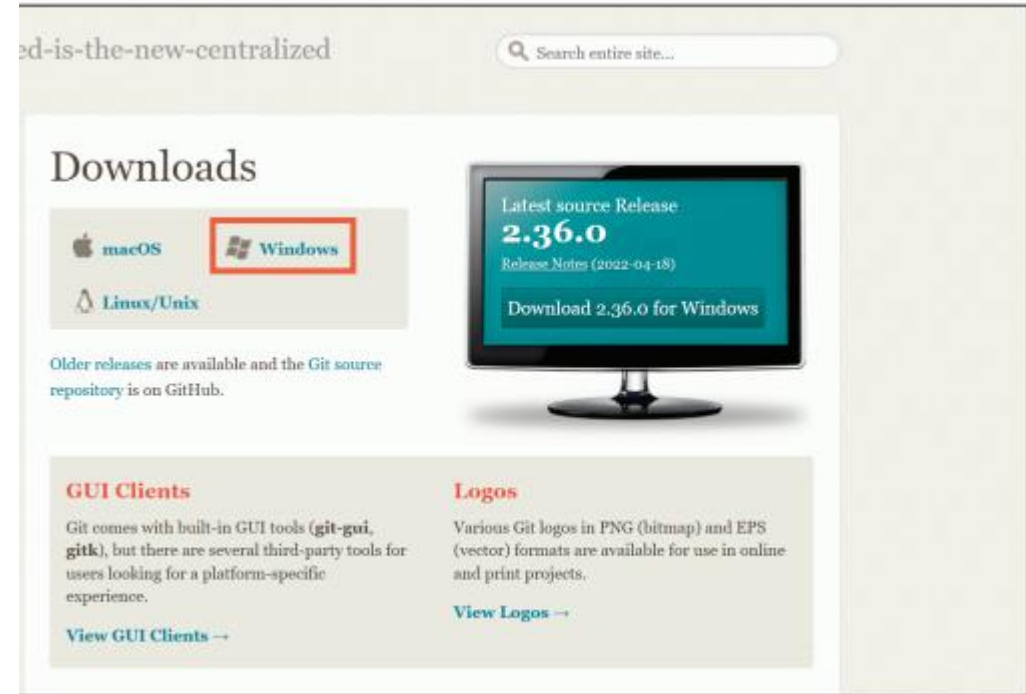


git 설치 , 배포

git 설치

git 설치

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



윈도 운영 체제에 설치할 예정이니 Windows를 선택
그러면 설치 파일이 자동으로 내려받아짐

Download for Windows

[Click here to download](#) the latest (2.40.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 19 days ago, on 2023-04-25.

Other Git for Windows downloads

Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

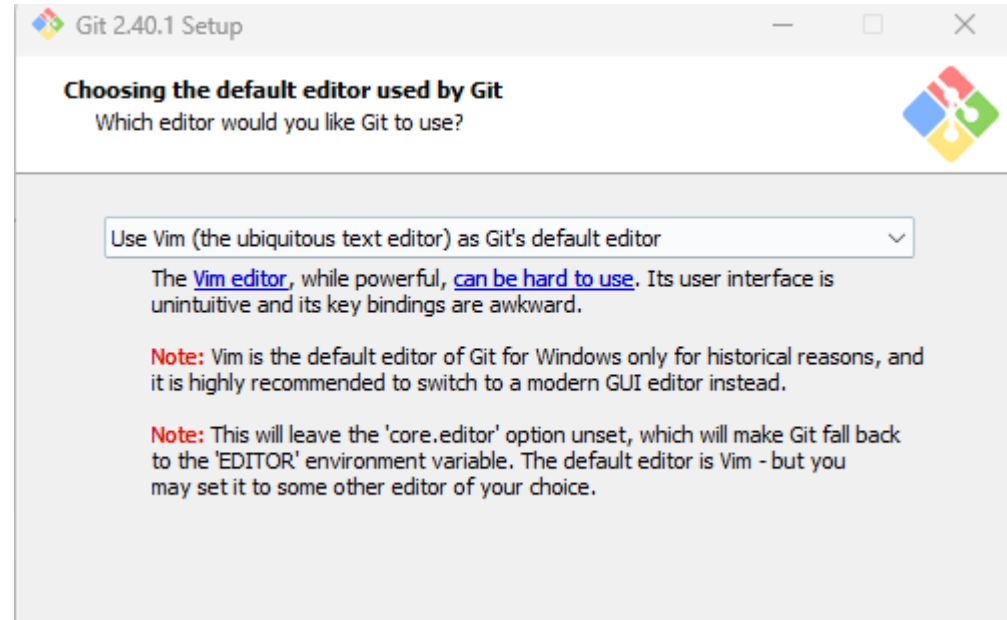
Portable ("thumbdrive edition")

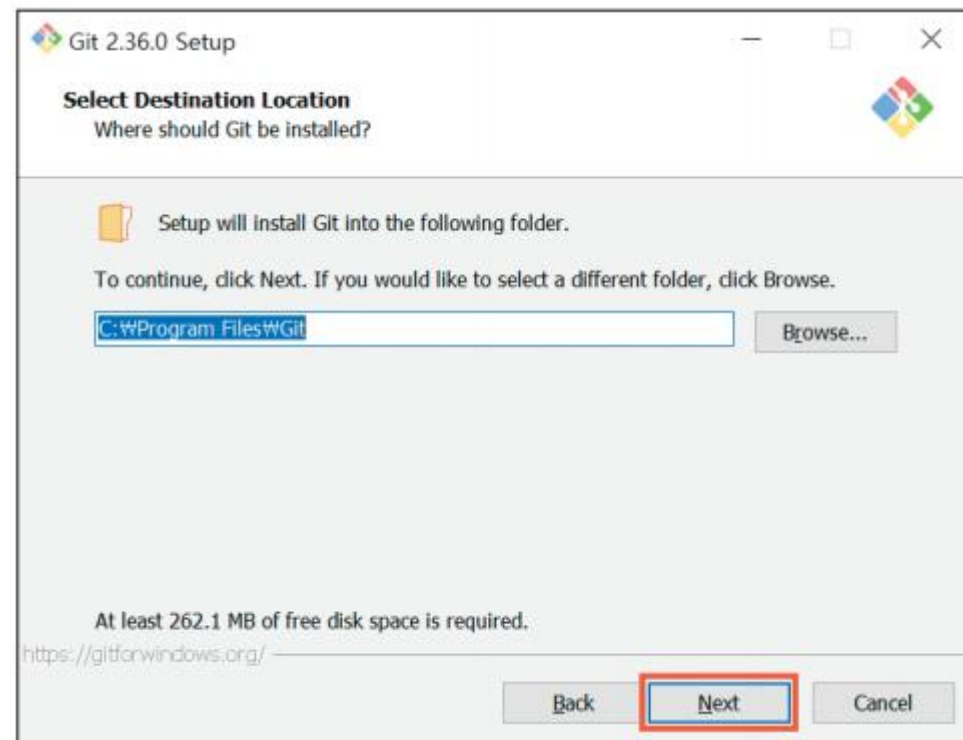
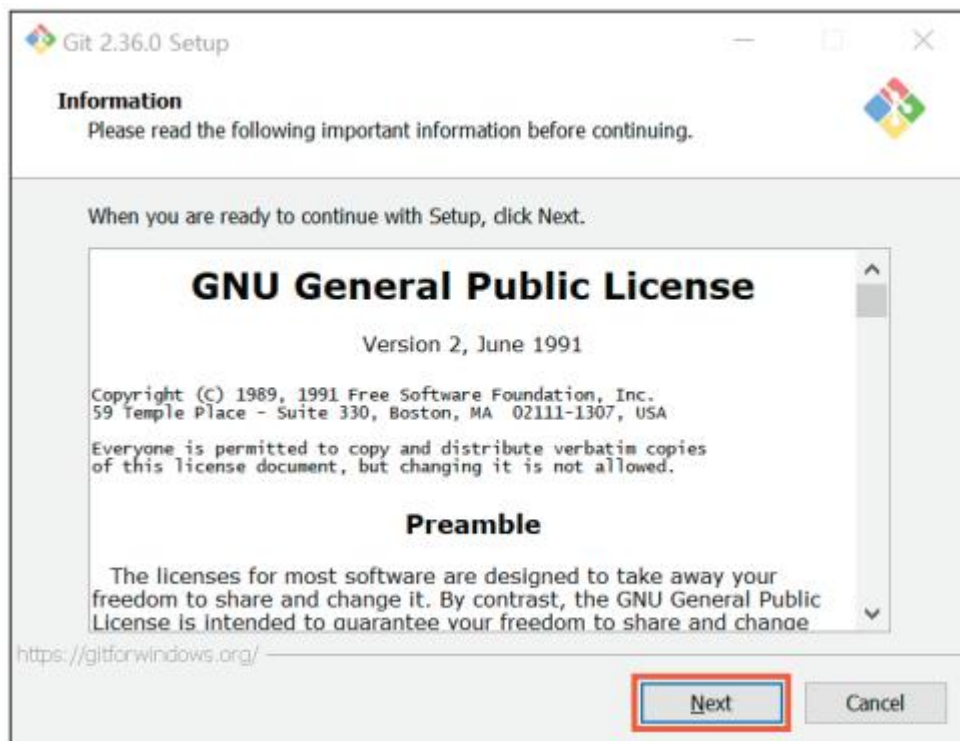
[32-bit Git for Windows Portable.](#)

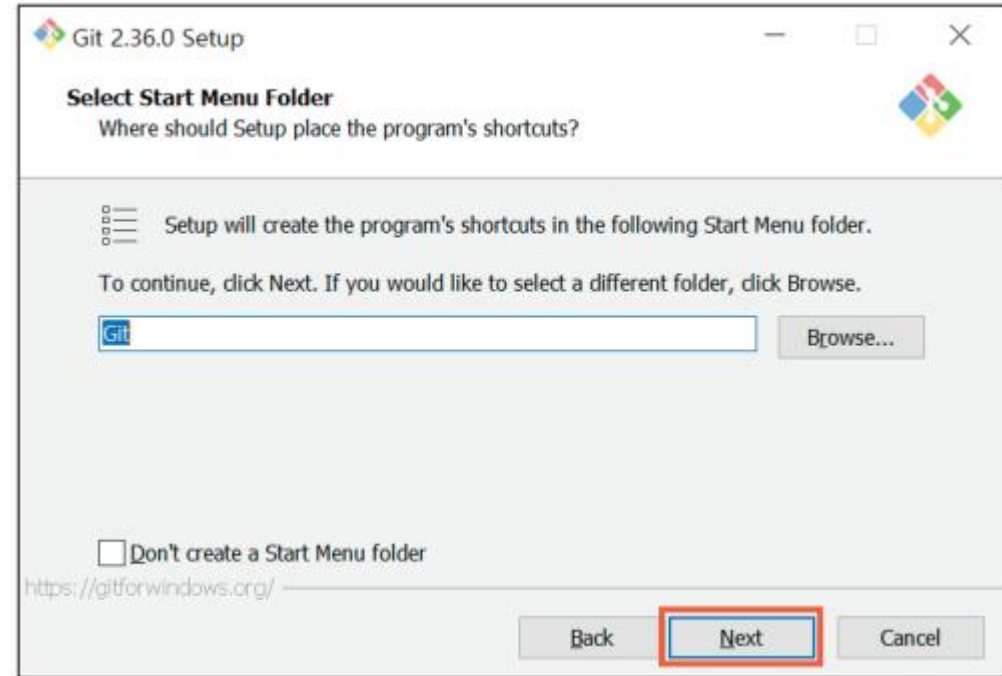
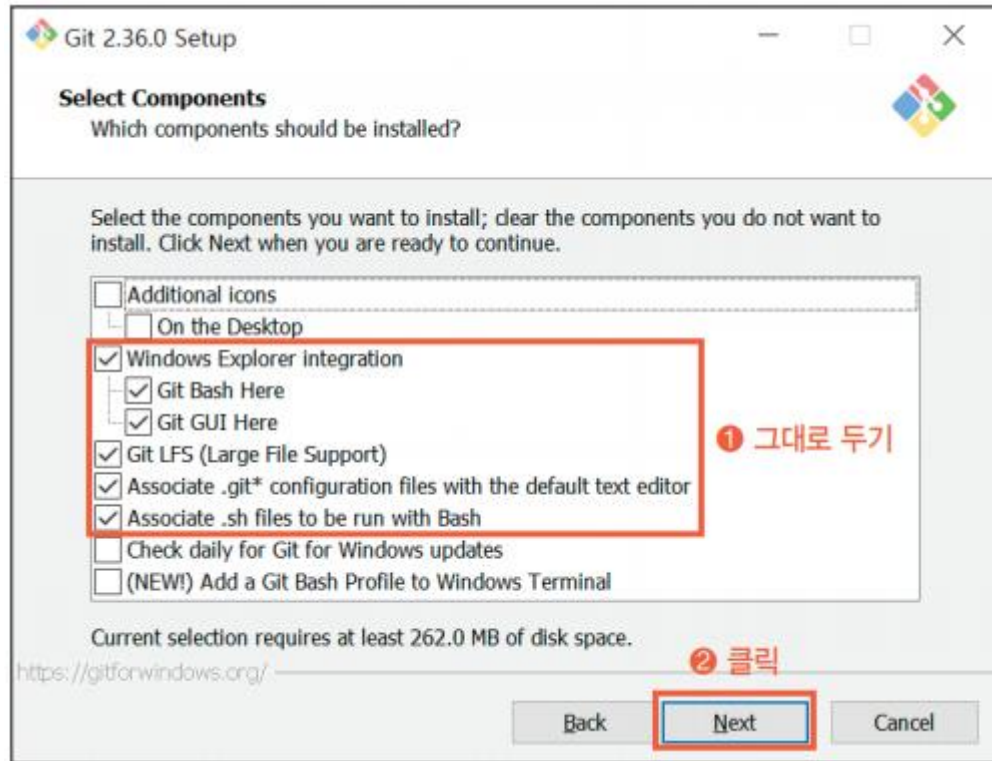
[64-bit Git for Windows Portable.](#)

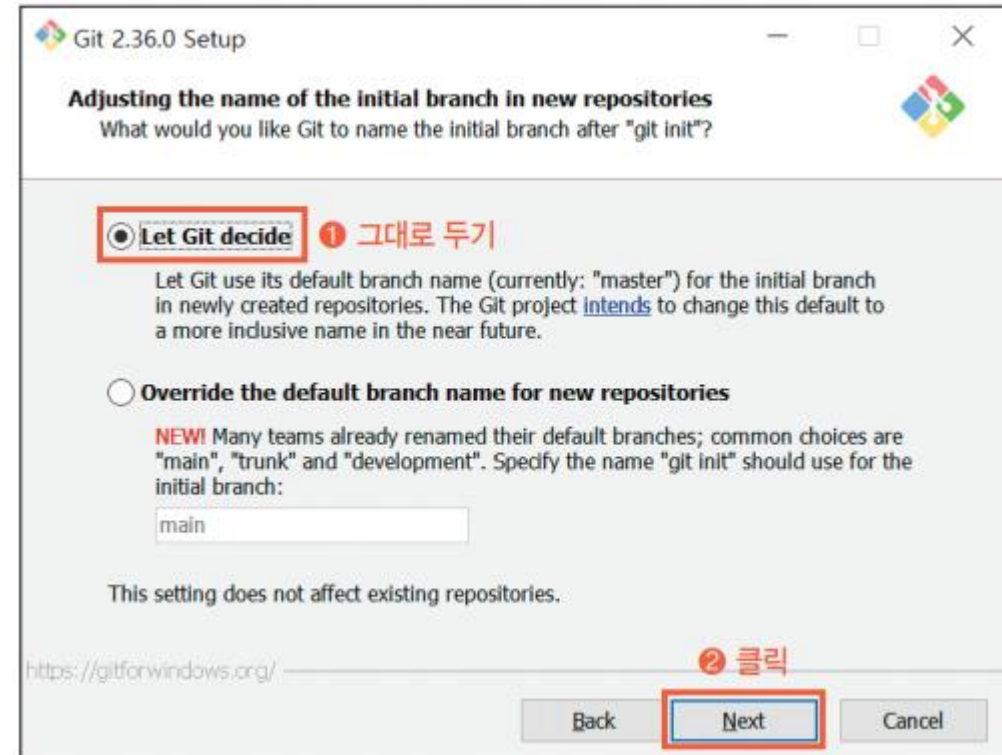
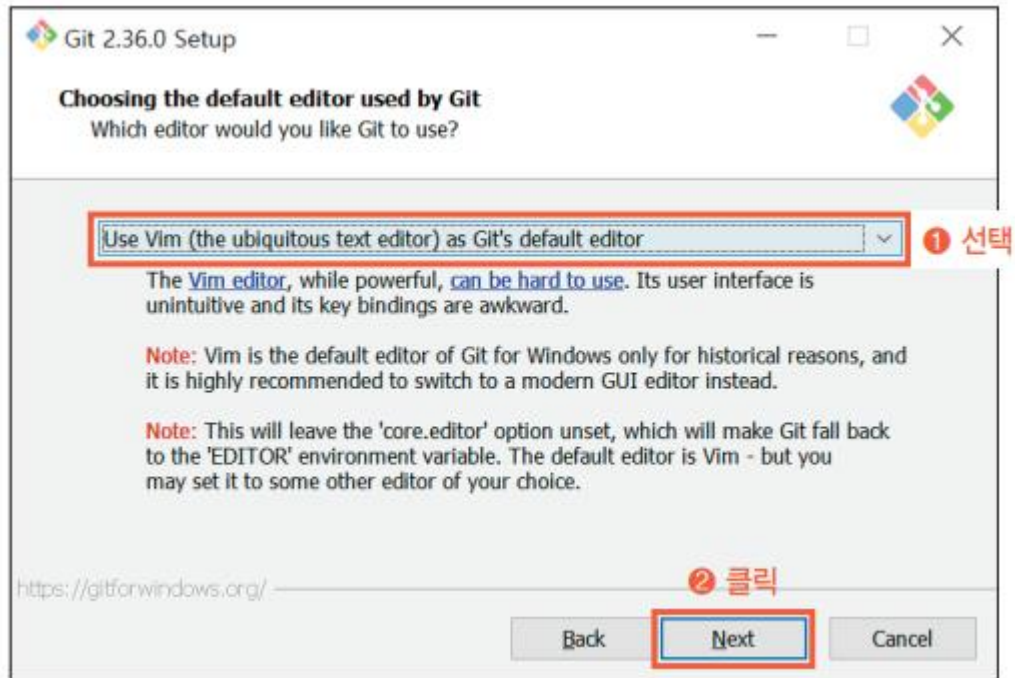
Using winget tool

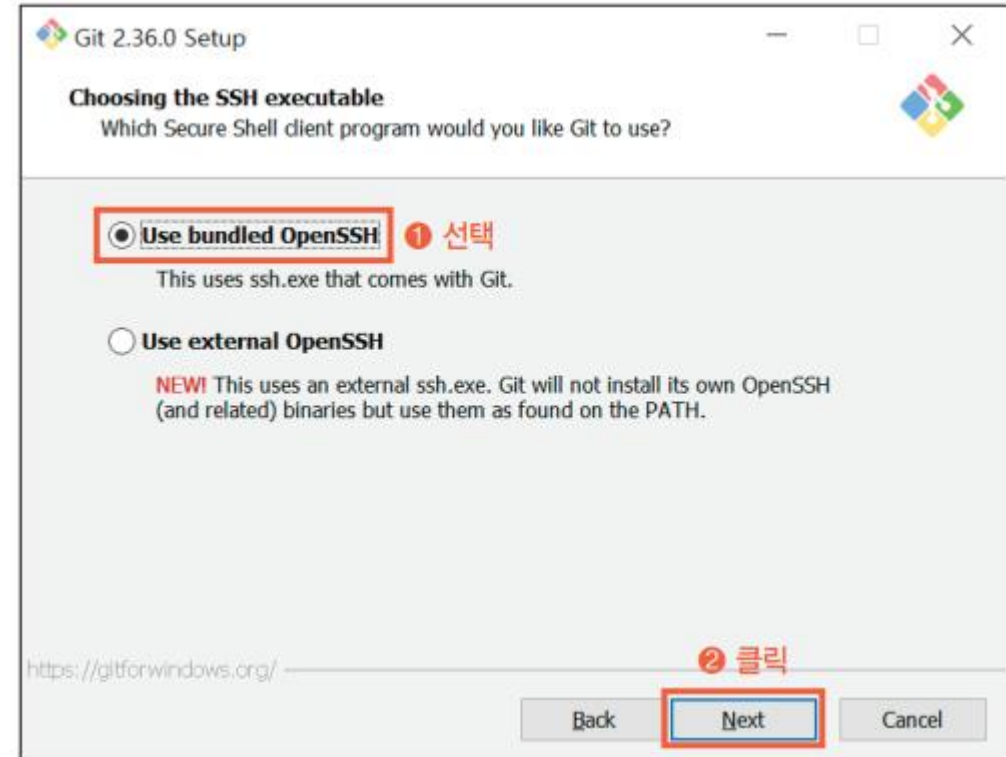
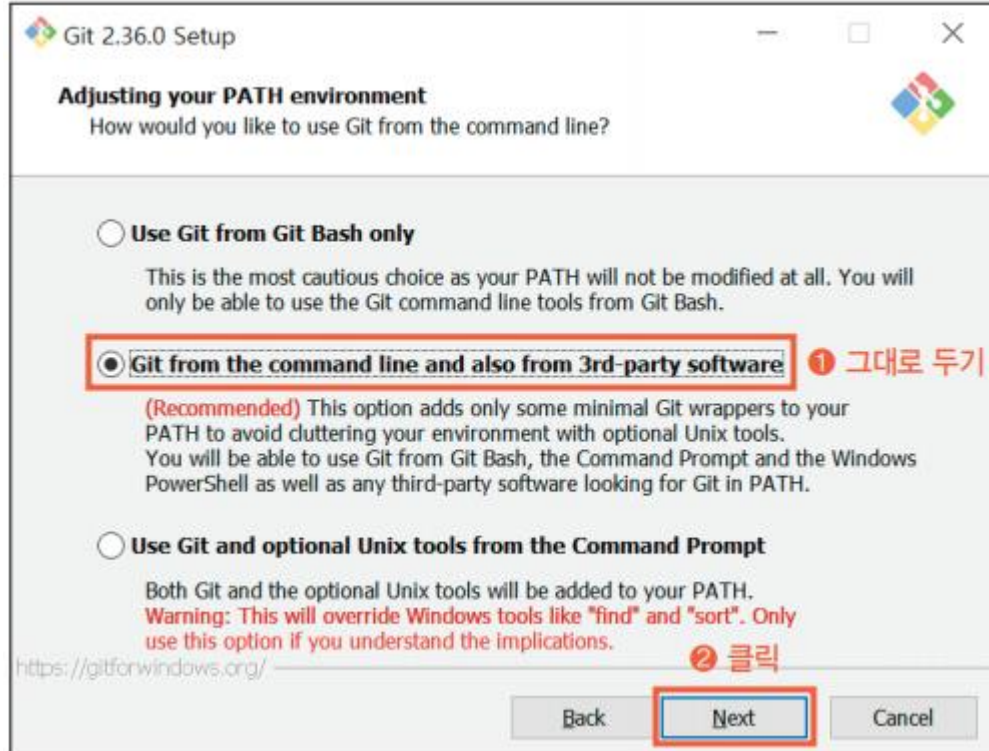
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell

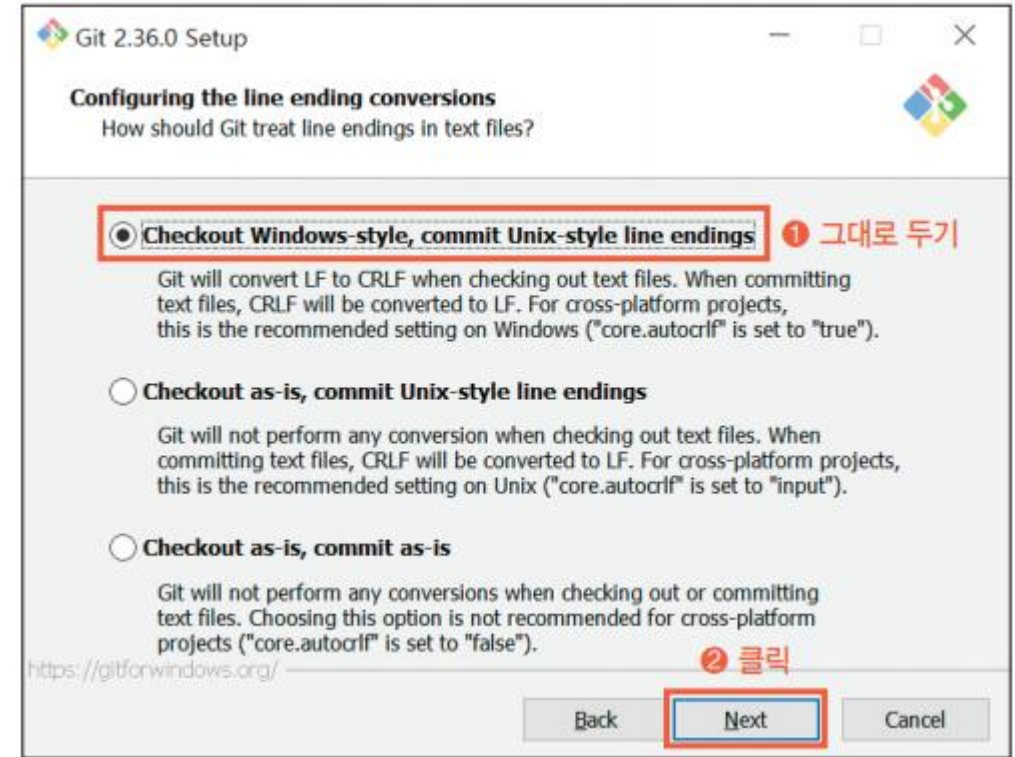
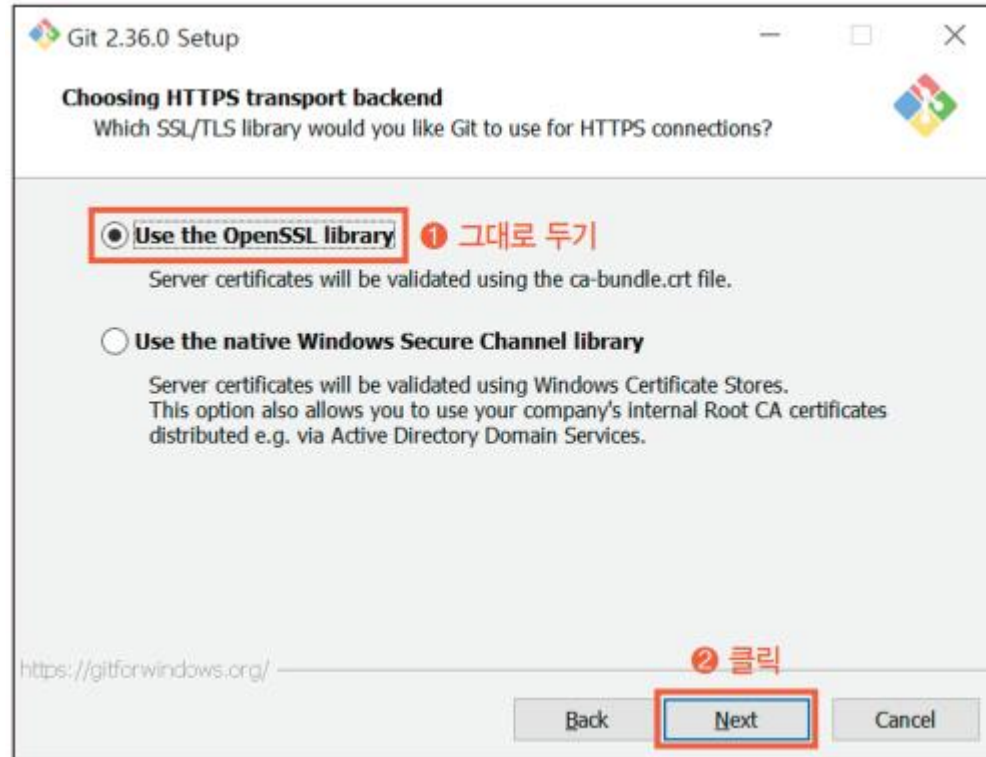


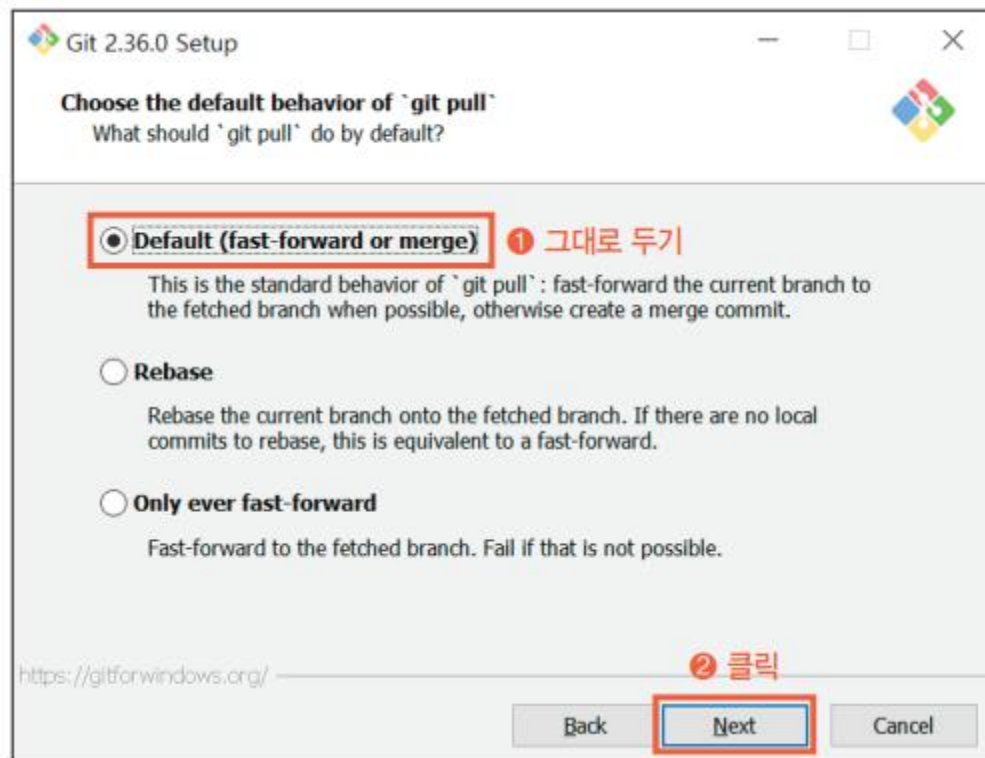
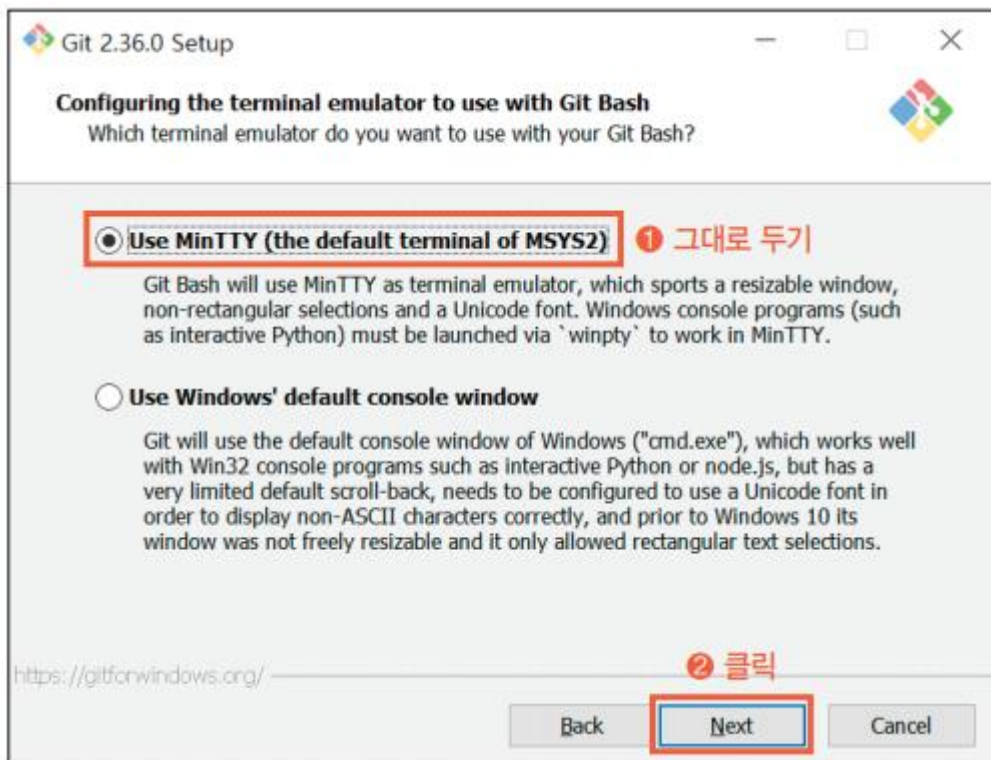


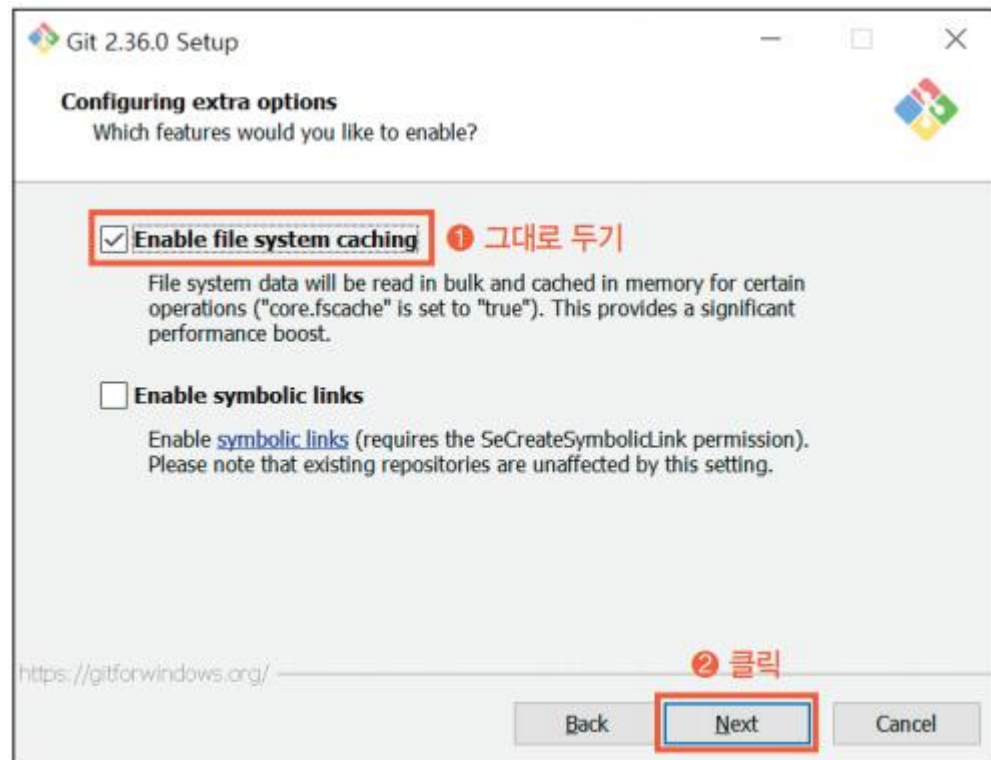
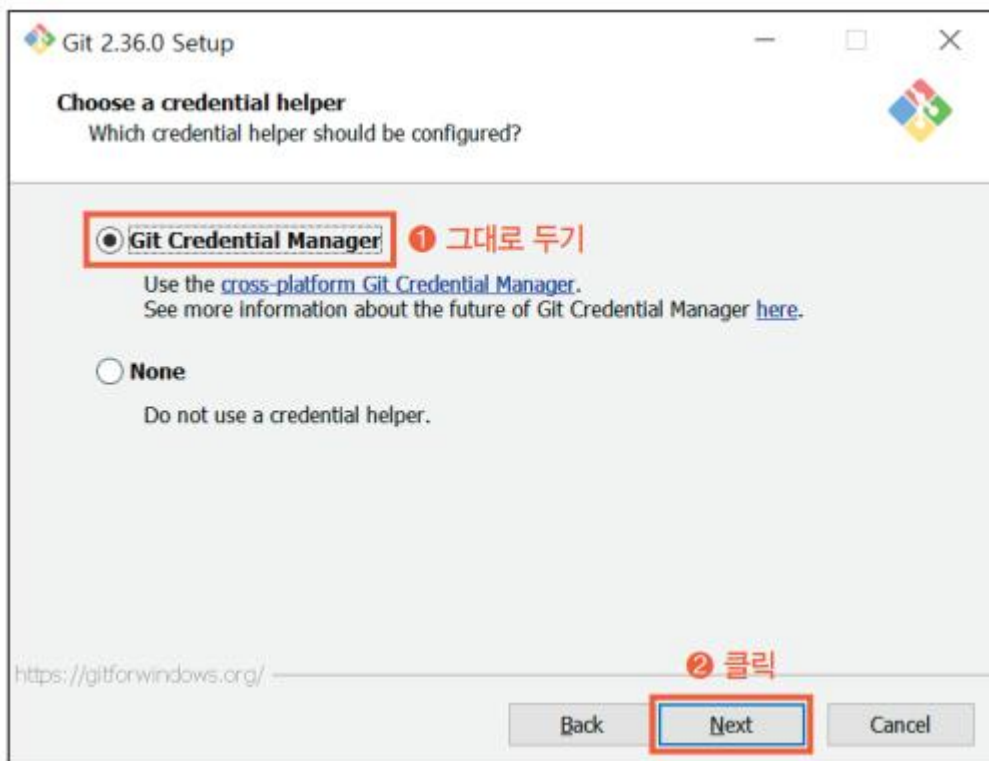


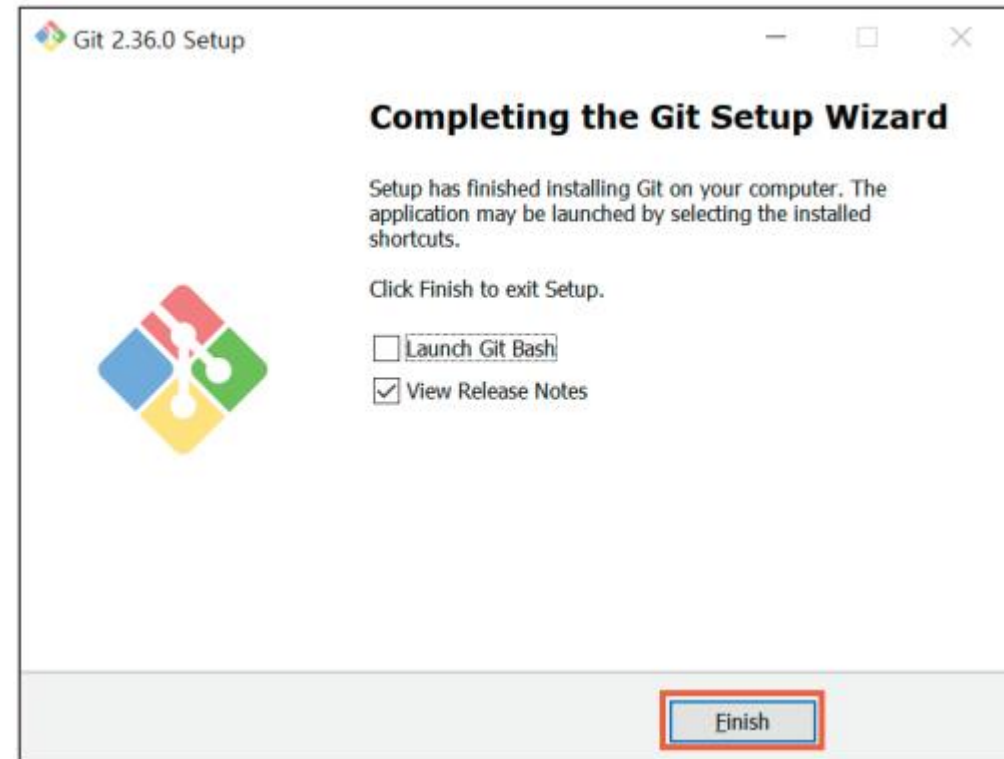
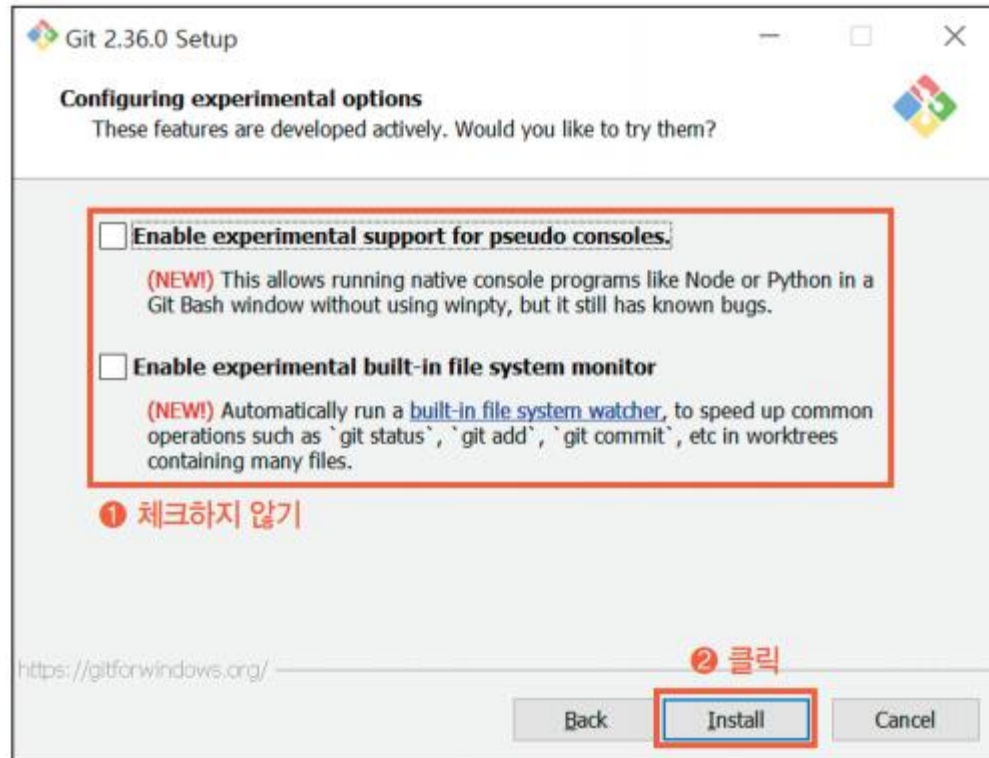












깃허브 회원가입

URL : <https://github.com>



github

원격 저장소란

- 깃에서는 지역 저장소와 원격 저장소(remote repository)를 연결해서 버전 관리하는 파일을 쉽게 백업할 수 있음
원격 저장소는 지역 저장소가 아닌 컴퓨터나 서버에 만든 저장소를 말함
- 원격 저장소는 지역 저장소와 연결되어 있으면서 '백업'과 '협업'이라는 중요한 역할을 함
특히 규모가 큰 프로젝트를 진행한다면 다른 사람과 협업할 때가 많은데, 이때 원격 저장소의 역할은 더욱 중요해짐
- 원격 저장소를 직접 구축할 수도 있지만 만들고 유지하는 것이 쉽지 않으므로 원격 저장소를 제공하는 서비스를 주로 사용
그중에서 깃과 관련해 가장 많이 사용하는 서비스가 바로 깃허브!

github

원격 저장소 만들기

1. 깃허브에 로그인한 후 화면 오른쪽 위에 있는 [+]를 클릭하고 [New repository]를 선택
2. 저장소 이름을 비롯해서 필요한 항목을 기입하고 [Create repository]를 클릭

① **Repository name:** 저장소 이름을 입력

영문과 숫자, 언더바(_), 붙임표(-) 등을 사용할 수 있음
공백이 포함되어 있으면 자동으로 붙임표(-)로 바꿈

② **Description:** 저장소를 소개하는 간단한 설명을 입력

이 부분은 옵션이므로 반드시 입력하지 않아도 됨

③ **Public / Private:** 저장소를 공개로 할지 비공개로 할지 선택

공개 저장소는 주소만 알면 누구나 볼 수 있으나 만일 다른 사람에게 보여 주고 싶지 않은 프로젝트를 관리한다면 저장소를 만들 때 비공개(private)로 하면 됨

The screenshot shows the 'Create a new repository' page on GitHub. It includes the following elements with numbered callouts:

- 1:** Points to the 'Repository name' input field, which contains 'test-1' and a checkmark.
- 2:** Points to the 'Description (optional)' text area.
- 3:** Points to the 'Public' radio button, which is selected. Below it, the 'Private' option is also visible.

Other visible text on the page includes: 'Create a new repository', 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.', 'Owner *' (with a dropdown showing 'jjlaon'), and 'Great repository names are short and memorable. Need inspiration? How about literate-octo-goggles?'.

github

원격 저장소 만들기

- ④ **Add a README:** 저장소를 소개하고 설명하는 내용을 작성하는 README 파일을 자동으로 만들려면 체크
여기에서는 체크하지 않도록 함
- ⑤ **Add .gitignore:** .gitignore 파일은 지역 저장소의 파일을 깃허브의 저장소로 푸시할 때 제외할 파일을 지정해 놓은 것 [▼]를 클릭한 후 어떤 언어와 관련된 것을 .gitignore 파일에 지정할지 선택
예를 들어 C++을 선택한다면 C++에서 사용하는 컴파일된 라이브러리나 실행 파일을 깃에서 무시하도록 .gitignore 파일을 자동으로 만들어 줌
- ⑥ **Choose a license:** 오픈 소스 프로젝트를 위한 저장소를 만들 때 해당 오픈 소스의 라이선스를 선택

The screenshot shows the GitHub 'Initialize this repository' screen. It has three main sections, each with a blue box and a circled number on the right:

- 4 Add a README file:** Includes a checkbox and a text area for a project description. A link 'Learn more.' is present.
- 5 Add .gitignore:** Includes a dropdown menu for '.gitignore template: None'. A link 'Learn more.' is present.
- 6 Choose a license:** Includes a dropdown menu for 'License: None'. A link 'Learn more.' is present.

Below these sections, there is a note: 'You are creating a public repository in your personal account.' and a 'Create repository' button.

github

원격 저장소 만들기

3. 원격 저장소가 만들어지면서 즉시 저장소 페이지로 이동
아직 아무 파일도 들어 있지 않고, 맨 위에는 저장소에 접속하는 방법이 나타남
저장소에 접속할 때는 HTTPS 방식이나 SSH 방식 중에서 선택할 수 있는데 우선 우리는 HTTPS 방식을 사용해 접속할 것
4. 화면에 나타난 HTTPS 주소를 사용해 언제든지 깃허브 저장소에 접속할 수도 있고 파일을 올릴 수 있음
깃허브 원격 저장소 주소만 알고 있다면 어디에서든 지역 저장소를 백업하거나 다른 사람과 협업할 수 있음

원격 저장소의 HTTPS 주소는 다음과 같은 형태

```
https://github.com/아이디/저장소명
```

github

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *



Repository name *

Great repository names are short and memorable. Need inspiration? How about **stunning-giggle?**

Description (optional)

☐ Public

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

☒ Private

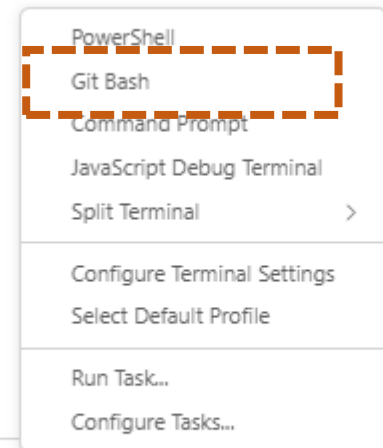
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

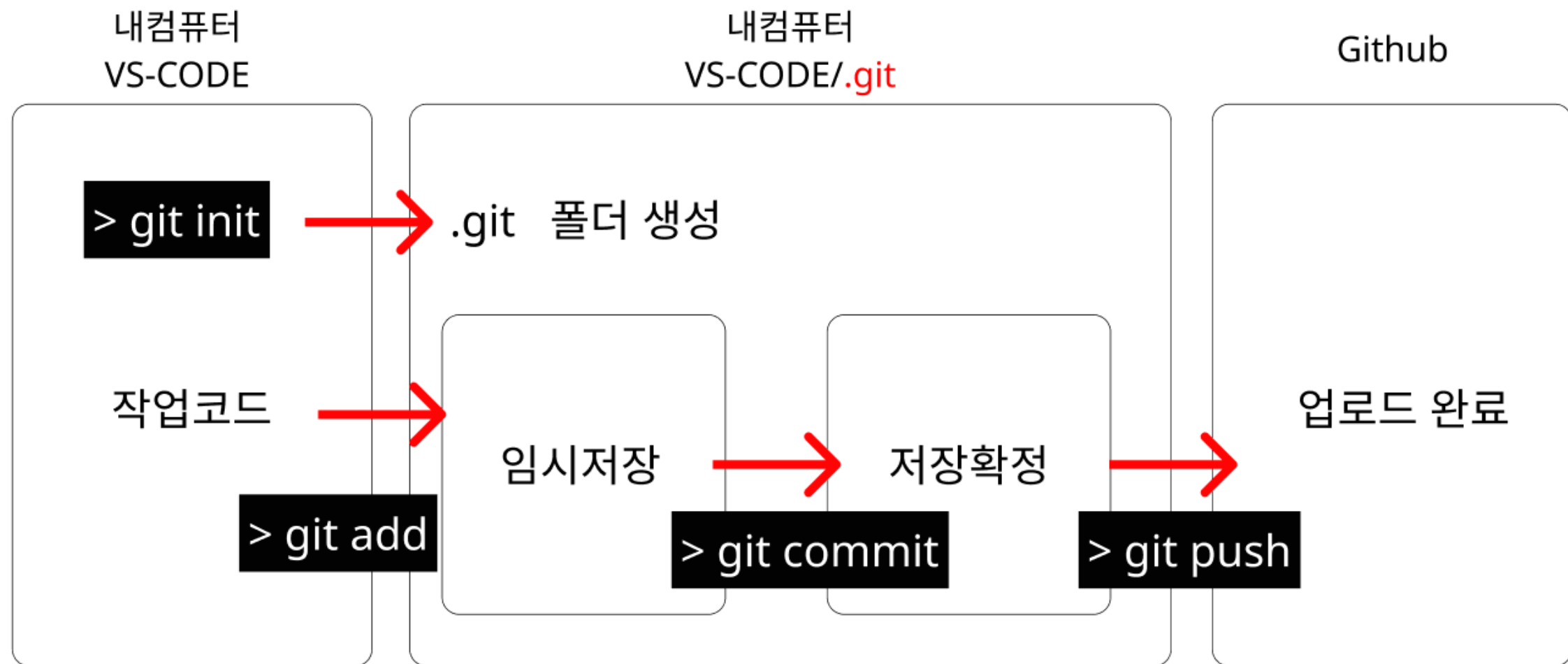
This is where you can write a long description for your project. [Learn more about READMEs.](#)

vscode - 터미널



powershell + - [icon] [icon] ...

- > git init : git 폴더 생성
- > git add : 임시 저장
- > git commit : 저장확정
- > git push : 업로드 완료



Git VSCode 터미널



The screenshot shows the VS Code interface with the '터미널' (Terminal) tab selected. The terminal output shows a Windows command prompt session. The user has navigated to 'D:_국기수업\01_html_css\test' and attempted to run 'git clone https://github.com/honeypunch97/teamproject'. An error message states that 'git' is not a recognized command. A context menu is open over the terminal, with 'Git Bash' highlighted. The menu also includes options like 'PowerShell', 'Command Prompt', 'JavaScript 디버그 터미널', and '터미널 분할'.

```
문제  출력  디버그 콘솔  터미널
Microsoft Windows [Version 10.0.19044.2965]
(c) Microsoft Corporation. All rights reserved.

D:\_국기수업\01_html_css\test>cd..

D:\_국기수업\01_html_css>git clone https://github.com/honeypunch97/teamproject
'git'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

D:\_국기수업\01_html_css>
* 복원된 기록

Microsoft Windows [Version 10.0.19044.2965]
(c) Microsoft Corporation. All rights reserved.

D:\_국기수업\01_html_css\test>
```

github

> git config --global user.name "사용자이름"

git config --global user.name "깃이름"

(로컬 git에 이름 설정)

> git config --global user.email [사용자이메일@...com](#)

(로컬 git에 메일주소 설정)

github

> **git init** (작업폴더 내에 .git폴더 생성)

> **git remote add origin (repository URL)** : .git과 github를 연결

```
$ git remote add origin 복사한 주소 붙여넣기
```

> **git remote -v** : 원격 저장소 연결됐는지 확인하려면 다음처럼 git remote 명령에 -v 옵션을 붙여서 입력

> **git add .** : (.git폴더 작업 내용 모두 추가) - add 한칸띄우기 .

origin https://github.com/eun00/test0.git (fetch)

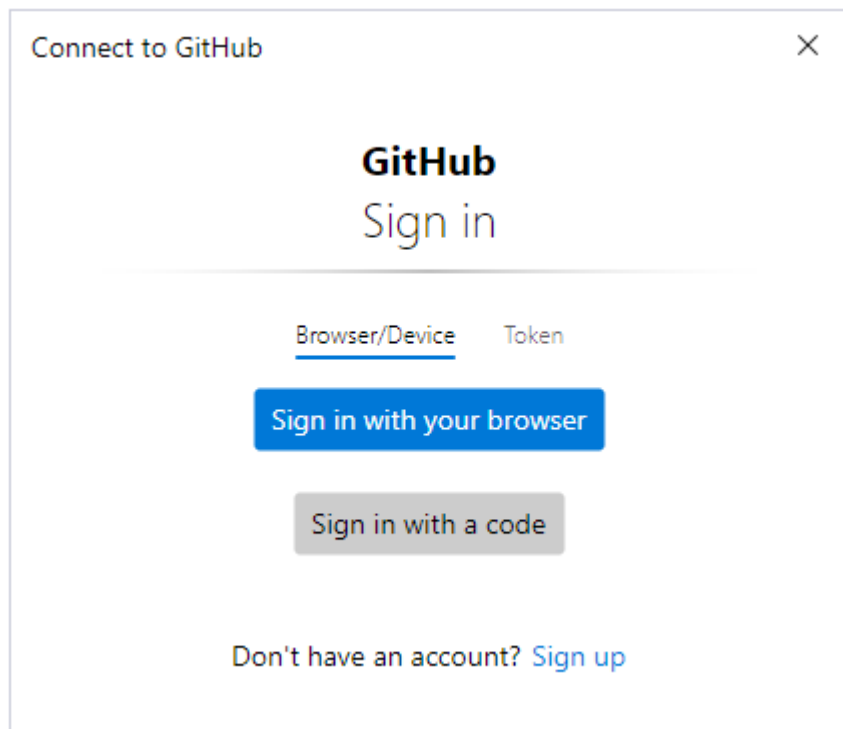
origin https://github.com/eun00/test0.git (push)

> **git status** : 작업폴더 확인

> **git commit -m '프로젝트명'** : (확정 저장및 코멘트 작성)

> **git push origin master** : (origin repository 의 master브랜치에 작업 내용 업로드)

github



Github 올리기

```
EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git config --global user.name "[redacted]"

EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git config --global user.email "[redacted]"

EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git init
Reinitialized existing Git repository in D:/_국기수업/01_html_css/test/.git/

EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git remote add origin https://github.com/eun00/test.git

EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git remote -v
origin https://github.com/eun00/test.git (fetch)
origin https://github.com/eun00/test.git (push)


EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git add .


EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git status
On branch master
nothing to commit, working tree clean


EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git commit -m "수정내용작성"
On branch master
nothing to commit, working tree clean

EZEN-216T@5-t MINGW64 /d/_국기수업/01_html_css/test (master)
$ git push origin master
Enumerating objects: 37, done.
```

Github 완료


 master ▾


 1 branch

 0 tags


Go to file




Add file ▾

 Code ▾

 eun00 배경수정

5b0993b 3 hours ago

 3 commits

 css	배경수정	3 hours ago
 images	프로젝트설명	5 hours ago
 index.html	프로젝트설명	5 hours ago

Add a README with an overview of your project.

Add a README

파일 수정 다시 업로드

- > git add. : (.git폴더 작업 내용 모두 추가)
- > git status : 작업폴더 확인
- > git commit -m '수정내용설명' : (수정내용 코멘트 작성)
- > git push origin master : (작업 내용 업로드)

저장소 수정 또는 삭제후 다시 업로드 할경우

리모트 확인

```
> git remote -v
```

기존 리모트 작업 삭제

```
> git remote rm origin
```

다시 추가

```
> git remote add origin https://~.git
```

맥에서 설치

런치패드에서 터미널 검색해서 열고

<https://brew.sh/> 긴 설치명령어를 터미널에 그대로 입력

- homebrew 설치 약간의 시간이 소요

- password 입력 맥북비번

homebrew 설치 끝나면 터미널에 `brew install git` 입력하면 git 설치 끝

vs Code 실행해서 `shift + command + p` 눌러서

Install "code" from VSCode commander 검색

그리고 터미널 들어가서

`git config --global core.editor "code --wait"`

////////////////////////////////////

homebrew 로 안할꺼면

<https://sourceforge.net/projects/git-osx-installer/>

가서 다운로드

파일 더블클릭하면 X

파일 마우스 우클릭 눌러서 설치

github.io

사이트 연결

Build and deployment

Source

Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. [Learn more.](#)

None Save

master 선택

Select branch

Select branch

master

✓ None

Try GitHub Enterprise risk-free for 30 days


[Learn more about the visibility of your GitHub Pages site](#)

사이트연결 - master 하고 나면 몇분정도 소요될수 있어요

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://eun00.github.io/site1/>

Last deployed by  eun00 6 minutes ago

 Visit site

...


Build and deployment

Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

 master ▾

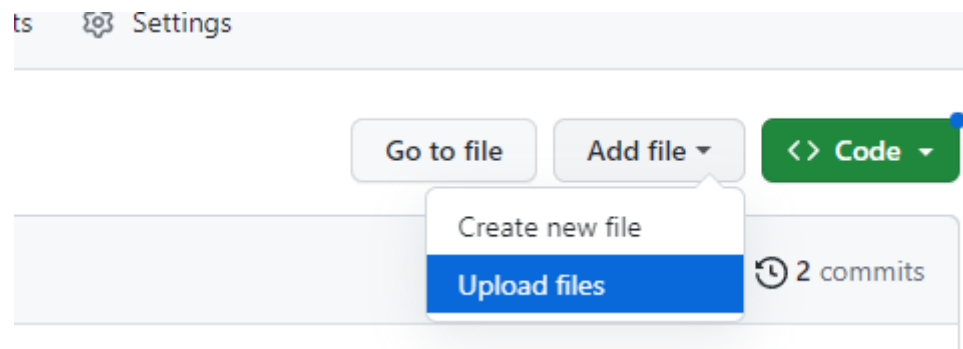
 / (root) ▾

Save

[Learn how to add a Jekyll theme to your site.](#)

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.

사이트연결 안될경우 : 드래그 이동하기

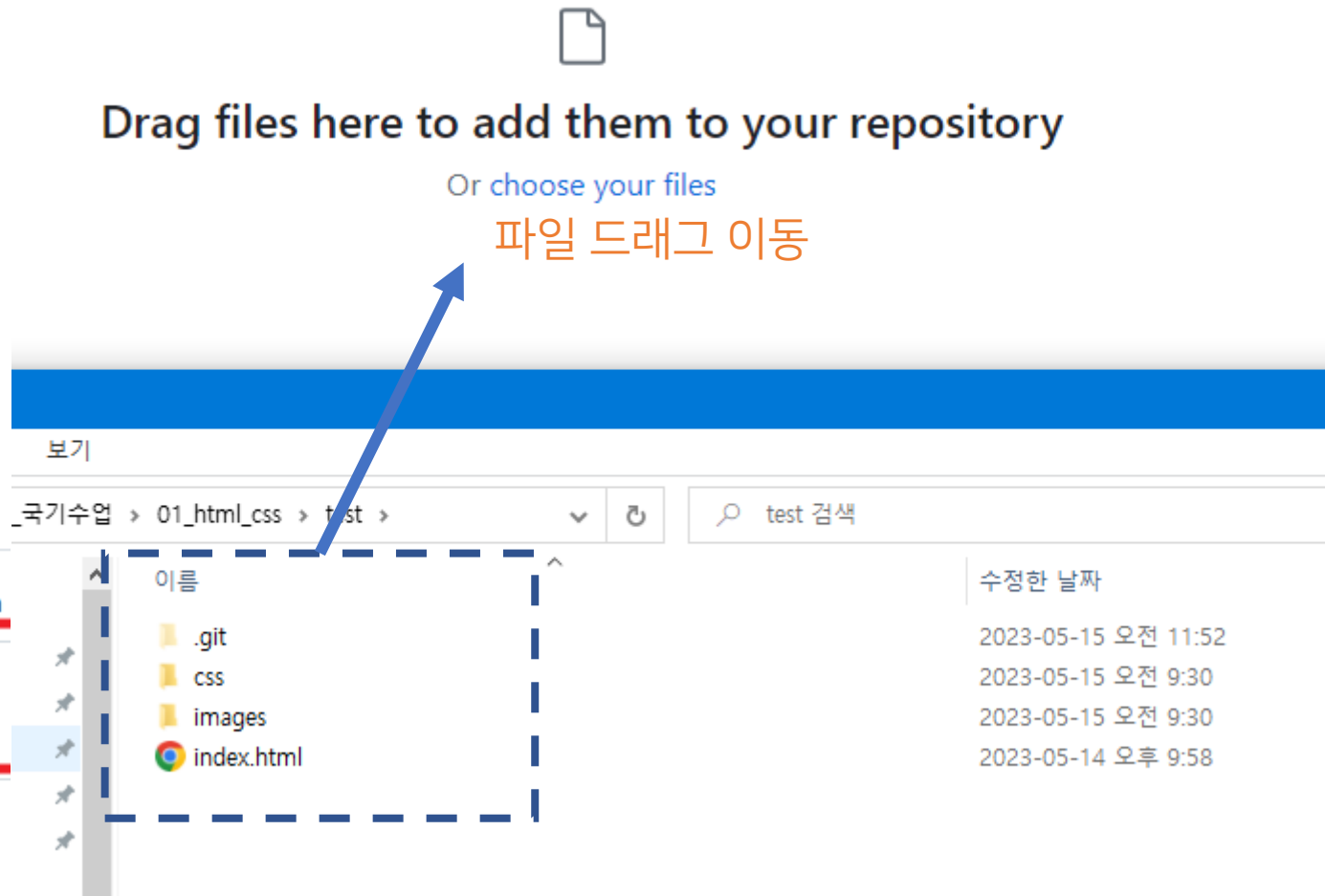


GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from

Your site is live at <https://eun00.github.io/site1/>

Last deployed by eun00 6 minutes ago



내려받기 : git clone

git clone <https://github.com/eun00/test.git>

git clone <https://github.com/eun00/test.git> 새로운이름

폴더 생성 : md 폴더명

폴더 이동 : cd 폴더명

상위폴더 : ../

폴더리스트 확인 : dir

폴더삭제 : del 폴더명

