



Git & GitHub

26기 TAG 이현우

우리가 배울것들

- Git & GitHub를 사용하는 이유
- 버전 관리 시스템이란?
- Git이란?
- Git의 사용법
- GitHub란?
- GitHub의 사용법

Git & GitHub를 사용하는 이유

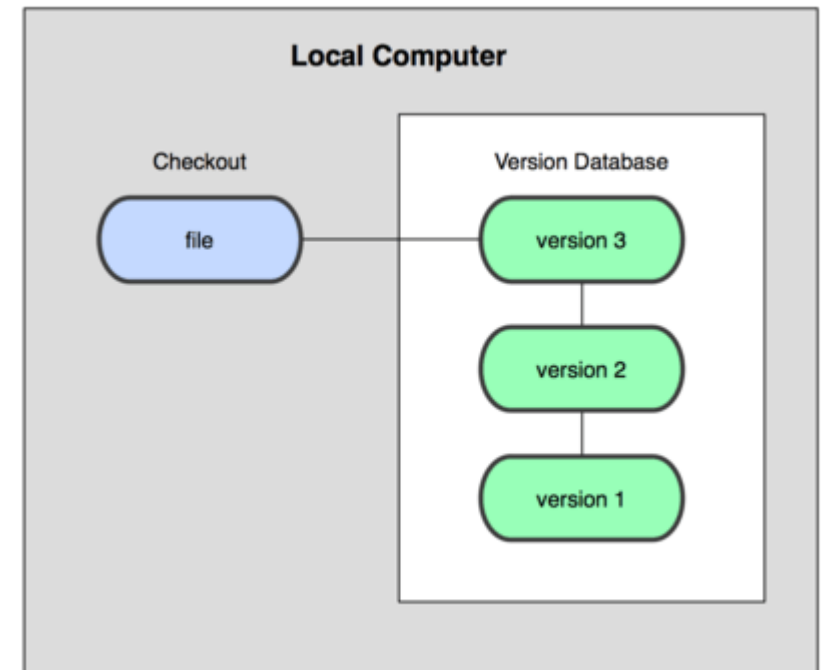
- Git은 버전 관리를 위한 도구
 - 프로젝트의 버전 & 이력 관리
- GitHub는 Git의 원격 저장소 서버를 제공
 - 로컬 PC에서 원격 서버 간 클라우드 서비스 제공
- 이처럼 소스코드의 버전 관리 및 클라우드는 개발자로서는 중요하다.

버전 관리 시스템(VCS)이란?

- 문서나 설계도, 소스 코드 등의 변경점을 관리해주는 소프트웨어 (출처: 위키백과)

- 종류

- 로컬 VCS: 서버 없이 로컬 컴퓨터 내에서 버전 관리
- 중앙집중식 VCS: 서버에 최종 버전 존재
- 분산 VCS: 서버 및 로컬 컴퓨터 모두 최종 버전 존재

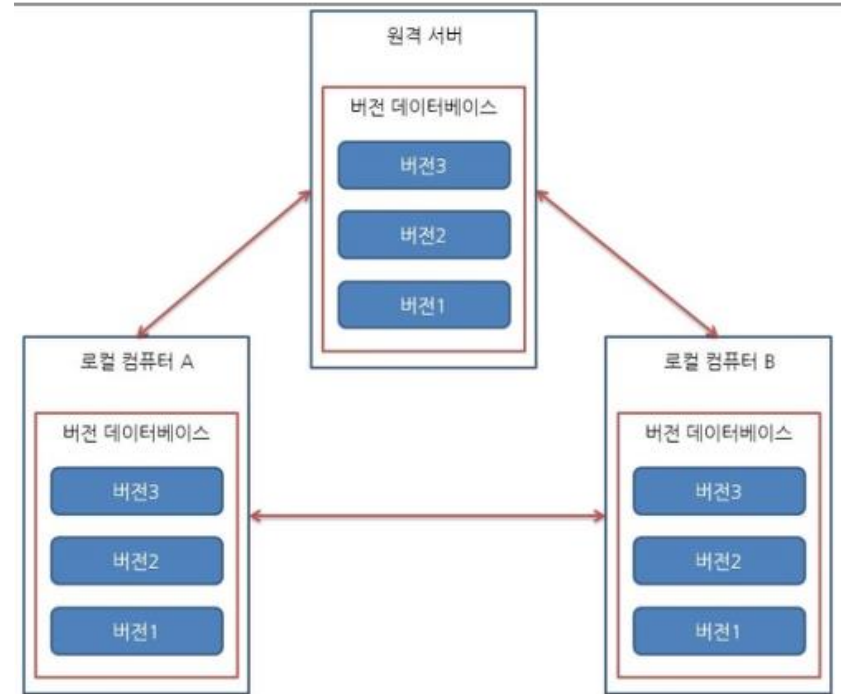


Git이란?



- 깃은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다. (출처: 위키백과)

분산 버전 관리 시스템 구성도



Git의 사용법⁽¹⁾



• 배울 것

- init: 현재 디렉토리에 git 저장소 생성
- add: 변경된 파일을 스테이지에 추가
- commit: 스테이지에 추가된 파일들을 커밋
- remote add: 새로운 원격 저장소 추가
- push: 저장소에 현재 로컬컴퓨터의 커밋 내용을 업로드

Git의 사용법⁽²⁾



- 실습에 앞서...
- Git 설치
- GitHub 가입
- GitHub 저장소 생성

Git의 사용법⁽³⁾



- Git 설치⁽¹⁾

- <https://git-scm.com/downloads> 접속

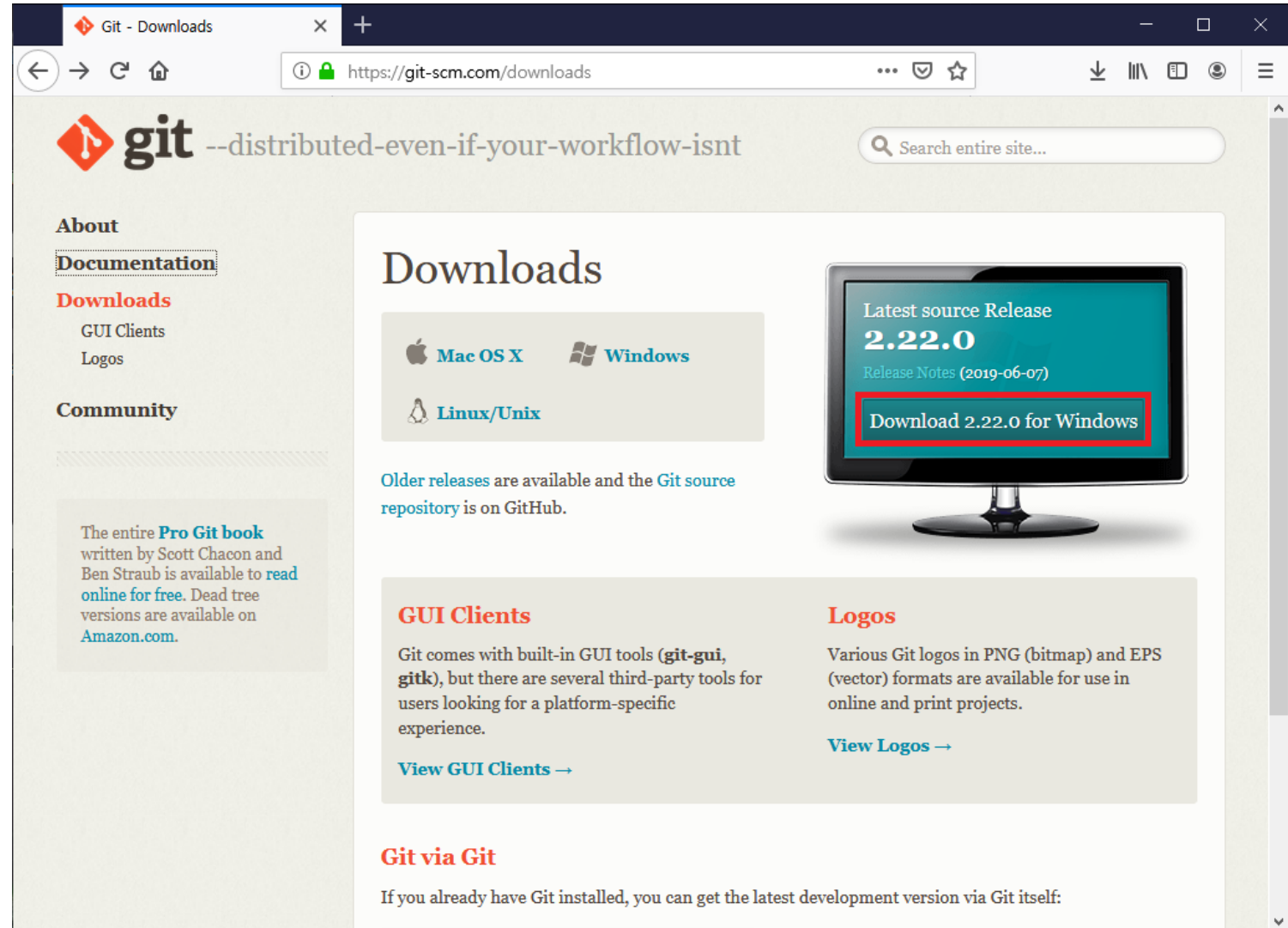
<https://git-scm.com/downloads>

Git의 사용법(3)



• Git 설치(2)

• 다운로드 클릭



Git의 사용법⁽⁴⁾



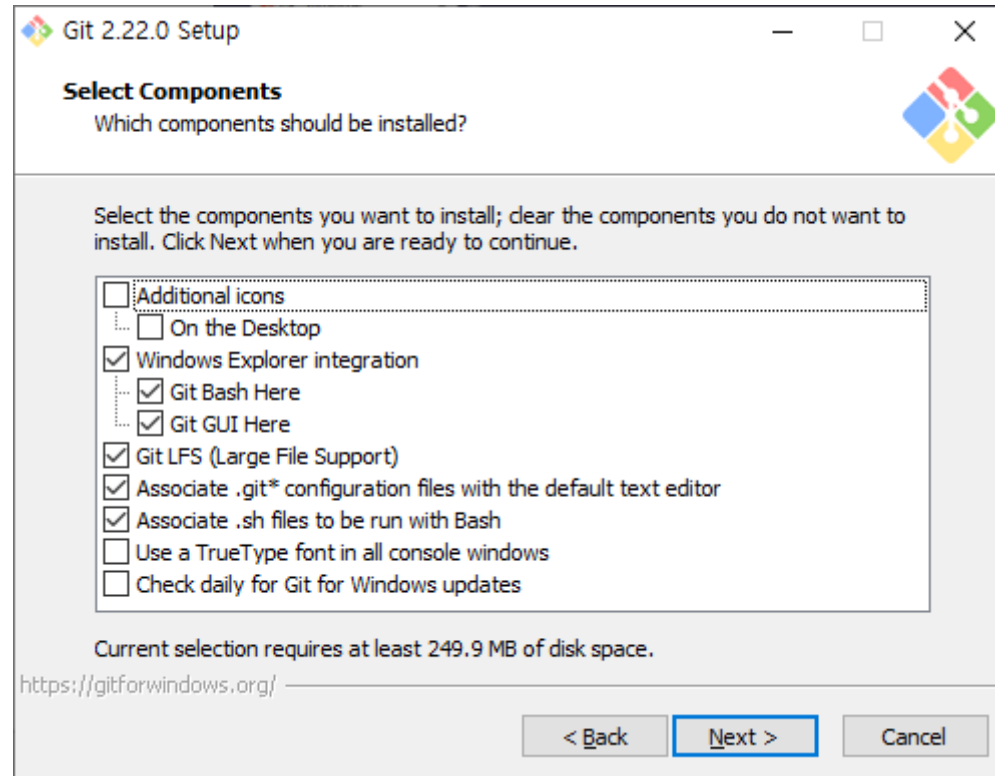
• Git 설치⁽³⁾



Git의 사용법(5)



• Git 설치(4)

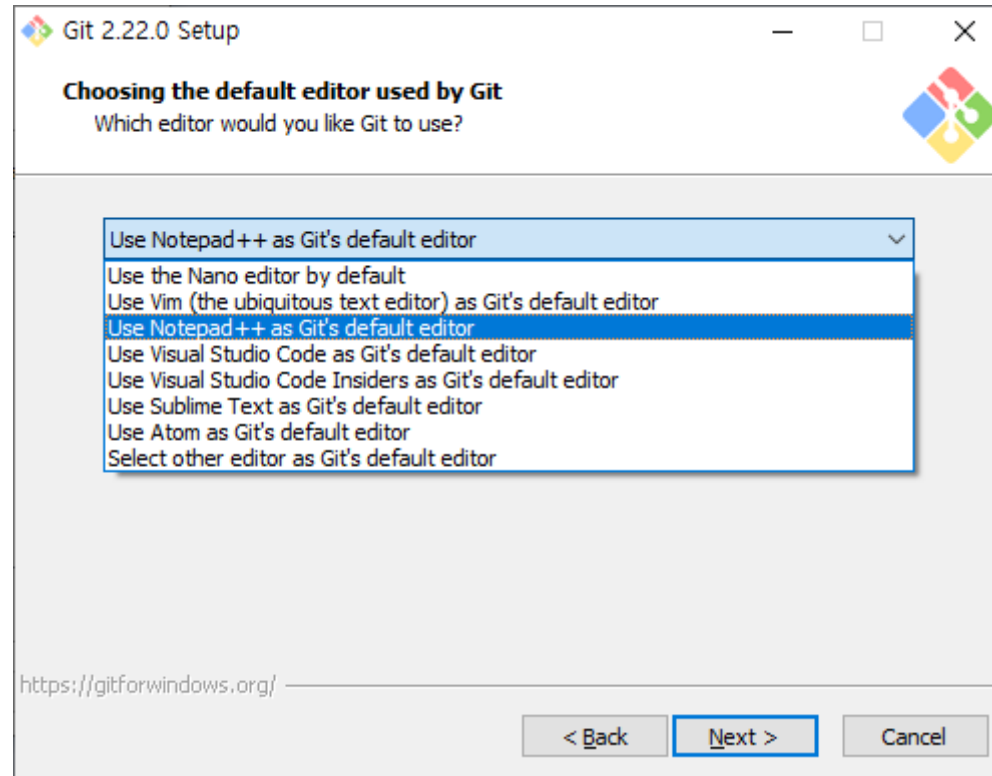


Git의 사용법⁽⁶⁾



- Git 설치⁽⁵⁾

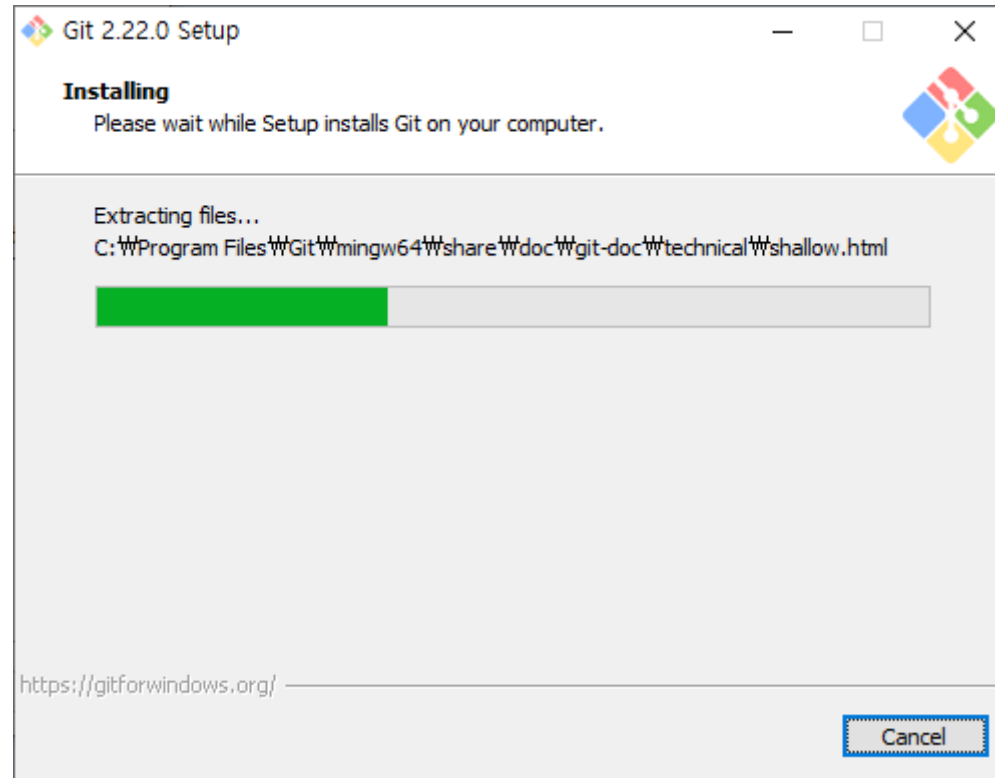
- Use Notepad++ as ..



Git의 사용법⁽⁷⁾



• Git 설치⁽⁶⁾

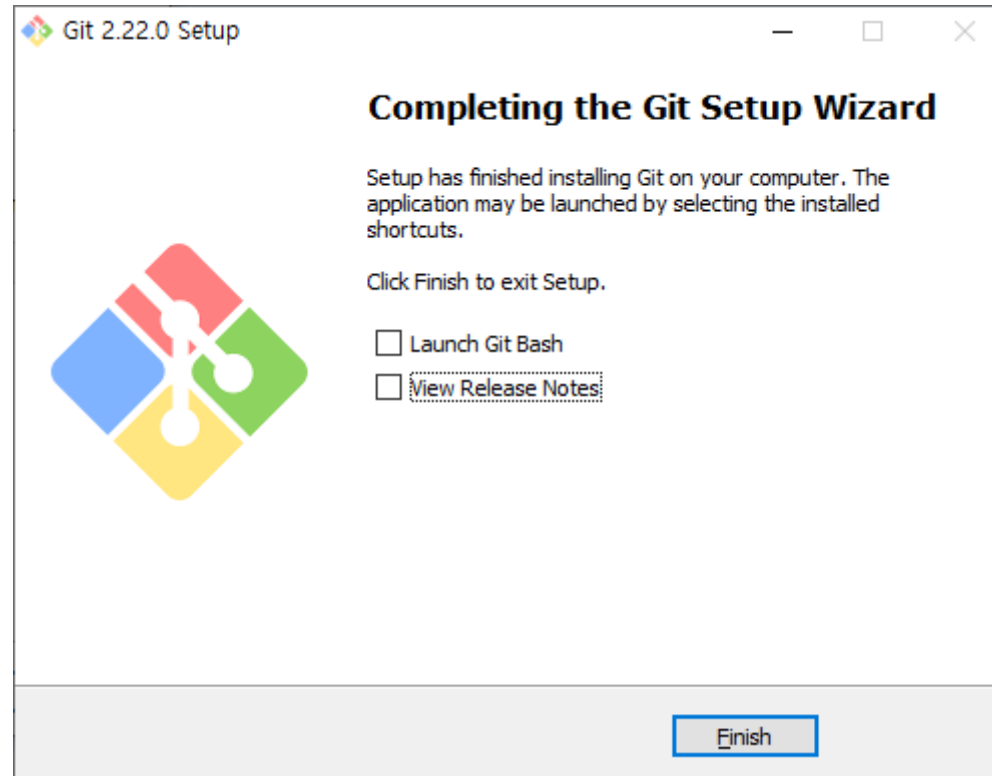


Git의 사용법⁽⁸⁾



- Git 설치⁽⁷⁾

- Finish!



Git의 사용법⁽⁹⁾



- Git 설치⁽⁸⁾

- git --version

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\wheonwoo>git --version
git version 2.22.0.windows.1

C:\Users\wheonwoo>_
```

Git의 사용법⁽¹⁰⁾



- GitHub 가입⁽¹⁾
 - <https://github.com> 접속

<https://github.com>

Git의 사용법⁽¹¹⁾



- GitHub 가입⁽²⁾

- Sign up 클릭

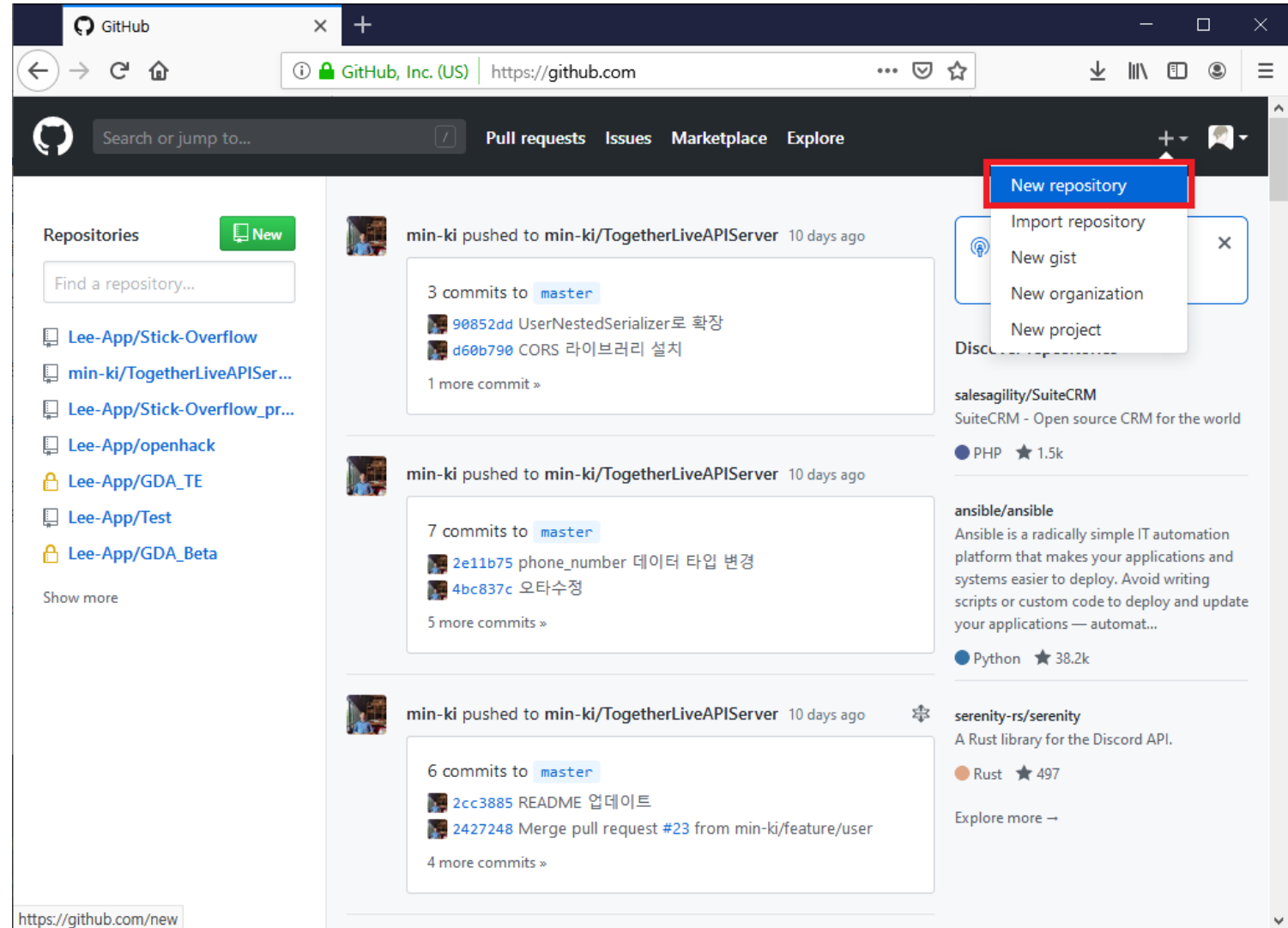
The screenshot shows the GitHub homepage in a web browser. The browser's address bar displays 'https://github.com'. The navigation bar at the top includes links for 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. A search bar is also present. On the right side of the navigation bar, the 'Sign in' and 'Sign up' buttons are visible, with 'Sign up' highlighted by a red rectangular box. The main content area features the text 'Built for developers' and a description of GitHub as a development platform. On the right side of the page, there is a sign-up form with the following fields: 'Username', 'Email', and 'Password'. Below the password field, there is a note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. At the bottom of the form is a green button labeled 'Sign up for GitHub'. Below the button, there is a disclaimer: 'By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.'

Git의 사용법⁽¹²⁾



• GitHub 저장소 생성⁽¹⁾

• New repository 클릭



Git의 사용법⁽¹³⁾



• GitHub 저장소 생성⁽²⁾

• 저장소 이름 입력

Create a New Repository

GitHub, Inc. (US) <https://github.com/new>

Create a new repository

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

Owner: Lee-App

Repository name *
test_github ✓

Great repository names are short and memorable. Need inspiration? How about **shiny-fiesta**?

Description (optional)

☒ Public
Anyone can see this repository. You choose who can commit.

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

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

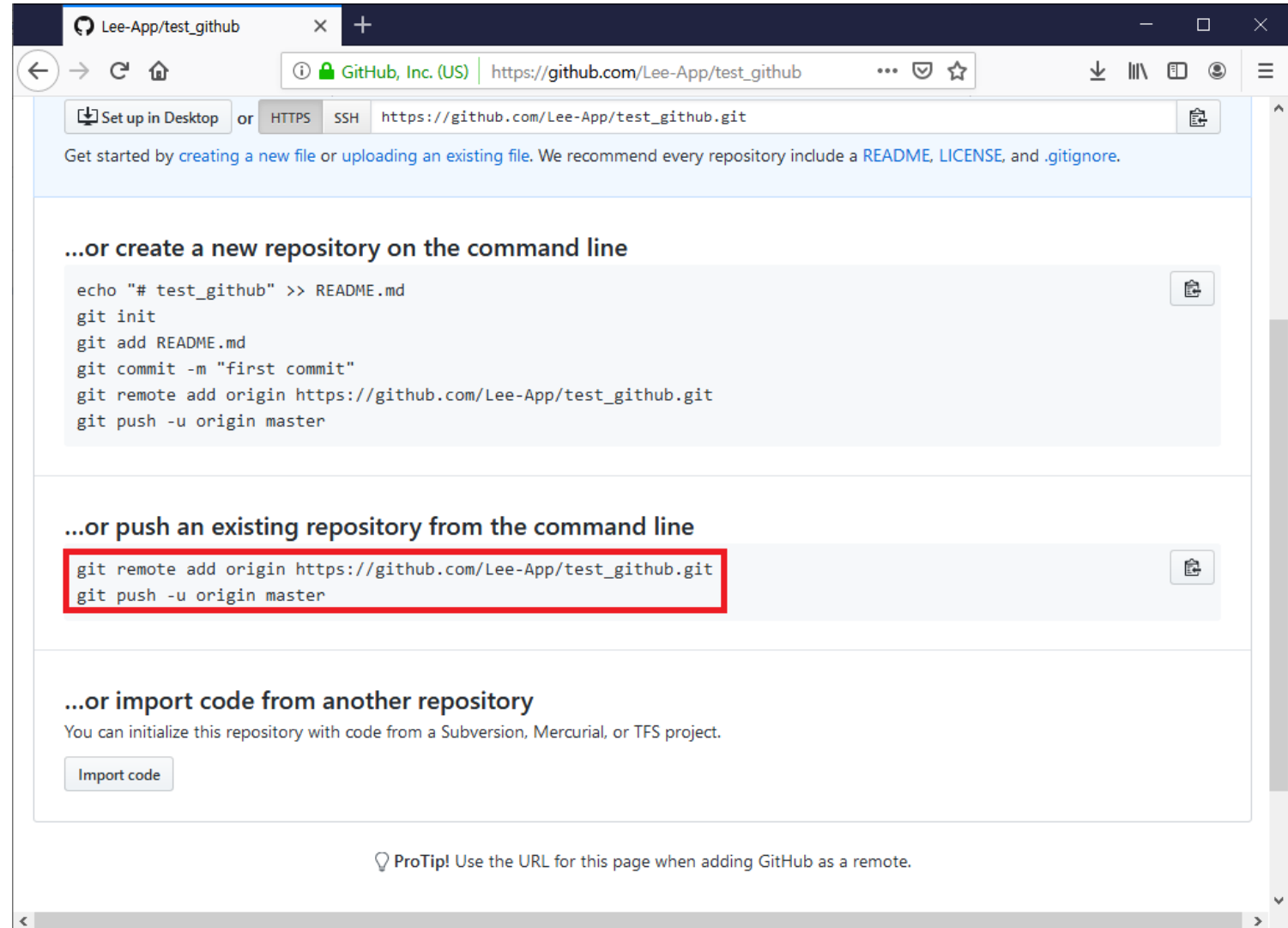
Create repository

Git의 사용법(14)



• GitHub 저장소 생성(3)

• 완료!



Git의 사용법⁽¹⁵⁾



- 배울 것 복습!

- init: 현재 디렉토리에 git 저장소 생성
- add: 변경된 파일을 스테이지에 추가
- commit: 스테이지에 추가된 파일들을 커밋
- remote add: 새로운 원격 저장소 추가
- push: 저장소에 현재 로컬컴퓨터의 커밋 내용을 업로드

Git의 사용법(16)



- 몇 가지 세팅!

- `git config --global user.name 유저명`
- `git config --global user.email 이메일`

```
C:\git_test>git config --global user.name "lee-app"  
C:\git_test>git config --global user.email "asow7345@hallym.ac.kr"  
C:\git_test>_
```

- `cd C:/`
- `mkdir git_test`
- `cd git_test`

```
C:\Users\wheonwoo>cd C:/  
C:\>mkdir git_test  
C:\>cd git_test  
C:\git_test>_
```

* **git config:** Git 설정 변경, **cd:** change directory, **mkdir:** make directory

Git의 사용법⁽¹⁷⁾

- init: 현재 디렉토리에 git 저장소 생성

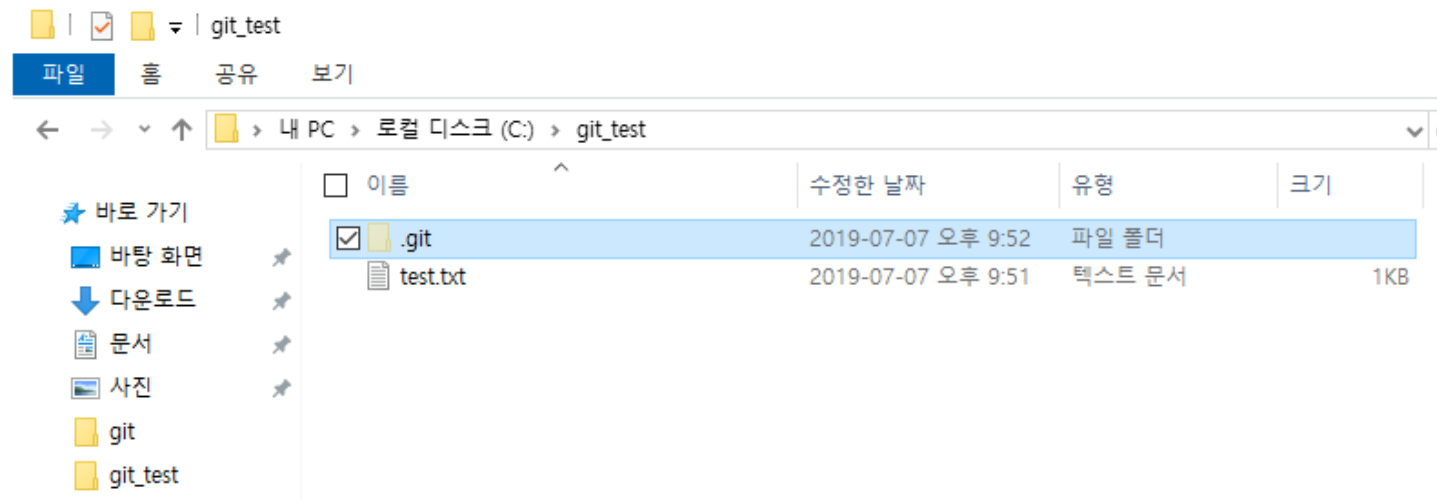
- git init

```
C:\git_test>git init
Initialized empty Git repository in C:/git_test/.git/

C:\git_test>_
```

- 실제 저장소로 사용되는 폴더에 .git 파일 생성

git 저장소라는 표시



Git의 사용법⁽¹⁸⁾



- add: 변경된 파일을 스테이지에 추가
 - git add *
 - git add test.txt
- 여기서 *은 현재 폴더의 모든 파일을 의미
- 스테이지 란?
- 커밋을 하려는 파일을 올려놓는 공간

```
C:\git_test>git add *  
  
C:\git_test>git commit -m "test.txt 파일 추가"  
[master (root-commit) e68b3d3] test.txt 파일 추가  
1 file changed, 1 insertion(+)  
create mode 100644 test.txt  
  
C:\git_test>_
```


Git의 사용법⁽¹⁹⁾



- commit: 스테이지에 추가된 파일들을 커밋

- `git commit -m` “커밋에 대한 내용”

```
C:\git_test>git add *  
  
C:\git_test>git commit -m "test.txt 파일 추가"  
[master (root-commit) e68b3d3] test.txt 파일 추가  
1 file changed, 1 insertion(+)  
create mode 100644 test.txt  
  
C:\git_test>_
```

- 커밋 이란?

- 어떤 순간 작업공간의 상태를 저장한 것

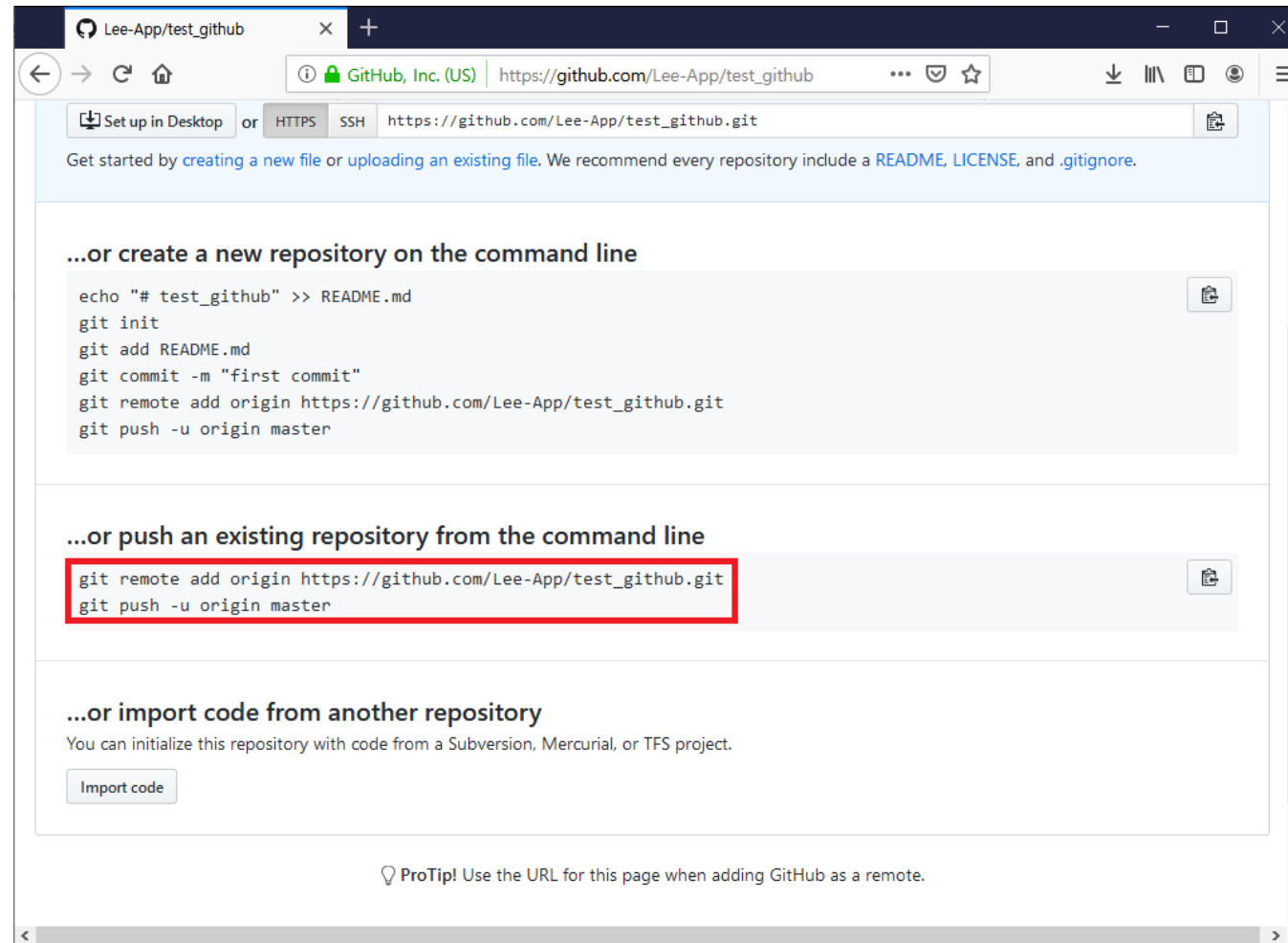
- 현재 작업공간의 상태를 커밋으로 만들어서 저장소에 저장

Git의 사용법(20)



- remote add: 새로운 원격 저장소 추가(1)

- 저장소를 만들었을 때 페이지



Git의 사용법⁽²¹⁾



- remote add: 새로운 원격 저장소 추가⁽²⁾
 - git remote add origin “자신의 원격 저장소 주소”
 - 현재의 로컬 저장소에
원격 저장소를 추가
 - 즉, 로컬 컴퓨터의 저장소와
GitHub 서버의 원격 저장소가 같다.

```
C:\git_test>git remote add origin https://github.com/Lee-App/test_github.git
C:\git_test>git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 461 bytes | 461.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/Lee-App/test_github.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
C:\git_test>.
```

Git의 사용법⁽²²⁾



- push: 저장소에 현재 로컬컴퓨터의 커밋 내용을 업로드

- git push -u origin master

- 현재의 로컬 저장소의 커밋 및 파일을 원격 저장소에 업로드한다.

- 로컬 컴퓨터에서 작업한 내용을 원격 저장소에 저장하는 것

```
C:\git_test>git remote add origin https://github.com/Lee-App/test_github.git
C:\git_test>git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 461 bytes | 461.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/Lee-App/test_github.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
C:\git_test>.
```

GitHub란? GitHub



- 깃허브는 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅 서비스이다. (출처: 위키백과)
- 원격 저장소. 즉, 클라우드의 역할을 한다.

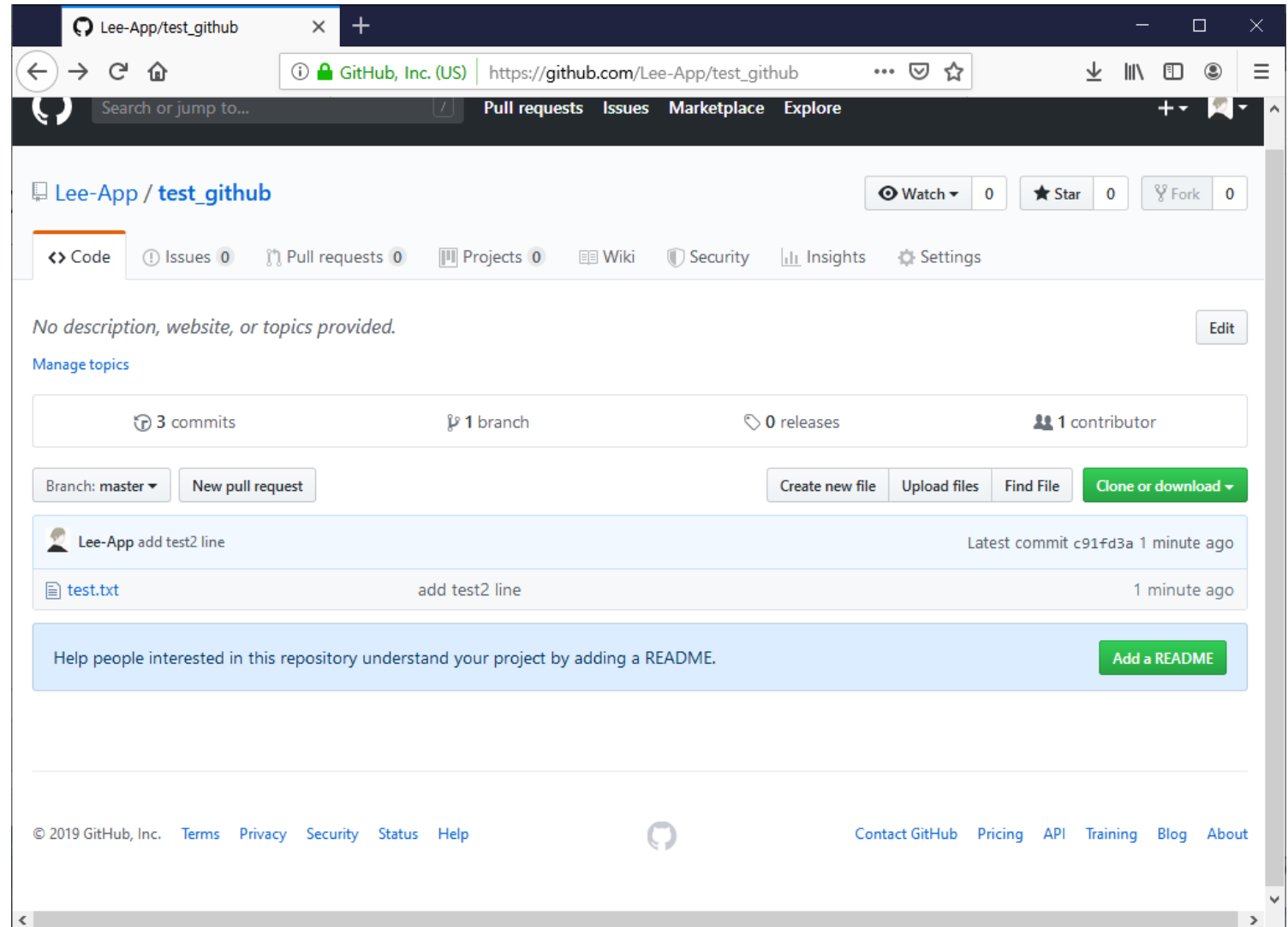


GitHub의 사용법⁽¹⁾



- GitHub 저장소 사용법⁽¹⁾

- 저장소 메인 화면

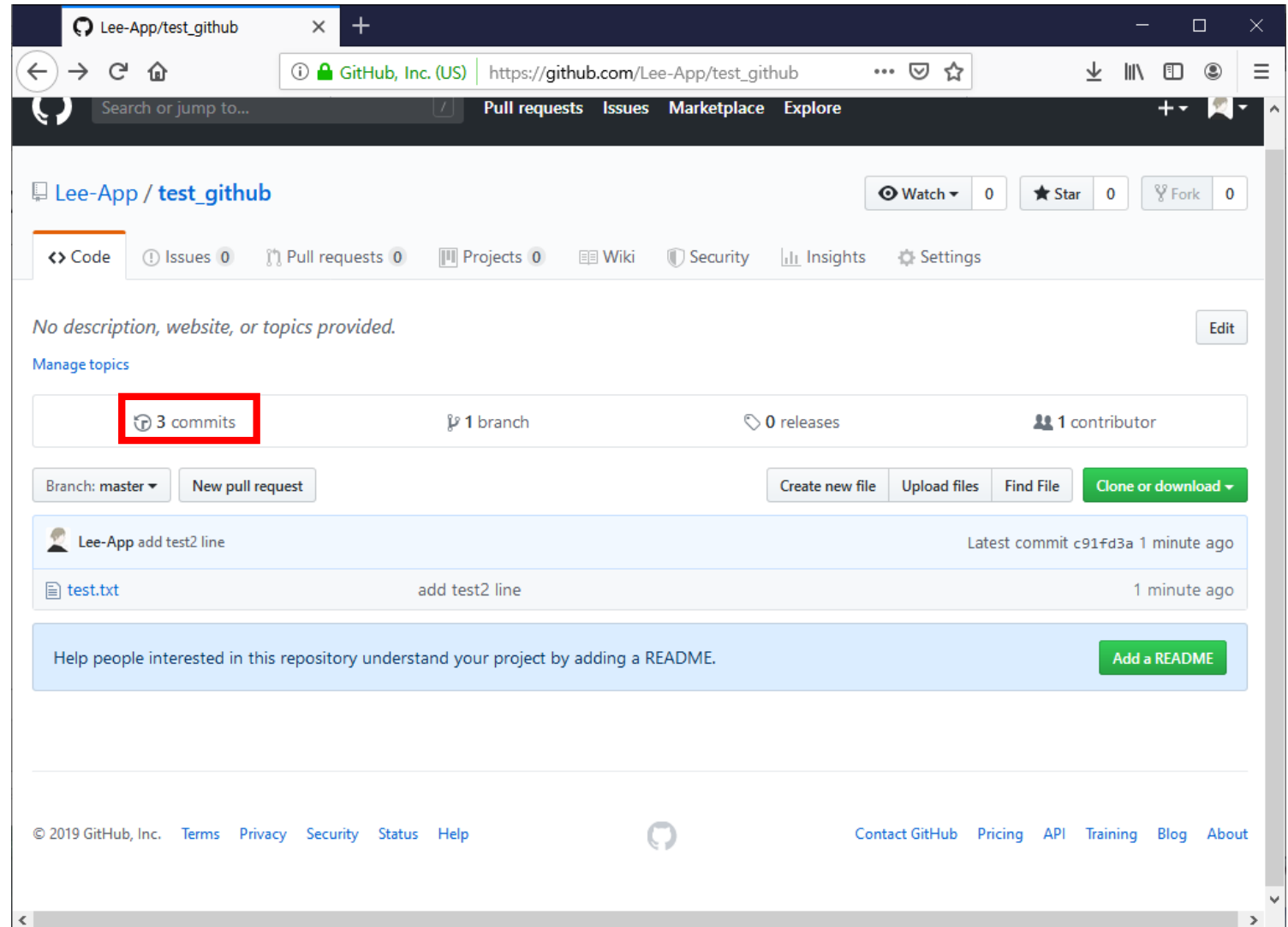


GitHub의 사용법(2)



• GitHub 저장소 사용법(2)

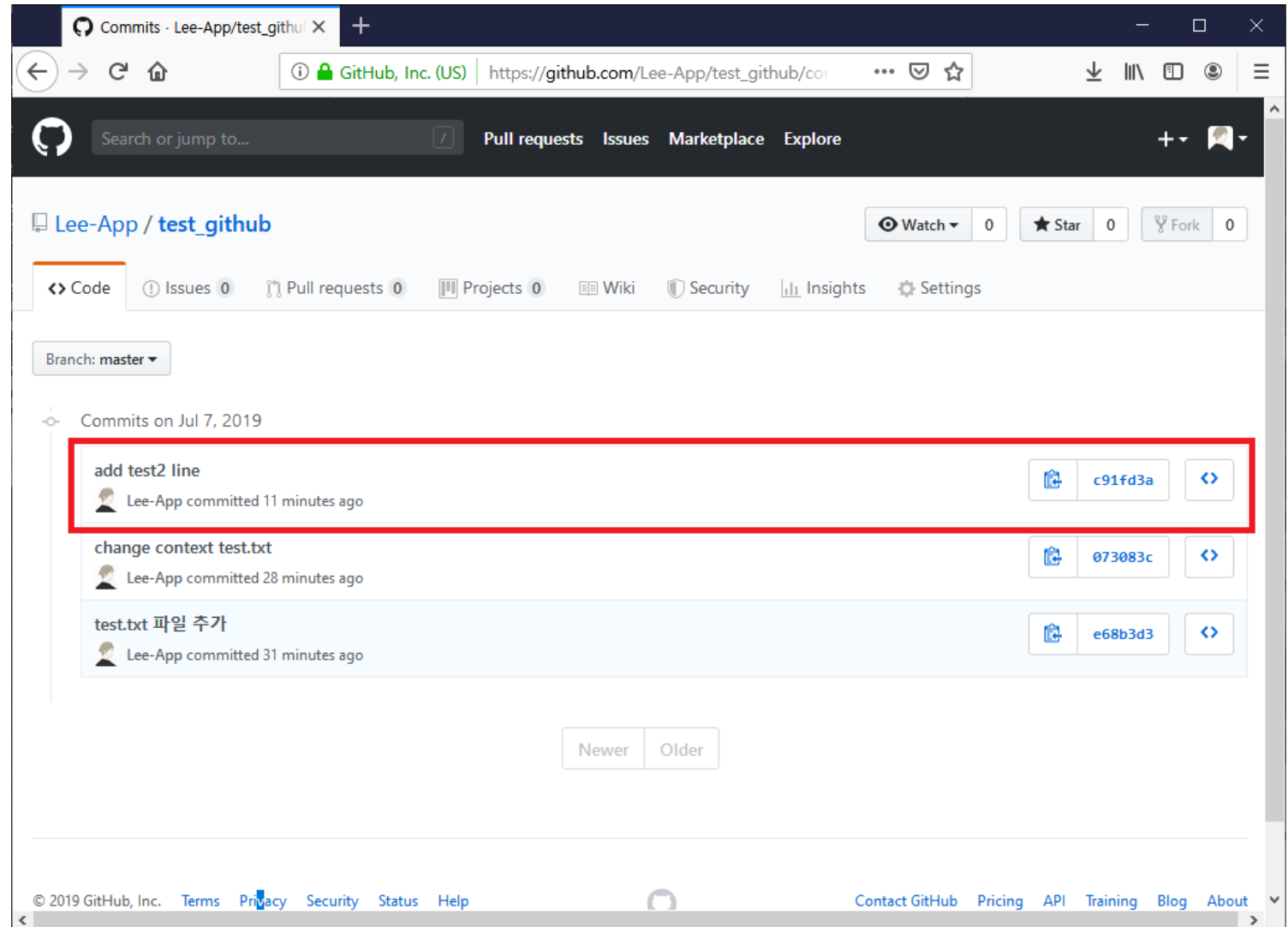
- commits 확인(1)
- 저장소 메인 페이지에서 commits 탭 클릭



GitHub의 사용법⁽³⁾



- GitHub 저장소 사용법⁽³⁾
 - commits 확인⁽²⁾
 - 모든 commit을 볼 수 있다.



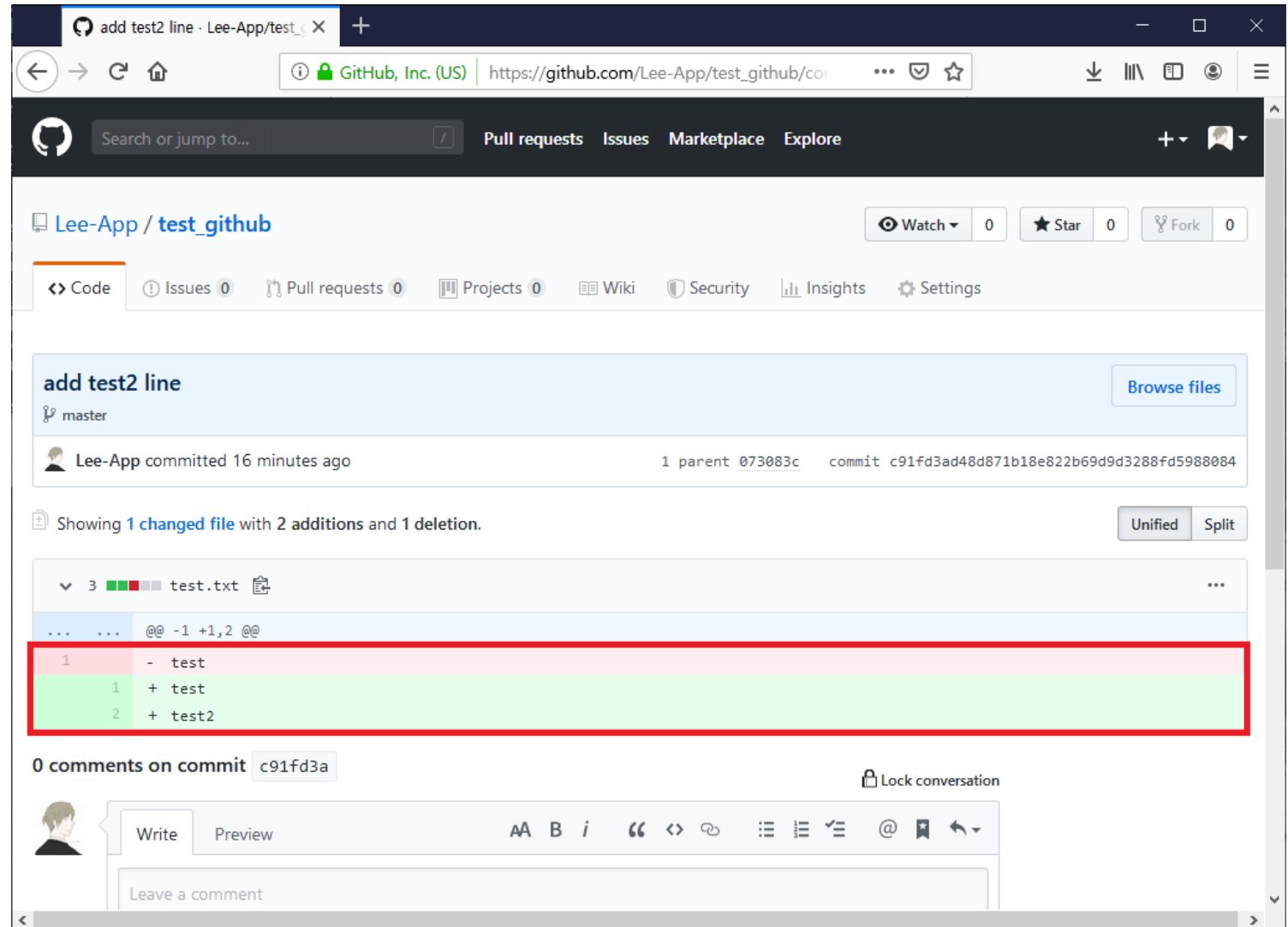
GitHub의 사용법(4)



- GitHub 저장소 사용법(4)

- commits 확인(3)

- commit 클릭 시 세부적인 내용을 볼 수 있다.

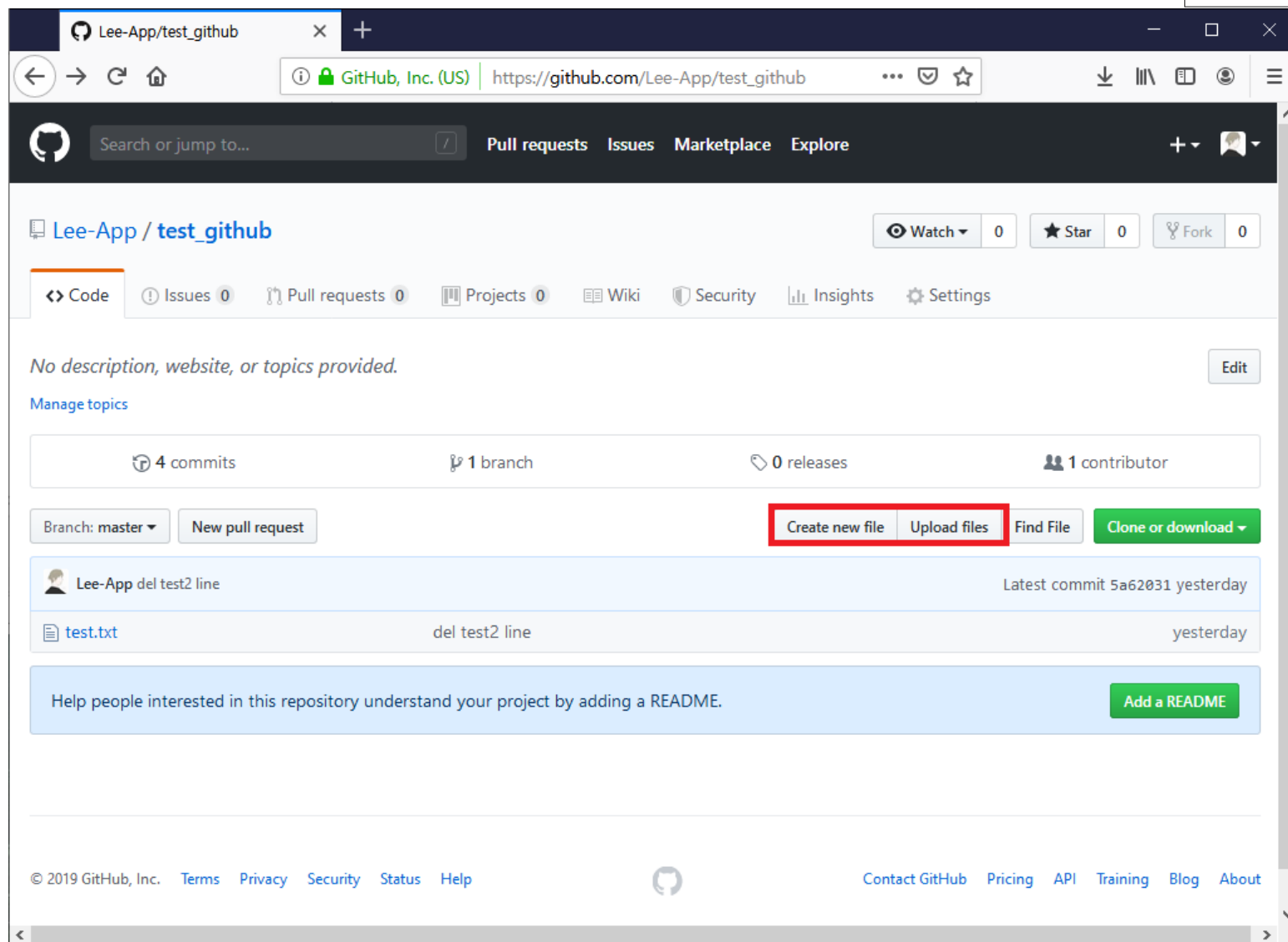


GitHub의 사용법(5)



• GitHub 저장소 사용법(5)

- 저장소에 파일 추가
- Create new file
새로운 파일을 웹에서 작성
- Upload file
이미 생성한 파일을 저장소에 업로드



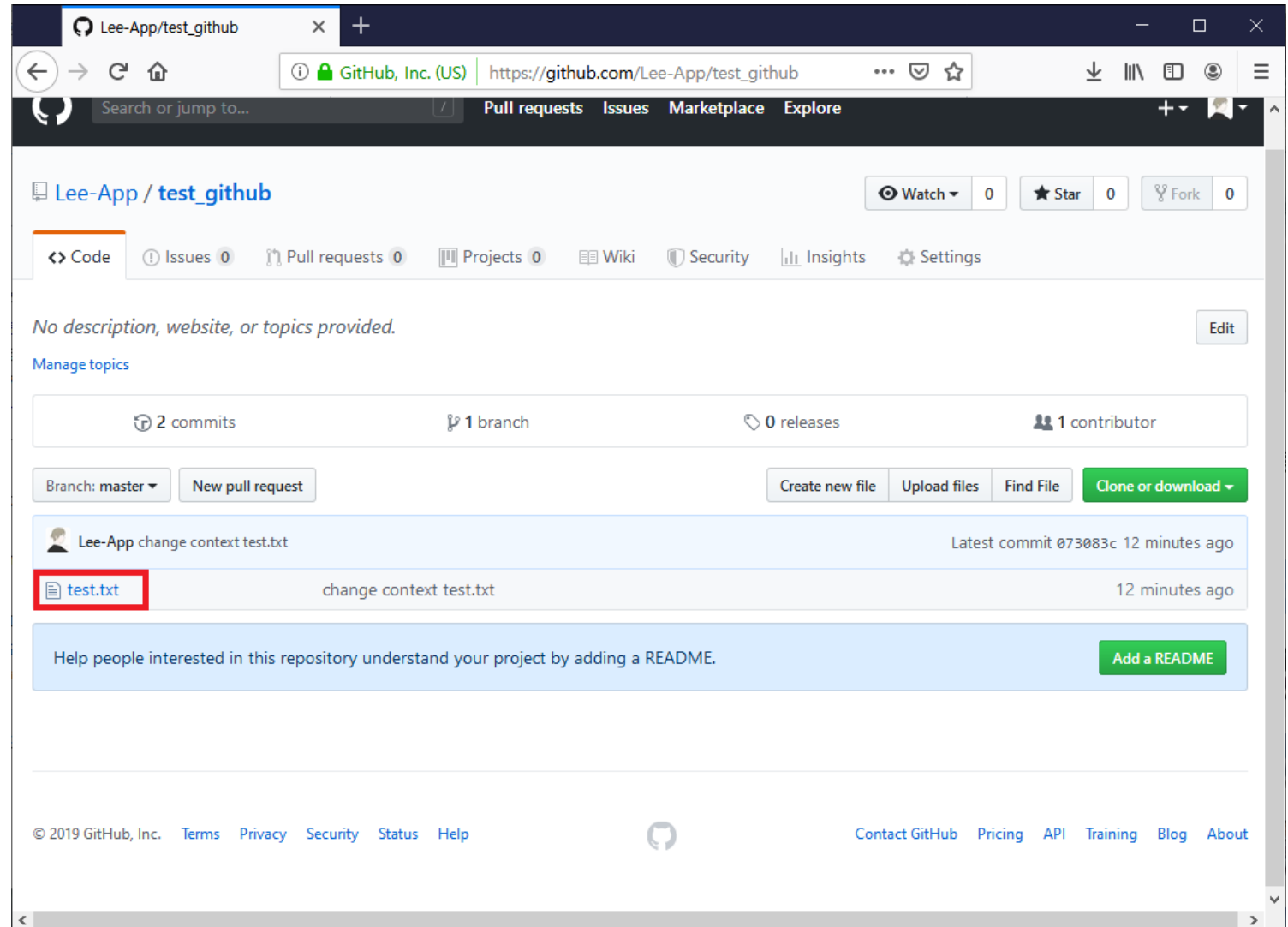
GitHub의 사용법(6)



• GitHub 저장소 사용법(6)

• 저장소에 파일 수정(1)

• 수정할 파일을 클릭



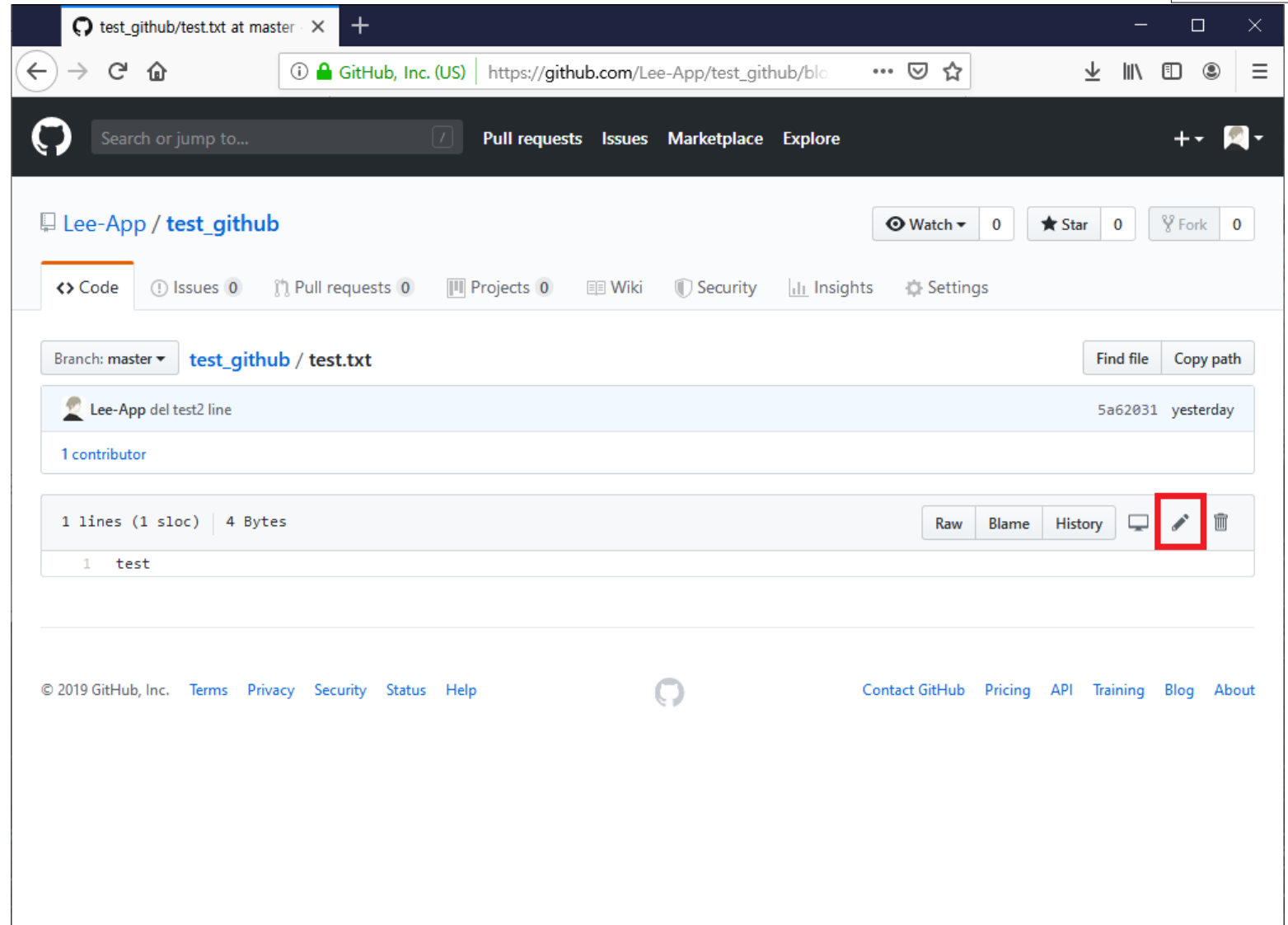
GitHub의 사용법⁽⁷⁾



• GitHub 저장소 사용법⁽⁷⁾

• 저장소에 파일 수정⁽²⁾

• 펜 모양 클릭



GitHub의 사용법⁽⁸⁾



• GitHub 저장소 사용법⁽⁸⁾

• 저장소에 파일 수정

- 파일 내용 수정 후
커밋 내용 작성
커밋 버튼 클릭

Editing test_github/test.txt at | X +

GitHub, Inc. (US) | https://github.com/Lee-App/test_github/edi

Commit changes

test.txt 수정

Add an optional extended description...

asow7345@hallym.ac.kr

Choose which email address to associate with this commit

☒ Commit directly to the master branch.

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

Commit changes Cancel

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

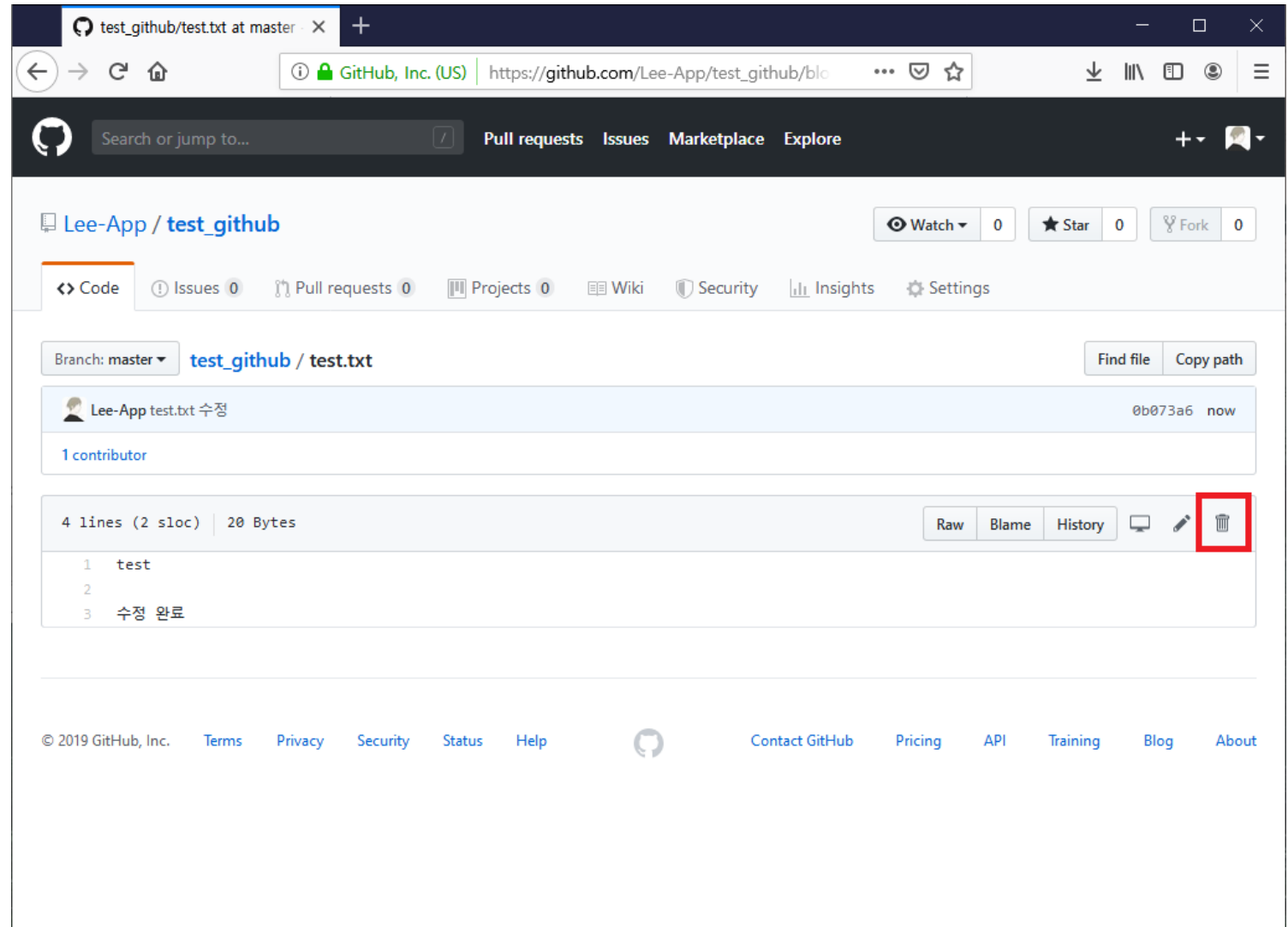
GitHub의 사용법⁽⁹⁾



• GitHub 저장소 사용법⁽⁹⁾

• 저장소에 파일 삭제

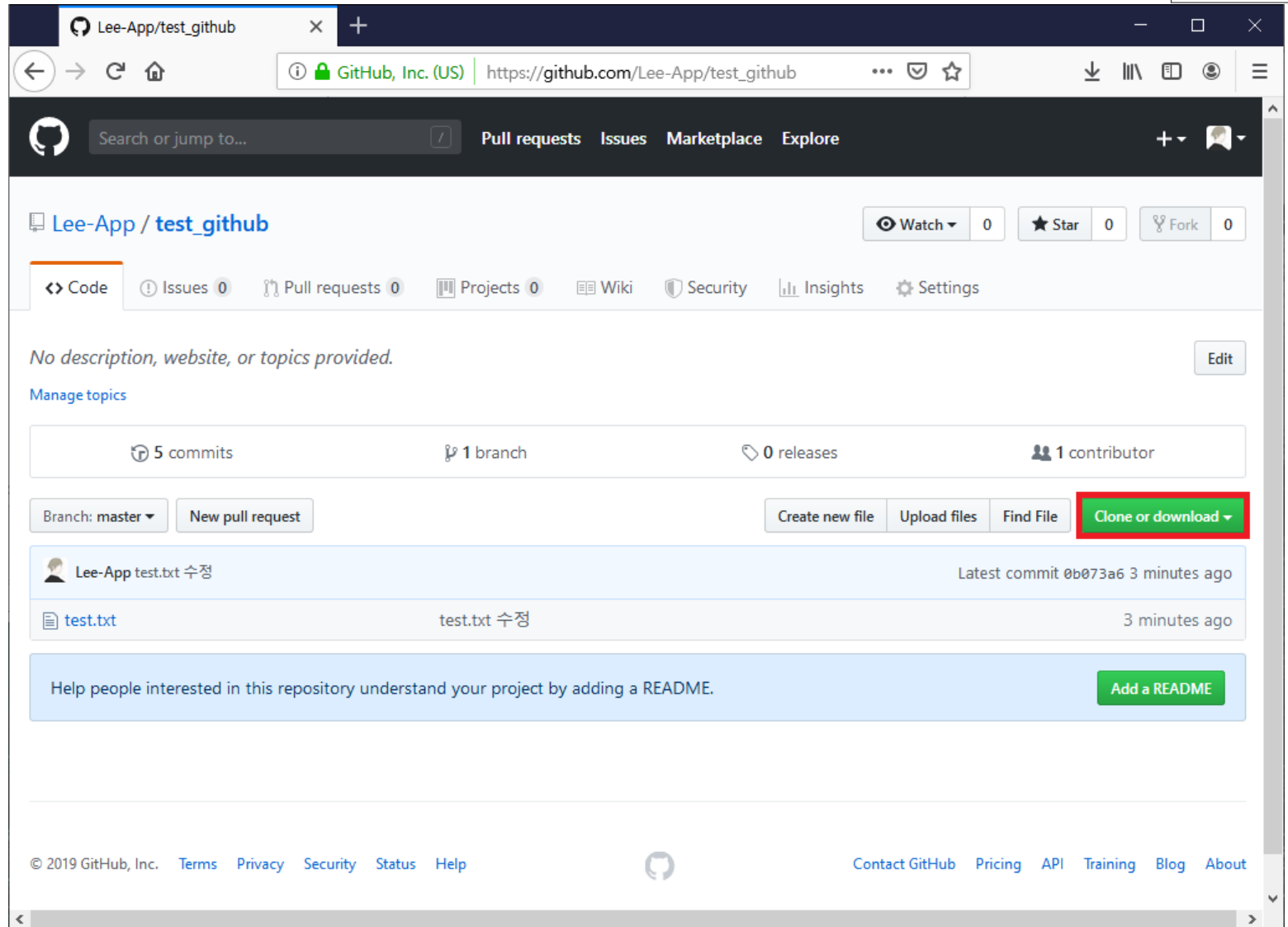
• 휴지통 모양 클릭



GitHub의 사용법⁽¹⁰⁾



- GitHub 저장소 다운로드⁽¹⁾
 - Clone or download 클릭



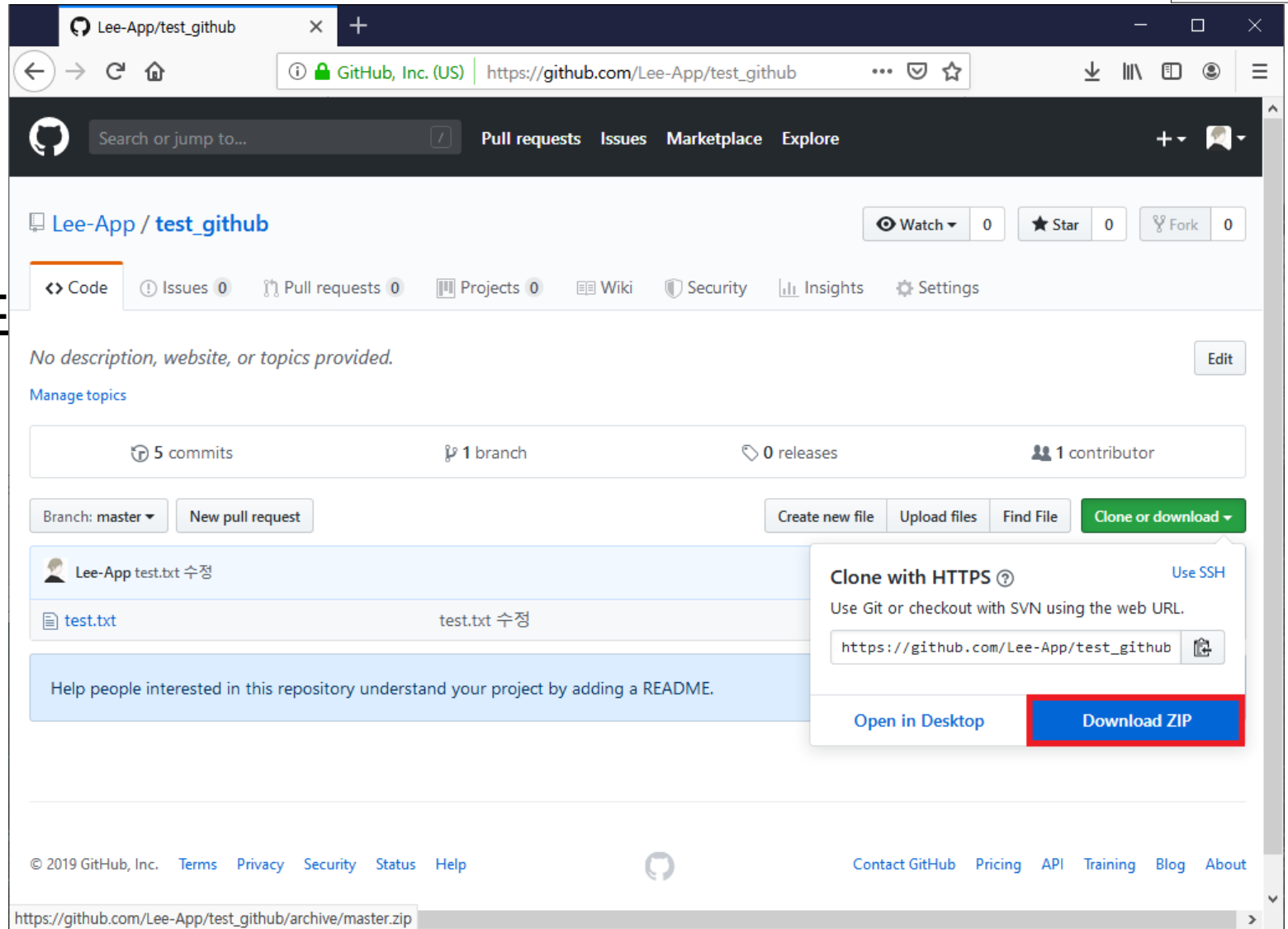
GitHub의 사용법⁽¹¹⁾



- GitHub 저장소 다운로드⁽²⁾

- Download ZIP 클릭

- 저장소가 압축파일로 다운로드





- 깃은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템
- 스테이지는 커밋을 하려는 파일을 올려 놓는 공간
- 커밋은 현재 작업공간의 상태를 커밋으로 만들어서 저장소에 저장
- 깃허브는 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅 서비스이다.
- 깃허브는 원격 저장소의 역할을 한다.

수고하셨습니다.