

GitHub

GitHub

❖ 개요

- ✓ Ruby on Rails로 작성된 분산 버전 관리 툴인 git의 원격 저장소 호스팅 서비스
- ✓ 영리적인 서비스와 오픈 소스를 위한 무상 서비스를 모두 제공
- ✓ git이 텍스트 명령어 입력 방식인데 반해 git hub는 그래픽 유저 인터페이스(GUI)를 제공
- ✓ git hub는 pastebin(텍스트 데이터를 저장하고 공유할 수 있는 온라인 저장소) 과 유사한 서비스인 Gist 와 Wiki를 Repository 마다 운영하고 있으며 git 저장소를 통해 고칠 수 있음
- ✓ <https://GitHub.com> 에 접속하여 계정 생성 및 로그인 후 작업 가능
- ✓ git hub는 깃 프로젝트 저장소 역할 외에도 다양한 기능을 제공하는데 GitHub Action 과 git hub Deployment API를 이용하면 빌드 및 배포 자동화를 구성할 수 있고 Project Boards를 이용해 프로젝트 협업 가능
- ✓ Public 저장소는 무료로 생성할 수 있고 Private 저장소는 작업자 3인 이하인 경우에는 무료 이고 설치형 버전인 Enterprise는 유료
- ✓ 가장 인기있는 git 저장소 호스팅 서비스

GitHub

❖ 현재 컴퓨터와 GitHub 계정 연결

✓ Personal Access Token 생성

- GitHub 로그인
- 우측 상단 프로필 → Settings
- Developer settings
- Personal access tokens → Tokens (classic) 또는 Fine-grained token
- Generate new token
- 권한은 최소한: repo 체크
- 생성 후 토큰 복사 (한 번만 보여줌)

✓ 토큰 등록

- Windows: 두가지 명령 중 하나만 수행
 - ◆ `git config --global credential.helper manager`
 - ◆ `git config --global credential.helper manager-core`
- Mac: `git config --global credential.helper osxkeychain`

명령어로 원격 저장소 사용

❖ 작업

- ✓ clone: 원격 저장소를 로컬 저장소로 복제
- ✓ fork: 원본 저장소(Upstream)를 내 계정으로 그대로 복사해서 내 마음대로 수정할 수 있는 독립 저장소를 생성
- ✓ remote: 원격 저장소를 추가하고 조회하고 삭제
- ✓ push: 원격 저장소에 밀어넣기
- ✓ fetch: 원격 저장소를 일단 가져만 오기
- ✓ pull: 원격 저장소를 가져와서 합치기

명령어로 원격 저장소 사용

❖ 원격 저장소 복제 – git clone

- ✓ 원격 저장소를 클론하는 명령은 `git clone <원격 저장소>`
- ✓ `https://GitHub.com/tensorflow/tensorflow.git` 에서 클론
 - `git clone https://GitHub.com/tensorflow/tensorflow.git`
- ✓ 특정 경로에 받고자 하는 경우에는 뒤에 다운로드 받을 경로를 추가
 - 문자열 형태로 설정해야 하고 빈 디렉토리 이어야 함
 - 현재 디렉토리에 `localtensorflow` 디렉토리를 생성
 - `git clone https://GitHub.com/tensorflow/tensorflow.git ./localtensorflow`

명령어로 원격 저장소 사용

❖ 원격 저장소 생성

- ✓ GitHub 사이트에서 New를 클릭한 후 작성
 - Owner는 원격 저장소의 소유자이고 Repository name은 원격 저장소의 이름
 - Description은 원격 저장소에 대한 설명을 적는 공간인데 생략 가능
 - 아래에 Public과 Private 중 하나를 선택할 수 있는 곳이 있는데 Public은 모두에게 공개된 저장소를 의미하고 Private은 모두에게 공개하지 않고 소유자와 소유자가 지정한 지정한 일부 사용자만 볼 수 있는 저장소를 의미
 - Add a README file은 원격 저장소를 생성할 적에 자동으로 README 파일을 생성해주는 항목
 - Add .gitignore를 선택하면 원격 저장소를 생성할 때 자동으로 .gitignore를 생성
 - Choose a license는 이 저장소에 담길 프로젝트의 라이선스를 선택하는 항목
- ✓ 생성된 원격 저장소의 URL은 <https://github.com/계정 이름/저장소 이름>
- ✓ 생성된 원격 저장소는 Settings 메뉴의 최하단에 있는 Delete this repository 를 클릭해서 삭제 가능

명령어로 원격 저장소 사용

❖ 원격 저장소 생성

- ✓ GitHub 사이트에서 New를 클릭한 후 작성

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).
Required fields are marked with an asterisk (*).

1 **General**

Owner *  itgangpae / **Repository name *** test-repo
✔ test-repo is available.

Great repository names are short and memorable. How about [cautious-octo-spork?](#)

Description
My First GitHub Repository
26 / 350 characters

2 **Configuration**

Choose visibility *
Choose who can see and commit to this repository Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#) Off

Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#) No .gitignore

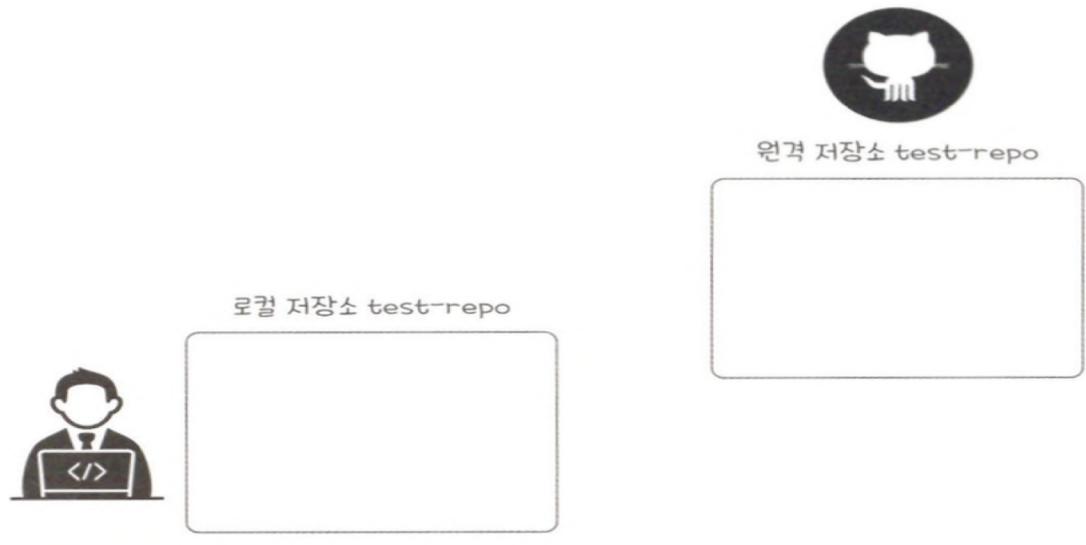
Add license
Licenses explain how others can use your code. [About licenses](#) No license

Create repository

명령어로 원격 저장소 사용

❖ git remote

- ✓ 원격 저장소를 추가하고 조회하고 삭제할 수 있는 명령어
- ✓ 로컬 저장소 생성
 - 컴퓨터에 test-repo 라는 디렉토리를 만들고 git init 명령 수행
- ✓ 현재 상황



명령어로 원격 저장소 사용

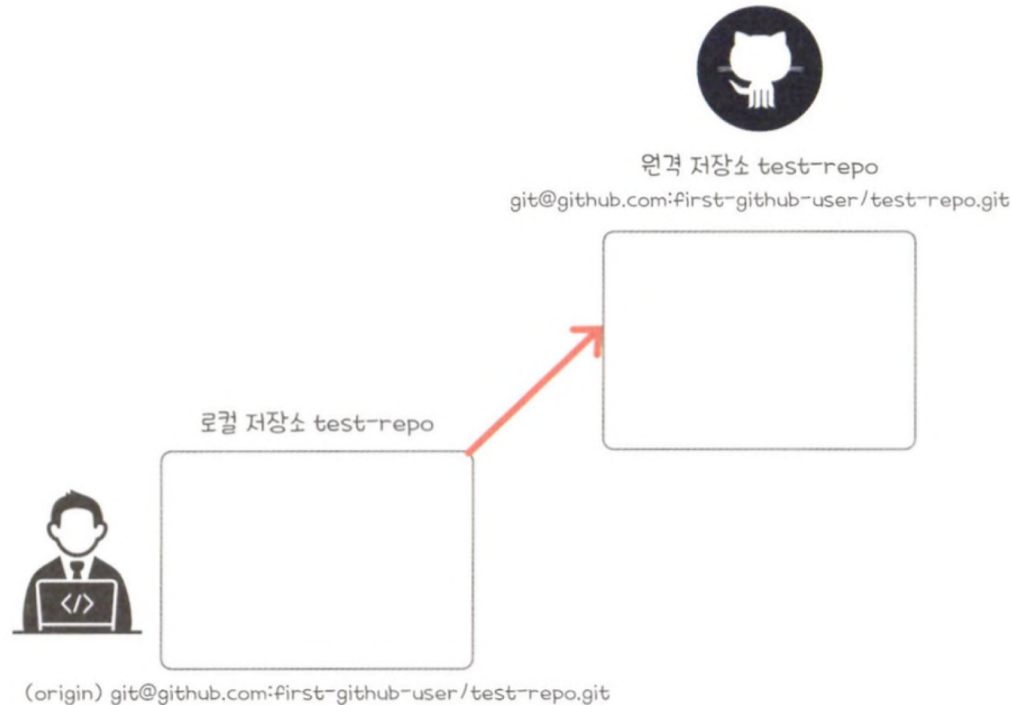
❖ git remote

✓ 로컬 저장소에 원격 저장소를 추가

- 명령: git remote add <원격 저장소 이름> <원격 저장소 경로>
- 원격 저장소 이름: <https://GitHub.com/유저이름/저장소이름>
- 이전에 만든 원격 저장소를 현재 로컬에 추가: git remote add origin <https://gitHub.com/itstudy001/test-repo.git>
- 원격 저장소 이름 확인: git remote
origin
- 원격 저장소 이름 과 경로 확인: git remote -v 또는 git remote --verbose
origin <https://gitHub.com/itstudy001/test-repo.git> (fetch)
origin <https://gitHub.com/itstudy001/test-repo.git> (push)

명령어로 원격 저장소 사용

- ❖ git remote
 - ✓ 현재까지 상황



명령어로 원격 저장소 사용

❖ git remote

- ✓ 원격 저장소 이름 변경: `git remote rename <기존 이름> <새로운 이름>`
 - origin 이라는 이름을 changed 로 변경
 - ◆ 변경: `git remote rename origin changed`
 - ◆ 확인: `git remote`
`changed`
 - ◆ 확인 `git remote -v`
`changed https://gitHub.com/itstudy001/test-repo.git (fetch)`
`changed https://gitHub.com/itstudy001/test-repo.git (push)`
- ✓ 원격 저장소 삭제: `git remote remove <원격 저장소 이름>`
 - `git remote remove changed`
 - `git remote`
 - `git remote -v`

명령어로 원격 저장소 사용

❖ git push

- ✓ 로컬 저장소의 내용을 원격 저장소에 업로드 하는 작업
- ✓ 로컬 저장소에 A 가 적힌 a.txt 파일을 생성
- ✓ stage에 추가: git add a.txt
- ✓ commit: git commit -m "first commit"

[main (root-commit) 3fa4be0] first commit

1 file changed, 1 insertion(+)

create mode 100644 a.txt

- ✓ 현재 상황



명령어로 원격 저장소 사용

❖ git push

✓ 업로드

- Remote Repository 연결: `git remote add origin https://github.com/itstudy001/test-repo.git`
- 현재 branch 이름 수정: `git branch -M main`
- 푸시: `git push -u origin main`

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 203 bytes | 203.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://GitHub.com/itstudy001/test-repo.git

* [new branch] main -> main

branch 'main' set up to track 'origin/main'.

...or push an existing repository from the command line

```
git remote add origin https://github.com/itstudy001/test-repo.git
git branch -M main
git push -u origin main
```



명령어로 원격 저장소 사용

❖ git push

✓ 업로드

- git remote add origin <https://GitHub.com/itstudy001/test-repo.git>
 - ◆ 원격 저장소 <https://GitHub.com/itstudy001/test-repo.git> 을 origin이라는 이름으로 추가하는 명령
- git branch -M main
 - ◆ git branch -M <branch 이름>은 현재 branch 이름을 <branch 이름>으로 바꾸는 명령
 - ◆ 이 명령은 현재 branch(master) 이름을 main 으로 변경하는 명령
 - ◆ 로컬 저장소의 기본 branch는 master이지만 Git Hub의 기본 branch는 main이기 때문에 로컬 저장소의 기본 branch(master)에서 만든 변경 사항을 Git Hub의 기본 branch(main) 로 push하기 위해서는 이와 같이 branch 이름을 main으로 변경하는 것이 편리

명령어로 원격 저장소 사용

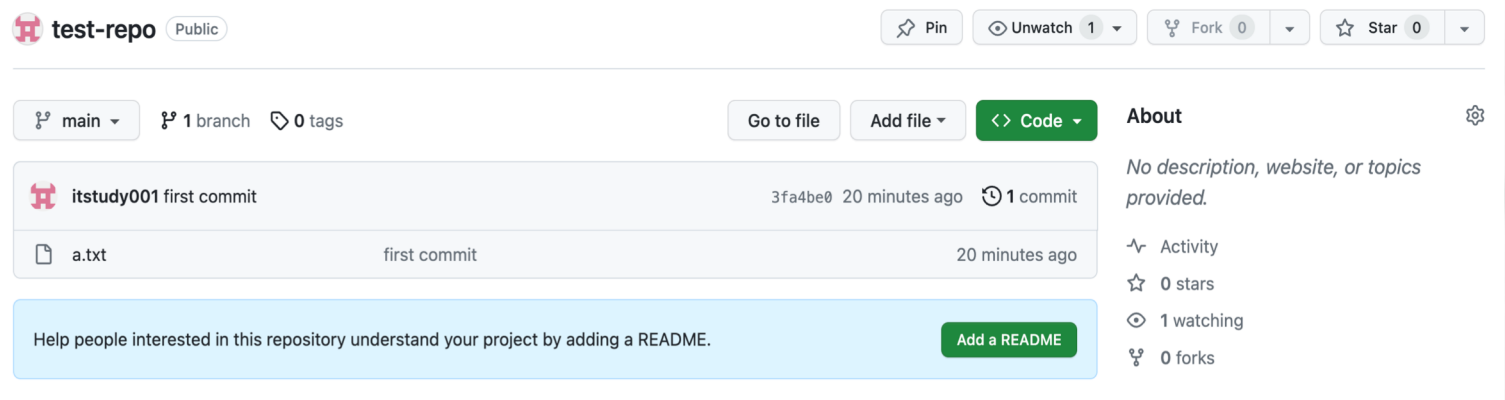
❖ git push

✓ 업로드

● git push -u origin main

- ◆ git push <원격 저장소 이름> <branch 이름>은 원격 저장소 이름으로 branch 이름을 push하는 명령
- ◆ git push origin main은 원격 저장소 origin으로 로컬 저장소 main branch의 변경 사항을 push하는 명령
- ◆ -u 옵션은 처음 push할 때 한 번만 사용하면 되는데 이 옵션과 함께 push하면 추후 간단히 git push 또는 git pull 명령 만으로 origin의 main branch로 푸시 또는 풀을 할 수 있음

● 원격 저장소 확인



명령어로 원격 저장소 사용

❖ git push

✓ 업로드

- a.txt 파일에 문자 B를 추가한 뒤 저장
- 스테이지에 추가: `git add a.txt`
- Commit: `git commit -m "second commit"`
[main e587e15] second commit
1 file changed, 2 insertions(+), 1 deletion(-)
- 푸시: `git push`
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 236 bytes | 236.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To <https://GitHub.com/itstudy001/test-repo.git>
3fa4be0..e587e15 main -> main

명령어로 원격 저장소 사용

❖ git fetch

- ✓ 원격 저장소의 변경 사항을 로컬 저장소에 병합하지 않고 일단 가져만 오는 방법
- ✓ 로컬 저장소의 현재 상태

● git log

commit e587e15e2eb65647e6138d49a8709dc30d5ffa42 (HEAD -> main, origin/main)

Author: itstudy001 <itstudy@kakao.com>

Date: Mon Oct 16 19:05:25 2023 +0900

second commit

commit 3fa4be0c4f8d042ceb5121ef4b208e609537b0b1

Author: itstudy001 <itstudy@kakao.com>

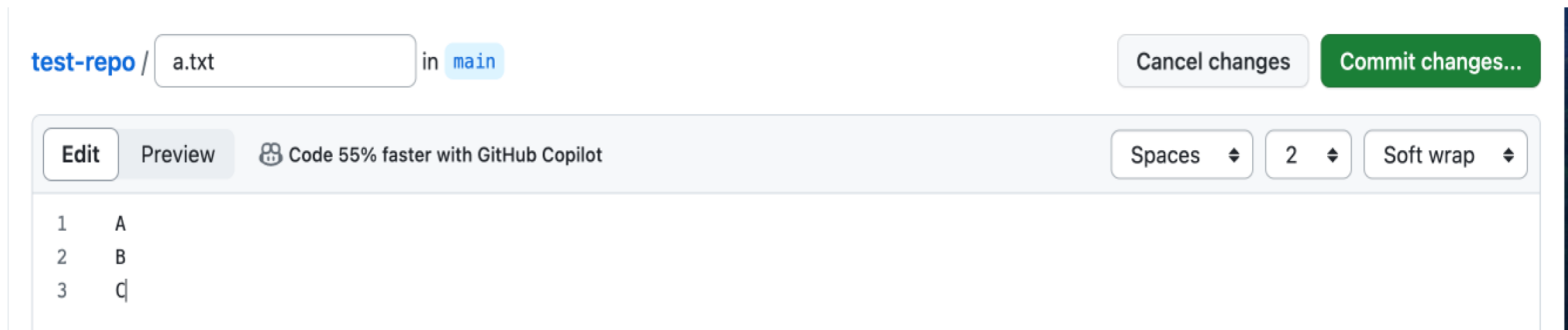
Date: Mon Oct 16 18:43:15 2023 +0900

first commit

명령어로 원격 저장소 사용

❖ git fetch

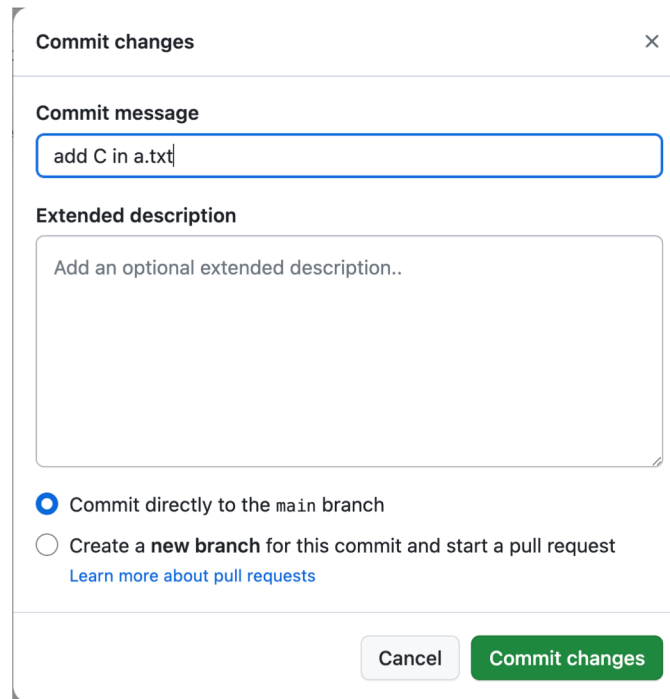
- ✓ Git Hub에서 a.txt 파일을 선택한 후 오른쪽의 연필 아이콘(편집)을 클릭하고 C를 추가한 후 화면에서 Commit Changes를 클릭



명령어로 원격 저장소 사용

❖ git fetch

- ✓ Git Hub에서 a.txt 파일을 선택한 후 오른쪽의 연필 아이콘(편집)을 클릭하고 C를 추가한 후 화면에서 Commit Changes를 클릭



Commit changes

Commit message

add C in a.txt

Extended description

Add an optional extended description..

☒ Commit directly to the main branch

☐ Create a new branch for this commit and start a pull request

[Learn more about pull requests](#)

Cancel Commit changes

명령어로 원격 저장소 사용

- ❖ git fetch
 - ✓ 현재 상황



명령어로 원격 저장소 사용

❖ git fetch

✓ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 619 bytes | 206.00 KiB/s, done.

From <https://GitHub.com/itstudy001/test-repo>

e587e15..0f8b9c3 main -> origin/main

✓ git status: 현재 1개의 commit 이 뒤짐

On branch main

Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.

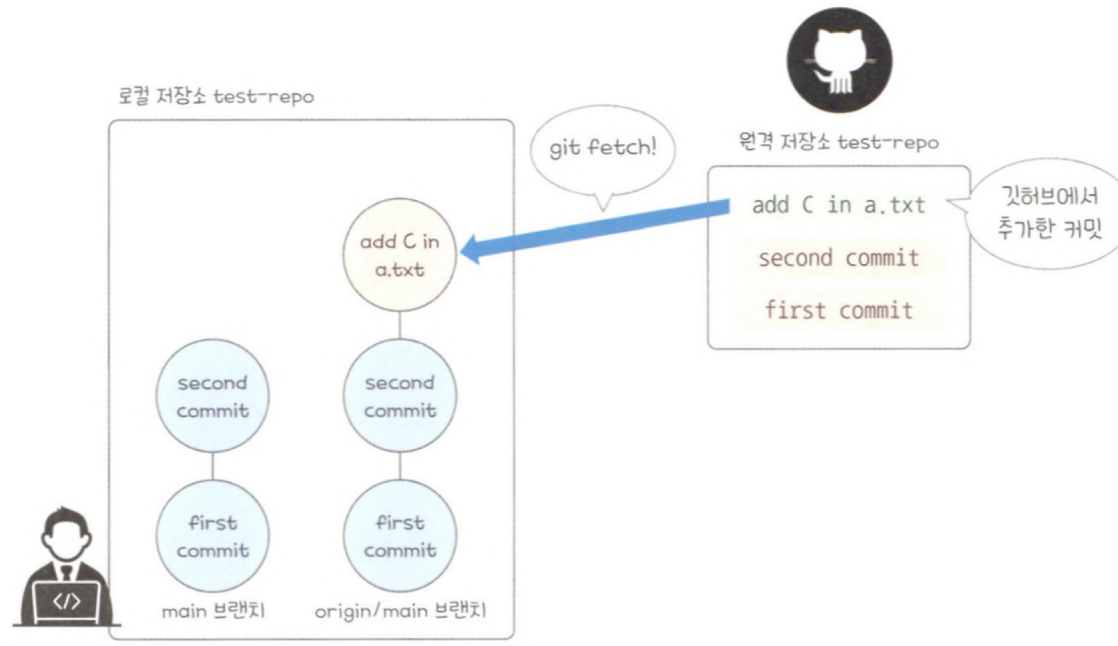
(use "git pull" to update your local branch)

nothing to commit, working tree clean

명령어로 원격 저장소 사용

❖ git fetch

- ✓ 패치 직후 main branch가 origin/main branch에 비해 Commit 하나가 뒤쳐지는 이유는 패치는 원격 저장소의 변경 사항을 origin/main branch로 가져올 뿐 main branch는 변함이 없기 때문
- ✓ 패치는 원격 저장소의 변경 사항을 일단 가지고 올 뿐 로컬 저장소의 branch로 병합하지는 않음



명령어로 원격 저장소 사용

❖ git fetch

- ✓ branch 비교 - git diff main origin/main

```
diff --git a/a.txt b/a.txt
```

```
index 2fe4ab0..b1e6722 100644
```

```
--- a/a.txt
```

```
+++ b/a.txt
```

```
@@ -1,2 +1,3 @@
```

```
A
```

```
-B
```

```
₩ No newline at end of file
```

```
+B
```

```
+C
```

명령어로 원격 저장소 사용

❖ git fetch

- ✓ origin/main으로 체크 아웃: `git checkout origin/main`

Note: switching to 'origin/main'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-c` with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

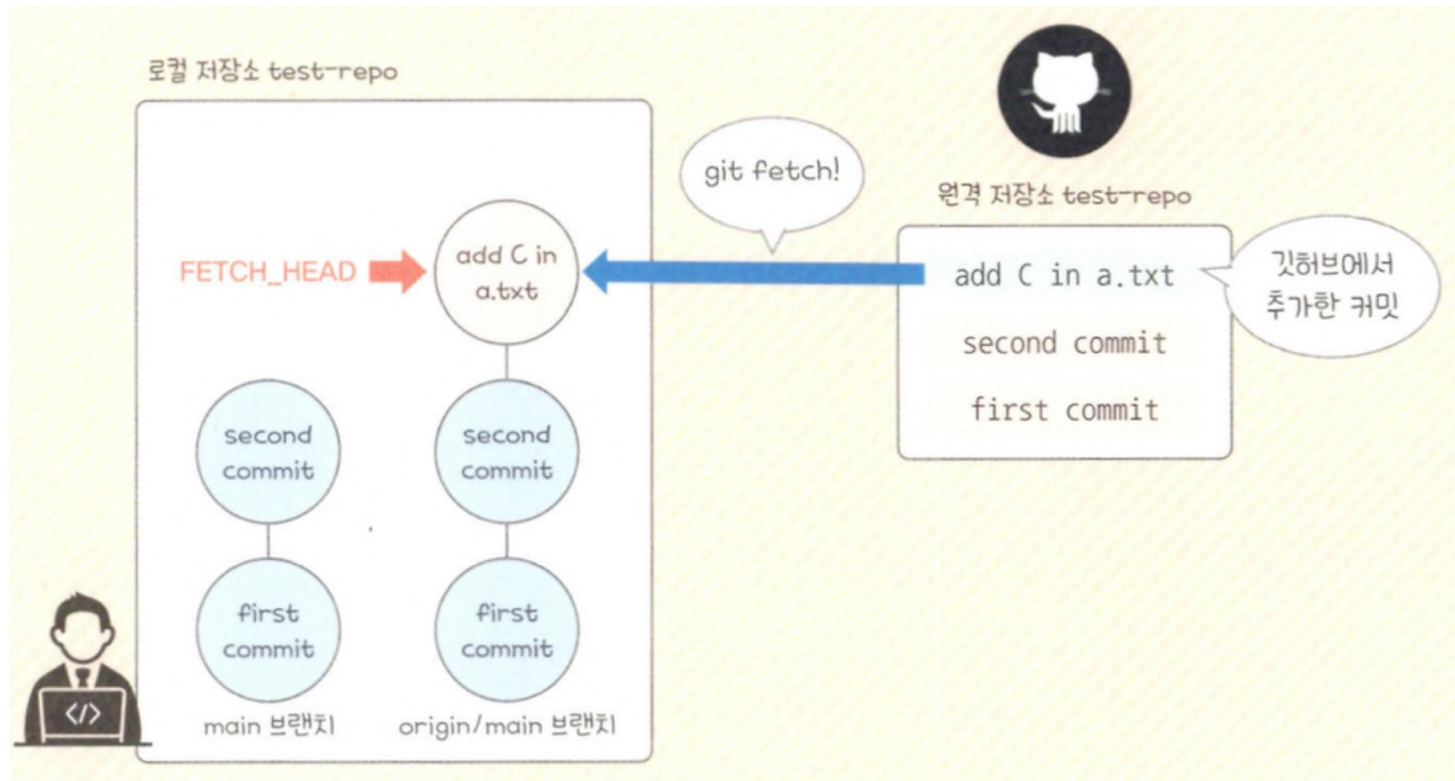
Turn off this advice by setting config variable `advice.detachedHead` to false

HEAD is now at 0f8b9c3 add C in a.txt

명령어로 원격 저장소 사용

❖ git fetch

- ✓ git checkout FETCH_HEAD 명령으로도 체크 아웃 할 수 있음
- ✓ FETCH_HEAD는 최근에 패치한 branch의 최신 Commit을 가리키기 때문
- ✓ FETCH_HEAD는 Git Hub에서 추가한 Commit(add C in a.txt)을 가리킴



명령어로 원격 저장소 사용

❖ git fetch

✓ 현재 Commit 확인 - git log

commit 0f8b9c3d0eff09a271026f0f1243a9e68da7ff28 (HEAD, origin/main)

Author: itstudy <72733143+itstudy001@users.noreply.GitHub.com>

Date: Mon Oct 16 10:09:45 2023 +0000

add C in a.txt

commit e587e15e2eb65647e6138d49a8709dc30d5ffa42 (main)

Author: itstudy001 <itstudy@kakao.com>

Date: Mon Oct 16 19:05:25 2023 +0900

second commit

commit 3fa4be0c4f8d042ceb5121ef4b208e609537b0b1

Author: itstudy001 <itstudy@kakao.com>

명령어로 원격 저장소 사용

❖ git fetch

✓ 변경된 내용을 main branch에 병합

● git checkout main

Previous HEAD position was 0f8b9c3 add C in a.txt

Switched to branch 'main'

Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.

(use "git pull" to update your local branch)

● git merge origin/main

Updating e587e15..0f8b9c3

Fast-forward

a.txt | 3 ++-

1 file changed, 2 insertions(+), 1 deletion(-)

명령어로 원격 저장소 사용

❖ git fetch

✓ git log

commit 0f8b9c3d0eff09a271026f0f1243a9e68da7ff28 (HEAD -> main, origin/main)

Author: itstudy <72733143+itstudy001@users.noreply.GitHub.com>

Date: Mon Oct 16 10:09:45 2023 +0000

add C in a.txt



명령어로 원격 저장소 사용

❖ git pull

- ✓ 원격 저장소를 가져와서 합치는 명령
- ✓ 풀은 패치와 병합을 동시에 하는 방식
- ✓ Git Hub에서 a.txt 파일을 클릭한 후 D를 추가하고 add D in a.txt 메시지를 작성하고 Commit
- ✓ 현재 상황



명령어로 원격 저장소 사용

❖ git pull

- ✓ 현재 로컬 Repository 상황 – git log

commit 57486ea690958b0e2ff80d201a7cbba5c2c10420 (HEAD -> main, origin/main)
Author: itstudy <72733143+itstudy001@users.noreply.github.com>
Date: Thu Apr 16 15:15:27 2026 +0900

Third Commit

commit d6c182d856a2119a1bceaa721aa9abee5e7ba636
Author: adam <itstudy@kakao.com>
Date: Thu Apr 16 15:01:17 2026 +0900

second commit

commit 367acd7f28ec5007fedd411fb7a767b67e6b6fa4
Author: adam <itstudy@kakao.com>
Date: Thu Apr 16 09:07:46 2026 +0900

a.txt 추가

명령어로 원격 저장소 사용

❖ git pull

✓ 원격 저장소 풀 – git pull

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 620 bytes | 310.00 KiB/s, done.

From <https://GitHub.com/itstudy001/test-repo>

0f8b9c3..ed6e5f1 main -> origin/main

Updating 0f8b9c3..ed6e5f1

Fast-forward

a.txt | 1 +

1 file changed, 1 insertion(+)

명령어로 원격 저장소 사용

❖ git pull

✓ 현재 로컬 Repository 상황 – git log

commit 0b5ae2aba37af361813aa558bc8a6933ad064beb (HEAD -> main, origin/main)
Author: itstudy <72733143+itstudy001@users.noreply.github.com>
Date: Thu Apr 16 15:23:36 2026 +0900

Fourth Commit

commit 57486ea690958b0e2ff80d201a7cbba5c2c10420
Author: itstudy <72733143+itstudy001@users.noreply.github.com>
Date: Thu Apr 16 15:15:27 2026 +0900

Third Commit

명령어로 원격 저장소 사용

❖ git pull

✓ 현재 로컬 Repository 상황 – git log

commit d6c182d856a2119a1bceaa721aa9abee5e7ba636

Author: adam <itstudy@kakao.com>

Date: Thu Apr 16 15:01:17 2026 +0900

second commit

commit 367acd7f28ec5007fedd411fb7a767b67e6b6fa4

Author: adam <itstudy@kakao.com>

Date: Thu Apr 16 09:07:46 2026 +0900

a.txt 추가

명령어로 원격 저장소 사용

❖ Pull Request 보내기

- ✓ 작업한 내용을 프로젝트에 합쳐달라고 요청을 보내는 것
- ✓ 기존 프로젝트에 Pull Request 보내기

● 과정

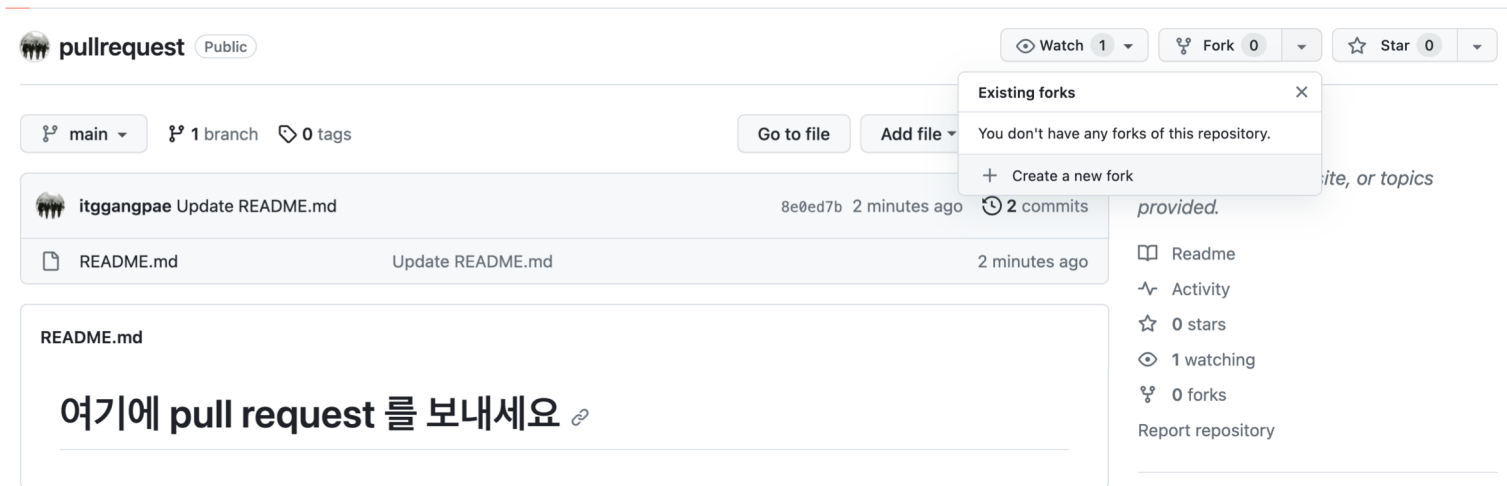
- ◆ 기여하려는 저장소를 본인 계정으로 Fork
- ◆ Fork한 저장소를 클론
- ◆ branch 생성 후 생성한 branch에서 작업
- ◆ 작업한 branch 푸시
- ◆ Pull Request 보내기

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- <https://github.com/ggangpae1/pr> 에 접속해서 Fork를 선택해서 create new fork를 선택



명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- <https://github.com/itgangpae/pullrequest.git> 에 접속해서 Fork를 선택해서 create new fork를 선택

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner *

 itstudy001 ▾

Repository name *

pullrequest

✔ pullrequest is available.

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

Description (optional)

fork

☒ Copy the **main** branch only

Contribute back to itgangpae/pullrequest by adding your own branch. [Learn more.](#)

ⓘ You are creating a fork in your personal account.

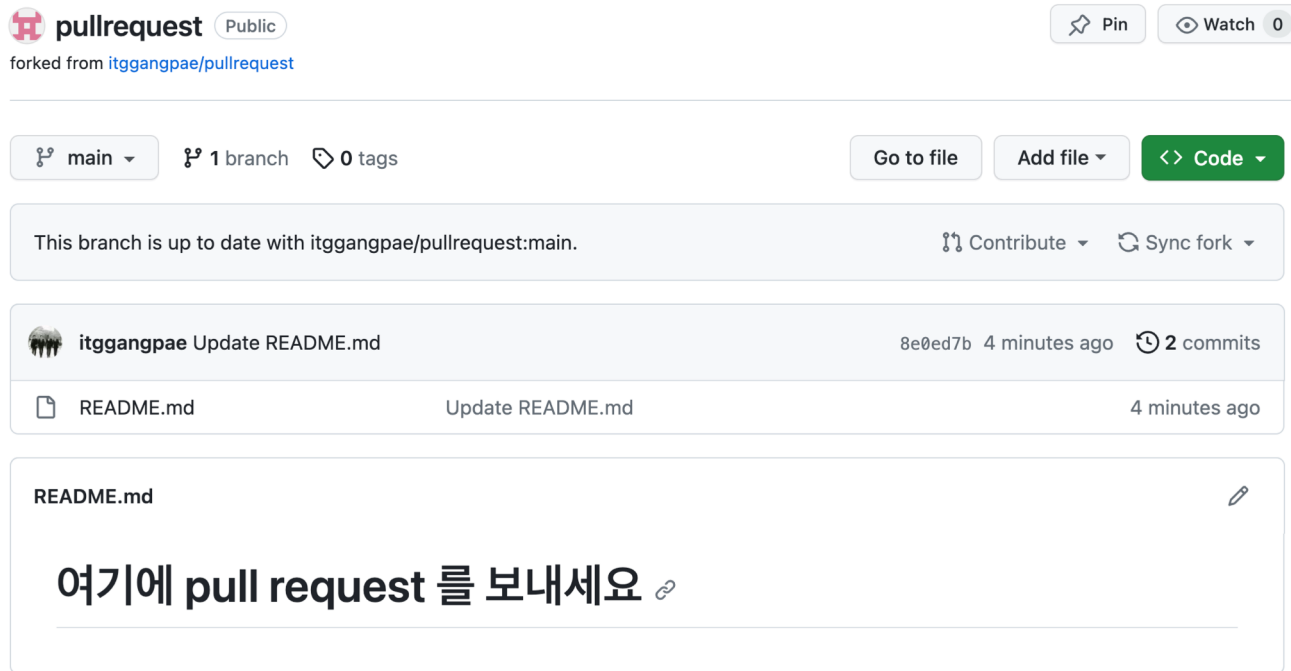
Create fork

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- <https://github.com/itgangpae/pr/fork>에 접속해서 fork를 선택해서 create new fork를 선택



명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- <https://github.com/itggangpae/pr/fork>에 접속해서 fork를 선택해서 create new fork를 선택



명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- 원격 저장소를 Fork한 이후로도 원래의 원격 저장소에는 계속해서 새로운 Commit이 쌓일 수 있는데 이 경우 Fork한 원격 저장소는 원본 저장소에 비해 몇 Commit 뒤쳐지게 됨



명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- 원격 저장소를 Fork한 이후로도 원래의 원격 저장소에는 계속해서 새로운 Commit이 쌓일 수 있는데 이 경우 Fork한 원격 저장소는 원본 저장소에 비해 몇 Commit 뒤쳐지게 됨

The screenshot shows a GitHub repository named 'pullrequest' which is public and forked from 'itgangpae/pullrequest'. The repository has 0 stars and 1 fork. The main branch is selected, and it is 1 commit behind the upstream 'main' branch. A recent commit by 'itgangpae' is shown, updating the README.md file. The README content is '여기에 pull request 를 보내세요'. The right sidebar shows the 'About' section with links to Readme, Activity, Stars, Watching, and Forks, and a 'Releases' section with a link to 'Create a new release'.

pullrequest Public
forked from itgangpae/pullrequest

Pin Watch 0 Fork 1 Star 0

main 1 branch 0 tags
Go to file Add file <> Code

This branch is 1 commit behind itgangpae:main. Contribute Sync fork

itgangpae Update README.md 8e0ed7b 9 minutes ago 2 commits

README.md Update README.md 9 minutes ago

README.md

여기에 pull request 를 보내세요

About
fork
Readme
Activity
0 stars
0 watching
1 fork

Releases
No releases published
Create a new release

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- Sync fork를 눌러서 원격 저장소가 원본 저장소에 비해 뒤쳐지지 않도록 맞춤

The screenshot shows a GitHub repository page for a user named 'pullrequest'. The repository is public and was forked from 'itgangpae/pullrequest'. The main branch is 'main', and there is 1 branch and 0 tags. A warning message states: 'This branch is 1 commit behind itgangpae:main.' Below this, a commit by 'itgangpae' is shown: 'Update README.md'. The README.md file is visible, with the text '여기에 pull request 를 보내세요' (Please send a pull request here). A modal window in the center of the screen displays the warning: 'This branch is out-of-date' and provides instructions to update the branch by syncing 1 commit from the upstream repository. It includes a 'Compare' button and a green 'Update branch' button. The right sidebar shows repository statistics: 0 stars, 0 watching, and 1 fork. The 'About' section lists 'Readme', 'Activity', and 'Releases'.

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- 프로젝트 Clone: `git clone https://github.com/itstudy001/pr.git`
- Clone된 프로젝트 내에서 새로운 branch 생성: `git branch itstudy`
- 생성된 branch로 체크 아웃: `git checkout itstudy`
- 새로운 branch에서 내용을 수정
- `git diff` 명령으로 변경된 내용 확인

```
diff --git a/README.md b/README.md
```

```
index d6b1d99..198a498 100644
```

```
--- a/README.md
```

```
+++ b/README.md
```

```
@@ -1,2 +1,3 @@
```

```
# 여기에 pull request 를 보내세요
```

```
새로운 Commit 을 추가
```

```
+itstudy가 내용을 추가
```

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- git add README.md

- git commit -m "itstudy"

[itstudy 5f5aac7] itstudy

1 file changed, 1 insertion(+)

- git push 명령으로 새로운 branch 푸시: git push origin itstudy

Total 0 (delta 0), reused 0 (delta 0), pack-reused 0

remote:

remote: Create a pull request for 'itstudy' on GitHub by visiting:

remote: <https://GitHub.com/itstudy001/pullrequest/pull/new/itstudy>

remote:

To <https://GitHub.com/itstudy001/pullrequest.git>

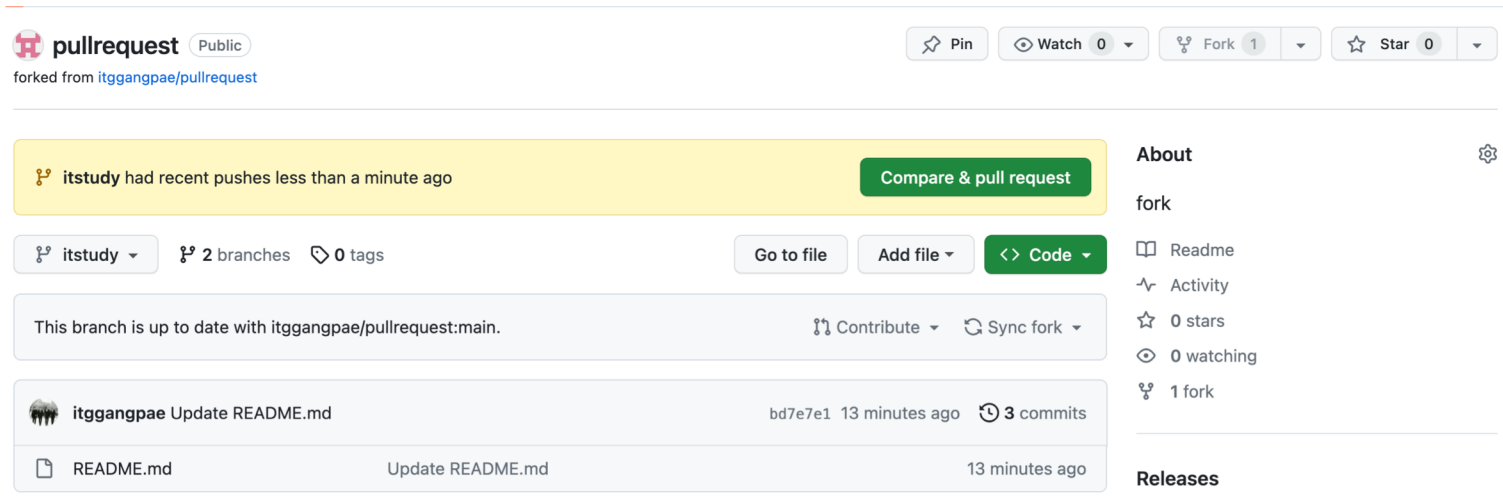
* [new branch] itstudy -> itstudy

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- git hub에서 Compare & pull request 버튼 클릭



The screenshot shows a GitHub pull request interface. At the top, the repository name 'pullrequest' is displayed as 'Public', with a note 'forked from itgangpae/pullrequest'. Action buttons include 'Pin', 'Watch 0', 'Fork 1', and 'Star 0'. A yellow notification bar states 'itstudy had recent pushes less than a minute ago' with a 'Compare & pull request' button. Below this, repository details show 'itstudy' with '2 branches' and '0 tags', and a 'Go to file' button. A status message indicates 'This branch is up to date with itgangpae/pullrequest:main.' with 'Contribute' and 'Sync fork' options. A commit history table shows a commit by 'itgangpae' titled 'Update README.md' with hash 'bd7e7e1' and '3 commits' made '13 minutes ago'. The commit details table shows a file 'README.md' with the action 'Update README.md' also from '13 minutes ago'. On the right, the 'About' section lists 'fork', 'Readme', 'Activity', '0 stars', '0 watching', and '1 fork'. The 'Releases' section is partially visible at the bottom.

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- git hub에서 Compare & pull request 버튼 클릭

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](#). [Learn more about diff comparisons here](#).

base repository: itgangpae/pullrequest base: main head repository: itstudy001/pullrequest compare: itstudy

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

itstudy

Write Preview

itstudy 가 추가함

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

☒ Allow edits by maintainers Create pull request

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

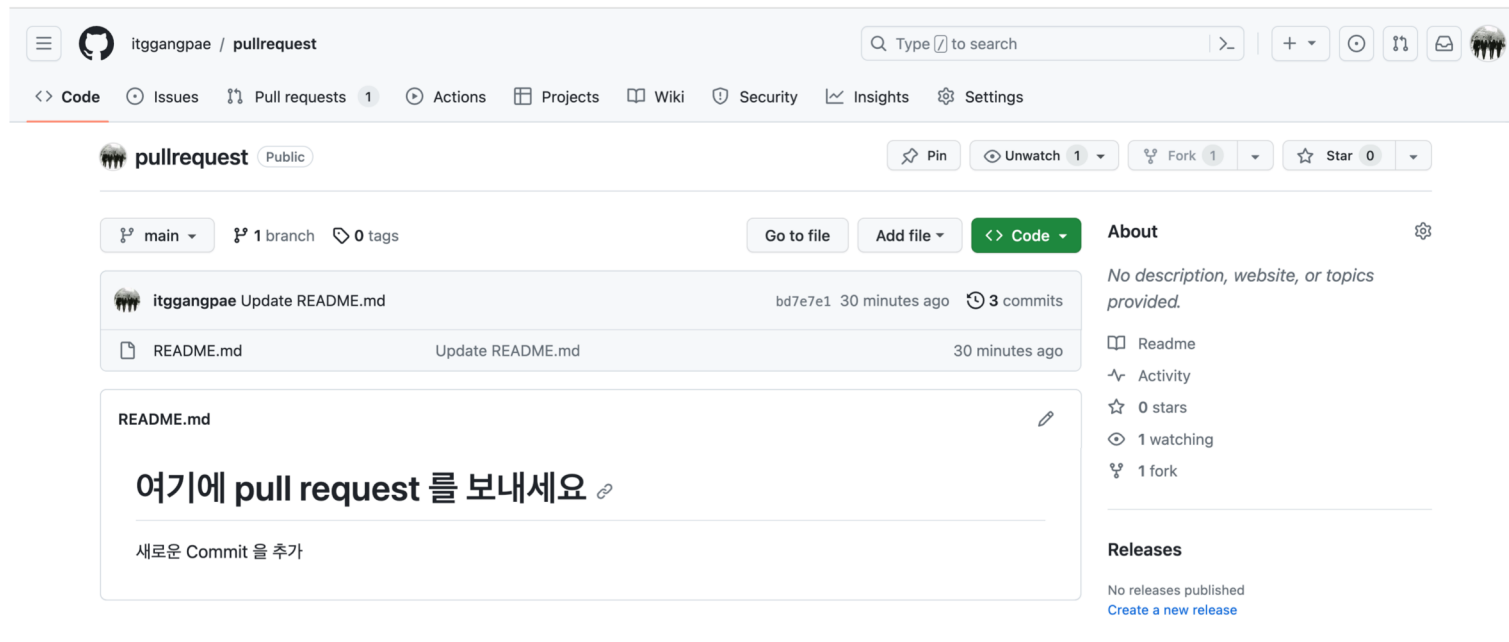
- git hub에서 Compare & pull request 버튼 클릭

The screenshot shows a GitHub Pull Request interface. At the top, the navigation bar includes links for Code, Issues, Pull requests (1), Actions, Projects, Security, and Insights. The main heading is 'itstudy #1' with 'Edit' and '<> Code' buttons. Below this, a green 'Open' button is followed by the text 'itstudy001 wants to merge 1 commit into itgangpae:main from itstudy001:itstudy'. A progress bar shows '+1 -0' changes. The 'Conversation' tab is active, showing a comment from 'itstudy001' stating 'itstudy 가 추가함' with a smiley face emoji. Below the comment is a commit diff snippet for 'itstudy' with hash '5f5aac7'. A green checkmark icon indicates 'This branch has no conflicts with the base branch', with a note that only those with write access can merge pull requests. On the right side, the 'Reviewers' section shows 'No reviews' and a 'Convert to draft' link. The 'Assignees' section shows 'No one assigned'. The 'Labels' and 'Projects' sections both show 'None yet'.

명령어로 원격 저장소 사용

❖ Pull Request 보내기

- ✓ 기존 프로젝트에 Pull Request 보내기
 - Itgangpae 계정의 repository



명령어로 원격 저장소 사용

❖ Pull Request 보내기

- ✓ 기존 프로젝트에 Pull Request 보내기
 - Itgangpae 계정의 repository

itgangpae / pullrequest

Q Type to search

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

Filters Q is:pr is:open Labels 9 Milestones 0 New pull request

1 Open 0 Closed

Author Label Projects Milestones Reviews Assignee Sort

itstudy

#1 opened 15 minutes ago by itstudy001

ProTip! Filter pull requests by the default branch with `base:main`.

명령어로 원격 저장소 사용

❖ Pull Request 보내기

- ✓ 기존 프로젝트에 Pull Request 보내기
 - Itggangpae 계정의 repository

itstudy #1

 **Open** itstudy001 wants to merge 1 commit into [itggangpae:main](#) from [itstudy001:itstudy](#) 

 Conversation 0  Commits 1  Checks 0  Files changed 1

 **itstudy001** commented 16 minutes ago First-time contributor ...

itstudy 가 추가함



  itstudy 5f5aac7

Add more commits by pushing to the **itstudy** branch on **itstudy001/pullrequest**.

 **Require approval from specific reviewers before merging**
[Branch protection rules](#) ensure specific people approve pull requests before they're merged. Add rule ×

 **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.

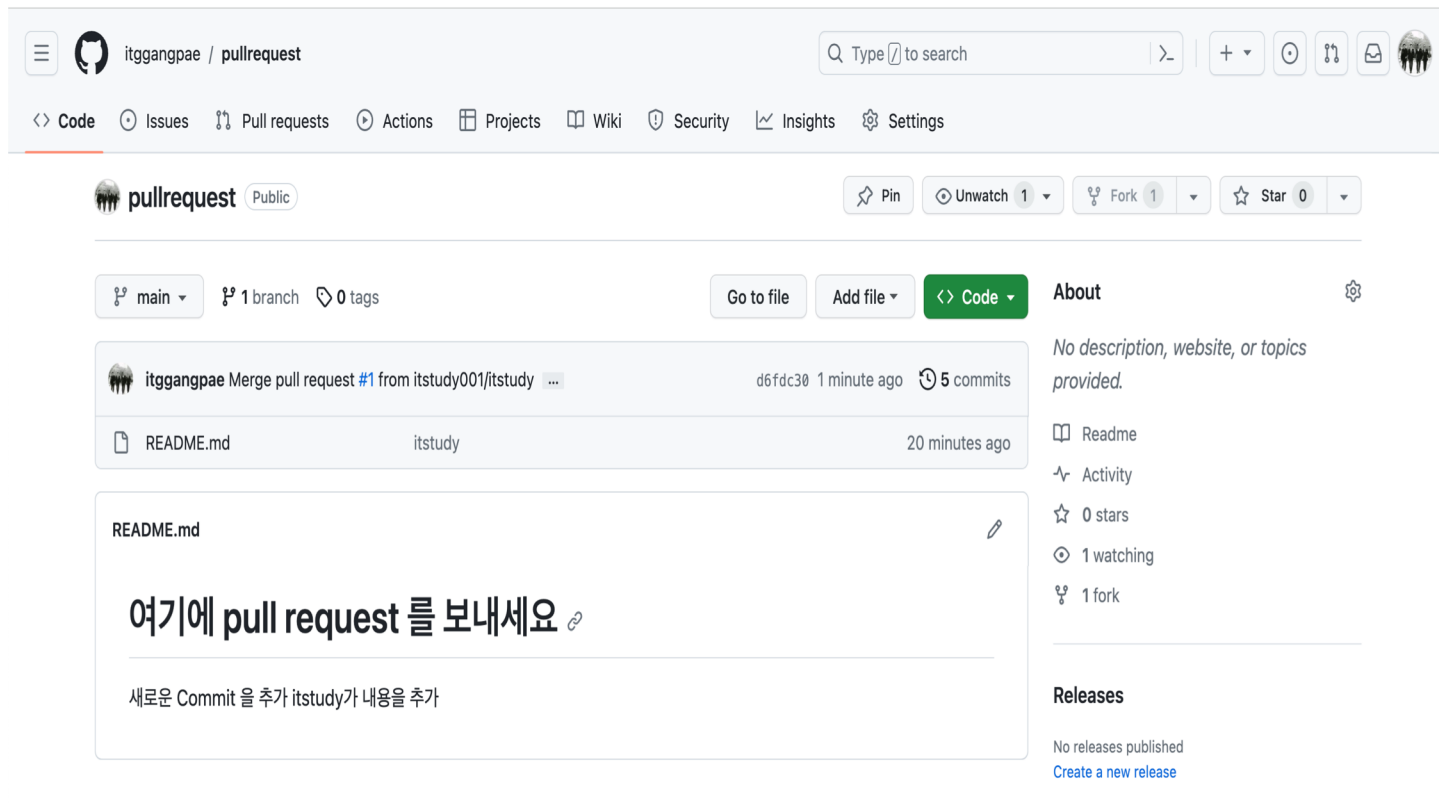
Merge pull request  You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- Itgangpae 계정의 repository에서 Merge를 선택하고 메시지 작성

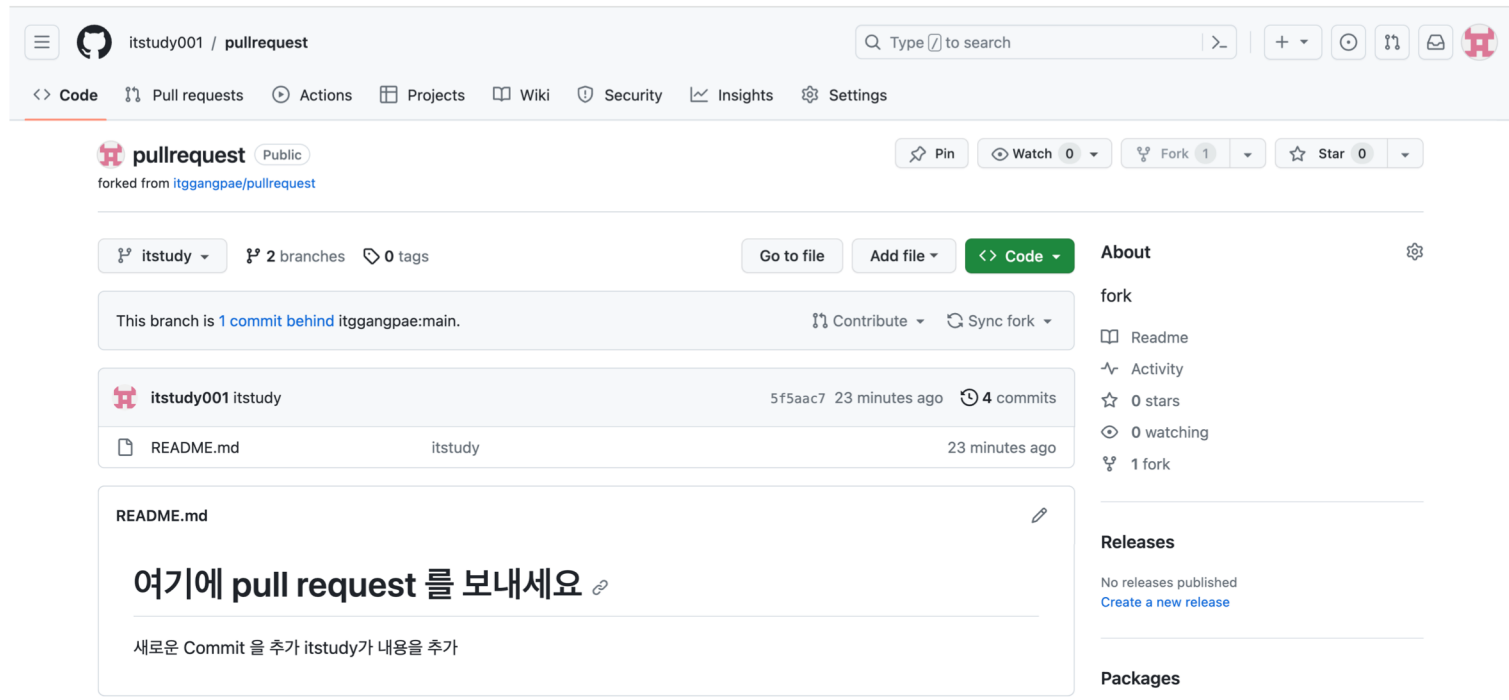


명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 기존 프로젝트에 Pull Request 보내기

- Pull request를 보낸 계정의 repository



명령어로 원격 저장소 사용

❖ Pull Request 보내기

✓ 최소 동의 수(Required Approvals)를 설정

- Repository Settings 진입: 해당 GitHub 레포지토리 상단의 메뉴 바에서 [Settings] 탭을 클릭
- Branches 메뉴 선택: 왼쪽 사이드바의 Code and automation 섹션에서 [Branches]를 클릭
- Branch protection rule 추가 또는 수정: 규칙이 없다면 [Add branch protection rule]을 클릭하고 규칙이 있다면 해당 규칙 옆의 [Edit]을 누름
- 대상 브랜치 지정: Branch name pattern 칸에 보호할 브랜치 이름(예: main 또는 develop)을 입력
- 보호 규칙 활성화: [Require a pull request before merging] 체크박스를 활성화
- 승인 인원 설정: 하위 옵션인 [Require approvals]에 체크하고 바로 아래 드롭다운 메뉴에서 최소 승인 인원(1~6명)을 선택
- 저장: 페이지 하단으로 내려가 [Create] 또는 **[Save changes]**를 클릭하여 설정을 완료