

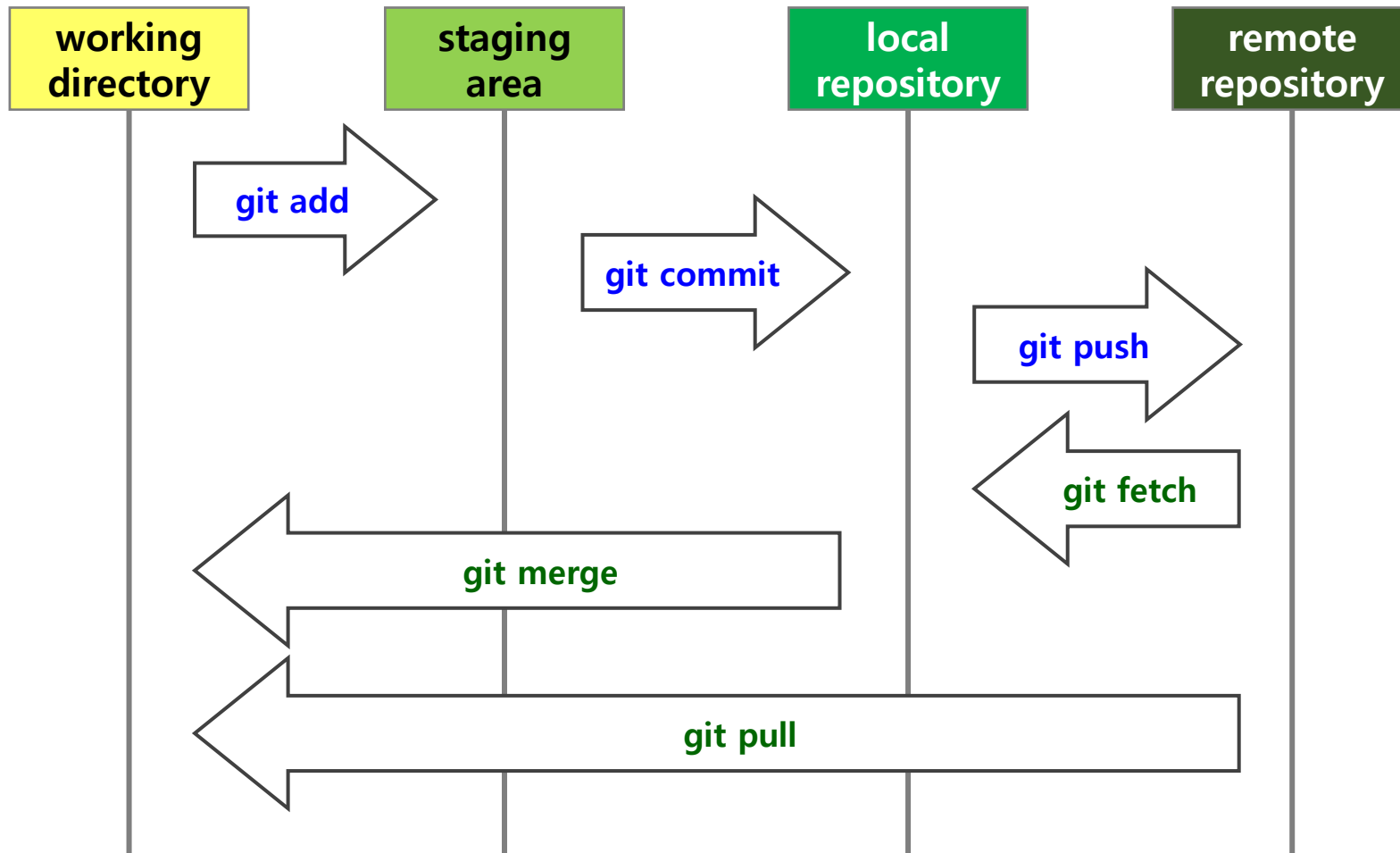
# GitHub 협업

**fork, pull request**

동의과학대학교 컴퓨터정보과

# 개인이 혼자 git과 github 사용 시 절차

- 자신의 local repo에서 자신의 remote repo로 동기화 할 때 절차



# windows 용 git 설치

- 다운로드 : <https://git-scm.com/>



The screenshot shows the Git website homepage. At the top left is the Git logo with the tagline "--fast-version-control". To the right is a search bar. Below the logo, there are two paragraphs of text describing Git as a free and open source distributed version control system. To the right of the text is a diagram showing a branching model with stacks of code and colored lines representing branches. At the bottom, there are four sections: "About", "Documentation", "Downloads", and "Community". The "Downloads" section is highlighted with a red box, and a red arrow points to the "Download 2.24.0 for Windows" button. To the right of the button, the word "다운로드" is written in red. On the right side of the page, there is a monitor displaying the latest source release "2.24.0" and a link to "Download 2.24.0 for Windows".

git --fast-version-control

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint** with **lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release  
**2.24.0**  
Release Notes (2019-11-04)  
**Download 2.24.0 for Windows**

다운로드

# Git – CLI

- windows용 git 을 설치하면 아래의 프로그램을 사용할 수 있음



- 위의 협업 예제는 Git CMD를 사용했으나 이 프로그램이 deprecated(곧 사용되지 않을 예정)임으로 Git Bash를 사용할 것을 추천함

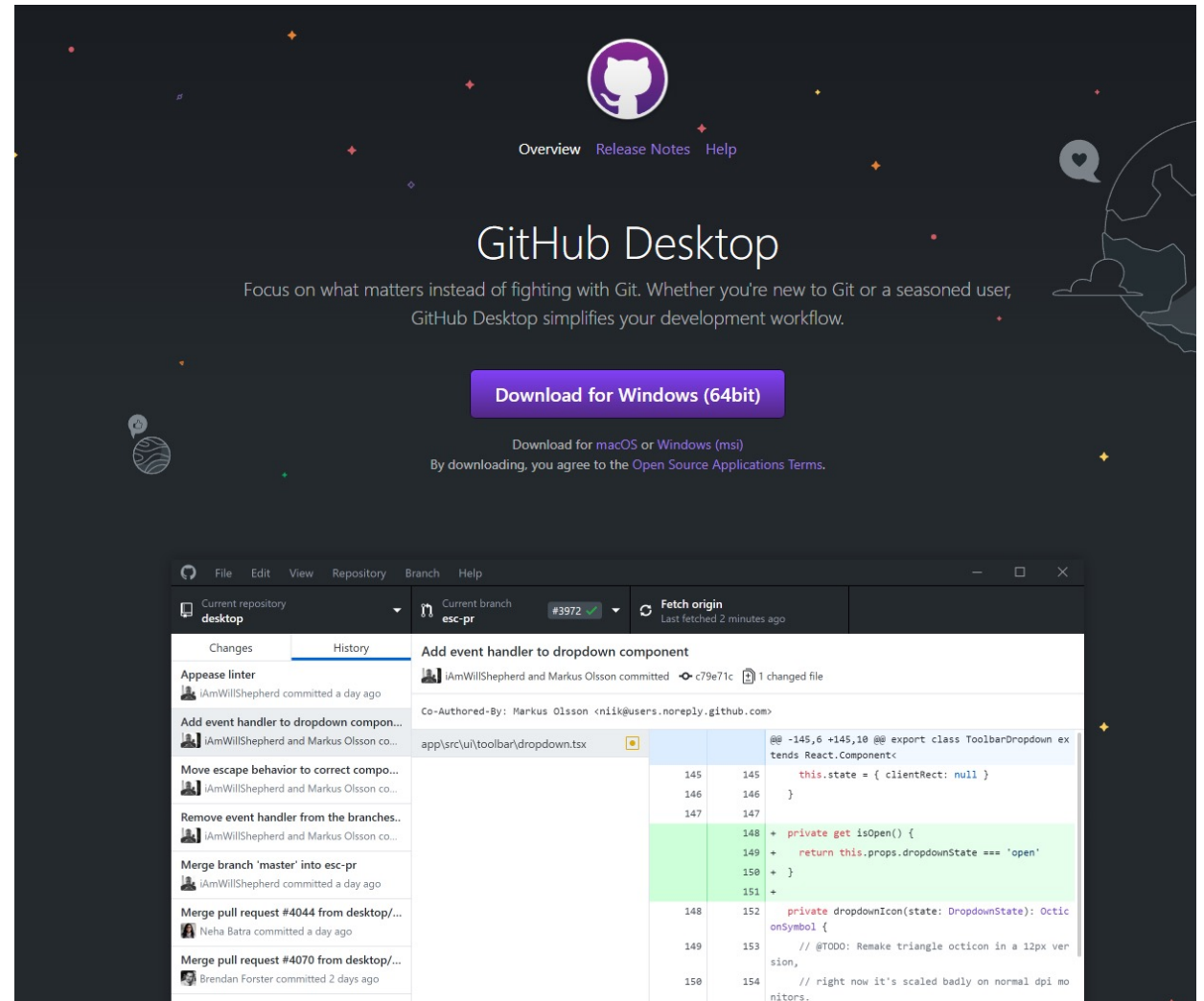
# Git – GUI Clients

- 플랫폼에 따른 다양한 git 관리 클라이언트가 있음
- 차후에 자신에게 맞는 프로그램을 사용하도록 함

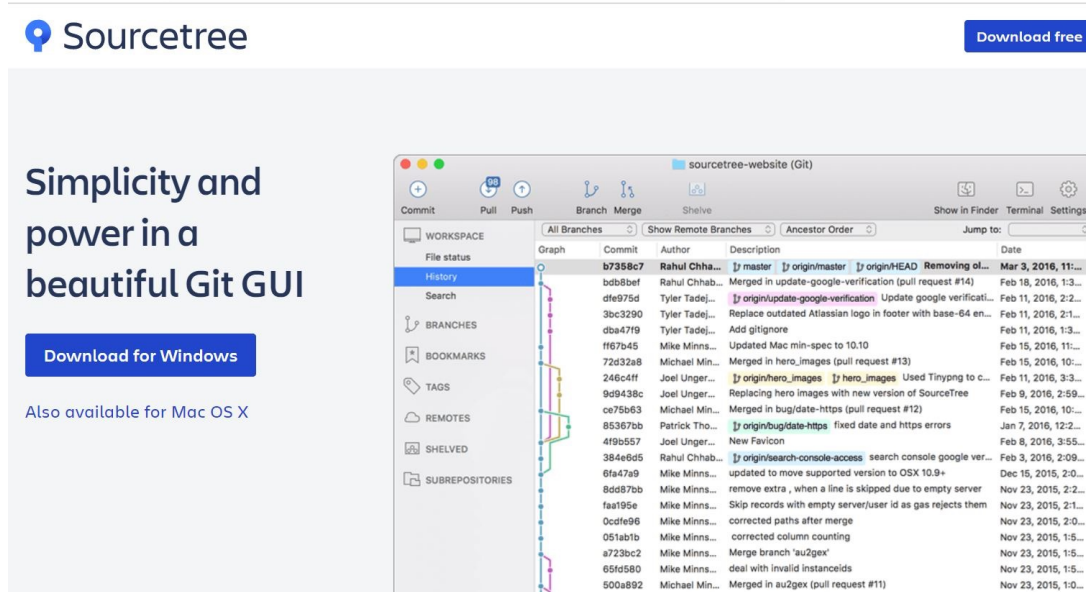
The screenshot shows the Git website's 'GUI Clients' page. At the top, there's a search bar and navigation links like 'About', 'Documentation', 'Downloads', 'GUI Clients', 'Logos', and 'Community'. The main heading is 'GUI Clients'. Below it, a paragraph states: 'Git comes with built-in GUI tools for committing ([git-gui](#)) and browsing ([gitk](#)), but there are several third-party tools for users looking for platform-specific experience.' Another paragraph says: 'If you want to add another GUI tool to this list, just [follow the instructions](#).' Below these are buttons for 'All', 'Windows', 'Mac', 'Linux', 'Android', and 'iOS'. The page displays several GUI client cards: SourceTree (Platforms: Mac, Windows; Price: Free; License: Proprietary), GitHub Desktop (Platforms: Mac, Windows; Price: Free; License: MIT), TortoiseGit (Platforms: Windows; Price: Free; License: GNU GPL), and Git Extensions (Platforms: Linux, Mac, Windows; Price: Free; License: GNU GPL). Each card includes a small image of the software's interface. On the left side, there's a text box mentioning the 'Pro Git' book by Scott Chacon and Ben Straub, available for free on Amazon.com.

# GitHub Desktop

- Git 핵심 기능을 위주로 쉽게 사용 가능 함
- GitHub에서 제작
- <https://desktop.github.com/>



# Source Tree



## A free Git client for Windows and Mac

Sourcetree simplifies how you interact with your Git repositories so you can focus on coding.  
Visualize and manage your repositories through Sourcetree's simple Git GUI.

- 많은 개발자들이 사용
- 다양한 Git 기능 제공
- <https://www.sourcetreeapp.com/>



# GitHub 협업 플로우

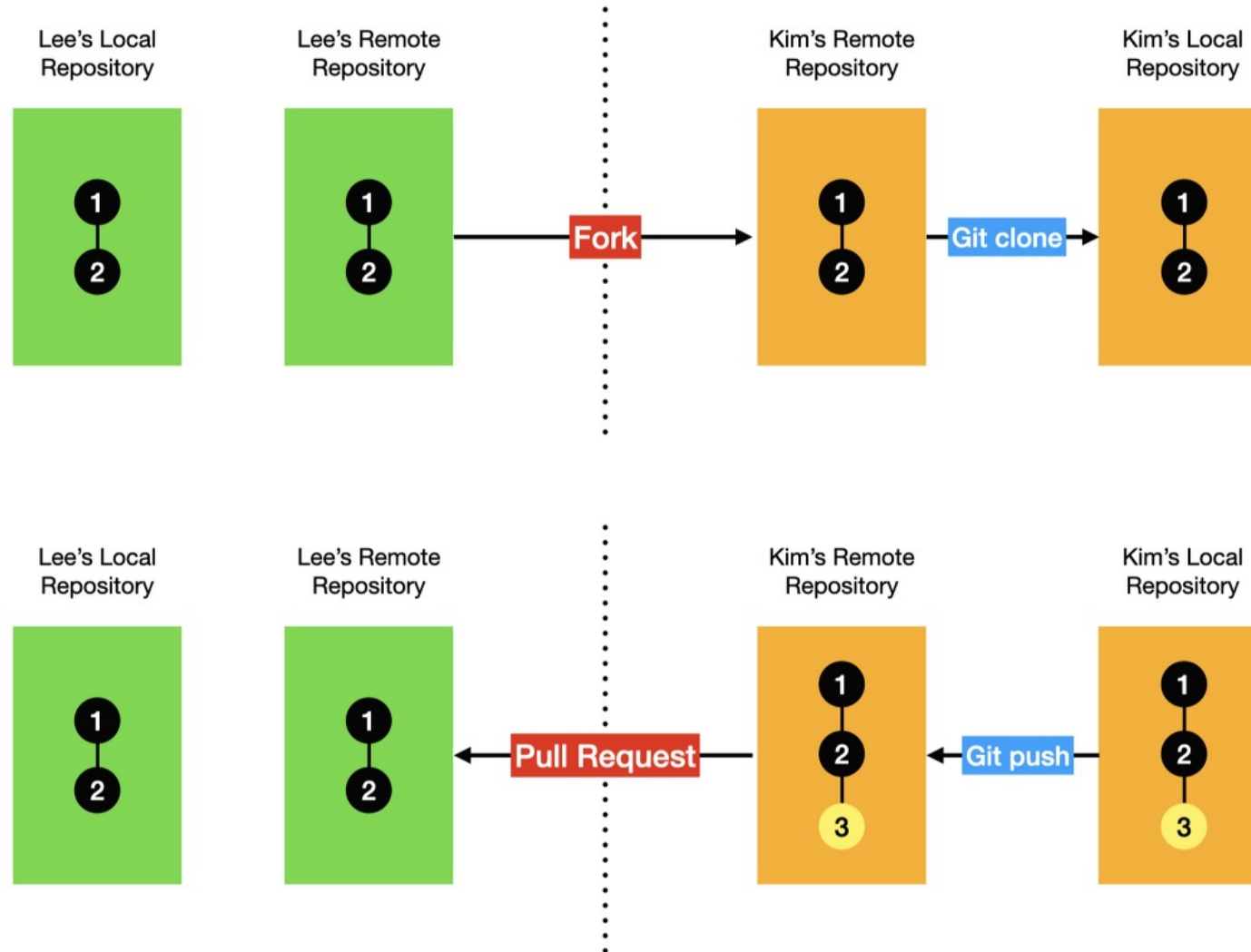
참조 : <https://git-scm.com/book/ko/v2>

1. 프로젝트를 fork 한다.
2. master 기반으로 토픽 브랜치를 만든다.
  - % git clone *GitHub URL*
  - % git branch *branch\_name*
3. 뭔가 수정해서 커밋한다.
4. 자신의 GitHub 프로젝트에 브랜치를 Push 한다.
5. GitHub에 Pull Request를 생성한다.
6. 토론하면서 그에 따라 계속 커밋한다.
7. 프로젝트 소유자는 Pull Request를 merge 하고 닫는다.



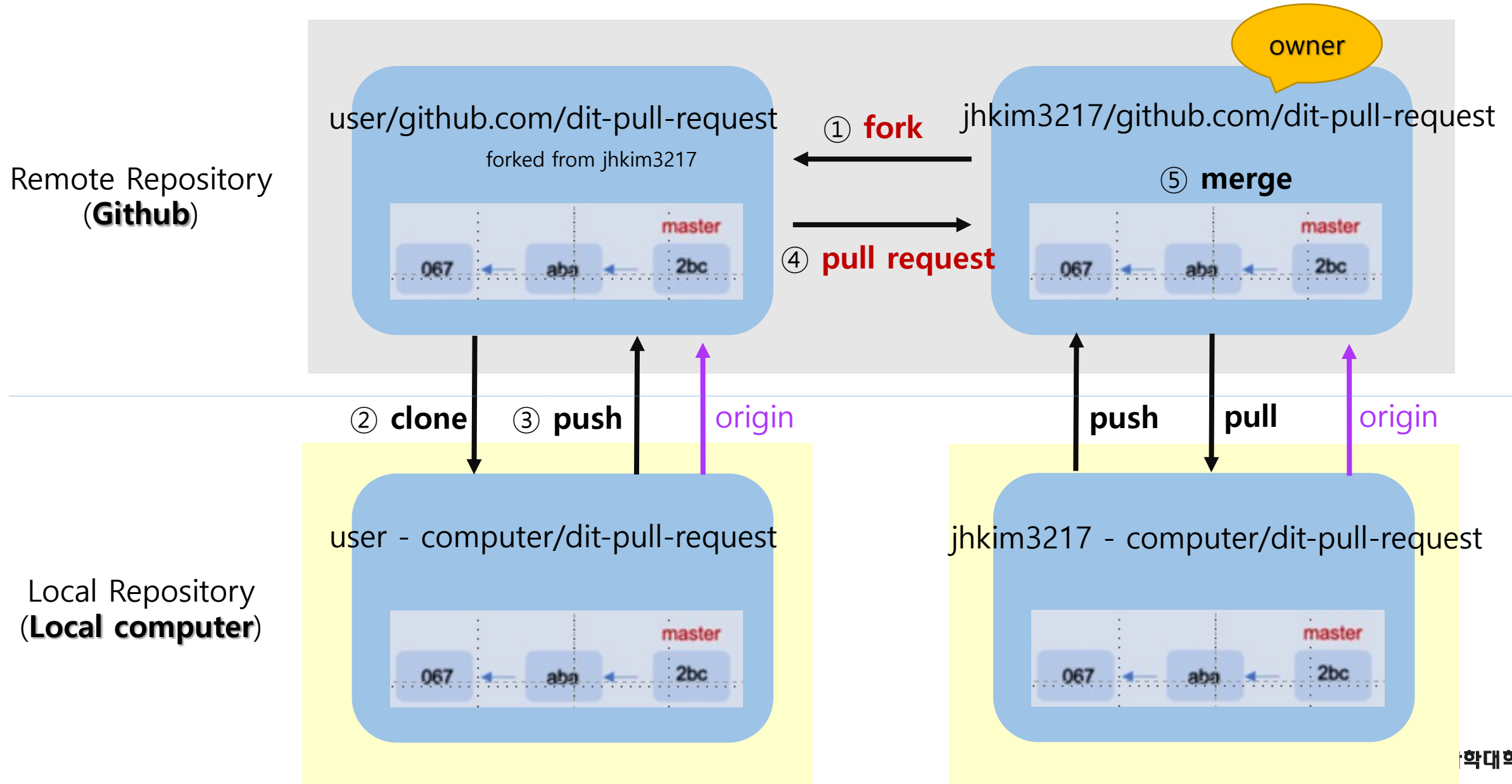
# GitHub Pull Request

참조 : <https://roniruny.tistory.com/113>



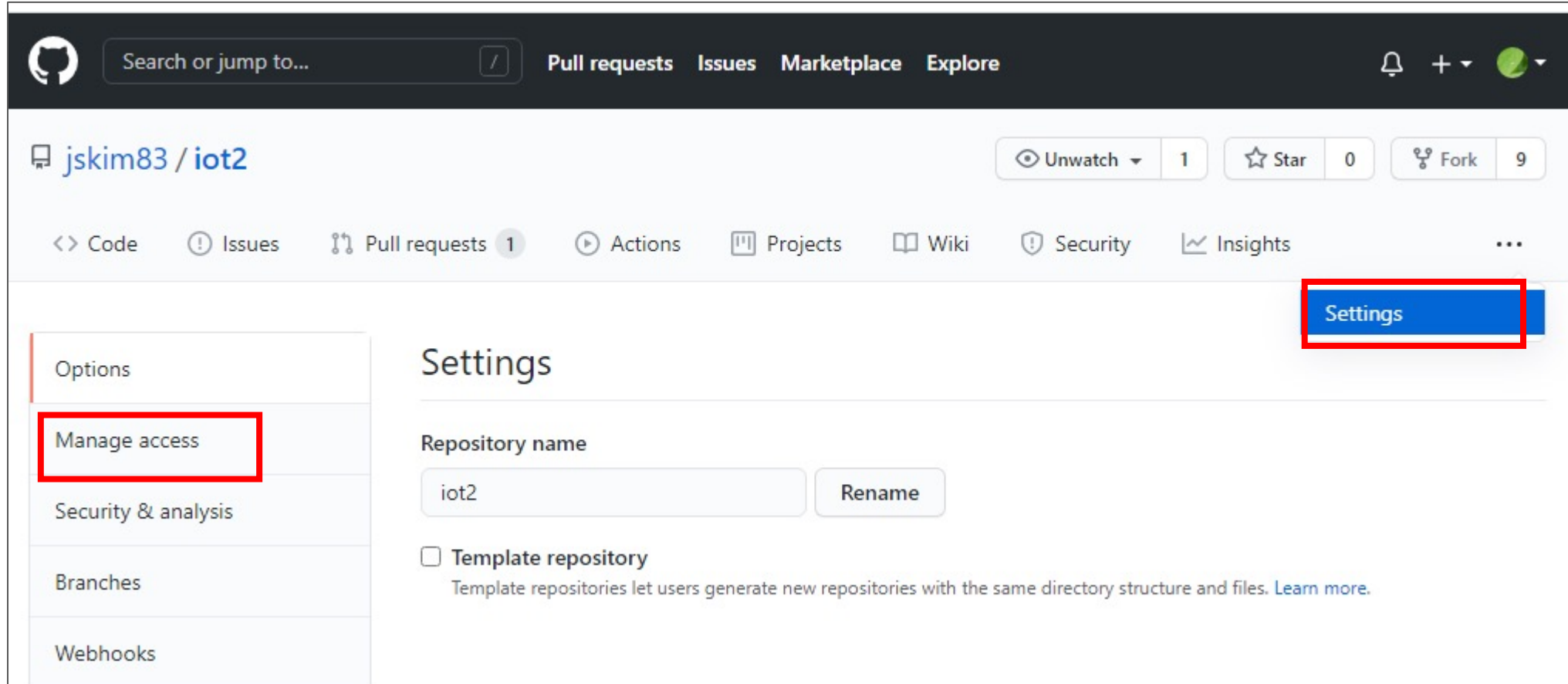
# 협업 실습 구성도

fork URL : <https://github.com/jhkim3217/dit-pull-request.git>



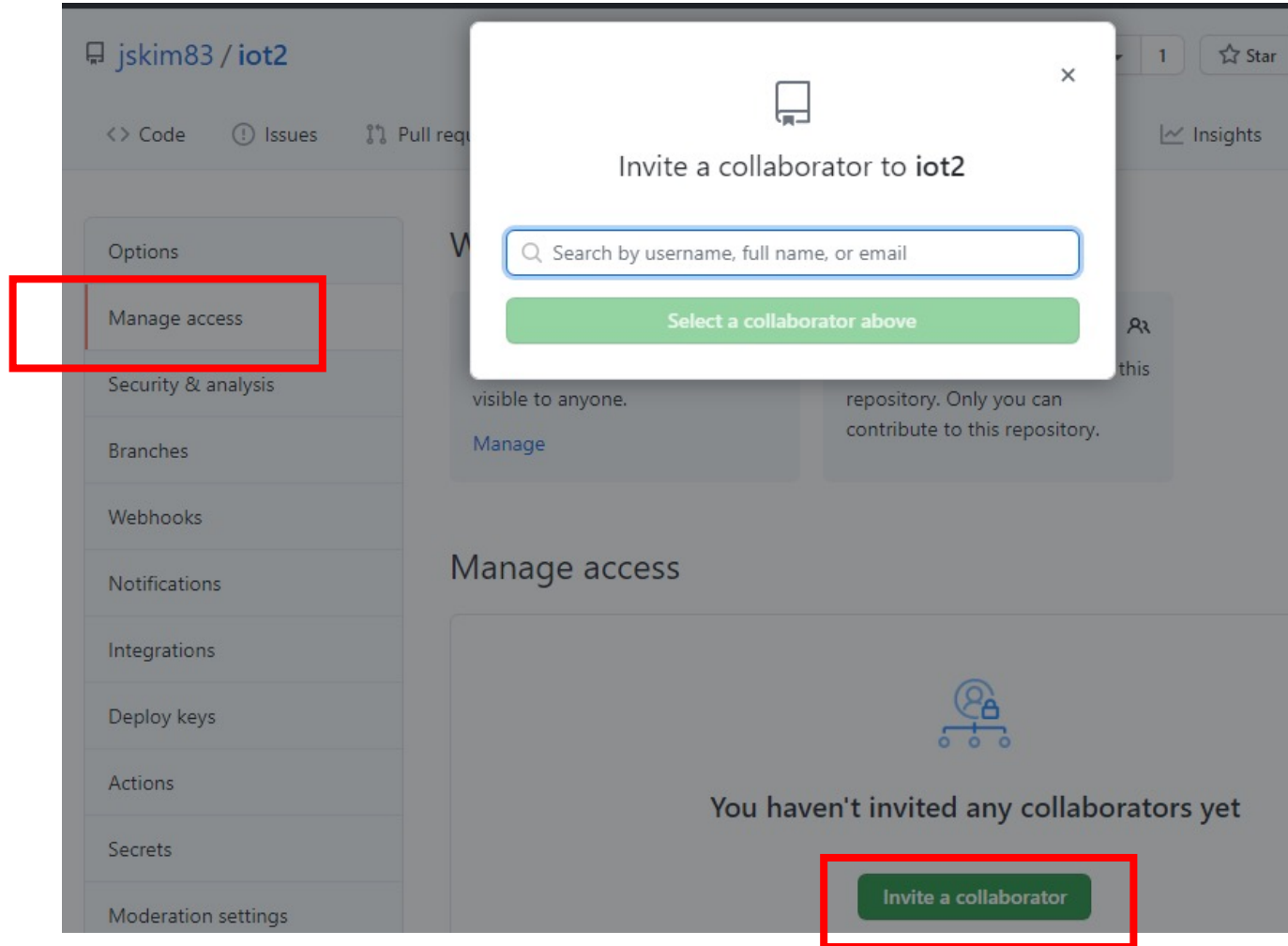
# GitHub 프로젝트에 동료 추가하기

- 해당 저장소에 commit 권한을 주고 싶은 동료가 있으면 "collaborator"로 추가
- 해당 저장소 – settings – Collaborator – 사용자 id 입력 - Add



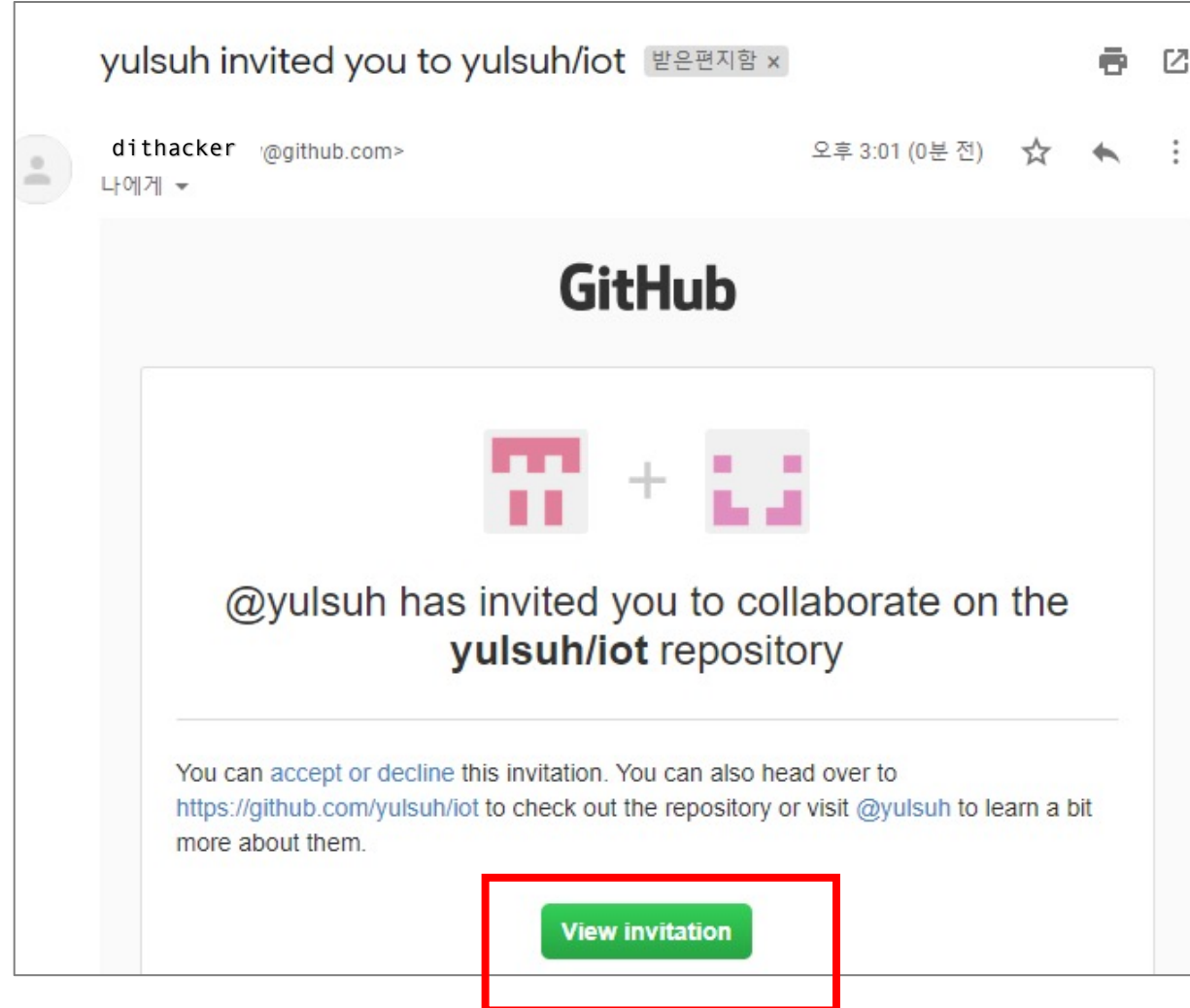
# github 프로젝트에 동료 추가하기

- Collaborator 를 추가하면 해당 동료에게 이메일이 발송된다.



# GitHub 프로젝트에 동료 추가하기

- 이메일을 확인하고 “view Invitation” 버튼을 눌러 초대를 수락한다.



# 실습 : pull request 하기

- <https://github.com/jhkim3217/dit-pull-request>


jhkim3217 / dit-pull-request Public

Unwatch 1 Star 0 Fork 1

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

main 1 branch 0 tags

Go to file Add file Code

 jhkim3217 Update README.md

10d389f 17 hours ago 17 commits

README.md	Update README.md	17 hours ago
request-names.txt	Update request-names.txt	3 days ago

README.md

## Pull-Request 과제

- pull-request 원격저장소(GitHub)를 지역저장소(자신의 PC)에 clone 한다.

```
% git clone https://github.com/jhkim3217/dit-pull-request.git
```
- 지역저장소에서 request-names.txt에 "홍길동(이름) 2반"을 추가 한다.

```
% git add request-names.txt  
% git commit -m "홍길동, 반 추가"  
% git push
```
- 웹브라우저로 자신의 GitHub 원격저장소(dit-pull-request)로 이동하여 Pull requests 요청을 한다.
- 자신의 이름이 request-names.txt에 병합(merge) 되었는지 확인, 병합은 owner가 pull-request를 accept(merge) 한 후 확인 가능

About

No description, website, or topics provided.

Readme


Releases


No releases published  
[Create a new release](#)

Packages

No packages published  
[Publish your first package](#)

Contributors 2

 jhkim3217 김종현Jonghyun Kim

 dithacker