

Github를 통한 협업의 기본 배우기 with SourceTree




Github는 무엇일까?

소프트웨어 개발 프로젝트를 위한 코드 관리 서비스




SourceTree는 무엇일까?


Git을 더 쉽게 사용하기위한 툴





사용방법 - Fork






[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)


 [sejonginterface / Study-Git](#) Public


 Pin  Watch 0  Fork 0  Star 0


[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


 main  1 branch  0 tags

[Go to file](#) [Add file](#) [Code](#)

 jkey20 SourceTree 설치 84f1f71 3 hours ago 3 commits

 README.md Update README.md 5 hours ago

 SourceTree 설치.md SourceTree 설치 3 hours ago


README.md 


Study-Git


깃허브 스터디


About

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



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 스타디

Fork Study-Git



Where should we fork Study-Git?



jkey20

Pin

Watch 0

Fork 0

Star 2

Code

About



깃허브 스타디

Readme

2 stars

0 watching

0 forks


Releases

No releases published




[Create a new release](#)


Packages

사용방법 - Fork







[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 up to date with [sejonginterface/Study-Git:main](#). [Contribute](#) [Fetch upstream](#)

 **jkey20** Git bash 설치 3de77d8 5 minutes ago 🕒 4 commits


 Git bash 설치.md	Git bash 설치	5 minutes ago
 README.md	Update README.md	6 hours ago
 SourceTree 설치.md	SourceTree 설치	4 hours 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](#)

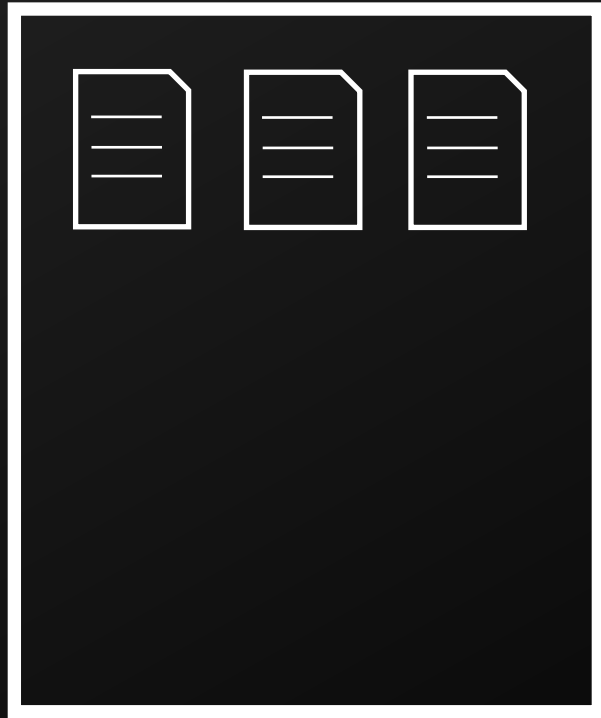
No packages published

Fork

Fork란?

- 협업 repo를 복사해서 개인 repo를 만듦

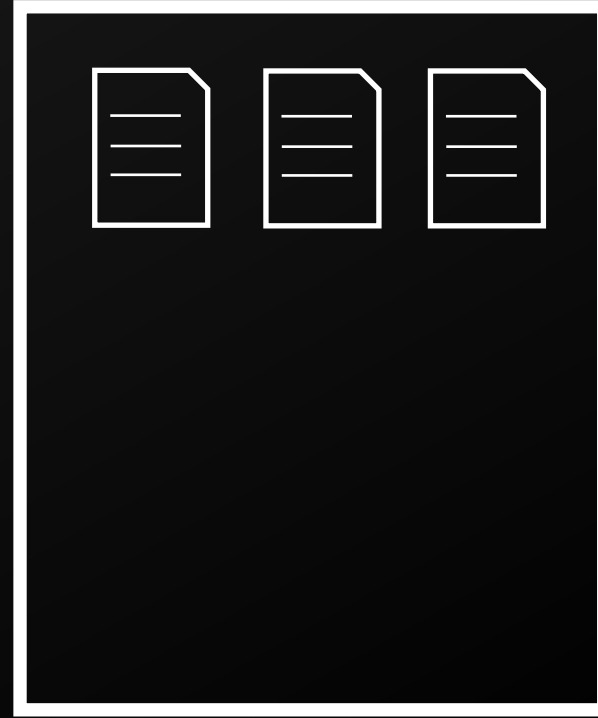
협업 repository(sejonginterface/Study-Git)




Fork






개인 repository([username]/Study-Git)




사용방법 - Clone





[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 up to date with [sejonginterface/Study-Git:main](#).

[Contribute](#) [Fetch upstream](#)

 [jkey20](#) [Git bash 설치](#) 3de77d8 11 minutes ago 🕒 4 commits

 Git bash 설치.md	Git bash 설치	11 minutes ago
 README.md	Update README.md	6 hours ago
 SourceTree 설치.md	SourceTree 설치	4 hours ago

README.md

Study-Git

깃허브 스터디

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



Search 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 up to date with [sejonginterface/Study-Git:main](#).

[jkey20](#) Git bash 설치

Git bash 설치.md [Git bash 설치](#)

README.md [Update README.md](#)

SourceTree 설치.md [SourceTree 설치](#)

README.md

Study-Git

깃허브 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

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

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

New tab


Local Remote Clone Add Create

Clone

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

소스 경로 / URL:

탐색

저장소 종류:  경로 / URL 노트

목적지 경로:

탐색

이름:

Local Folder:

[루트]

> 고급 옵션

클론

사용방법 - Clone

ifgitsydy

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

탐색

저장소 종류:  Git 저장소입니다

목적지 경로:

탐색

Study-Git

Local Folder:

[루트]

> 고급 옵션

! Please correct the following input errors:

Destination cannot be empty

클론

사용방법

ifgitstydy

gitstudyfork

+

-

커밋

Pull

Push

패치

브랜치

병합

스태시

폐기

태그

깃 플로우...

원격

터미널

탐색기

설정

WORKSPACE

파일 상태

History

Search

브랜치

main

태그

원격

스태시

모든 브랜치

원격 브랜치

날짜순으로 정렬

그래프	설명	날짜	작성자	커밋 ID
	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5f1e1
	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23221e1

파일 상태순 정렬

Git bash 설치.md

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 + ![image](https://user-images.githubusercontent.com/32587845/160089436-4d15d6c9-2935-49e5-8b
3 +
4 + ## 2. 설치
5 + ![image](https://user-images.githubusercontent.com/32587845/160090996-583580cc-aab4-480a-b4
6 +
7 + ## 3. 경로지정
8 + ![image](https://user-images.githubusercontent.com/32587845/160091045-e8def3b0-e712-4a2b-a6
9 +
10 + ## 4. 설치 옵션
11 + next 누르기
12 + ![image](https://user-images.githubusercontent.com/32587845/160091170-4040d343-0181-4919-89
13 +
14 + ## 5. 시작폴더 경로
```

Commit

커밋(Commit)이란?

- 코드의 변화를 시간순으로 기록하는 것

사용방법 - Commit

ifgitstudy

gitstudyfork

커밋

Pull

Push

패치

브랜치

병합

스테시

폐기

태그

WORKSPACE

파일 상태

History

Search

브랜치

main

태그

원격

스테시

모든 브랜치

원격 브랜치

날짜순으로 정렬

그래프	설명	날짜	작성자	커밋
	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f7
	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5fe
	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23227

파일 상태순 정렬

Git bash 설치.md

코딩용치 되돌리기

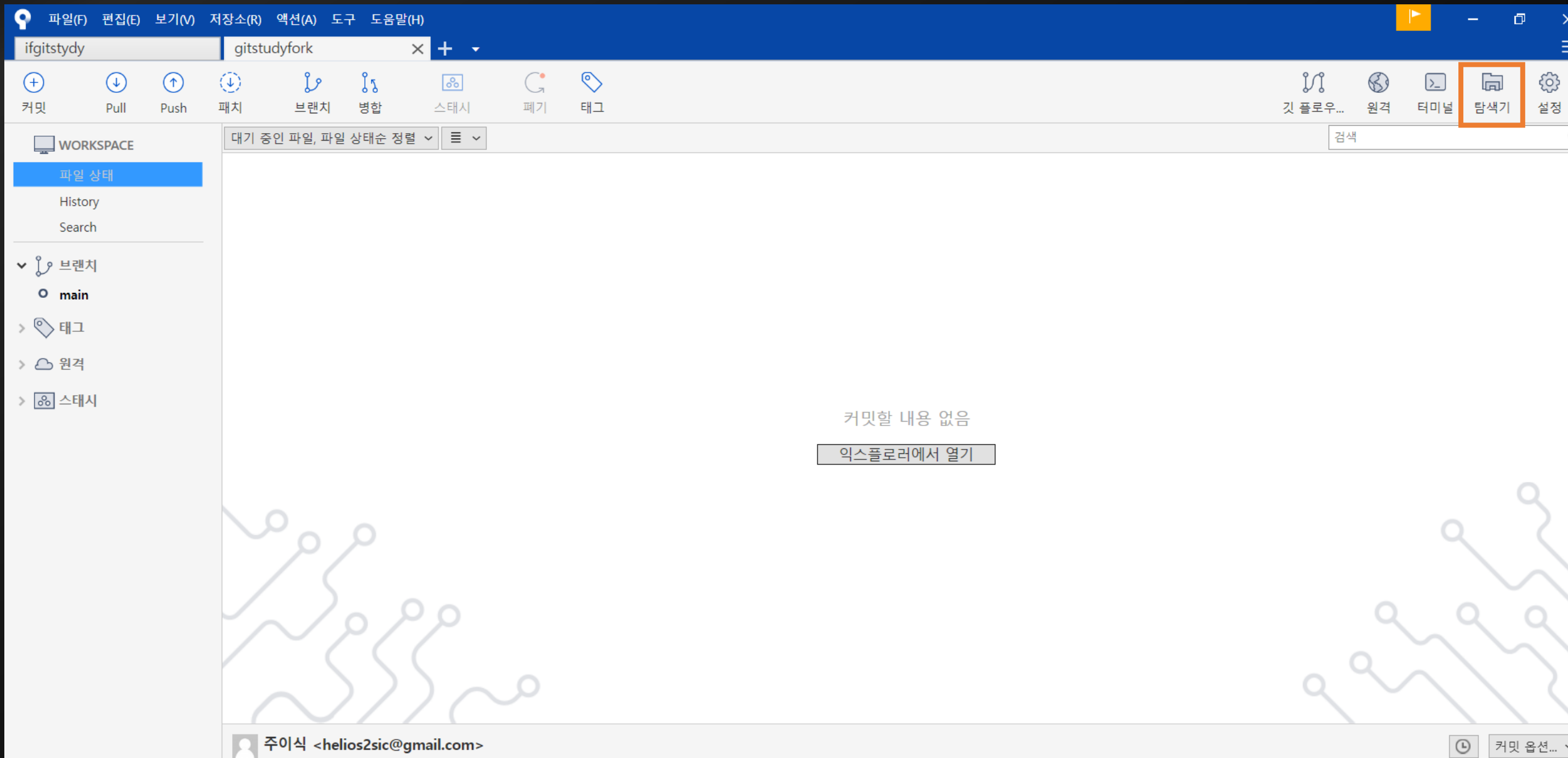
커밋: 3de77d8872194ecc428e9f68c195d4d7129fdf61 [3de77d8]
상위 항목: 84f1f71397
작성자: 주이식 <helios2sic@gmail.com>
날짜: 2022년 3월 25일 금요일 오후 6:25:48
커밋한 사람: 주이식

Git bash 설치

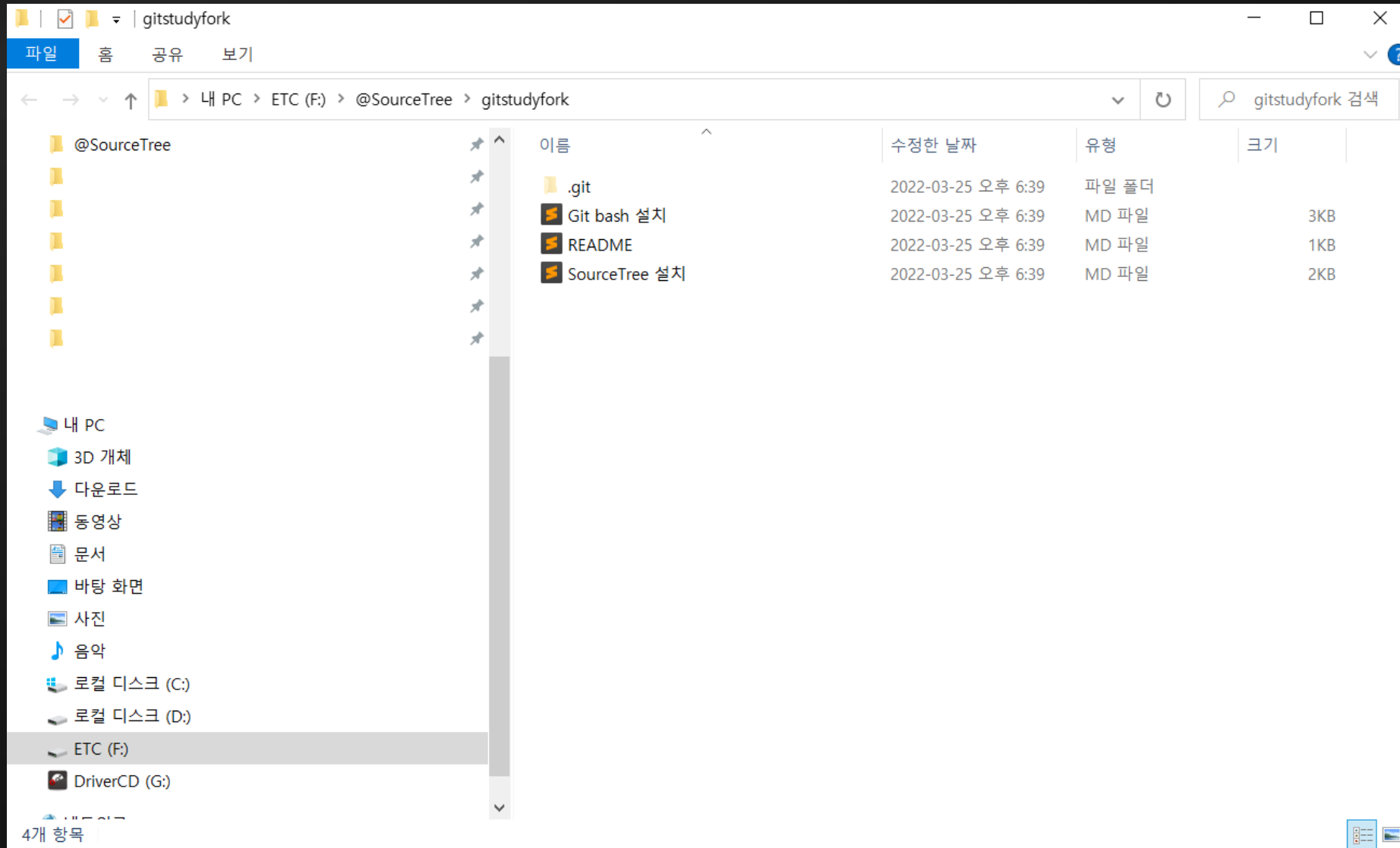
Git bash 설치.md

```
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
```

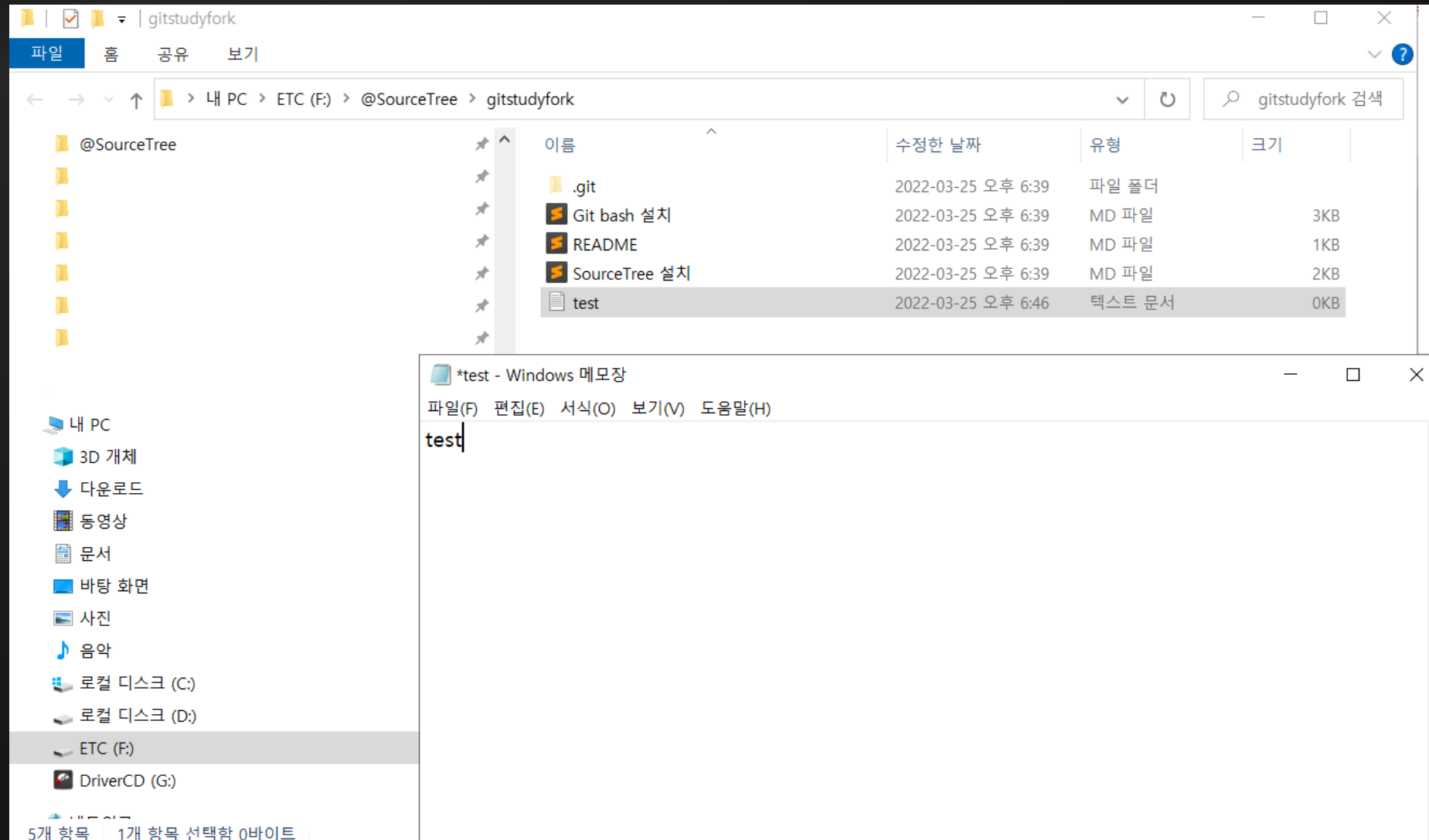
사용방법 - Commit



사용방법 - Commit

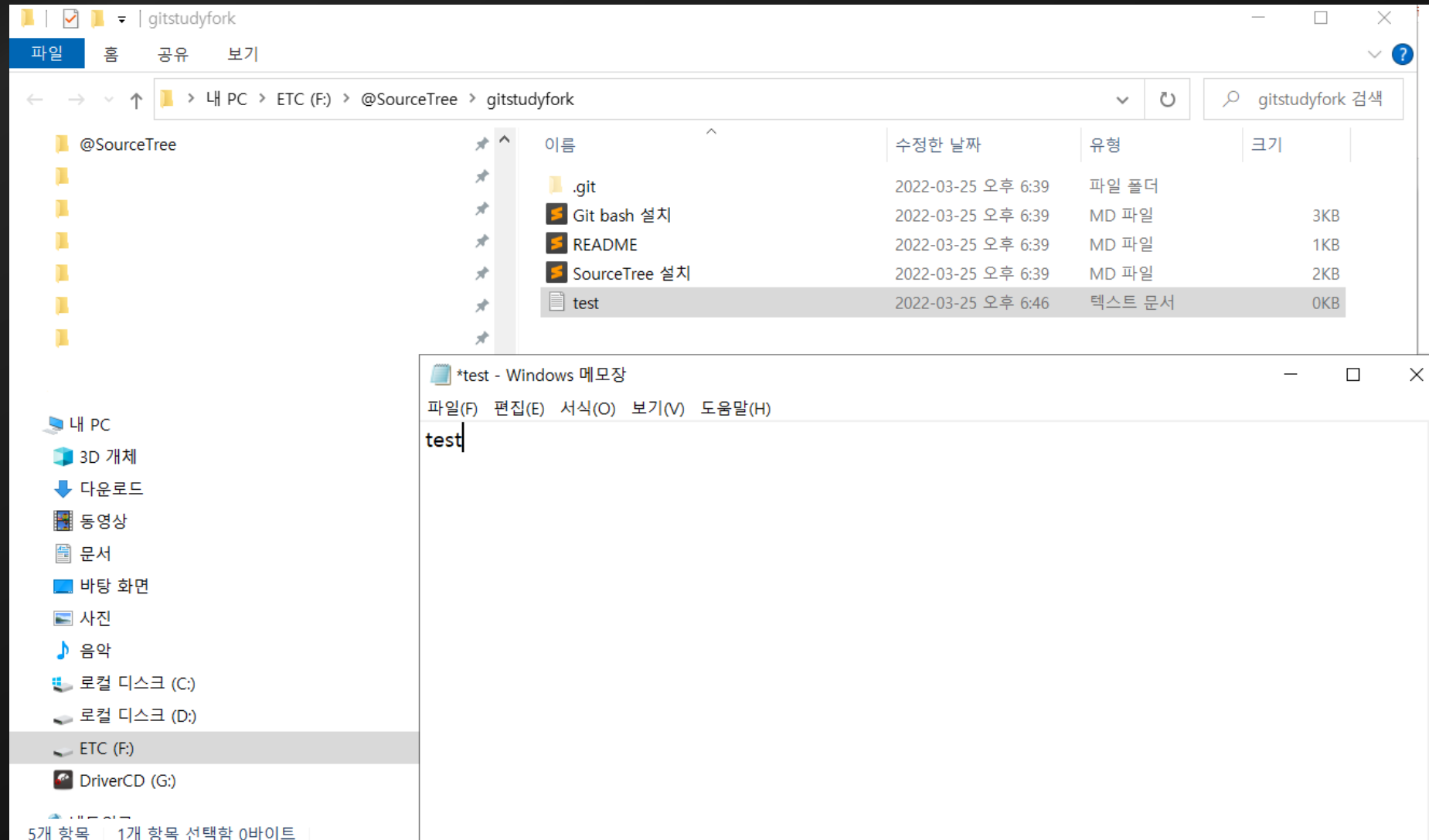


사용방법 - Commit



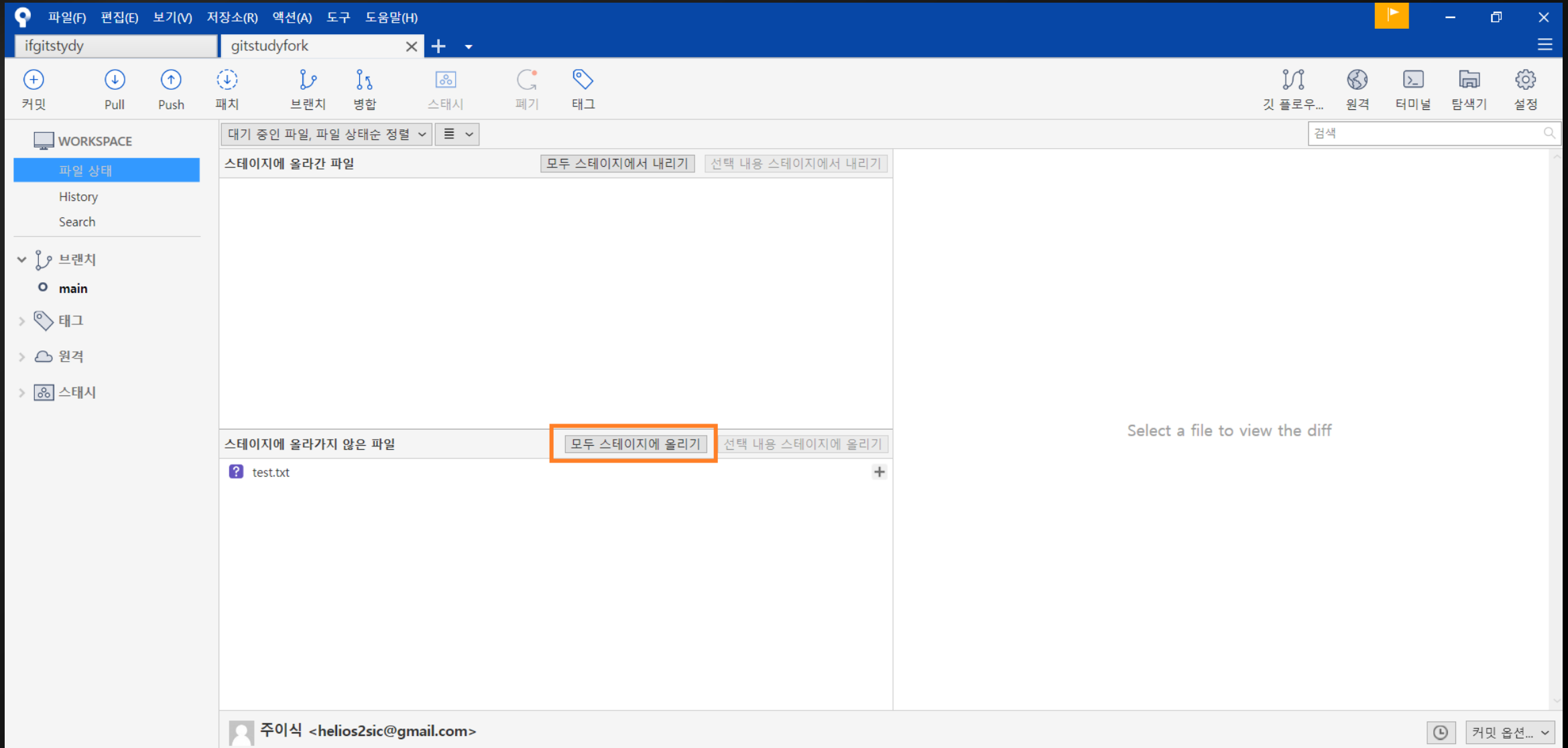
폴더 우클릭 -> 새로 만들기 -> 텍스트 문서 -> test로 이름변경 -> 내용 test로 수정

사용방법 - Commit



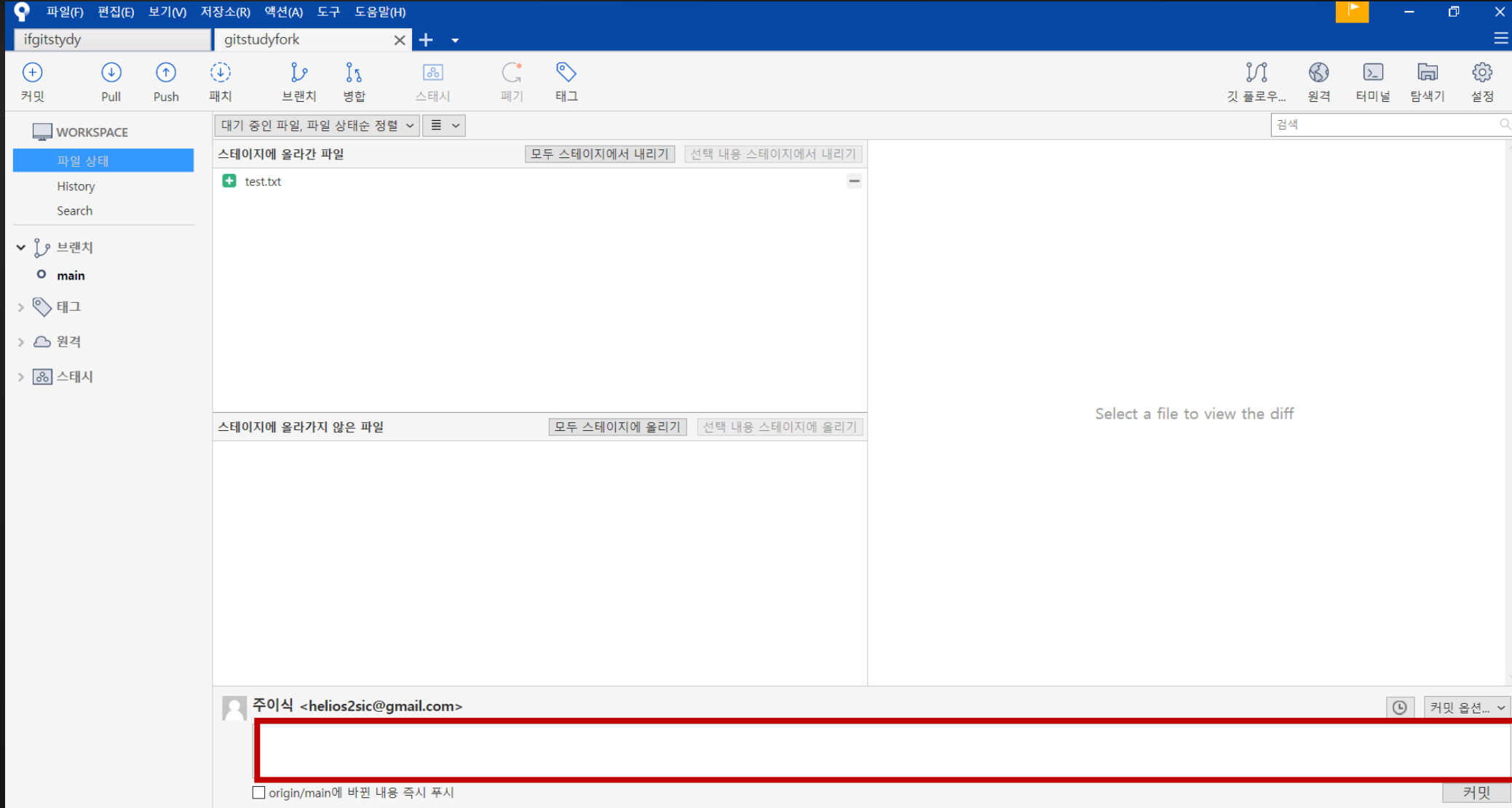
폴더 우클릭 -> 새로 만들기 -> 텍스트 문서 -> test로 이름변경 -> 내용 test로 수정

사용방법 - Commit



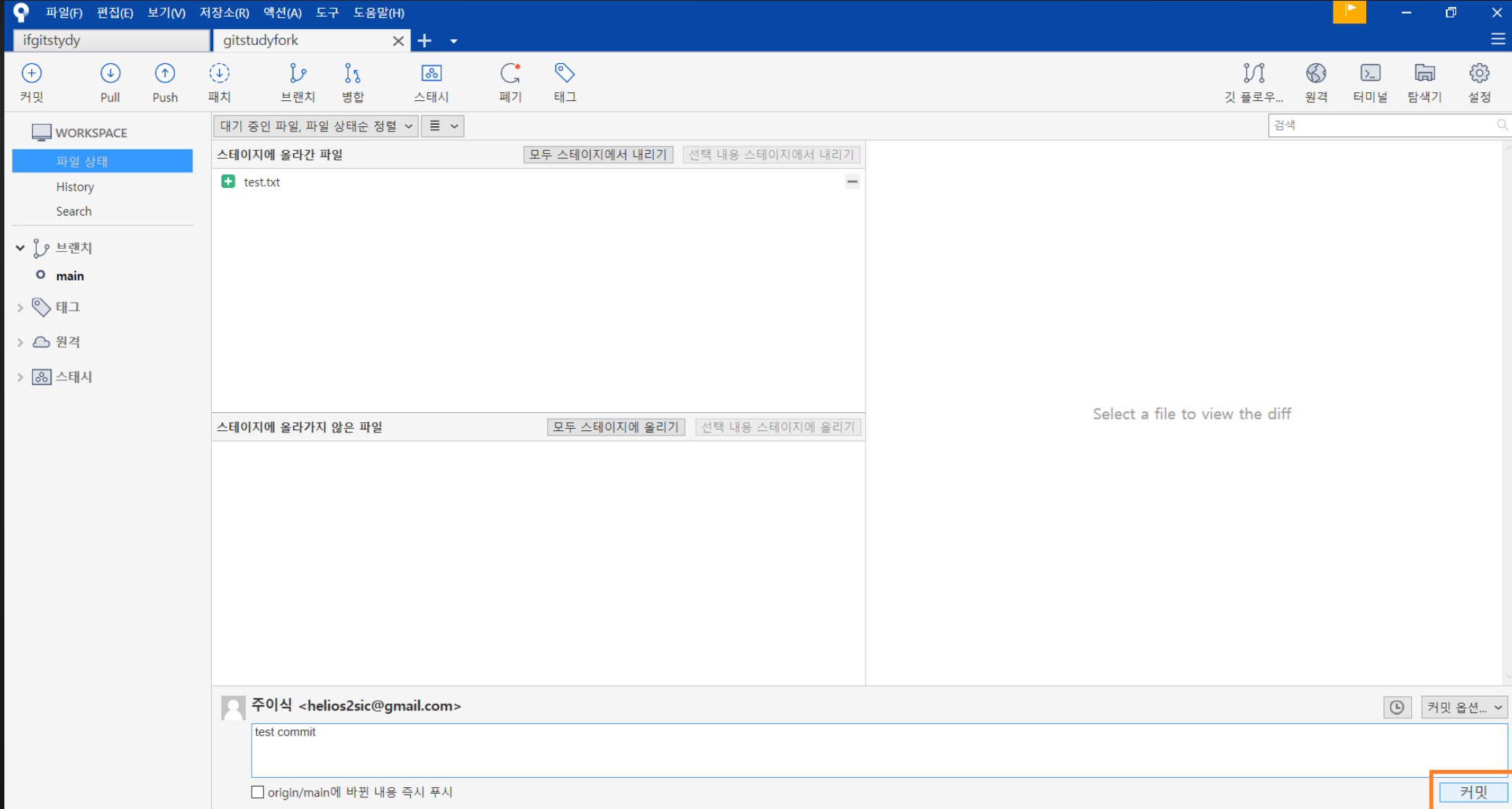
SourceTree로 가면 다음과 같이 바뀜 -> 모두 스테이지에 올리기 클릭

사용방법 - Commit



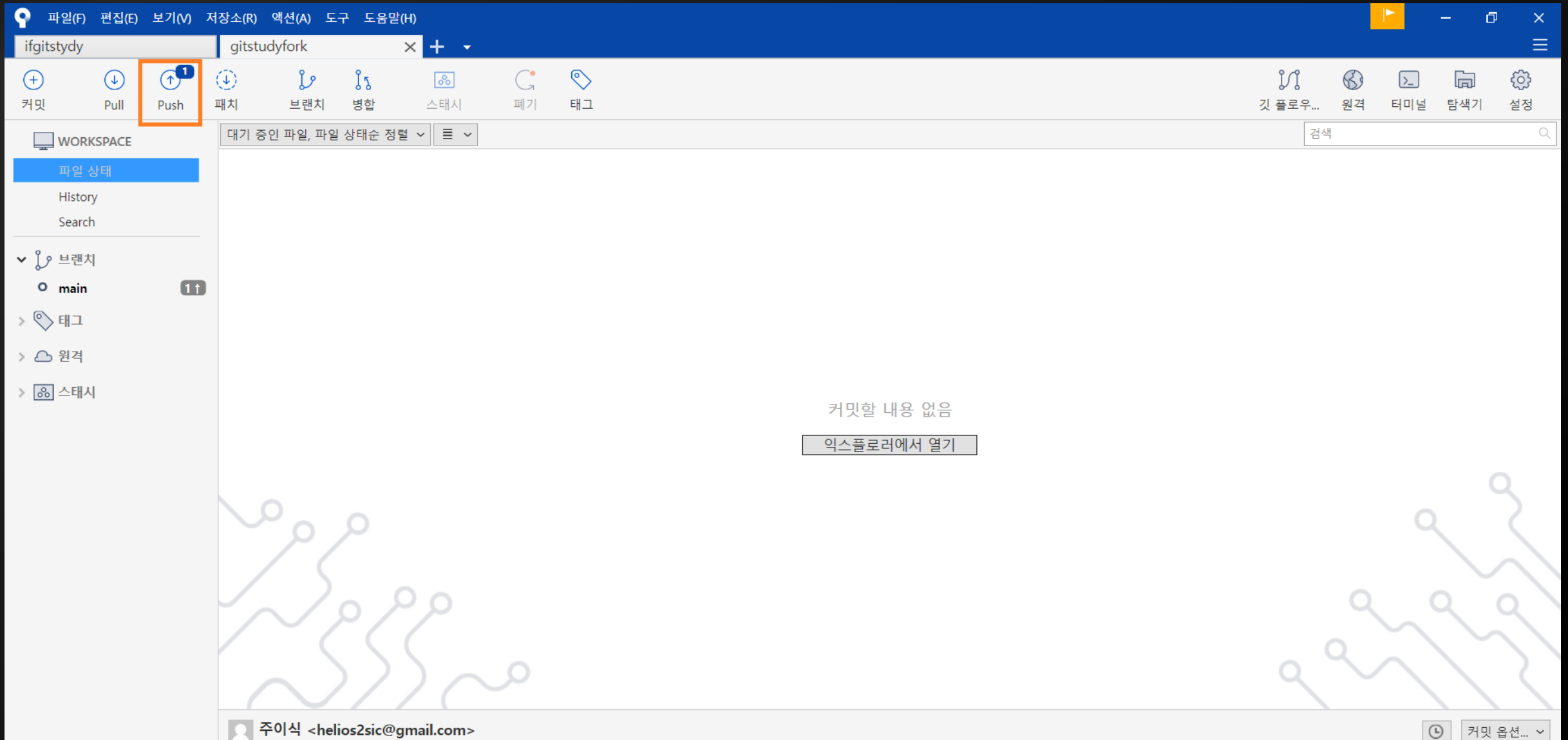
빨간곳에 커밋 메시지를 쓸 수 있음

사용방법 - Commit



다음과 같이 커밋 메시지를 적고 우측 하단에 커밋을 누르면 커밋완료

사용방법 - Push



커밋이 완료되면 다음과같이 push 우측 상단에 숫자가 생김

Push

푸시(Push)란?

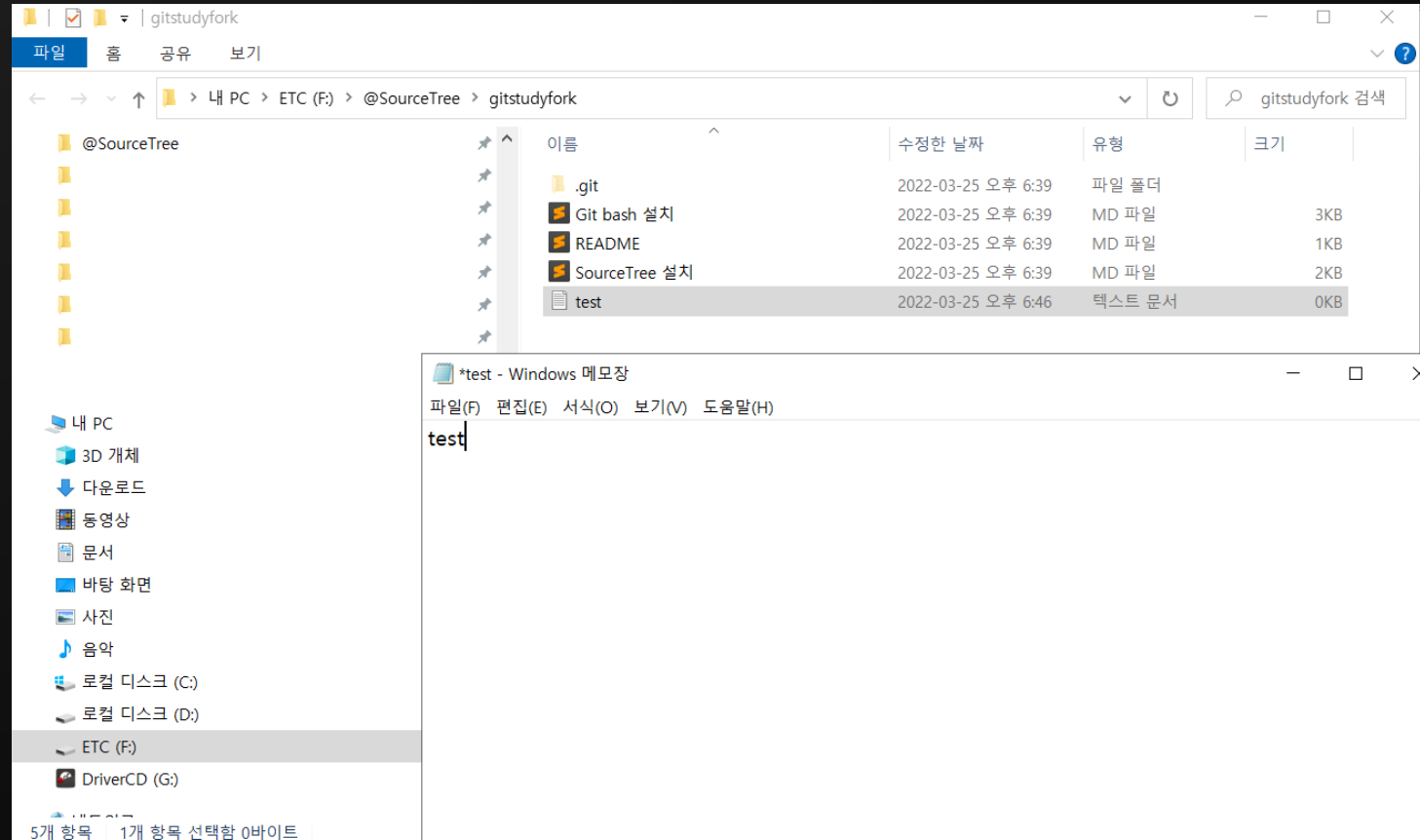
- 저장소로 파일을 업로드 하는 것

Push

개인 repository([username]/Study-Git)



로컬 repo

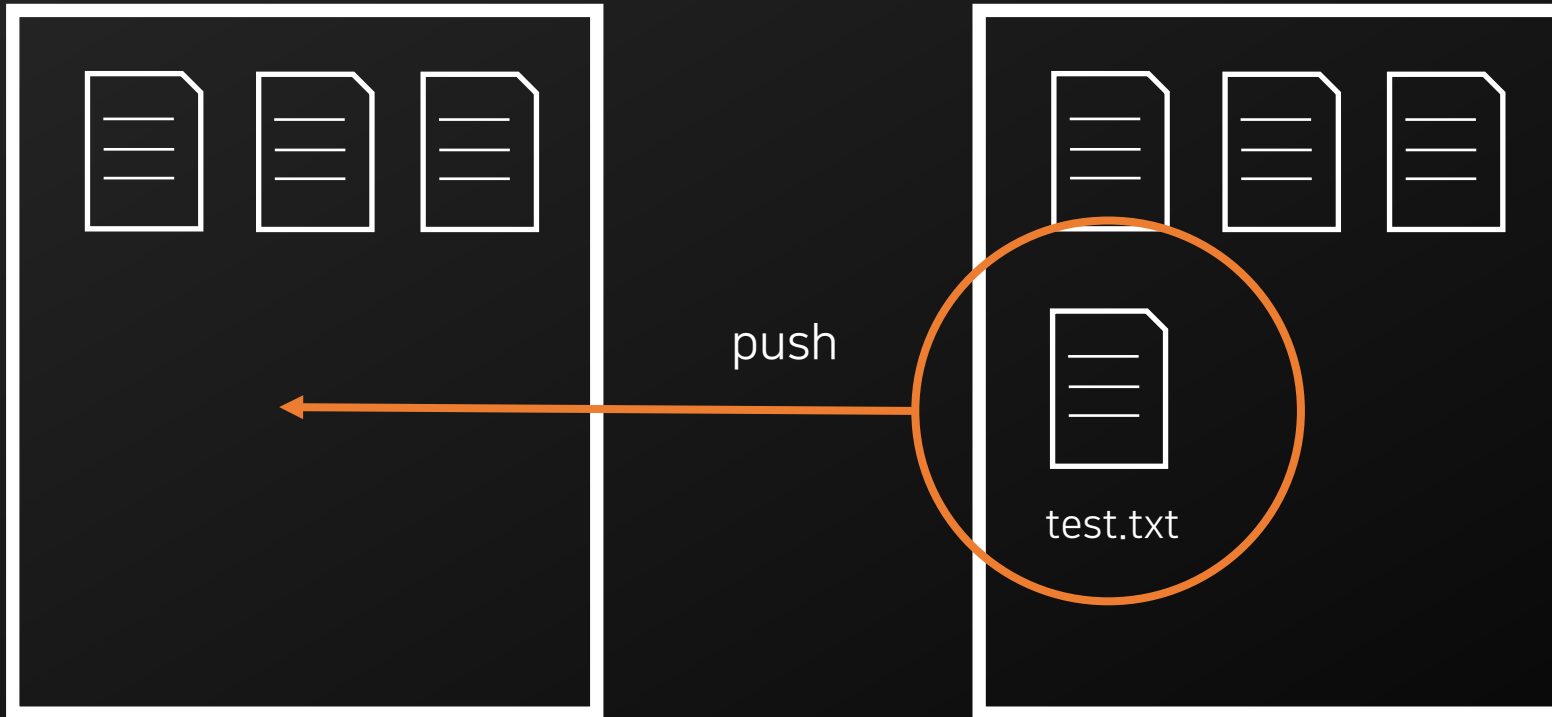


로컬 repo에서 test란 파일을 만들면 개인 repo에는 적용 안되어있음
따라서 개인 repo에 파일 업로드해서 변경사항을 업데이트 시킴

Push

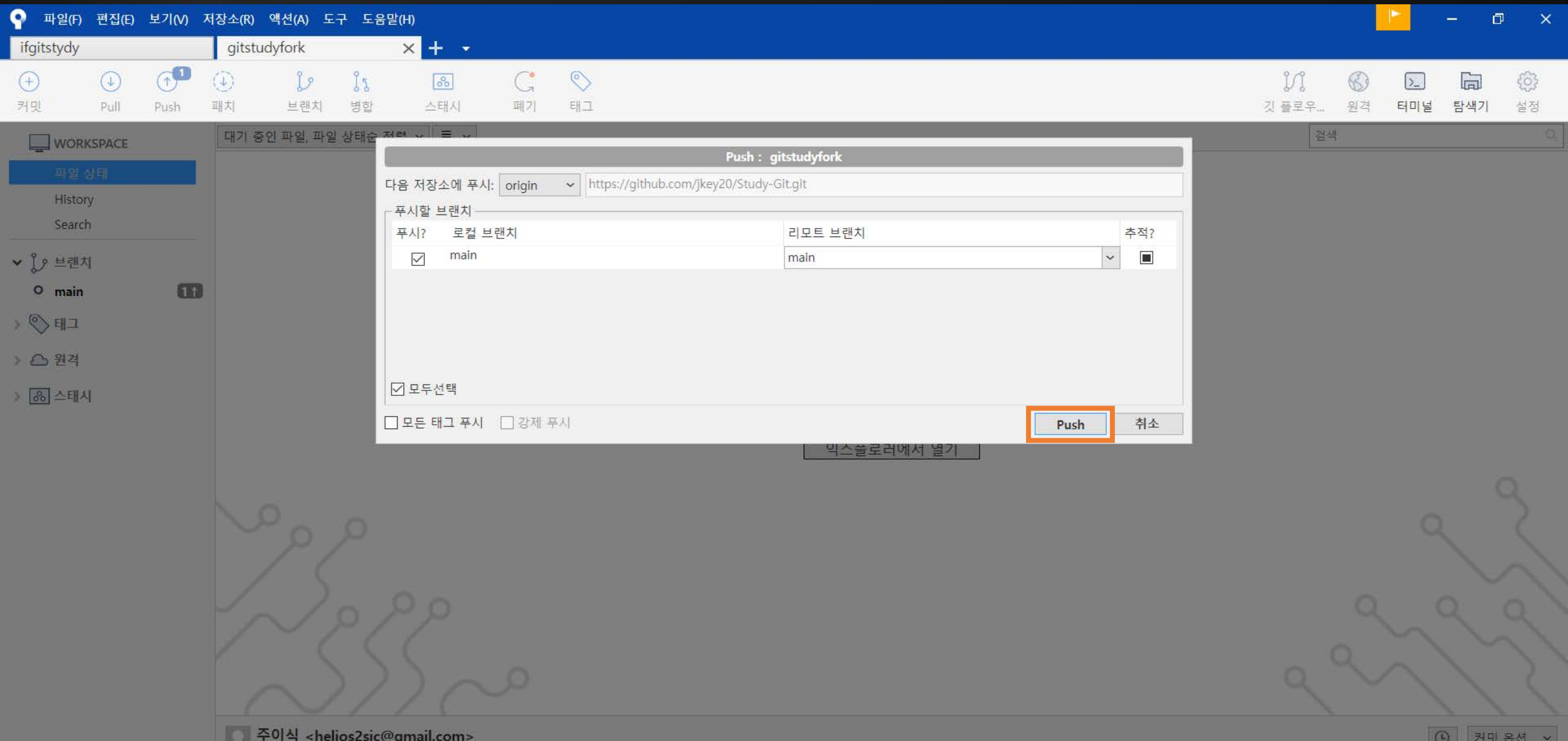
개인 repository([username]/Study-Git)

로컬 repo



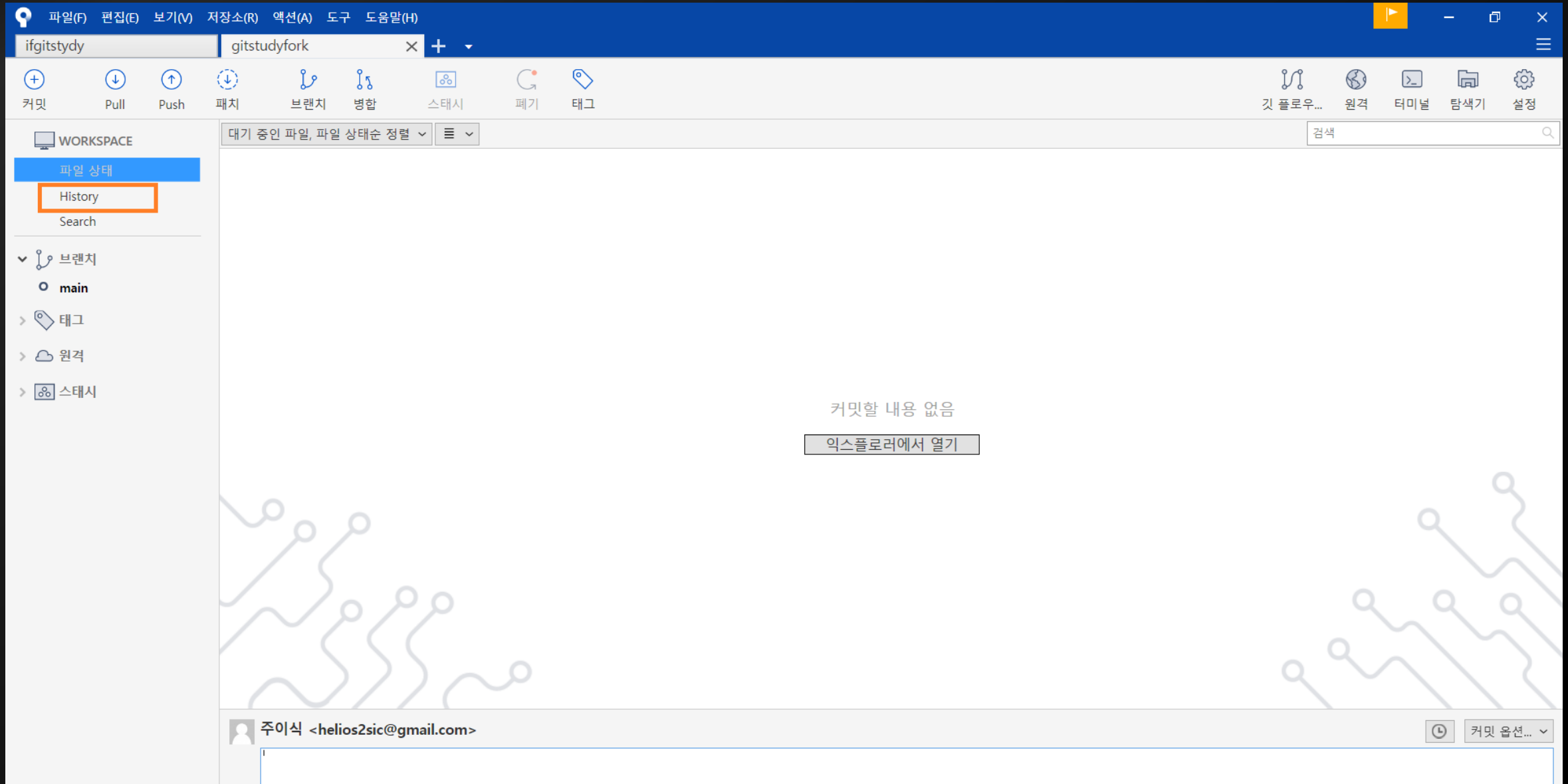
로컬 repo에서 생성된 test 파일을 푸시하면 개인 repo에도 test 파일이 생김

사용방법 - Push



push 버튼을 누르면 다음과 같은 화면이 나오고 push버튼을 한번 더 누름

사용방법 - Push



push가 완료되면 History를 누름

사용방법 - Push

ifgitsydy | gitstudyfork

커밋 Pull Push 패치 브랜치 병합 스테시 폐기 태그

WORKSPACE
파일 상태
History
Search

브랜치
main
태그
원격
스테시

모든 브랜치	원격 브랜치	날짜순으로 정렬	건너뛰기
그래프	설명	날짜	작성자
○ main	test commit	25 3 2022 18:55	주이식 <helios2sic@gmail.com> aba040c318cd16feb7d6c32aabc104cfe7295aeb [aba040c]
●	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com> 3de77d8872
●	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com> 84f1f7
●	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com> aa5fe1
●	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com> 23227

파일 상태순 정렬

커밋: aba040c318cd16feb7d6c32aabc104cfe7295aeb [aba040c]
상위 항목: 3de77d8872
작성자: 주이식 <helios2sic@gmail.com>
날짜: 2022년 3월 25일 금요일 오후 6:55:24
커밋한 사람: 주이식

test commit


+ test.txt

수정 전 파일 열기 | 수정된 후 파일 열기




No changes in this file have been detected, or it is a binary file


커밋이 잘 된것을 볼 수 있음

사용방법 - Push





[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 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

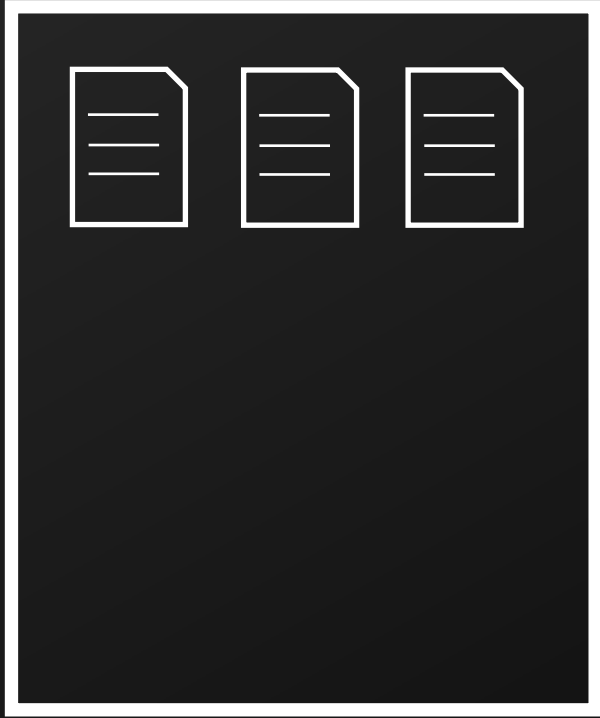
No packages published

[Publish your first package](#)

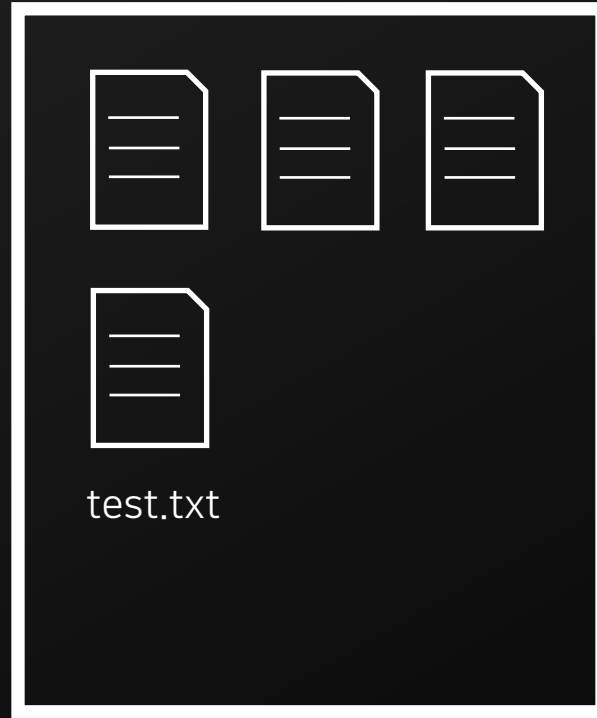
개인 repo에 가면 test.txt 파일이 있는걸 확인할 수 있음

현재 상태

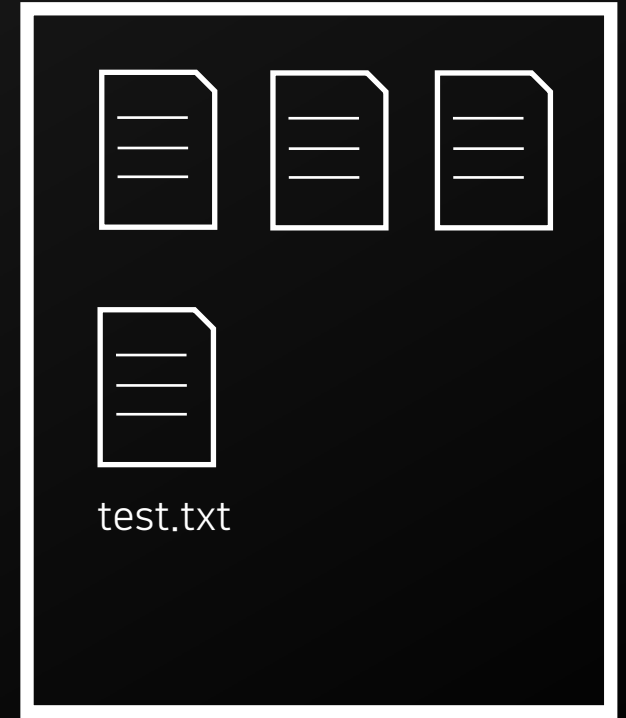
협업 repository(sejonginterface/Study-Git)



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





[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



[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](#)

[Filters](#) 


[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!** Updated in the last three days: [updated:>2022-03-22](#).

Windows 제품 키즈

사용방법 - Pull Request

GitHub interface showing the Pull Request workflow for the repository `sejonginterface / Study-Git`.

The interface displays the comparison between the `base repository: sejonginterface/Study-Git` and the `head repository: jkey20/Study-Git`. The status indicates **✓ Able to merge**, meaning these branches can be automatically merged.

A blue box highlights the **Create pull request** button, which is the next step in the process.

Below the button, the interface shows the commit history for the comparison, including the `test commit` by `jkey20` committed 20 minutes ago.

At the bottom, it shows the file changes: **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

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Open a pull request

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

 base repository: [sejonginterface/Study-Git](#) ▼


base: [main](#) ▼

←

head repository: [jkey20/Study-Git](#) ▼

compare: [main](#) ▼










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




test commit


Write

Preview

H B I         

Leave a comment

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

☒ Allow edits by maintainers 

Create pull request ▼

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone


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

사용방법 - Pull Request

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Open a pull request

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

 base repository: sejonginterface/Study-Git ▼


base: main ▼

←

head repository: jkey20/Study-Git ▼

compare: main ▼










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




test commit


Write

Preview

H B I         

Leave a comment

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

☒ Allow edits by maintainers 

Create pull request ▼

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone


Pull Request의 제목을 정하는 부분이다.

사용방법 - Pull Request

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Open a pull request

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

 base repository: sejonginterface/Study-Git ▼


base: main ▼

←

head repository: jkey20/Study-Git ▼

compare: main ▼










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




test commit


Write

Preview

H B I         

Leave a comment

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

☒ Allow edits by maintainers 

Create pull request ▼

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone


Pull Request의 내용을 정하는 부분이다.

사용방법 - Pull Request

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Open a pull request

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

 base repository: sejonginterface/Study-Git ▼


base: main ▼

←

head repository: jkey20/Study-Git ▼

compare: main ▼










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




test commit


Write

Preview

H B I         

Leave a comment

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

☒ Allow edits by maintainers 

Create pull request ▼

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects


None yet

Milestone

No milestone

모든 설정이 끝나면 Create pull request를 누른다.

사용방법 - Pull Request

 [sejonginterface](#) / [Study-Git](#) Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

pull request title #1

Edit [Code](#)

[Open](#) jkey20 wants to merge 1 commit into [sejonginterface:main](#) from [jkey20:main](#)


Conversation 0

Commits 1



Checks 0

Files changed 1


+0 -0


 jkey20 commented now Member

pull request content

  test commit aba040c

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

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

사용방법 - Pull Request

The screenshot displays the GitHub interface for a pull request titled "pull request title #1". The repository is "sejonginterface / Study-Git" and is public. The pull request is open, showing that "jkey20" wants to merge 1 commit into "sejonginterface:main".

On the left sidebar, the "Pull requests" tab is selected, showing 1 pull request. Below it, the "Conversation" tab is active, showing a comment from "jkey20" stating "pull request content". A commit "test commit" is listed below the comment.

The main content area shows the pull request details. A comment from "jkey20" is visible, stating "pull request content". Below the comment, a commit "test commit" is listed. A message indicates: "Add more commits by pushing to the main branch on jkey20/St".

At the bottom, a green box highlights the "Merge pull request" button. A tooltip is visible, stating: "Continuous integration has not been set up. GitHub Actions and several other apps can be used to auto...". Below this, a green checkmark indicates: "This branch has no conflicts with the base brai. Merging can be performed automatically." The "Merge pull request" button is highlighted with a yellow border.

On the right, the "Merge pull request #1 from jkey20/main" dialog is open. It shows the pull request title, the email address "helios2sic@gmail.com", and a "Confirm merge" button (highlighted with a yellow border) and a "Cancel" button. Below the dialog, a comment box is visible with the text "Leave a comment".

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

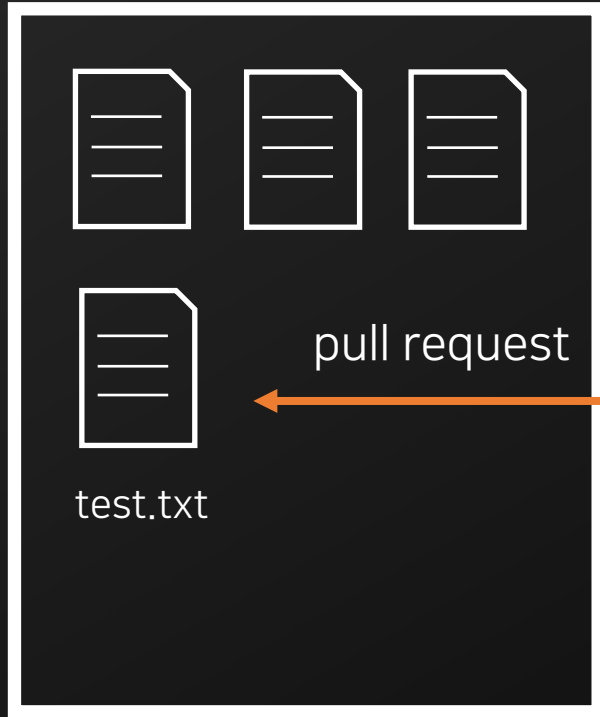
Merge

Merge(머지)란?

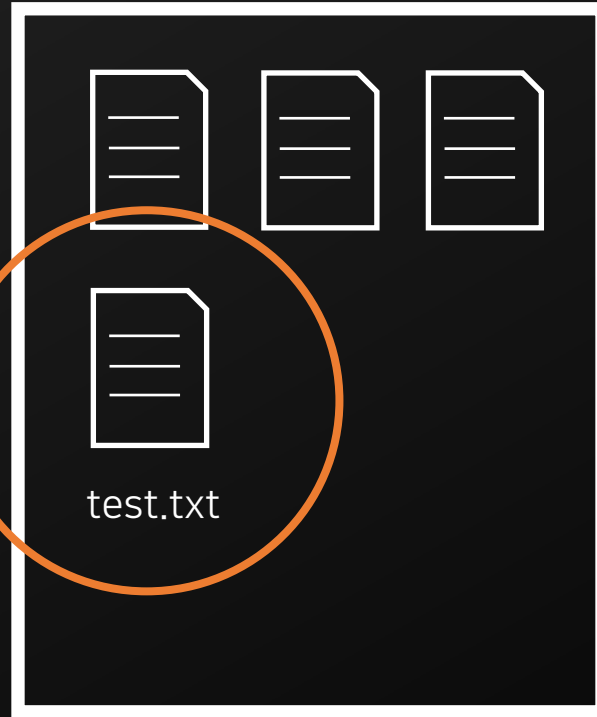
- 분기된 커밋을 하나로 합치는 작업 -> 추후에 더 자세히 배움

현재 상태

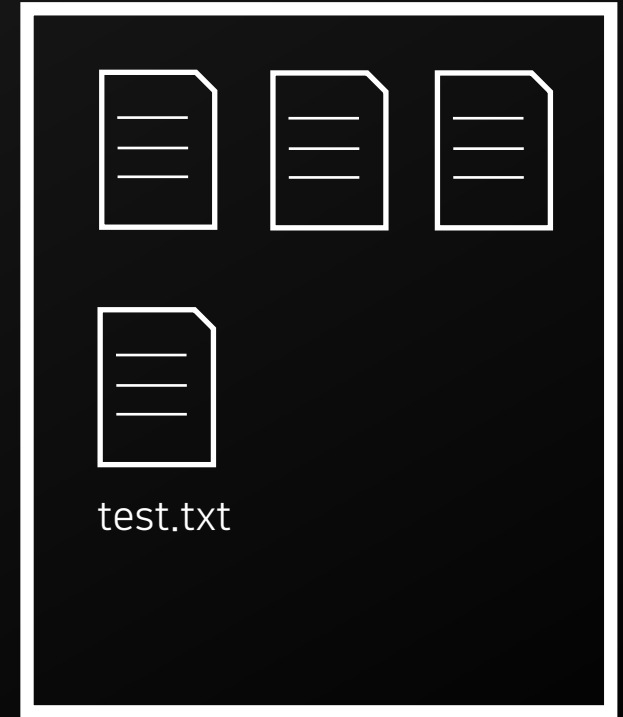
협업 repository(sejonginterface/Study-Git)



개인 repository([username]/Study-Git)



로컬 repo




pull request







정상적으로 pull request를 merge시키면 다음과 같이 반영된다.







현재 상태





[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


About

깃허브 스터디

 [Readme](#)

 [2 stars](#)

 [0 watching](#)

 [1 fork](#)

Releases

No releases published

[Create a new release](#)

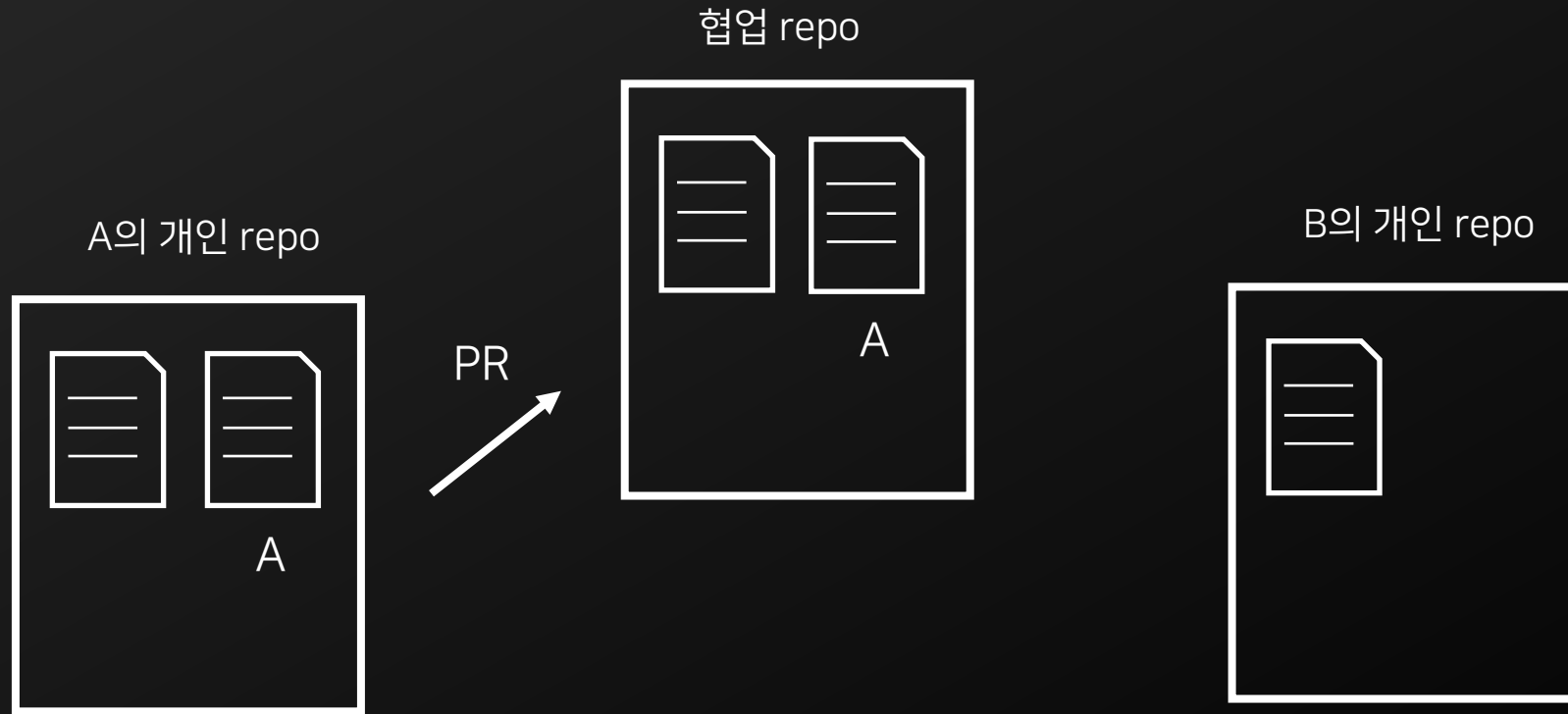
Packages

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

This screenshot shows the GitHub repository page for 'sejonginterface / Study-Git'. The repository is public and has a 'main' branch with 1 branch and 0 tags. The file list shows the following files and their commit messages:

File	Commit Message
Git bash 설치.md	Git bash 설치
README.md	Update README.md
SourceTree 설치.md	SourceTree 설치
test.txt	test commit
test2.txt	test2

The README.md file is visible at the bottom of the page.

This screenshot shows the GitHub repository page for 'jkey20 / Study-Git', which is a fork of 'sejonginterface / Study-Git'. The repository is public and has a 'main' branch with 1 branch and 0 tags. A message indicates that this branch is 2 commits behind 'sejonginterface / Study-Git:main'. The file list shows the following files and their commit messages:

File	Commit Message	Commit Hash	Time Ago
Git bash 설치.md	Git bash 설치	aba048c	2 hours ago
README.md	Update README.md		8 hours ago
SourceTree 설치.md	SourceTree 설치		6 hours ago
test.txt	test commit		2 hours ago

The README.md file is visible at the bottom of the page.

test2라는 파일이 협업 repo에 반영되어있고 개인 repo에는 존재하지 않는다.

Pull

The screenshot shows the SourceTree application interface. The top menu bar includes options like '파일(F)', '편집(E)', '보기(V)', '저장소(R)', '액션(A)', '도구', and '도움말(H)'. Below the menu is a toolbar with icons for '커밋', 'Pull', 'Push', '패치', '브랜치', '병합', '스테시', '폐기', and '태그'. On the right side of the toolbar, there are icons for '깃 플로우...', '원격', '터미널', '탐색기', and '설정' (Settings), which is highlighted with an orange box. The left sidebar shows the 'WORKSPACE' section with '파일 상태', 'History', and 'Search' options. Below this, the '브랜치' (Branches) section shows the 'main' branch selected. The main area displays a commit history table with columns for '그래프', '설명', '날짜', '작성자', and '커밋'. The table lists several commits, with the most recent one being 'test commit' by '주이식 <helios2sic@gmail.com>' with commit hash 'aba040c'. Below the table, there is a section for '파일 상태' (File Status) showing the current commit details. At the bottom, there is a file view for 'test.txt' showing the message 'No changes in this file have been detected, or it is a binary file'.

그래프	설명	날짜	작성자	커밋
○	test commit	25 3 2022 18:55	주이식 <helios2sic@gmail.com>	aba040c
●	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
●	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
●	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5fe1e
●	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23227a5

파일 상태: aba040c318cd16feb7d6c32aabc104cfe7295aeb [aba040c]
상위 항목: 3de77d8872
작성자: 주이식 <helios2sic@gmail.com>
날짜: 2022년 3월 25일 금요일 오후 6:55:24
커밋한 사람: 주이식

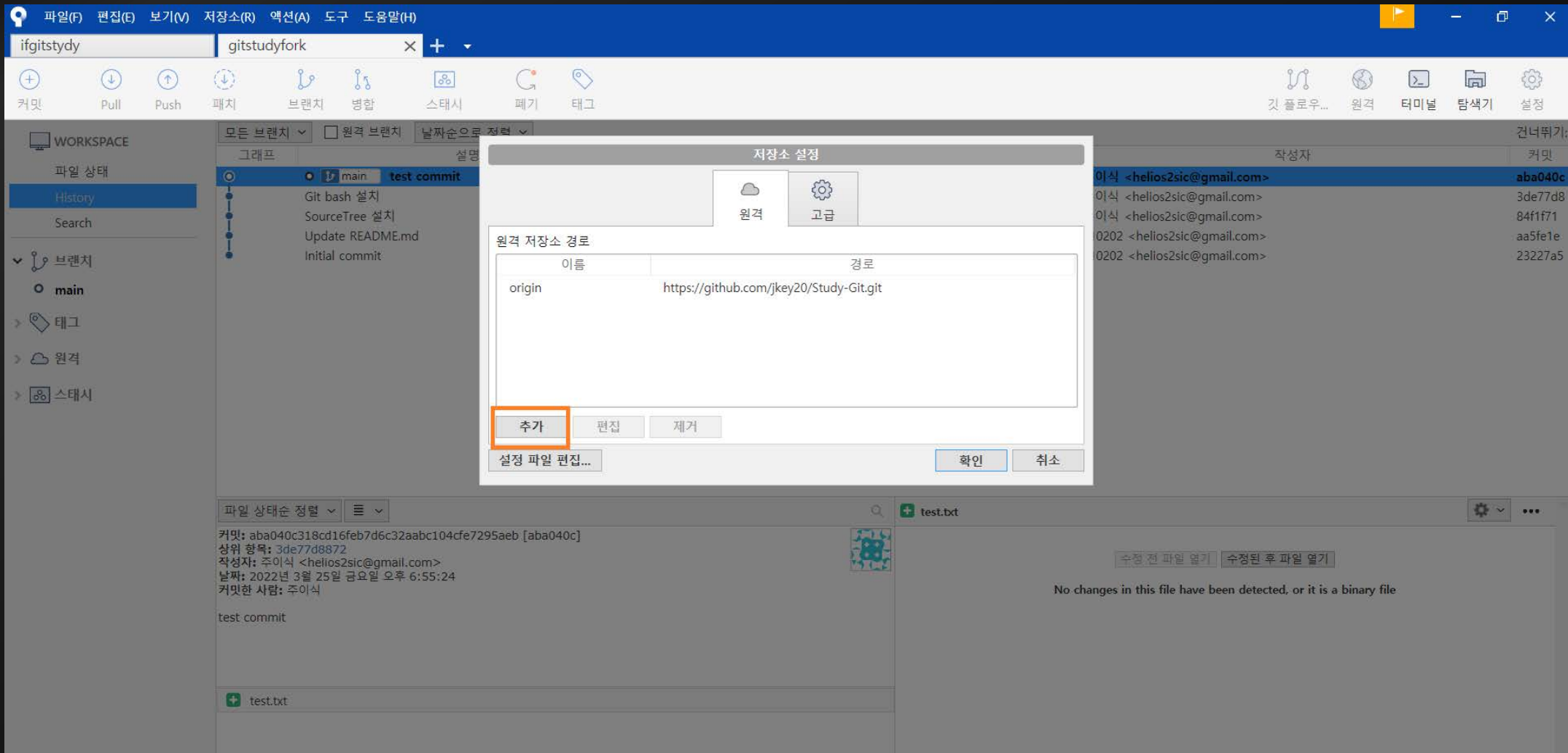
test commit

test.txt

No changes in this file have been detected, or it is a binary file

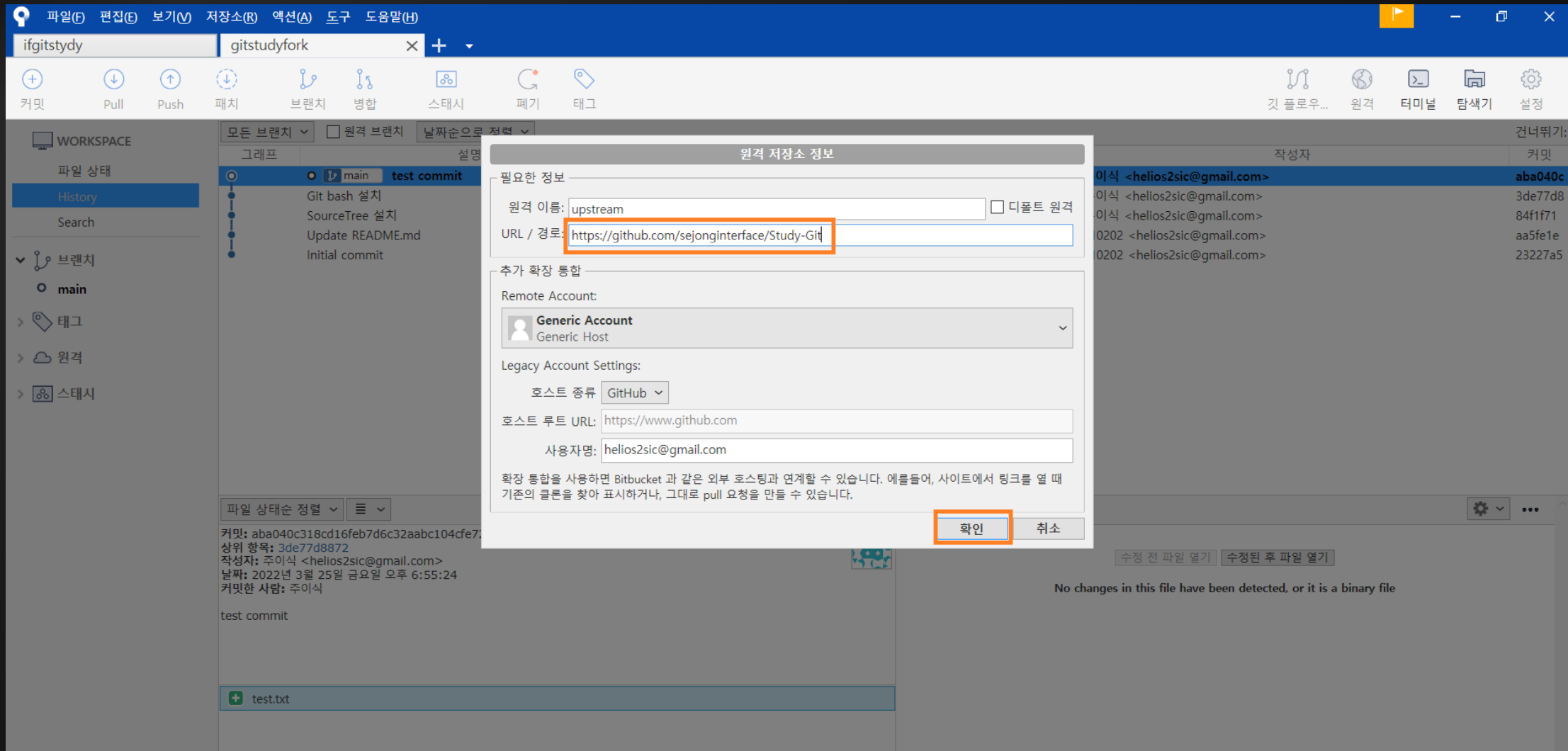
협업 저장소에서 파일을 가져오기 위해서 설정을 눌러 협업 저장소로부터 가져오게 설정한다.

Pull



추가 버튼을 누른다.

Pull



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

Pull

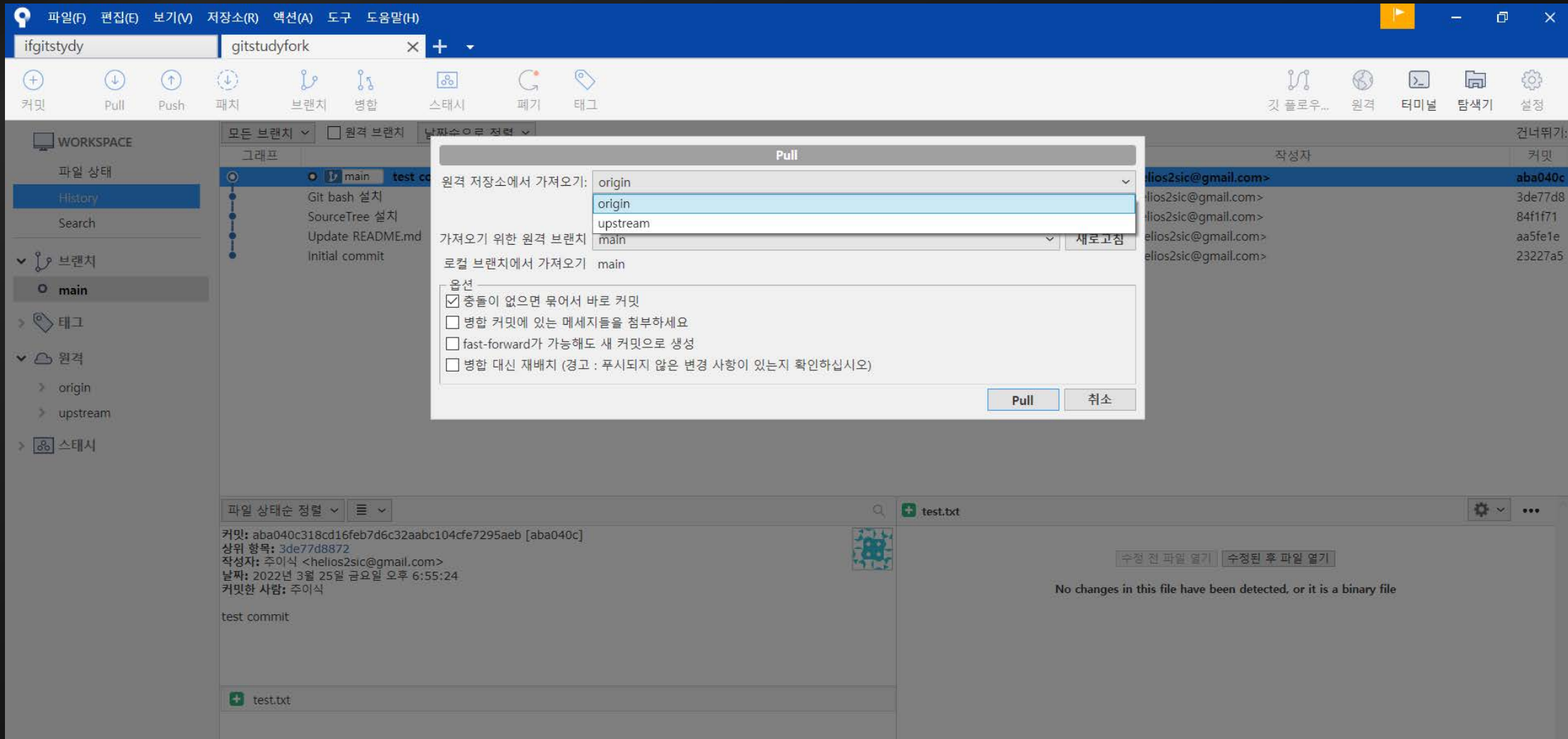
The screenshot shows the SourceTree application window. The top toolbar contains several icons, with the 'Pull' icon (a downward arrow) highlighted with an orange box. The main area displays a commit history table for the 'main' branch.

그래프	설명	날짜	작성자	커밋
○	test commit	25 3 2022 18:55	주이식 <helios2sic@gmail.com>	aba040c
●	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
●	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
●	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5fe1e
●	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23227a5

Below the commit history, the 'test.txt' file is selected. The commit details for 'aba040c' are shown, including the commit message 'test commit' and the author '주이식 <helios2sic@gmail.com>'. The file view shows 'test.txt' with the message 'No changes in this file have been detected, or it is a binary file'.

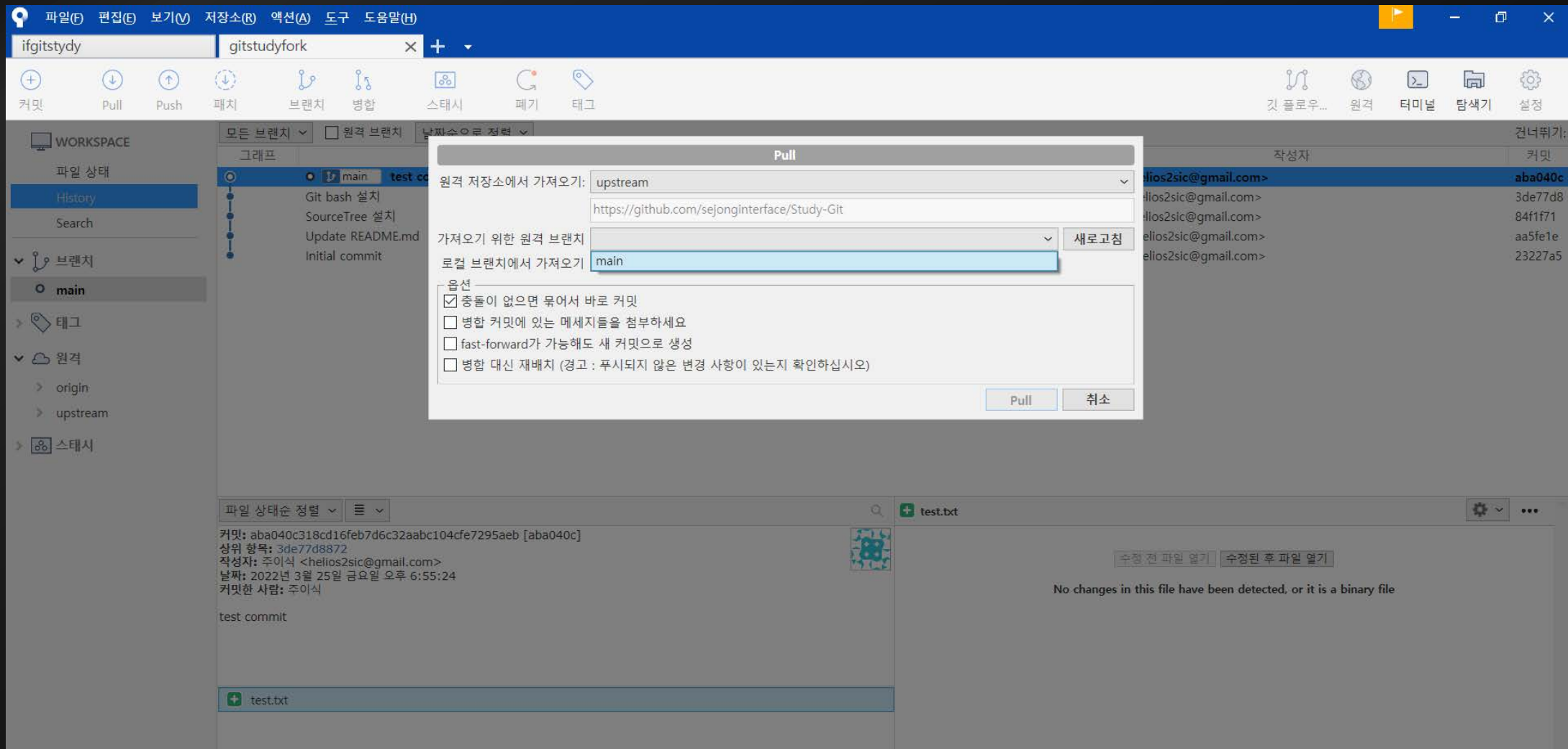
Pull을 누른다.

Pull



설정한 upstream이 새로 나온다.

Pull



가져오기 위한 원격 브랜치를 main으로 설정하고 pull을 누른다.

Pull

The screenshot shows the Git GUI interface with a pull request being merged. The commit 'test commit' is highlighted in blue, indicating it is the current commit being merged. The commit history shows a sequence of commits from 'Initial commit' to 'Merge pull request #1 from jkey20/main'. The file 'test.txt' is shown in the diff view, indicating no changes.

그래프	설명	날짜	작성자	커밋
○ main 2	test2	25 3 2022 20:28	주이식 <helios2sic@gmail.com>	4571dbe
○	Merge pull request #1 from jkey20/main	25 3 2022 20:17	210202 <helios2sic@gmail.com>	2cf686b
○	test commit	25 3 2022 18:55	주이식 <helios2sic@gmail.com>	aba040c
○	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
○	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
○	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5fe1e
○	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23227a5

커밋: aba040c318cd16feb7d6c32aabc104cfe7295aeb [aba040c]
상위 항목: 3de77d8872
작성자: 주이식 <helios2sic@gmail.com>
날짜: 2022년 3월 25일 금요일 오후 6:55:24
커밋한 사람: 주이식

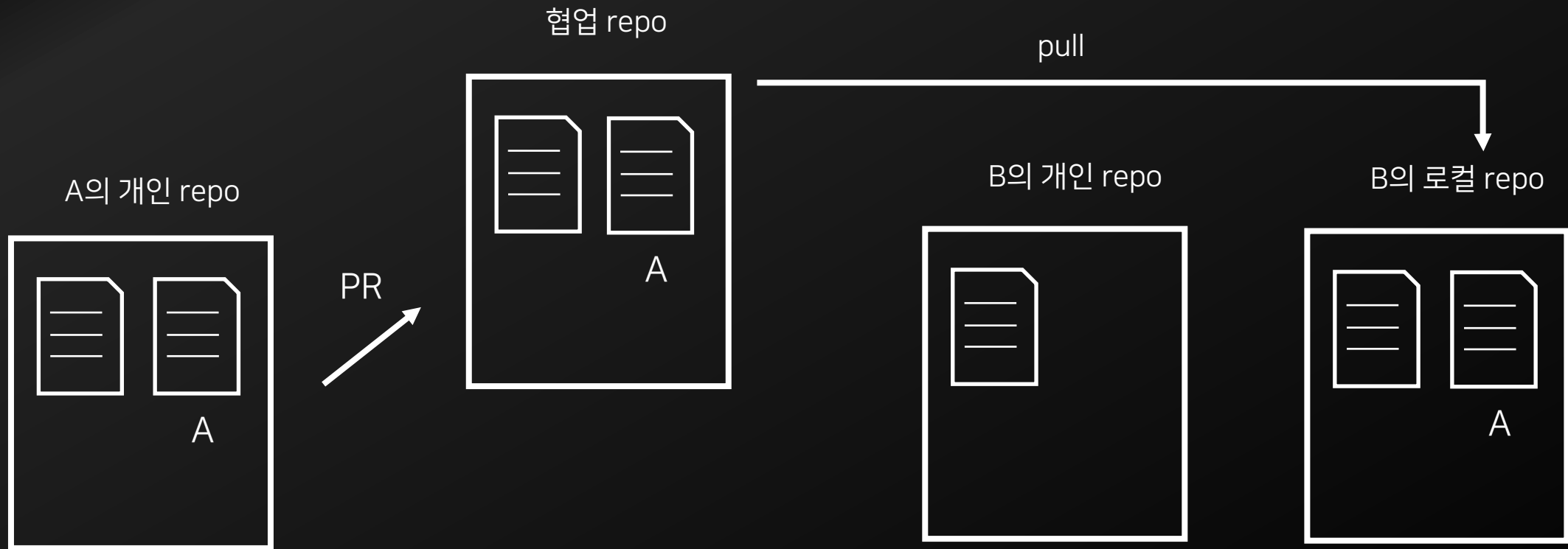
test commit

+ test.txt

No changes in this file have been detected, or it is a binary file

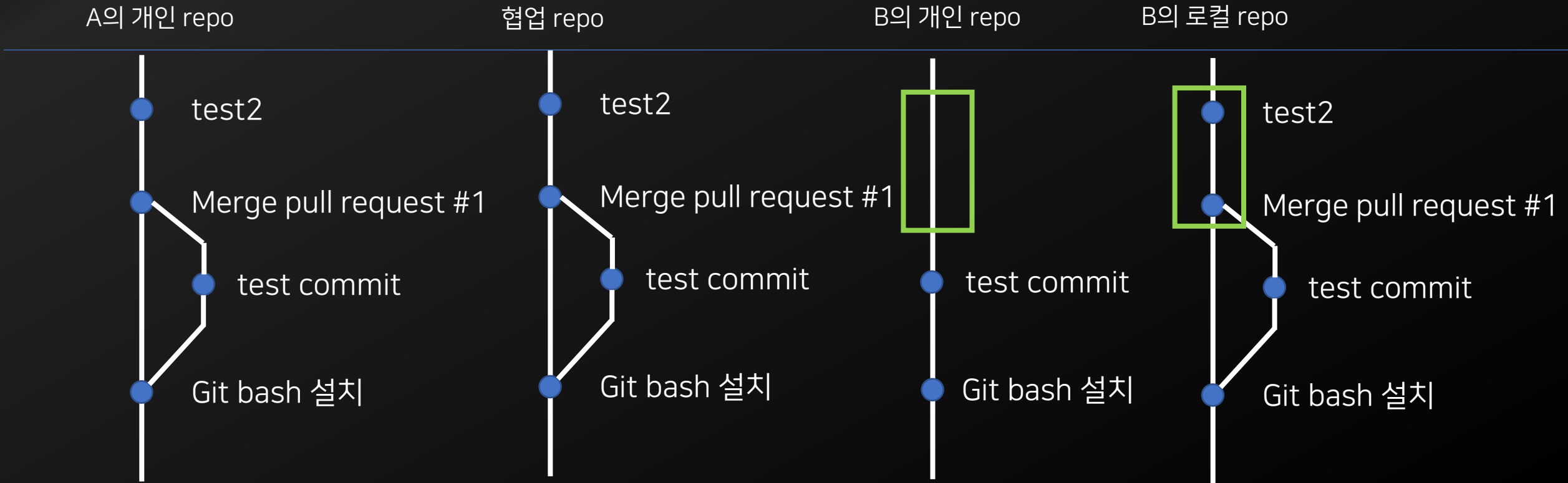
pull이 끝나면 다음과 같이 다른 사람의 커밋이 새로 표시되고 push에 숫자가 나온다.

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 차이내기때문. push를 완료하면 모든 repo에 똑같이 반영됨.

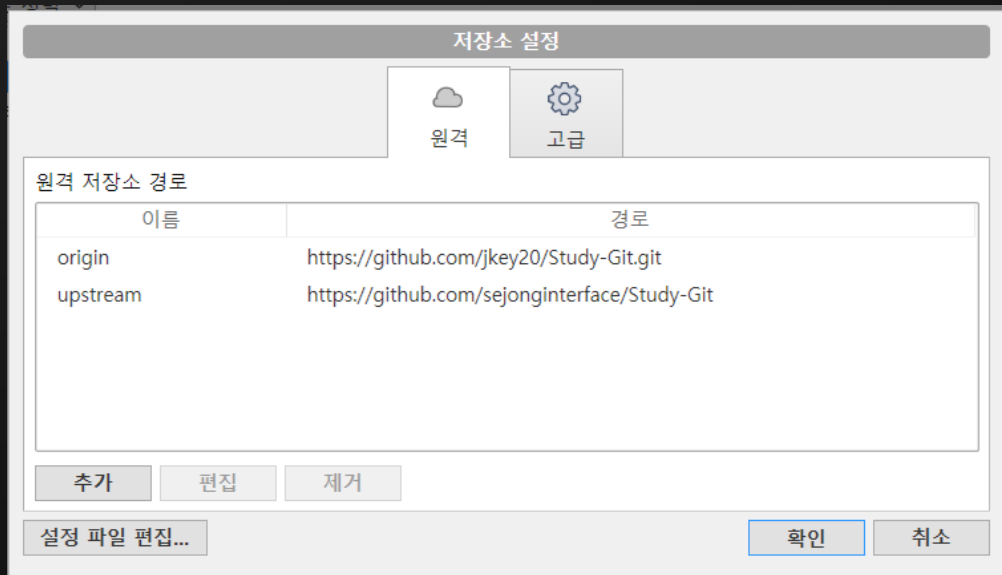
원격 저장소

- SourceTree상에서 설정한 origin의 경로

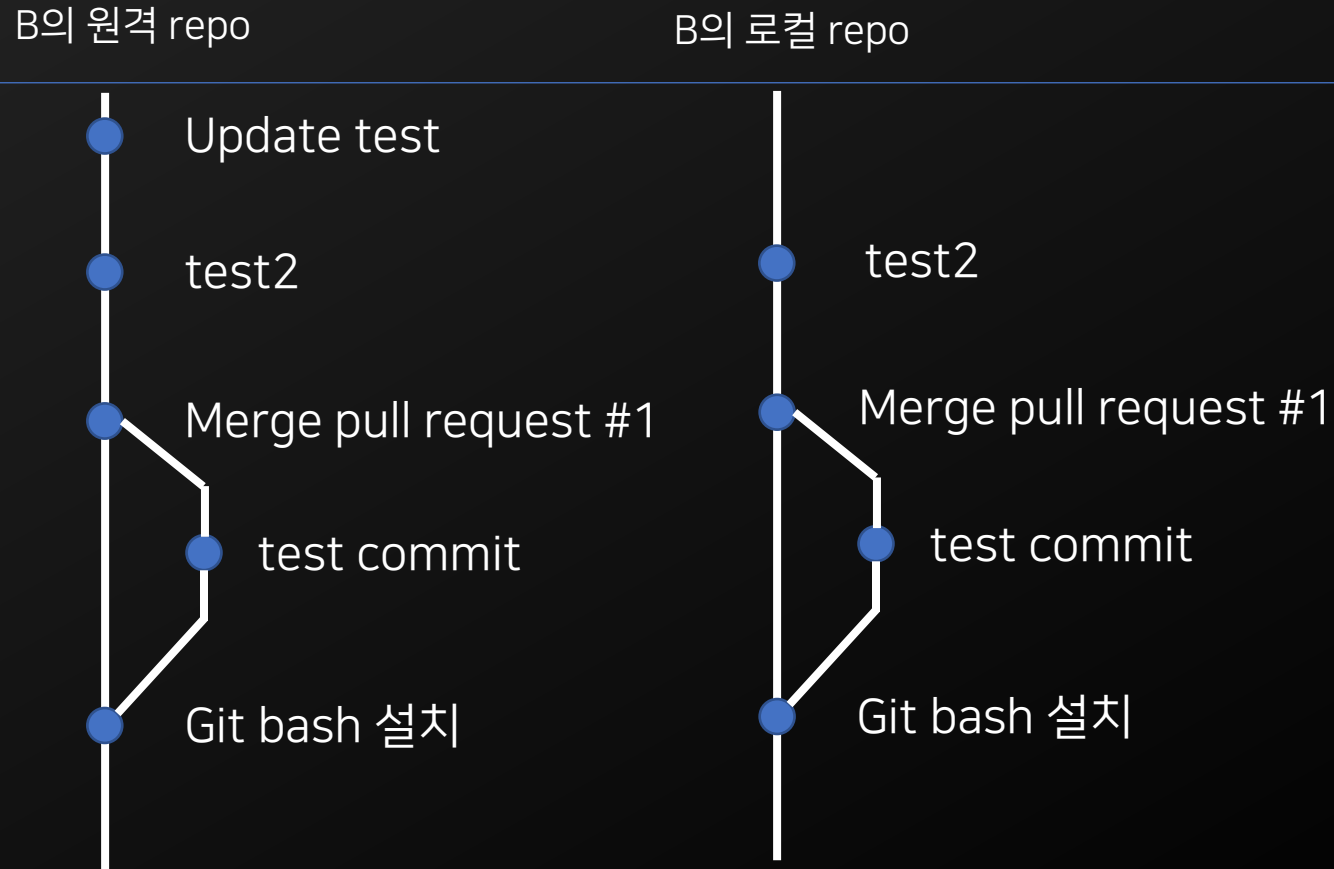
현재 원격 저장소 : 개인 repo

원격 저장소는 개인 repo도 가능하고 협업 repo도 가능함.

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



협업할 때 - 원격 저장소의 변경사항이 생겼을 때

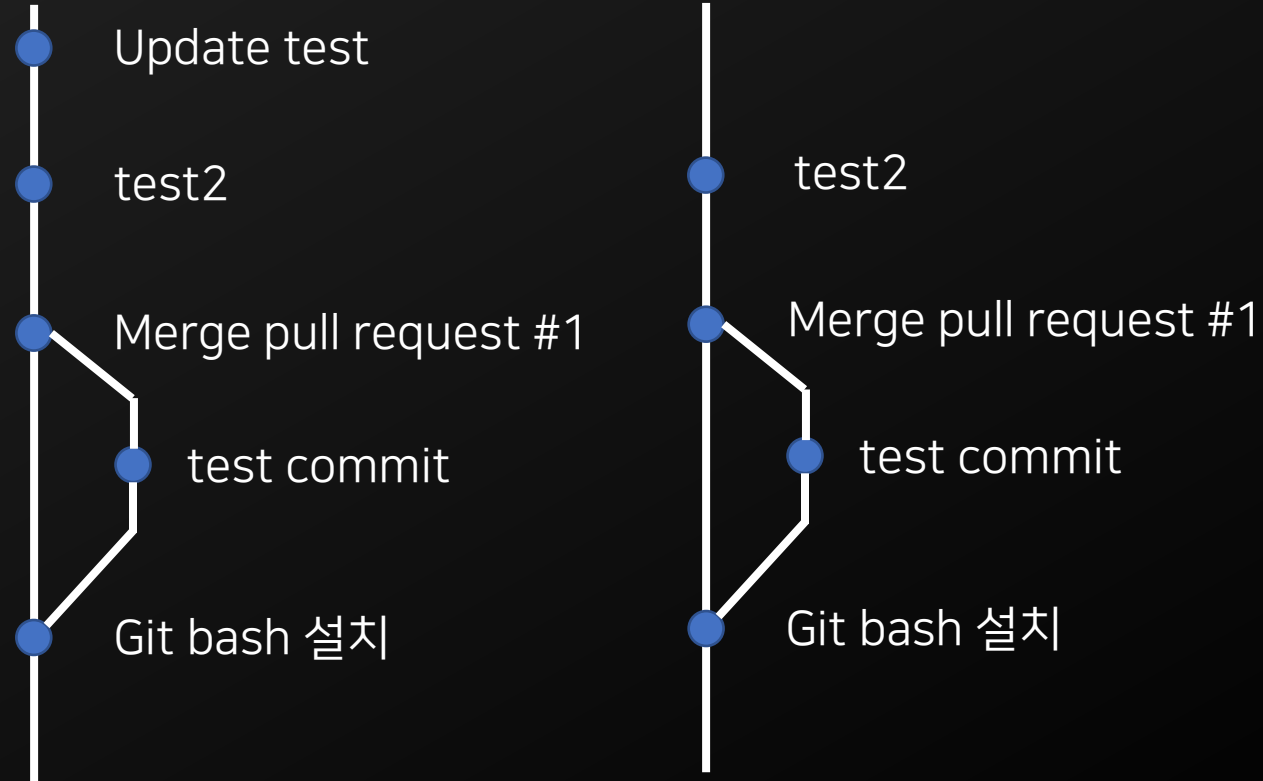


원격 repo에 커밋이 생겼을때 로컬repo에 커밋을 반영하지 않고 작업하면? -> 커밋을 무시한채로 작업해서 오류생김

협업할 때 - 원격 저장소의 변경사항이 생겼을 때

B의 개인 repo

B의 로컬 repo



그럼 pull을해서 받고 작업해야하는가?

-> pull은 아예 덮어쓰우는 작업. 로컬 repo에서 작업한것과 원격 저장소에서 작업한 파일이 덮어쓰워지는 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

The screenshot displays the Fetch IDE interface. The top menu bar includes options like 파일(F), 편집(E), 보기(V), 저장소(R), 액션(A), 도구, and 도움말(H). Below the menu is a toolbar with icons for 커밋, Pull, Push, 패치, 브랜치, 병합, 스테시, 폐기, and 태그. The left sidebar shows the WORKSPACE with sections for 파일 상태, History, Search, 브랜치 (main), 태그, 원격 (origin, upstream), and 스테시.

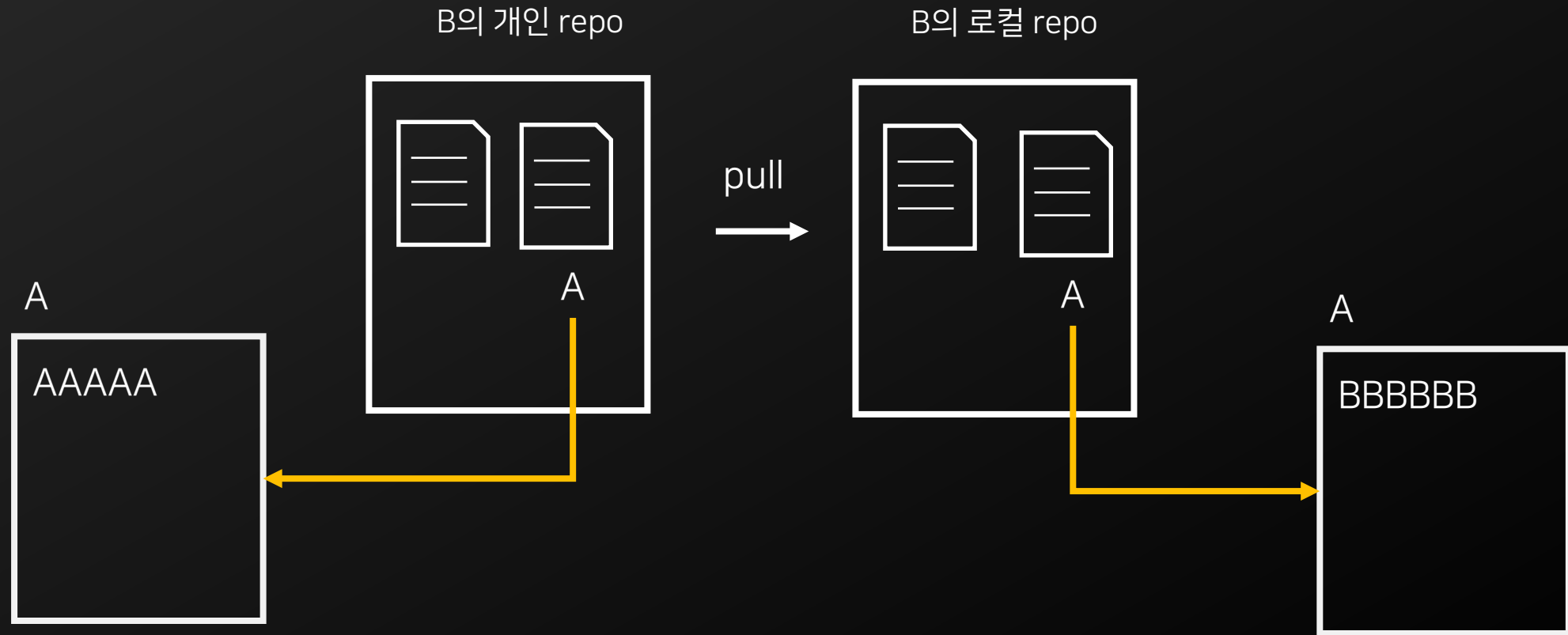
The main area features a commit history table with columns: 그래프, 설명, 날짜, 작성자, and 커밋. The table lists several commits, with the most recent being a merge pull request #1 from jkey20/main.

그래프	설명	날짜	작성자	커밋
	Merge pull request #1 from jkey20/main	25 3 2022 20:28	주이식 <helios2sic@gmail.com>	4571dbe
	test commit	25 3 2022 20:17	210202 <helios2sic@gmail.com>	2cf686b
	Git bash 설치	25 3 2022 18:55	주이식 <helios2sic@gmail.com>	aba040c
	SourceTree 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
	Update README.md	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
	Initial commit	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5fe1e
	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23227a5

Below the table, the '파일 상태' (File Status) section shows the commit hash 4571dbe2cfa0b0c03432927c74cb31a3db3e962b [4571dbe] and the commit message 'test2'. The 'test2.txt' file is shown with a diff view, indicating a new line at the end of the file.

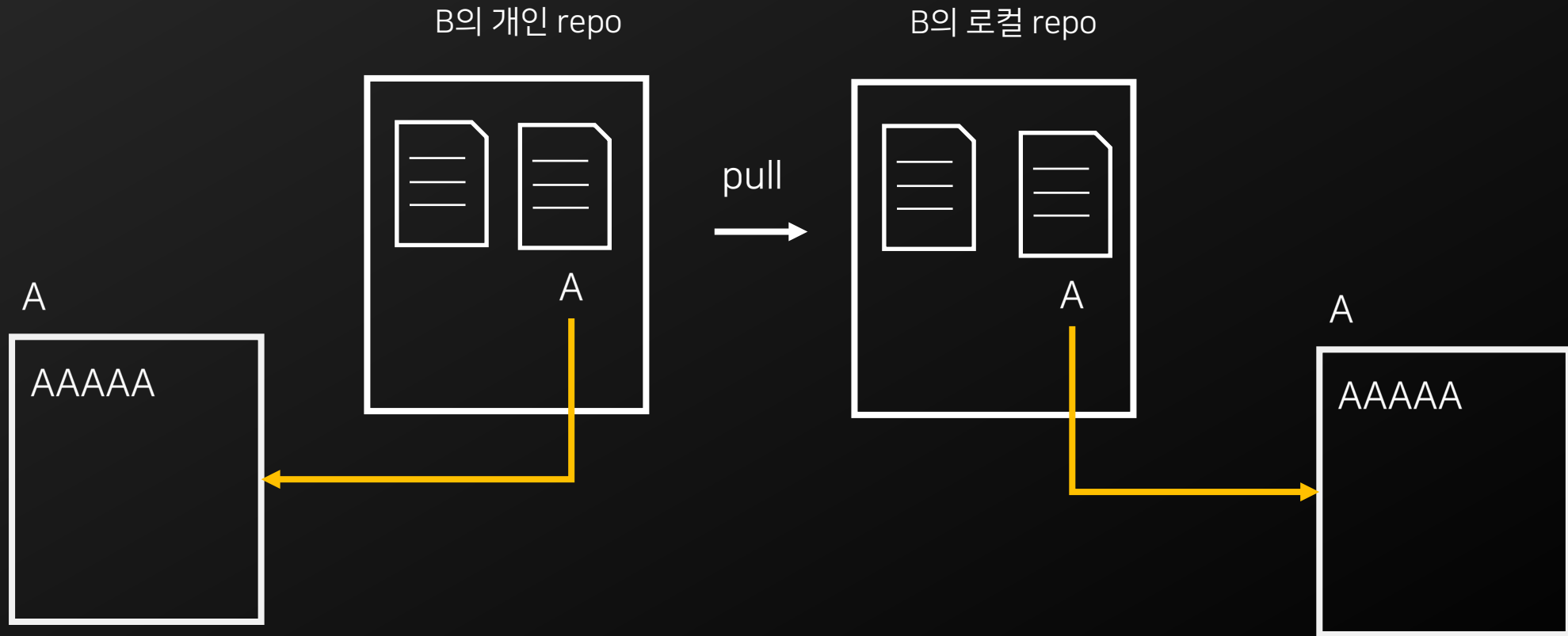
패치를 누르면 커밋을 가져와 변경사항을 확인하고 pull할게 있으면 알려줌

협업할 때 - 원격 저장소에 변경된 데이터가 부분적으로 겹칠때



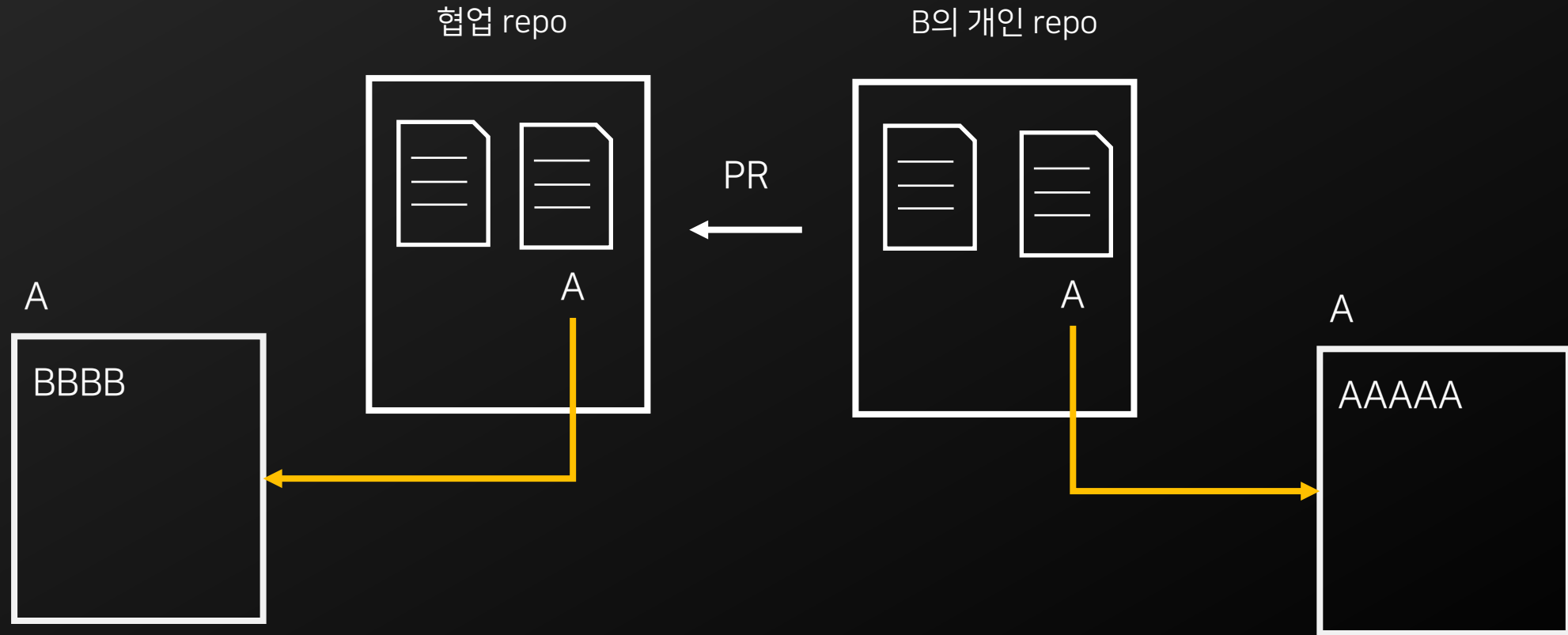
만약 A라는 파일의 내용이 로컬에는 다음과 같고 B의 개인 repo에서 다음과 같을때 pull하면?

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) interface. At the top, the navigation bar includes links for Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. The PR title is "pr #2" and it is currently open. The description states "jkey20 wants to merge 2 commits into sejonginterface:main from jkey20:main". Below the description, there are tabs for Conversation (0), Commits (2), Checks (0), and Files changed (1). A comment from jkey20 says "pr". Below the comment, a commit history shows two commits by jkey20, both updating test.txt, with a "Verified" badge and commit hashes 5a199f4 and 7b729b3. A message at the bottom of the commit history says "Add more commits by pushing to the main branch on jkey20/Study-Git." 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". To the right of this message is a button labeled "Resolve conflicts". On the right side of the PR, there are sections for Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), and Milestone (No milestone).

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

협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

sejonginterface / Study-GitPublic

Pin

Watch0

Fork1

Star2

<> Code

Issues

Pull requests1

Actions

Projects

Wiki

Security

Insights

Settings

pr #2

Resolving conflicts between jkey20:main and sejonginterface:main and committing changes → jkey20:main

1 conflicting file

test.txt

test.txt

<<<<<< main
AAAAAB

AAAAA
>>>>>> main

6

1 conflict

Prev

Next

Mark as resolved

파랑색 부분이 내가 수정한 데이터, 주황색이 원격 저장소의 데이터

협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

The screenshot shows a GitHub Pull Request (PR) for the repository 'sejonginterface / Study-Git'. The PR is titled 'pr #2' and is in the state of 'Resolving conflicts between jkey20:main and sejonginterface:main and committing changes → jkey20:main'. The interface includes a navigation bar with links to Code, Issues, Pull requests (1), Actions, Projects, Wiki, Security, Insights, and Settings. On the right, there are buttons for Pin, Watch (0), Fork (1), and Star (2). The main content area shows a conflict in the file 'test.txt'. The conflict is highlighted with a red bar at the top, indicating '1 conflict'. Below the conflict bar, the file 'test.txt' is shown with a diff view. The diff shows a change from 'AAAAAB' to 'AAAAAB' (no change). The 'Mark as resolved' button is highlighted with an orange border.

sejonginterface / Study-Git Public

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

pr #2

Resolving conflicts between jkey20:main and sejonginterface:main and committing changes → jkey20:main

1 conflicting file

test.txt

1 conflict Prev Next

test.txt

test.txt

1 AAAAAB

2

3

Mark as resolved

수정을 마치면 Mark as resolved를 누름.

협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

The screenshot shows a GitHub Pull Request (PR) for the repository 'sejonginterface / Study-Git'. The PR is titled 'pr #2' and is in the state of 'Resolving conflicts between jkey20:main and sejonginterface:main and committing changes → jkey20:main'. A green 'Commit merge' button is highlighted with an orange border. Below the title, it indicates '1 conflicting file' and shows a list of files with a green checkmark next to 'test.txt'. The content of 'test.txt' is displayed, showing three lines of text: '1 AAAAA', '2', and '3'. The status '✓ Resolved' is shown in the top right corner of the file view area.

sejonginterface / Study-Git Public

<> Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

pr #2

Resolving conflicts between jkey20:main and sejonginterface:main and committing changes → jkey20:main

Commit merge

1 conflicting file

test.txt

test.txt

test.txt

test.txt

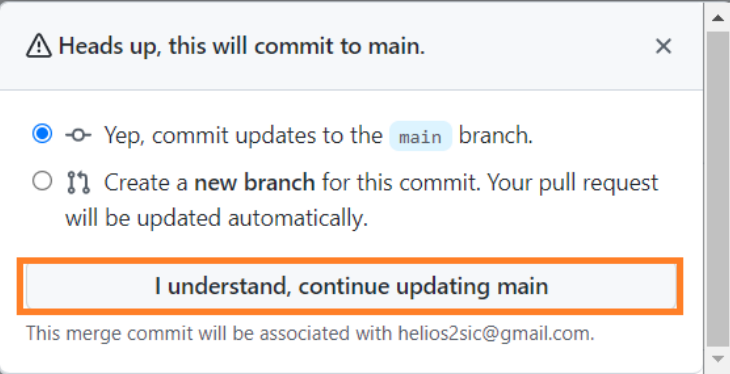
1 AAAAA

2

3

✓ Resolved

Commit merge를 눌러 적용



협업할 때 - 원격 저장소에 겹치는 데이터를 PR보낼때

The screenshot displays a GitHub Pull Request (PR) for the repository 'sejonginterface:main' from the branch 'jkey20:main'. The PR is titled 'jkey20 wants to merge 3 commits into sejonginterface:main from jkey20:main'. The interface shows a conversation tab with a comment from 'jkey20' stating 'pr'. Below the comment, a commit history is shown with three commits: 'Update test.txt' (Verified, 5a199f4), 'Update test.txt' (Verified, 7b729b3), and 'Merge branch 'main' into main' (Verified, 1ba3974). A green box highlights the status: 'Continuous integration has not been set up' and 'This branch has no conflicts with the base branch'. A green button labeled 'Merge pull request' is visible. The right sidebar shows sections for Reviewers, Assignees, Labels, Projects, Milestone, Development, and Notifications. The bottom of the interface shows a 'Write' tab and a 'Preview' tab.

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 Member

pr

jkey20 added 3 commits 18 hours ago

- Update test.txt Verified 5a199f4
- Update test.txt Verified 7b729b3
- 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](#).

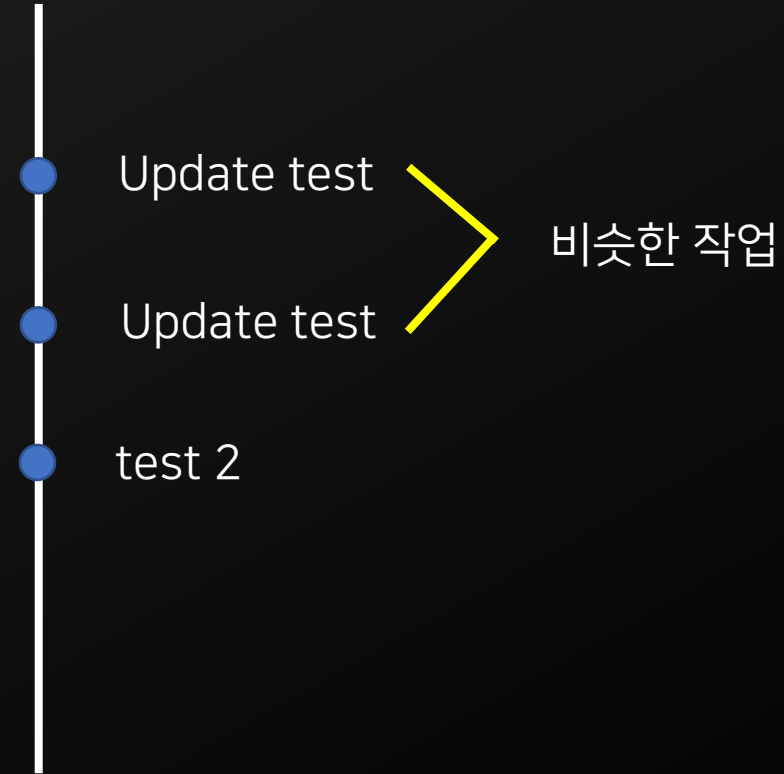
Write Preview H B I ≡ <> @ ↩

Notifications Customize Unsubscribe

Merge pull request가 정상적으로 나온다.

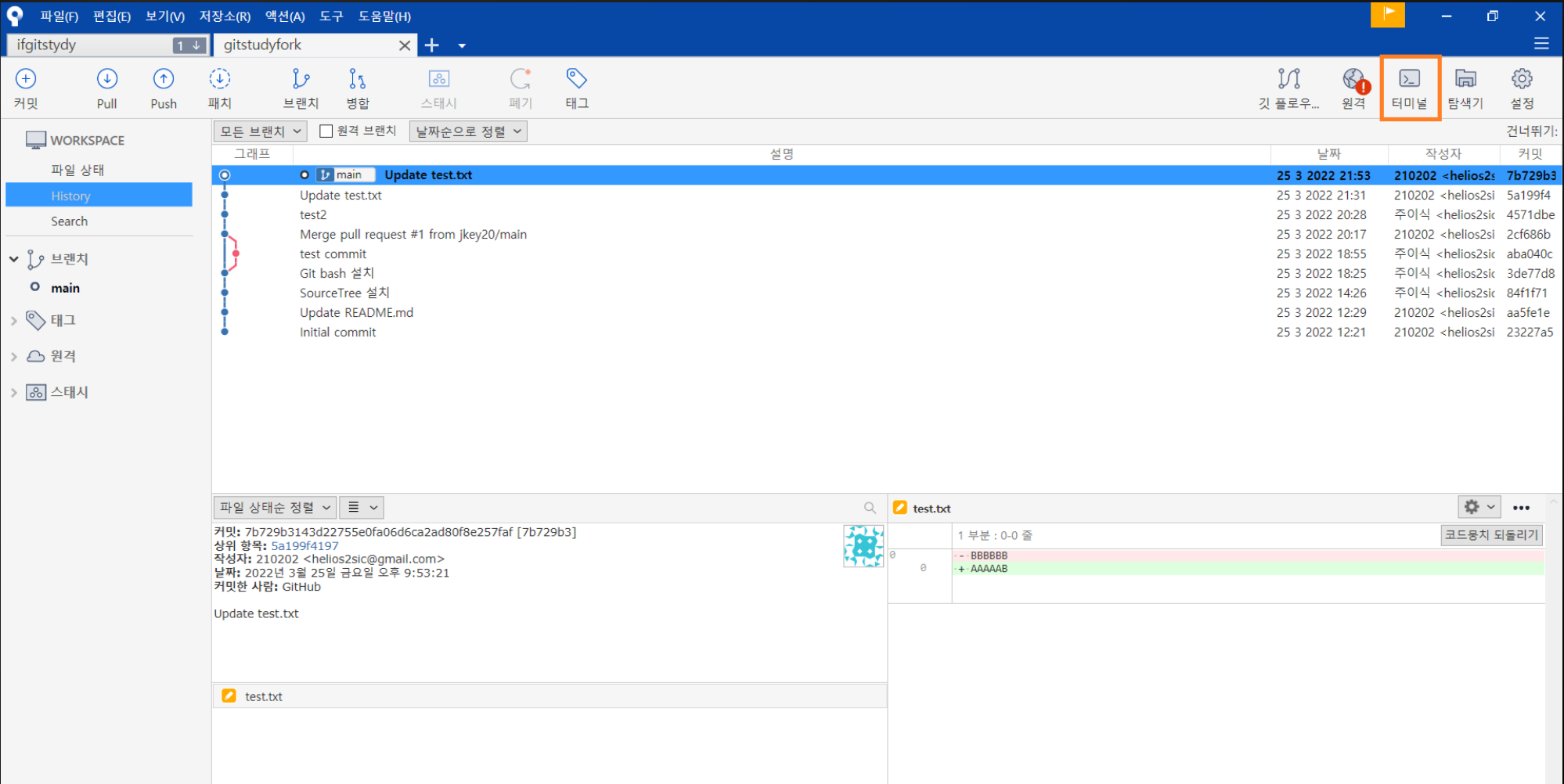
협업할 때 - 커밋 합치기

B의 로컬 repo



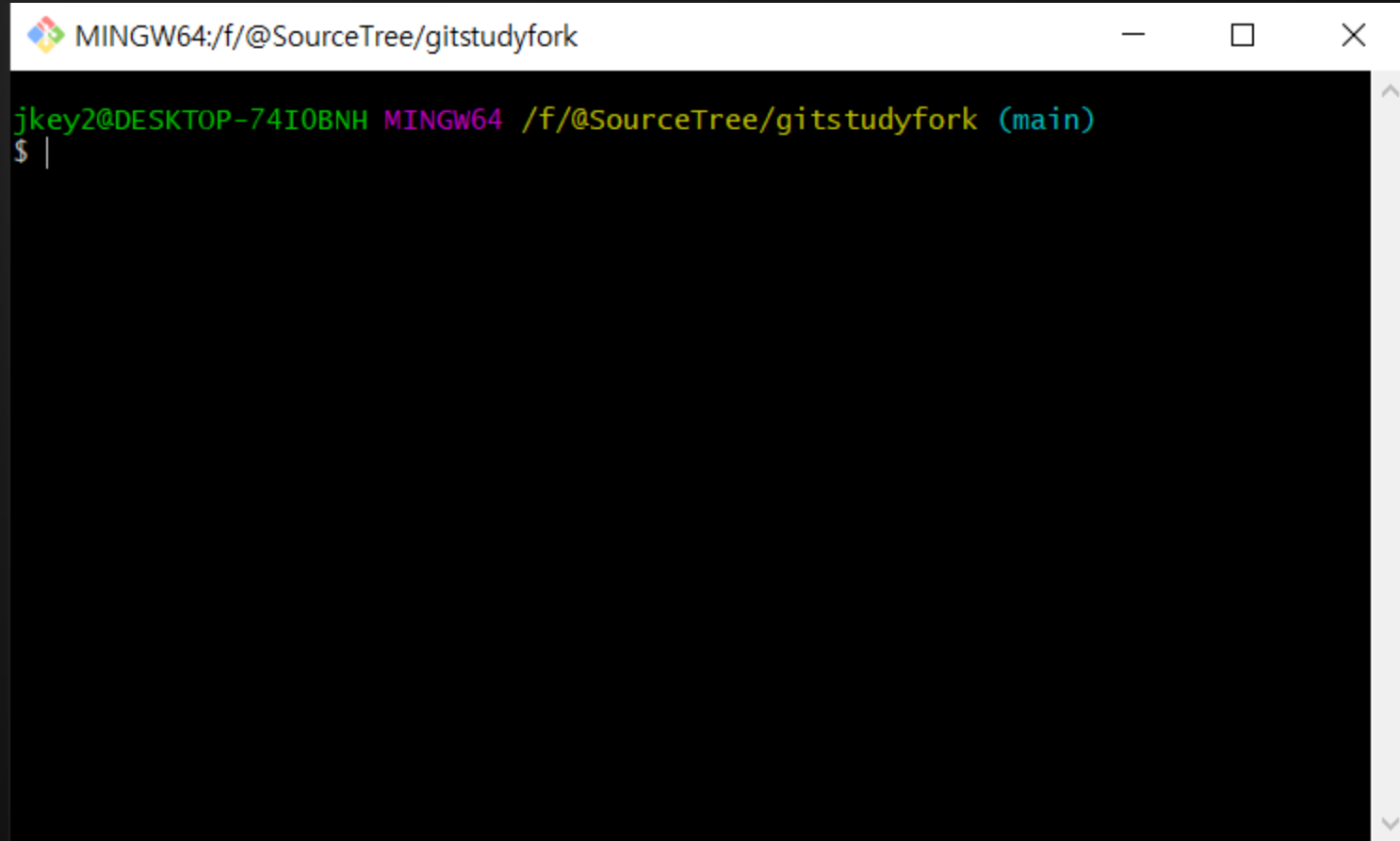
커밋을 여러개 했는데 비슷한 작업인 경우가 생김. 그냥 push를 하면 너무 커밋이 많아지는데 어떻게 할까?

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

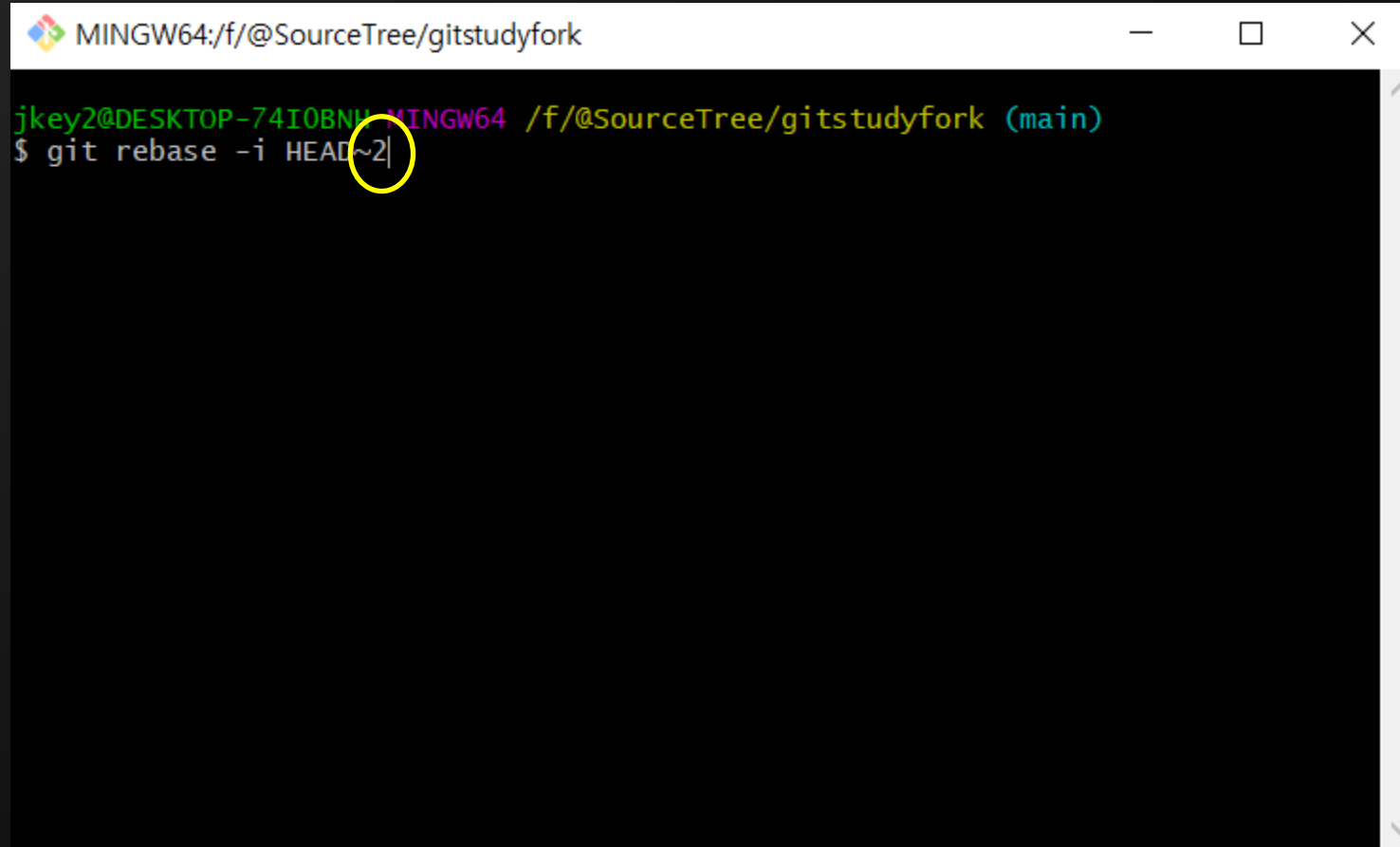


A screenshot of a Windows command prompt window. The title bar shows the icon for a Windows application, followed by the text "MINGW64:/f/@SourceTree/gitstudyfork". The window has standard minimize, maximize, and close buttons. The command prompt shows the user "jkey2@DESKTOP-74I0BNH" in a "MINGW64" environment, at the path "/f/@SourceTree/gitstudyfork" on the "main" branch. The prompt is "\$ |" and the command "git commit -m 'add test'" has been entered.

```
MINGW64:/f/@SourceTree/gitstudyfork
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git commit -m 'add test'
```

다음과 같이 검은창이 나오면 성공

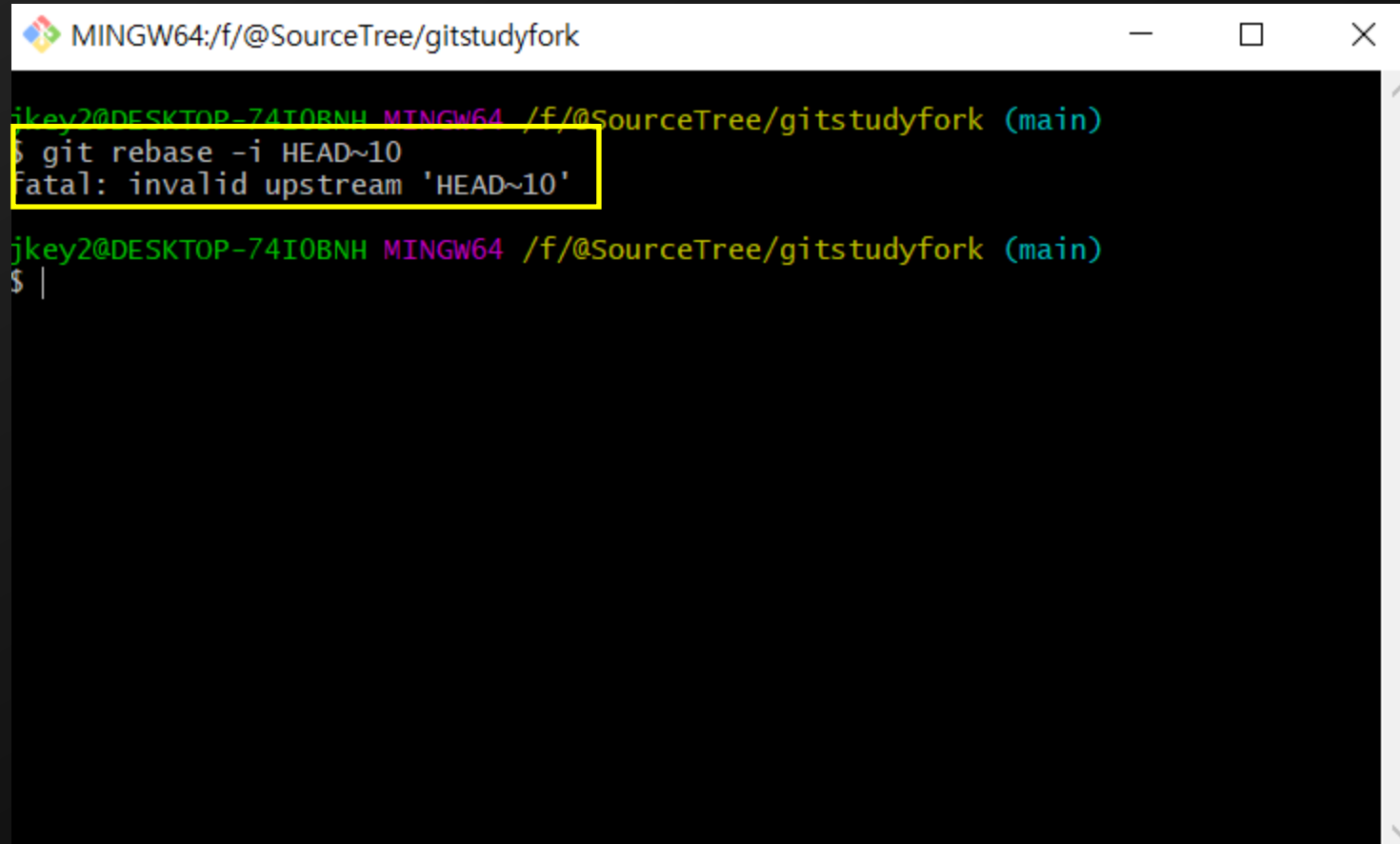
협업할 때 - 커밋 합치기



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

다음과 같이 커맨드를 입력함 2는 합칠 커밋의 개수

협업할 때 - 커밋 합치기

A screenshot of a Windows terminal window titled 'MINGW64:/f/@SourceTree/gitstudyfork'. The window shows a user attempting to run 'git rebase -i HEAD~10', which results in a 'fatal: invalid upstream 'HEAD~10'' error. The error message is highlighted with a yellow box. The terminal text is as follows:

```
MINGW64:/f/@SourceTree/gitstudyfork  
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)  
$ |
```

만약 합칠 커밋의 개수가 현재 존재하는 커밋보다 많다면 다음과 같이 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
<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
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

```
MINGW64:/f/@SourceTree/gitstudyfork
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

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

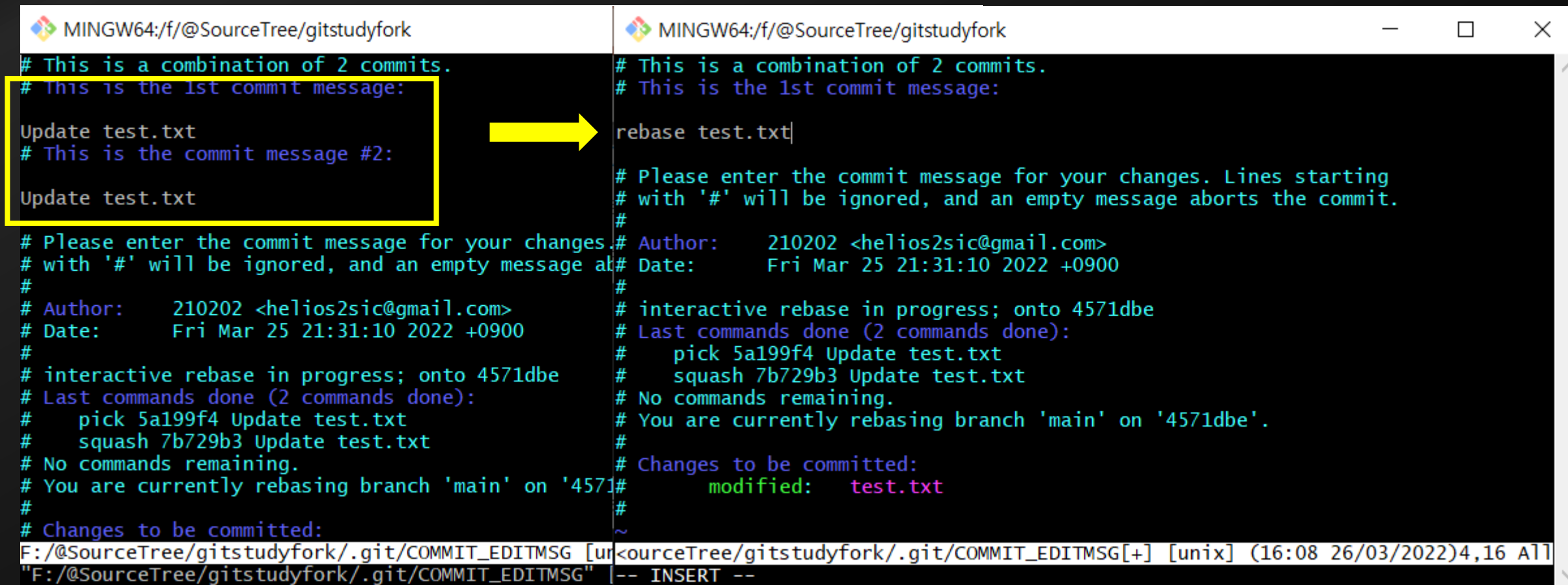
Update test.txt
# This is the commit message #2:

Update test.txt

# Please enter the commit message for your changes. 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 만들

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. 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:
#   modified:   test.txt

F:/@SourceTree/gitstudyfork/.git/COMMIT_EDITMSG [un]
"F:/@SourceTree/gitstudyfork/.git/COMMIT_EDITMSG" -- INSERT --

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

~
sourceTree/gitstudyfork/.git/COMMIT_EDITMSG[+] [unix] (16:08 26/03/2022)4,16 All
-- INSERT --
```

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

협업할 때 - 커밋 합치기

```
MINGW64:/f/@SourceTree/gitstudyfork

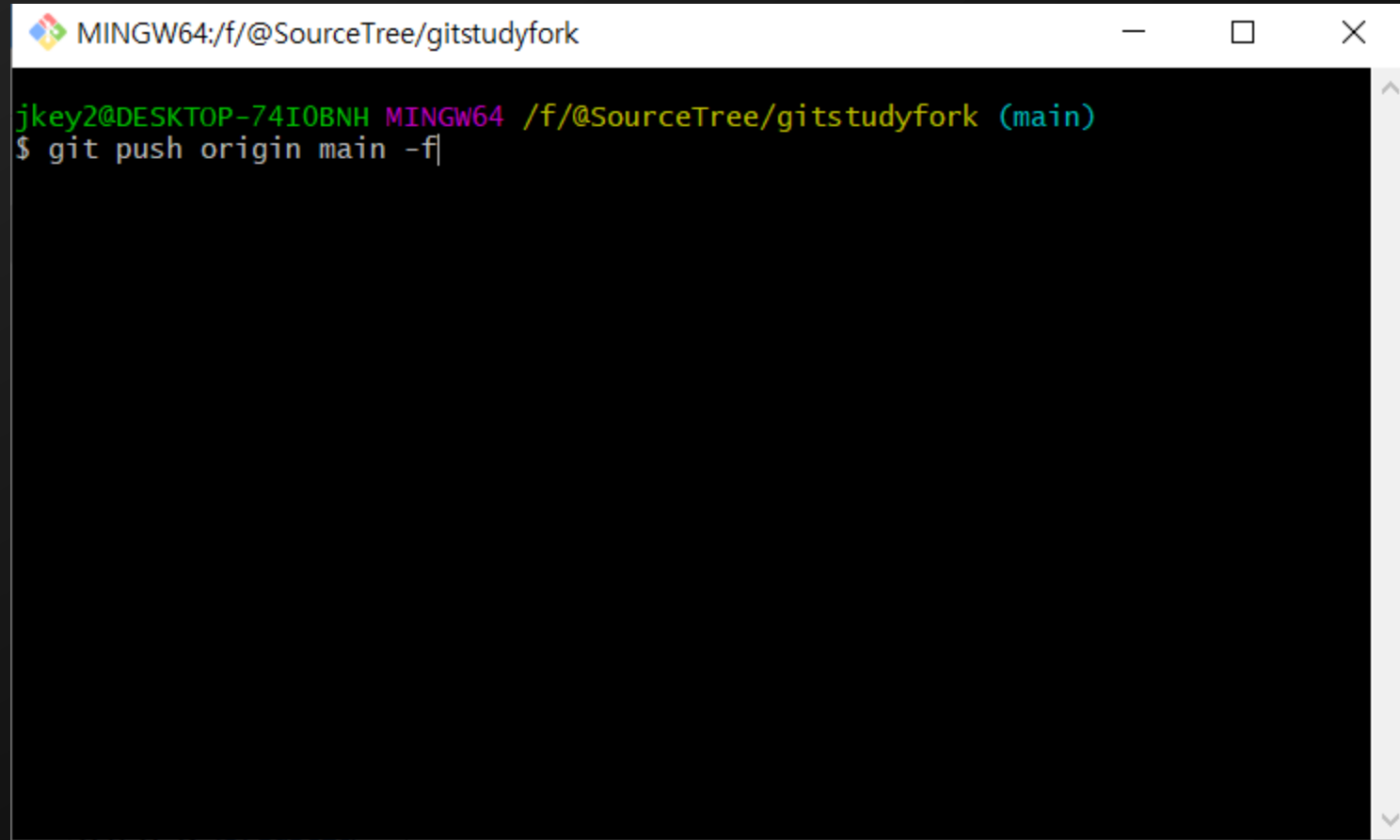
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git rebase -i HEAD~10
fatal: invalid upstream 'HEAD~10'

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



```
MINGW64:/f/@SourceTree/gitstudyfork
jkey2@DESKTOP-74I0BNH MINGW64 /f/@SourceTree/gitstudyfork (main)
$ git push origin main -f
```

이제 변경사항을 푸시하기 위해서 다음과 같이 커맨드 입력

협업할 때 - 커밋 합치기

```
MINGW64:/f/@SourceTree/gitstudyfork
+ 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-Cit.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 하기위해서 현재 버전의 Branch를 만들어 다른 작업을 이어나가는 것

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

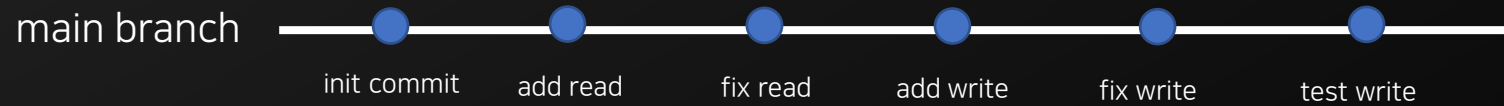
또한 A,B,C 기능을 만들다고 했을 때 A,B,C 기능을 각자 작업할때 커밋들이 순서대로 있자않고 너무 많아져서 정렬하기 힘들.
그래서 branch를 통해서 현재 작업의 분기를 만들어서 작업함.

모든 repo는 처음 생성하면 main 브랜치가 존재.

Branch

예시 상황) A는 read, B는 write 기능을 만들려고 한다.

협업 repo

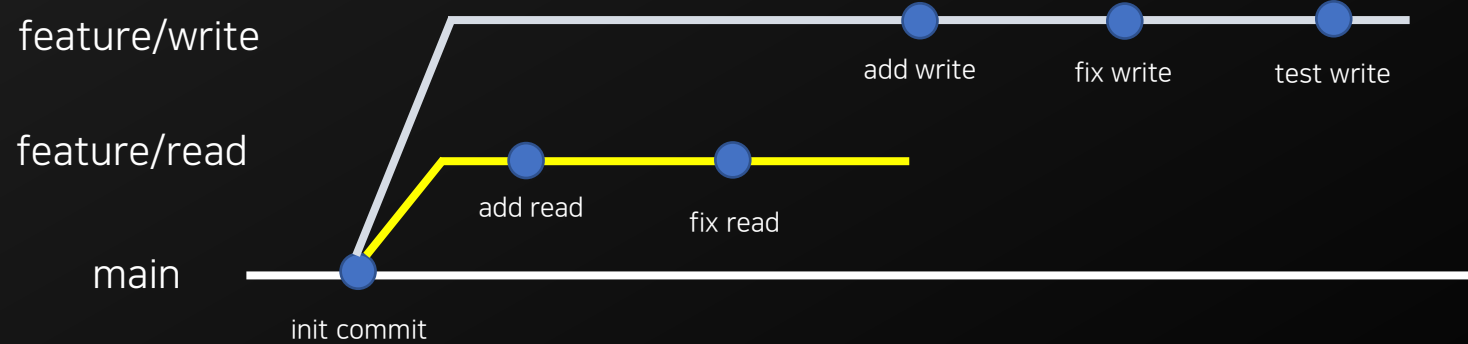


브랜치를 사용하지 않으면 다음과 같이 커밋들이 섞여있어서 알아보기 힘들.

Branch

예시 상황) A는 read, B는 write 기능을 만들려고 한다.

협업 repo

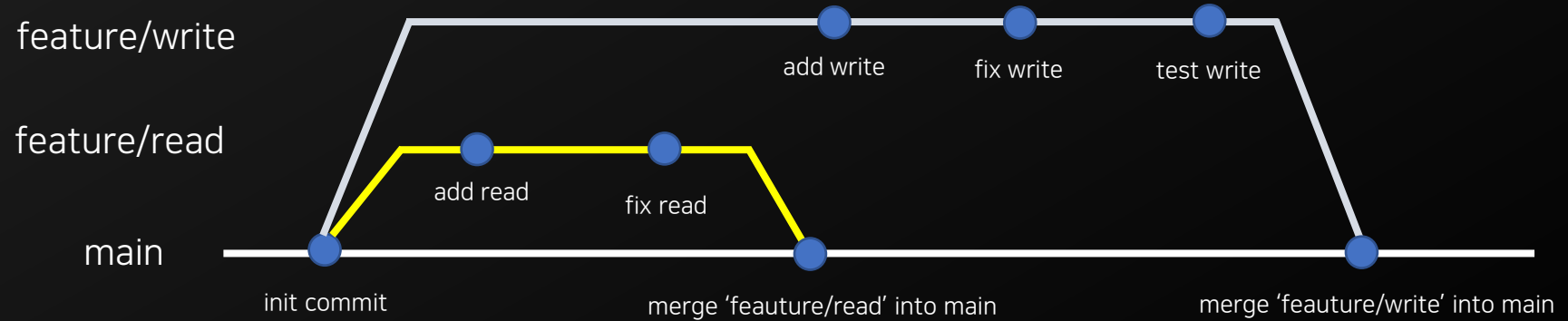


브랜치를 나누면 다음과 같이 커밋을 한 시간순서대로 분류되고 기능에 맞게 분류되어서 보기편함

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

The screenshot shows the Git GUI application window. The top toolbar contains several icons, and the '브랜치' (Branch) icon, which depicts a branching diagram, is highlighted with a red rectangle. Below the toolbar, the 'WORKSPACE' sidebar on the left lists '파일 상태' (File Status), 'History', 'Search', '브랜치' (Branch), '태그' (Tag), '원격' (Remote), and '스태시' (Stash). The '브랜치' section is expanded, showing the 'main' branch. The main area displays a commit history table with columns for '그래프' (Graph), '설명' (Description), '날짜' (Date), '작성자' (Author), and '커밋' (Commit). The table lists several commits, with the most recent one being 'rebase test.txt' by '210202 <helios2sic@gmail.com>' with commit hash 'c02625b'. Below the table, the '파일 상태' (File Status) section shows the current commit's details, including the commit hash, author, date, and a list of files. The 'test.txt' file is shown with a diff view, indicating a change from '0' to 'AAAAAB'.

그래프	설명	날짜	작성자	커밋
● main	rebase test.txt	26 3 2022 16:08	210202 <helios2sic@gmail.com>	c02625b
●	test2	25 3 2022 20:28	주이식 <helios2sic@gmail.com>	4571dbe
●	Merge pull request #1 from jkey20/main	25 3 2022 20:17	210202 <helios2sic@gmail.com>	2cf686b
●	test commit	25 3 2022 18:55	주이식 <helios2sic@gmail.com>	aba040c
●	Git bash 설치	25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
●	SourceTree 설치	25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
●	Update README.md	25 3 2022 12:29	210202 <helios2sic@gmail.com>	aa5fe1e
●	Initial commit	25 3 2022 12:21	210202 <helios2sic@gmail.com>	23227a5

커밋: c02625b29b78d5133b9268ff08caa6777cd7e602 [c02625b]
상위 항목: 4571dbe2cf
작성자: 210202 <helios2sic@gmail.com>
날짜: 2022년 3월 25일 금요일 오후 9:31:10
커밋한 사람: 주이식
커밋 날짜: 2022년 3월 26일 토요일 오후 4:08:12

rebase test.txt

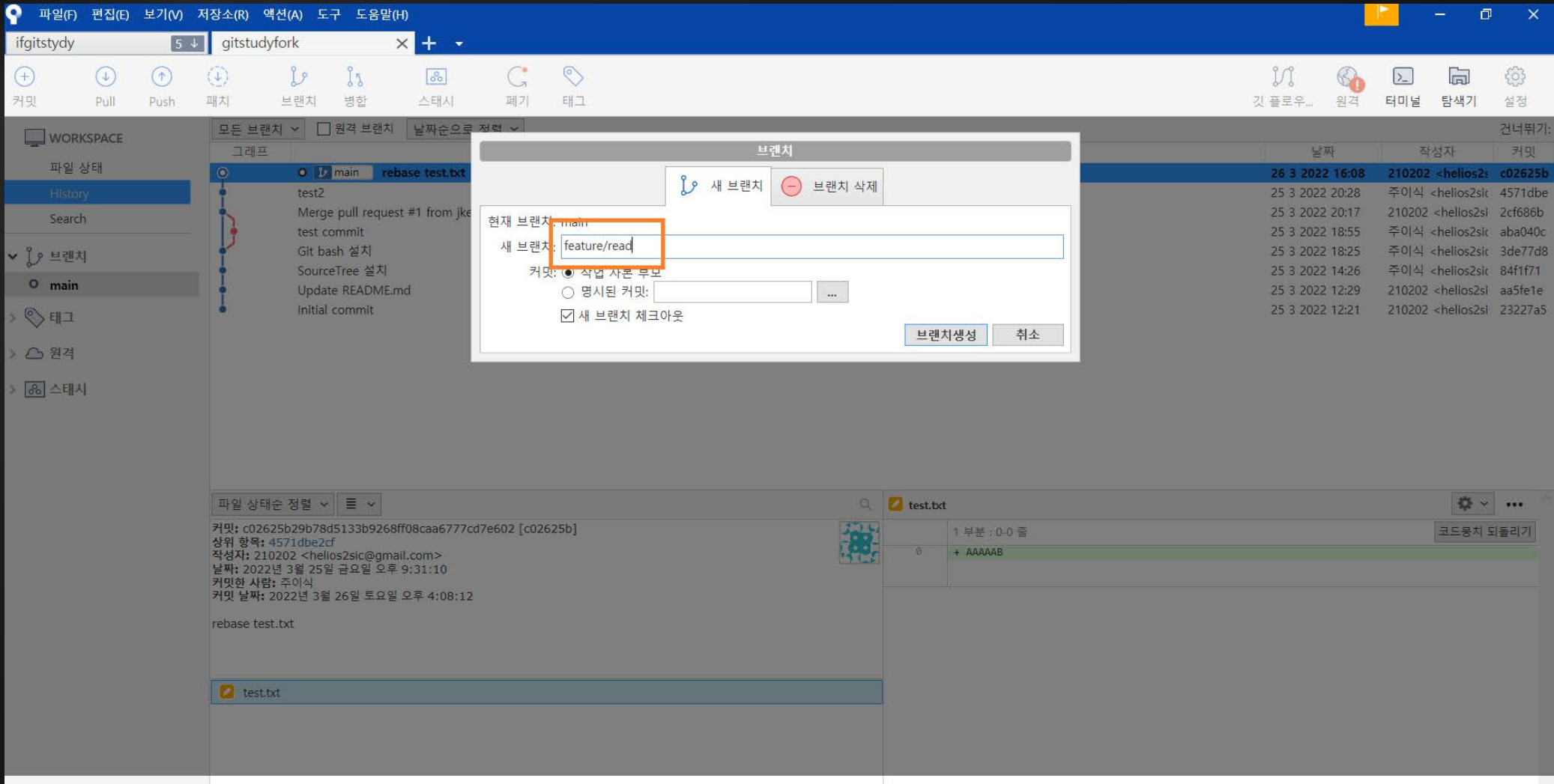
test.txt

1 부분: 0-0 줄
+ AAAAAB

코드용치 되돌리기

브랜치 버튼을 누른다.

사용방법 - Branch



브랜치의 이름을 입력하고 브랜치 생성을 누른다.

사용방법 - Branch

The screenshot shows the Git GUI interface with the following components:

- Top Bar:** Contains menu items (파일(F), 편집(E), 보기(V), 저장소(R), 액션(A), 도구, 도움말(H)) and a toolbar with icons for commit, pull, push, patch, branch, merge, stash, and tag.
- Left Panel (WORKSPACE):** Shows the file state, history, search, and branches. The 'branches' section is expanded, showing a 'feature' branch with a sub-branch 'read' highlighted by an orange box.
- Graph View:** Displays a commit graph with a red line indicating the current branch. The commit history is shown in a table with columns for commit hash, date, author, and commit message.
- Commit History Table:**

날짜	작성자	커밋
26 3 2022 16:16	주이식 <helios2si>	dda4645
25 3 2022 20:17	210202 <helios2si>	2cf686b
25 3 2022 18:55	주이식 <helios2sic>	aba040c
25 3 2022 18:25	주이식 <helios2sic>	3de77d8
25 3 2022 14:26	주이식 <helios2sic>	84f1f71
25 3 2022 12:29	210202 <helios2si>	aa5fe1e
25 3 2022 12:21	210202 <helios2si>	23227a5
- File View:** Shows the file 'test.txt' with a diff view. The diff shows a change from '0' to 'AAAAAB'.

그러면 왼쪽에 브랜치가 생성된 모습을 볼 수 있다.

사용방법 - Branch

The screenshot shows the Git GUI interface with the 'feature/read' branch selected. The commit history table is as follows:

커밋	날짜	작성자	커밋
792eaa7	26 3 2022 16:46	주이식 <helios2si>	test3
dda4645	26 3 2022 16:16	주이식 <helios2si>	rebase test.txt
2cf686b	25 3 2022 20:17	210202 <helios2si>	Merge pull request #1 from jkey20/main
aba040c	25 3 2022 18:55	주이식 <helios2si>	test commit
3de77d8	25 3 2022 18:25	주이식 <helios2si>	Git bash 설치
84f1f71	25 3 2022 14:26	주이식 <helios2si>	SourceTree 설치
aa5fe1e	25 3 2022 12:29	210202 <helios2si>	Update README.md
23227a5	25 3 2022 12:21	210202 <helios2si>	Initial commit

Below the table, the commit details for 'test3' are shown:

커밋: 792eaa767715542b7ec7262eb98eb4c355e16d54 [792eaa7]
상위 항목: dda4645dba
작성자: 주이식 <helios2si@gmail.com>
날짜: 2022년 3월 26일 토요일 오후 4:46:50
커밋한 사람: 주이식

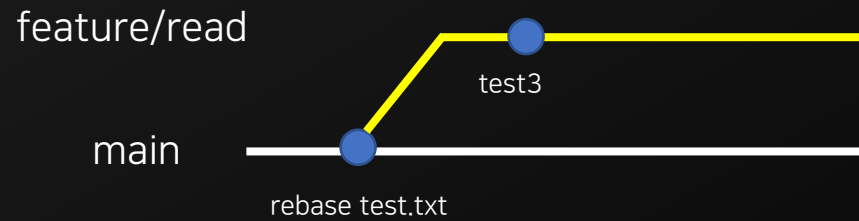
The file 'test3.txt' is shown in the bottom pane with the message: 'No changes in this file have been detected, or it is a binary file'.

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 the GitHub interface for the repository **jkey20 / Study-Git**. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the repository name, there are buttons for Pin, Watch (0), Fork (1), and Star (0). The main navigation bar shows the 'Pull requests' tab selected. A yellow banner at the top of the pull requests section indicates that the **feature/read** branch has recent pushes 4 minutes ago, with a 'Compare & pull request' button. Below this, there is a search bar with the filter 'is:pr is:open' and buttons for Labels (9) and Milestones (0). A green button labeled 'New pull request' is highlighted with an orange box. The main content area displays a 'Welcome to pull requests!' message, explaining that pull requests help collaborate on code and that they will appear in a searchable and filterable list. A 'ProTip!' at the bottom suggests clicking a checkbox on the left to edit multiple issues at once.

개인 repo의 pull request로 가서 new pull request를 누른다.

사용방법 - Branch

sejonginterface / Study-Git Public

<> 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

Choose a Base Repository

Filter repos

sejonginterface/Study-Git

jkey20/Study-Git

1 commit

2 files changed

1 contributor

Commits on Mar 26, 2022

rebase test.txt

jkey20 committed 38 minutes ago

Showing 2 changed files with 2 additions and 0 deletions.

Split Unified

1 test.txt

여기서 개인 repo를 선택해준다.

사용방법 - Branch

forked from [sejonginterface/Study-Git](#)

<> Code 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: main ← compare: main

Choose a head ref

Find a branch

Branches Tags

✓ main default

feature/read

here isn't anything to compare.

Use two different branch names to get a valid comparison. Check out some of these sample comparisons.

Example comparisons	
feature/read	8 minutes ago
main@{1 day}...main	24 hours ago

Showing 0 changed files with 0 additions and 0 deletions.

Split Unified

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

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

사용방법 - Branch

The screenshot displays the SourceTree application interface. The top toolbar includes buttons for Commit, Pull, Push, Patch, Branch, Merge, Stash, Revert, and Tag. The left sidebar shows the Workspace with options for File Status, History, Search, Branches, Tags, Remotes, and Stashes. The main area is divided into three panes: a commit graph on the left, a commit list in the center, and a file view on the right.

The commit list pane shows a table of commits. The commit 'rebase test.txt' (commit hash dda4645dbac81d5573620447e3971f21134a374d) is highlighted, indicating it is the current commit. The commit message for this commit is 'Merge pull request #1 from jkey20/main'. The commit list also shows the previous commit 'Merge pull request #1 from jkey20/feature/read' (commit hash bb13c3e).

The file view pane shows the file 'test.txt' with a new addition (green line) containing the text 'AAAAAB'.

날짜	작성자	커밋
26 3 2022 16:56	210202 <helios2s...	bb13c3e
26 3 2022 16:46	주이식 <helios2sic@gmail.com>	792eaa7
26 3 2022 16:16	주이식 <helios2sic@gmail.com>	dda4645
25 3 2022 20:17	210202 <helios2si...	2cf686b
25 3 2022 18:55	주이식 <helios2sic@gmail.com>	aba040c
25 3 2022 18:25	주이식 <helios2sic@gmail.com>	3de77d8
25 3 2022 14:26	주이식 <helios2sic@gmail.com>	84f1f71
25 3 2022 12:29	210202 <helios2si...	aa5fe1e
25 3 2022 12:21	210202 <helios2si...	23227a5

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