

# **LAB II LECTURE**

## **Git Remote / GitHub**

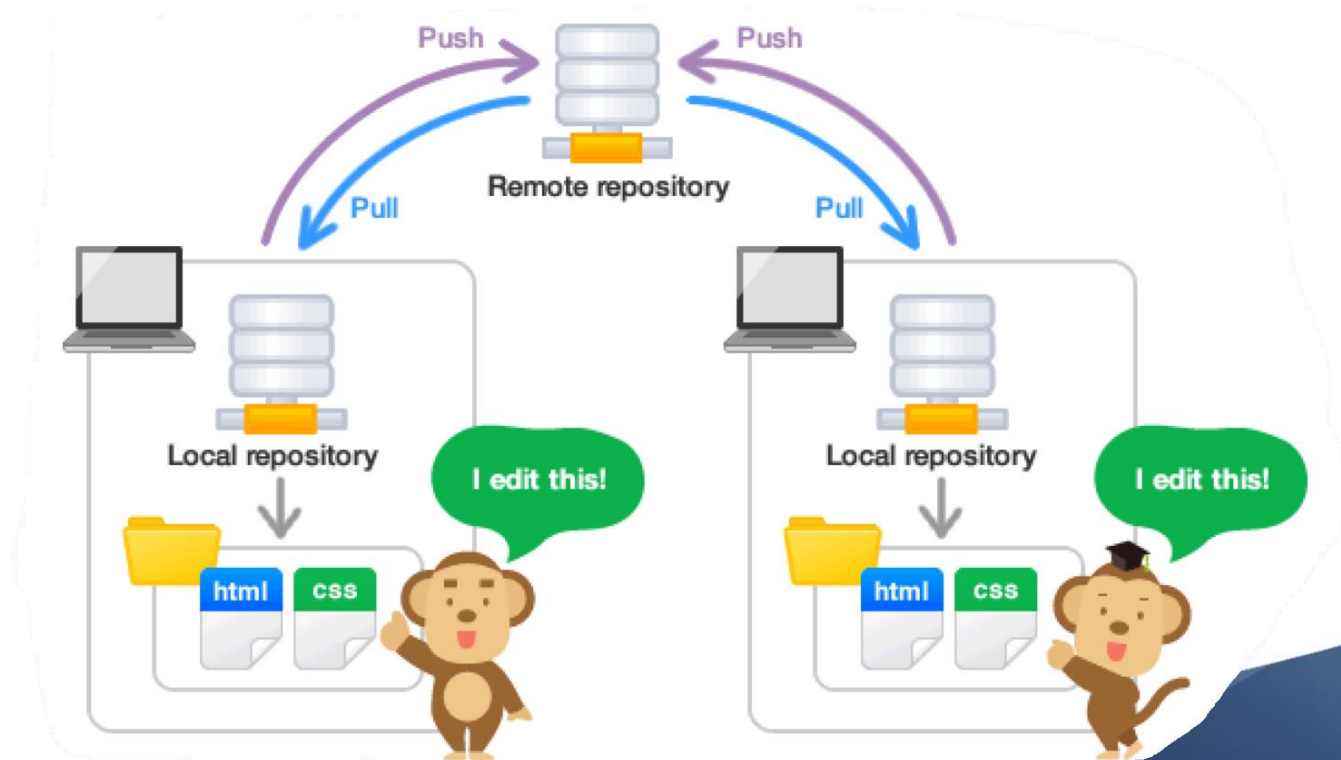
Seoul National University  
Graphics & Media Lab

# Today's Contents

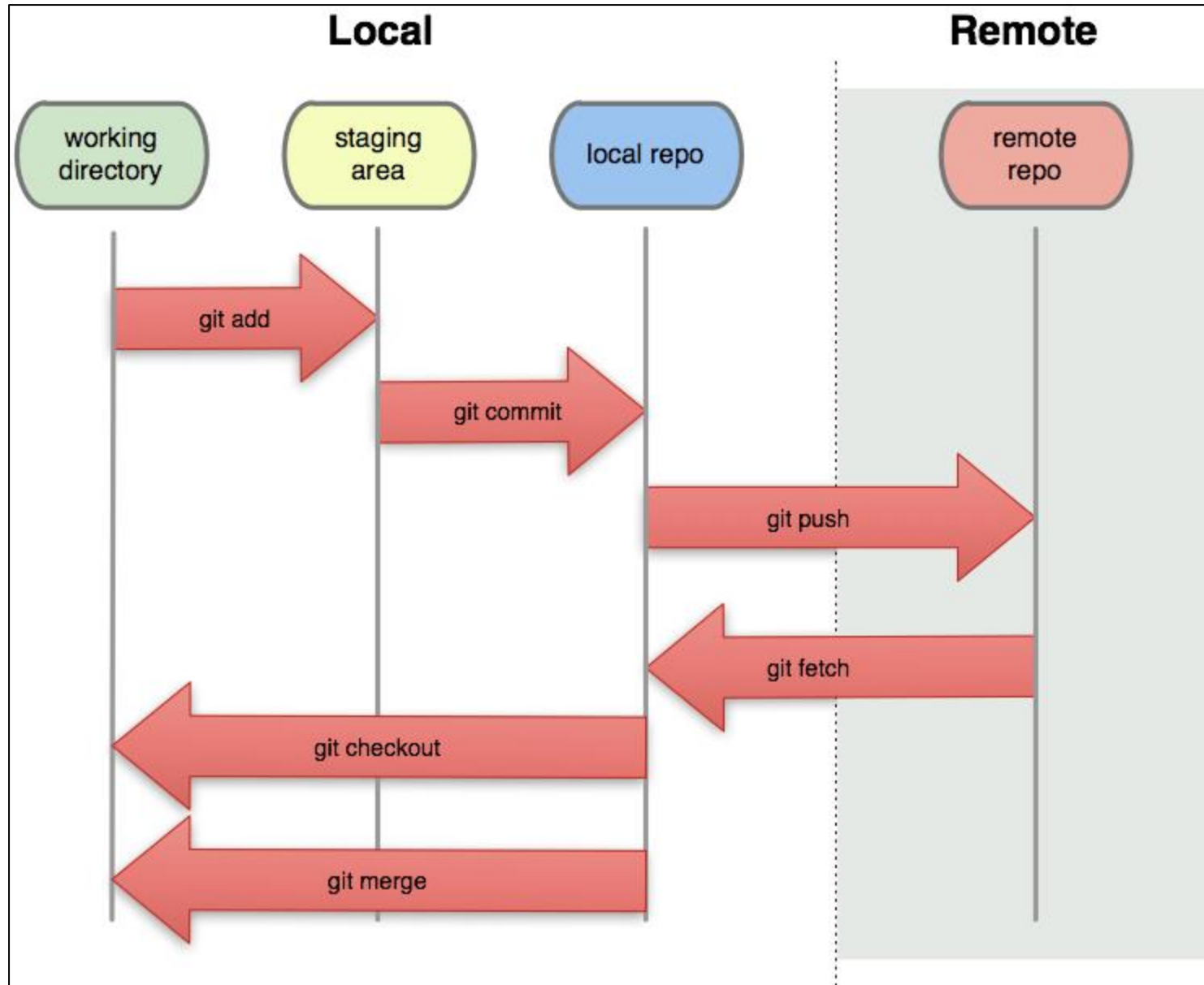
- Working with Remote
- Introduce to GitHub / Setup
- Contributing to a GitHub Project

# Local vs Remote

- Local Repository
  - Reside on a local machine of an individual user
- Remote repository
  - Reside on a remote server
  - Shared among multiple team members



# Working With Remotes



# Creating Remote Repository (GitHub)

- <https://github.com>
- The single largest host for Git repositories
- A large percentage of all git repositories are hosted on GitHub, and many open-source projects use it for Git hosting, issue tracking, code review, and other things

The screenshot displays the GitHub homepage for a user named Snu-ProgMeth. The top navigation bar includes the GitHub logo, a search bar, and links to Pull requests, Issues, Marketplace, and Explore. The main content area is divided into several sections:

- Create your first project:** A section encouraging users to create a new repository or import an existing one. It features a green "Create repository" button and a blue "Import repository" link.
- Recent activity:** A section showing links to recent activity across the user's GitHub profile.
- The home for all developers — including you:** A central section with a welcome message and a "Start writing code" button.
- Start a new repository:** A section with a form to create a new repository. It includes a text input for the repository name, a dropdown menu to select the visibility (Public or Private), and a green "Create a new repository" button.
- Introduce yourself with a profile README:** A section with a form to create a profile README. It includes a text input for the README content, a dropdown menu to select the README name, and a green "Create" button.
- Use tools of the trade:** A section with links to various tools and services, including GitHub Desktop and Visual Studio Code.

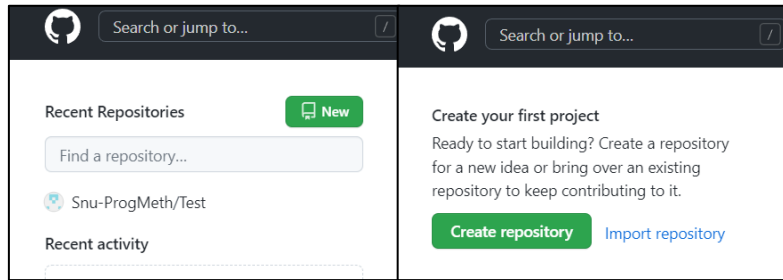
On the right side of the page, there are two promotional banners:

- Universe 2022:** A banner for the global developer event for cloud, security, community, and AI. It includes a "Register now" button.
- GitHub Copilot:** A banner for the AI-powered code completion tool. It includes a "Learn more about Copilot" button.

At the bottom right, there is a "Latest changes" section with a list of recent updates.

# Create a Remote Repository on GitHub

- Click "New" / "Create repository" button



- Set your Repository
  - Repository Name
  - Public vs Private
  - Add .gitignore
    - Set according to your working environment
    - C++, visualstudio, python, etc...
  - Select license

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* Snu-ProgMeth ▾ Repository name \*

Great repository names are short and memorable. Need inspiration? How about [didactic-waffle?](#)

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)  
.gitignore template: **None ▾**

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)  
License: **None ▾**

① You are creating a public repository in your personal account.

[Create repository](#)

# Github Basically Tells You What to Do

Quick setup — if you've done this kind of thing before

 Set up in Desktop or

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# GitRemoteExample" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Snu-ProgMeth/GitRemoteExample.git
git push -u origin main
```


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

```
git remote add origin https://github.com/Snu-ProgMeth/GitRemoteExample.git
git branch -M main
git push -u origin main
```


...or import code from another repository




You can initialize this repository with code from a Subversion, Mercurial, or TFS project.


# Create New File on Remote

 Snu-ProgMeth / GitRemoteExample Public Pin Unpin

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

 master had recent pushes 5 minutes ago [Compare & pull request](#)


 main ▾  2 branches  0 tags [Go to file](#) [Add file ▾](#) [Code ▾](#)

 Snu-ProgMeth Create .gitignore 20 1 commit

[Create new file](#) [Upload files](#)

GitRemoteExample / .gitignore in main

Want to use a .gitignore template? [Choose .gitignore: VisualStudio ▾](#)

 **Commit new file**

Create .gitignore

Add an optional extended description...

☒ Commit directly to the main branch.

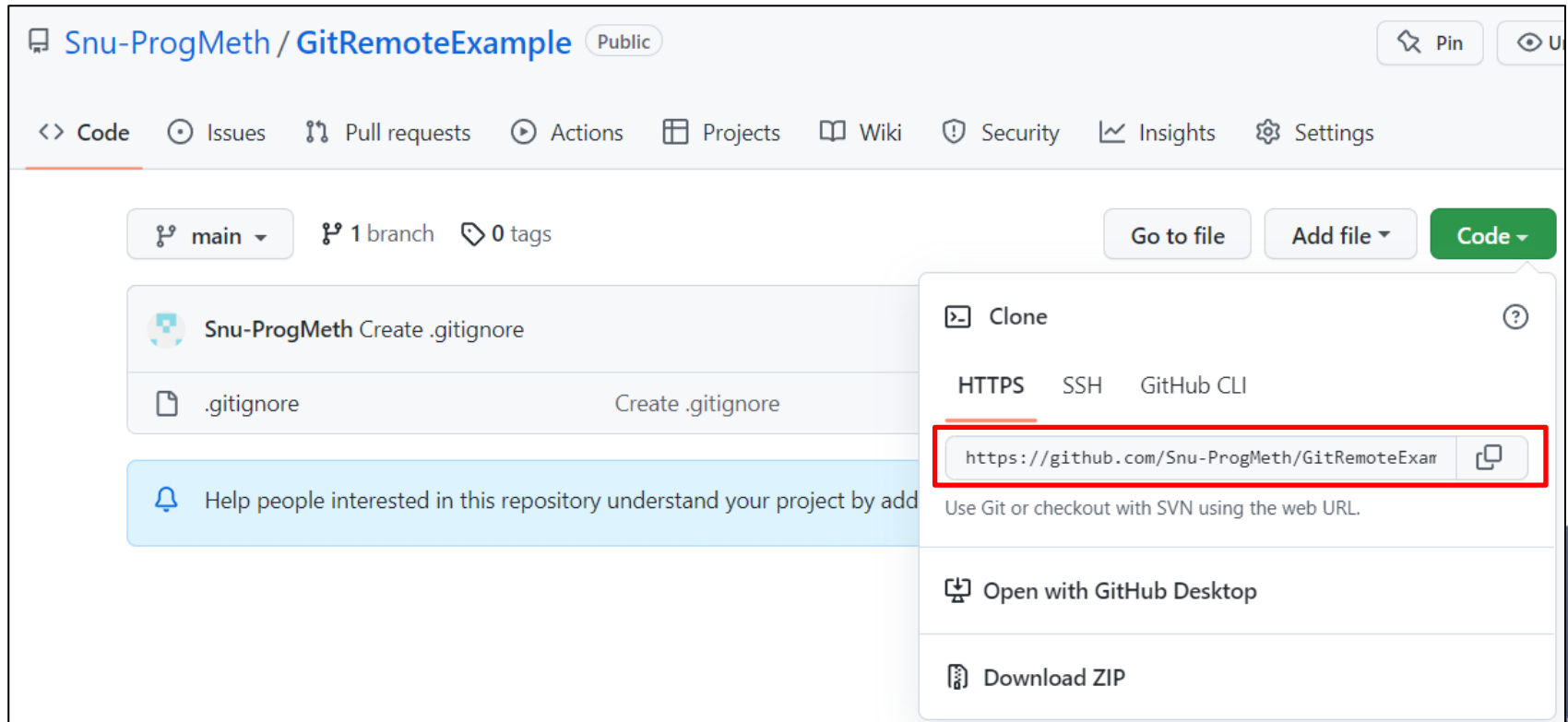
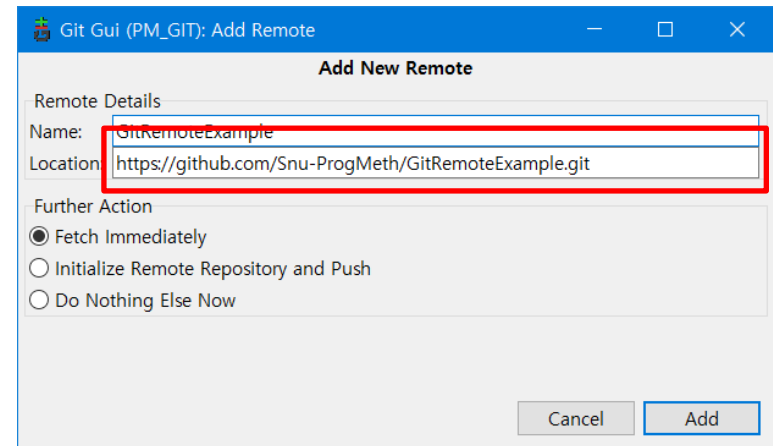
☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

[Commit new file](#) [Cancel](#)



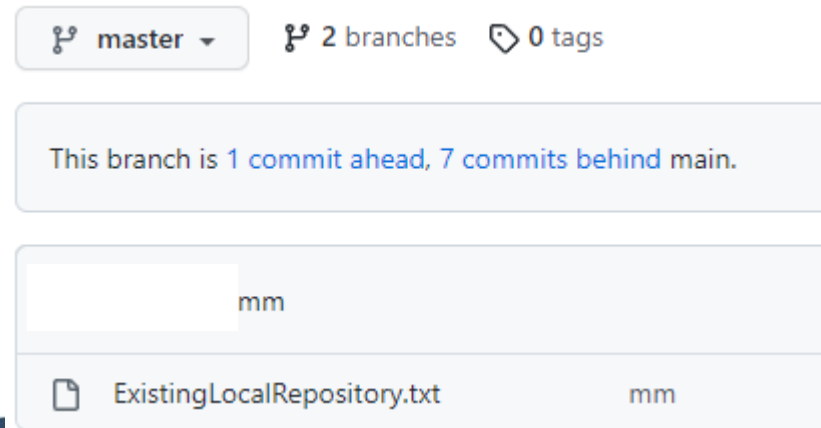
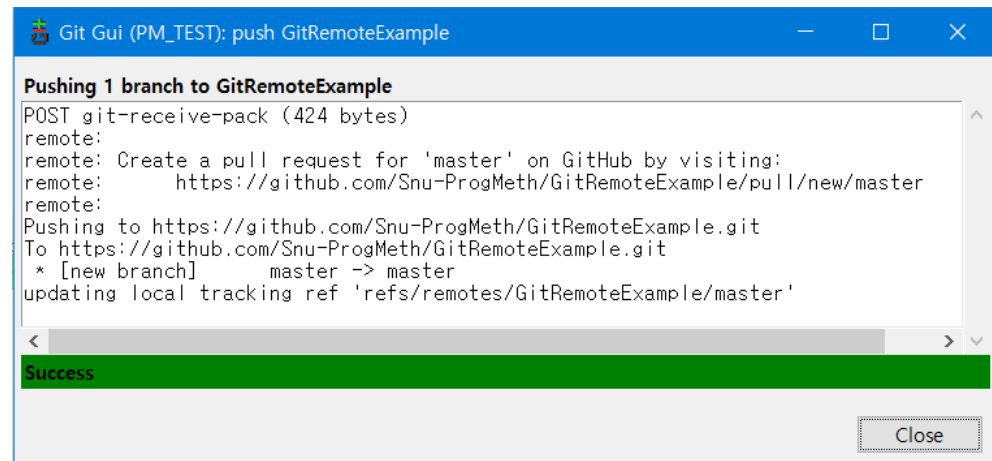
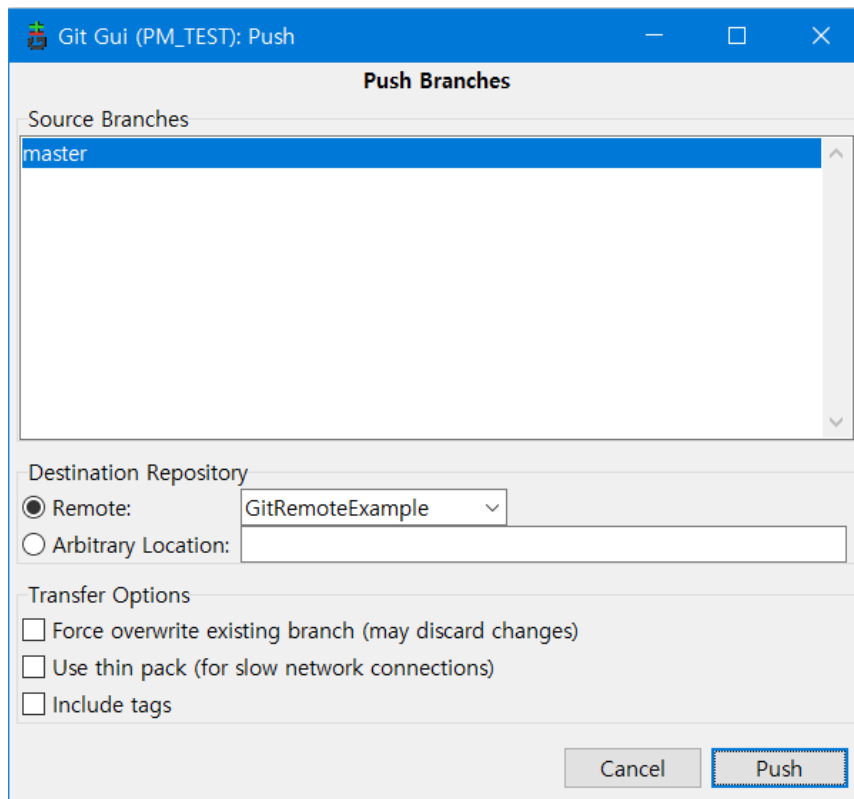
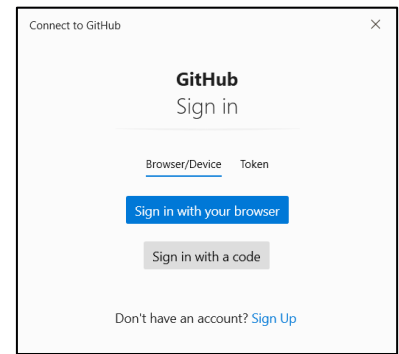
# Add Existing Local Repository

- Git GUI "Remote">"Add"



# Pushing to Remote Repository

- Git GUI "Remote"->"Push"



# Logout Github

- When using git gui, GitHub login information save automatically within windows
- 제어판>사용자 계정>자격 증명 관리자
  - Windows 자격증명>일반 자격증명

The screenshot shows the Windows Control Panel with the 'User Accounts' window open. The 'Credential Manager' link is highlighted under the 'User Accounts' section. The 'Credential Manager' window is also open, showing the 'Windows Credentials' tab. The 'Windows Credentials' list is empty, with a message stating 'Windows 자격 증명이 없습니다.' (There are no Windows credentials).

**컴퓨터 설정 변경** 보기 기준: 1

- 시스템 및 보안  
컴퓨터 상태 검토  
파일 히스토리로 파일의 백업 복사본 저장  
백업 및 복원(Windows 7)
- 네트워크 및 인터넷  
네트워크 상태 및 작업 보기
- 하드웨어 및 소리  
장치 및 프린터 보기  
장치 추가
- 프로그램  
프로그램 제거
- 사용자 계정  
계정 유형 변경 | 사용자 계정 제거
- 자격 증명 관리자  
웹 자격 증명 관리 | Windows 자격 증명 관리
- Mail (Microsoft Outlook 2016)

**자격 증명 관리**

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그인 정보를 보고 삭제합니다.

**웹 자격 증명**

자격 증명 백업(B) | 자격 증명 복원(R)

Windows 자격 증명 [Windows 자격 증명 추가](#)

Windows 자격 증명이 없습니다.

인증서 기반 자격 증명 [인증서 기반 자격 증명 추가](#)

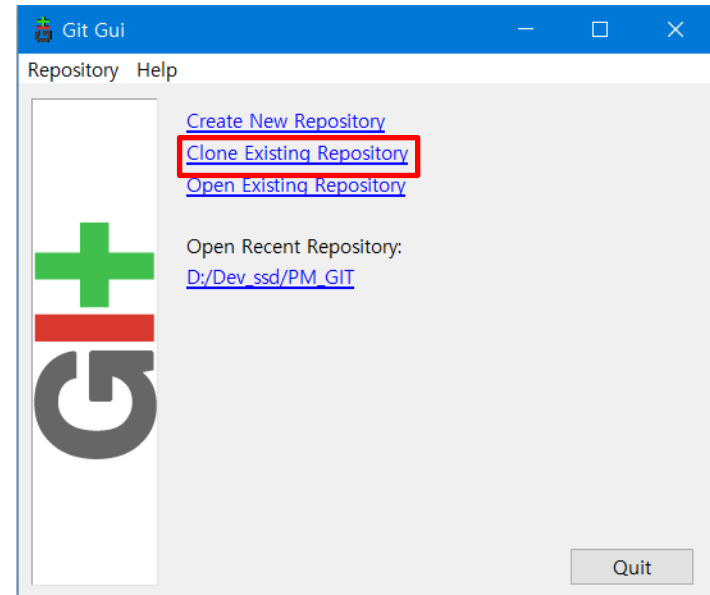
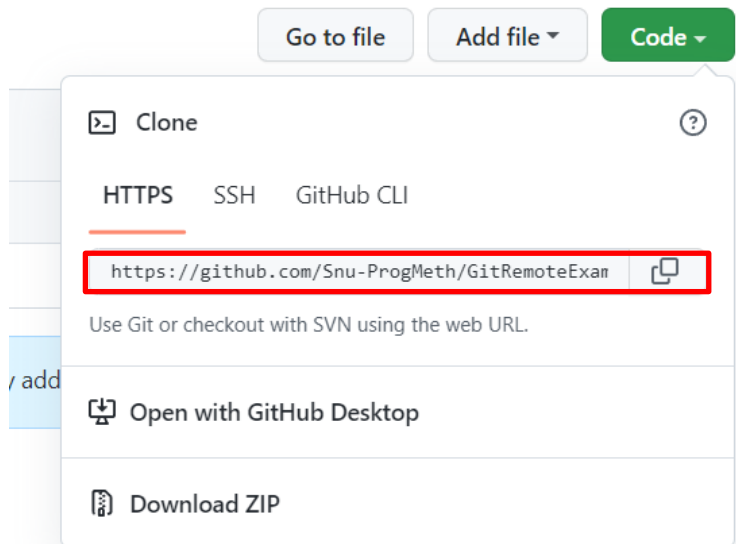
인증서가 없습니다.

일반 자격 증명 [일반 자격 증명 추가](#)

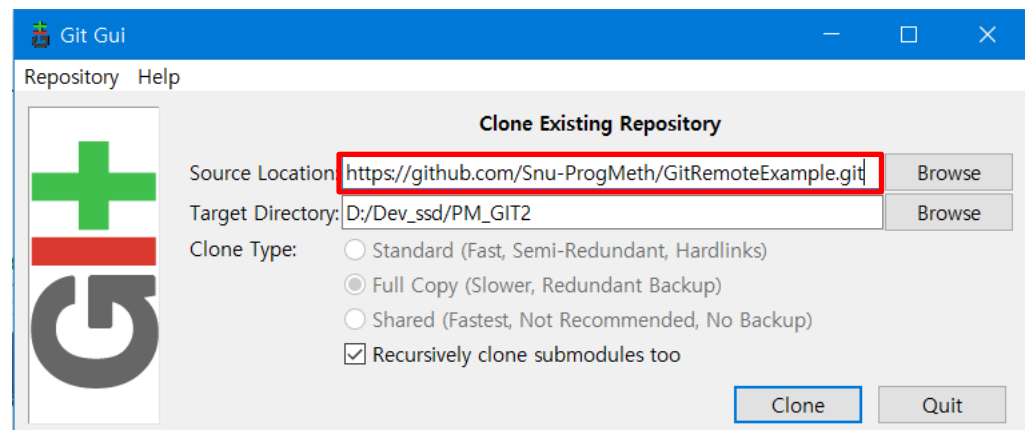
MicrosoftOffice16_Data:SSPI:handdgi@snu.ac.kr	수정한 날짜: 2019-09-04
Adobe User Info(Part2)	수정한 날짜: 2019-09-05
git:https://github.com	수정한 날짜: 오늘
MicrosoftAccount:user=handdgi@gmail.com	수정한 날짜: 오늘

# Clone Remote Repository

- Create local copy on your computer from existing remote repository

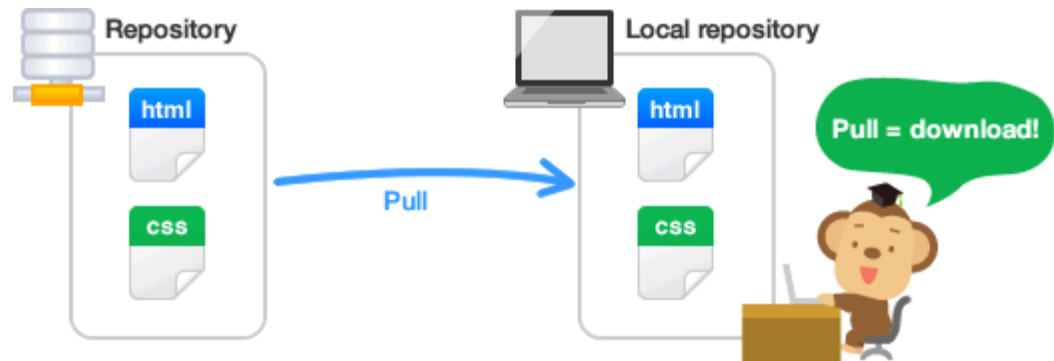
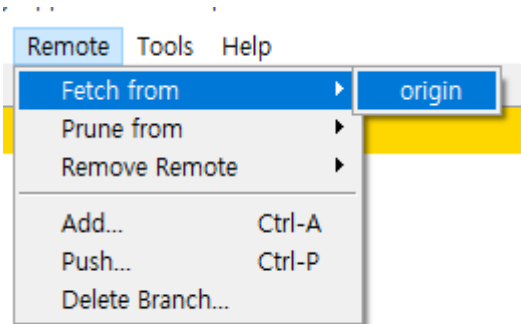


- Target directory should not exist at the time of cloning



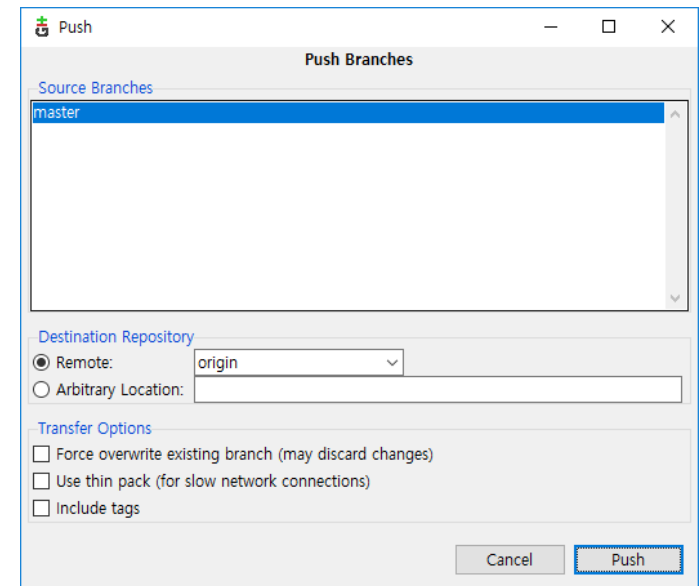
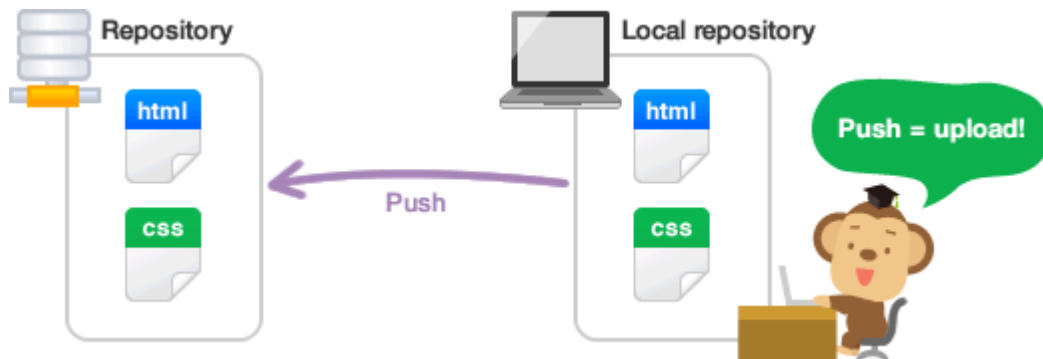
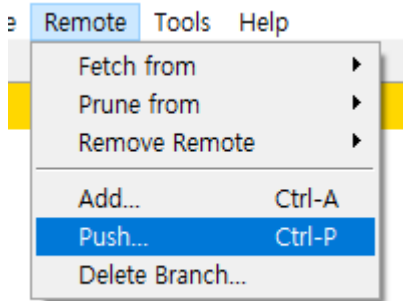
# Working with Remotes Pull

- If your local repository has been set with or cloned from remote, keep up to date with the remote using following operations
- Fetch
  - Download the latest contents from remote
  - Fetched changes should be merged using merge menu
  - On CLI, pull command
    - Fetch + merge



# Working with Remotes Push

- Push command
  - Update remote repository with local one
  - Only users have access to the remote can push their modifications into the remote



# Collaborate with Github

- Setting menu
  - Collaborators
  - Collaborators should sign up for GitHub

The screenshot shows the GitHub interface for the repository 'handdgi / 20191108'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings' (which is selected). On the left sidebar, there are links for 'Options', 'Collaborators' (highlighted), 'Branches', 'Webhooks', 'Notifications', 'Integrations & services', and 'Deploy keys'. The main content area is titled 'Collaborators' and includes a link 'Push access to the repository'. It contains a message: 'This repository doesn't have any collaborators yet. Use the form below to add a collaborator.' Below this is a search section titled 'Search by username, full name or email address' with a note: 'You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.' There is a text input field and an 'Add collaborator' button.

handdgi / 20191108

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Options

**Collaborators**

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Collaborators Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

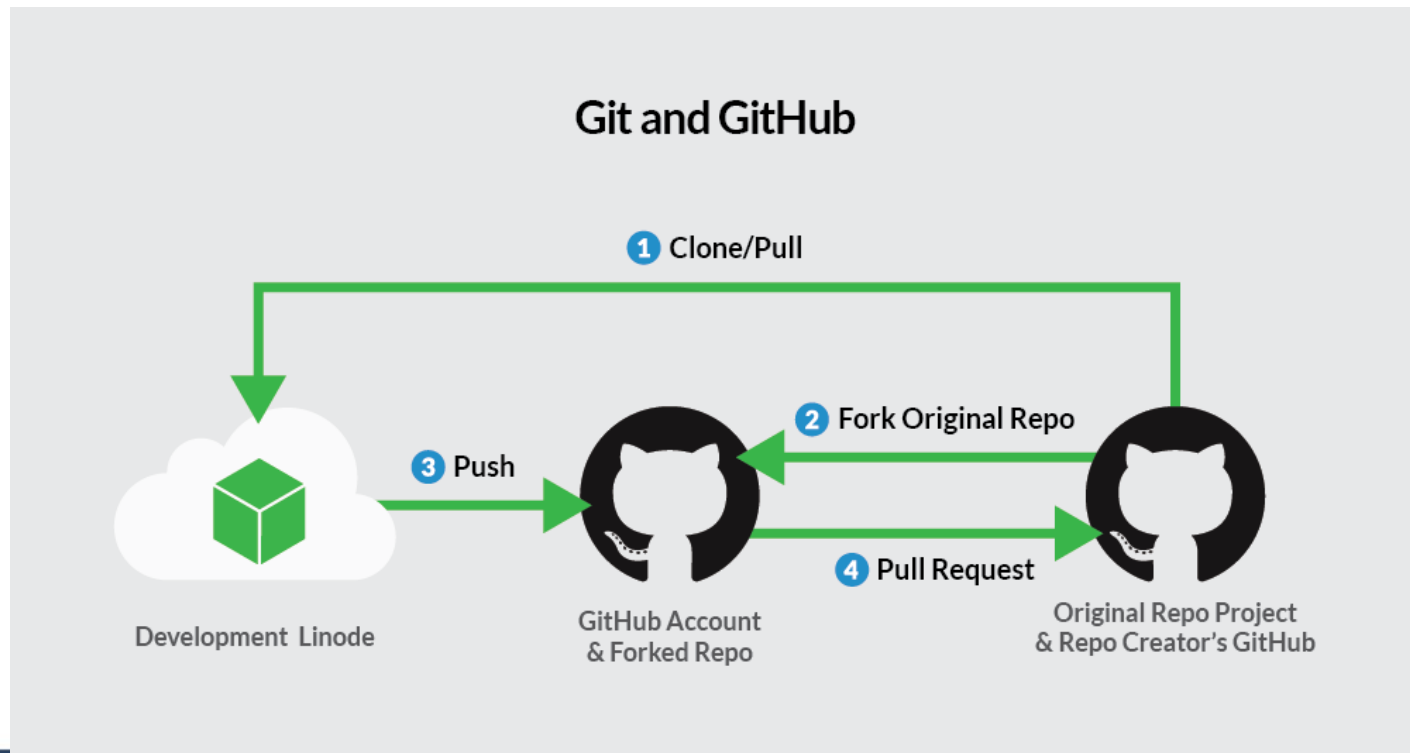
Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

# Contributing to a GitHub Project

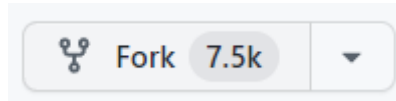
- Unless you are working as one of the core developers in a project (able to push to the repo directly), participating in a GitHub project starts from **forking** (copying) the repo to your account
- Create a topic branch in the forked repo, and make changes
- If modifications done, a **pull request** could be issued to merge your branch into the original project



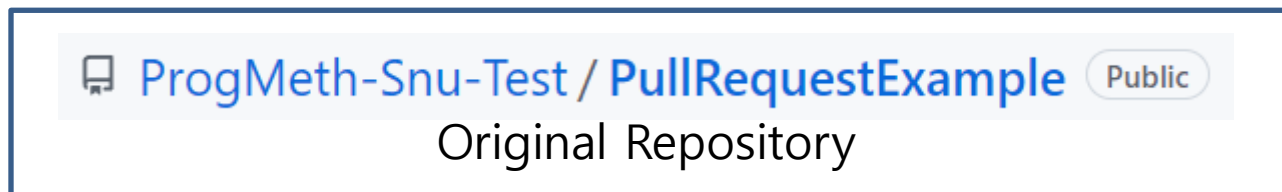


# Forking a project

- Forking a project
  - Clicking 'Fork' button on GitHub page will fork the project into your account

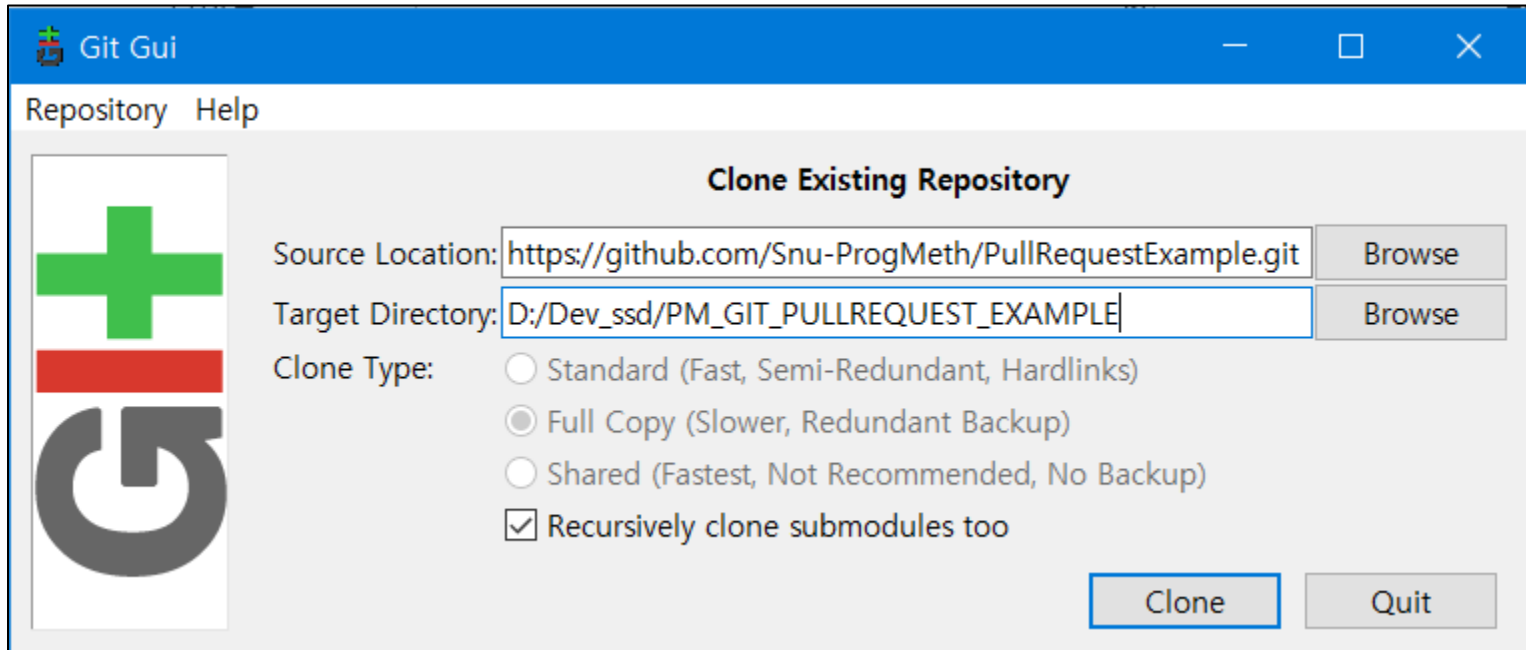


- Forked project is indicated with a fork icon



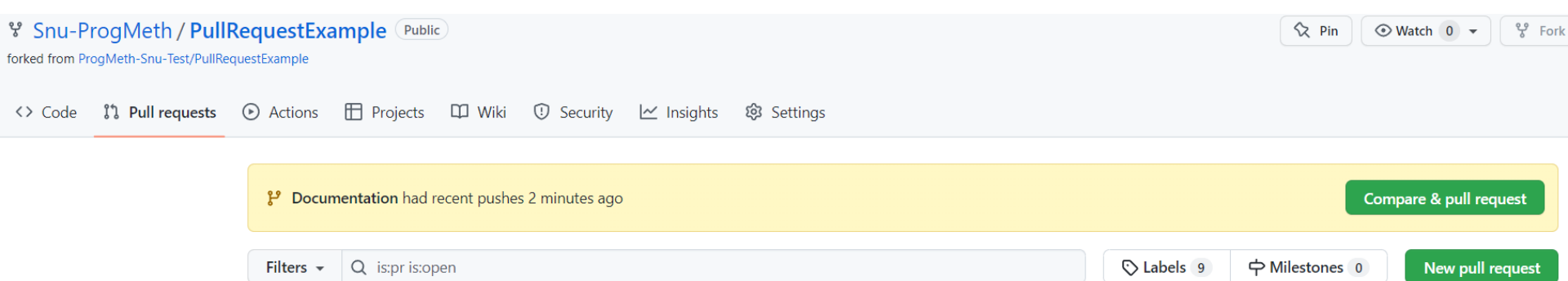
# Contributing to a GitHub Project

- Clone the forked project



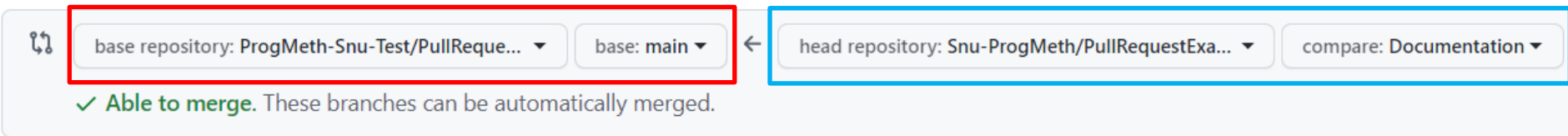
# Contributing to a GitHub Project

- Make changes and push
  - First create a topic branch which you will work in
  - Modify the code as you want, then commit and push to 'forked' repo
- Create a pull request
  - Go to the forked project in GitHub
  - Either
    - Click 'Compare & pull request'
    - Click 'Pull requests' > 'New pull request'



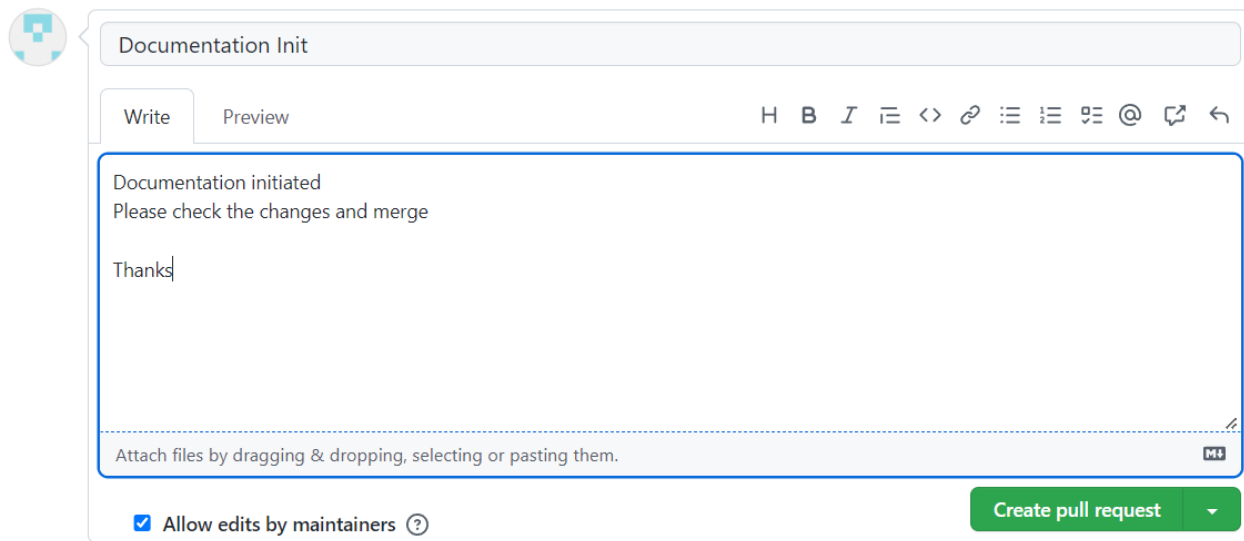
# Contributing to a Github Project

- Create a pull request
  - Make sure that your topic branch is merging into original's branch



The screenshot shows the GitHub pull request creation interface. It features two main dropdown menus for repository selection, separated by a left-pointing arrow. The left dropdown is labeled 'base repository: ProgMeth-Snu-Test/PullReque...' and has a sub-dropdown 'base: main'. The right dropdown is labeled 'head repository: Snu-ProgMeth/PullRequestExa...' and has a sub-dropdown 'compare: Documentation'. A green checkmark and the text 'Able to merge. These branches can be automatically merged.' are displayed below the dropdowns.

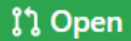
- Add message and click 'Create pull request'
  - Make sure that your topic branch is merging into original's branch



The screenshot shows the GitHub pull request message editor. It has a title bar 'Documentation Init' and two tabs: 'Write' and 'Preview'. The 'Write' tab is active, showing a text area with the message 'Documentation initiated' and 'Please check the changes and merge'. Below this, the word 'Thanks' is typed. At the bottom, there is a checkbox labeled 'Allow edits by maintainers' which is checked, and a green button labeled 'Create pull request'.

# Contributing to a Github Project

## Documentation Init #1



Snu-ProgMeth wants to merge 1 commit into `ProgMeth-Snu-Test:main` from `Snu-ProgMeth:Documentation`



Conversation 0



Commits 1



Checks 0



Files changed 1



Snu-ProgMeth commented 2 minutes ago

First-time contributor



Documentation initiated  
Please check the changes and merge  
  
Thanks



**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



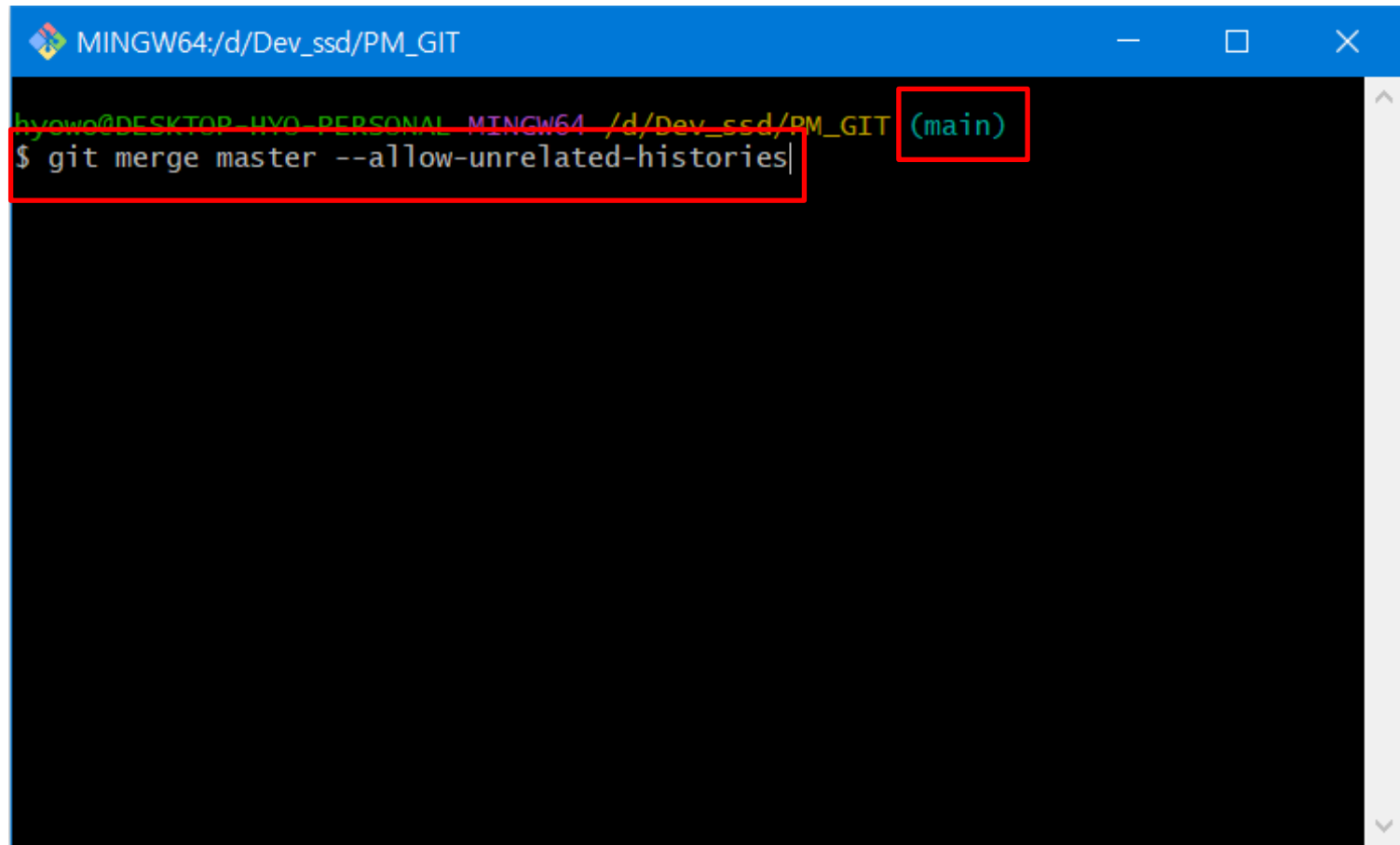
You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

# Mission

- Fork repo,  
[https://github.com/Snu-ProgMeth/22Fall\\_PM\\_L2\\_GIT\\_REMOTE.git](https://github.com/Snu-ProgMeth/22Fall_PM_L2_GIT_REMOTE.git)
- Clone your forked repo into your local workspace
- Create a cpp file with a name like this
  - 'Lab2-YOUR\_STUDENT\_NUMBER'
  - E.g. 'Lab2-2022-12345'
- Commit and push to the forked repo
- Create a pull request into the original repo submission branch

# --allow-unrelated-histories

- Force merge unrelated branches (for instance by adding remote)



A screenshot of a Windows terminal window titled "MINGW64:/d/Dev\_ssd/PM\_GIT". The terminal shows a prompt "hyowe@DESKTOP-HYO-PERSONAL MINGW64 /d/Dev\_ssd/PM\_GIT (main)" where the command "git merge master --allow-unrelated-histories" is being entered. The command and the "(main)" branch indicator are highlighted with red boxes. The terminal has a blue title bar and standard window controls.

```
MINGW64:/d/Dev_ssd/PM_GIT
hyowe@DESKTOP-HYO-PERSONAL MINGW64 /d/Dev_ssd/PM_GIT (main)
$ git merge master --allow-unrelated-histories
```