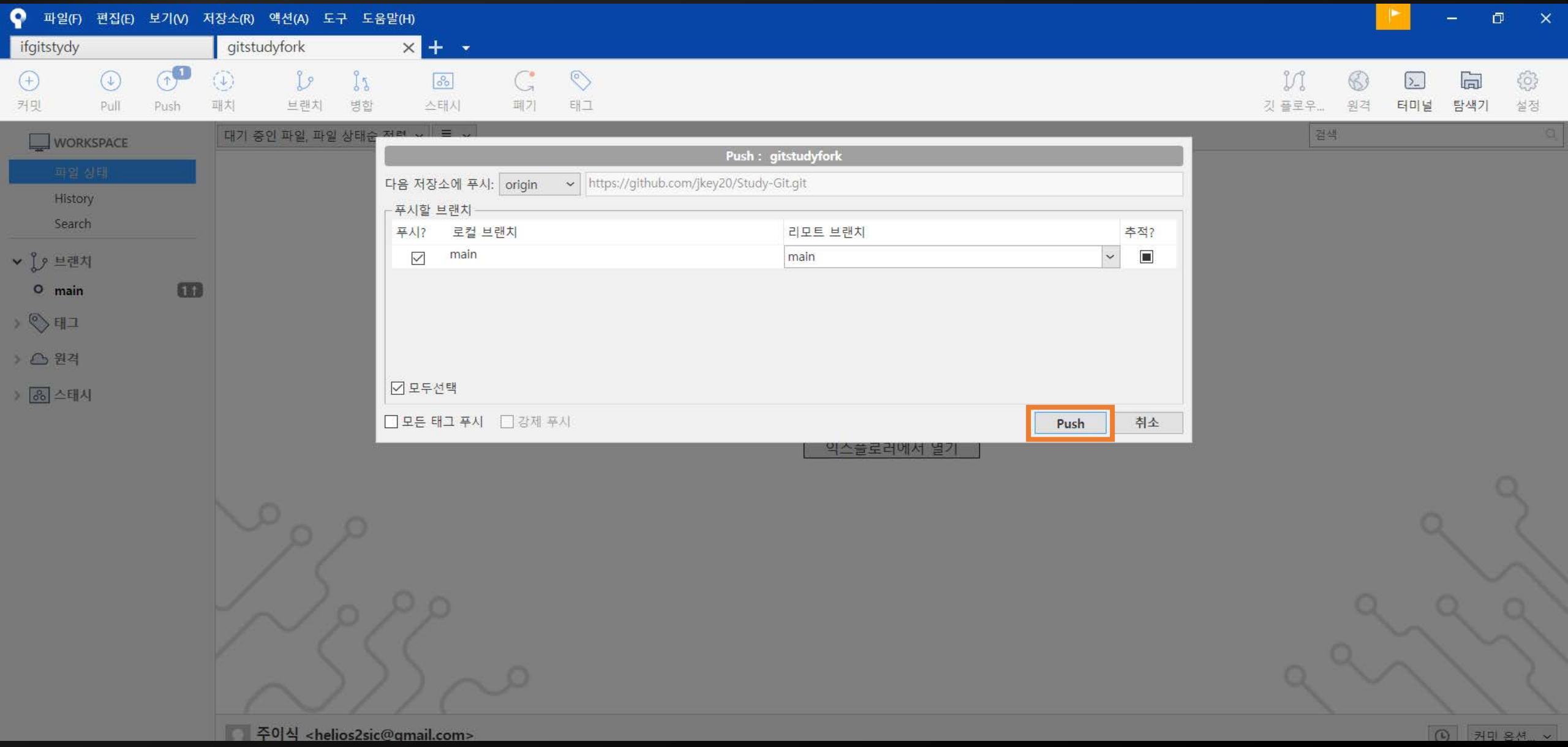


push



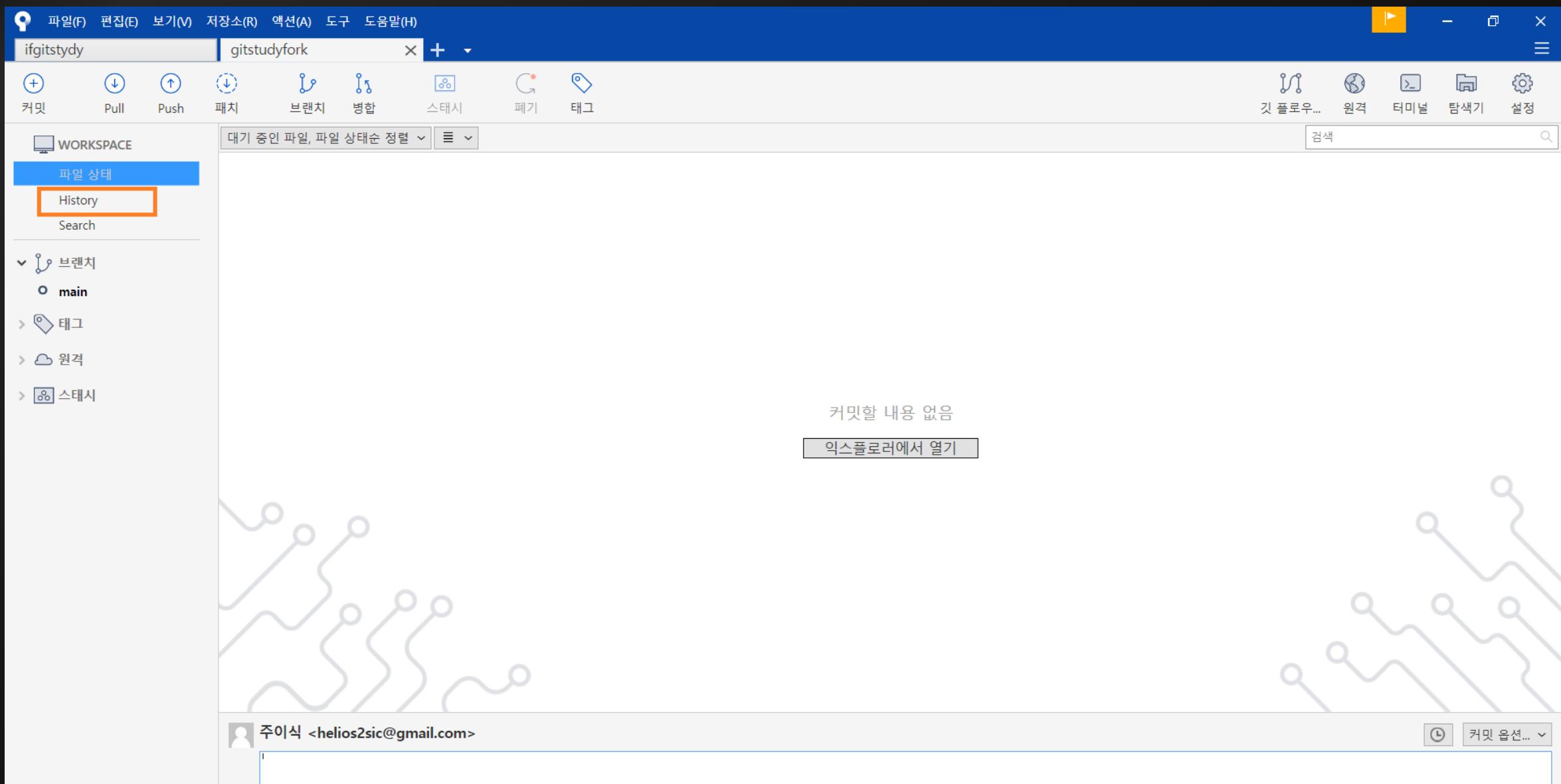
로컬 repo에서 생성된 test 파일을 푸시하면 개인 repo에도 test 파일이 생김

# 사용방법 - Push



push 버튼을 누르면 다음과 같은 화면이 나오고 push버튼을 한번 더 누름

# 사용방법 - Push



push가 완료되면 History를 누름

# 사용방법 - Push

The screenshot shows the SourceTree application interface. The top navigation bar includes File(F), 편집(E), 보기(V), 저장소(R), 옵션(A), 도구, and 도움말(H). The main window displays a commit history for the repository 'gitstudyfork'. The commit history shows the following details:

| 작성자                           | 날짜              | 설명               |
|-------------------------------|-----------------|------------------|
| 주이식 <helios2sic@gmail.com>    | 25 3 2022 18:55 | main test commit |
| 주이식 <helios2sic@gmail.com>    | 25 3 2022 18:25 | Git bash 설치      |
| 주이식 <helios2sic@gmail.com>    | 25 3 2022 14:26 | SourceTree 설치    |
| 210202 <helios2sic@gmail.com> | 25 3 2022 12:29 | Update README.md |
| 210202 <helios2sic@gmail.com> | 25 3 2022 12:21 | Initial commit   |

The bottom section shows a file viewer for 'test.txt' with the content "No changes in this file have been detected, or it is a binary file".

커밋이 잘 된것을 볼 수 있음

# 사용방법 - Push

The screenshot shows a GitHub repository page for 'jkey20 / Study-Git'. The repository is public and was forked from 'sejonginterface/Study-Git'. The main branch is 'main' (1 commit ahead), and there are 0 tags. The repository contains 5 commits by 'jkey20' (test commit) made 13 minutes ago, including changes to 'Git bash 설치.md', 'README.md', 'SourceTree 설치.md', and 'test.txt'. The 'Code' tab is selected. The right sidebar shows the repository's details: 'About' (깃허브 스터디), 'Readme', '0 stars', '0 watching', and '1 fork'. The 'Releases' section indicates no releases have been published, and the 'Packages' section indicates no packages have been published.

jkey20 / Study-Git Public

forked from sejonginterface/Study-Git

Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

This branch is 1 commit ahead of sejonginterface/Study-Git:main. Contribute Fetch upstream

jkey20 test commit aba040c 13 minutes ago 5 commits

Git bash 설치.md Git bash 설치 43 minutes ago

README.md Update README.md 7 hours ago

SourceTree 설치.md SourceTree 설치 5 hours ago

test.txt test commit 13 minutes ago

README.md

Study-Git

About

깃허브 스터디

Readme

0 stars

0 watching

1 fork

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

개인 repo에 가면 test.txt 파일이 있는걸 확인할 수 있음

# 현재 상태

---

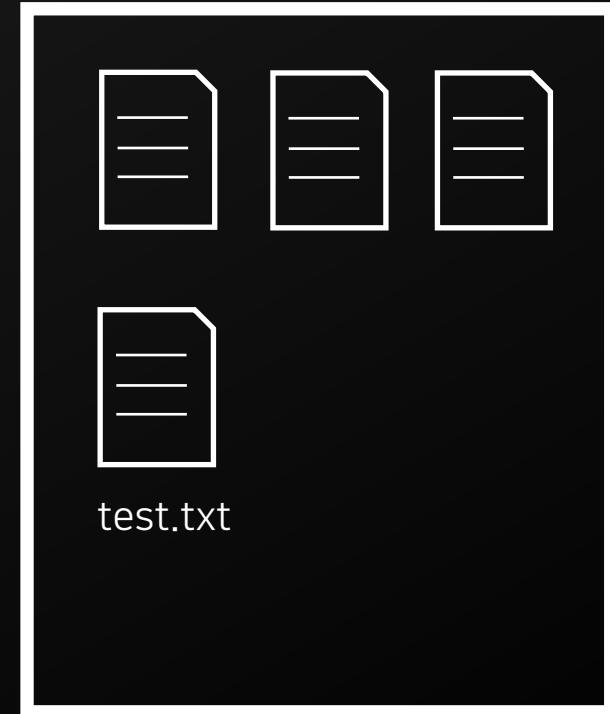
협업 repository(sejonginterface/Study-Git)



개인 repository([username]/Study-Git)



로컬 repo



협업 repo에도 test.txt를 반영시키려면 어떻게 해야할까?

# Pull Request

---

Pull Request란?

- 내가 작업한 코드를 검토후 원격 저장소에 업로드

# 사용방법 - Pull Request

Searched or jump to... / Pull requests Issues Marketplace Explore

jkey20 / Study-Git Public  
forked from sejonginterface/Study-Git

Pin Watch 0 Fork 1 Star 0

Code Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

This branch is 1 commit ahead of sejonginterface/Study-Git:main. Contribute Fetch upstream

jkey20 test commit aba040c 19 minutes ago 5 commits

Git bash 설치.md Git bash 설치 1 hour ago

README.md Update README.md 7 hours ago

SourceTree 설치.md SourceTree 설치 5 hours ago

test.txt test commit 19 minutes ago

README.md

Study-Git

About

깃허브 스터디

Readme

0 stars

0 watching

1 fork

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

# 사용방법 - Pull Request

The screenshot shows a GitHub repository page for `jkey20 / Study-Git`. The page has a dark theme. At the top, there is a search bar, a navigation bar with links for `Pull requests`, `Issues`, `Marketplace`, and `Explore`, and a user profile icon. Below the navigation bar, the repository name is displayed along with its status as `Public` and the fact that it was forked from `sejonginterface/Study-Git`. To the right of the repository name are buttons for `Pin`, `Watch` (with 0 notifications), `Fork` (with 1 fork), and `Star` (with 0 stars). A dropdown menu is also visible.

The main content area features a navigation bar with tabs: `Code`, `Pull requests` (which is the active tab, indicated by an orange underline), `Actions`, `Projects`, `Wiki`, `Security`, `Insights`, and `Settings`. Below this is a search bar with filters set to `is:pr is:open`, a `Labels` section (9 labels), a `Milestones` section (0 milestones), and a prominent green `New pull request` button, which is highlighted with an orange border.

The central part of the page displays a message: `Welcome to pull requests!` It explains that pull requests help collaborate on code and provides a link to `create a pull request`. At the bottom, there is a `ProTip!` message stating that the page was updated in the last three days: `updated:>2022-03-22`.

# 사용방법 - Pull Request

The screenshot shows a GitHub repository page for 'sejonginterface / Study-Git'. The 'Code' tab is selected. The main content area is titled 'Comparing changes' and instructs the user to choose two branches to see what's changed or to start a new pull request. It also mentions that the branches can be automatically merged. A green button labeled 'Create pull request' is highlighted with an orange border. Below this, a summary shows 1 commit, 1 file changed, and 1 contributor. A specific commit from 'jkey20' is shown with the message 'test commit' and the date 'Committed on Mar 25, 2022'. At the bottom, it says 'Showing 1 changed file with 0 additions and 0 deletions.' and includes 'Split' and 'Unified' options.

Search or jump to... /

Pull requests Issues Marketplace Explore

sejonginterface / Study-Git Public

Pin Watch 0 Fork 1 Star 2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

### Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base repository: sejonginterface/Study-Git base: main head repository: jkey20/Study-Git compare: main

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

Discuss and review the changes in this comparison with others. [Learn about pull requests](#)

Create pull request

-o 1 commit 1 file changed 1 contributor

Commits on Mar 25, 2022

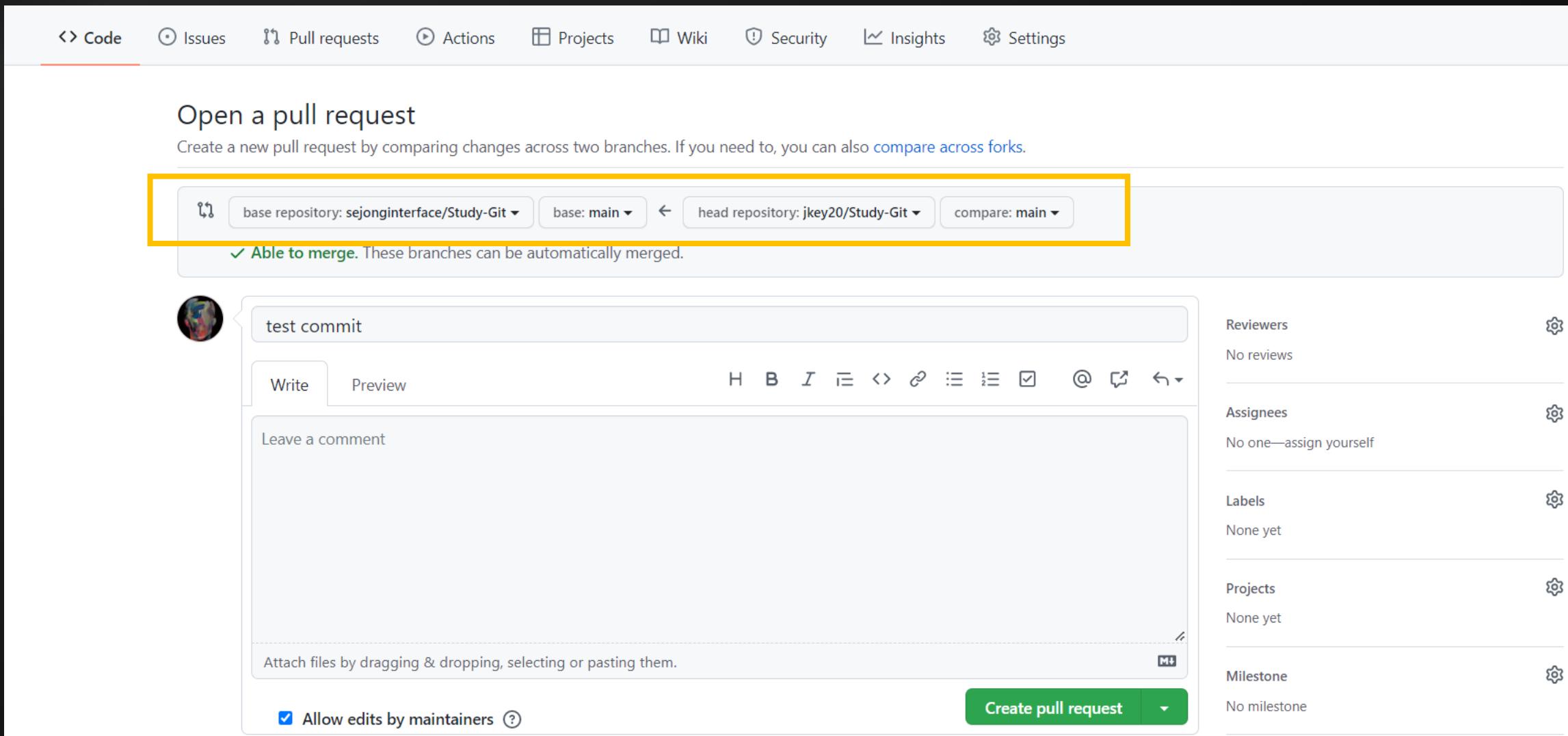
test commit  
jkey20 committed 20 minutes ago

Split Unified

Showing 1 changed file with 0 additions and 0 deletions.

다음과 같이 파일 변경사항을 비교하고 변경이 생긴 파일의 코드를 보여준다. 주황색 버튼을 눌러서 계속 진행하면 된다.

# 사용방법 - Pull Request



The screenshot shows the GitHub interface for creating a pull request. At the top, there is a navigation bar with links: Code, Issues, Pull requests (which is the active tab), Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the title "Open a pull request" is displayed, followed by the sub-instruction: "Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks." A yellow box highlights the "base repository" dropdown set to "sejonginterface/Study-Git", the "base" dropdown set to "main", the "head repository" dropdown set to "jkey20/Study-Git", and the "compare" dropdown set to "main". Below these dropdowns, a green checkmark indicates "Able to merge. These branches can be automatically merged." To the left, there is a commit message input field containing "test commit" and a rich text editor toolbar. To the right, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), and "Milestone" (No milestone). At the bottom, there is a checkbox for "Allow edits by maintainers" and a large green "Create pull request" button.

왼쪽의 base repository가 협업 저장소, head repository가 개인 저장소가 되면 된다.

# 사용방법 - Pull Request

The screenshot shows the GitHub interface for creating a pull request. At the top, there is a navigation bar with links: Code, Issues, Pull requests (which is the active tab), Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the page title is "Open a pull request". A sub-instruction says, "Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks." There are dropdown menus for "base repository" (set to "sejonginterface/Study-Git"), "base" (set to "main"), "head repository" (set to "jkey20/Study-Git"), and "compare" (set to "main"). A green success message states, "✓ Able to merge. These branches can be automatically merged." On the left, there is a user profile icon and a text input field containing "test commit", which is highlighted with a yellow box. Below the input field are tabs for "Write" (selected) and "Preview", and a toolbar with various icons. A large text area labeled "Leave a comment" is followed by a file attachment section. At the bottom, there is a checkbox for "Allow edits by maintainers" and a green "Create pull request" button with a dropdown arrow. To the right of the main form, there are sections for "Reviewers" (no reviews), "Assignees" (no one—assign yourself), "Labels" (none yet), "Projects" (none yet), and "Milestone" (no milestone). Each section has a gear icon for settings.

Pull Request의 제목을 정하는 부분이다.

## 사용방법 - Pull Request

Pull Request의 내용을 정하는 부분이다.

# 사용방법 - Pull Request

The screenshot shows the GitHub interface for creating a pull request. At the top, there is a navigation bar with links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Pull requests' link is highlighted.

The main area is titled 'Open a pull request' and contains instructions: 'Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.' Below this, there are dropdown menus for 'base repository' (sejonginterface/Study-Git), 'base' (main), 'head repository' (jkey20/Study-Git), and 'compare' (main). A green success message says '✓ Able to merge. These branches can be automatically merged.'

The central part of the screen features a rich text editor for writing commit messages. It includes a preview tab, a toolbar with various icons (H, B, I, etc.), and a comment input field. Below the editor, there is a file attachment section and a checkbox for 'Allow edits by maintainers'.

To the right, there are several configuration sections with gear icons:

- Reviewers: No reviews
- Assignees: No one—assign yourself
- Labels: None yet
- Projects: None yet
- Milestone: No milestone

A large green button at the bottom right is labeled 'Create pull request' and is highlighted with a yellow box.

모든 설정이 끝나면 Create pull request를 누른다.

# 사용방법 - Pull Request

The screenshot shows a GitHub repository page for 'sejonginterface / Study-Git'. The 'Pull requests' tab is selected, showing one open pull request. The pull request details are as follows:

- Title:** pull request title #1
- Status:** Open
- Merge Details:** jkey20 wants to merge 1 commit into `sejonginterface:main` from `jkey20:main`
- Activity:** jkey20 commented now: pull request content
- Commits:** 1 (test commit)
- Reviewers:** No reviews. Still in progress? Convert to draft.
- Assignees:** No one—assign yourself.
- Labels:** None yet.
- Projects:** None yet.
- Milestone:** No milestone.

The pull request interface includes sections for CI status, conflict resolution, and merging options. A green 'Merge pull request' button is visible at the bottom left.

다음과 같이 생성된다. 이 상태는 협업하는 사람들이 해당 변경사항을 검토하여 협업 저장소에 반영할지 정하는 상태다.

# 사용방법 - Pull Request

The screenshot shows a GitHub repository page for 'sejonginterface / Study-Git'. The 'Pull requests' tab is selected, showing one open pull request titled 'pull request title #1'. The pull request details show that 'jkey20' wants to merge 1 commit into the 'sejonginterface:main' branch from the 'jkey20:main' branch. A comment from 'jkey20' is visible, and there is a 'test commit' listed. The interface includes sections for conversation, commits, and checks. A modal window is open for merging the pull request, prompting the user to 'Merge pull request #1 from jkey20/main'. The 'Confirm merge' button is highlighted with a yellow box.

pull request title #1

Open jkey20 wants to merge 1 commit into sejonginterface:main

Conversation 0 Commits 1 Checks 0

jkey20 commented now

pull request content

-o test commit

Add more commits by pushing to the main branch on jkey20/St

Continuous integration has not been set up

This branch has no conflicts with the base branch

Merge pull request

pull request title #1

jkey20 commented 1 hour ago

pull request content

-o test commit

Add more commits by pushing to the main branch on jkey20/St

Merge pull request #1 from jkey20/main

pull request title

helios2sic@gmail.com

Choose which email address to associate with this commit

Confirm merge Cancel

Leave a comment

merge pull request를 누르면 오른쪽 화면처럼 나오고 Confirm merge를 누르면 협업 저장소에 반영된다.

# Merge

---

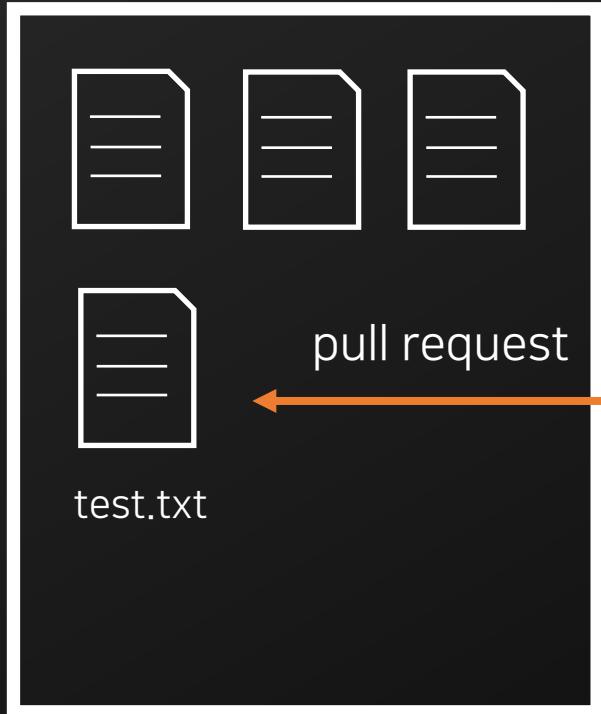
Merge(머지)란?

- 분기된 커밋을 하나로 합치는 작업 -> 추후에 더 자세히 배움

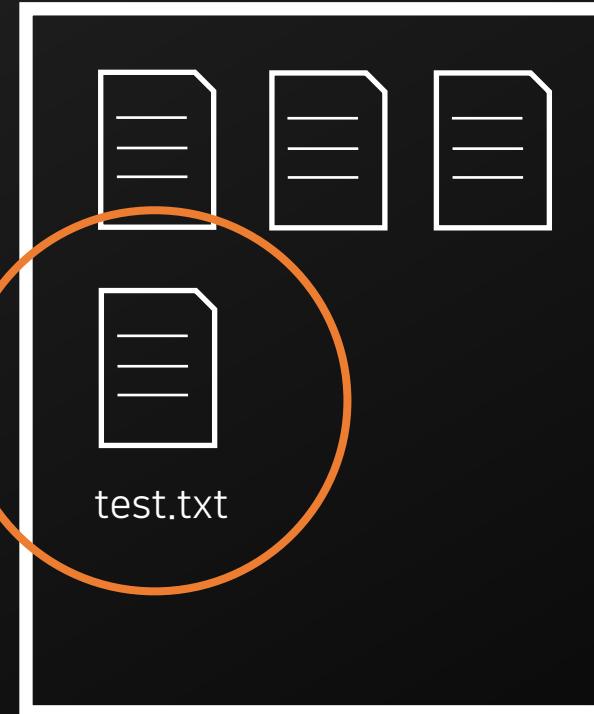
# 현재 상태

---

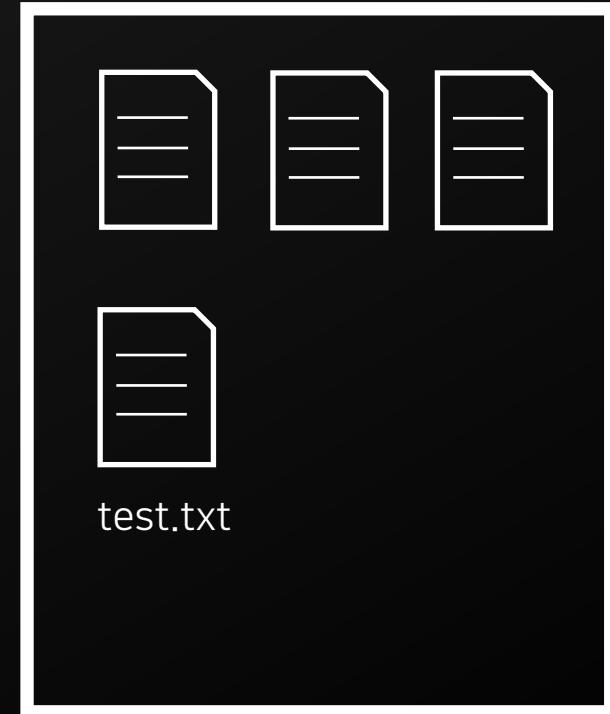
협업 repository(sejonginterface/Study-Git)



개인 repository([username]/Study-Git)



로컬 repo



정상적으로 pull request를 merge시키면 다음과 같이 반영된다.

# 현재 상태

The screenshot shows a GitHub repository page for 'sejonginterface / Study-Git'. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A recent merge pull request from 'jkey20/main' is shown, along with several commits related to setting up tools like Git bash and SourceTree. The repository has 2 stars, 0 watchers, 1 fork, and 2 contributors. It also includes sections for Releases and Packages, both of which are currently empty.

Search or jump to...  Pull requests Issues Marketplace Explore

sejonginterface / Study-Git Public

Pin Watch 0 Fork 1 Star 2

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

jkey20 Merge pull request #1 from jkey20/main ... 2cf686b now 6 commits

Git bash 설치.md Git bash 설치 2 hours ago

README.md Update README.md 8 hours ago

SourceTree 설치.md SourceTree 설치 6 hours ago

test.txt test commit 1 hour ago

README.md

Study-Git

깃허브 스터디

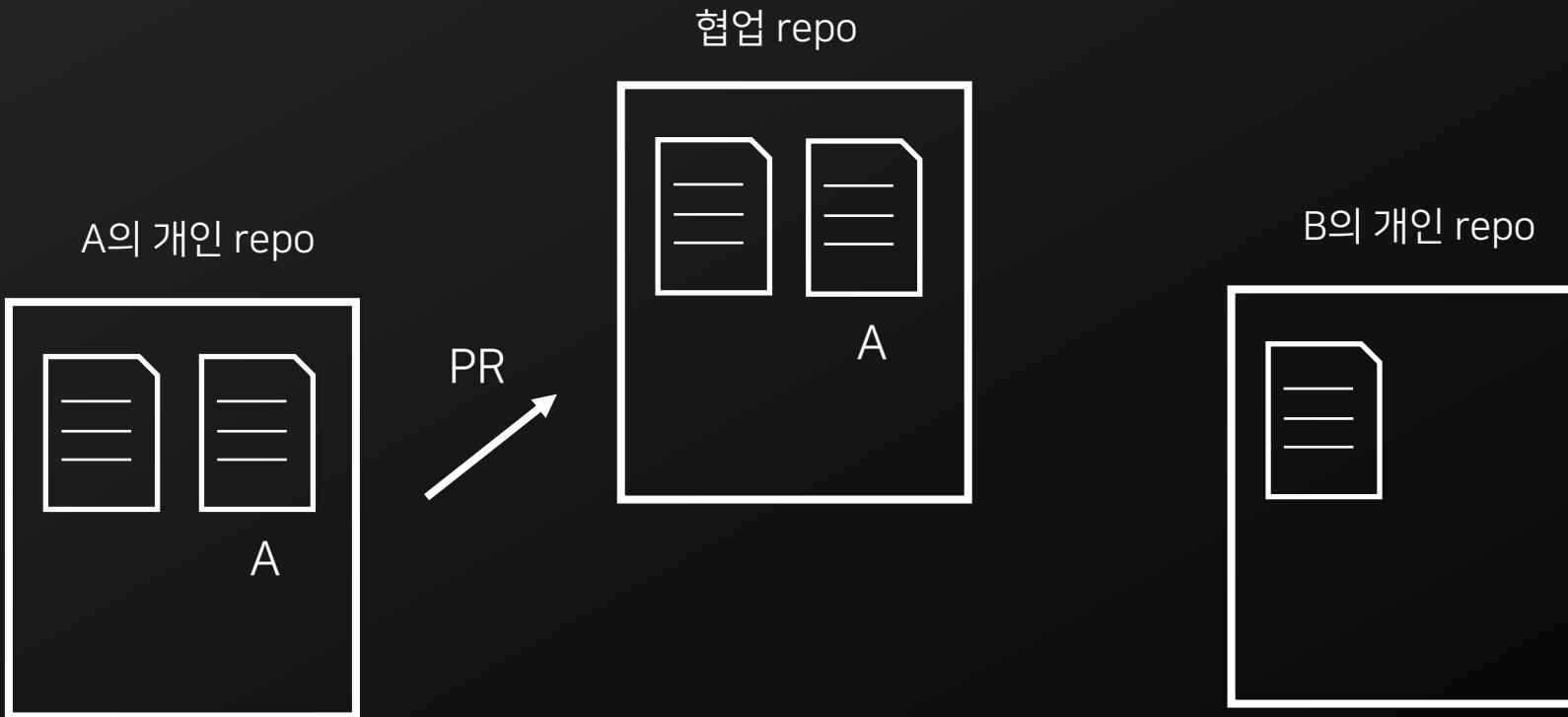
Readme 2 stars 0 watching 1 fork

No releases published Create a new release

No packages published Publish your first package

협업 저장소에 정상적으로 반영된것을 볼 수 있다.

# 협업할 때 - 다른 사람이 협업 저장소에 pr 반영



A가 파일 A를 만들어서 협업 repo에 반영을 시키면 B도 마찬가지로 A의 파일을 받아서 repo에 업데이트 해줘야한다.

# Pull

The image shows two GitHub repository pages side-by-side for comparison.

**Left Repository (sejonginterface/Study-Git):**

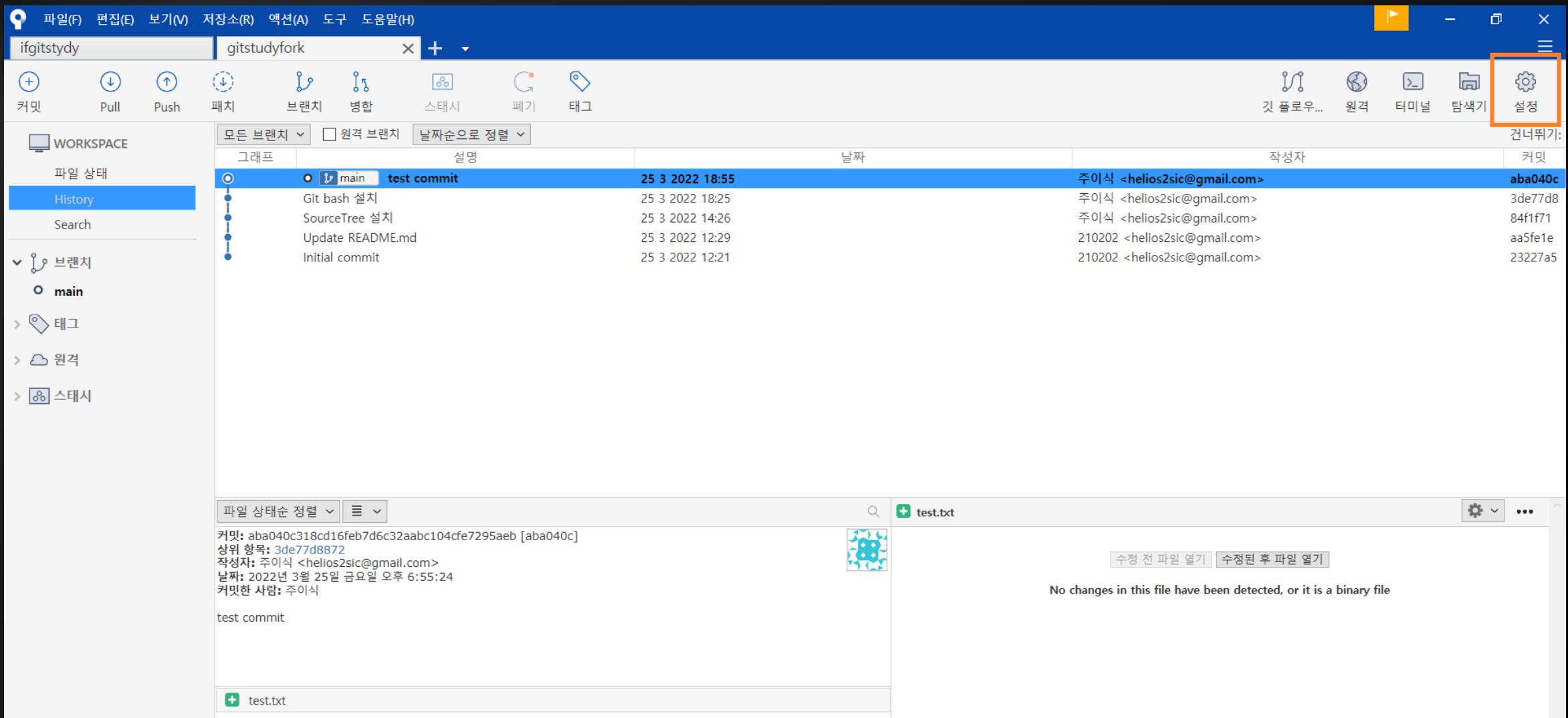
- Code tab:** Shows the main branch with 1 branch and 0 tags.
- Commits:**
  - jkey20 test2 (2 hours ago)
  - Git bash 설치.md (Git bash 설치)
  - README.md (Update README.md)
  - SourceTree 설치.md (SourceTree 설치)
  - test.txt (test commit)
  - test2.txt (test2)
- README.md:** Contains the text "Study-Git".

**Right Repository (jkey20/Study-Git):**

- Code tab:** Shows the main branch with 1 branch and 0 tags.
- Message:** "This branch is 2 commits behind sejonginterface/Study-Git:main."
- Commits:**
  - jkey20 test commit (aba048c 2 hours ago)
    - Git bash 설치.md (Git bash 설치)
    - README.md (Update README.md)
    - SourceTree 설치.md (SourceTree 설치)
    - test.txt (test commit)
- README.md:** Contains the text "Study-Git".

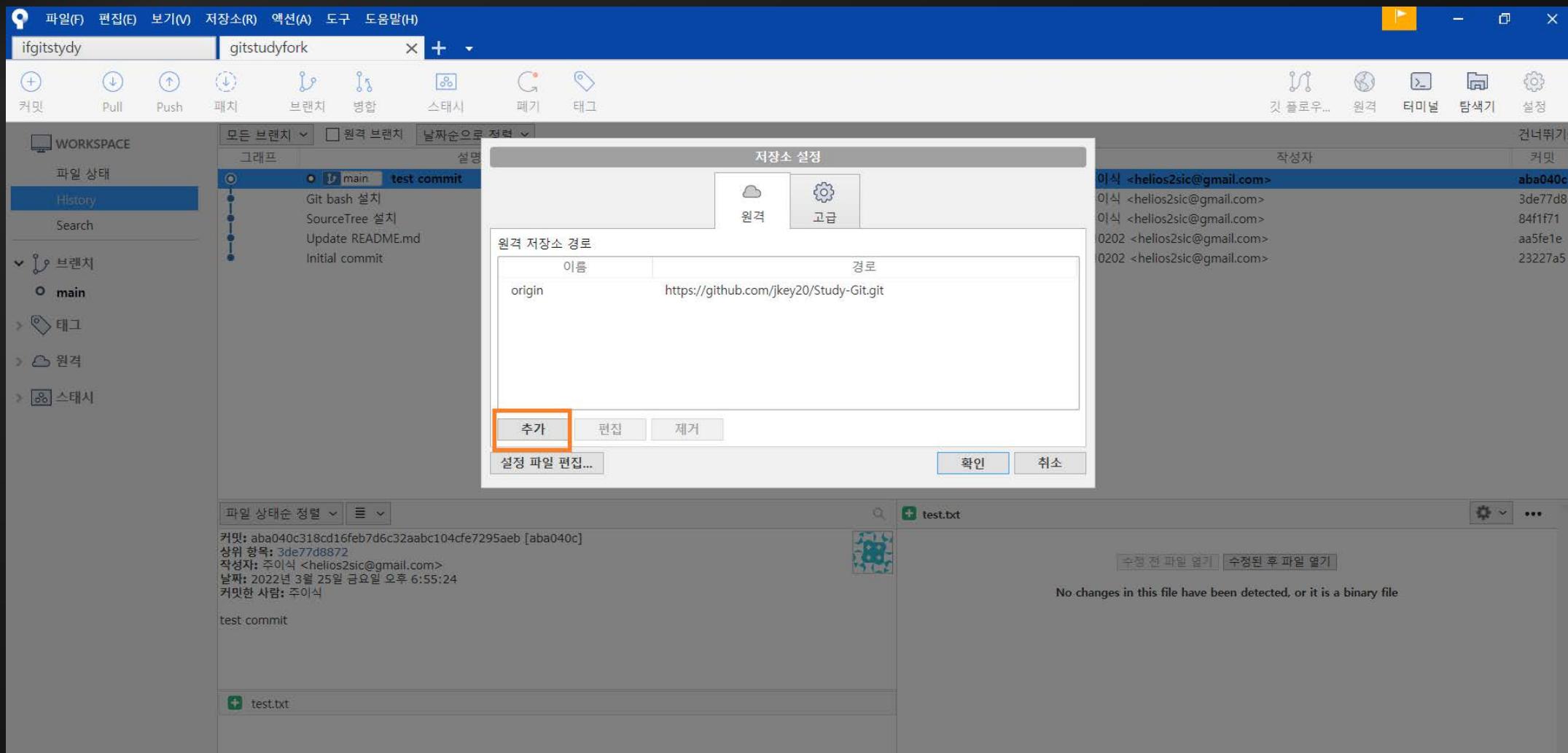
test2라는 파일이 협업 repo에 반영되어있고 개인 repo에는 존재하지 않는다.

# Pull



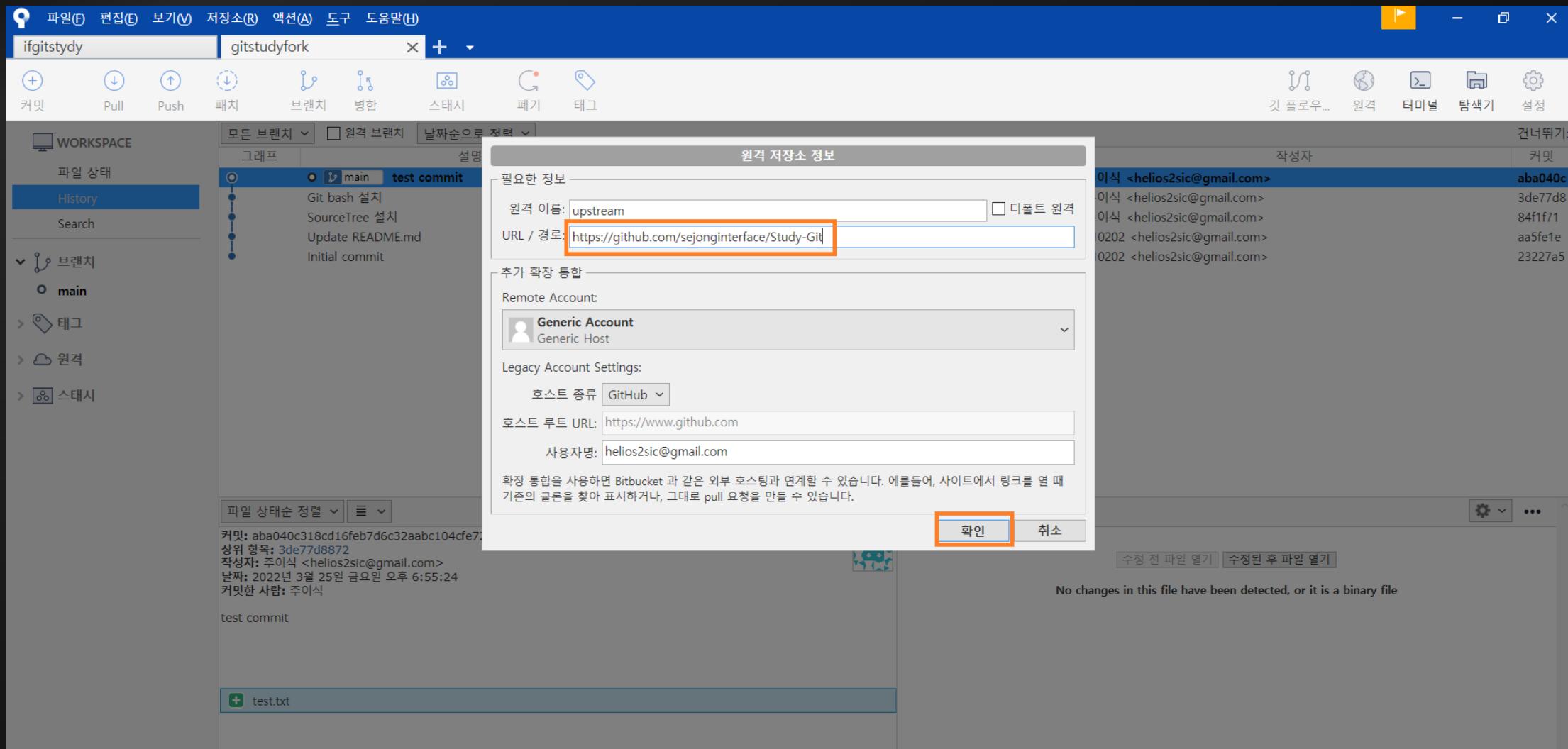
협업 저장소에서 파일을 가져오기 위해서 설정을 눌러 협업 저장소로부터 가져오게 설정한다.

# Pull



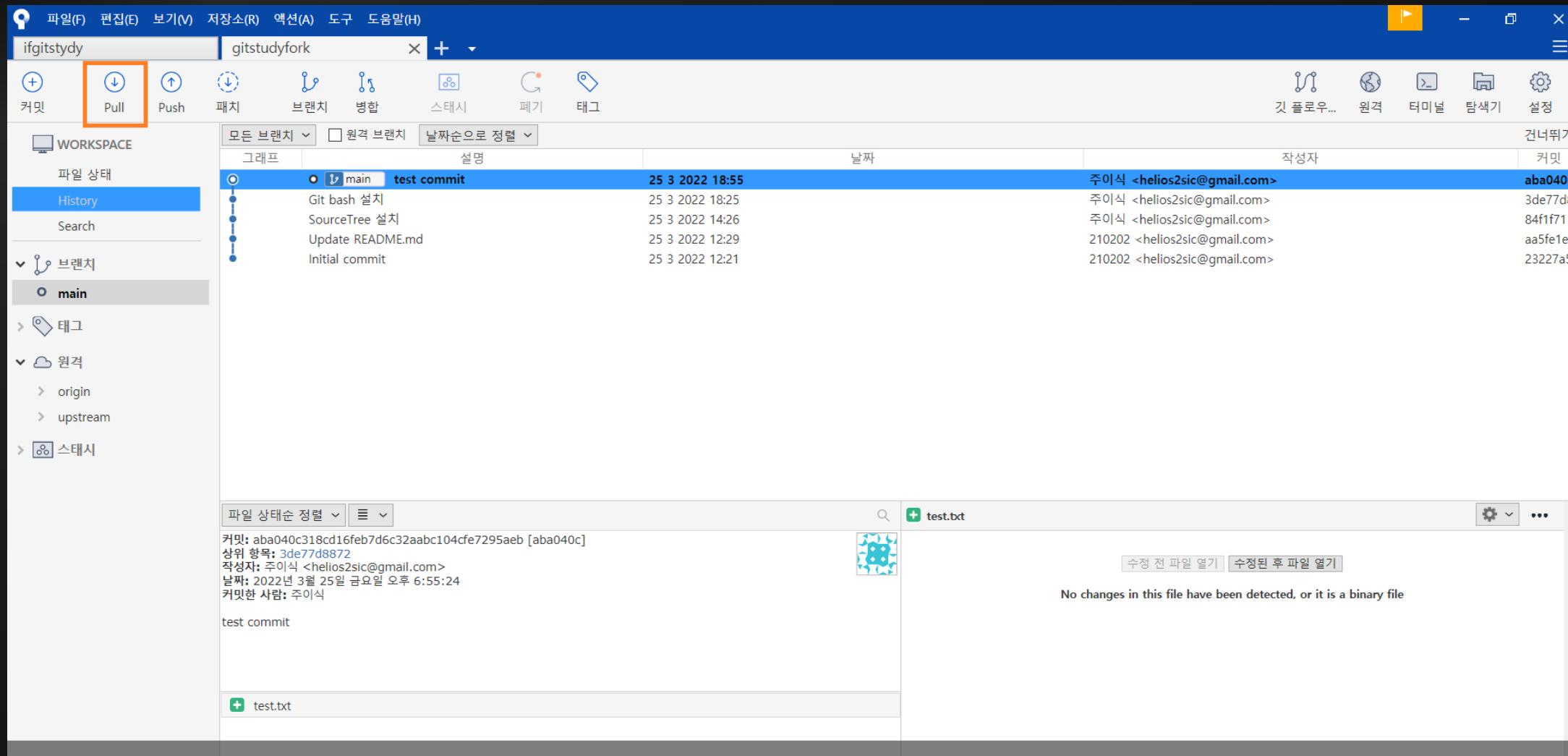
추가 버튼을 누른다.

# Pull



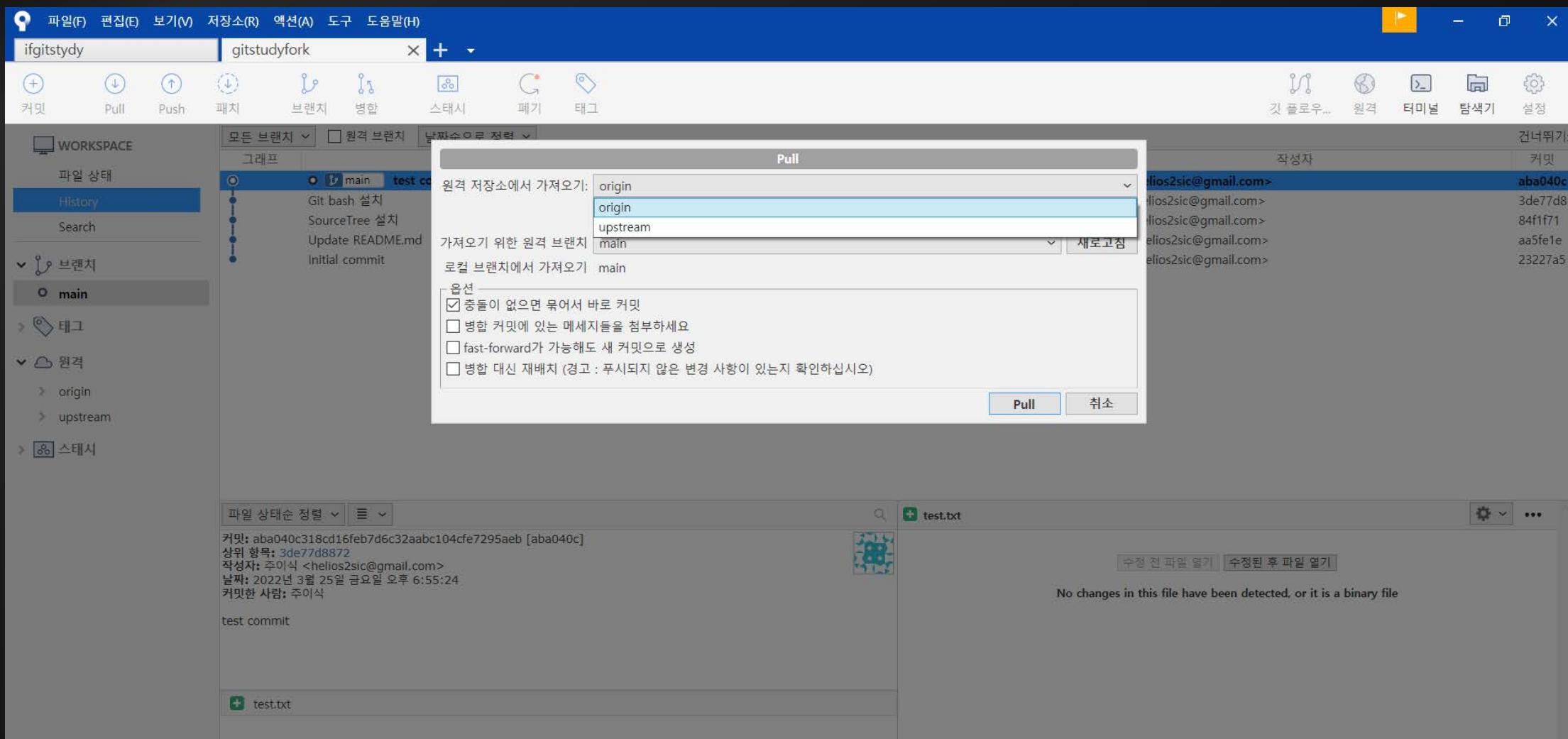
원격의 이름은 upstream으로 설정하고 URL/경로에 협업 repo의 url을 적고 확인을 누른다.

# Pull



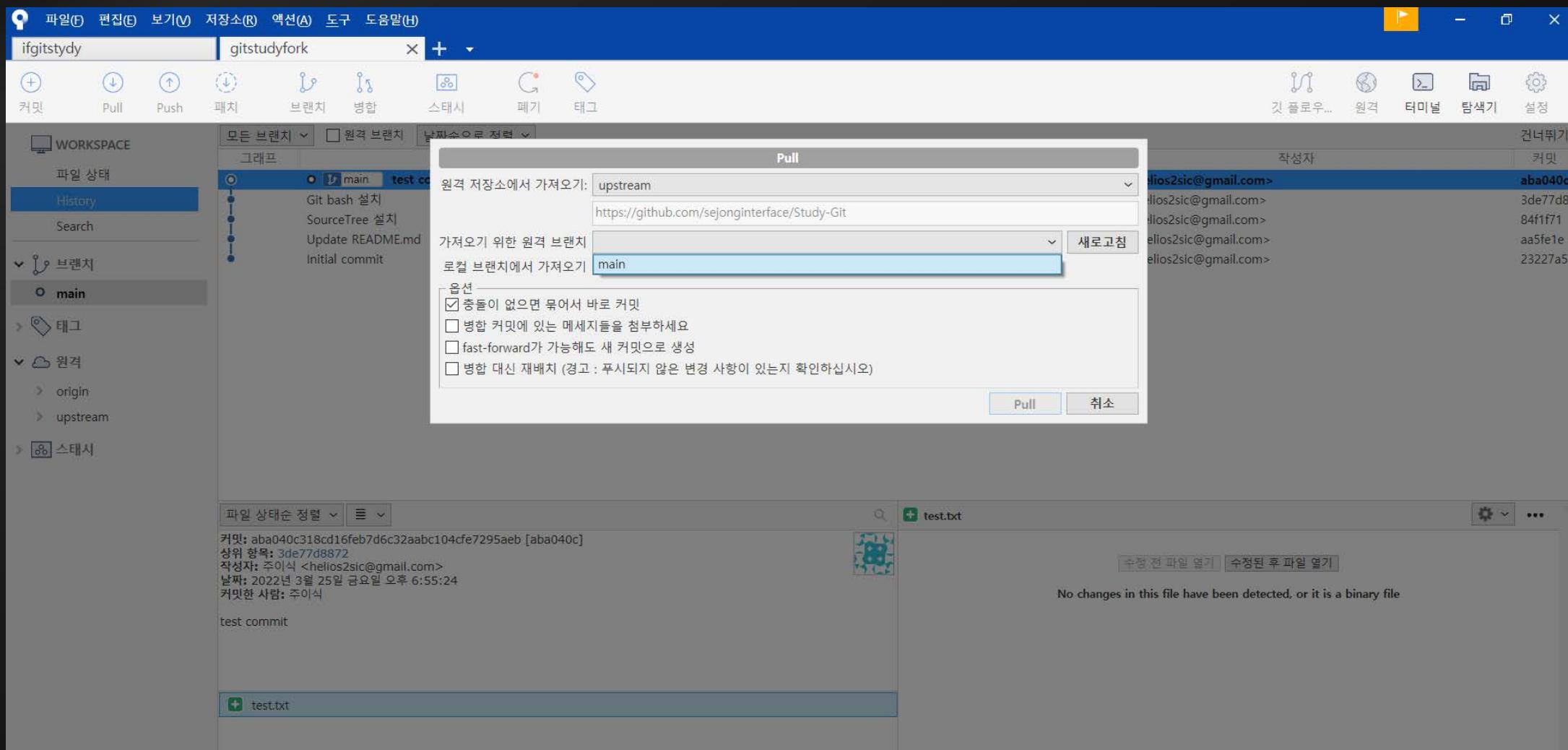
Pull을 누른다.

# Pull



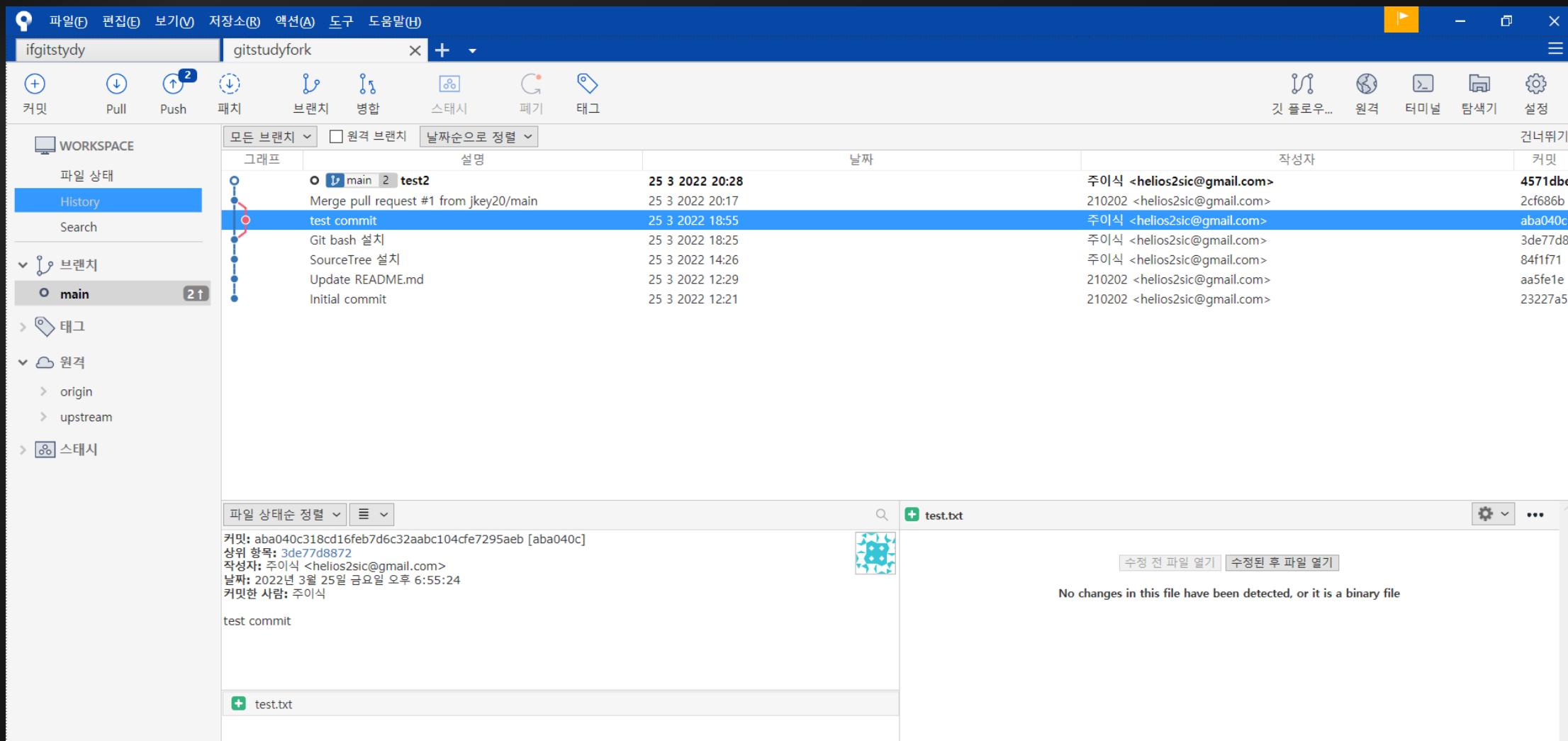
설정한 upstream이 새로 나온다.

# Pull



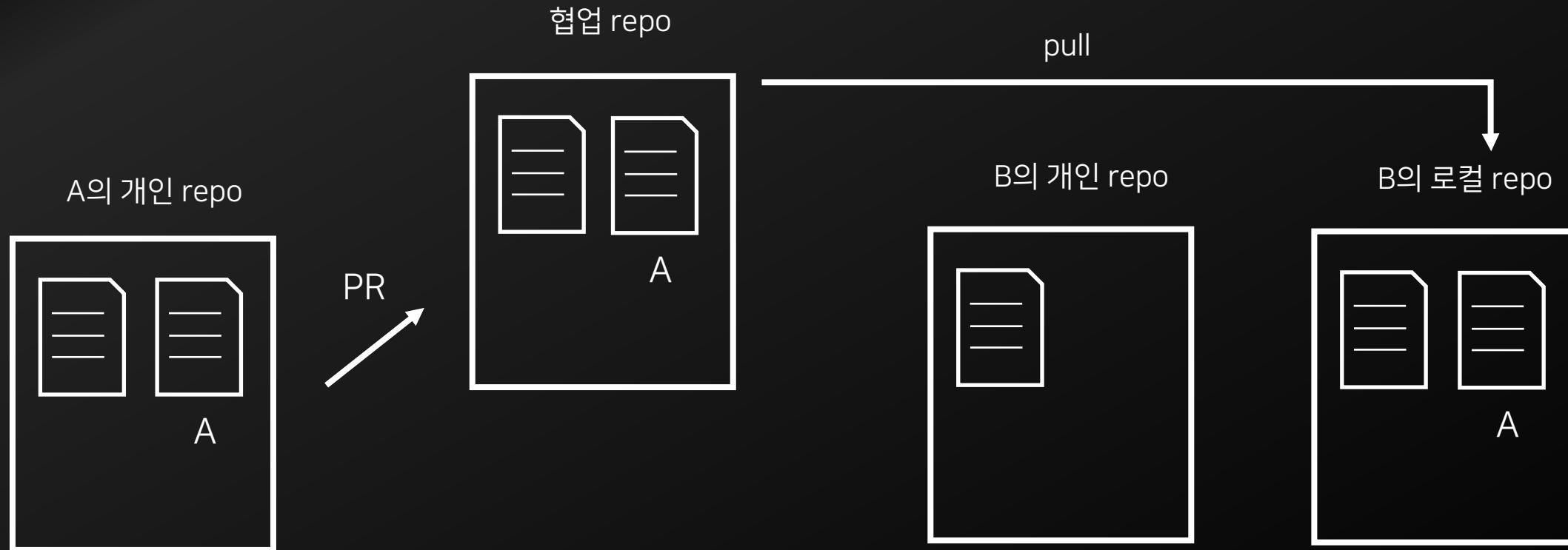
가져오기 위한 원격 브랜치를 main으로 설정하고 pull을 누른다.

# Pull



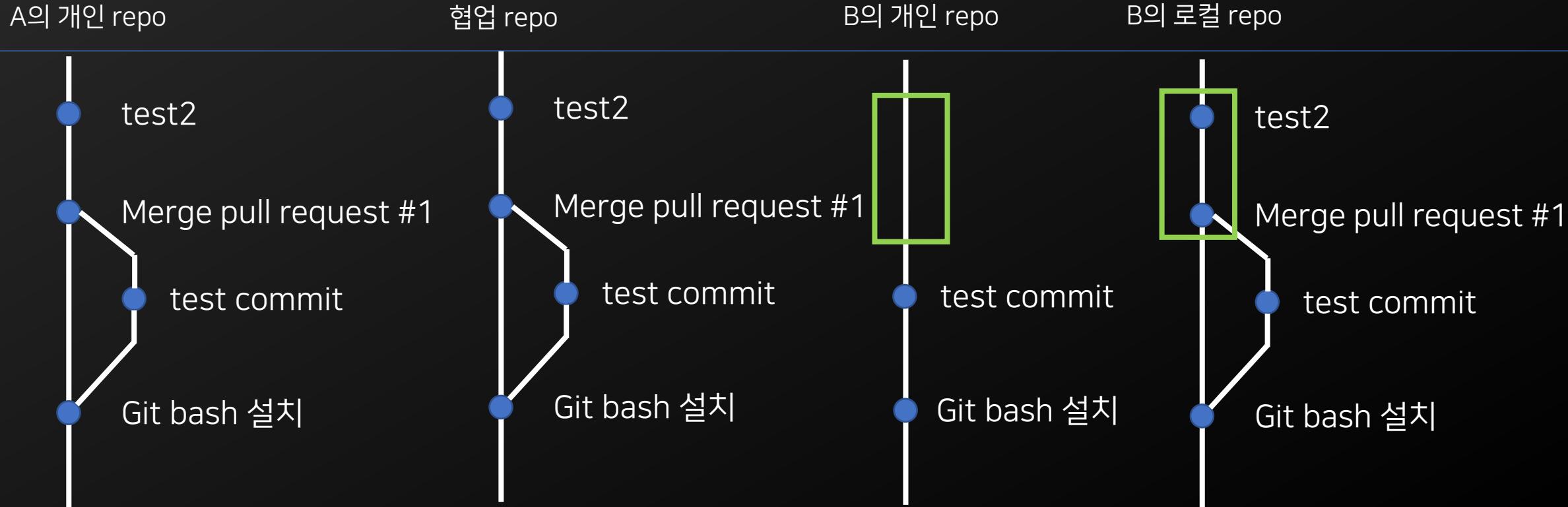
pull이 끝나면 다음과 같이 다른 사람의 커밋이 새로 표시되고 push에 숫자가 나온다.

# 현재 상태



협업 repo에서 로컬 repo로 pull을 통해 A 파일을 받아온 상태여서 개인 repo에 반영해줘야하므로 push에 숫자가 나온것

# 현재 상태 - 커밋



push에 숫자가 2가 나온이유는 커밋이 2개가 차이나기때문. push를 완료하면 모든 repo에 똑같이 반영됨.

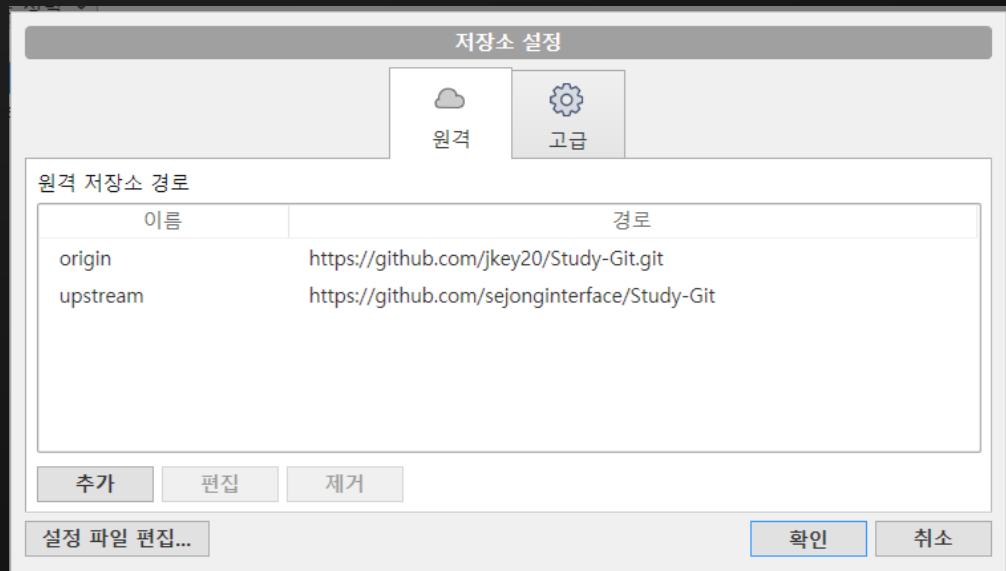
# 원격 저장소

- SourceTree상에서 설정한 origin의 경로

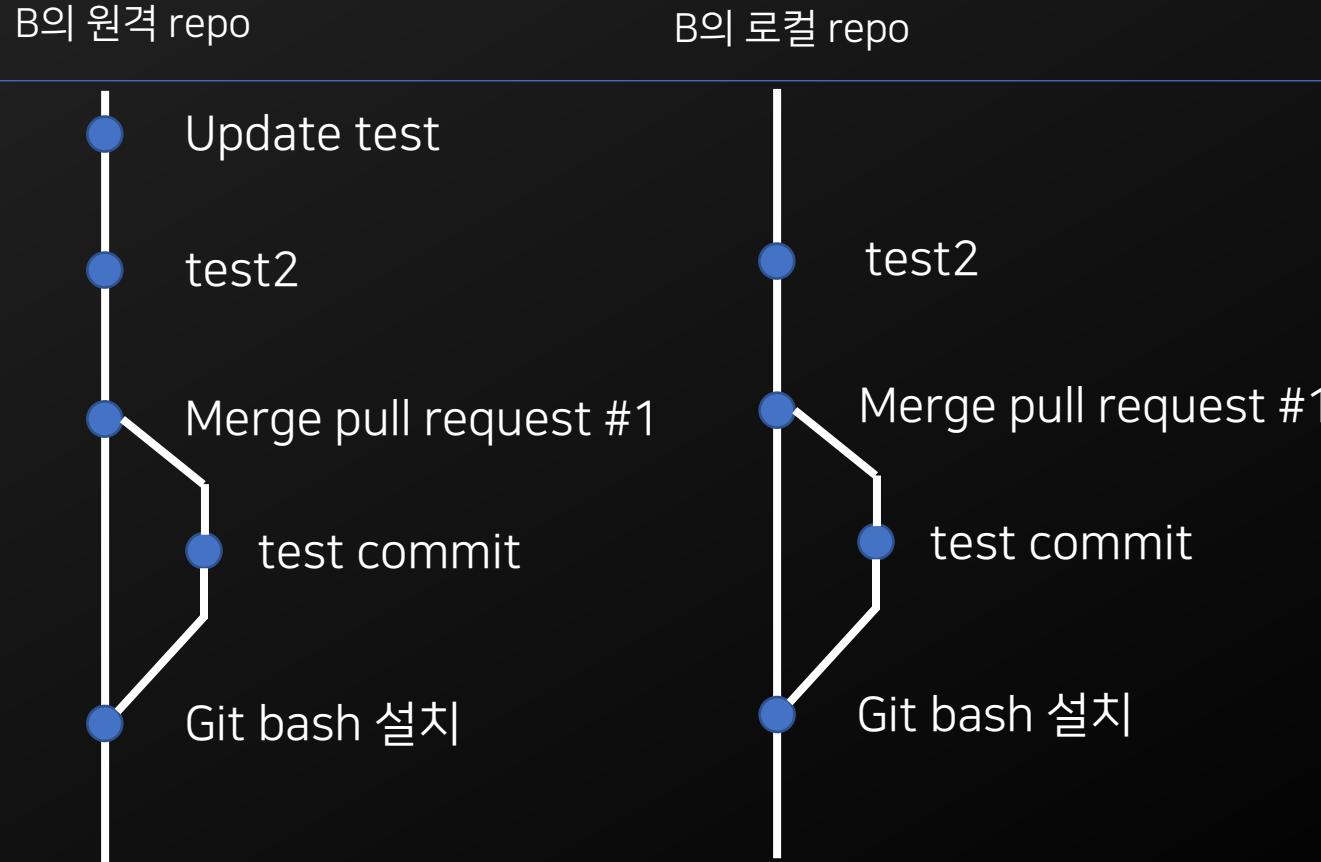
현재 원격 저장소 : 개인 repo

원격 저장소는 개인 repo도 가능하고 협업 repo도 가능함.

\* 단 협업 repo를 origin으로 하면 협업 repo에 pull request 없이 직접 푸쉬하게되어서 위험함

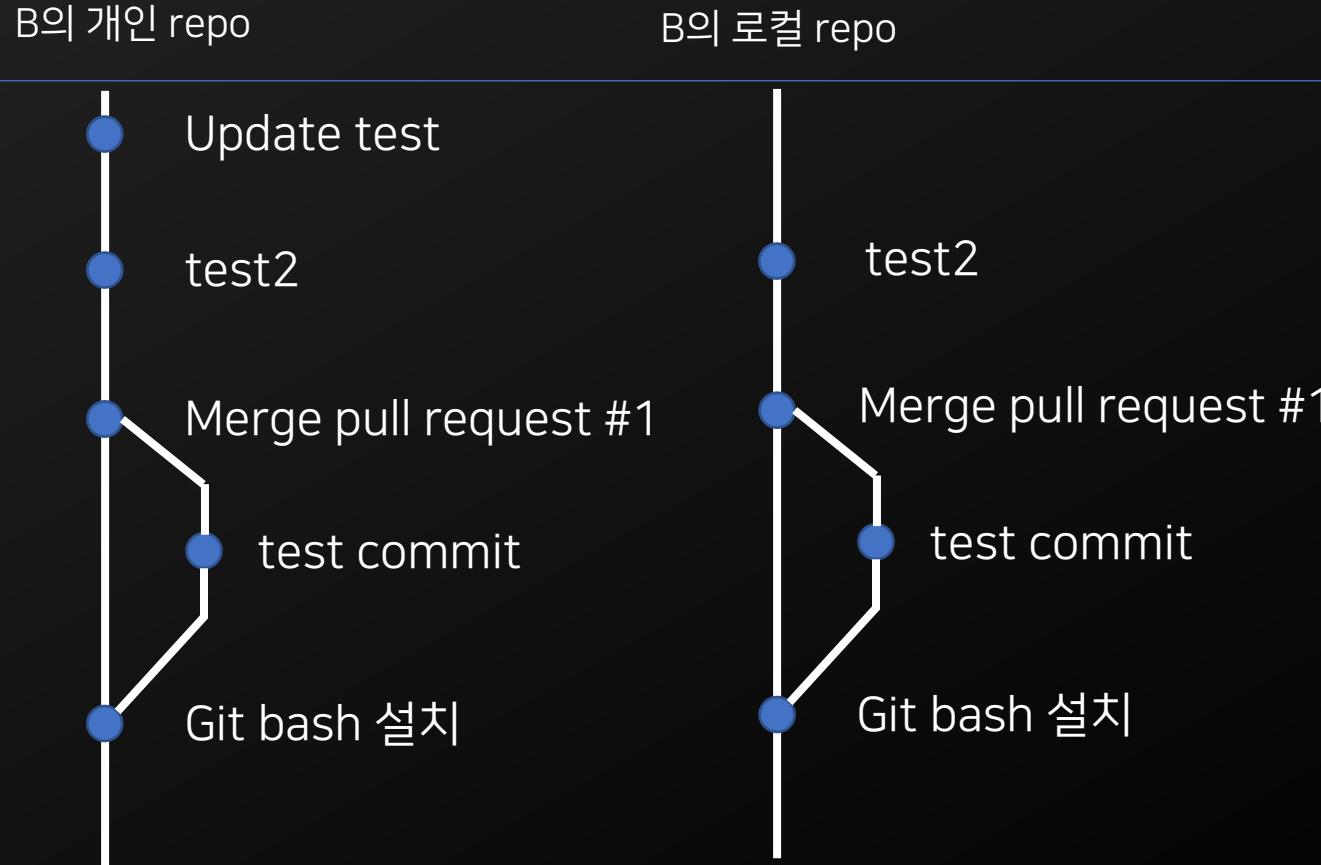


# 협업할 때 - 원격 저장소의 변경사항이 생겼을 때



원격 repo에 커밋이 생겼을때 로컬repo에 커밋을 반영하지 않고 작업하면? -> 커밋을 무시한채로 작업해서 오류생김

# 협업할 때 - 원격 저장소의 변경사항이 생겼을 때



그럼 pull을 해서 받고 작업해야하는가?

-> pull은 아예 덮어씌우는 작업. 로컬 repo에서 작업한것과 원격 저장소에서 작업한 파일이 덮어씌워지는 위험성 존재

# Fetch

---

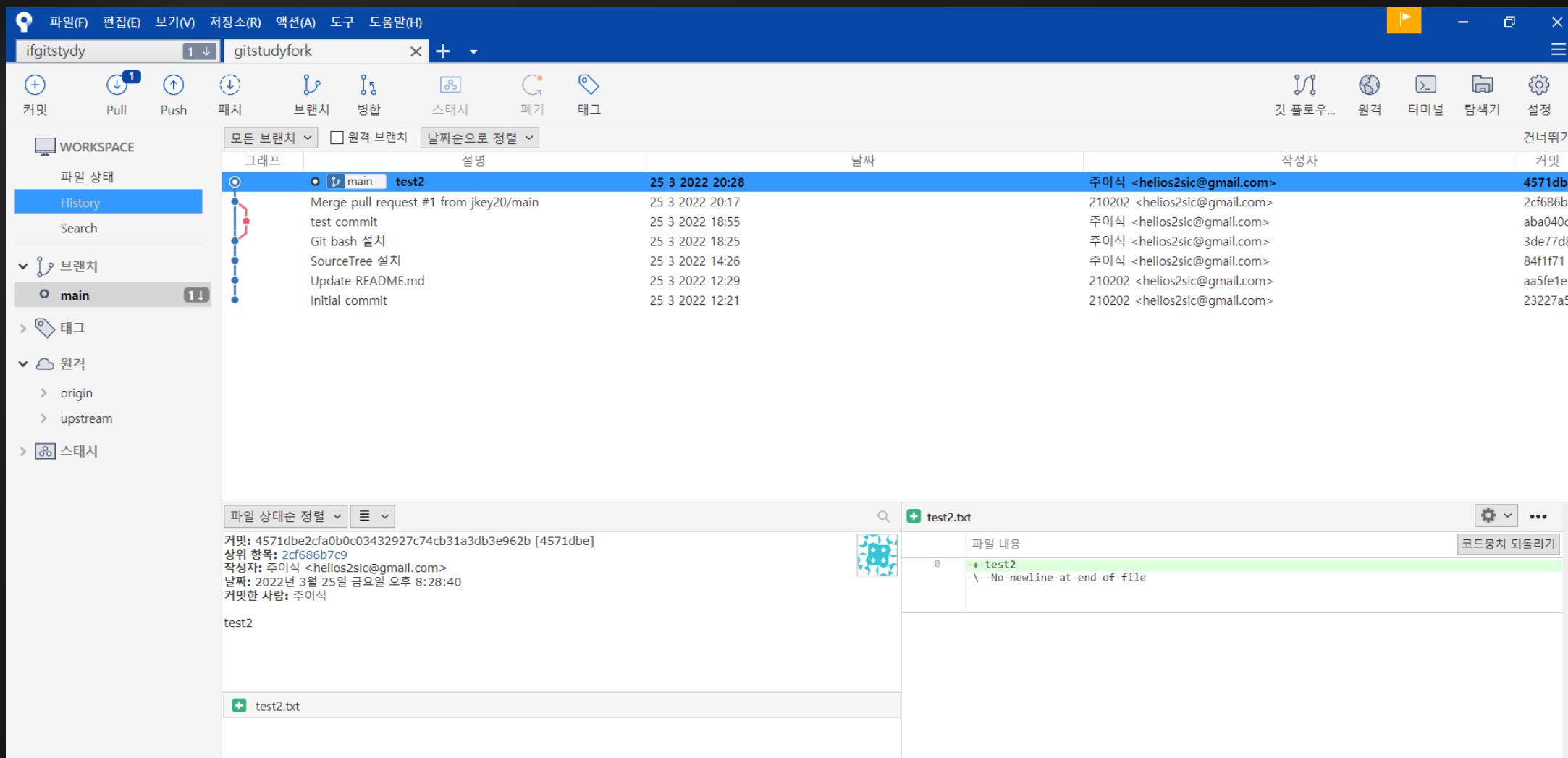
- 원격 repo의 커밋들을 로컬 repo로 가져오고 변경사항을 확인만 함.

pull과 다른점?

- pull은 원격 repo의 커밋들을 로컬 repo로 가져오고 병합(merge)를 함.

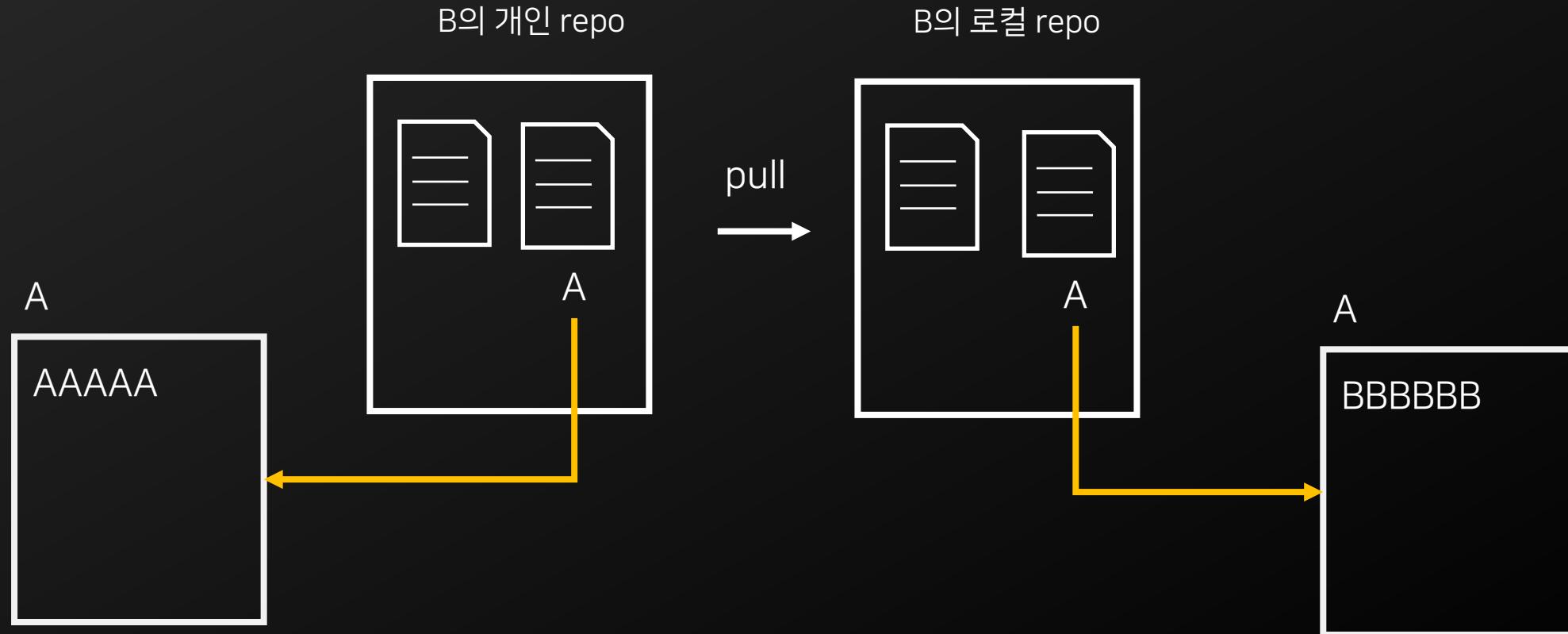
-> 패치를 사용해서 변경사항을 먼저 확인 로컬 repo에서 작업한것과 원격 저장소에서 작업한 파일이 덮어씌워지는걸 막음

# Fetch



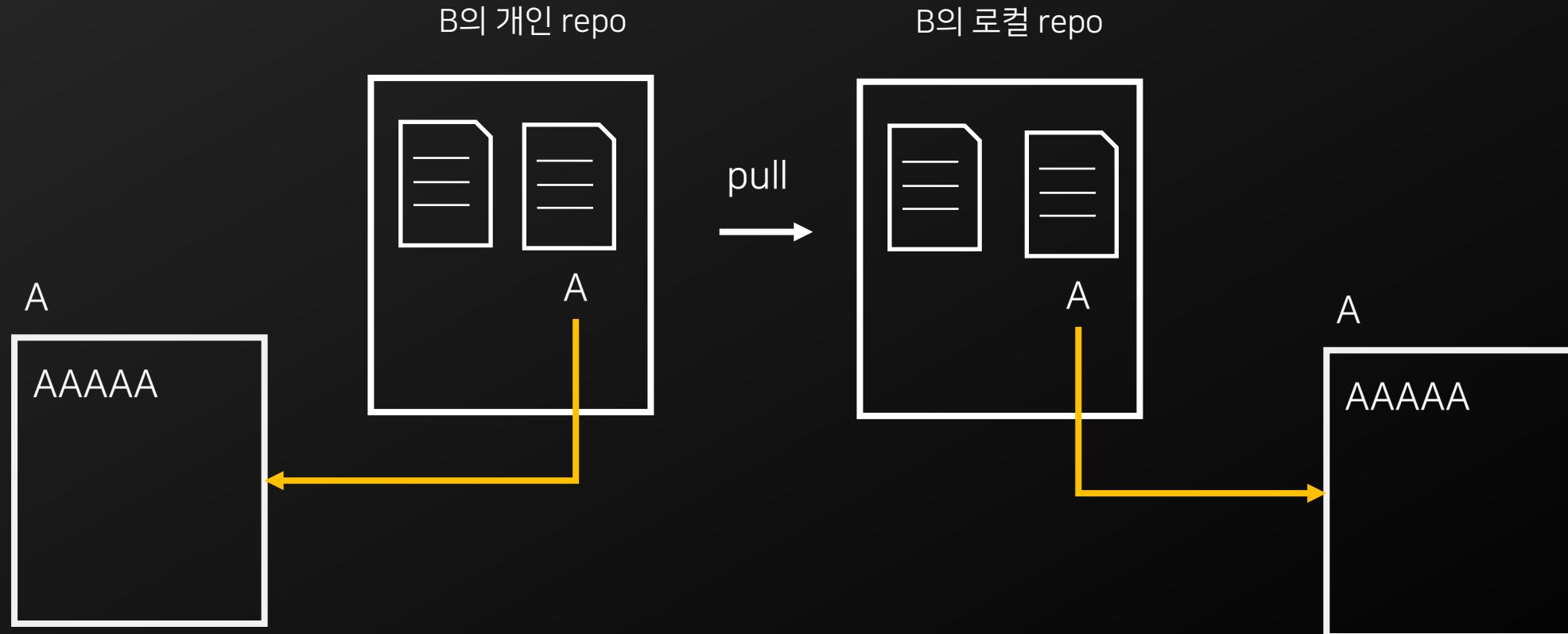
패치를 누르면 커밋을 가져와 변경사항을 확인하고 pull할게 있으면 알려줌

# 협업할 때 - 원격 저장소에 변경된 데이터가 부분적으로 겹칠때



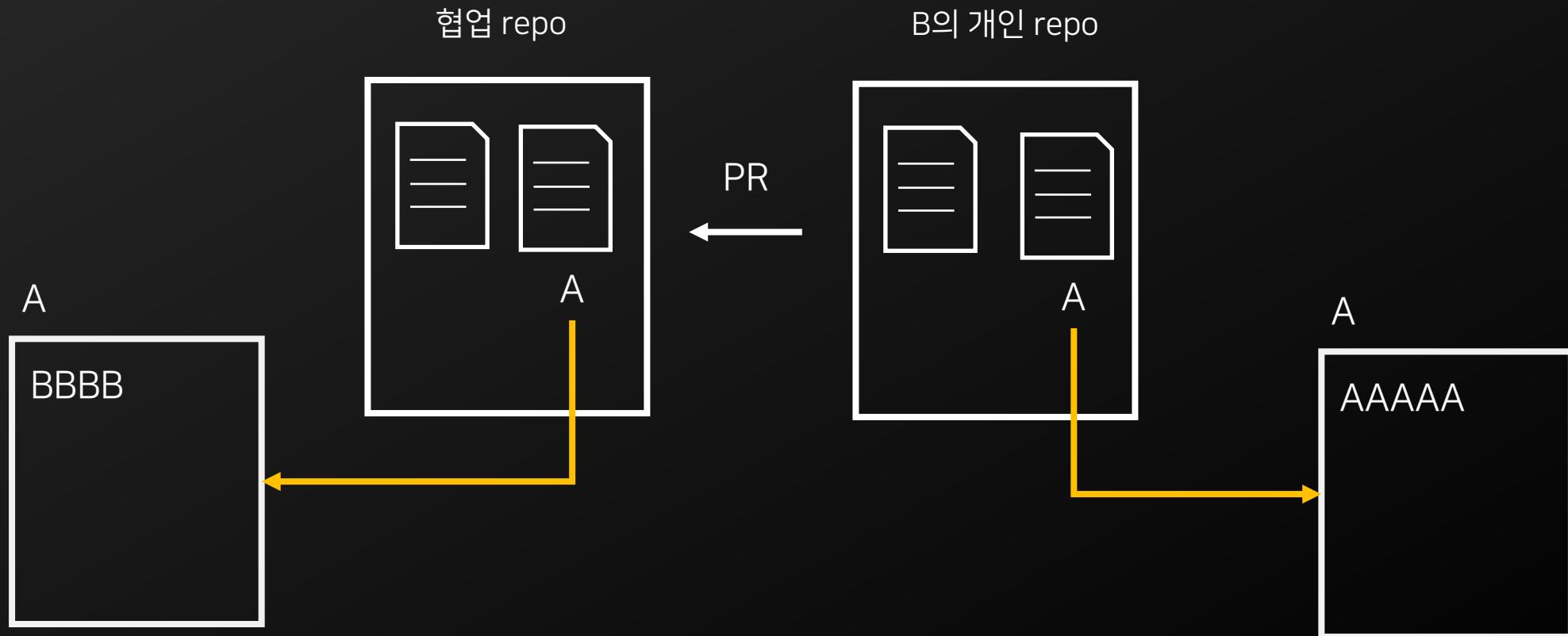
만약 A라는 파일의 내용이 로컬에는 다음과 같고 B의 개인 repo에서 다음과 같을때 pull하면?

# 협업할 때 - 원격 저장소에 변경된 데이터가 부분적으로 겹칠때



개인 repo의 파일이 로컬 repo로 변경됨!

# 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때



B의 개인 repo에서 겹치는 데이터를 PR보내면? 방금전과 마찬가지로 파일이 변경되나?

# 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

The screenshot shows a GitHub Pull Request (PR #2) for the repository `jkey20/Study-Git`. The PR is open and merges two commits from the branch `jkey20:main` into the main branch `sejonginterface:main`.

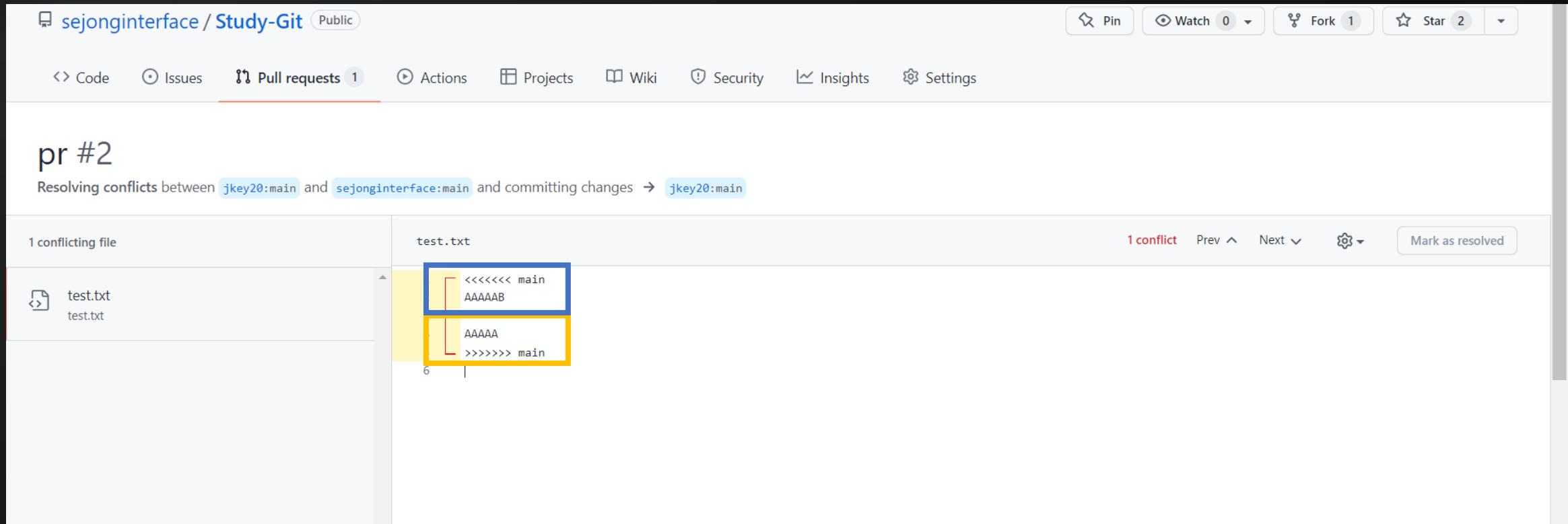
Key details from the PR interface:

- Commits:** 2 (2 commits added by jkey20)
- Files changed:** 1 (1 file: `test.txt`)
- Reviewers:** No reviews
- Assignees:** No one—assign yourself
- Labels:** None yet
- Projects:** None yet
- Milestone:** No milestone

A yellow box highlights a warning message: "This branch has conflicts that must be resolved. Use the web editor or the command line to resolve conflicts. Conflicting files". A yellow button labeled "Resolve conflicts" is also highlighted.

다음과 같이 Conflict가 발생했다고 merge할 수 없다고 나옴. -> Resolve conflicts를 눌러 Conflict를 해결해야함.

# 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼 때



파랑색 부분이 내가 수정한 데이터, 주황색이 원격 저장소의 데이터

# 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

The screenshot shows a GitHub pull request interface for a repository named 'sejonginterface / Study-Git'. The pull request is titled 'pr #2' and is resolving conflicts between the 'jkey20:main' branch and the 'sejonginterface:main' branch. The main content area displays a file named 'test.txt' with the following content:

```
1      AAAAAAB
2
3
```

There is one conflict, indicated by the red text '1 conflict' at the top right of the file view. A button labeled 'Mark as resolved' is also visible there, which is highlighted with a red box.

수정을 마치면 Mark as resolved를 누름.

# 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

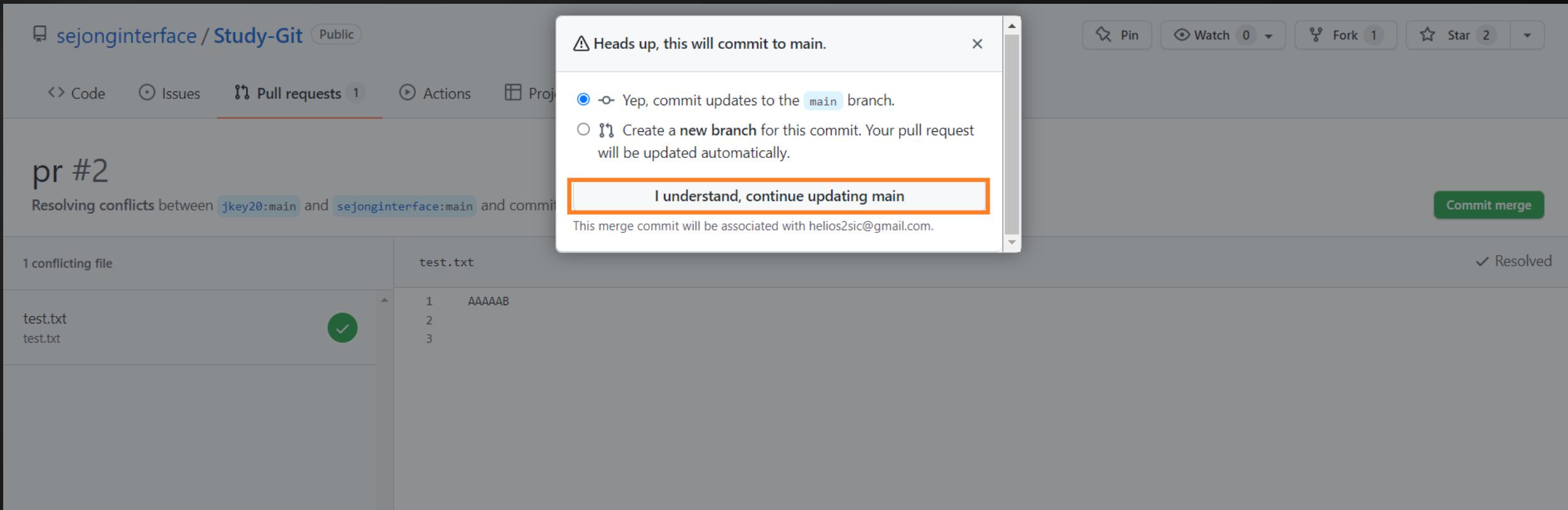
The screenshot shows a GitHub Pull Request (PR #2) interface. The repository is `sejonginterface / Study-Git`. The PR is resolving conflicts between `jkey20:main` and `sejonginterface:main`, specifically for the file `test.txt`. The conflict is resolved, indicated by a green checkmark icon next to the file name. The merged content of `test.txt` is shown as:

```
1 AAAAAAB  
2  
3
```

A green button labeled `Commit merge` is visible, with a red box highlighting it. The status of the conflict is marked as `✓ Resolved`.

Commit merge를 눌러 적용

# 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때



주황색 네모에 위치한 버튼을 누르면 완료.

## 협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼 때

**Open** jkey20 wants to merge 3 commits into [sejonginterface:main](#) from [jkey20:main](#)

Conversation 0 Commits 3 Checks 0 Files changed 1 +2 -1

jkey20 commented 18 hours ago

pr

jkey20 added 3 commits 18 hours ago

- o [Update test.txt](#) Verified 5a199f4
- o [Update test.txt](#) Verified 7b729b3
- o [Merge branch 'main' into main](#) Verified 1ba3974

Add more commits by pushing to the [main](#) branch on [jkey20/Study-Git](#).

 Continuous integration has not been set up  
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

 This branch has no conflicts with the base branch  
Merging can be performed automatically.

**Merge pull request** You can also open this in GitHub Desktop or view command line instructions.

Reviewers  
No reviews  
Still in progress? Convert to draft

Assignees  
No one—assign yourself

Labels  
None yet

Projects  
None yet

Milestone  
No milestone

Development  
Successfully merging this pull request may close these issues.  
None yet

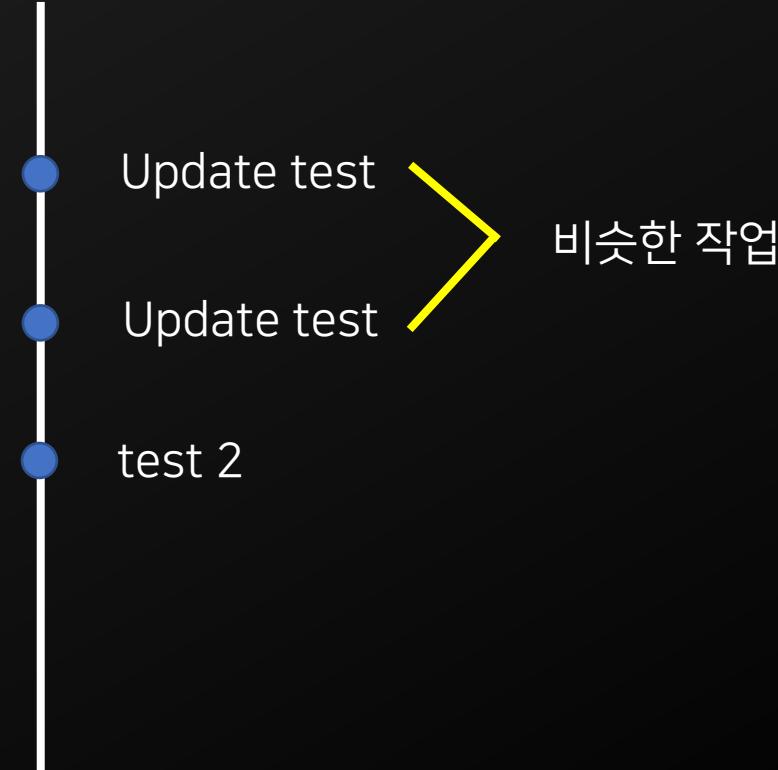
Notifications  
Customize  
Unsubscribe

Merge pull request가 정상적으로 나온다.

# 협업할 때 - 커밋 합치기

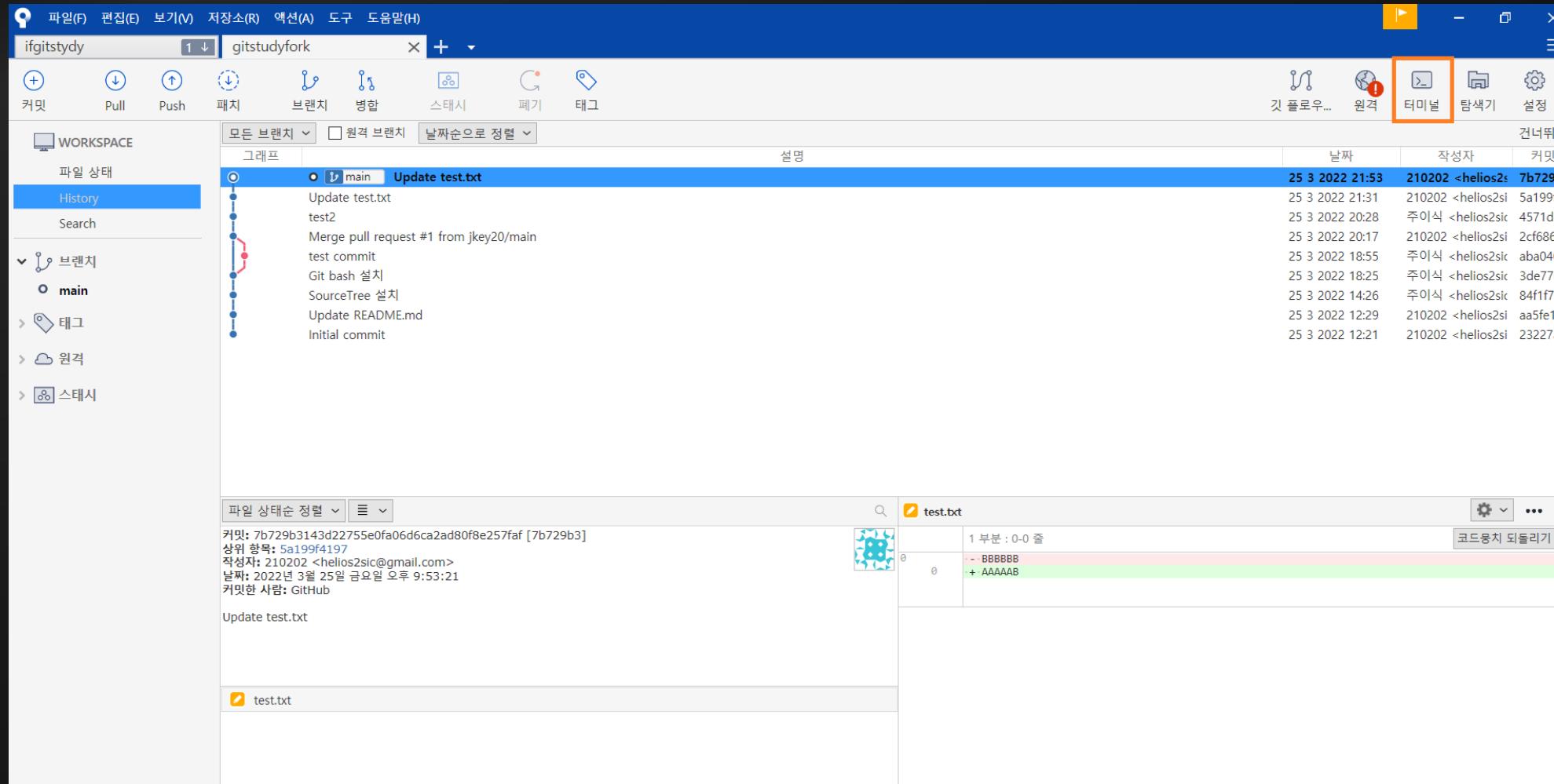
---

B의 로컬 repo



커밋을 여러개 했는데 비슷한 작업인 경우가 생김. 그냥 push를 하면 너무 커밋이 많아지는데 어떻게 할까?

# 협업할 때 - 커밋 합치기



SourceTree에서 터미널을 누름. 만약 터미널을 눌렀는데 아무런 반응이 없으면 터미널 옆에 탐색기를 누르고 열린 폴더에서 우클릭 -> Git bash Here 클릭

# 협업할 때 - 커밋 합치기

---

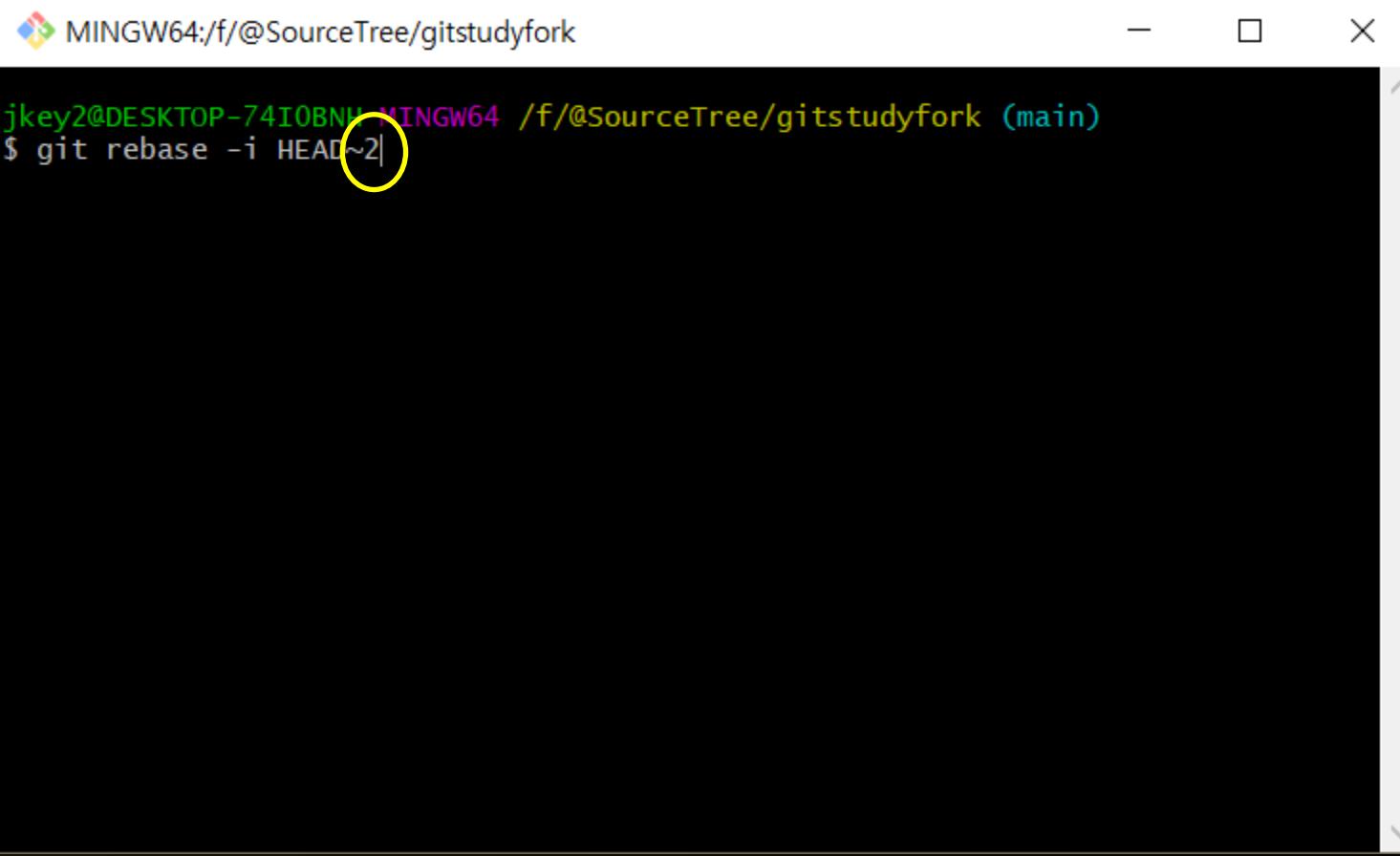


A screenshot of a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The window shows a command-line interface with the following text:  
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)  
\$ |

다음과 같이 검은창이 나오면 성공

# 협업할 때 - 커밋 합치기

---



A screenshot of a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The window shows a command-line interface with the following text:

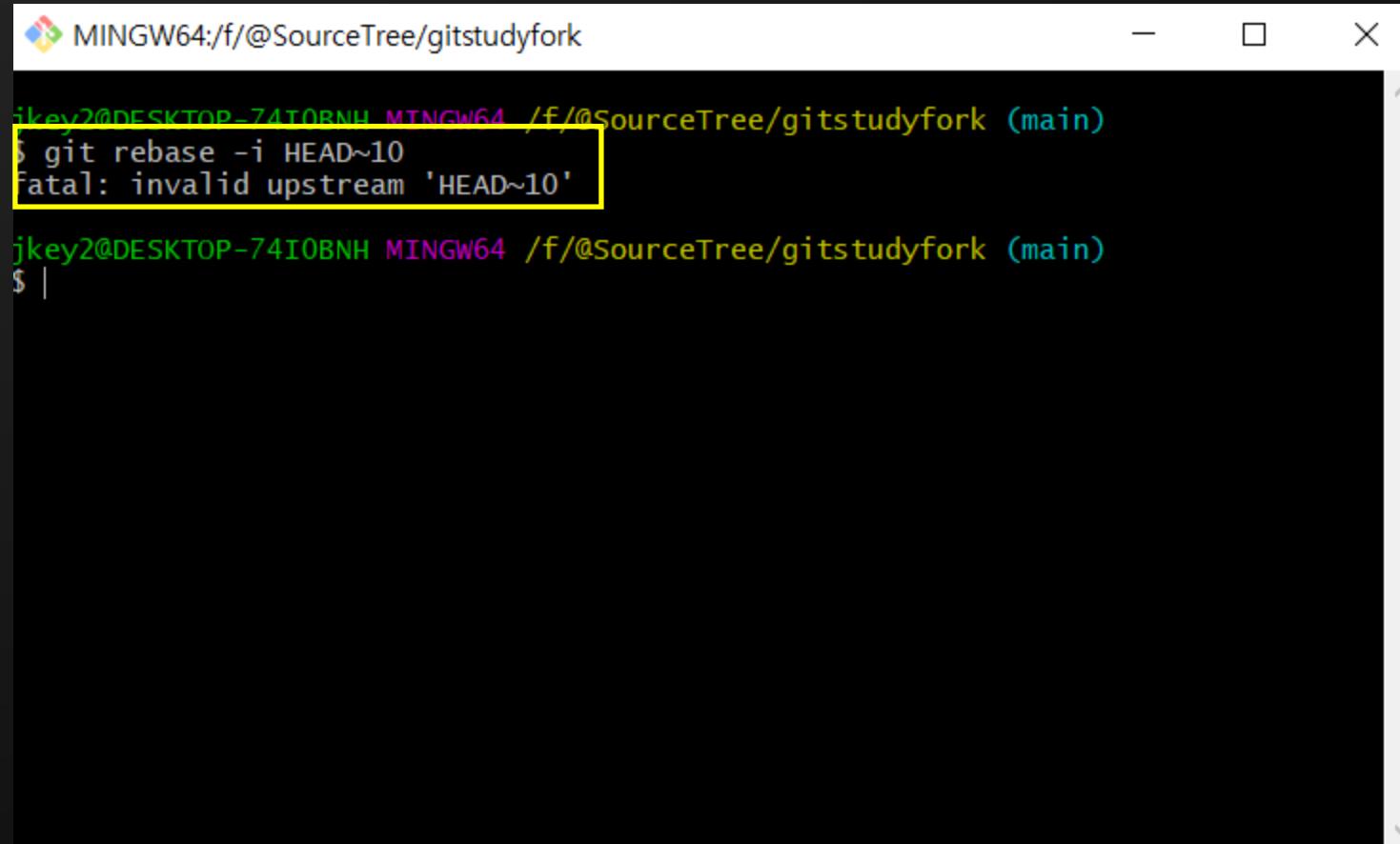
```
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git rebase -i HEAD~2|
```

The command "\$ git rebase -i HEAD~2" is highlighted with a yellow circle around the number "2".

다음과 같이 커맨드를 입력함 2는 합칠 커밋의 개수

# 협업할 때 - 커밋 합치기

---

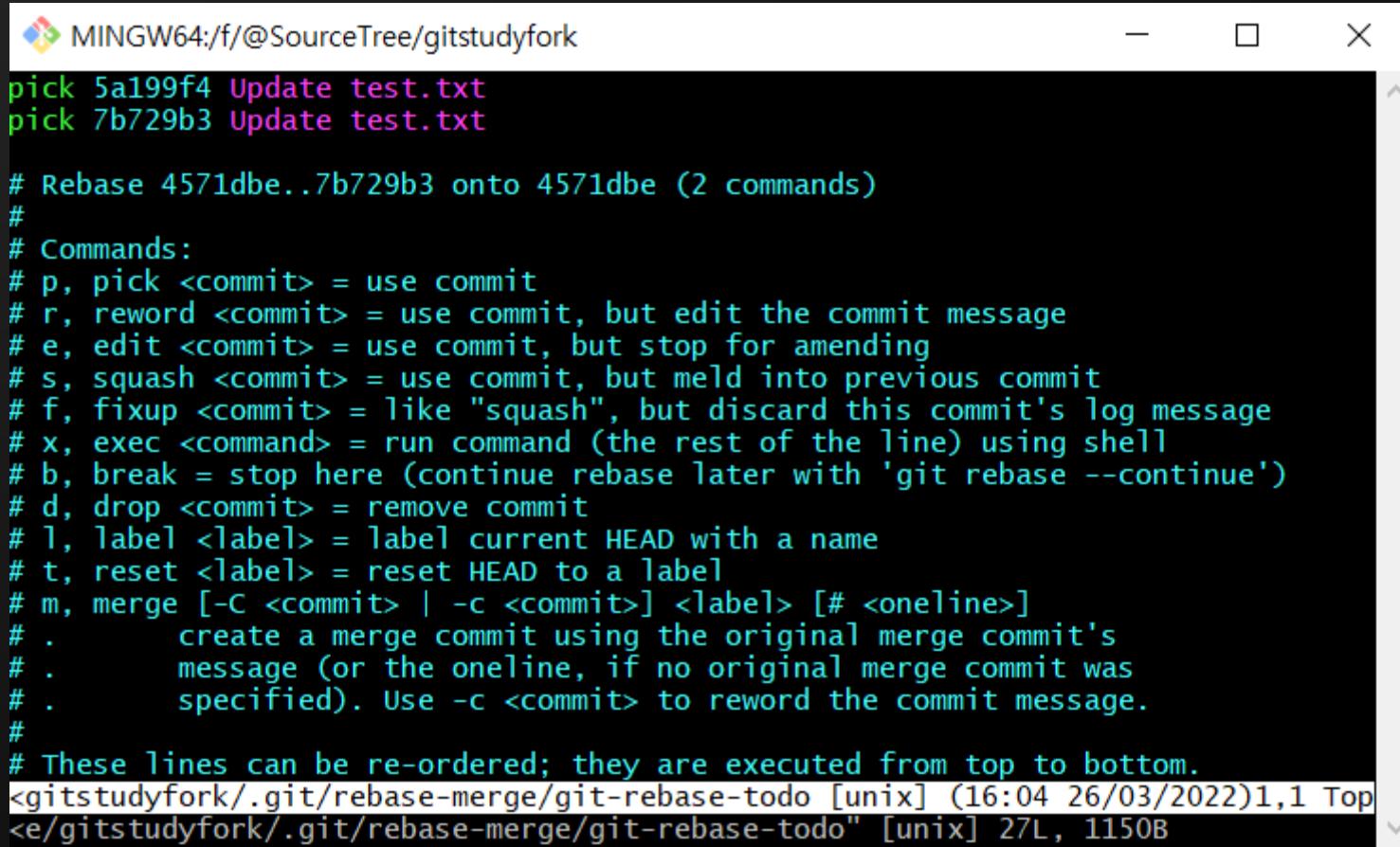


A screenshot of a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The window shows a command being run: "git rebase -i HEAD~10". The output indicates an error: "fatal: invalid upstream 'HEAD~10'". The entire error message is highlighted with a yellow box.

```
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git rebase -i HEAD~10
fatal: invalid upstream 'HEAD~10'
```

만약 합칠 커밋의 개수가 현재 존재하는 커밋보다 많다면 다음과 같이 나옴

# 협업할 때 - 커밋 합치기



The screenshot shows a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The window displays a "git rebase todo" list. At the top, there are two commits listed: "pick 5a199f4 update test.txt" and "pick 7b729b3 update test.txt". Below these, a comment "# Rebase 4571dbe..7b729b3 onto 4571dbe (2 commands)" is followed by a series of commit operation codes with their descriptions:

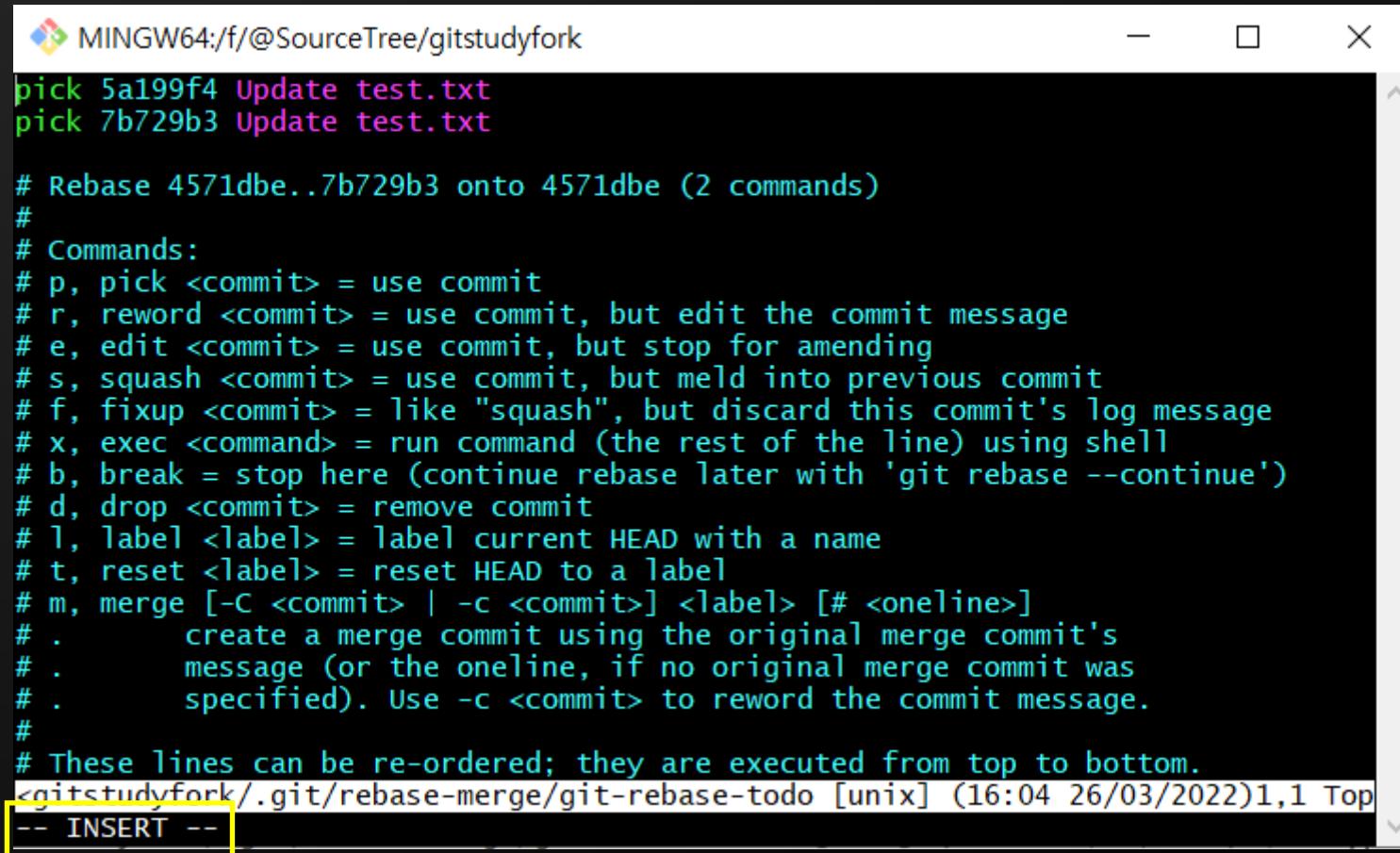
- # Commands:
  - # p, pick <commit> = use commit
  - # r, reword <commit> = use commit, but edit the commit message
  - # e, edit <commit> = use commit, but stop for amending
  - # s, squash <commit> = use commit, but meld into previous commit
  - # f, fixup <commit> = like "squash", but discard this commit's log message
  - # x, exec <command> = run command (the rest of the line) using shell
  - # b, break = stop here (continue rebase later with 'git rebase --continue')
  - # d, drop <commit> = remove commit
  - # l, label <label> = label current HEAD with a name
  - # t, reset <label> = reset HEAD to a label
  - # m, merge [-C <commit> | -c <commit>] <label> [<oneline>]
    - # . create a merge commit using the original merge commit's
    - # . message (or the oneline, if no original merge commit was
    - # . specified). Use -c <commit> to reword the commit message.
- #

# These lines can be re-ordered; they are executed from top to bottom.

<gitstudyfork/.git/rebase-merge/git-rebase-todo [unix] (16:04 26/03/2022)1,1 Top<e/gitstudyfork/.git/rebase-merge/git-rebase-todo" [unix] 27L, 1150B

\$ git rebase -i HEAD~2 를 입력하고 엔터를 누르면 다음과같이 나옴

# 협업할 때 - 커밋 합치기



MINGW64:/f/@SourceTree/gitstudyfork

```
pick 5a199f4 Update test.txt
pick 7b729b3 Update test.txt

# Rebase 4571dbe..7b729b3 onto 4571dbe (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [<oneline>]
# .
#   create a merge commit using the original merge commit's
#   message (or the oneline, if no original merge commit was
#   specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
<gitstudyfork/.git/rebase-merge/git-rebase-todo [unix] (16:04 26/03/2022)>1,1 Top
-- INSERT --
```

이 창에서 i를 누르면 다음과 같이 INSERT 혹은 끼워넣기라고 나옴

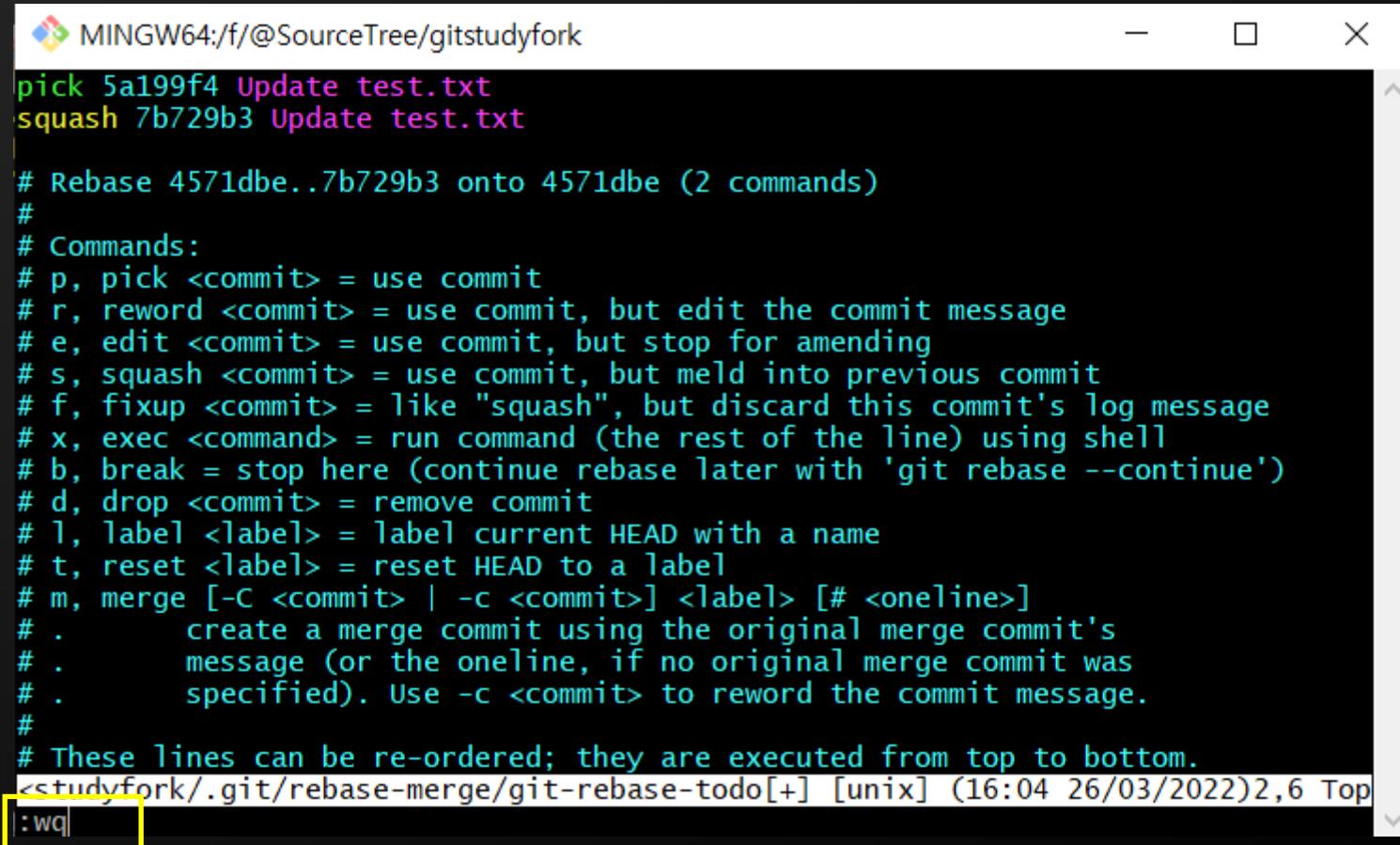
## 협업할 때 - 커밋 합치기

```
MINGW64:f/@SourceTree/gitstudyfork - □ X
pick 5a199f4 Update test.txt
squash 7b729b3 Update test.txt

# Rebase 4571dbe..7b729b3 onto 4571dbe (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .      create a merge commit using the original merge commit's
# .      message (or the oneline, if no original merge commit was
# .      specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
<studyfork/.git/rebase-merge/git-rebase-todo[+] [unix] (16:04 26/03/2022)2,7 Top
-- INSERT --
```

그 다음 아래쪽 커밋에 pick으로 되어있는걸 지우고 squash로 변경

# 협업할 때 - 커밋 합치기



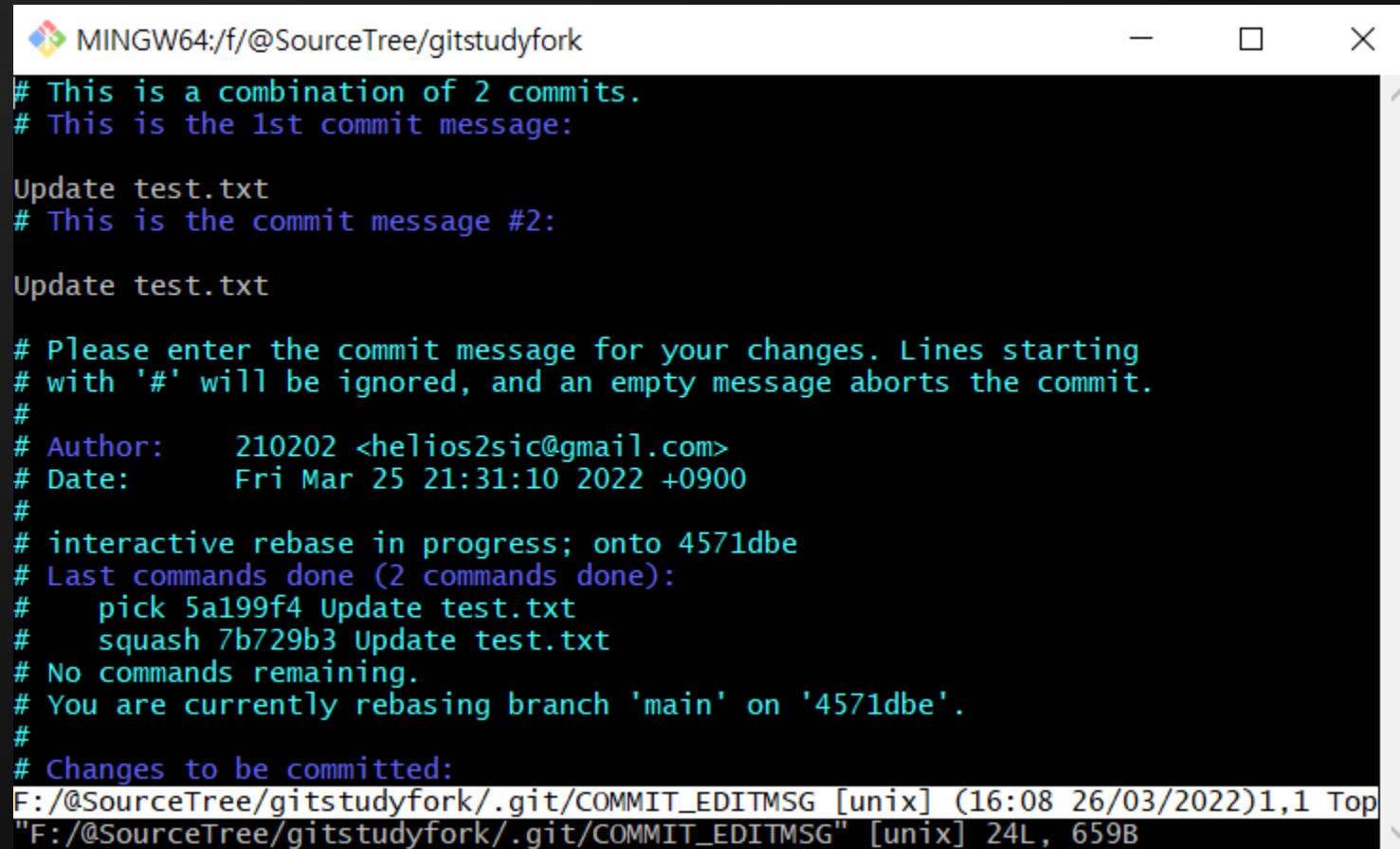
```
pick 5a199f4 update test.txt
squash 7b729b3 Update test.txt

# Rebase 4571dbe..7b729b3 onto 4571dbe (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .      create a merge commit using the original merge commit's
# .      message (or the oneline, if no original merge commit was
# .      specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
<studyfork/.git/rebase-merge/git-rebase-todo[+] [unix] (16:04 26/03/2022)2,6 Top
[:wq]
```

변경이후 esc를 누르면 INSERT가 없어지고 그 상태에서 :wq 입력 후 엔터

# 협업할 때 - 커밋 합치기

---



The screenshot shows a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The window displays a git commit message with the following content:

```
# This is a combination of 2 commits.
# This is the 1st commit message:

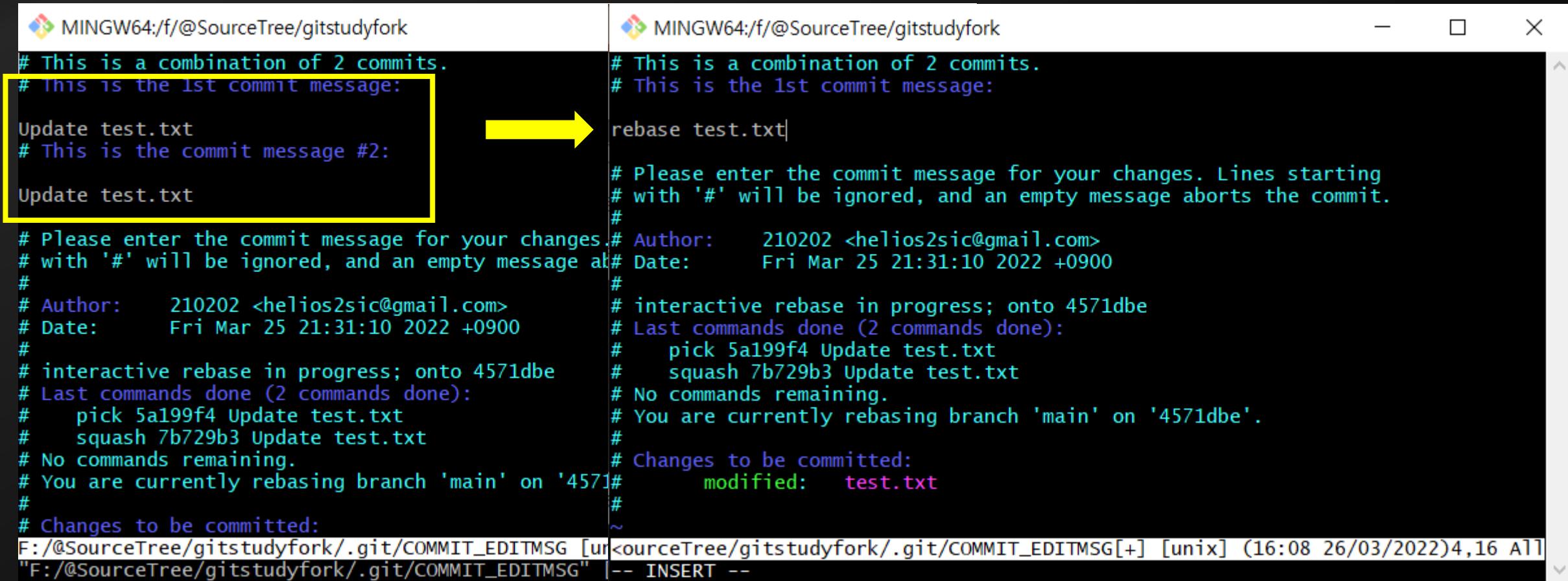
Update test.txt
# This is the commit message #2:

Update test.txt

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Author: 210202 <helios2sic@gmail.com>
# Date: Fri Mar 25 21:31:10 2022 +0900
#
# interactive rebase in progress; onto 4571dbe
# Last commands done (2 commands done):
#   pick 5a199f4 Update test.txt
#   squash 7b729b3 Update test.txt
# No commands remaining.
# You are currently rebasing branch 'main' on '4571dbe'.
#
# Changes to be committed:
F:/@SourceTree/gitstudyfork/.git/COMMIT_EDITMSG [unix] (16:08 26/03/2022)1,1 Top
"F:/@SourceTree/gitstudyfork/.git/COMMIT_EDITMSG" [unix] 24L, 659B
```

그럼 다음과 같이 나오고 마찬가지로 i를 눌러 INSERT가 뜨게 만듬

# 협업할 때 - 커밋 합치기



```
MINGW64:/f/@SourceTree/gitstudyfork
# This is a combination of 2 commits.
# This is the 1st commit message:

Update test.txt
# This is the commit message #2:

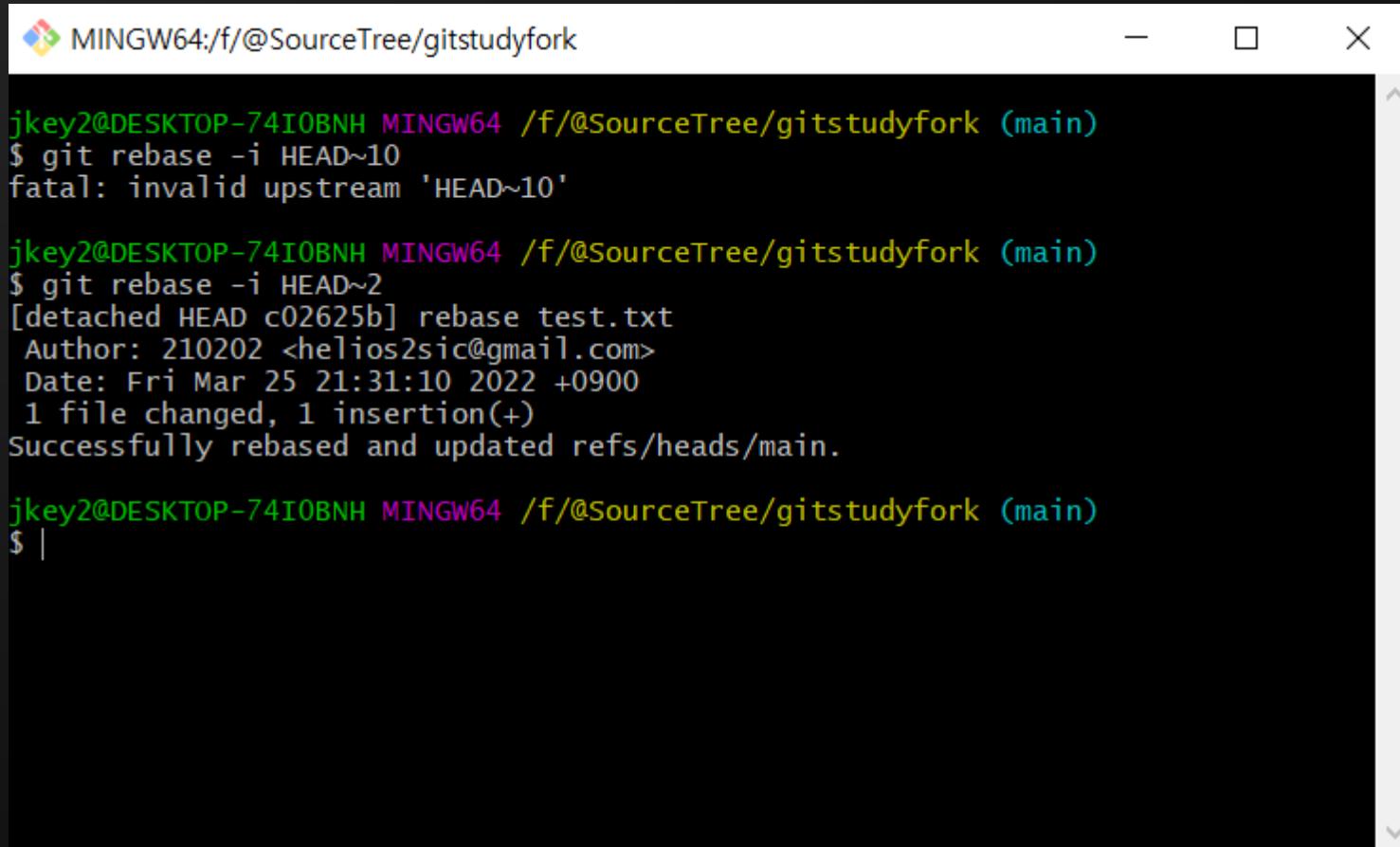
Update test.txt

# Please enter the commit message for your changes.
# with '#' will be ignored, and an empty message aborts the commit.
#
# Author: 210202 <helios2sic@gmail.com>
# Date: Fri Mar 25 21:31:10 2022 +0900
#
# interactive rebase in progress; onto 4571dbe
# Last commands done (2 commands done):
#   pick 5a199f4 Update test.txt
#   squash 7b729b3 Update test.txt
# No commands remaining.
# You are currently rebasing branch 'main' on '4571dbe'.
#
# Changes to be committed:
#       modified: test.txt
#
~
F:@SourceTree/gitstudyfork/.git/COMMIT_EDITMSG [ur<sourceTree/gitstudyfork/.git/COMMIT_EDITMSG[+] [unix] (16:08 26/03/2022)4,16 All]
"F:@SourceTree/gitstudyfork/.git/COMMIT_EDITMSG" -- INSERT --
```

노란색 네모 부분을 지우고 1st commit message를 자신이 원하는 커밋메시지로 변경

# 협업할 때 - 커밋 합치기

---



The screenshot shows a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The terminal output is as follows:

```
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git rebase -i HEAD~10
fatal: invalid upstream 'HEAD~10'

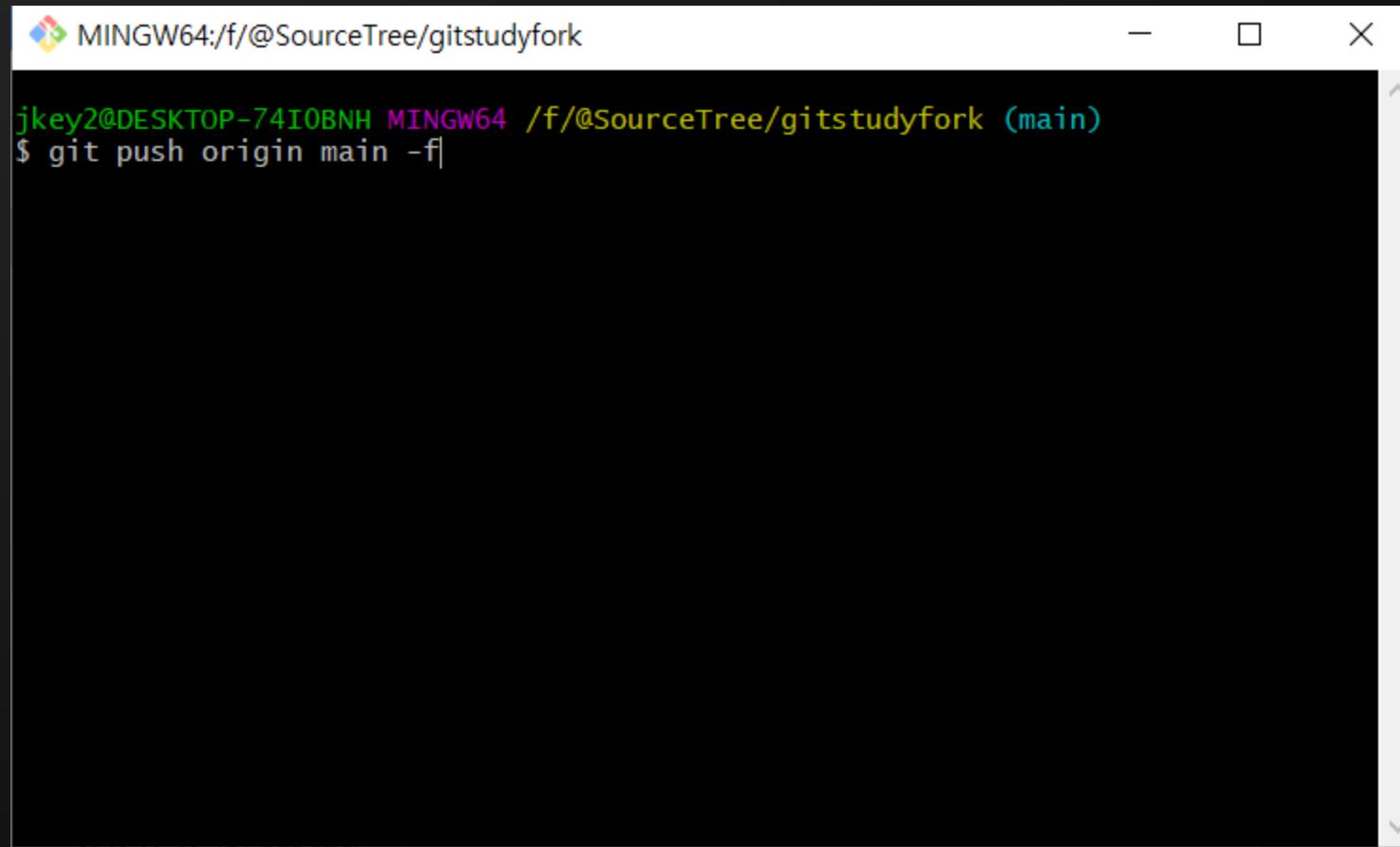
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git rebase -i HEAD~2
[detached HEAD c02625b] rebase test.txt
Author: 210202 <helios2sic@gmail.com>
Date: Fri Mar 25 21:31:10 2022 +0900
 1 file changed, 1 insertion(+)
Successfully rebased and updated refs/heads/main.

jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ |
```

esc -> :wq 후 엔터를 누르면 다음과 같이 나옴

# 협업할 때 - 커밋 합치기

---

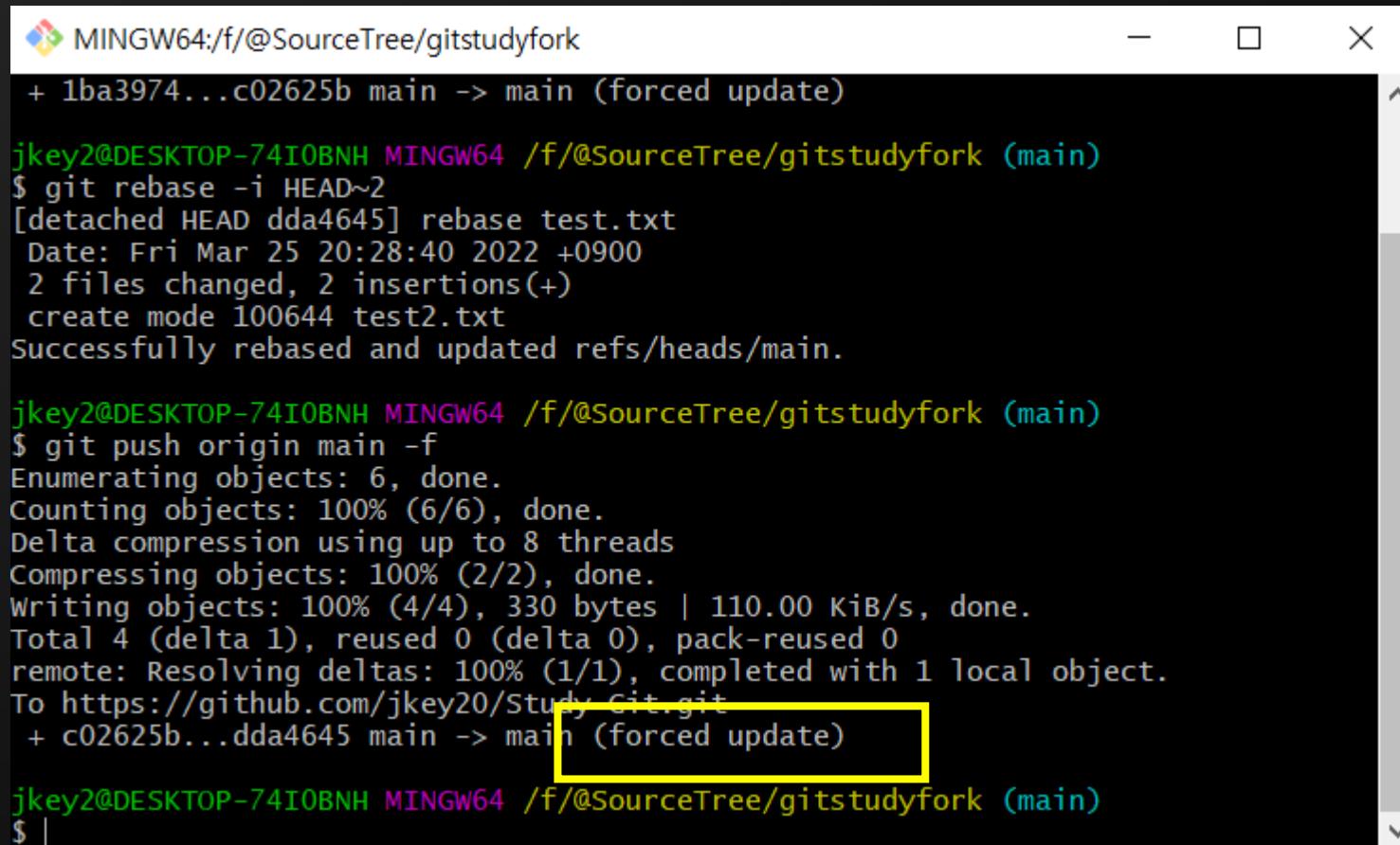


The screenshot shows a terminal window with the following details:

- Title Bar:** MINGW64:/f/@SourceTree/gitstudyfork
- User and Hostname:** jkey2@DESKTOP-74I0BNH MINGW64
- Working Directory:** /f/@SourceTree/gitstudyfork (main)
- Command Line:** \$ git push origin main -f

이제 변경사항을 푸시하기 위해서 다음과 같이 커맨드 입력

# 협업할 때 - 커밋 합치기



The screenshot shows a terminal window titled "MINGW64:/f/@SourceTree/gitstudyfork". The command \$ git rebase -i HEAD~2 was run, resulting in a detached HEAD state (dd4645). The user then rebased the test.txt file, which added two new files: test2.txt and test3.txt. The command \$ git push origin main -f was run to force-push the changes to the remote repository. The output shows the progress of the push, including object enumeration, compression, and writing to disk. The final status shows the main branch has been updated with the rebased history.

```
+ 1ba3974...c02625b main -> main (forced update)

jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git rebase -i HEAD~2
[detached HEAD dda4645] rebase test.txt
  Date: Fri Mar 25 20:28:40 2022 +0900
  2 files changed, 2 insertions(+)
    create mode 100644 test2.txt
Successfully rebased and updated refs/heads/main.

jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git push origin main -f
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 330 bytes | 110.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jkey20/Study_Git.git
 + c02625b...dda4645 main -> main (forced update)

jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ |
```

다음과 같이 나오면 완료

# Branch

---

Branch(브랜치)란?

- 동시에 다양한 작업을 하기 위해서 현재 버전의 Branch를 만들어 다른 작업을 이어나가는 것

우리가 지금까지 배운대로만하면 협업할때 모든 작업들을 계속 push pull하면서 해줘야해서 번거로움.

또한 A,B,C 기능을 만들다고 했을 때 A,B,C 기능을 각자 작업할때 커밋들이 순서대로 있지 않고 너무 많아져서 정렬하기 힘듬.

그래서 branch를 통해서 현재 작업의 분기를 만들어서 작업함.

모든 repo는 처음 생성하면 main 브랜치가 존재.

# Branch

---

예시 상황) A는 read, B는 write 기능을 만들려고 한다.

협업 repo

---

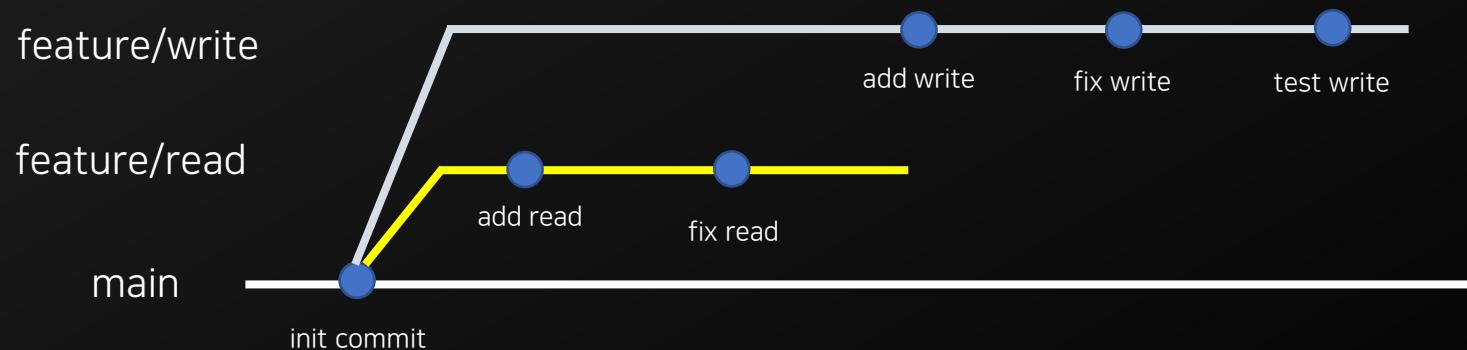


브랜치를 사용하지 않으면 다음과 같이 커밋들이 섞여있어서 알아보기 힘듬.

# Branch

예시 상황) A는 read, B는 write 기능을 만들려고 한다.

협업 repo

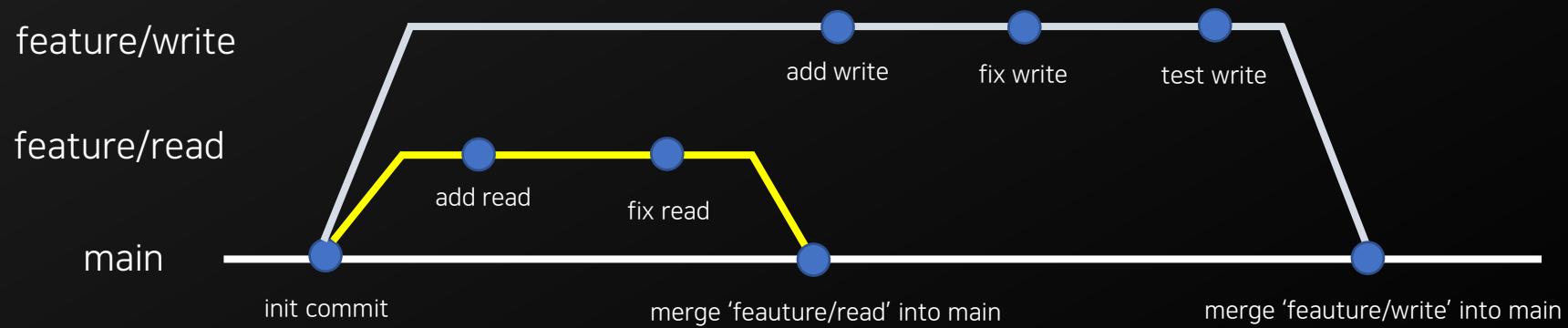


브랜치를 나누면 다음과 같이 커밋을 한 시간순서대로 분류되고 기능에 맞게 분류되어서 보기편함

# Branch

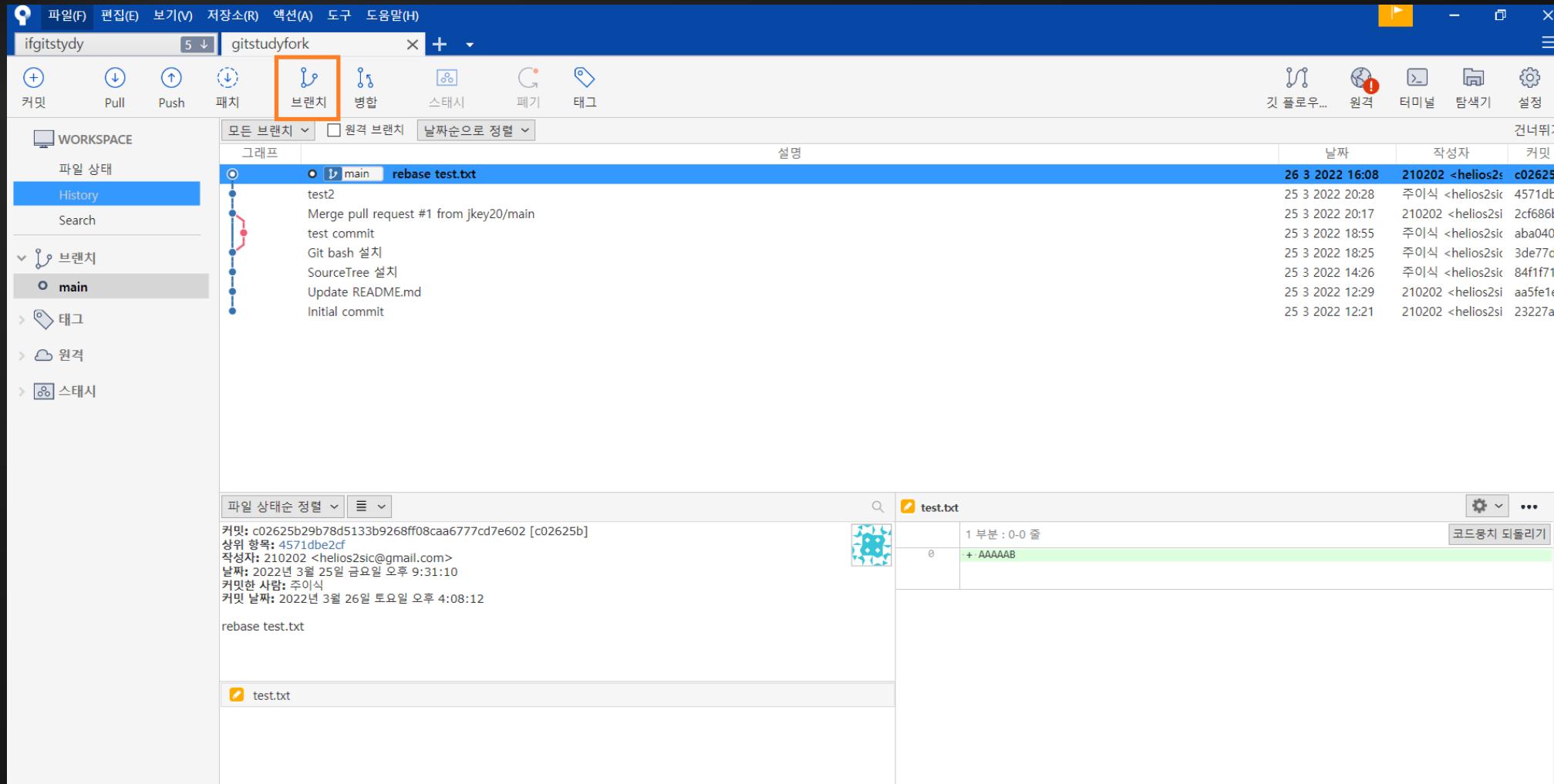
브랜치에서 작업한것을 통합할때 merge를 사용한다.

협업 repo



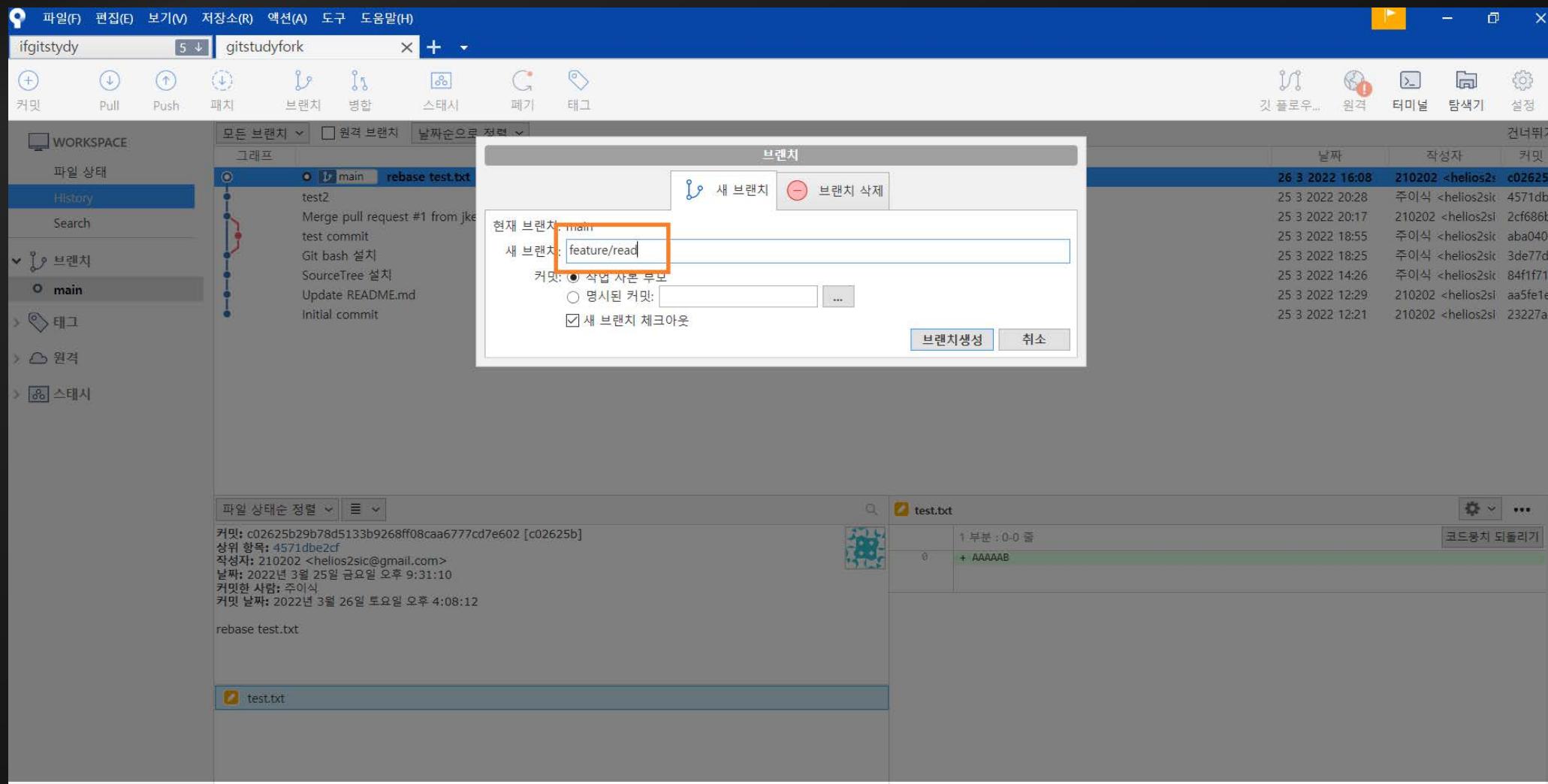
pull request -> merge를 통해서 다음과 같이 병합하게 된다. 머지를 하게되면 머지를 했다는 커밋이 생기게 된다.

# 사용방법 - Branch



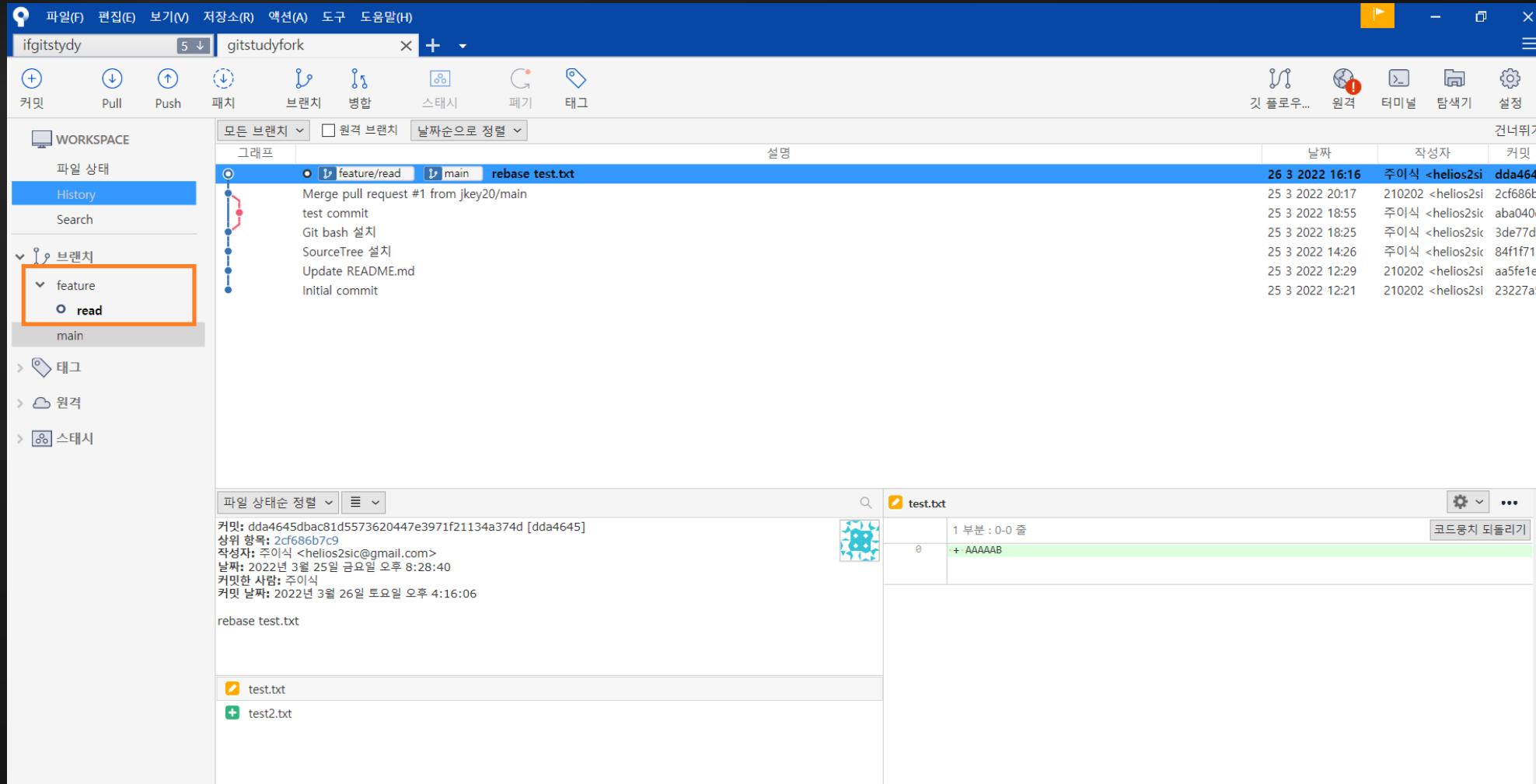
브랜치 버튼을 누른다.

# 사용방법 - Branch



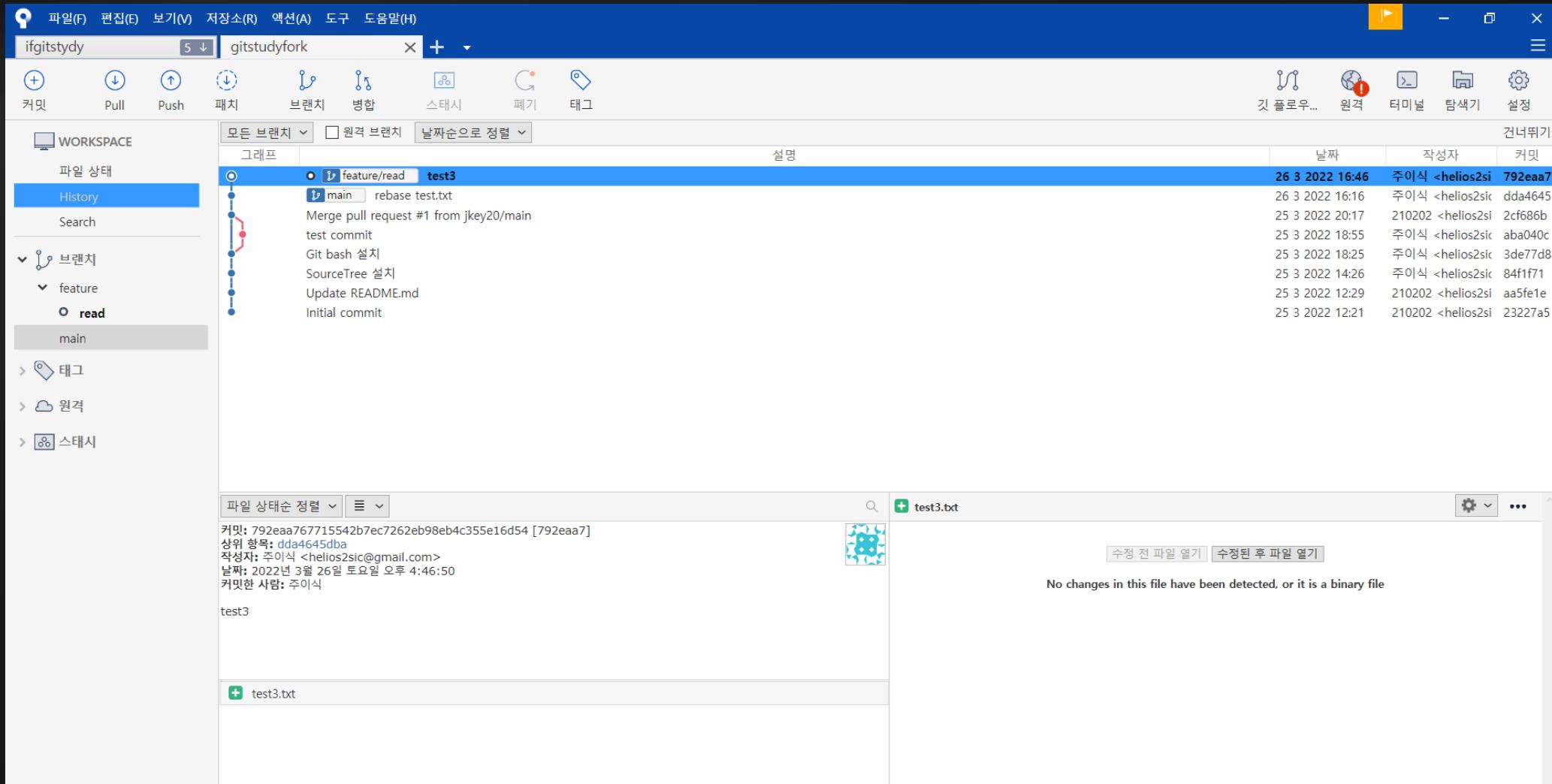
브랜치의 이름을 입력하고 브랜치 생성을 누른다.

# 사용방법 - Branch



그러면 왼쪽에 브랜치가 생성된 모습을 볼 수 있다.

# 사용방법 - Branch

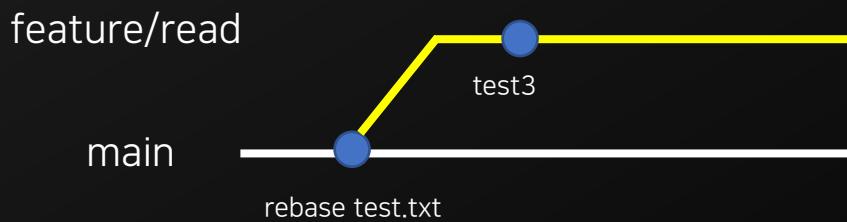


그 다음 테스트용으로 탐색기를 눌러 test3.txt 파일을 만들고 커밋을 해서 다음과 같이 나오게 해보자.

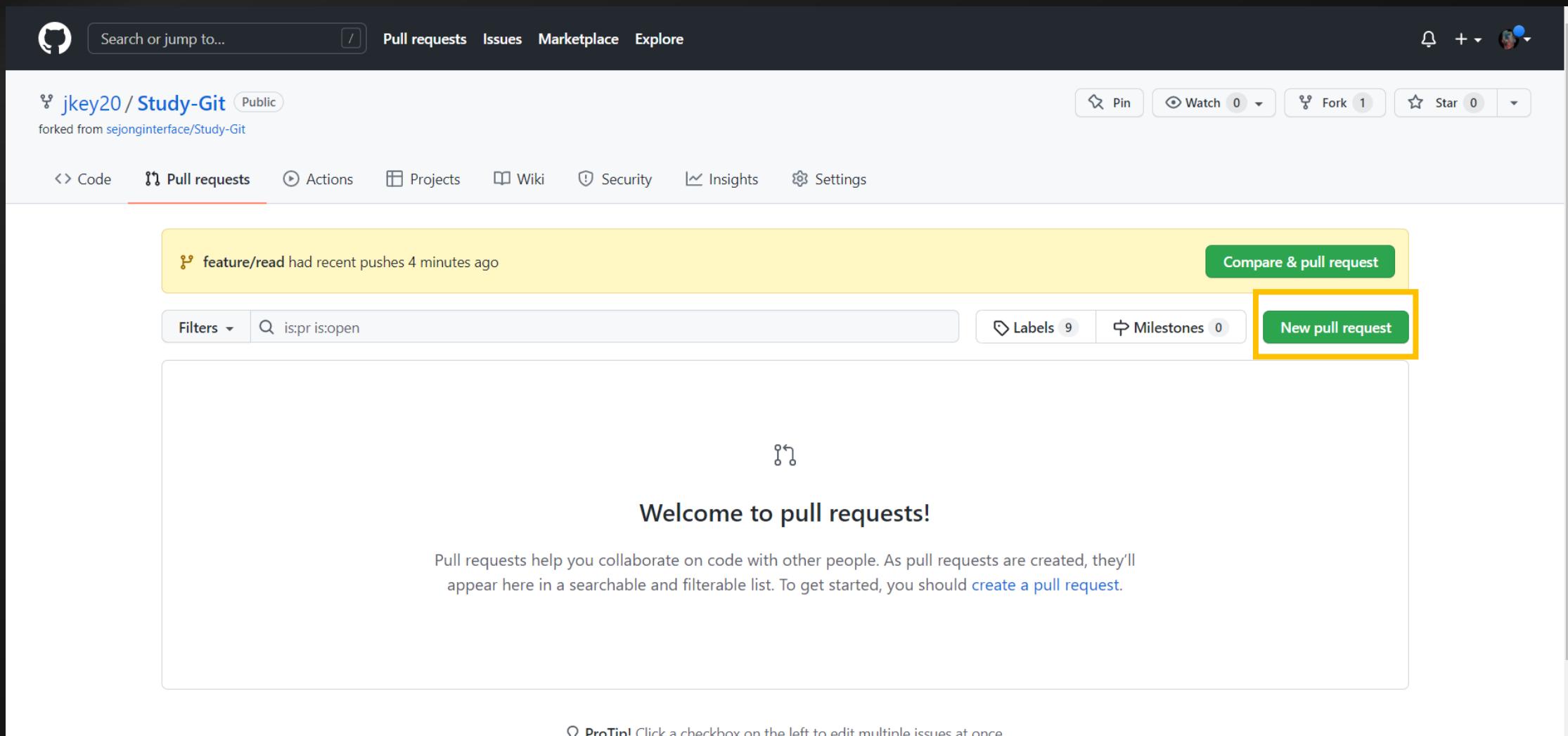
# 사용방법 - Branch

현재 상태는 다음과 같다. 이 상태에서 merge를 해보자.

B 개인 repo



# 사용방법 - Branch



The screenshot shows a GitHub repository page for 'jkey20 / Study-Git'. The 'Pull requests' tab is selected. At the top, there is a yellow notification bar with the message: 'feature/read had recent pushes 4 minutes ago'. To the right of this bar is a green 'Compare & pull request' button. Below the notification bar is a search bar with the query 'is:pr is:open' and a 'Filters' dropdown. To the right of the search bar are buttons for 'Labels 9' and 'Milestones 0'. A prominent green button labeled 'New pull request' is located on the right side of the search bar, with a yellow box and border highlighting it. The main content area displays a large 'Welcome to pull requests!' message with a small icon above it. Below this message, a paragraph explains what pull requests are and how to create one. At the bottom of the page, there is a 'ProTip!' link.

feature/read had recent pushes 4 minutes ago

Compare & pull request

Filters is:pr is:open

Labels 9

Milestones 0

New pull request

Welcome to pull requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).

ProTip! Click a checkbox on the left to edit multiple issues at once.

개인 repo의 pull requests로 가서 new pull request를 누른다.

# 사용방법 - Branch

The screenshot shows the GitHub interface for comparing changes between repositories. At the top, it displays the repository path `sejonginterface / Study-Git` and various navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main section is titled "Comparing changes" and instructs the user to choose two branches to see what's changed or to start a new pull request. It shows the base repository as `sejonginterface/Study-Git` and the head repository as `jkey20/Study-Git`, both set to the "main" branch. A green "Create pull request" button is visible. Below this, a list of commits is shown, starting with a commit from `jkey20` on March 26, 2022, titled "rebase test.txt". The commit details show it was committed 38 minutes ago by `jkey20`. At the bottom, it indicates "Showing 2 changed files with 2 additions and 0 deletions." A yellow box highlights the "jkey20/Study-Git" repository selection area.

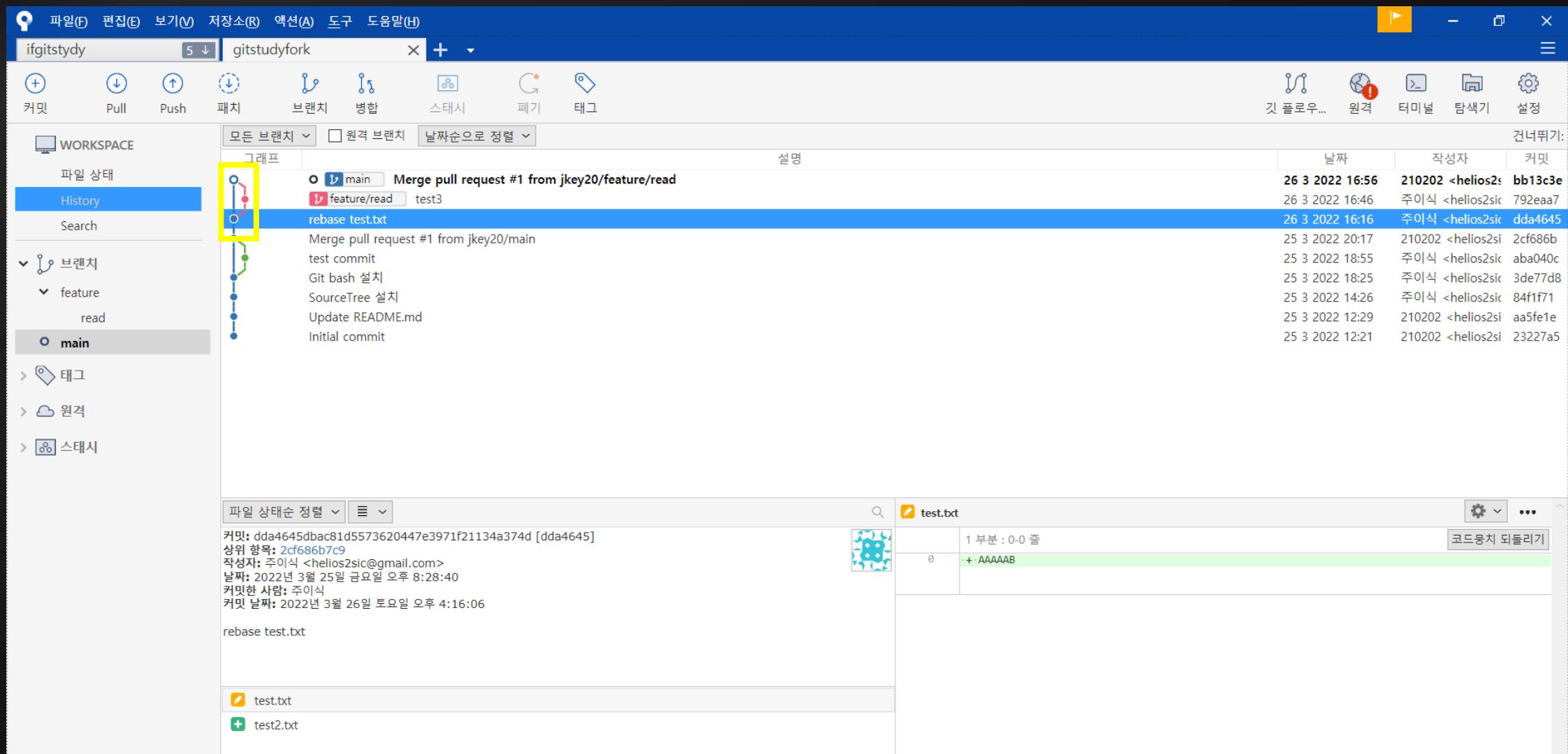
여기서 개인 repo를 선택해준다.

# 사용방법 - Branch

The screenshot shows the GitHub interface for comparing changes between two branches. At the top, there are navigation links: Code (highlighted in red), Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the title "Comparing changes" is displayed, followed by the instruction: "Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks." A modal window titled "Choose a head ref" is open, showing a search bar "Find a branch" and a list of branches. The "main" branch is selected as the base branch (indicated by a checkmark). The "feature/read" branch is selected as the compare branch (indicated by a blue border). A message in the modal states: "here isn't anything to compare. use two different branch names to get a valid comparison. check out some of these sample comparisons." Below the modal, the main content area shows the comparison results: "Showing 0 changed files with 0 additions and 0 deletions." At the bottom right, there are "Split" and "Unified" buttons. The footer contains the GitHub logo and links to Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

그다음 base는 합쳐질 곳, compare은 합치는 branch를 선택해주고 pr을 진행한다.

# 사용방법 - Branch



pr을 만들고 merge한뒤 SourceTree로 돌아와서 main 브랜치를 선택하고 pull을 받으면 다음과 같이 브랜치가 병합된 것을 볼 수 있다.