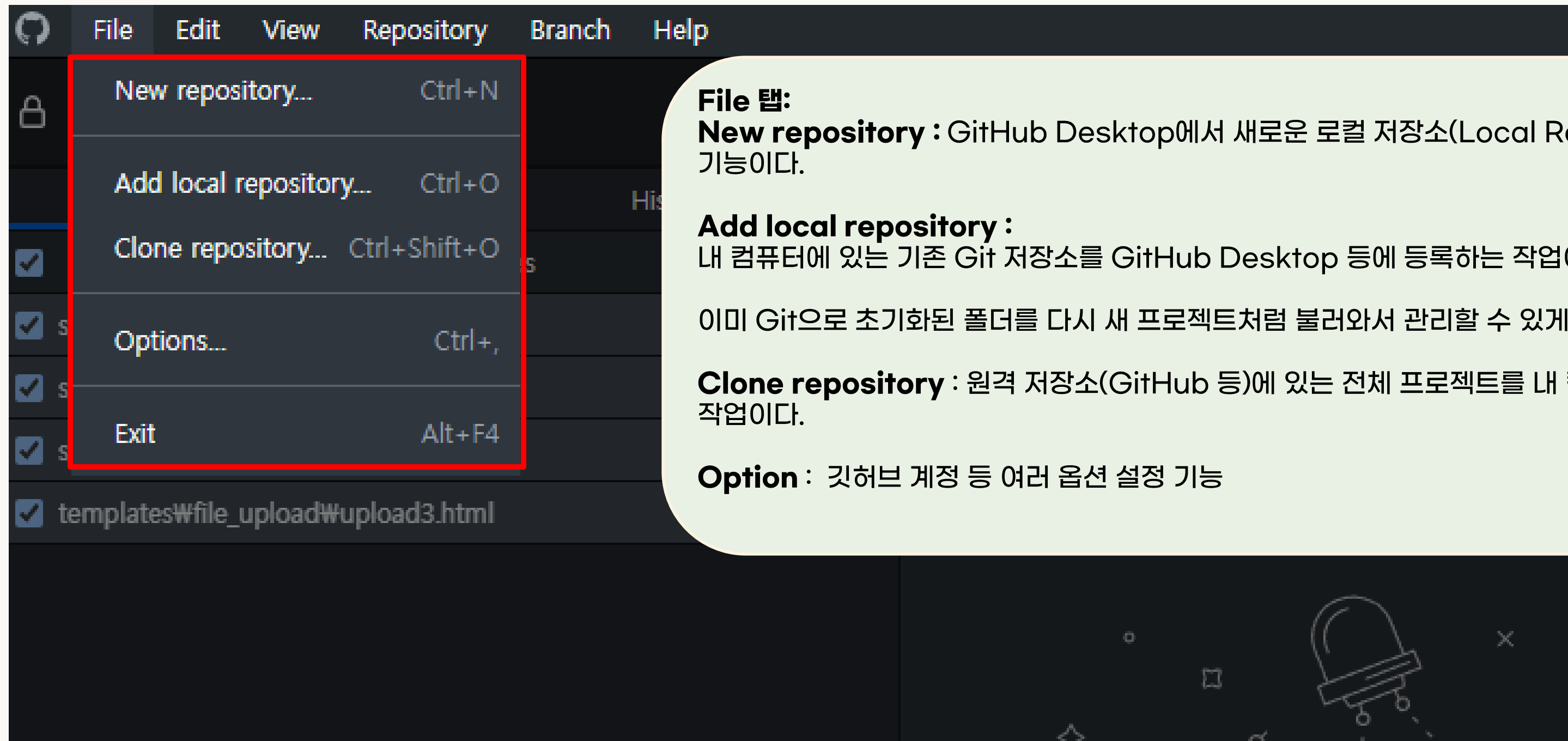


# Github Desktop

## 가이드



# File 탭



## File 탭:

**New repository** : GitHub Desktop에서 새로운 로컬 저장소(Local Repository)를 생성하는 기능이다.

## Add local repository :

내 컴퓨터에 있는 기존 Git 저장소를 GitHub Desktop 등에 등록하는 작업이다.

이미 Git으로 초기화된 폴더를 다시 새 프로젝트처럼 불러와서 관리할 수 있게 해주는 기능이다.

**Clone repository** : 원격 저장소(GitHub 등)에 있는 전체 프로젝트를 내 컴퓨터로 복사해오는 작업이다.

**Option** : 깃허브 계정 등 여러 옵션 설정 기능

# New repository

**Create a new repository**

1 Name repository name

2 Description

3 Local path D:\W Choose...

4 ☐ Initialize this repository with a README

5 Git ignore None

6 License None

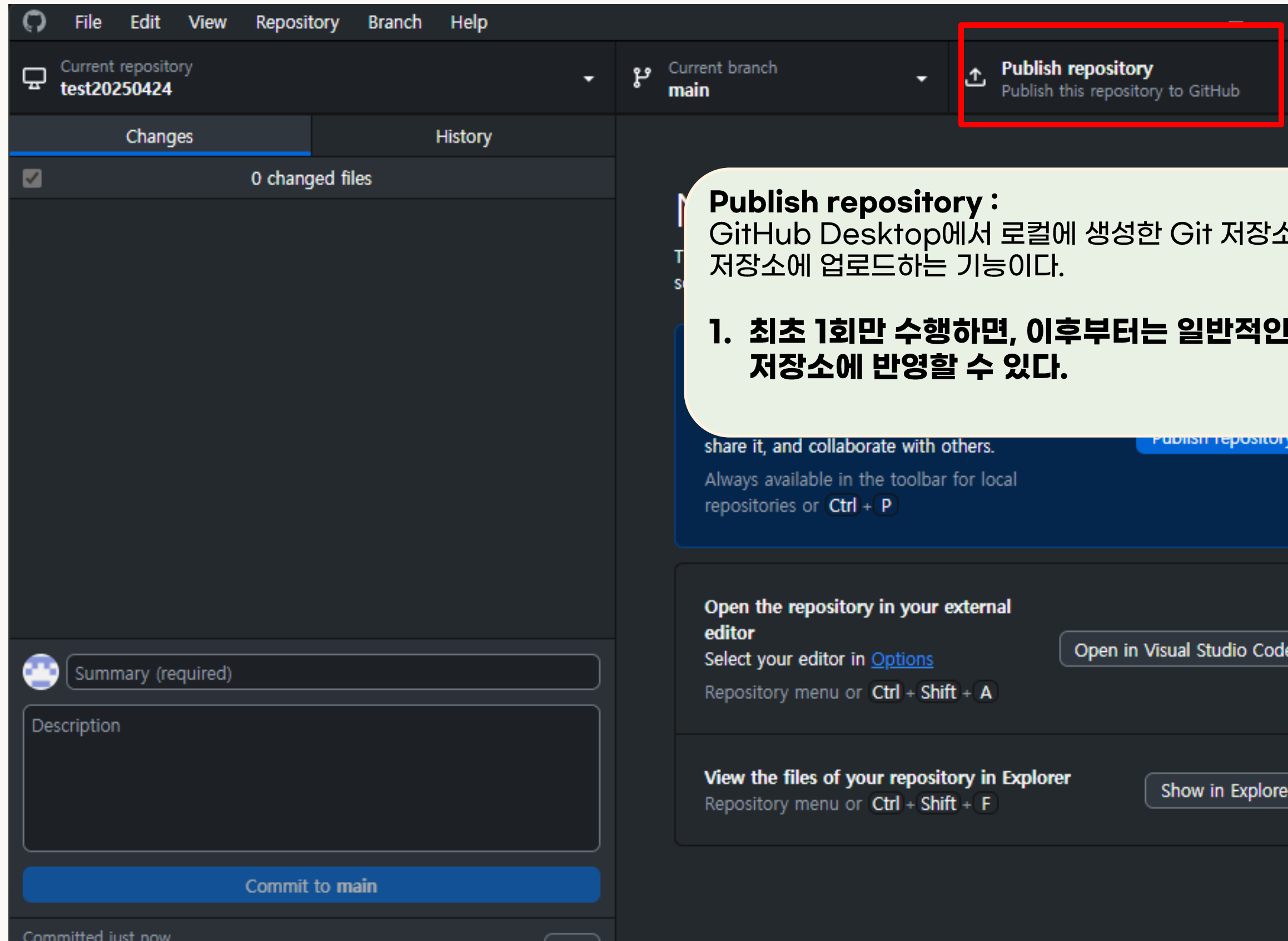
7 Create repository Cancel

## New repository :

GitHub Desktop에서 새로운 로컬 저장소(Local Repository)를 생성하는 기능이다.

1. repository(저장소) 이름 설정
2. 저장소에 대한 설명
3. 폴더 생성 경로 지정
4. Readme 파일 생성 (선택)
5. Git이 추적하지 않아야 할 파일이나 폴더를 지정하는 설정 파일
6. 소프트웨어나 프로젝트에 대한 사용 권한과 조건을 명시한 문서
7. 로컬 저장소 생성

# Publish repository

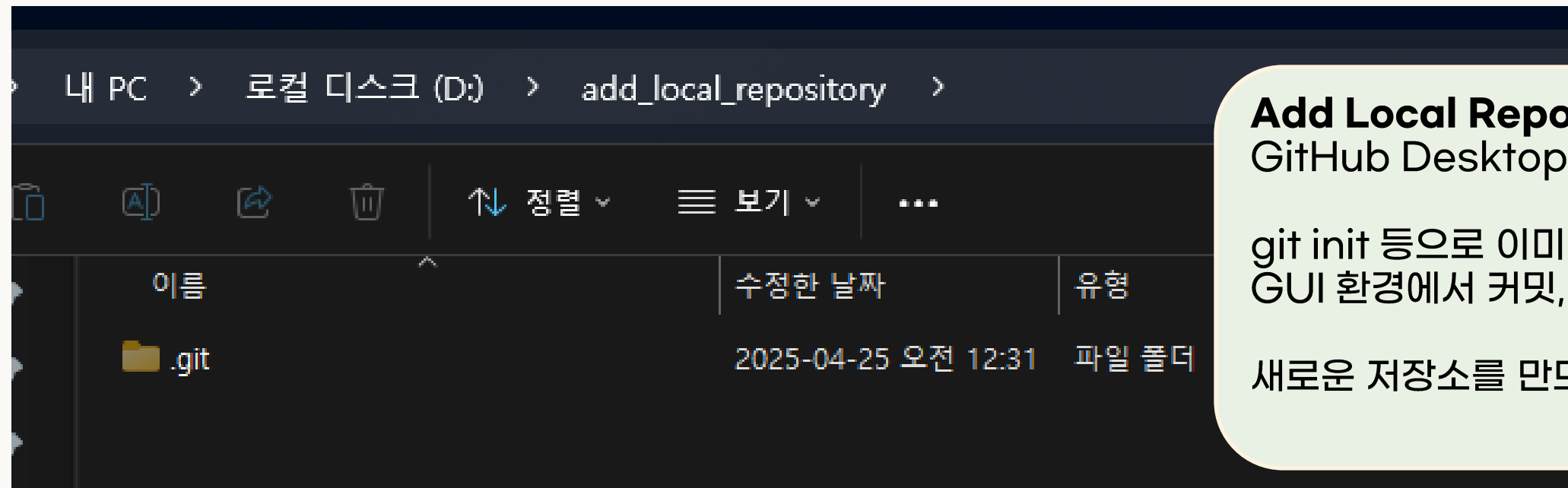


## Publish repository :

GitHub Desktop에서 로컬에 생성한 Git 저장소를 GitHub 원격 저장소에 업로드하는 기능이다.

1. 최초 1회만 수행하면, 이후부터는 일반적인 push로 계속 원격 저장소에 반영할 수 있다.

# Add Local Repository

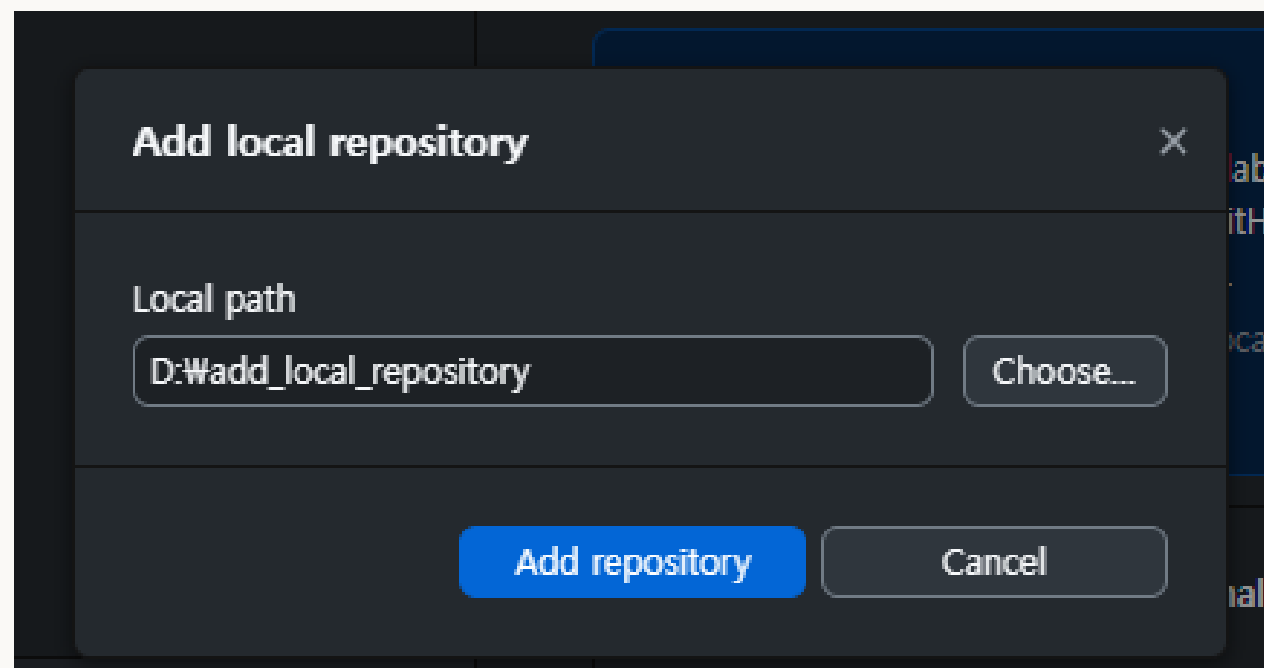


## Add Local Repository :

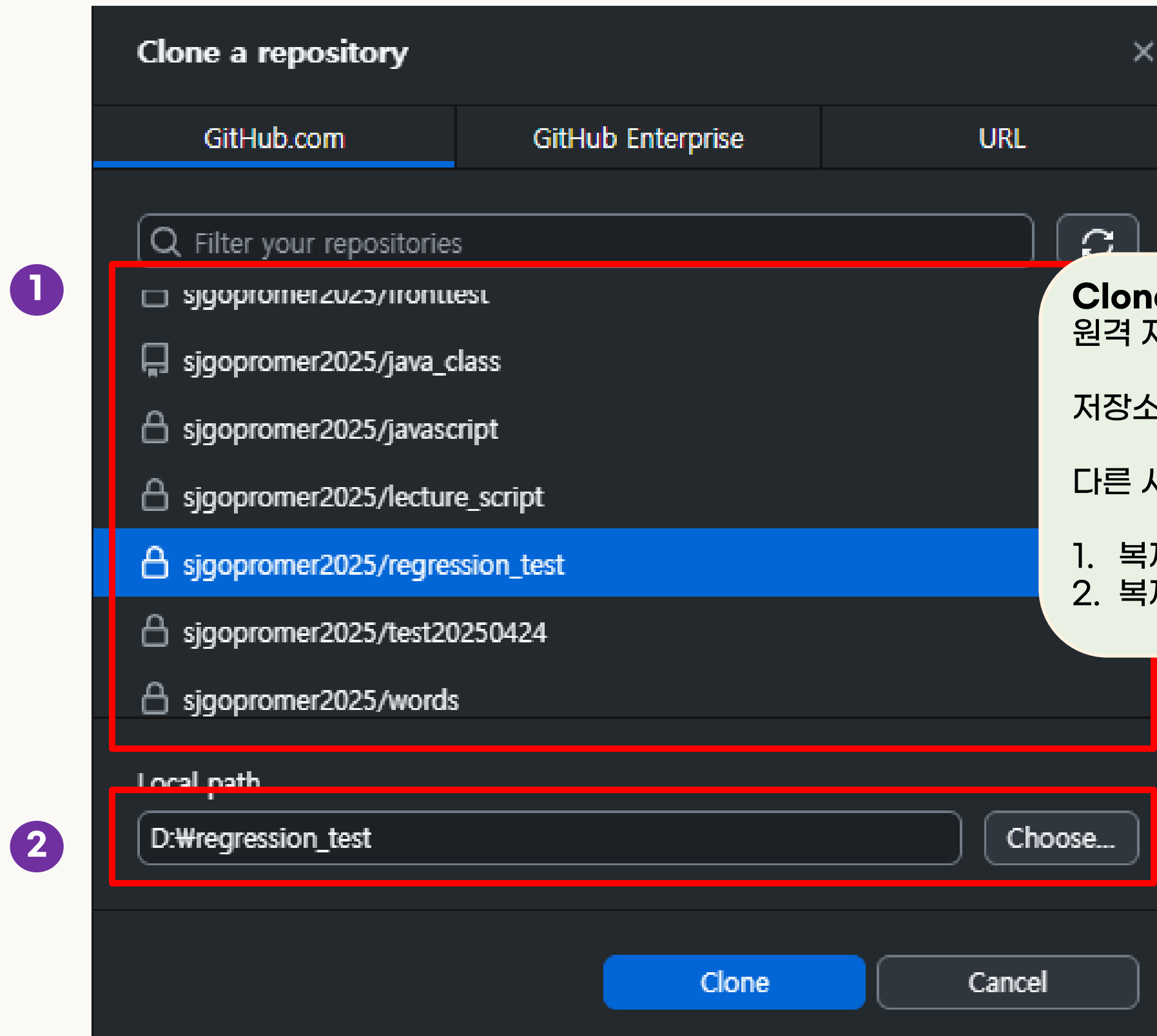
GitHub Desktop에서 이미 내 컴퓨터에 존재하는 Git 저장소를 불러오는 기능이다.

git init 등으로 이미 Git으로 관리 중인 폴더를 GitHub Desktop에 등록하여 GUI 환경에서 커밋, 푸시 등의 작업을 쉽게 할 수 있도록 해주는 기능이다.

새로운 저장소를 만드는 것이 아니라, 기존의 로컬 저장소를 연결하는 것이다.



# Clone repository



## Clone Repository :

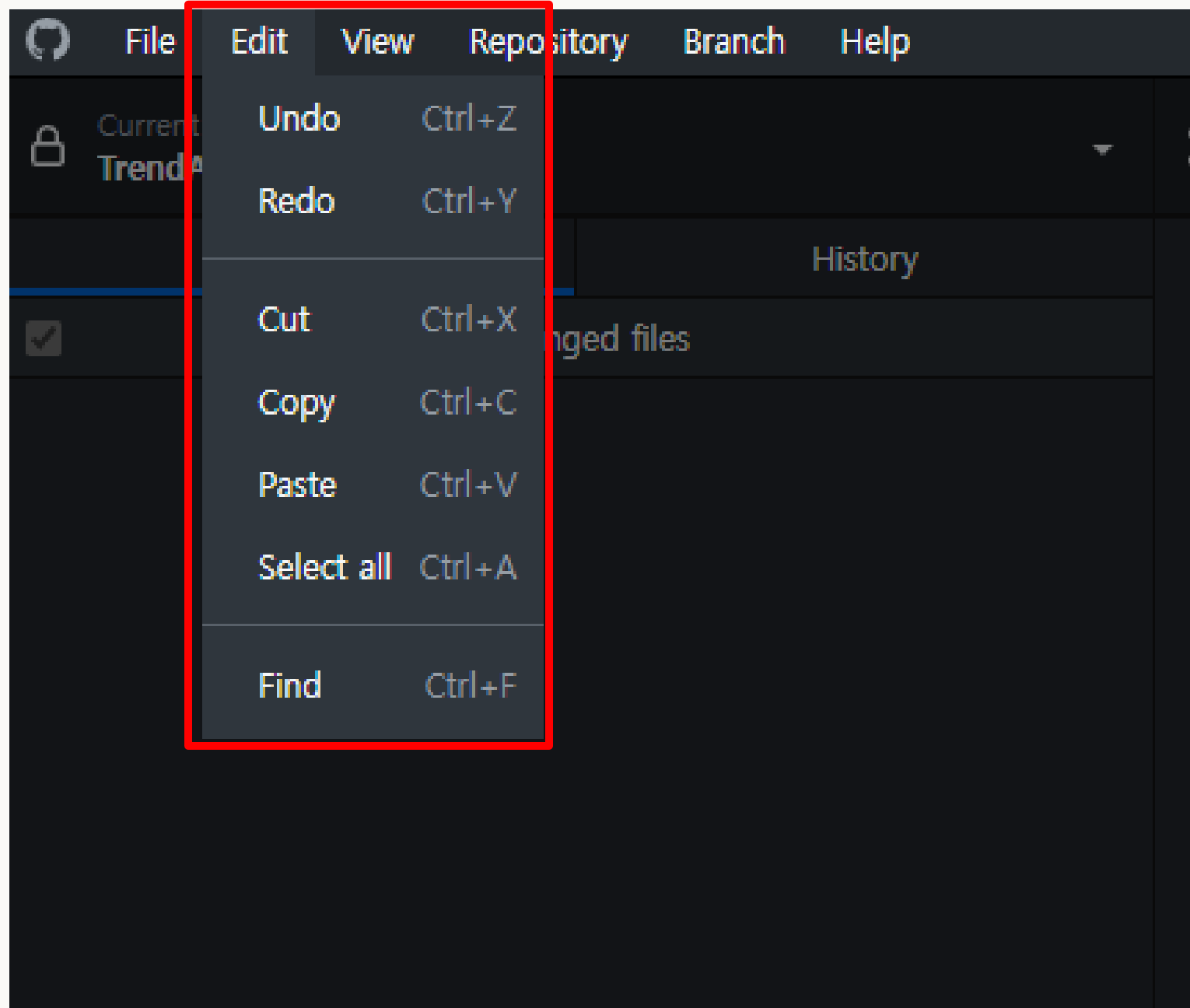
원격 저장소(GitHub 등)에 있는 전체 프로젝트를 내 컴퓨터로 복제하는 작업이다.

저장소의 파일, 폴더, 커밋 기록 등 모든 Git 이력을 그대로 로컬로 가져오는 것이다.

다른 사람이 만든 프로젝트를 내 환경에서 실행하거나 수정하려고 할 때 사용된다.

1. 복제할 프로젝트 선택
2. 복제할 프로젝트 경로 선택

# Edit 탭



**Undo : 되돌리기**

**Redo : 되돌리기 취소 (앞으로)**

**Cut : 잘라내기**

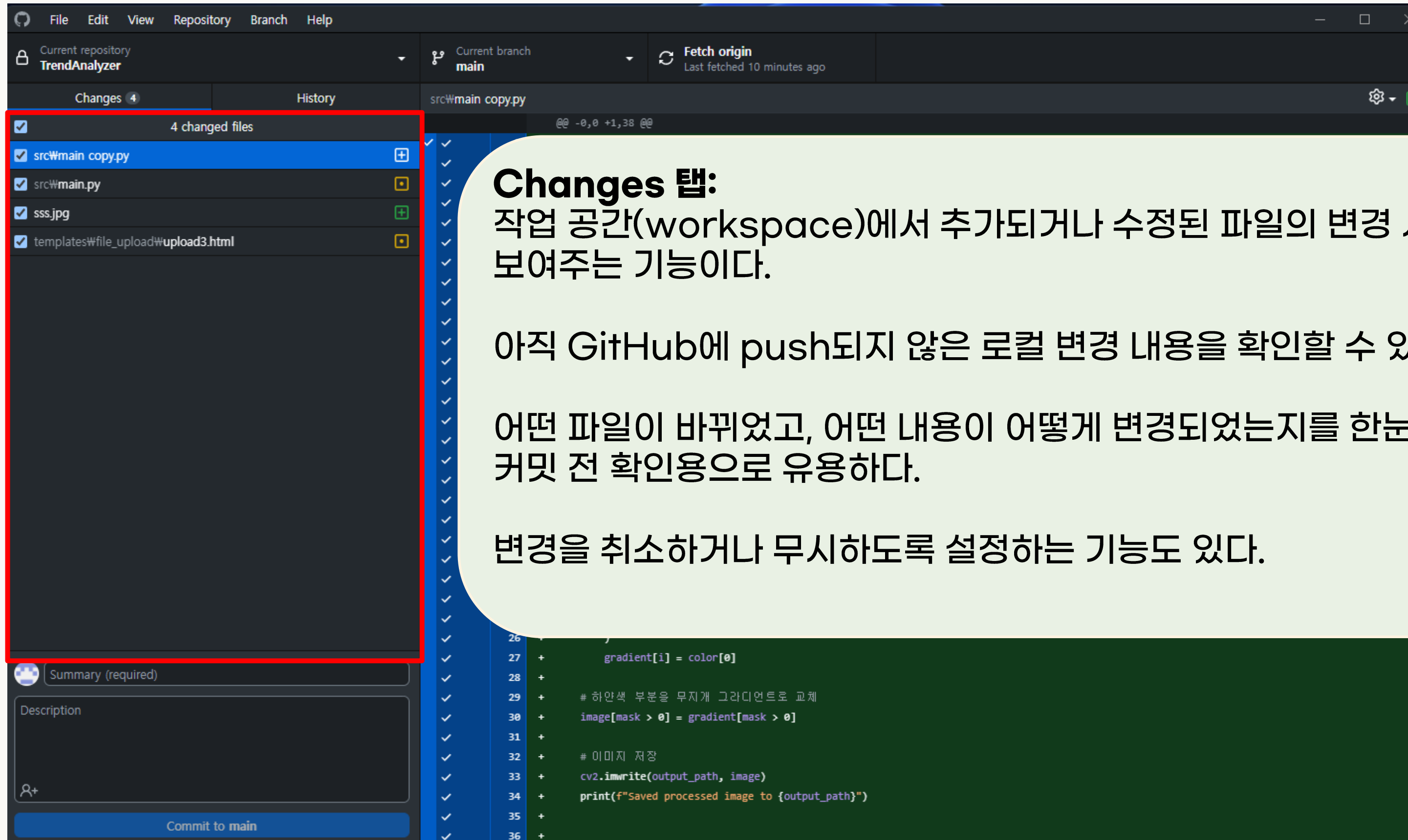
**Copy: 복사**

**Paste:붙여넣기**

**Select all: 전체 선택**

**Find : 찾기**

# Changes 탭



## Changes 탭:

작업 공간(workspace)에서 추가되거나 수정된 파일의 변경 사항을 사용자에게 보여주는 기능이다.

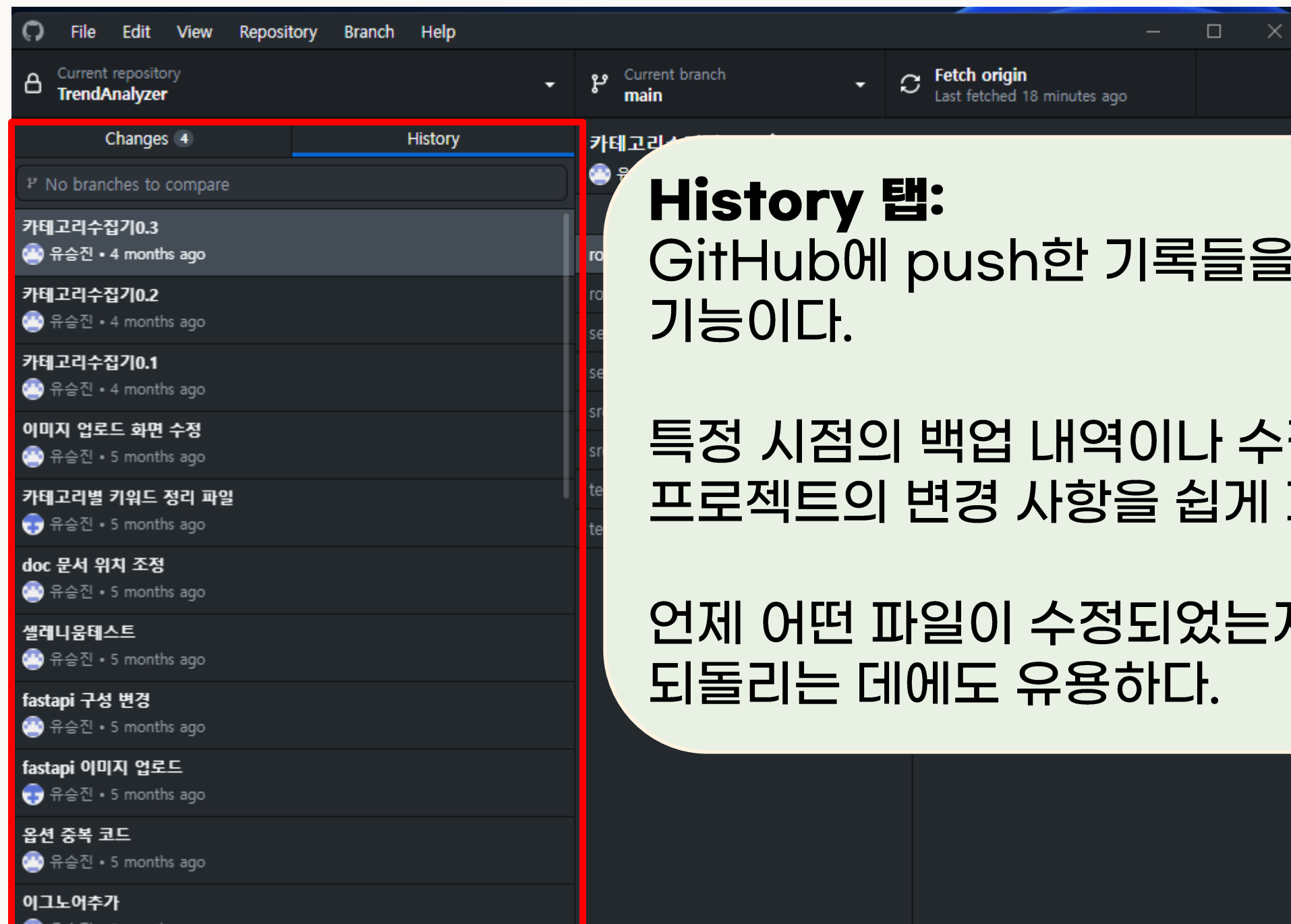
아직 GitHub에 push되지 않은 로컬 변경 내용을 확인할 수 있다.

어떤 파일이 바뀌었고, 어떤 내용이 어떻게 변경되었는지를 한눈에 파악할 수 있어 커밋 전 확인용으로 유용하다.

변경을 취소하거나 무시하도록 설정하는 기능도 있다.



# History 탭



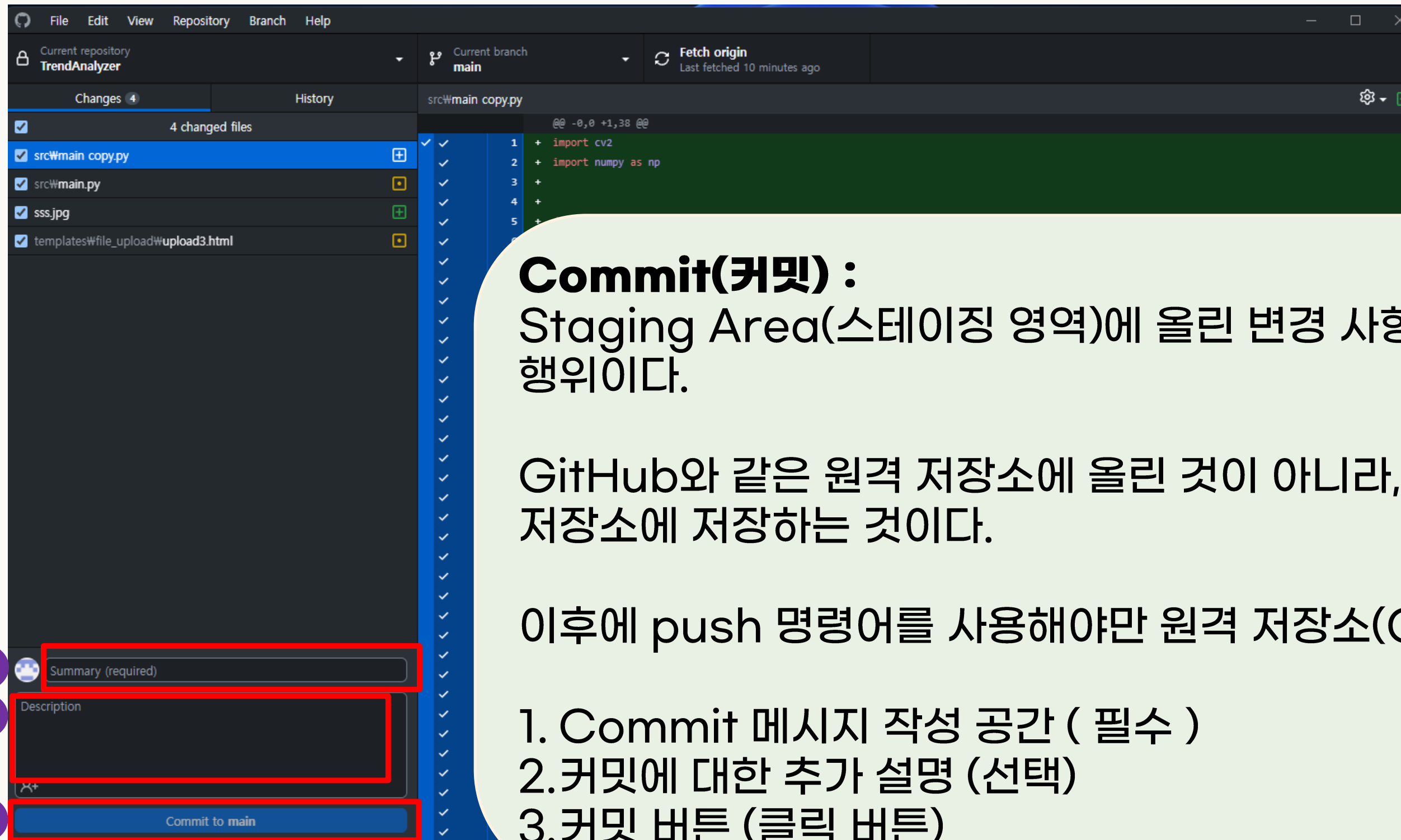
## History 탭:

GitHub에 push한 기록들을 시간 순으로 확인할 수 있는 기능이다.

특정 시점의 백업 내역이나 수정 기록을 살펴볼 수 있어 프로젝트의 변경 사항을 쉽게 파악할 수 있다.

언제 어떤 파일이 수정되었는지 알 수 있어 이전 버전으로 되돌리는 데에도 유용하다.

# Commit



## Commit(커밋) :

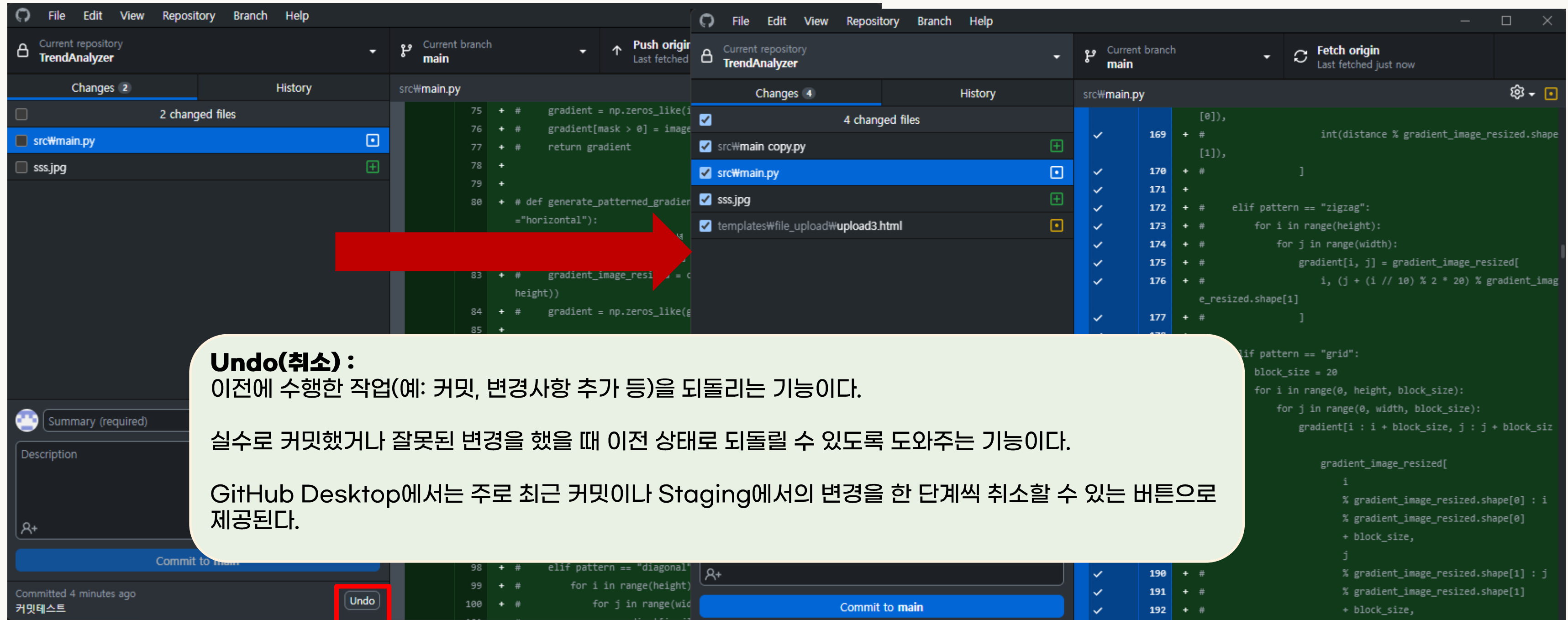
Staging Area(스테이징 영역)에 올린 변경 사항을 로컬 저장소에 저장하는 행위이다.

GitHub와 같은 원격 저장소에 올린 것이 아니라, 내 컴퓨터 안의 Git 로컬 저장소에 저장하는 것이다.

이후에 push 명령어를 사용해야만 원격 저장소(GitHub 등)에 업로드된다.

1. Commit 메시지 작성 공간 ( 필수 )
2. 커밋에 대한 추가 설명 (선택)
3. 커밋 버튼 (클릭 버튼)

# Undo (취소)



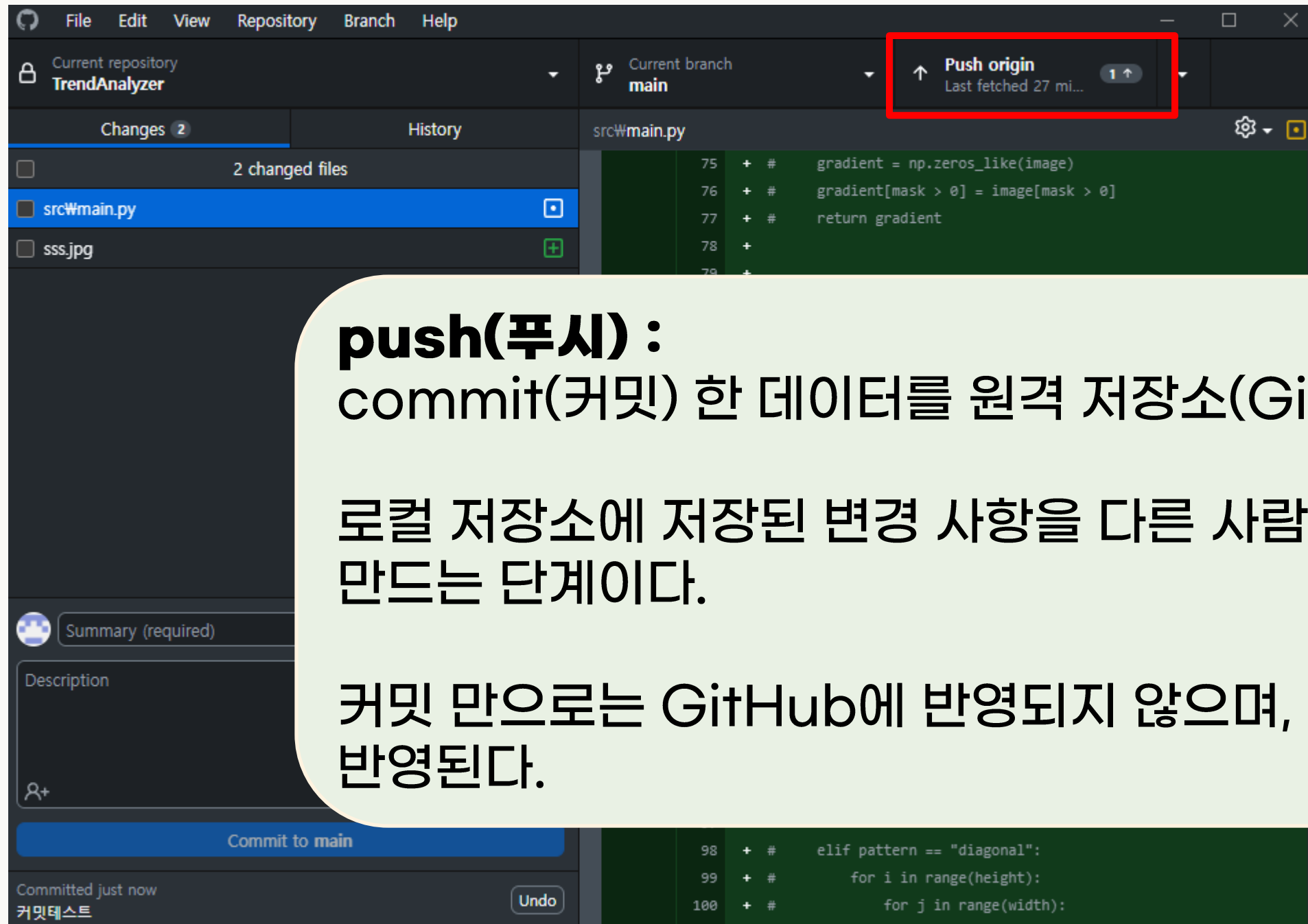
The screenshot displays the GitHub Desktop application interface. The top bar shows the repository name 'TrendAnalyzer' and the current branch 'main'. The left sidebar lists 'Changes' (2) and 'History'. The main area shows a diff view for 'src/main.py'. A red arrow points from the 'src/main.py' file in the 'Changes' list to the diff view. The bottom bar shows the commit summary area with a red box highlighting the 'Undo' button. The commit message is '커밋 테스트' and it was committed 4 minutes ago.

**Undo(취소) :**  
이전에 수행한 작업(예: 커밋, 변경사항 추가 등)을 되돌리는 기능이다.

실수로 커밋했거나 잘못된 변경을 했을 때 이전 상태로 되돌릴 수 있도록 도와주는 기능이다.

GitHub Desktop에서는 주로 최근 커밋이나 Staging에서의 변경을 한 단계씩 취소할 수 있는 버튼으로 제공된다.

# PUSH



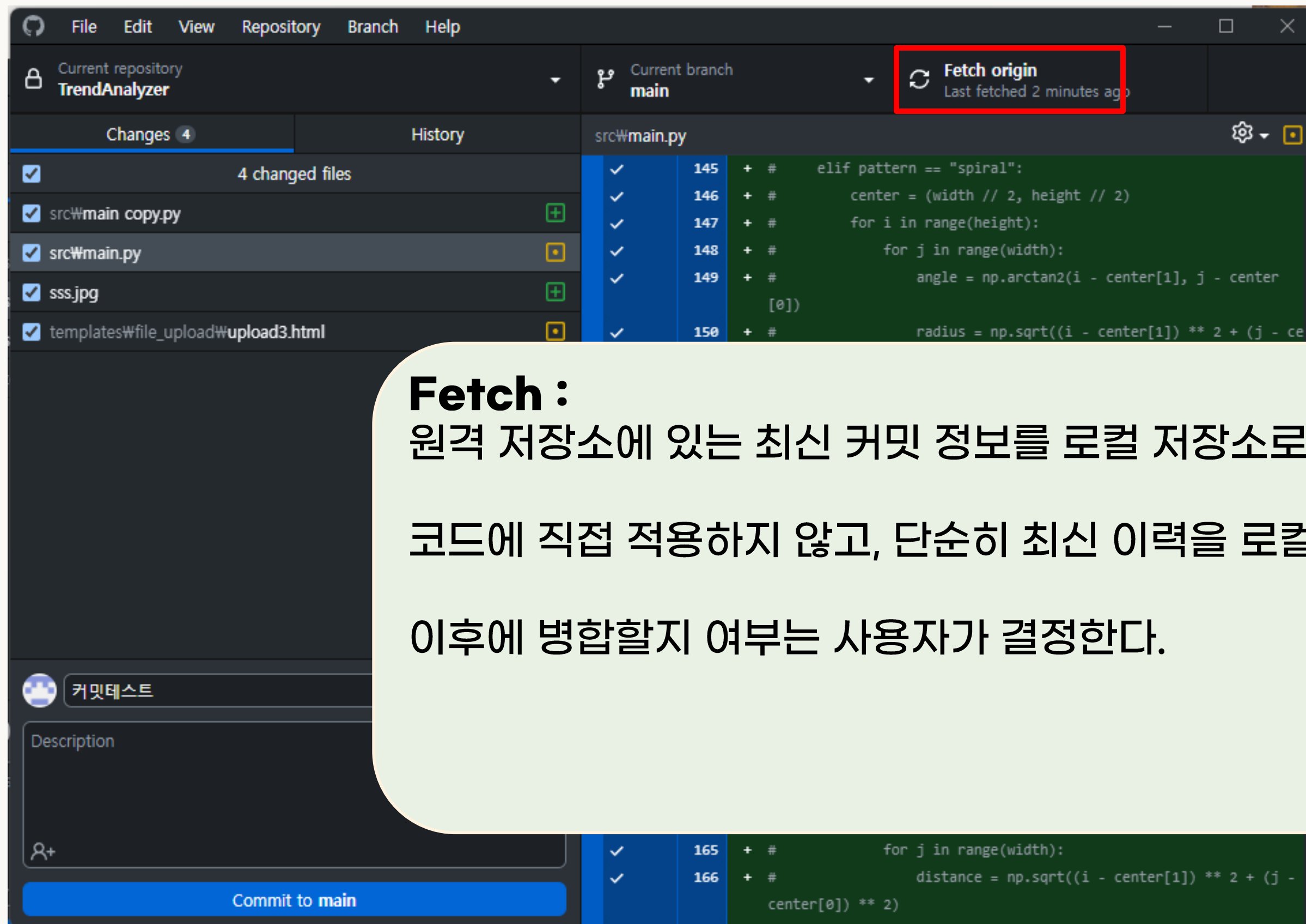
## push(푸시) :

commit(커밋) 한 데이터를 원격 저장소(GitHub 등)에 올리는 작업이다.

로컬 저장소에 저장된 변경 사항을 다른 사람들과 공유하거나 백업할 수 있게 만드는 단계이다.

커밋 만으로는 GitHub에 반영되지 않으며, push를 통해서만 원격 저장소에 반영된다.

# Fetch



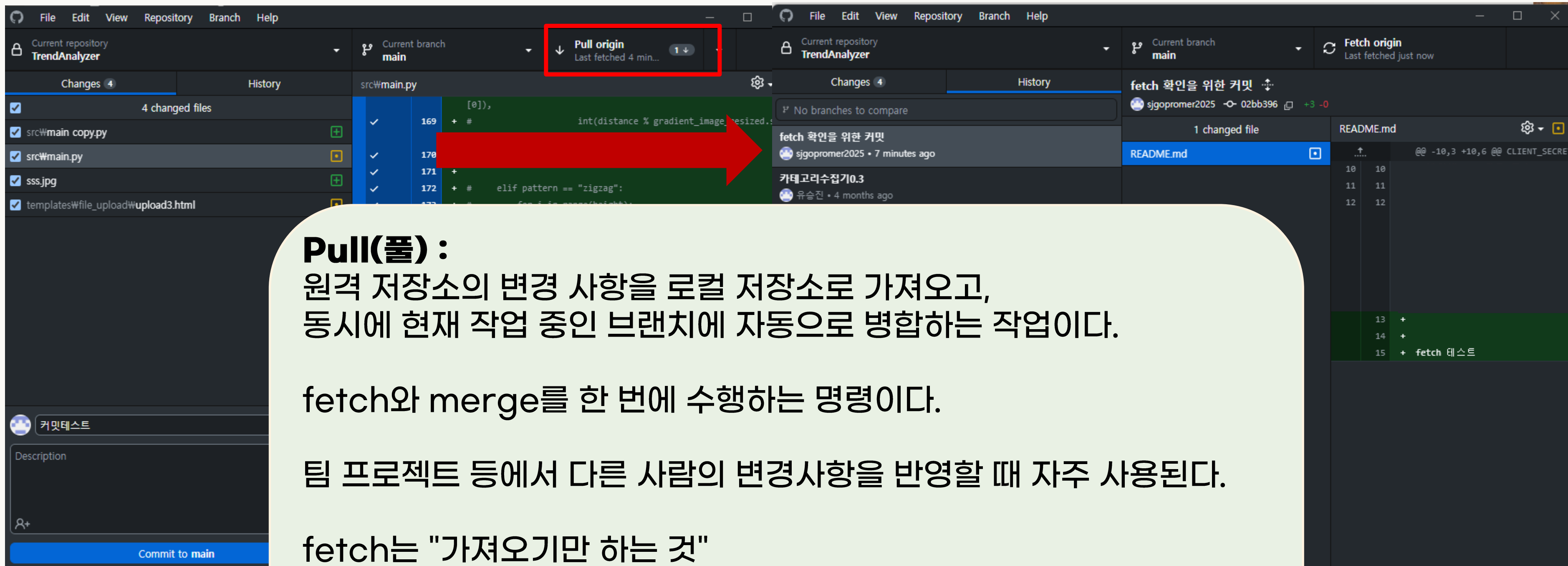
## Fetch :

원격 저장소에 있는 최신 커밋 정보를 로컬 저장소로 가져오는 작업이다.

코드에 직접 적용하지 않고, 단순히 최신 이력을 로컬에만 저장하는 것이다.

이후에 병합할지 여부는 사용자가 결정한다.

# Pull (풀)



## **Pull(풀) :**

원격 저장소의 변경 사항을 로컬 저장소로 가져오고,  
동시에 현재 작업 중인 브랜치에 자동으로 병합하는 작업이다.

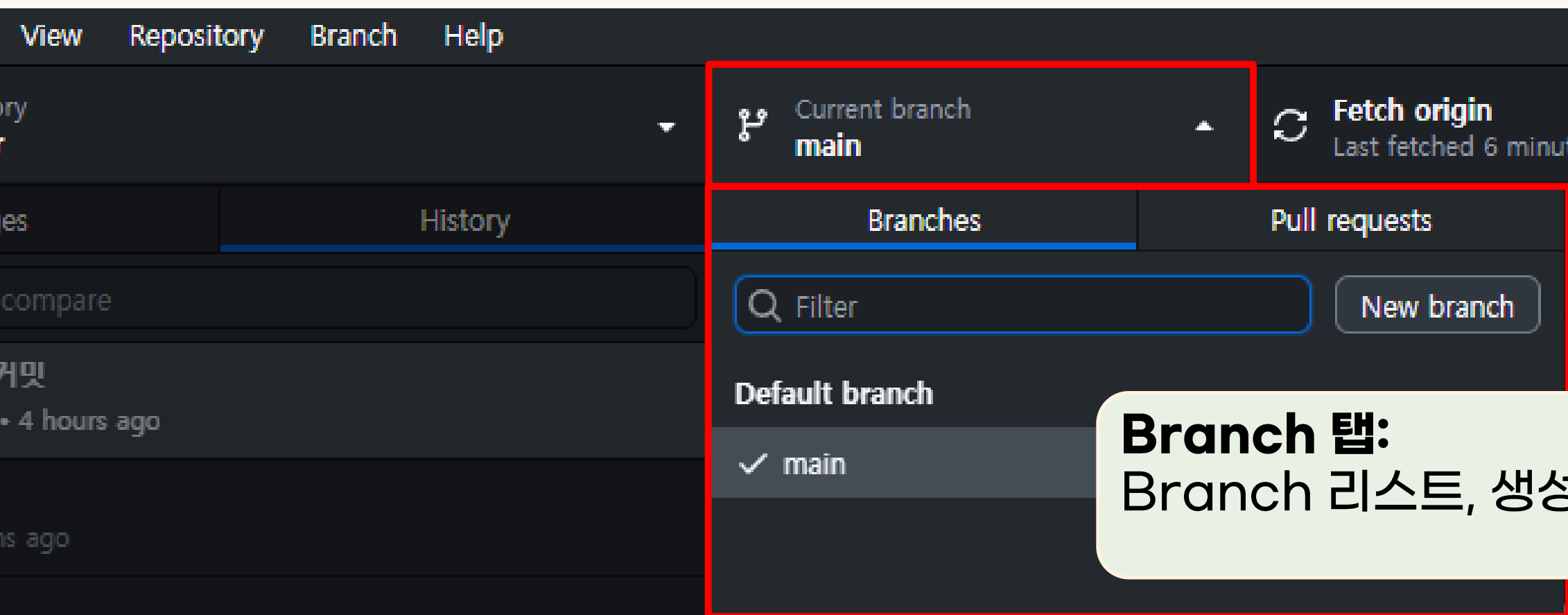
fetch와 merge를 한 번에 수행하는 명령이다.

팀 프로젝트 등에서 다른 사람의 변경사항을 반영할 때 자주 사용된다.

fetch는 "가져오기만 하는 것"

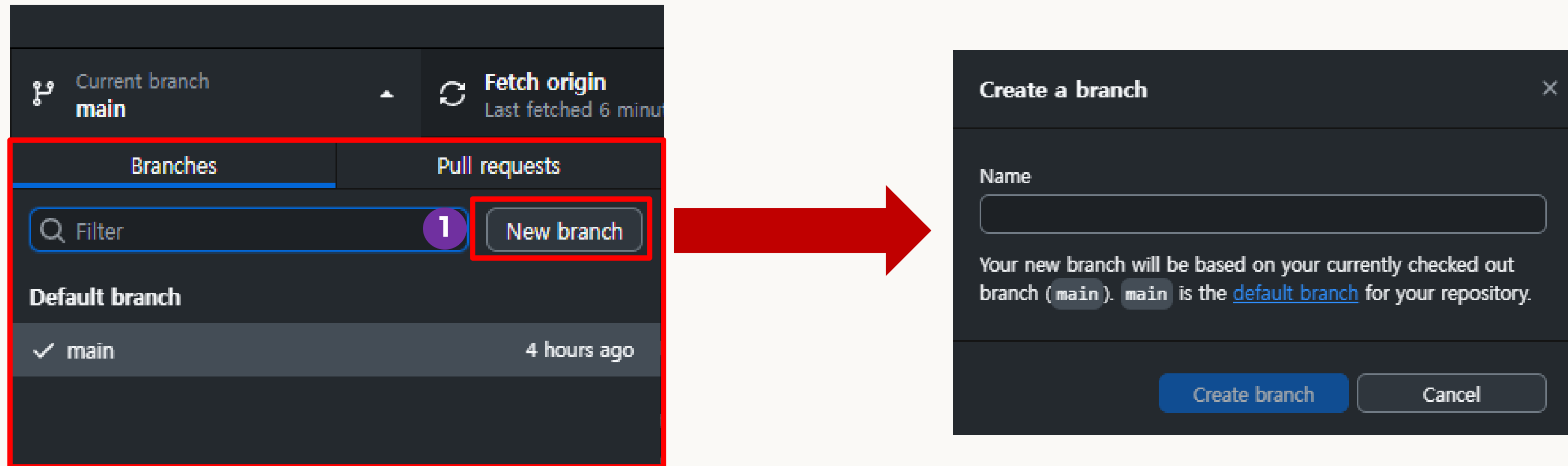
pull은 "가져오고, 내 코드에 적용까지 하는 것"

# Branch 생성 (1)



**Branch 탭:**  
Branch 리스트, 생성, 삭제 등을 할 수 있는 기능을 가진 영역

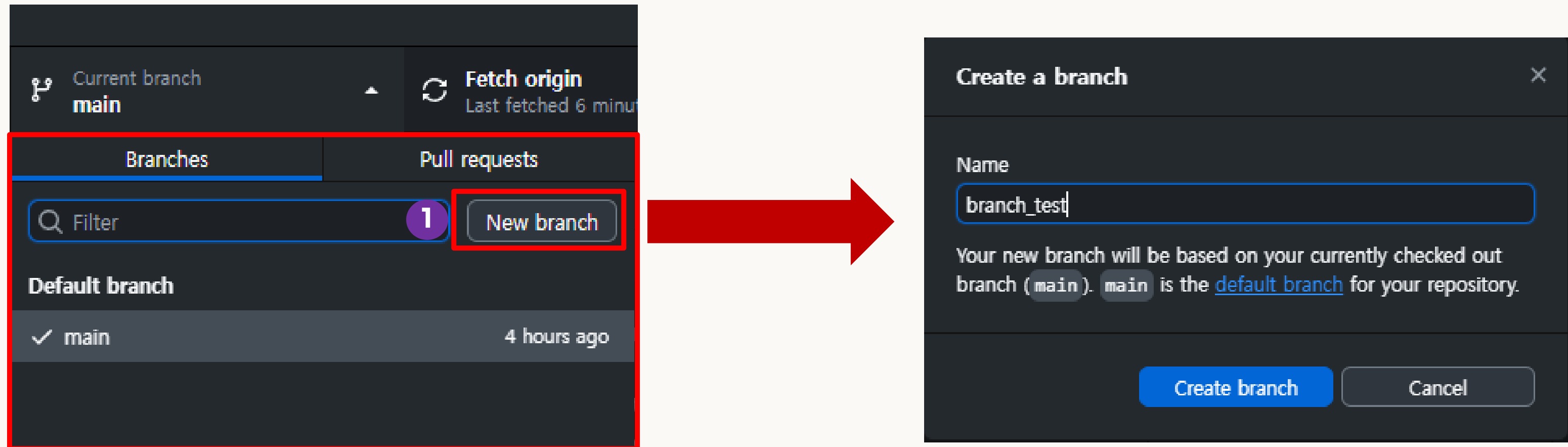
# Branch 생성 (2)



New Branch 버튼 클릭 시 branch 생성 창 표시  
생성하는 branch 현재 사용하고 있는 branch를 기준으로  
복제된다.

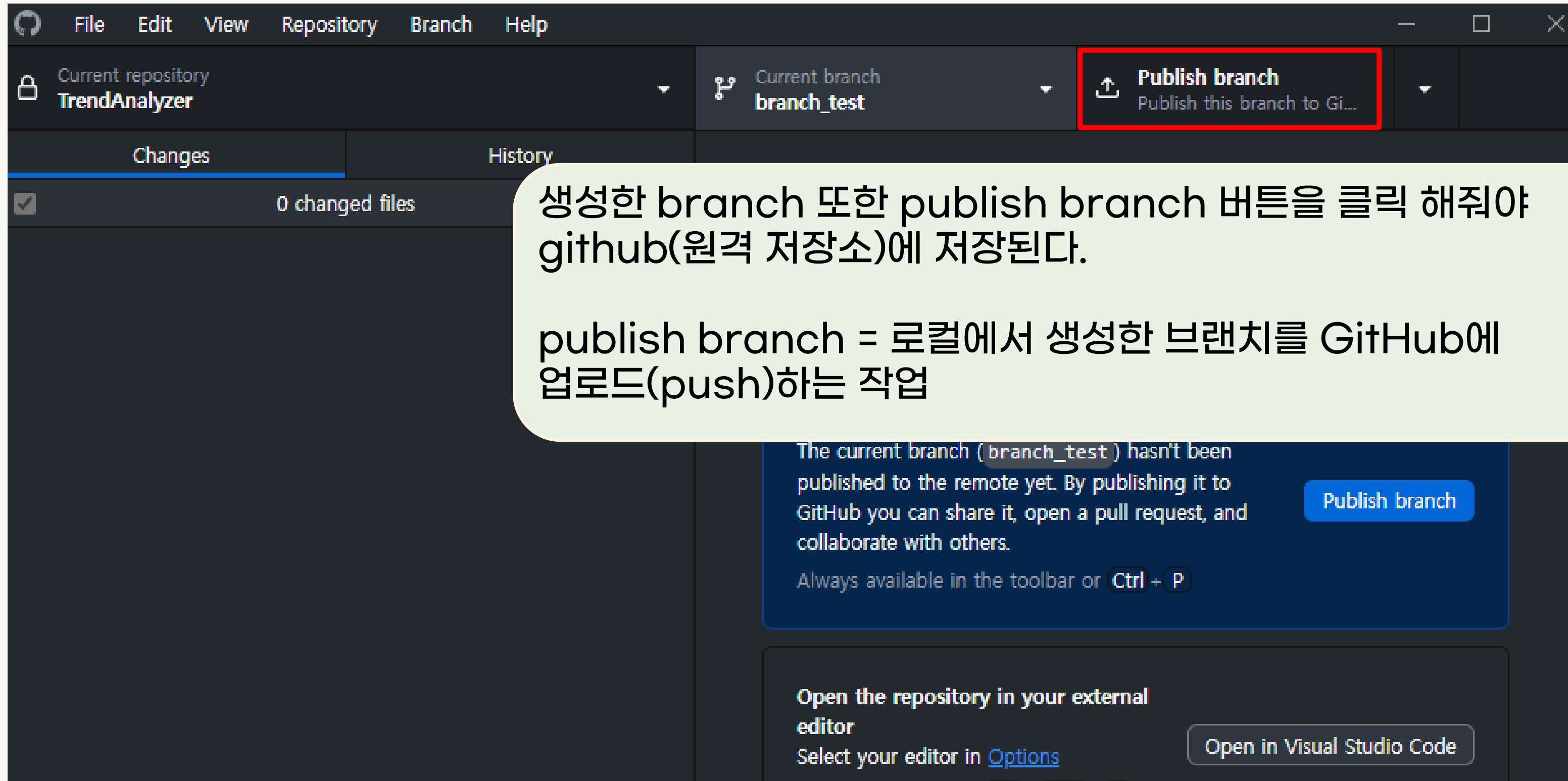


# Branch 생성 (3)

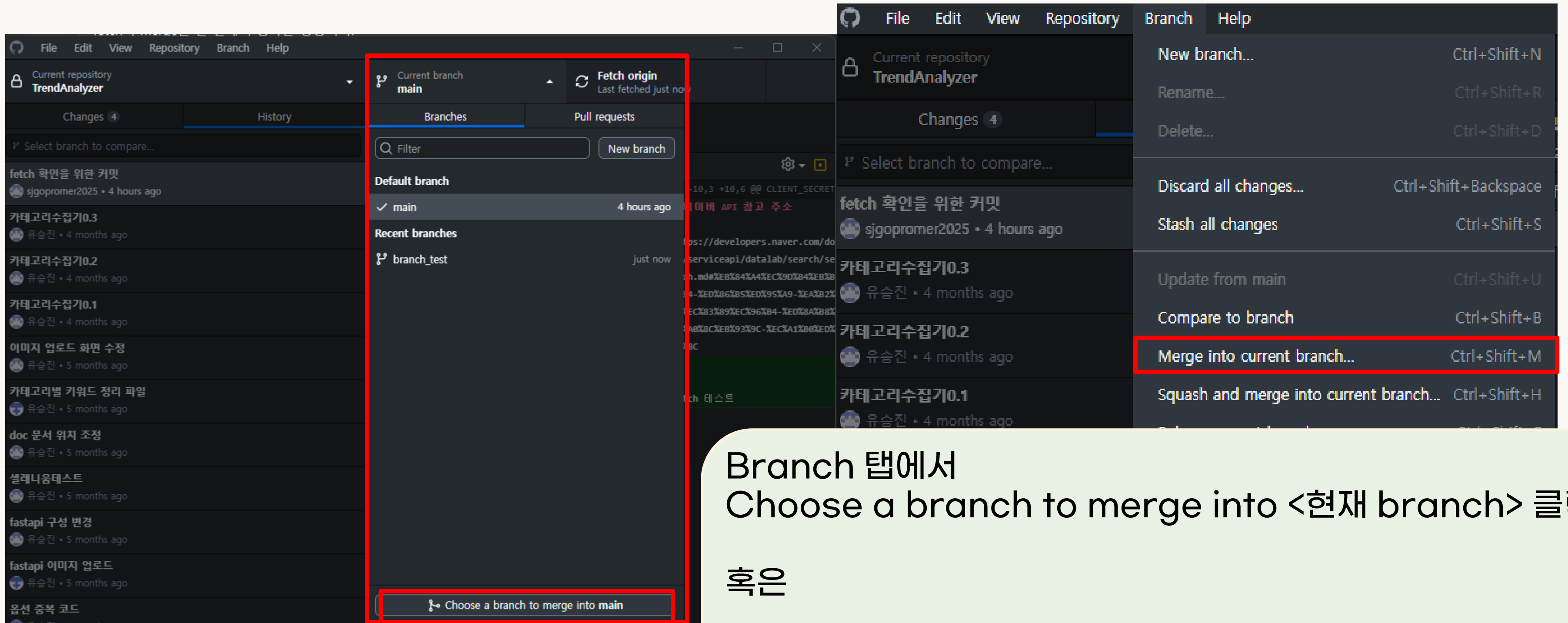


New Branch 버튼 클릭 시 branch 생성 창 표시  
생성하는 branch 현재 사용하고 있는 branch를 기준으로  
복제된다.

# Branch 생성 (4)



# Merge 합병 (1)

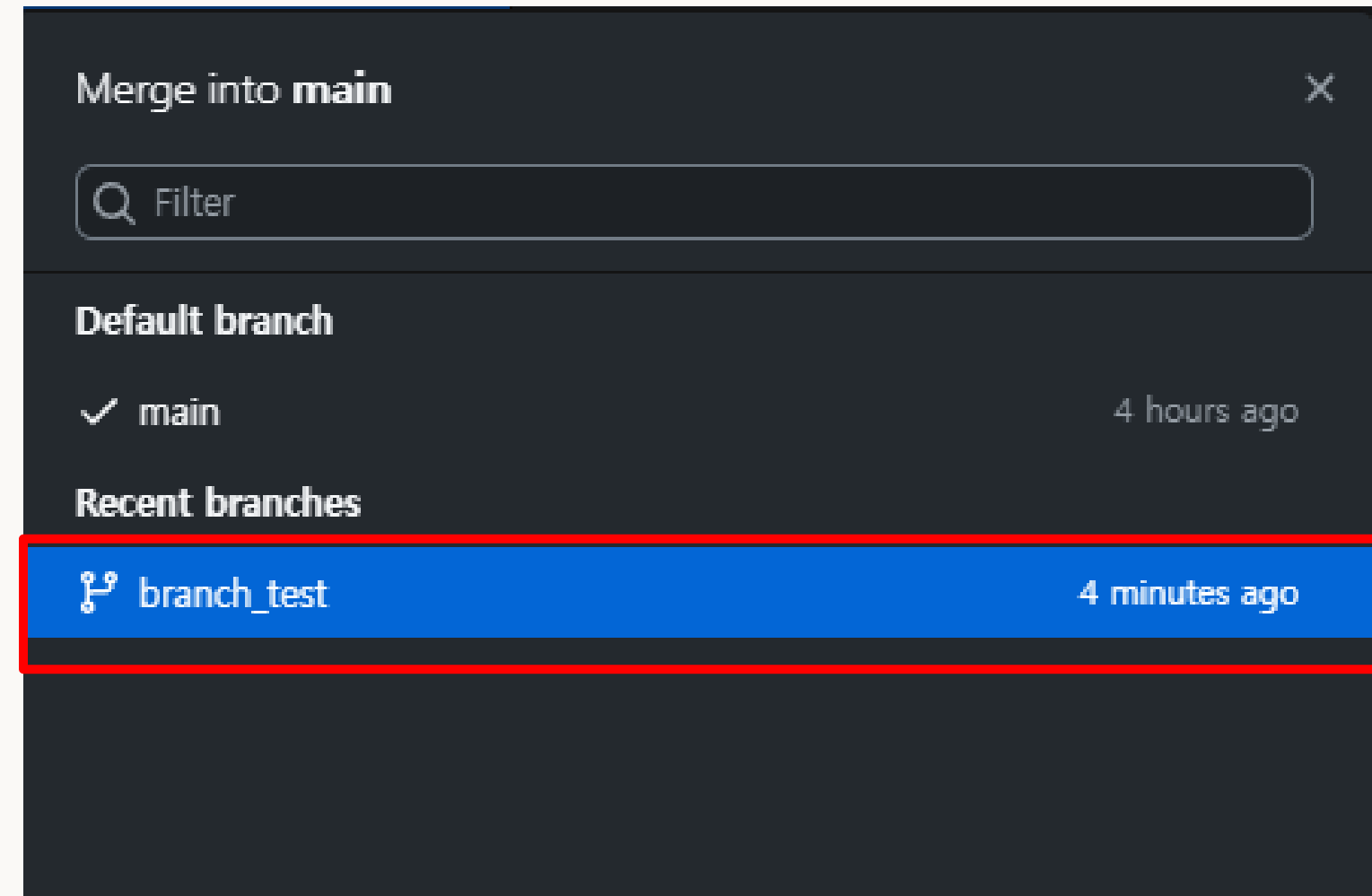


Branch 탭에서  
Choose a branch to merge into <현재 branch> 클릭

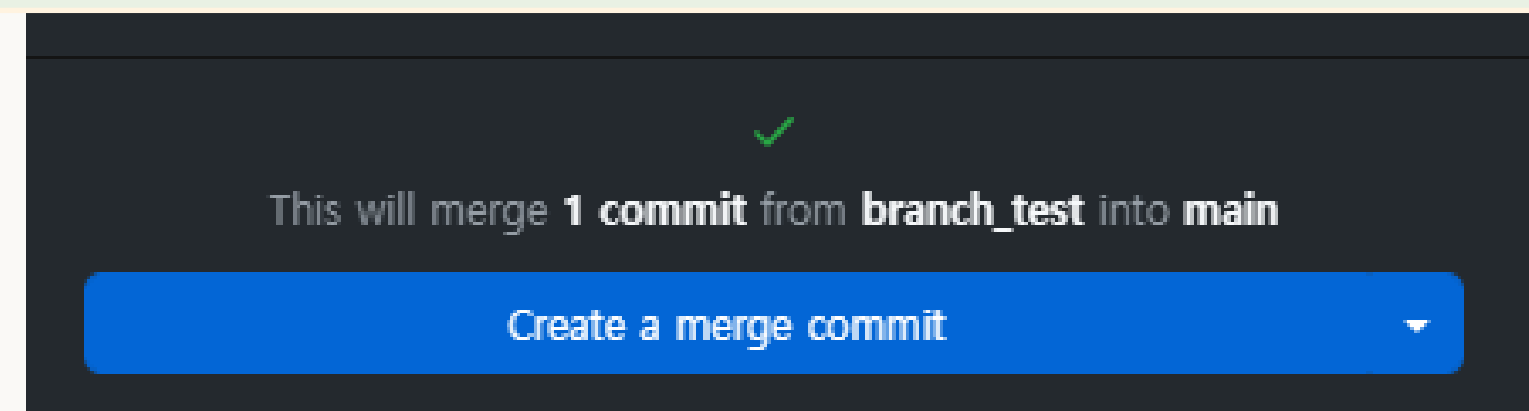
혹은

상단 Branch 탭에서  
merge into current branch (현재 브랜치로 합병)

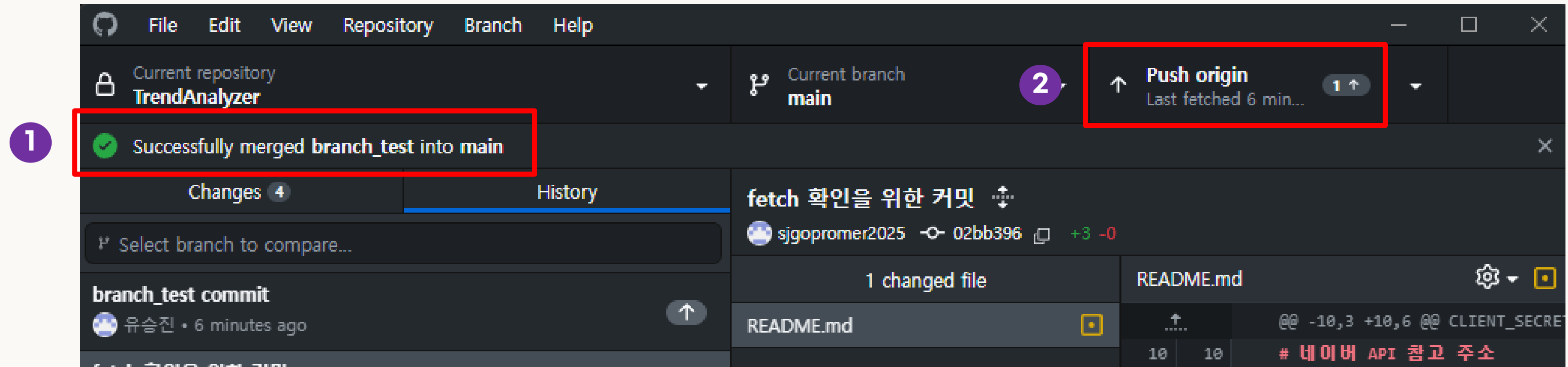
# Merge 합병 (2)



병합할 branch 선택 후 create a merge commit 버튼 클릭



# Merge 합병 (3)



1. [현재 브랜치 명]에 성공적으로 merge(합병) 이 되었으며 commit 까지 같이 진행된다.
2. 변경된 사항을 github(원격저장소)에 push 반영한다.

**고생하셨습니다!**

