Lecture Note 4.
GitHub

April 02, 2025

Kwanghee Lee
Dept. of Software
Dankook University

kh-lee@dankook.ac.kr

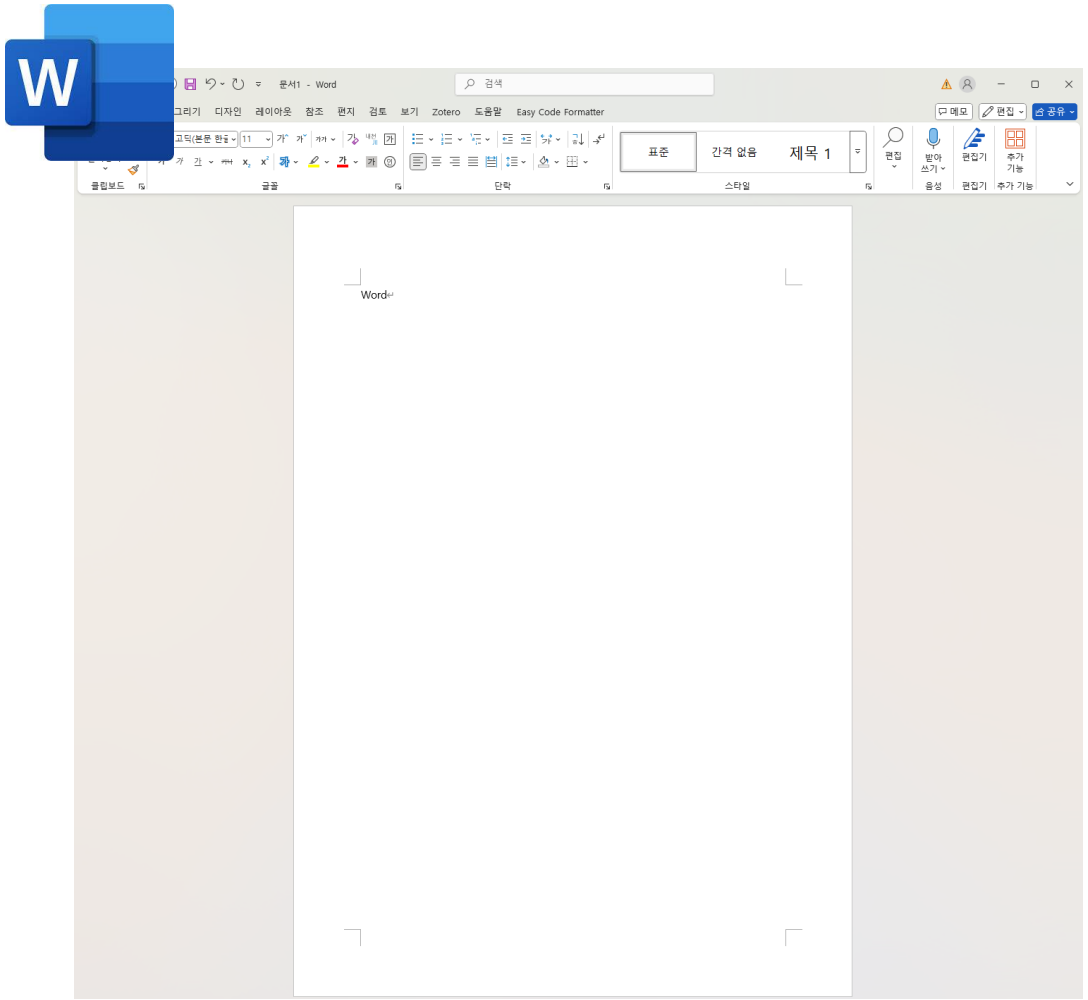
(본 교재는 2025년도 과학기술정보통신부 및 정보통신기획평가원의 'SW중심대학사업' 지원을 받아 제작 되었습니다.)

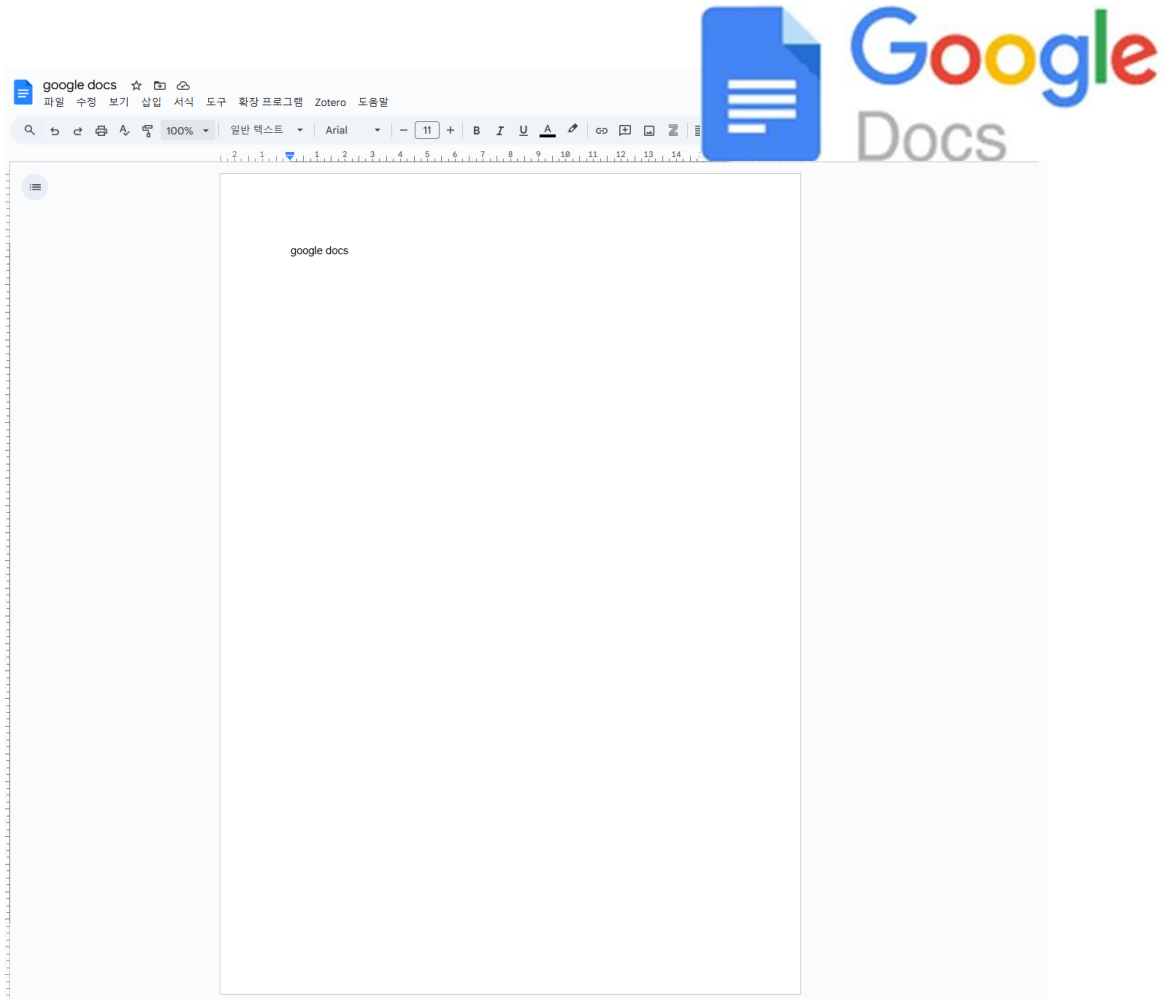
Contents

- Local Repository and Remote Repository
- GitHub
  - What is github?
  - How to use github?
- Vim
  - What is Vim?
  - How to use vim in Linux?
- Practice

Local Repository and Remote Repository (1/3)

- **Local Repository, Remote Repository**





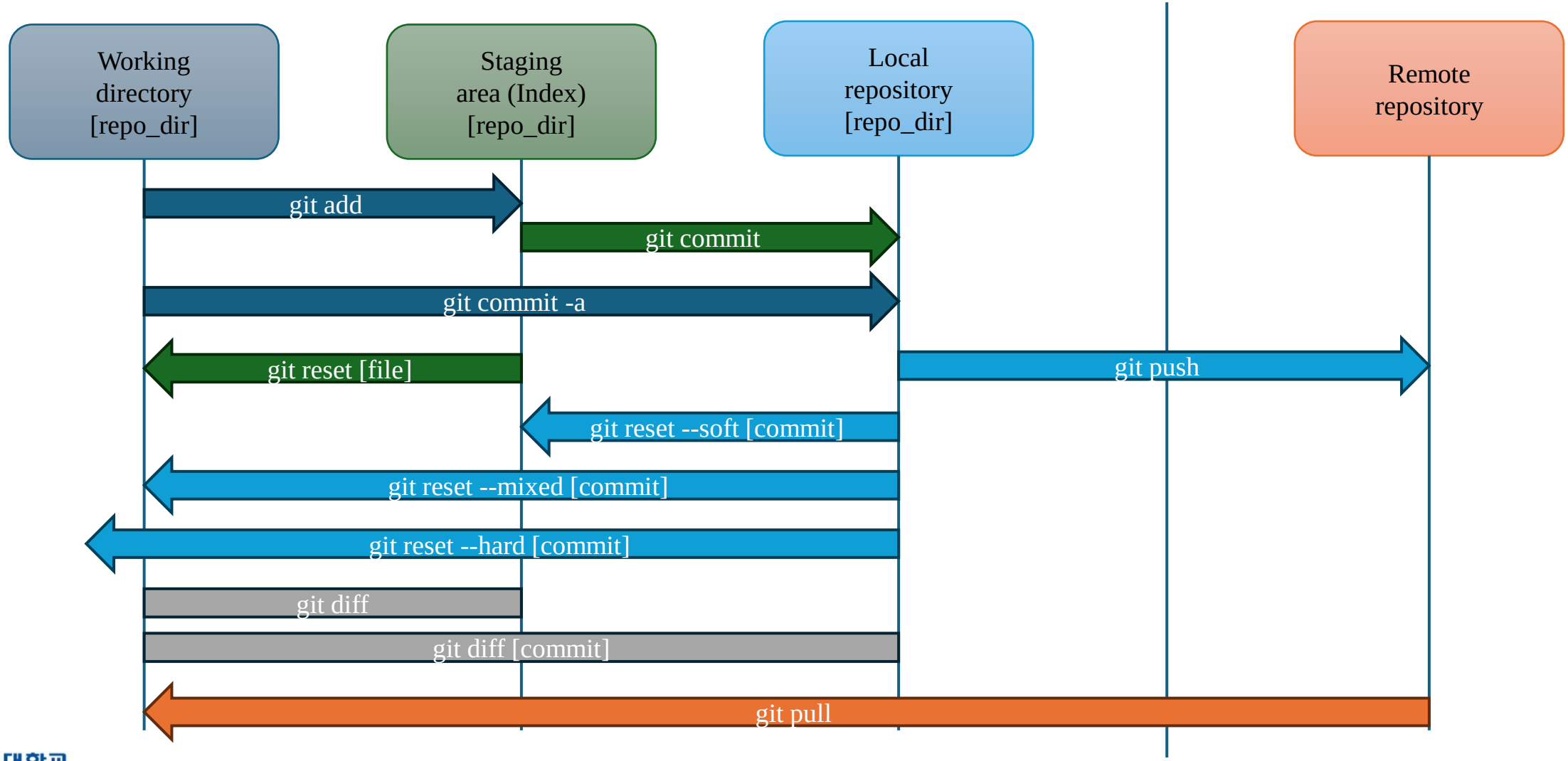
Local Repository and Remote Repository (2/3)

- **Remote Repository**
  - **SourceForge**
    - 오래된 오픈소스 프로젝트 저장소
  - **GitHub**
    - Microsoft
    - 오픈소스 커뮤니티와 협업에 강함
    - 가장 인기 많은 원격 저장소
  - **GitLab**
    - 내부 서버 설치 가능
    - 비용 대비 많은 저장용량
    - 커스터마이징 가능
    - 기업 사내 서버로 운영 적합
  - **Bitbucket**
  - **AWS CodeCommit**
  - **Azure DevOps Repos**

ref.: google image

Local Repository and Remote Repository (3/3)

- **Local Repository, Remote Repository**



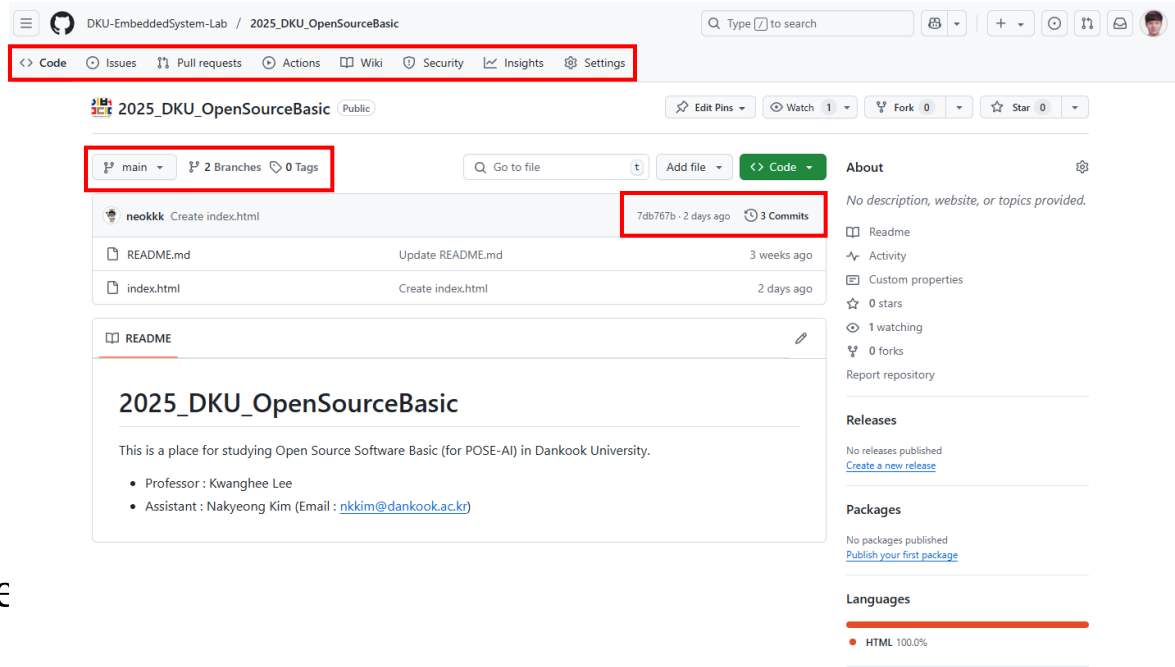
GitHub (1/20)

- **What is GitHub?**
  - A service for managing and sharing Git repositories on the internet.
  - Acquired by Microsoft on June 4, 2018.

- **Why use GitHub?**
  - Collaboration
    - Multiple people can work on the same codebase simultaneously.
  - Backup
    - Code is safely stored on the internet.
  - History
    - Track who changed what and when.
  - Testing
    - Automatically run tests with GitHub Actions.
  - Sharing
    - Great for contributing to open-source projects or showcasing your work.

GitHub (2/20)

- **What is GitHub?**
  - Code
    - View the actual code in the repository
  - Pull requests
    - View and manage requests to merge code into branches
  - Actions
    - Set up and manage CI/CD workflows and automation
  - Issues
    - Track bugs, feature requests, and other tasks
  - Wiki
    - Host documentation for the project
  - Insights
    - Analyze the project's activity and contributions
  - Commits
    - View the commit history for changes made to the code
  - Branches
    - View the list of branches in the repository

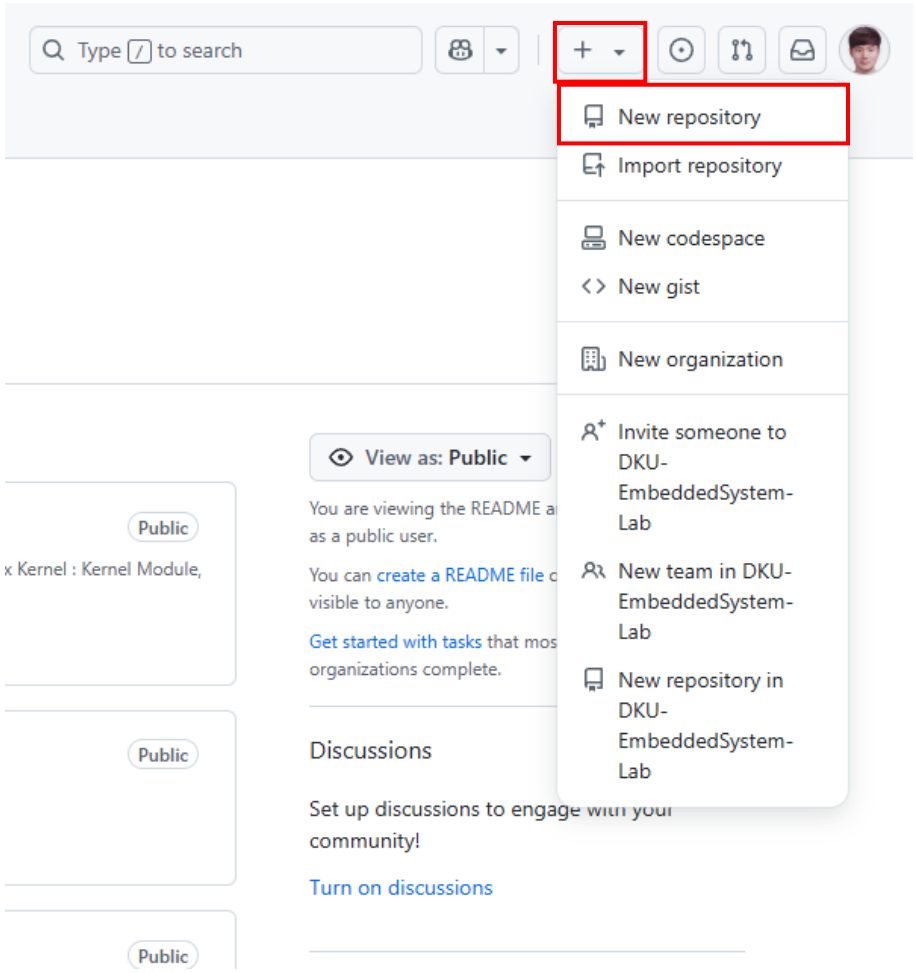


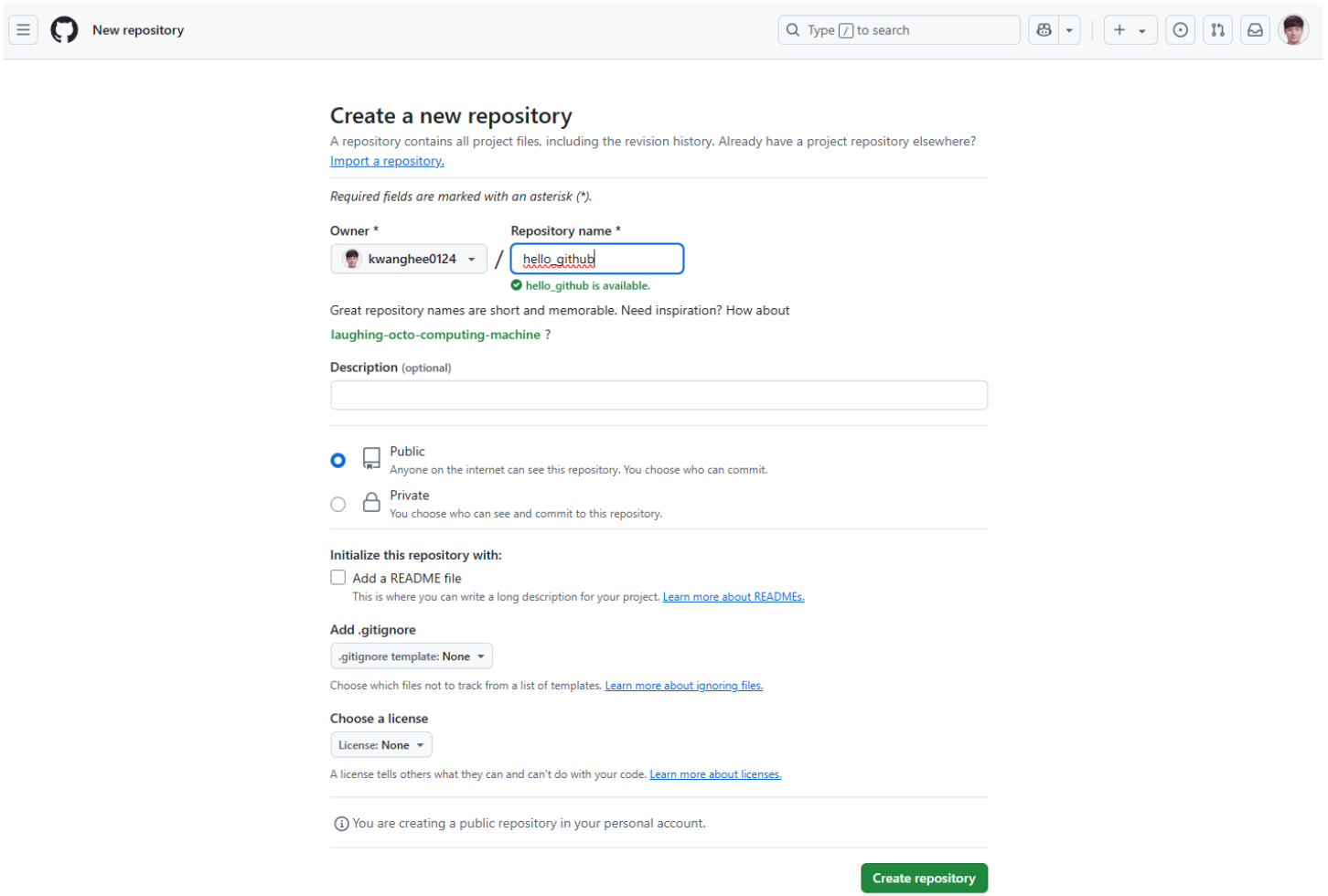
GitHub – How to use GitHub? (3/20)

- **Git Remote Commands**
  - git remote: View the list of registered remote repositories
    - -v: View detailed URLs of remote repositories (distinguishes fetch/push)
    - add [name] [url]: Add a new remote repository (commonly named origin)
  - git clone: Clone a remote repository to the local machine.
  - git fetch
    - [remote]: Fetch changes from the remote repository without automatically merging them.
  - git pull
    - [remote] [branch]: Automatically fetch and merge changes from the remote repository in one step.
  - git push
    - [remote] [branch]: Upload a local branch to the remote repository.

GitHub – How to use GitHub? (4/20)

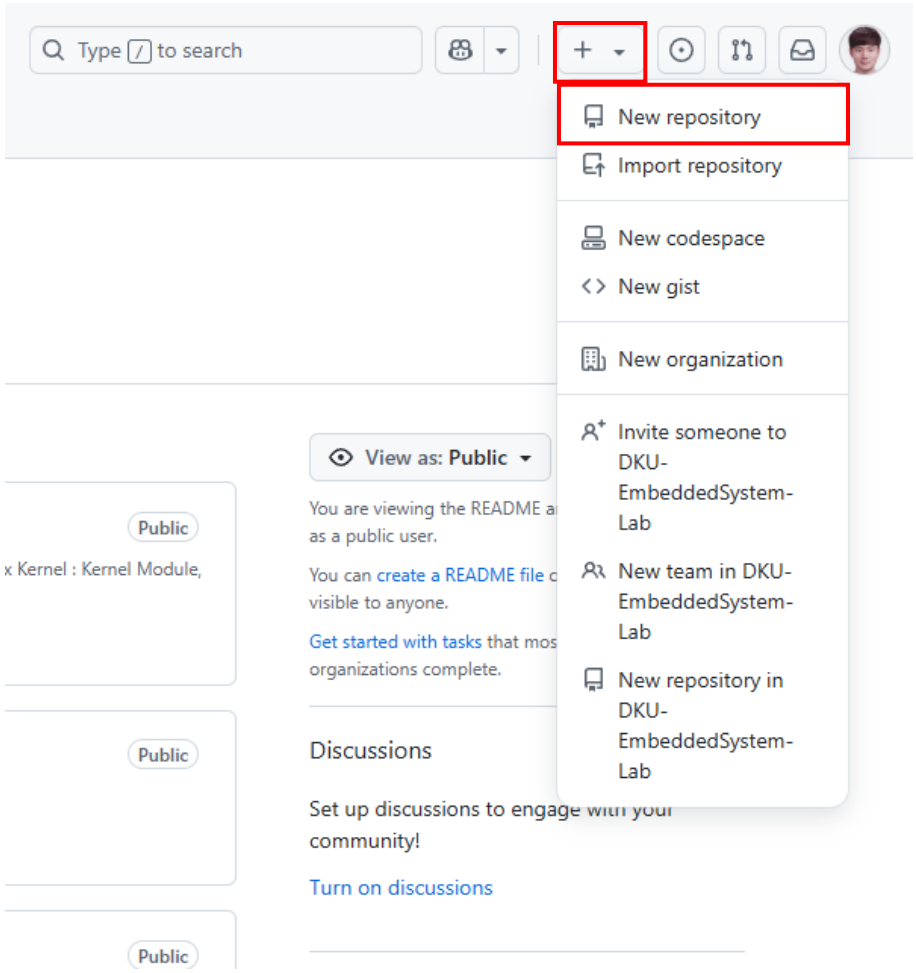
- **Make Remote Repository**

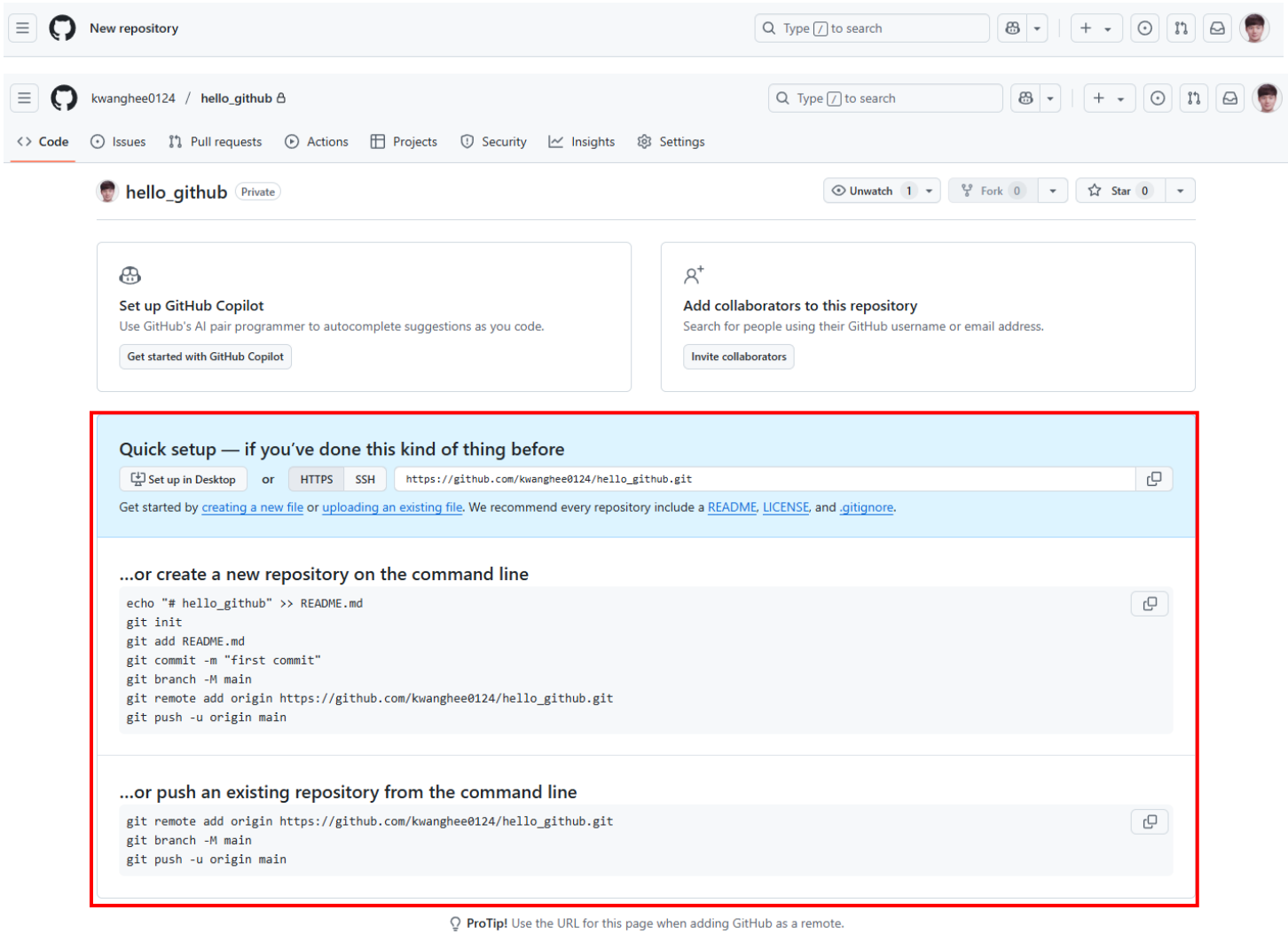




GitHub – How to use GitHub? (5/20)

- **Make Remote Repository**





GitHub – How to use GitHub? (6/20)

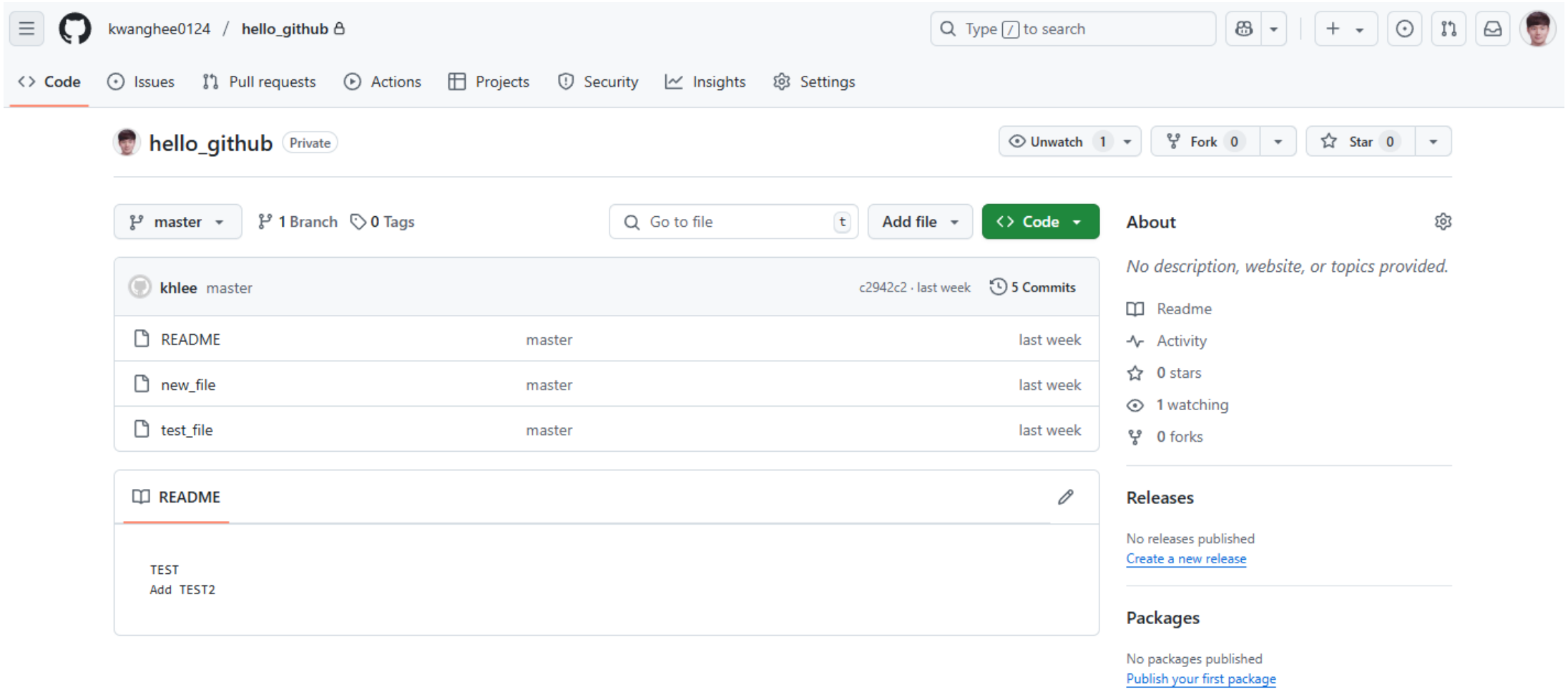
- **git remote and push**

```
~/Lecture/git/repo_dir
root# ls
new_file  README  test_file
~/Lecture/git/repo_dir
root# git remote add origin https://github.com/kwanghee0124/hello_github.git
~/Lecture/git/repo_dir
root# git branch
* master
  testing
~/Lecture/git/repo_dir
root# git branch -r
~/Lecture/git/repo_dir
root# git branch -a
* master
  testing
~/Lecture/git/repo_dir
root# git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (13/13), 1.05 KiB | 1.05 MiB/s, done.
Total 13 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/kwanghee0124/hello_github.git
 * [new branch]      master -> master
```

```
~/Lecture/git/repo_dir
root# git branch
* master
  testing
~/Lecture/git/repo_dir
root# git branch -r
  origin/master
~/Lecture/git/repo_dir
root# git branch -a
* master
  testing
  remotes/origin/master
```

GitHub – How to use GitHub? (7/20)

- **git remote and push**



GitHub – How to use GitHub? (8/20)

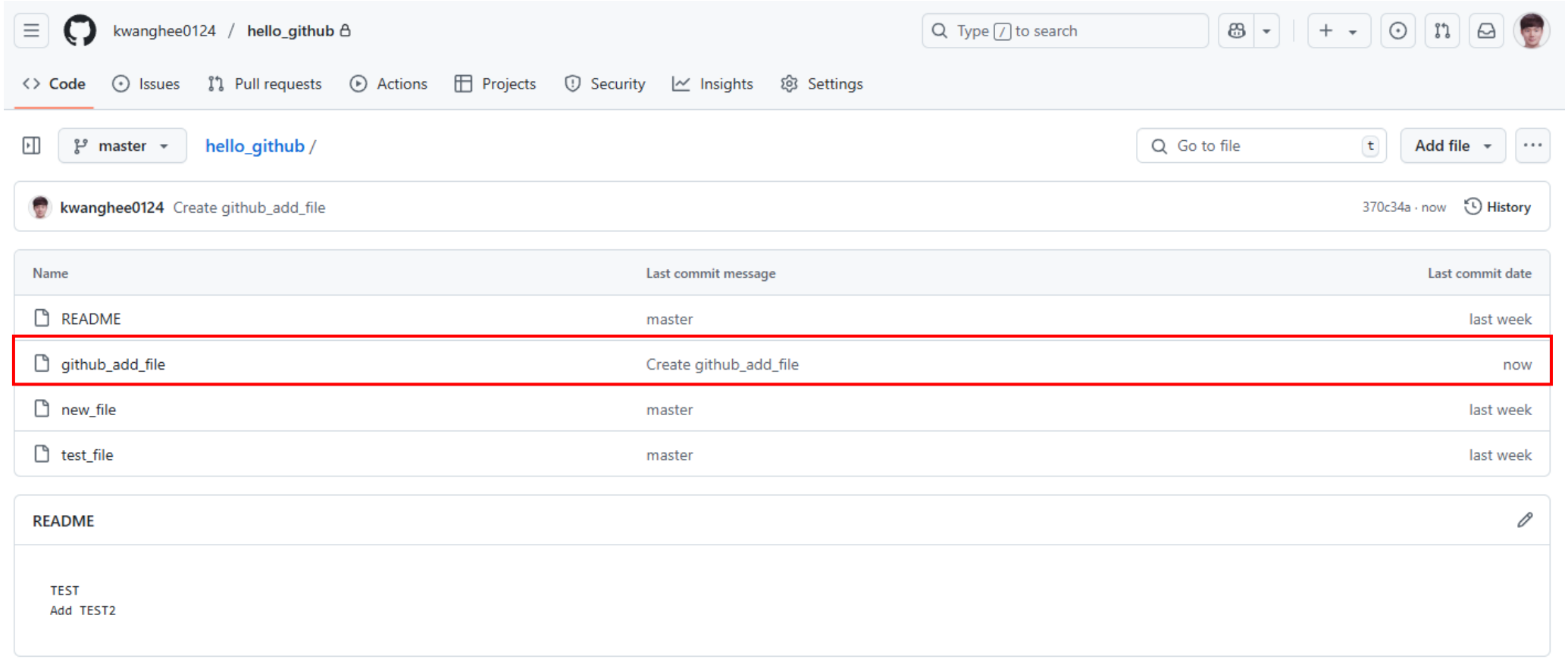
- **git push option**
  - -u: Push the local branch and set it as the upstream (tracking) branch.

```
~/Lecture/git/repo_dir
root# git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master
```

```
~/Lecture/git/repo_dir
root# git push -u origin master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Everything up-to-date
~/Lecture/git/repo_dir
root# git push
Everything up-to-date
```

GitHub – How to use GitHub? (9/20)

- **git pull**



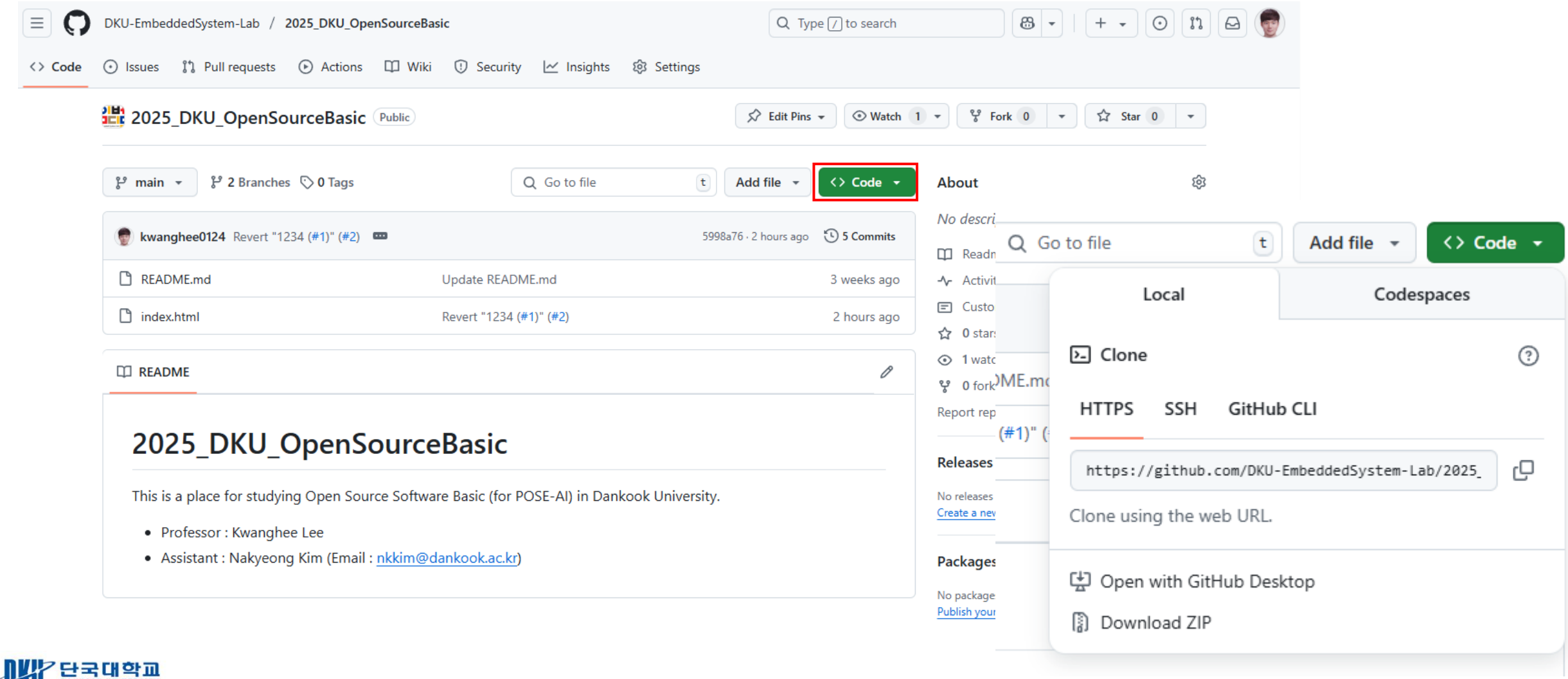
GitHub – How to use GitHub? (10/20)

- **git pull**

```
~/Lecture/git/repo_dir
root# ls
new_file  README  test_file
~/Lecture/git/repo_dir
root# git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1000 bytes | 1000.00 KiB/s, done.
From https://github.com/kwanghee0124/hello_github
 * branch            master     -> FETCH_HEAD
   489f141..1482cba  master     -> origin/master
Updating 489f141..1482cba
Fast-forward
 github_add_file | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 github_add_file
~/Lecture/git/repo_dir
root# ls
github_add_file  new_file  README  test_file
```

GitHub – How to use GitHub? (11/20)

- **git clone**



GitHub – How to use GitHub? (12/20)

- **git clone**

```
~/Lecture/git
root# ls
repo_dir
~/Lecture/git
root# git clone https://github.com/DKU-EmbeddedSystem-Lab/2025_DKU_OpenSourceBasic.git
Cloning into '2025_DKU_OpenSourceBasic'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 16 (delta 2), reused 3 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (16/16), 6.30 KiB | 6.30 MiB/s, done.
Resolving deltas: 100% (2/2), done.
~/Lecture/git
root# ls
2025_DKU_OpenSourceBasic  repo_dir
~/Lecture/git
root# cd 2025_DKU_OpenSourceBasic/
~/Lecture/git/2025_DKU_OpenSourceBasic
root# ls
index.html  README.md
```

GitHub – How to use GitHub? (13/20)

- **git merge**

```
~/Lecture/git/repo_dir
root# ls
new_file  README  test_file
~/Lecture/git/repo_dir
root# git merge testing
Auto-merging test_file
CONFLICT (content): Merge conflict in test_file
Automatic merge failed; fix conflicts and then commit the result.
```

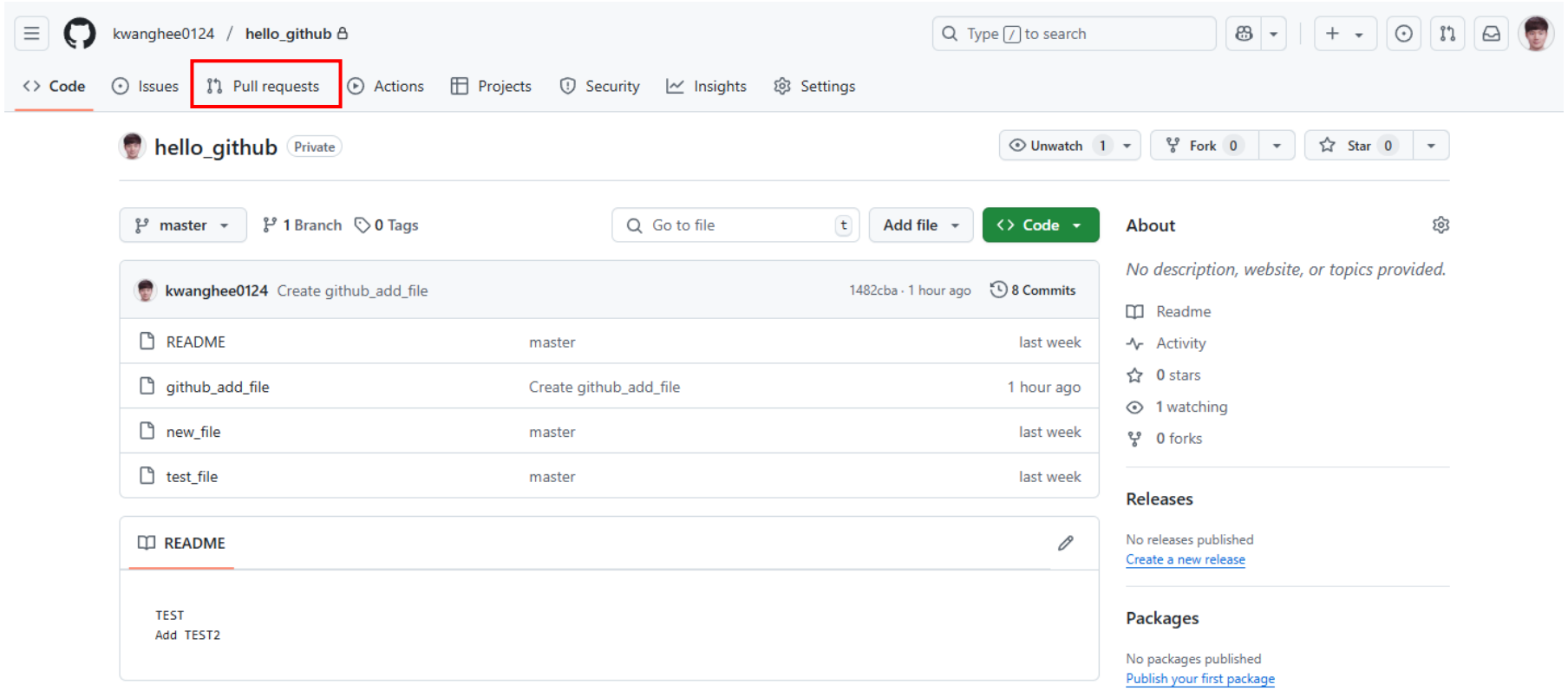
```
test_file ●
 5 <<<<<<< HEAD
 4 Master branch!
 3 =======
 2 Testing branch!
 1 >>>>>>> testing
```

```
test_file ●
 3 Master branch!
```

```
~/Lecture/git/repo_dir
root# git add test_file
~/Lecture/git/repo_dir
root# git commit
[master 953c851] Merge branch 'testing'
```

GitHub – How to use GitHub? (14/20)

- **Pull requests**



GitHub – How to use GitHub? (15/20)

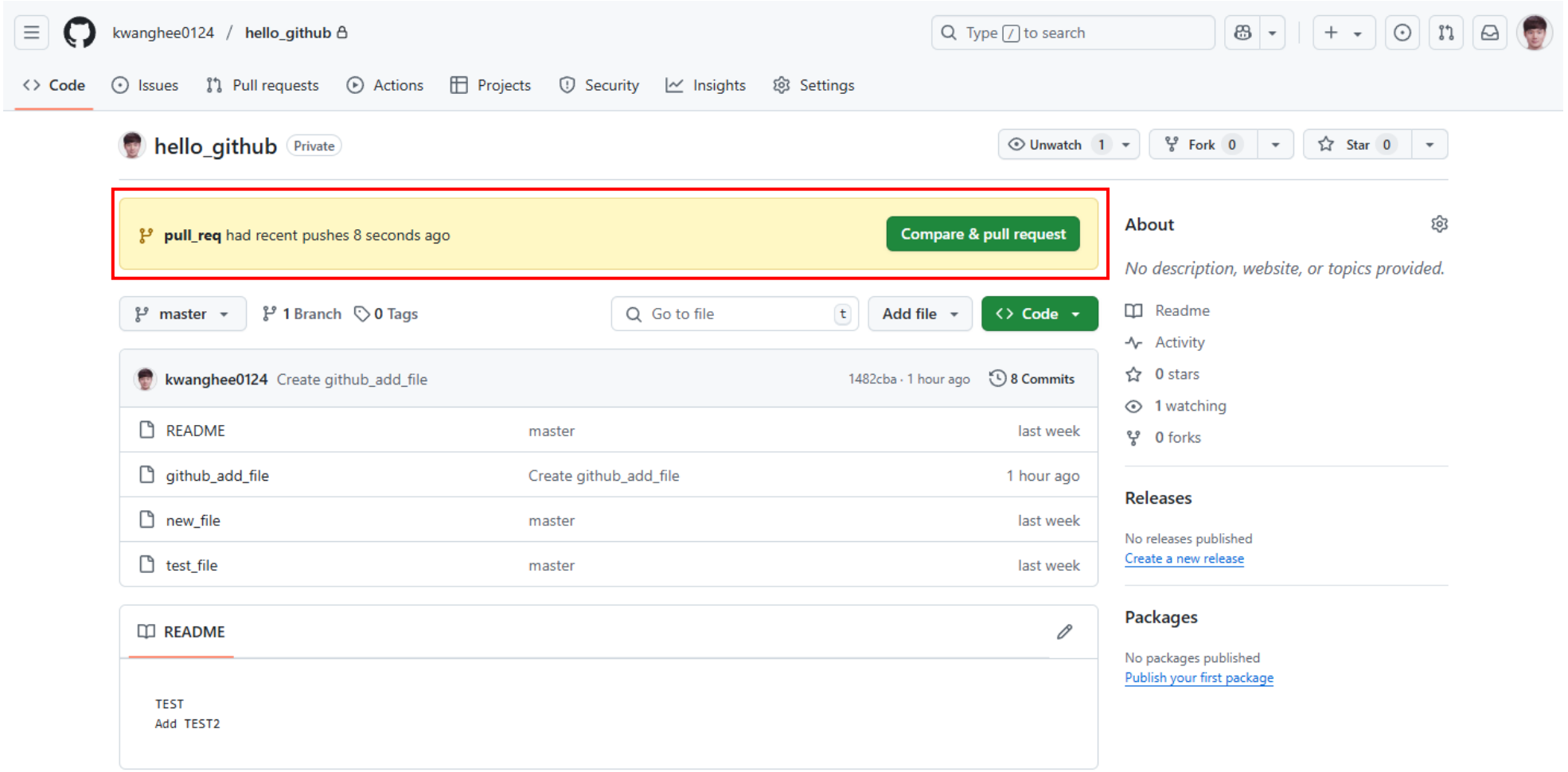
- **Pull requests**

```
~/Lecture/git/repo_dir
root# git branch
* main
  testing
~/Lecture/git/repo_dir
root# git branch pull_req
~/Lecture/git/repo_dir
root# git switch pull_req
Switched to branch 'pull_req'
~/Lecture/git/repo_dir
root# git branch
  main
* pull_req
  testing
```

```
~/Lecture/git/repo_dir
root# echo "pull request" > new_file
~/Lecture/git/repo_dir
root# ls
new_file  README  test_file
~/Lecture/git/repo_dir
root# cat new_file
pull request
~/Lecture/git/repo_dir
root# git add new_file
~/Lecture/git/repo_dir
root# git commit -m "new_file"
[pull_req 877a4d8] new_file
 1 file changed, 1 insertion(+)
~/Lecture/git/repo_dir
root# git push origin pull_req
Enumerating objects: 26, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (21/21), 1.93 KiB | 1.93 MiB/s, done.
Total 21 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
remote:
remote: Create a pull request for 'pull_req' on GitHub by visiting:
remote:      https://github.com/kwanghee0124/hello_github/pull/new/pull_req
remote:
To https://github.com/kwanghee0124/hello_github.git
 * [new branch]      pull_req -> pull_req
```

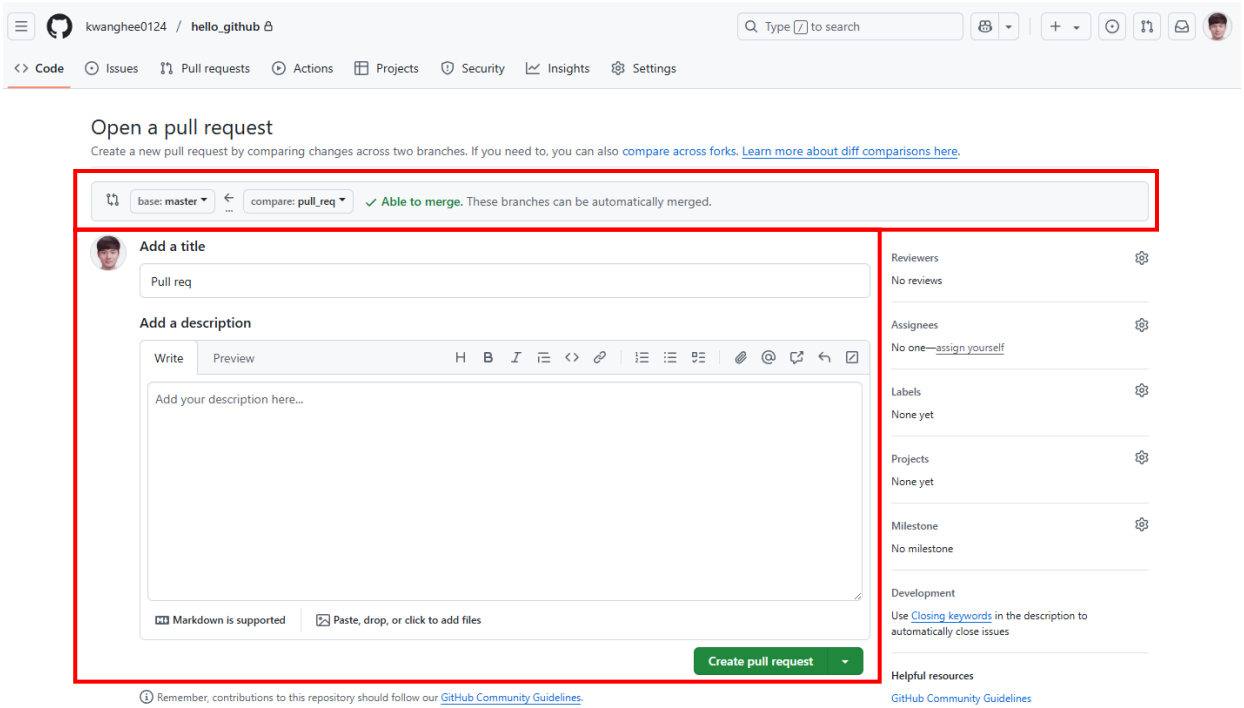
GitHub – How to use GitHub? (16/20)

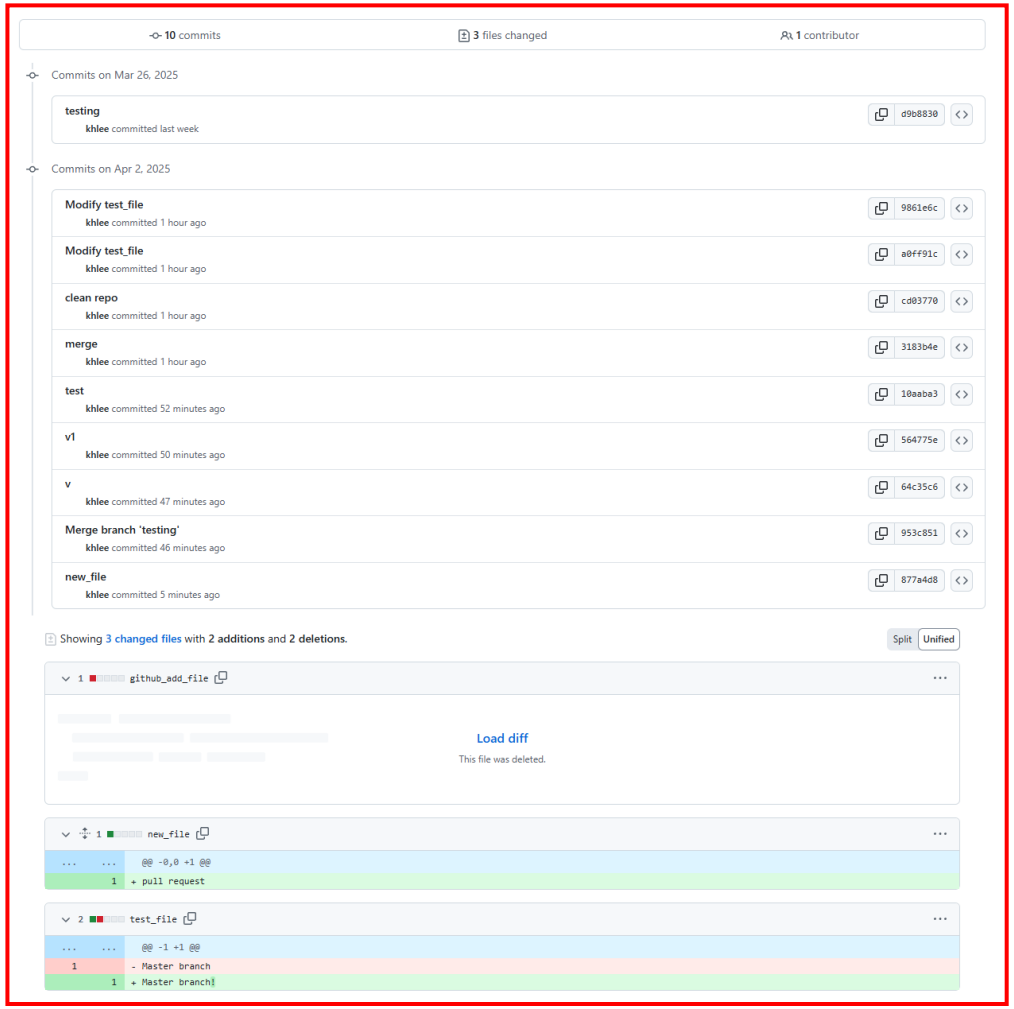
- **Pull requests**



GitHub – How to use GitHub? (17/20)

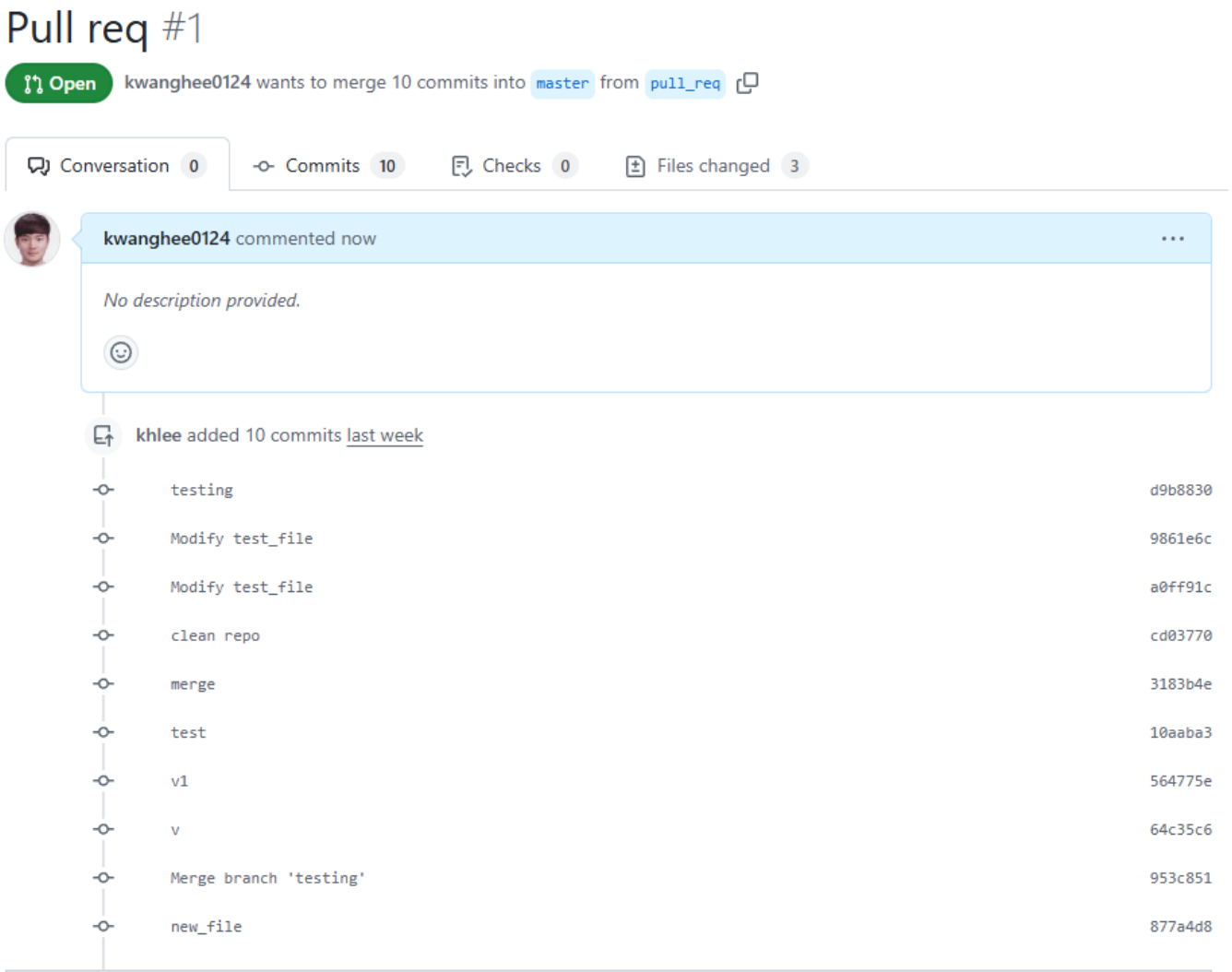
- **Pull requests**





GitHub – How to use GitHub? (18/20)

- **Pull requests**



GitHub – How to use GitHub? (19/20)

- **.gitconfig**
  - git token: ghp_VzGOGvQ[redacted]

```
~/Lecture/git/repo_dir
root# git config --global credential.helper cache
~/Lecture/git/repo_dir
root# git config --global credential.helper store
```

15분

```
[user]
        name = opensource
        email = sslab@dankook.ac.kr
[credential]
        helper = store
```

GitHub – branch name (20/20)

- **git master branch → main branch**
  - git branch -m/M [current name] [new name]
  - git branch -m/M [new name]

```
~/Lecture/git/repo_dir
root# git branch
* master
  testing
~/Lecture/git/repo_dir
root# git branch -m main
~/Lecture/git/repo_dir
root# git branch
* main
  testing
```

What is vim? (1/4)

- **Editor**
  - ed: line editor for the Unix OS from Aug. 1969

```
a
ed is the standard Unix text editor.
This is line number two.
.
2i

.
,l
ed is the standard Unix text editor.$
$
This is line number two.$
3s/two/three/
,l
ed is the standard Unix text editor.$
$
This is line number three.$
w text
65
q
```

```
ed is the standard Unix text editor.

This is line number three.
```

  - Vi(Visual) editor
    - BSD, C shell, Vi by Bill Joy
    - Developed **“vi”** by adding the **“ed”** plugin
    - Not opensource => Opensource project

What is vim? (2/4)

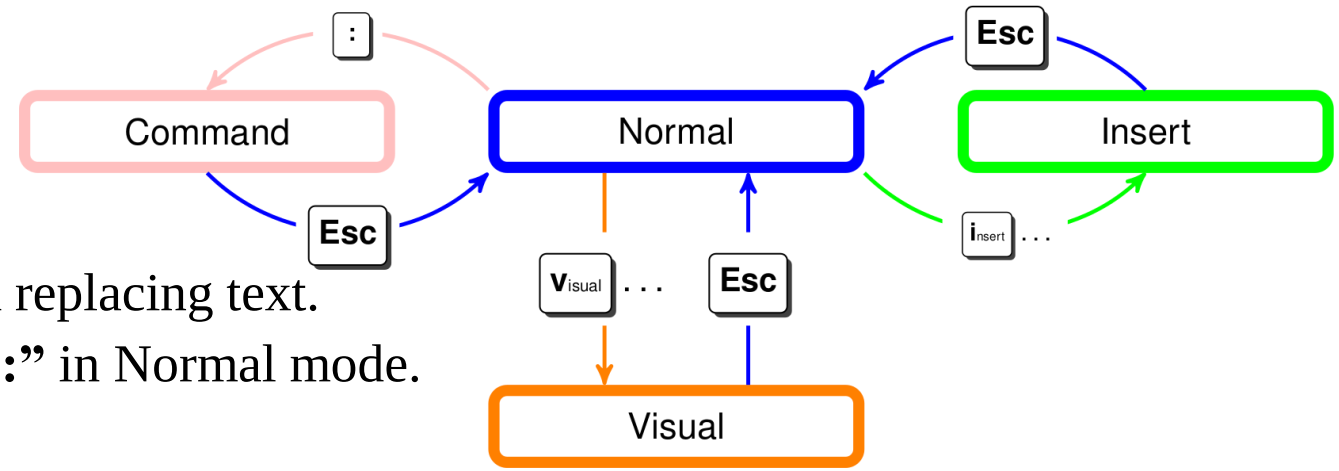
- **Editor**
  - Vim: **Vi** + **IM**proved
    - Initial name: Vi + imitation
    - By Bram Moolenaar from 1991
    - Since vim is aliased (shortcut, link, connection) to vi, even if you type vi, it connects to vim.
    - Linux and Unix command: vimtutor
    - New project: Neovim

```
                    VIM - Vi IMproved

                     version 8.2.2637
                 by Bram Moolenaar et al.
       Modified by <bugzilla@redhat.com>
Vim is open source and freely distributable

          Become a registered Vim user!
type  :help register<Enter>   for information

type  :q<Enter>               to exit
type  :help<Enter>  or  <F1>  for on-line help
type  :help version8<Enter>   for version info
                                                0,0-1         All
```

```
===============================================================================
=   W e l c o m e   t o   t h e   V I M   T u t o r   -   Version 1.7      =
===============================================================================

     Vim is a very powerful editor that has many commands, too many to
     explain in a tutor such as this.  This tutor is designed to describe
     enough of the commands that you will be able to easily use Vim as
     an all-purpose editor.

     The approximate time required to complete the tutor is 30 minutes,
     depending upon how much time is spent with experimentation.

     ATTENTION:
     The commands in the lessons will modify the text.  Make a copy of this
     file to practice on (if you started "vimtutor" this is already a copy).

     It is important to remember that this tutor is set up to teach by
     use.  That means that you need to execute the commands to learn them
     properly.  If you only read the text, you will forget the commands!

     Now, make sure that your Caps-Lock key is NOT depressed and press
     the   j   key enough times to move the cursor so that lesson 1.1
     completely fills the screen.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                        Lesson 1.1:  MOVING THE CURSOR
```

What is vim? (3/4)

- **Vim: mode**
  - Normal mode
    - Default mode, The first mode entered when running vim
    - Does not edit text, instead performs a command
      - h, j, k, l(cursor), dd(current line delete), yy(current line copy), p(paste)
    - Enter normal mode through **ESC**

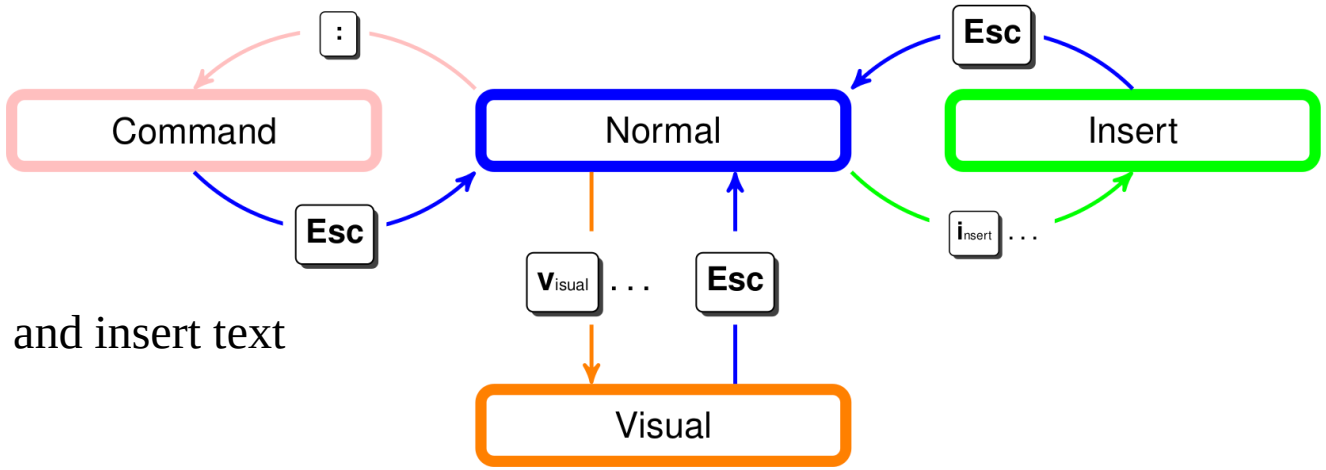


  - Command mode
    - Mode for saving, quitting, searching, and replacing text.
    - Command-line mode entered by typing **“:”** in Normal mode.
    - Command
      - :w(save the file), :q(quit vim), :wq(save and quit), :q!(quit without saving)

Ref : https://github.com/nilp0inter/Vim-for-Muggles

What is vim? (4/4)

- **Vim: mode**
  - Insert mode
    - Mode in which text can be edited
    - **Enter input mode**
      - i: Insert text before the cursor
      - a: Insert text after the cursor
      - o: Open a new line below the current line and insert text



  - Visual mode
    - Mode for selecting text blocks and executing commands.
    - Entered by typing **“v”** in Normal mode.
    - Type of Visual mode
      - v(select by character), V(select by line), Ctrl+v(select by block)

Ref : https://github.com/nilp0inter/Vim-for-Muggles

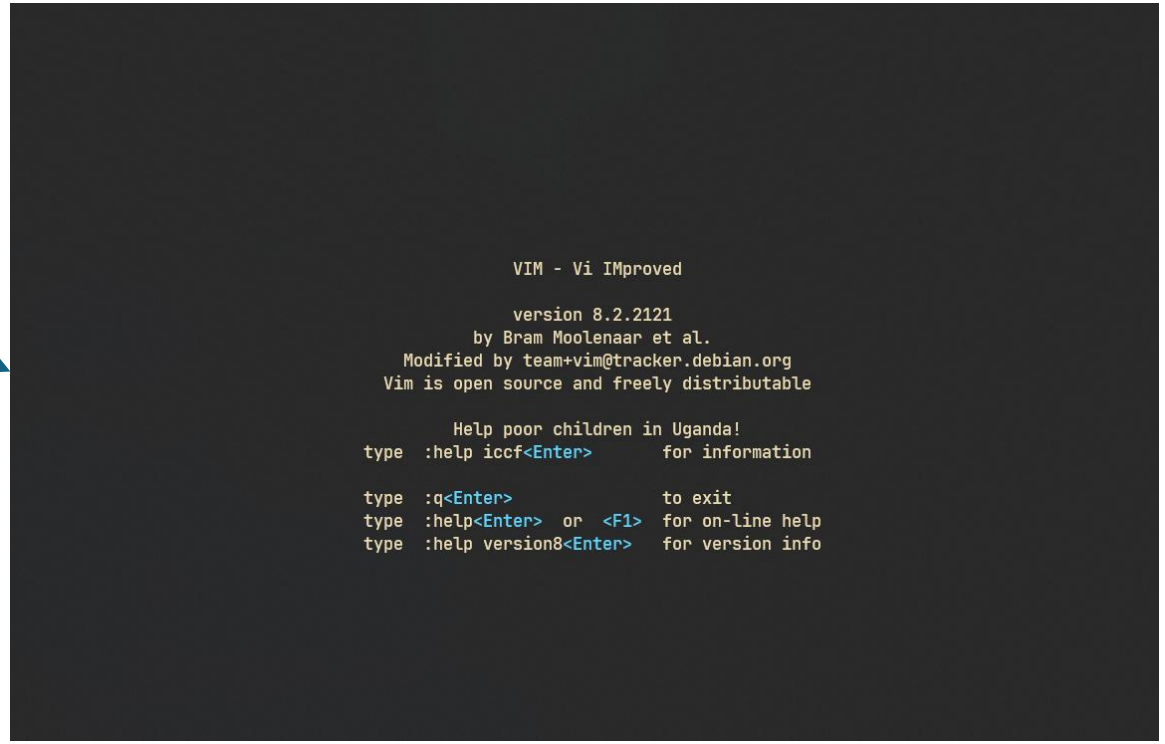
How to use vim in Linux? (1/2)

- **Vim**
  - install

```
~/Lecture/git/repo_dir
root# sudo apt install vim
```

  - open
    - vim [file name]

```
~/Lecture/git/repo_dir
root# vim
```

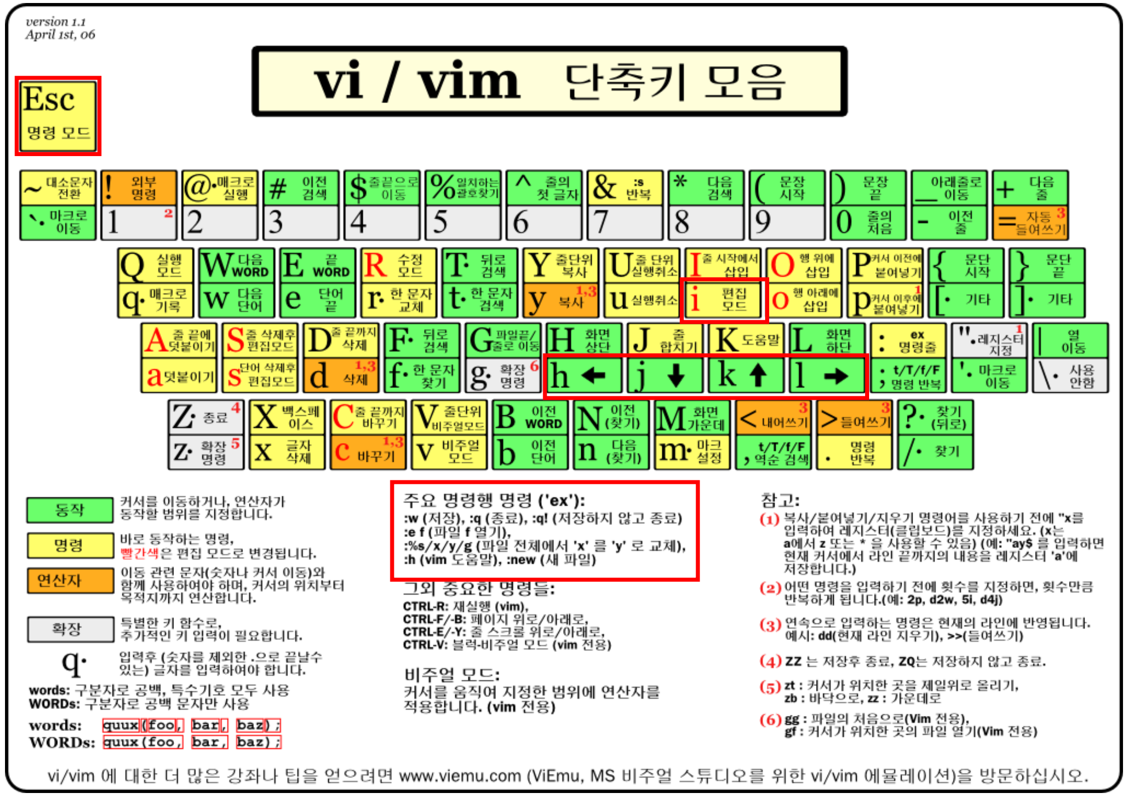


How to use vim in Linux? (2/2)

- **Vim**

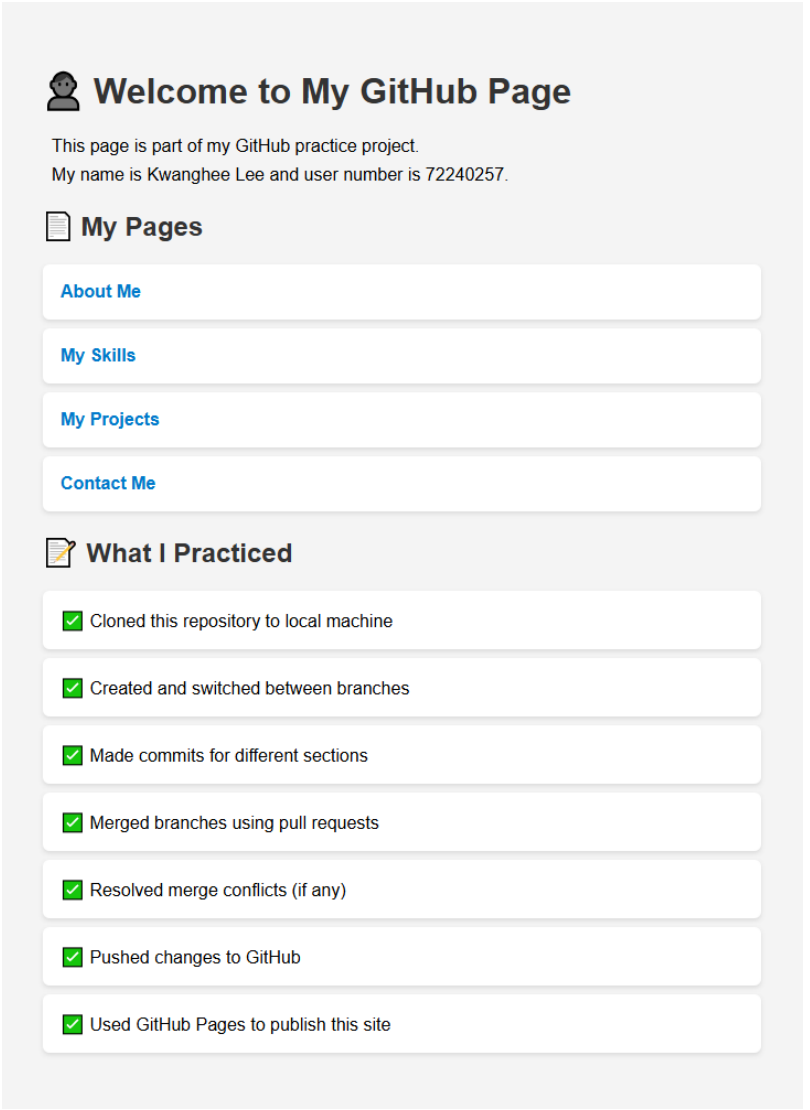
```
~/Lecture/git/repo_dir
root# ls
new_file   README   test_file
~/Lecture/git/repo_dir
root# vim new_file
```

```
pull request
~
~
~
~
"new_file" 1L, 13B                                          1,1           All
```



Practice

- **Follow the scenario**
  - Clone repository
  - Create and switch branch
  - Update file and commit
  - Merge ‘test’ branch and resolve conflicts
  - Make Pull Request



Practice

1. Clone Repository

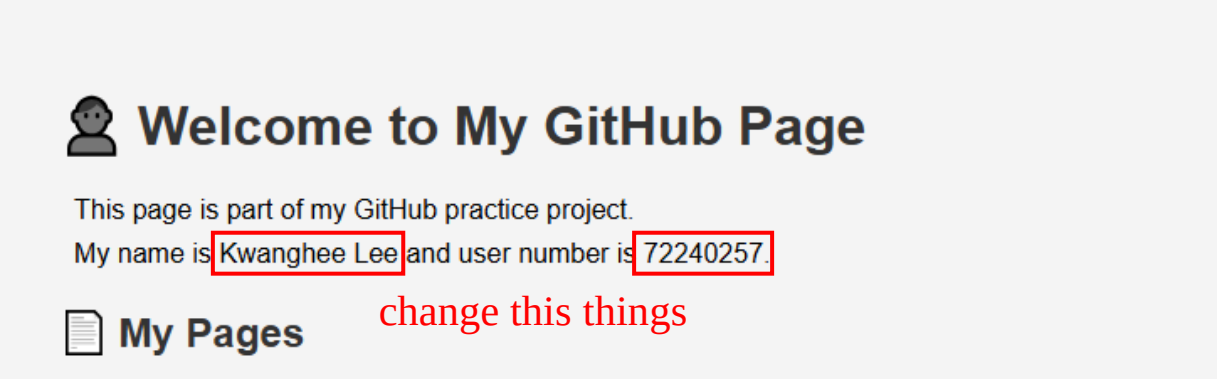
- https://github.com/DKU-EmbeddedSystem-Lab/2025_DKU_OpenSourceBasic
- Start from `main` branch

2. Create and Switch Branch

- Name branch by user number (e.g., 72240257)
- Work from a non-main branch

3. Update File and Commit

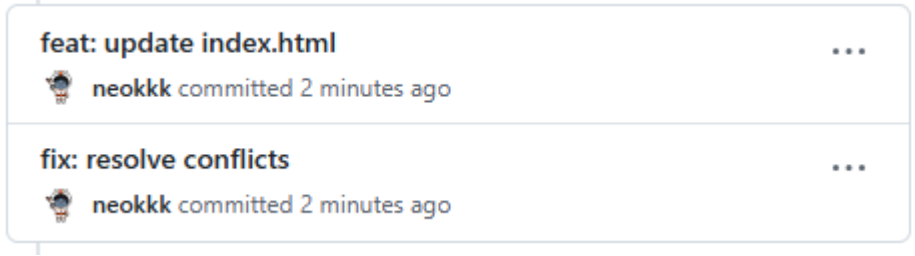
- Modify index.html
- Write meaningful commit messages



Practice

4. Merge ‘test’ Branch and Resolve Conflicts

- Try to merge ‘test’ branch from your own branch
  - If you try to merge from ‘test’, it will be restricted
- Occur conflicts on purpose
- Keep commit log for conflicts (just for resolving: empty diff)
- Push changes to remote repository (on your own branch)



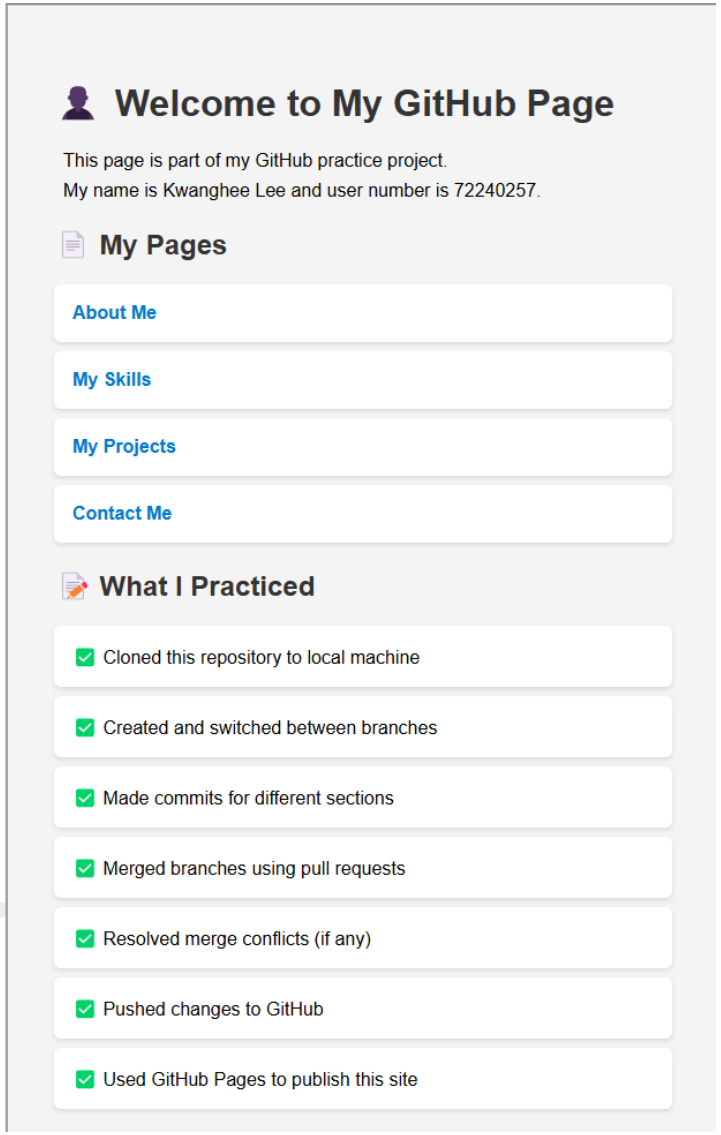
5. Make Pull Request

- Compare changes across branch: main ↔ own
- Set PR title to ‘[<user_number>] Update index.html’

Appendix

- **HTML viewer**
  - https://html.onlineviewer.net

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>My Personal GitHub Page</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 2rem;
      background-color: #f4f4f4;
    }
    h1, h2 {
      color: #333;
    }
    ul {
      list-style: none;
      padding-left: 0;
    }
    li {
      background: #fff;
      margin-bottom: 0.5rem;
      padding: 1rem;
      border-radius: 6px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    }
    p {
      margin: 0.5rem;
    }
    a {
      text-decoration: none;
      color: #007acc;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <h1>👤 Welcome to My GitHub Page</h1>
  <p>This page is part of my GitHub practice project.</p>
  <p>My name is Kwanghee Lee and user number is 72240257.</p>

  <h2>📄 My Pages</h2>
  <ul>
    <li><a href="about.html">About Me</a></li>
    <li><a href="skills.html">My Skills</a></li>
    <li><a href="projects.html">My Projects</a></li>
    <li><a href="contact.html">Contact Me</a></li>
  </ul>

  <h2>📝 What I Practiced</h2>
  <ul>
    <li>✅ Cloned this repository to local machine</li>
    <li>✅ Created and switched between branches</li>
    <li>✅ Made commits for different sections</li>
    <li>✅ Merged branches using pull requests</li>
    <li>✅ Resolved merge conflicts (if any)</li>
    <li>✅ Pushed changes to GitHub</li>
    <li>✅ Used GitHub Pages to publish this site</li>
  </ul>
</body>
</html>
```



Summary

- **Local Repository and Remote Repository**

- **GitHub**
  - What is github?
  - How to use github?

- **Vim**
  - What is Vim?
  - How to use vim in Linux?

Assignment 3

1. Practice

- **제출 요건**
  - Include student ID and date (using whoami, date)
  - 기한: 일주일
  - 양식: 포맷 없음, 장수 제한 없음, pdf (파일명: 오픈소스SW기초_{분반}_{이름}_{학번}.pdf)
  - 제출: e-Campus => 과제

Aknowledgement

- 본 교재는 2025년도 과학기술정보통신부 및 정보통신기획평가원의 ‘SW중심대학사업’ 지원을 받아 제작 되었습니다.

- 본 결과물의 내용을 전재할 수 없으며, 인용(재사용)할 때에는 반드시 과학기술정보통신부와 정보통신기획평가원이 지원한 ‘SW중심대학’의 결과물이라는 출처를 밝혀야 합니다.