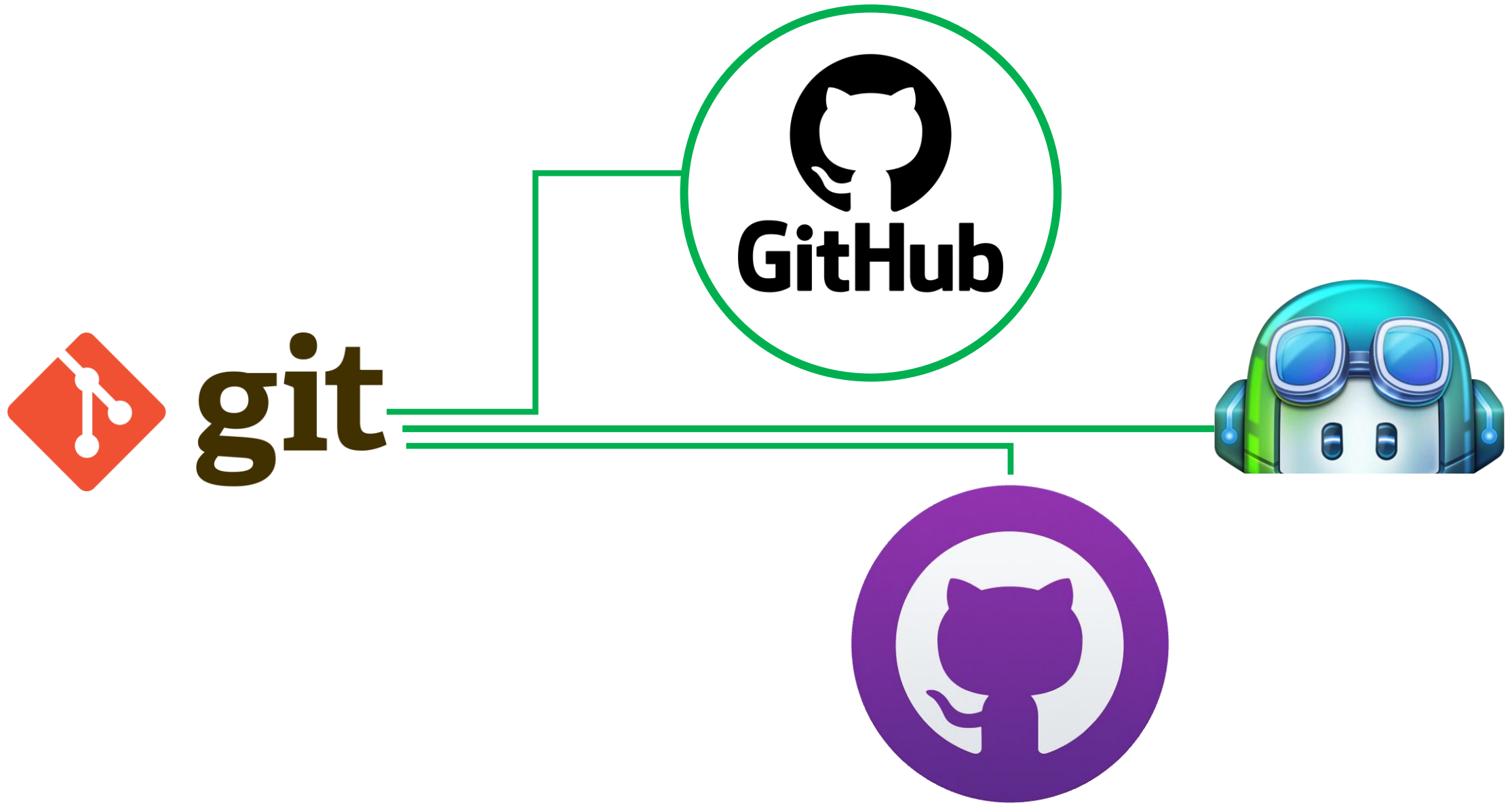


실무에서 활용하는 Git 특강

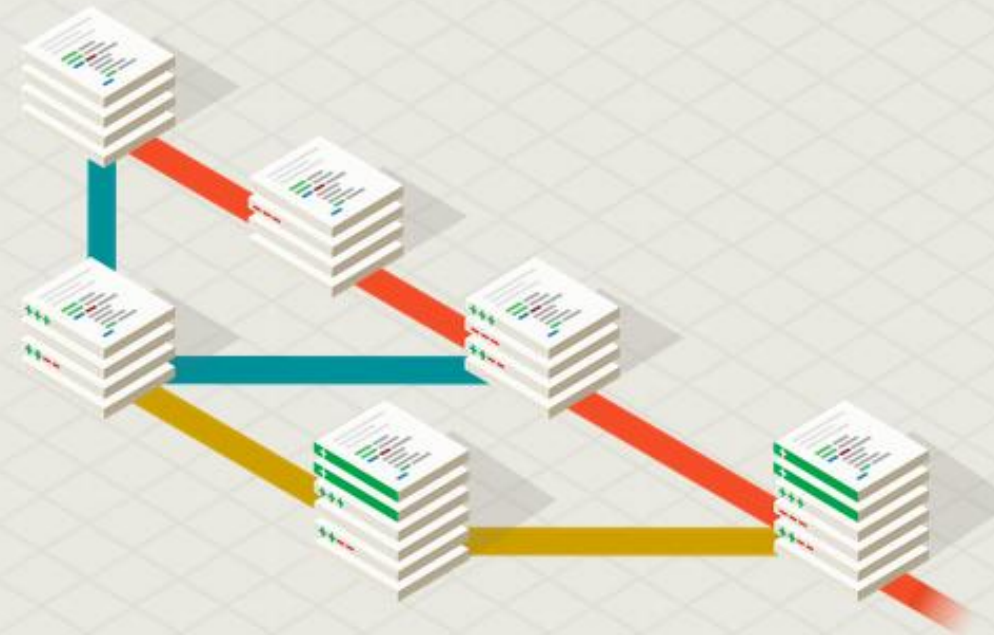


Git이란 무엇인가?

- 분산 버전 관리 시스템.
- 소스 코드의 변경 사항을 추적하고 여러 개발자들이 동시에 작업할 수 있게 도와 줌
- 주요 기능 : 소스 코드의 버전 관리, 브랜치(branch) 관리, 내용 병합, 충돌 해결 등

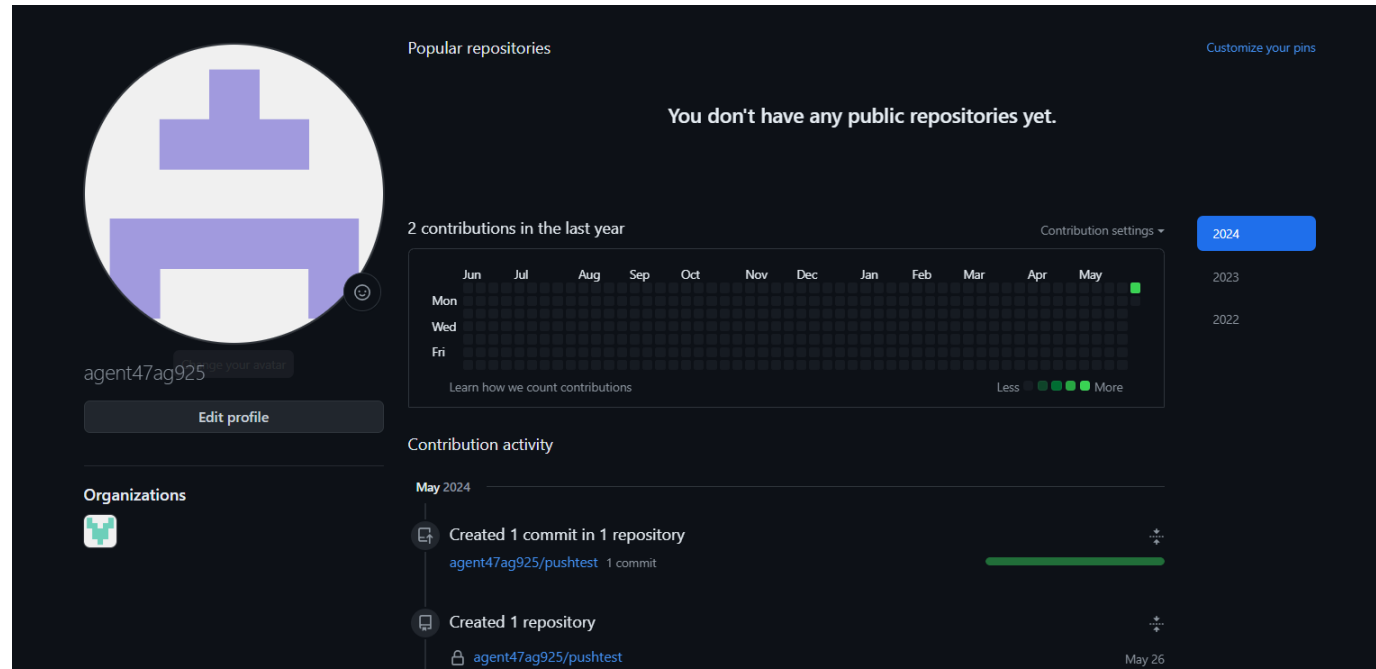
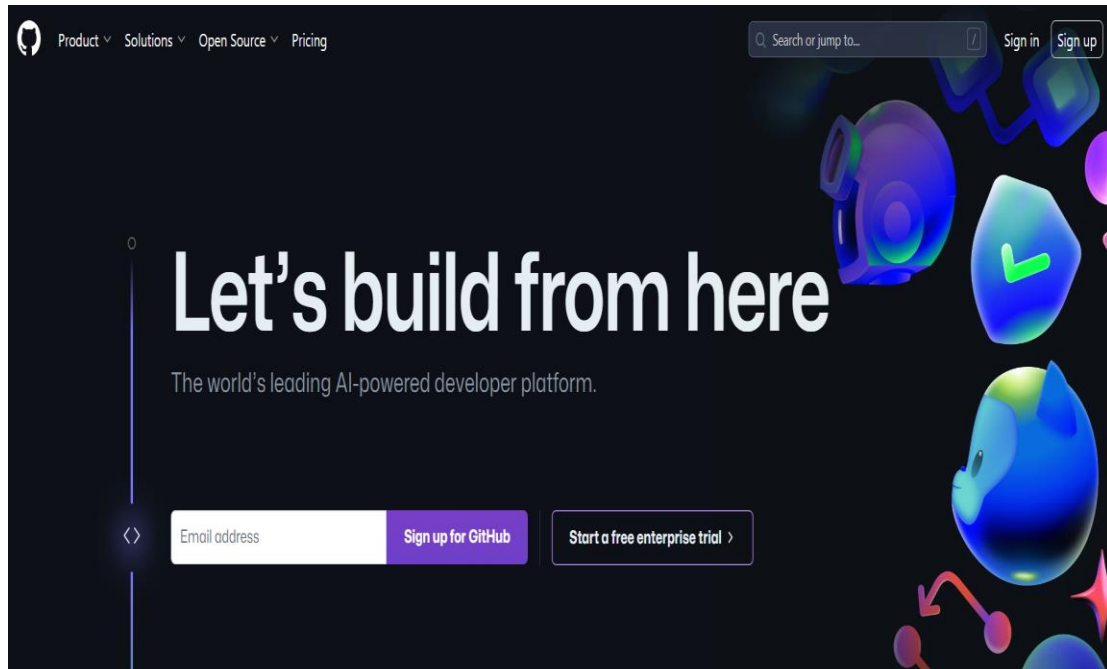
Git is a 무료 오픈 소스 distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is 배우기 쉽고 and has a 엄청 가볍고 빠르게 동작함 performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like 가벼운 로컬 브랜칭 , convenient staging areas, and 편리하게 여러 개의 워크플로우를 만들 수 있음



GitHub란 무엇인가?

- <https://github.com/>
- GitHub는 Git을 기반으로 하는 웹 호스팅 서비스
- 소스 코드 저장소를 온라인에서 관리할 수 있게 해 줌
- 주요 기능 : 원격 저장소 제공, 협업 관리, 이슈 추적, 프로젝트 관리, Actions



GitHub와 비슷한 서비스는?

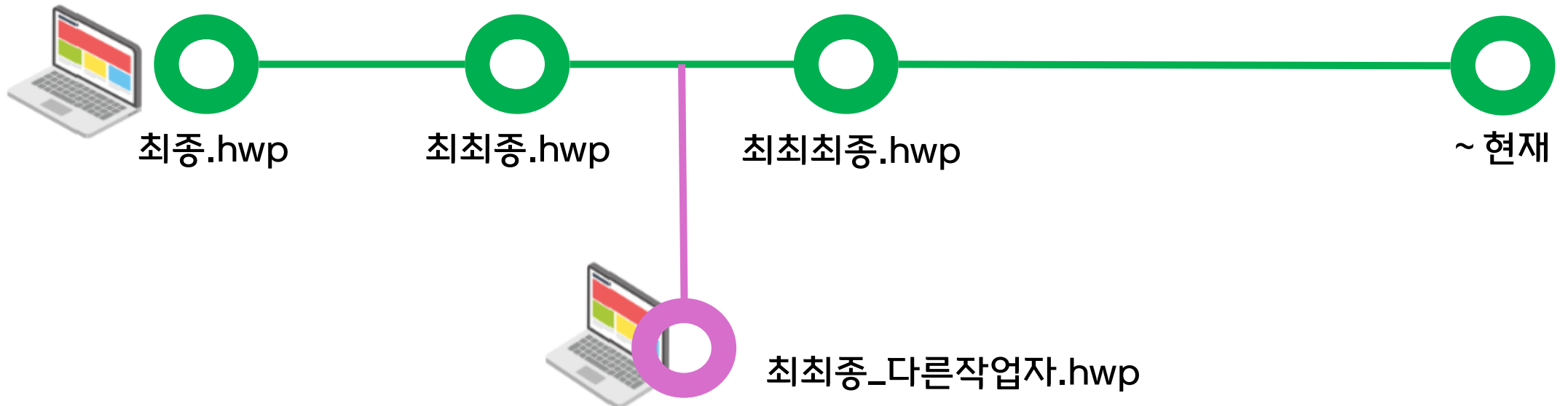
- Git 시스템을 기반으로 온라인 저장소와 소스코드 협업을 제공하되, 특색 있는 서비스들이 많음

GitLab	Bitbucket	Source Forge	AWS Code Commit	Azure Repos
<ul style="list-style-type: none">온라인 저장소CI/CD 파이프라인이슈 트래킹코드 리뷰	<ul style="list-style-type: none">온라인 저장소CI/CD 파이프라인	<ul style="list-style-type: none">온라인 저장소프로젝트 호스팅이슈 트래킹	<ul style="list-style-type: none">온라인 저장소AWS 다른 서비스와 통합이 용이	<ul style="list-style-type: none">온라인 저장소CI/CD 파이프라인

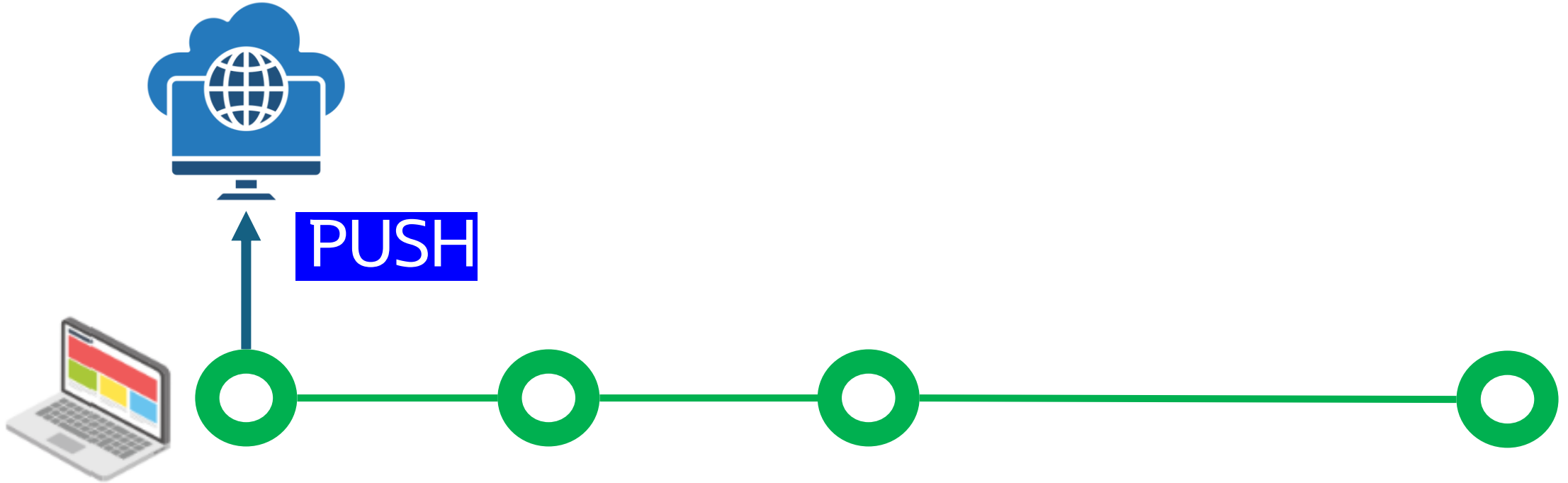


Git의 개념

- 저장소(repo) : 프로젝트 파일과 변경 이력을 관리하는 디렉토리. 로컬저장소와 원격저장소로 나누어짐
- 브랜치(branch) : 저장소에서 작업하는 워크플로우. 기본 브랜치는 main이나 master로 불리며,
필요하면 별도의 브랜치를 만들어 작업할 수 있음
- 커밋(commit) : 파일의 변경 사항을 저장소에 기록하는 작업. 특정 시점의 코드 상태를 저장하여,
추후에 돌아갈 수(Rollback) 있음
- 병합(merge) : 2명 이상, 혹은 원격 저장소와 로컬 저장소의 같은 파일을 하나의 내용으로 합치는 작업.

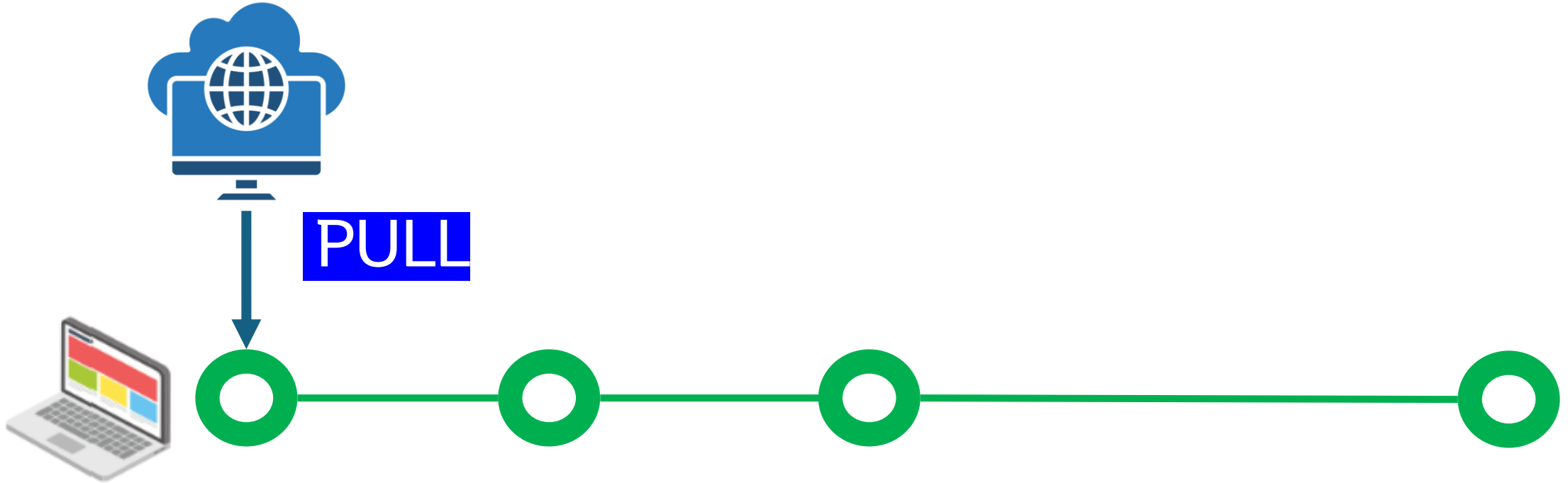


Git의 주요 행동 -push



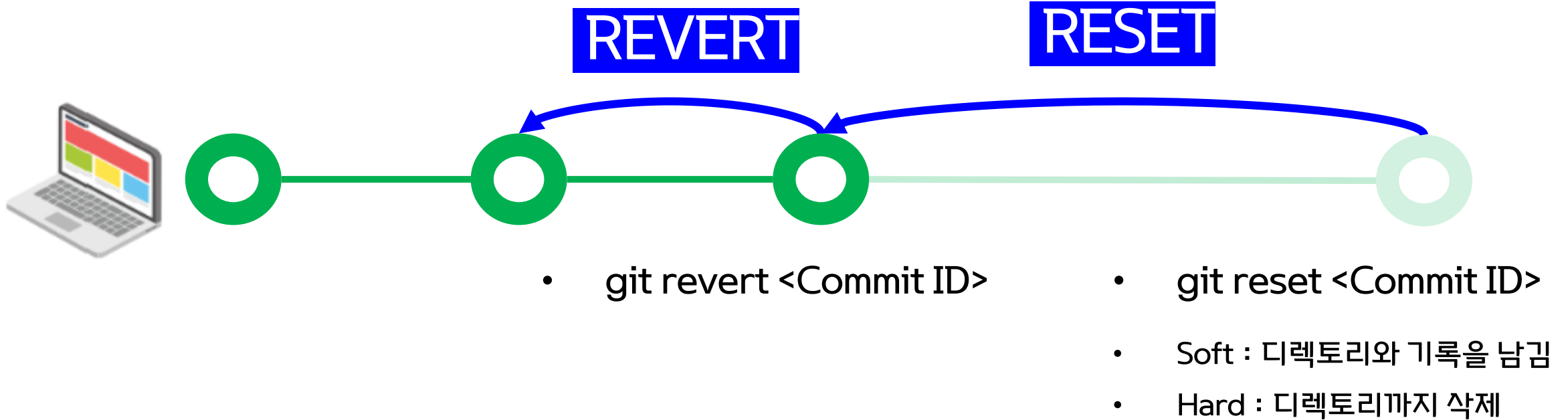
- `git init`
- `git add .`
- `git commit -m “커밋 메시지”`
- `git push origin main(master)`

Git의 주요 행동 -pull(pull request)



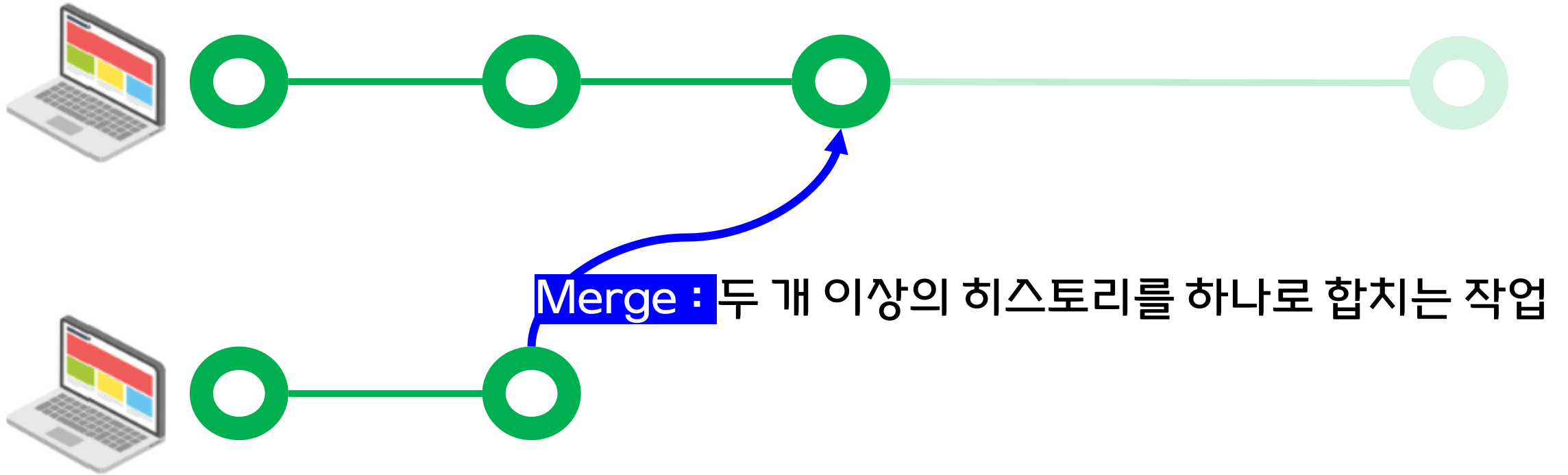
- `git pull origin main(master)`
 - `git pull --allow-unrelated-histories origin main`
- `git fetch origin main(master)`
- `git merge origin main(master)`

Git의 주요 행동 - 시점 변경



Git의 주요 행동 - 시점 변경

- `git merge <Commit ID>`



GitHub 계정 만들기 실습

<https://github.com/>

Git 계정 세팅하기

- 로그인 방법 : github의 계정으로 로그인하여 저장소에 접근
- 액세스 토큰 만들기 : github의 설정에서 액세스 토큰 만들어 저장소에 접근
- SSH key 방법 : ssh 키를 만들고, 기기에서 이를 인증하여 저장소에 접근

Git Bash 터미널 설치

Downloads



Older releases are available and the [Git source repository](#) is on GitHub.



Download for Windows

[Click here to download](#) the latest (**2.45.1**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **12 days ago**, on 2024-05-14.

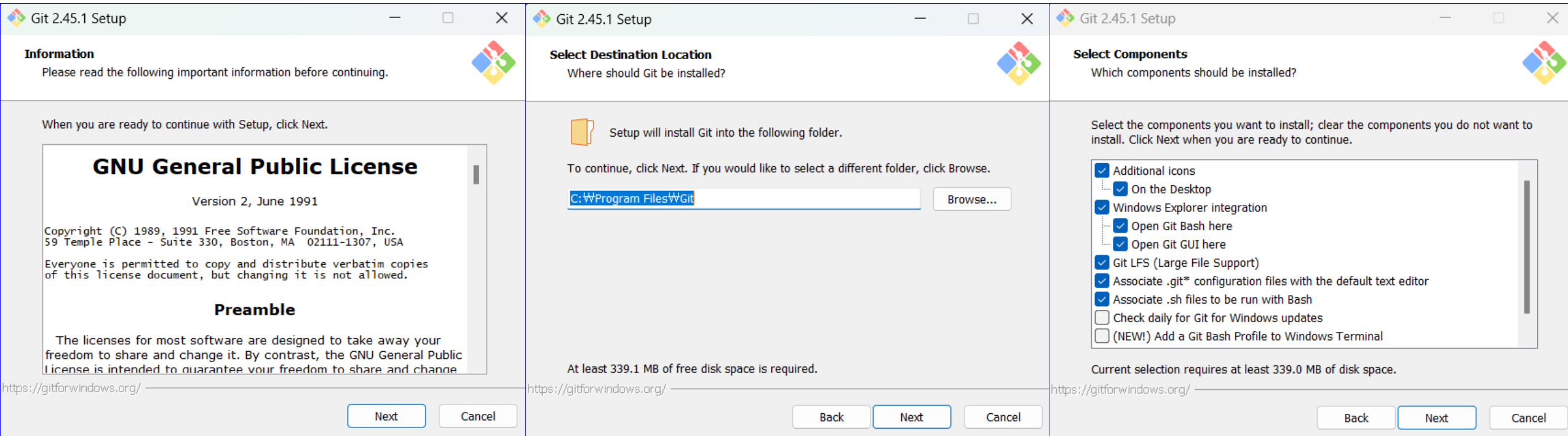
Other Git for Windows downloads

Standalone Installer

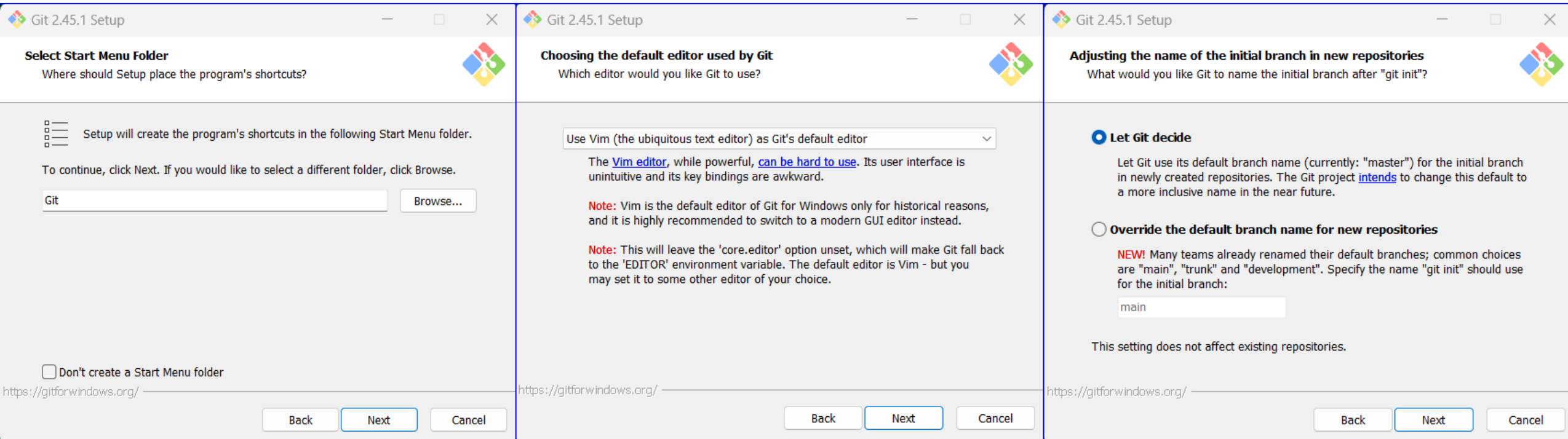
[32-bit Git for Windows Setup.](#)

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

Git Bash 터미널 설치



Git Bash 터미널 설치



Git Bash 터미널 설치

Git 2.45.1 Setup	Git 2.45.1 Setup	Git 2.45.1 Setup
Adjusting your PATH environment How would you like to use Git from the command line?	Choosing the SSH executable Which Secure Shell client program would you like Git to use?	Choosing HTTPS transport backend Which SSL/TLS library would you like Git to use for HTTPS connections?
<p><input type="radio"/> Use Git from Git Bash only This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.</p> <p><input checked="" type="radio"/> Git from the command line and also from 3rd-party software (Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.</p> <p><input type="radio"/> Use Git and optional Unix tools from the Command Prompt Both Git and the optional Unix tools will be added to your PATH. Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.</p>	<p><input checked="" type="radio"/> Use bundled OpenSSH This uses ssh.exe that comes with Git.</p> <p><input type="radio"/> Use external OpenSSH NEW! This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.</p>	<p><input checked="" type="radio"/> Use the OpenSSL library Server certificates will be validated using the ca-bundle.crt file.</p> <p><input type="radio"/> Use the native Windows Secure Channel library Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.</p>
<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>	<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>	<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>

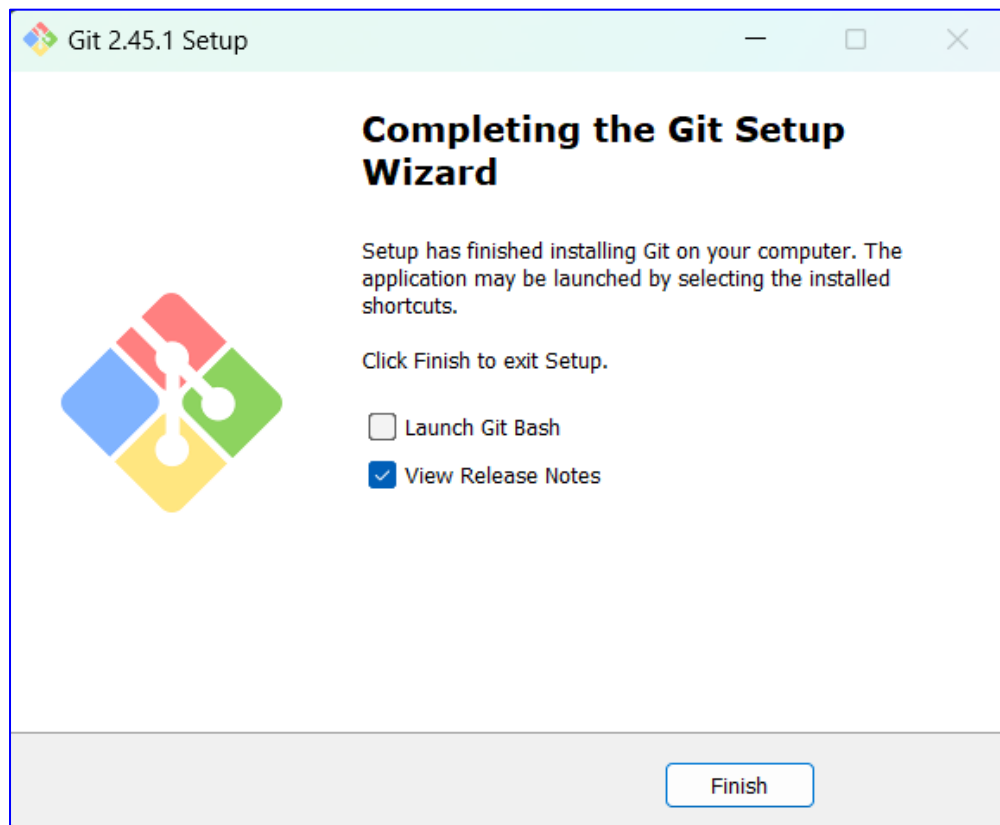
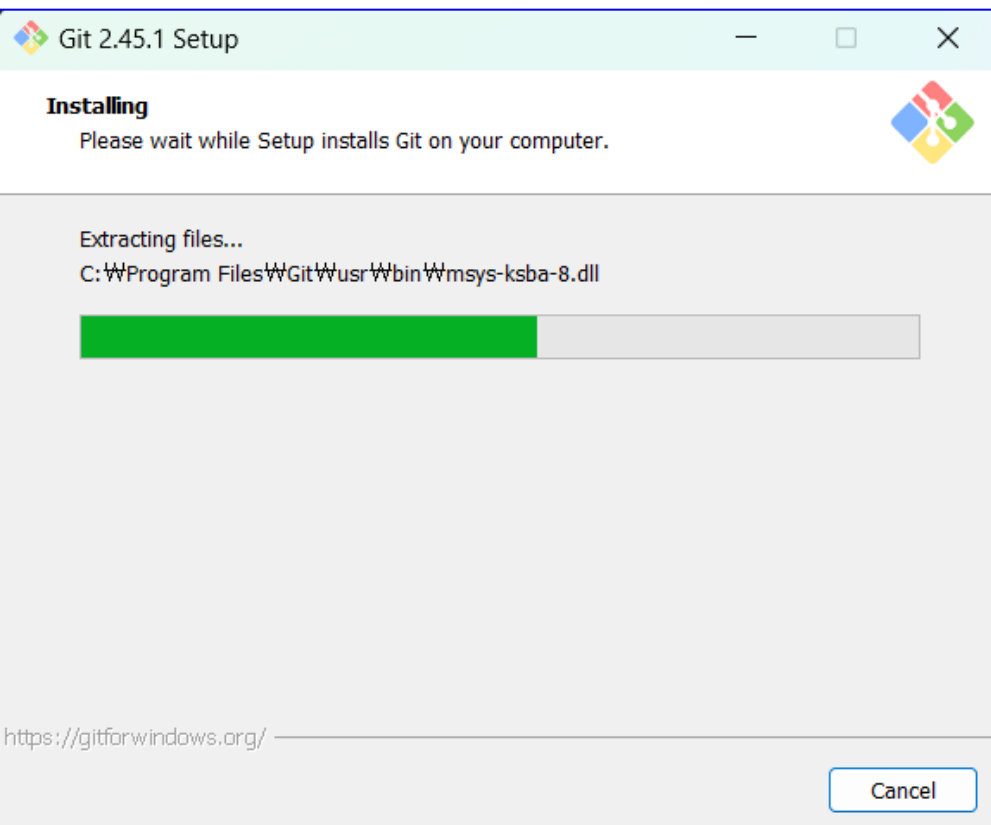
Git Bash 터미널 설치

Git 2.45.1 Setup	Git 2.45.1 Setup	Git 2.45.1 Setup
Configuring the line ending conversions How should Git treat line endings in text files?	Configuring the terminal emulator to use with Git Bash Which terminal emulator do you want to use with your Git Bash?	Choose the default behavior of `git pull` What should `git pull` do by default?
<p><input checked="" type="radio"/> Checkout Windows-style, commit Unix-style line endings</p> <p>Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").</p> <p><input type="radio"/> Checkout as-is, commit Unix-style line endings</p> <p>Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").</p> <p><input type="radio"/> Checkout as-is, commit as-is</p> <p>Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").</p>	<p><input checked="" type="radio"/> Use MinTTY (the default terminal of MSYS2)</p> <p>Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.</p> <p><input type="radio"/> Use Windows' default console window</p> <p>Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.</p>	<p><input checked="" type="radio"/> Fast-forward or merge</p> <p>Fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.</p> <p><input type="radio"/> Rebase</p> <p>Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.</p> <p><input type="radio"/> Only ever fast-forward</p> <p>Fast-forward to the fetched branch. Fail if that is not possible. This is the standard behavior of `git pull`.</p>
<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>	<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>	<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>

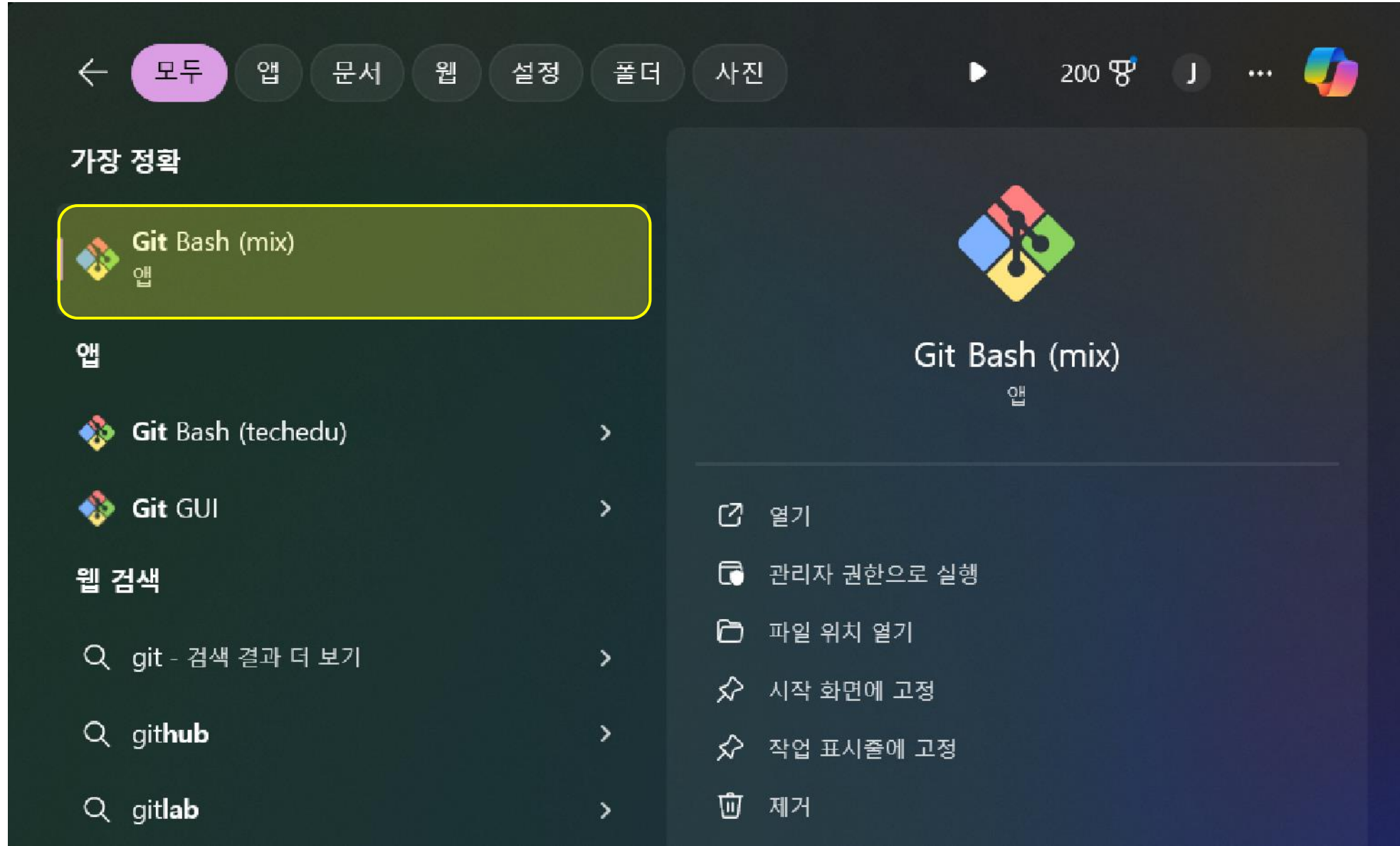
Git Bash 터미널 설치

Git 2.45.1 Setup	Git 2.45.1 Setup	Git 2.45.1 Setup
Choose a credential helper Which credential helper should be configured?	Configuring extra options Which features would you like to enable?	Configuring experimental options These features are developed actively. Would you like to try them?
<p><input checked="" type="radio"/> Git Credential Manager Use the cross-platform Git Credential Manager. See more information about the future of Git Credential Manager here.</p> <p><input type="radio"/> None Do not use a credential helper.</p>	<p><input checked="" type="checkbox"/> Enable file system caching File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.</p> <p><input checked="" type="checkbox"/> Enable symbolic links Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.</p>	<p><input type="checkbox"/> Enable experimental support for pseudo consoles. This allows running native console programs like Node or Python in a Git Bash window without using winpty, but is unfortunately not quite stable yet.</p> <p><input type="checkbox"/> Enable experimental built-in file system monitor (NEW!) Automatically run a built-in file system watcher, to speed up common operations such as <code>`git status`</code>, <code>`git add`</code>, <code>`git commit`</code>, etc in worktrees containing many files.</p>
<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>	<p>https://gitforwindows.org/</p> <p>Back Next Cancel</p>	<p>https://gitforwindows.org/</p> <p>Back Install Cancel</p>

Git Bash 터미널 설치



Git Bash 터미널 설치



GitHub 레포 만들기

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner *



agent47ag925

Repository name *

레포지토리 이름

Great repository names are short and memorable. Need inspiration? How about [ideal-computing-machine](#) ?

Description (optional)



Public

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

퍼블릭(타인이 볼 수 있음)



Private

You choose who can see and commit to this repository.

프라이빗(잠금, 소수만)

Initialize this repository with:



Add a README file

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

README 설명파일

Add .gitignore

.gitignore template: None

.gitignore (깃 추적 무시 파일)

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

라이선스 설정

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

GitHub 레포 연결하기

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest
$ git init
Initialized empty Git repository in C:/Users/jeong/Documents/GitTest/pushtest/.git/

jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ :
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ git remote set-url main https://agent47ag925@github.com/agent47ag925/pushtest.git
```

잘못되어 지울 때 : git remote remove origin

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ git remote -v
origin https://github.com/agent47ag925/pushtest.git (fetch)
origin https://github.com/agent47ag925/pushtest.git (push)
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ echo "NEW REPOSITORY TEST" >> README.text
```

GitHub 레포 연결하기

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ ls
README.text
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ git status
On branch master
```

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

README.text

nothing added to commit but untracked files present (use "git add" to track)

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ git add .
```

warning: in the working copy of 'README.text', LF will be replaced by CRLF the next time Git touches it

GitHub 레포 연결하기

Run

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)  
$ git config --global user.email 47ag925park@gmail.com
```


```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)  
$ git config --global user.name agent47ag925
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)  
$ git config --list  
user.email=47ag925park@gmail.com  
user.name=agent47ag925
```

GitHub 레포 연결하기

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ git commit -m "+First commit"
[master (root-commit) 62447ac] +First commit
1 file changed, 1 insertion(+)
create mode 100644 README.text
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ git push origin master
remote: Repository not found.
fatal: repository 'https://github.com/agent47ag925/pushtest.git/' not found
```



SSH KEY 생성 인증

GitHub 레포 연결하기

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ ssh-keygen -t ed25519 -C "47ag925park@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/jeong/.ssh/id_ed25519):
Created directory '/c/Users/jeong/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/jeong/.ssh/id_ed25519
Your public key has been saved in /c/Users/jeong/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:xziU7FR88QUid704vTAmVHhONHtS5U+IQig6pWNckPg 47ag925park@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|oo      o+oB.=oo|
|o .    o..+o++X *|
| o  +  .= o== = o|
|  E *  + + ..= ..|
|   o o  S + = . .|
|      o o o .    |
|                  |
+-----[SHA256]-----+
```

ssh-keygen -t ed25519 -C "이메일"

ssh-keygen -t rsa -b 4096 -C "이메일"

GitHub 레포 연결하기 - 내 깃헙 settings에 연결하기

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (master)
$ clip < /c/Users/jeong/.ssh/id_ed25519.pub
```

macOS \$ pbcopy < ~/.ssh/id_ed25519.pub

agent47ag925 (agent47ag925)
Your personal account [Switch settings context](#)

Public profile
Account
Appearance
Accessibility
Notifications

Access

Billing and plans
Emails
Password and authentication
Sessions

SSH and GPG keys

SSH keys

There are no SSH keys associated with your account.

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

Vigilant mode

agent47ag925

Set status

Your profile
Add account

Your repositories
Your projects
Your Copilot
Your organizations
Your enterprises
Your stars
Your sponsors
Your gists

Upgrade
Try Enterprise
Feature preview
Settings

GitHub 레포 연결하기 - 내 깃헙 settings에 연결하기

Add new SSH Key

Title

MyWindowDesktop

Key type

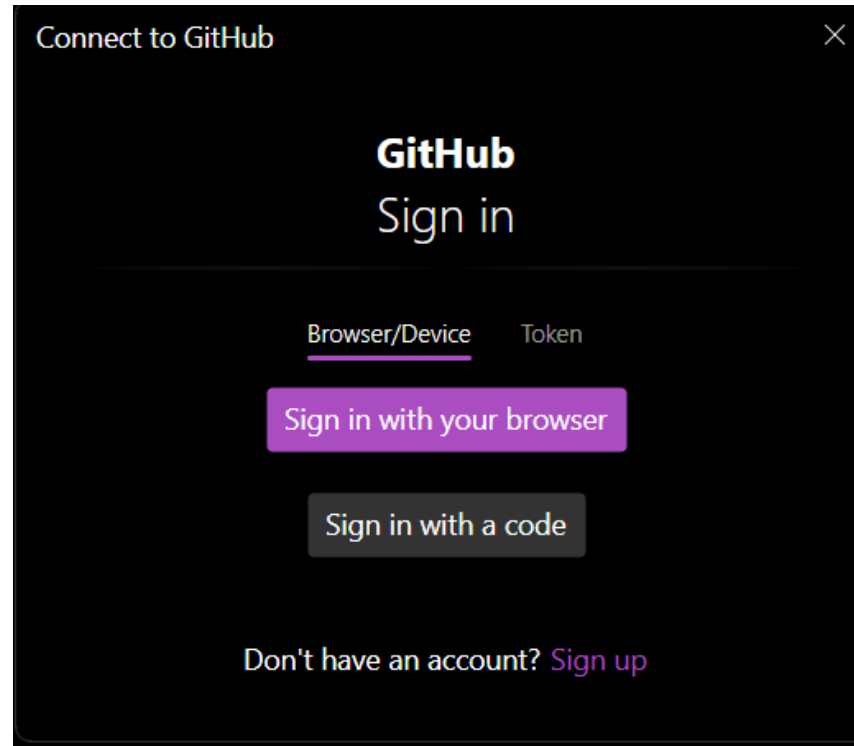
Authentication Key

Key

ssh-ed25519

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (main)
$ git push origin master
```

GitHub 레포 연결하기



```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (main)
$ git push origin master
Everything up-to-date
```

Gitbash 1. 변경사항 새로 만들고 커밋, 푸시하기

- `git add .`
- `git status`
- `git commit -m “커밋 메시지”`
- `git push origin master` (`git push origin main`)

Gitbash 2. 레포지토리에 콜라보레이터 추가하기

- 레포지토리 이름 하단의 “settings” 메뉴 선택
- 새로 열린 메뉴 창 왼쪽에 Collaborators 선택
- Manager acces에 Add people 하여, 메일주소로 콜라보레이터 추가

Gitbash 3. 브랜치를 만들고 체크아웃하기

- `git checkout -B [브랜치 이름]`
- `git checkout [원래 브랜치 이름]`

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (main)
$ git checkout -B local
Switched to a new branch 'local'
```

```
jeong@swd47ag925 MINGW64 ~/Documents/GitTest/pushtest (local)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

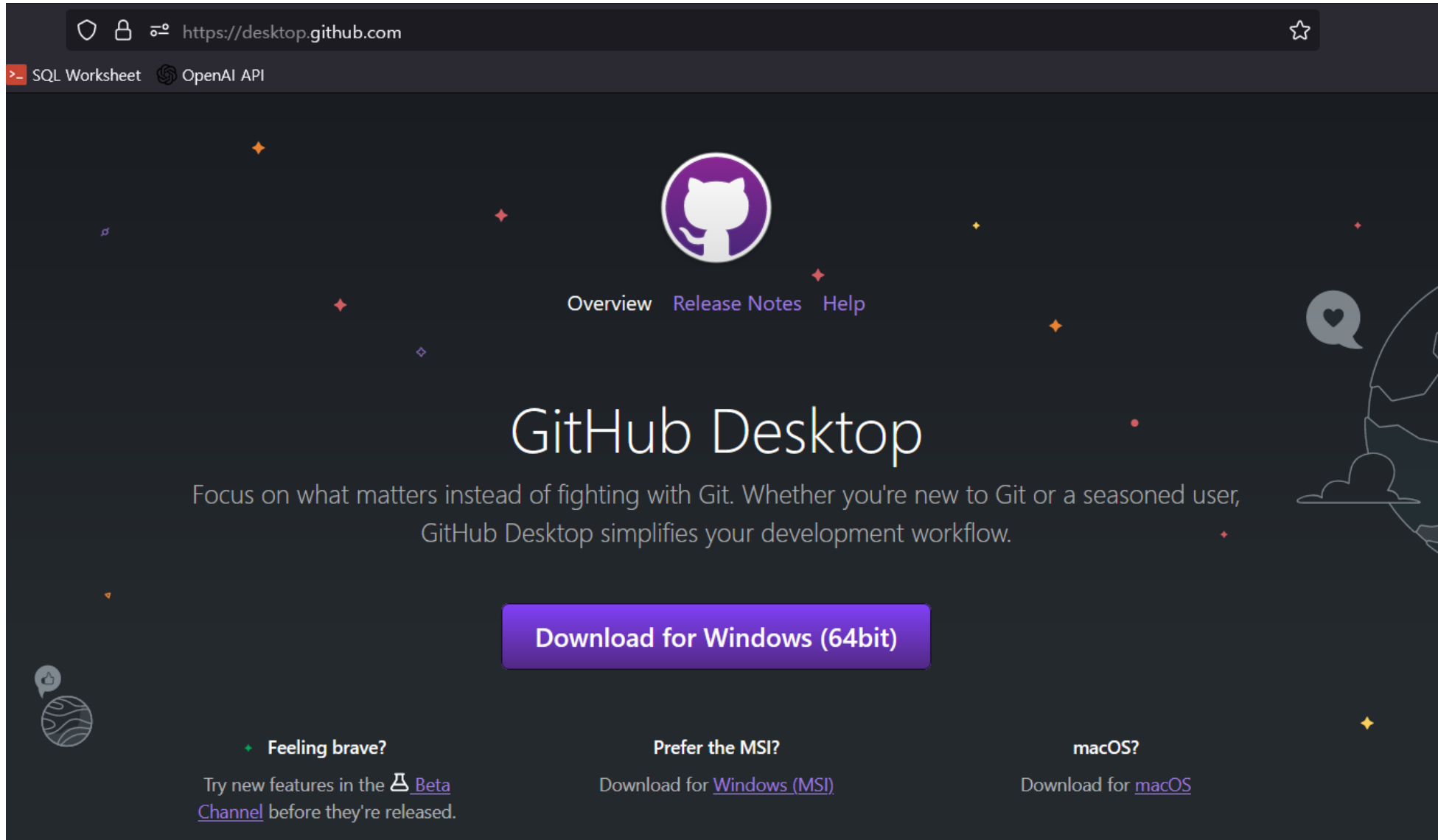
Gitbash 4. 새 디렉토리 만들고, 맘에 드는 공개 레포 클론하기

- `cd [폴더 이름]` : 폴더 이름 위치로 이동
- `mkdir [새 폴더 이름]` : [새 폴더 이름]을 가진 폴더 만들기
- `git clone [깃레포 주소]` : 공개된 깃 레포를 주소로 다운로드

Gitbash 5. 레포지토리 지우기

- 레포지토리 이름 하단의 “settings” 메뉴 선택
- 새로 열린 메뉴 창 왼쪽에 General 선택
- 제일 아래쪽 Danger Zone 확인,
- Delete this repository

Git GUI – Github Desktop



Git GUI – Github Desktop

Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

[Sign in to GitHub.com](#)

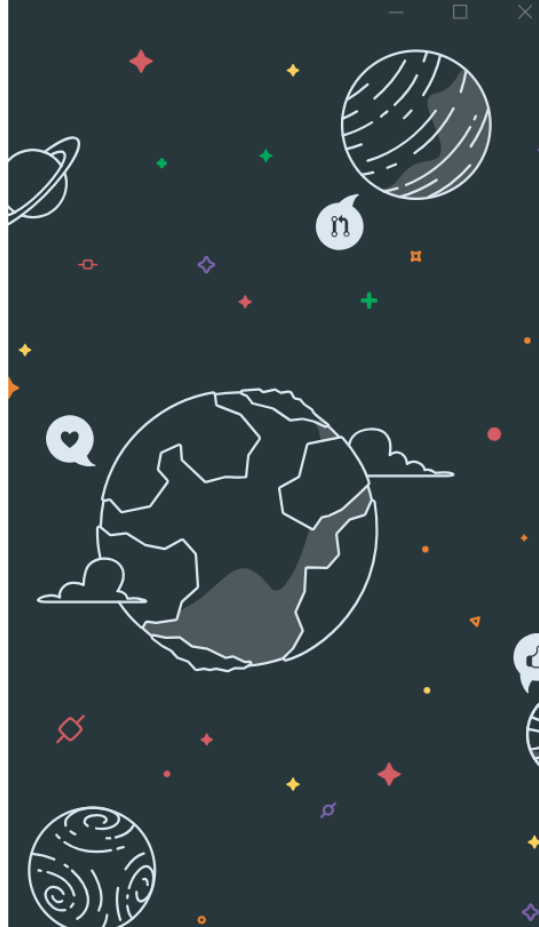
[Sign in to GitHub Enterprise](#)

New to GitHub? [Create your free account.](#)

Skip this step

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#).

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. [Learn more about user metrics.](#)



Authorize GitHub Desktop



GitHub Desktop by [GitHub Desktop](#)

wants to access your [agent47ag925](#) account



Repositories

Public and **private**



Personal user data

Full access



Workflow

Update GitHub Action Workflow files.

Organization access



Follow-fastai ✓

Cancel

Authorize desktop

Authorizing will redirect to

Git GUI - Github Desktop

Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

- ☒ Use my GitHub account name and email address
☐ Configure manually

Name

agent47ag925

Email

102883790+agent47ag925@users.noreply.github.com

Finish

Cancel

Example commit

Fix all the things

 agent47ag925 • 30 minutes ago



Create a tutorial repository...



Clone a repository from the Internet...



Create a New Repository on your local drive...

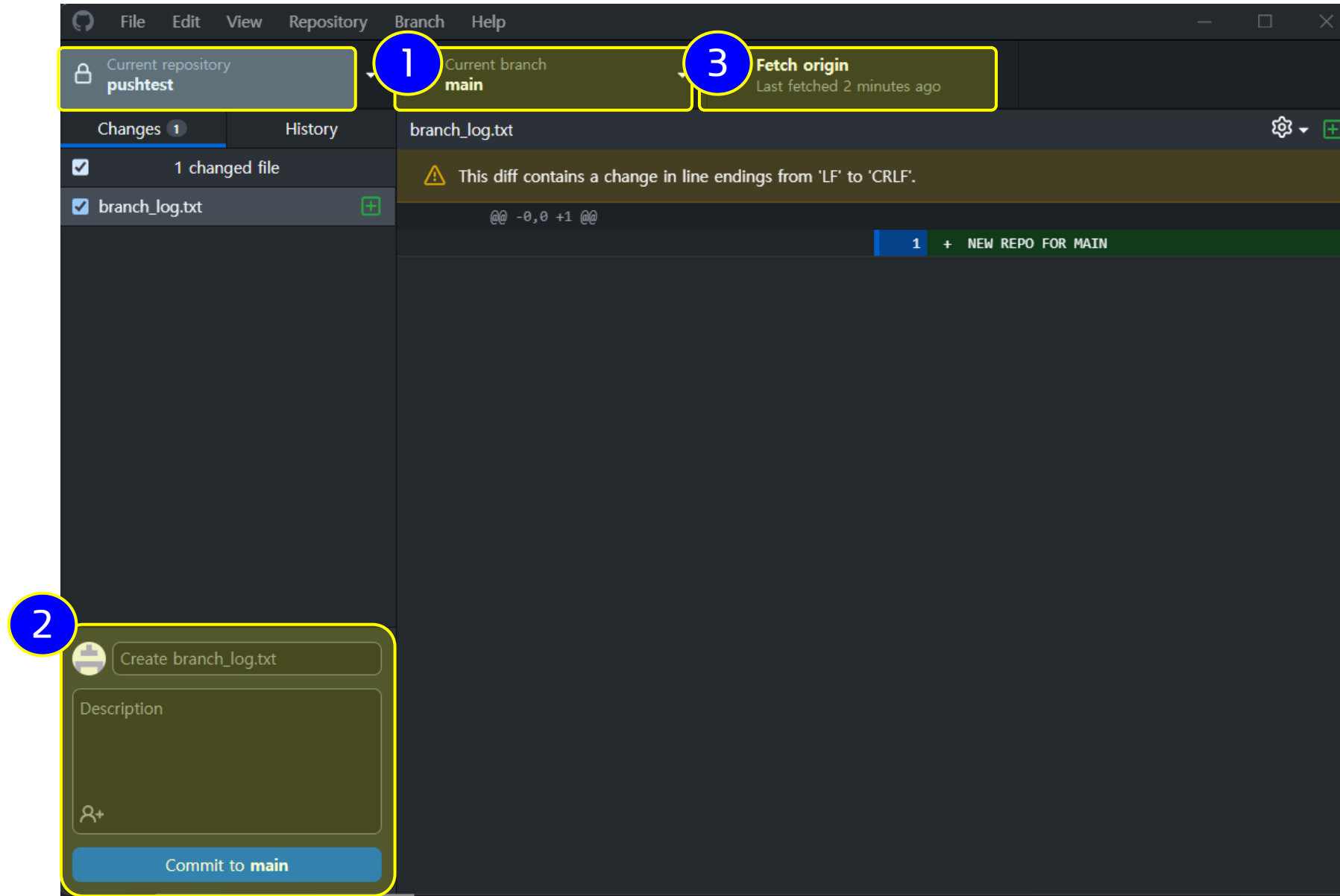


Add an Existing Repository from your local drive...





ProTip! You can drag & drop an existing repository folder here to add it to Desktop


Git GUI - Github Desktop




Git GUI - Github Desktop

FileEditViewRepositoryBranchHelp

Current repository
pushtest

Current branch
master


**Fetch origin**
Last fetched just now

Changes

History

Select branch to compare...

+First commit

 agent47ag925 • 1 hour ago

1 changed file

README.text

agent47ag925

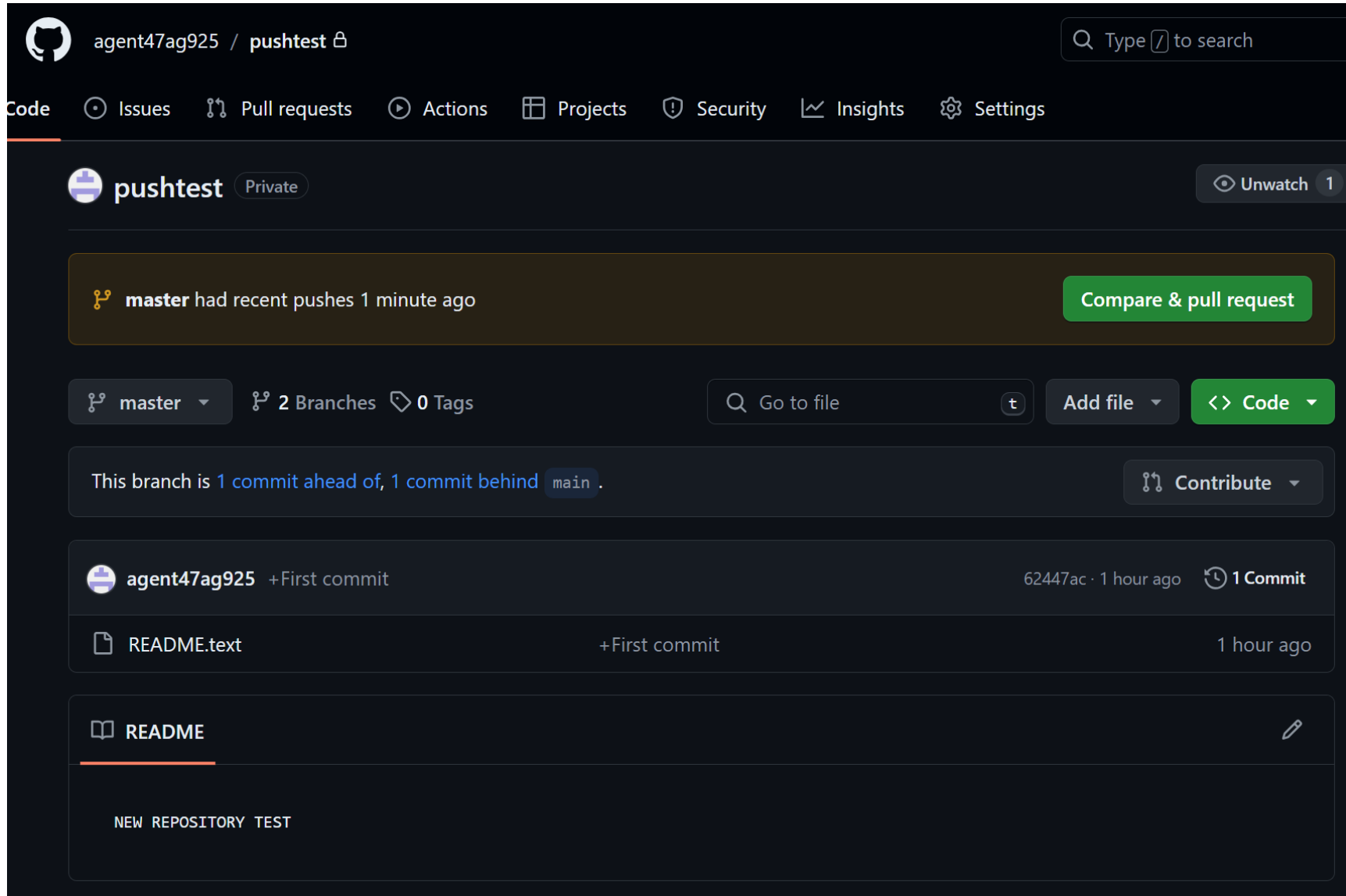
62447ac

+1-0

README.text

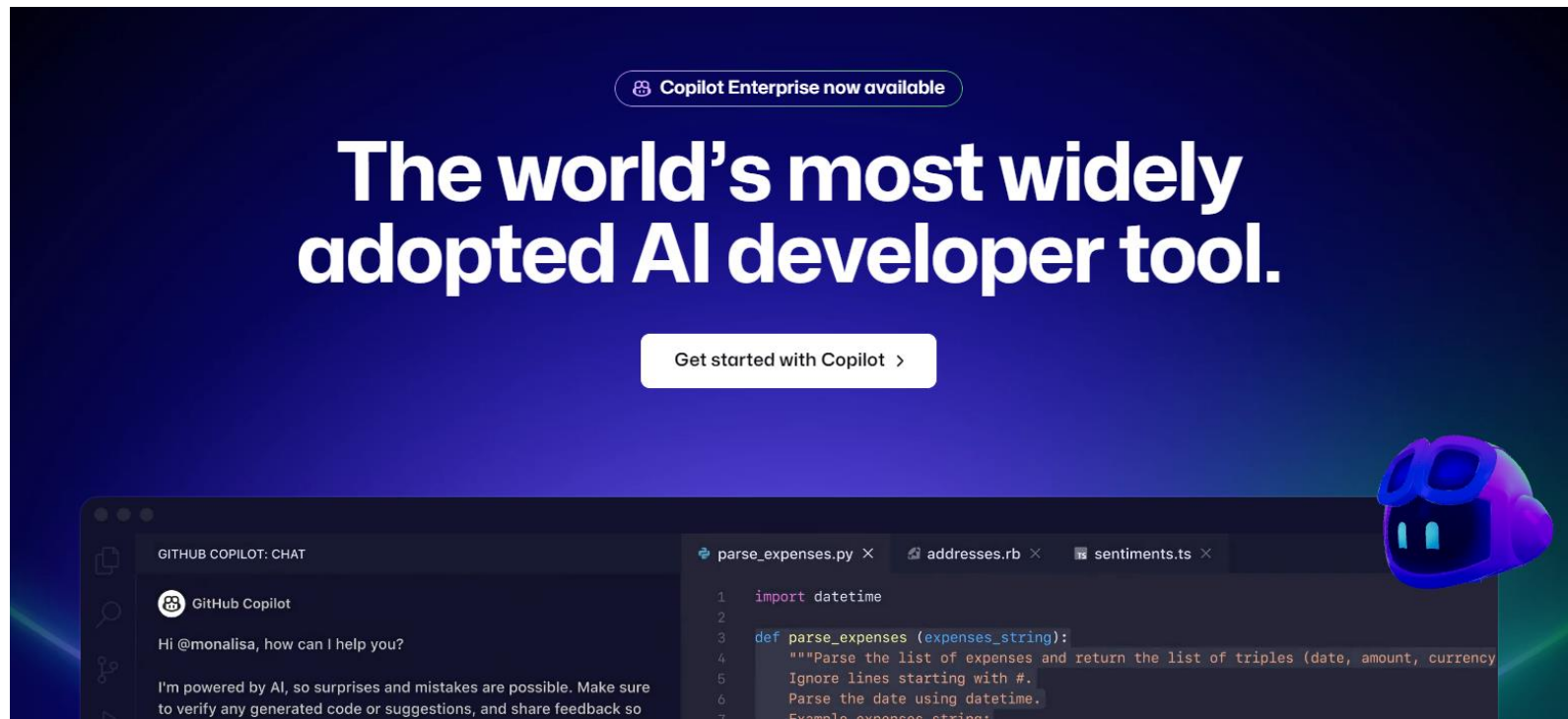
@@ -0,0 +1 @@
+ NEW REPOSITORY TEST

Git GUI – Github Desktop



Github Copilot

- Github와 openai가 공동 개발한 ai 기반 코드 작성 도구
- 개발자가 코드를 작성할 때 실시간으로 코드 제안을 제공
- 코드 작성 속도를 높이고, 효율성을 극대화하는 데 도움을 줌
- Visual Studio Code, JetBrains, Neovim 등 코드 편집기에서 사용할 수 있음



Github Copilot

- 1달여 간의 Trial 기간 제공
- 신용카드 및 지불 가능한 수단을 등록하여 Trial 시작

Copilot Individual

For individual developers, freelancers, students, and educators that want to code faster and happier.

\$ 10 USD
per month / \$100 USD per year

Start a free trial >

Free for verified students, teachers, and maintainers of popular open source projects.

What's included

Chat

- ✓ Unlimited messages and interactions
- ✓ Context-aware coding support and explanations

Most popular

Copilot Business

For organizations ready to improve engineering velocity, code quality, and developer experience.

\$ 19 USD
per user / month

Buy now >

Contact sales >

What's included

Chat

- ✓ Unlimited messages and interactions
- ✓ Context-aware coding support and explanations

New

Copilot Enterprise

For companies looking to customize GitHub Copilot to their organization and infuse AI across the developer workflow.

\$ 39 USD
per user / month


Contact sales >

What's included

Chat

- ✓ Unlimited messages and interactions
- ✓ Context-aware coding support and explanations

Billing Information

 **agent47ag925**
Personal account

First name *

jeongeun

Last name *

park

Address (P.O. box, company name, c/o) *

Address line 2 (Apartment, suite, unit)

City *

seoul

Country/Region *

Korea, South

State/Province

seoul

Postal/Zip code

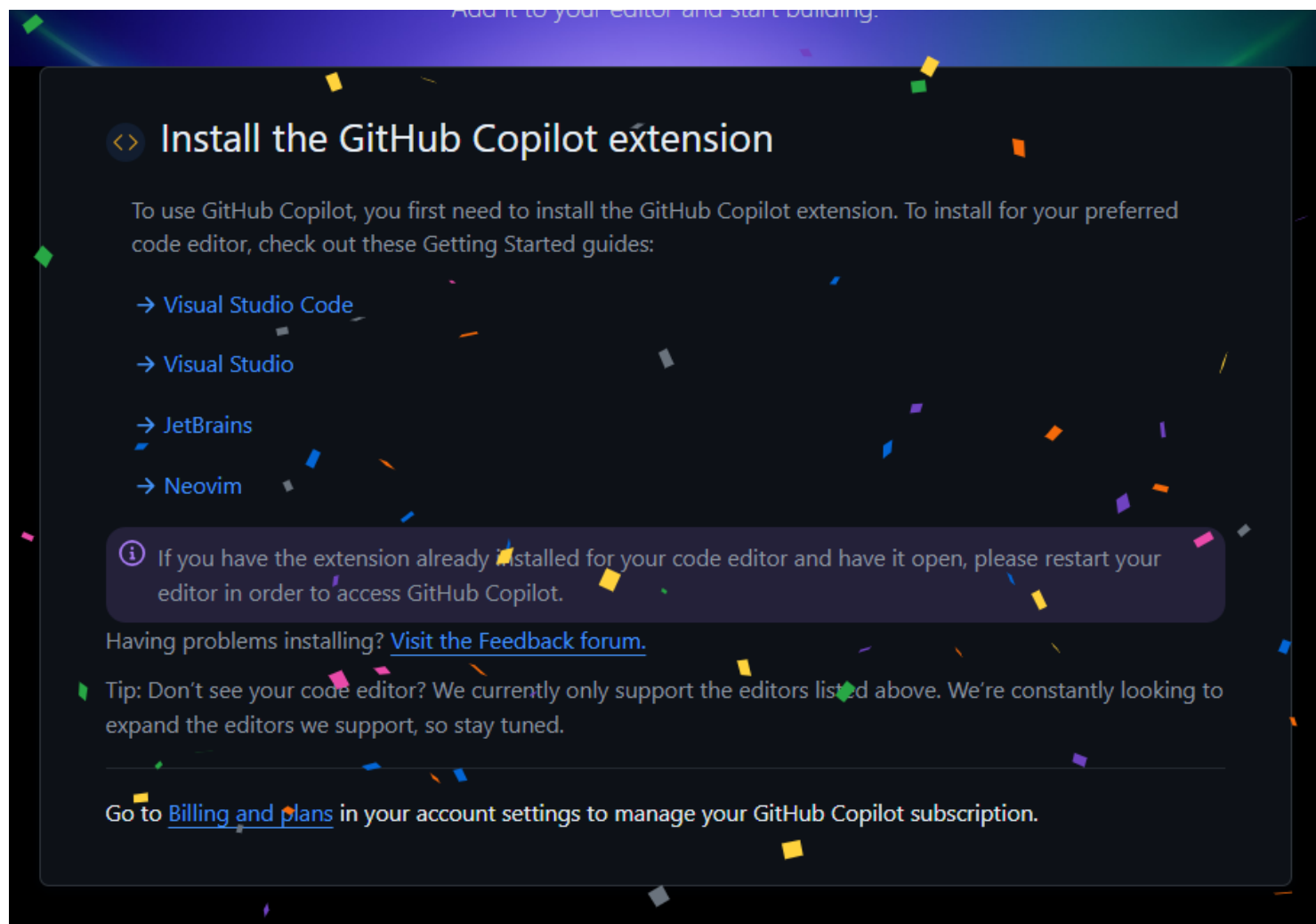
Required for certain countries

Required for certain countries

Save

Github Copilot


- Vscodero 연결 가능





EXTENSIONS: MARKETPLACE

하기_2 (1).ipynbML_cheonan_03_빅데이터분석과정.ipynbML_cheonan_04_지도학습_분류.ipynb

copilot


**GitHub Copilot**
Your AI pair programmer
GitHub **Install**

**GitHub Copilot Chat**
AI chat features powered by Copilot
GitHub **Install**




GitHub Copilot




v1.195.0

GitHub  [github.com](#) | 16,638,238 | ★★★★★ (1034)


Your AI pair programmer


Install 

DETAILSFEATURESCHANGELOGEXTENSION PACK



Authorize GitHub for VS Code

**GitHub for VS Code** by [Visual Studio Code](#)
wants to access your agent47ag925 account

**Personal user data**
Email addresses (read-only)

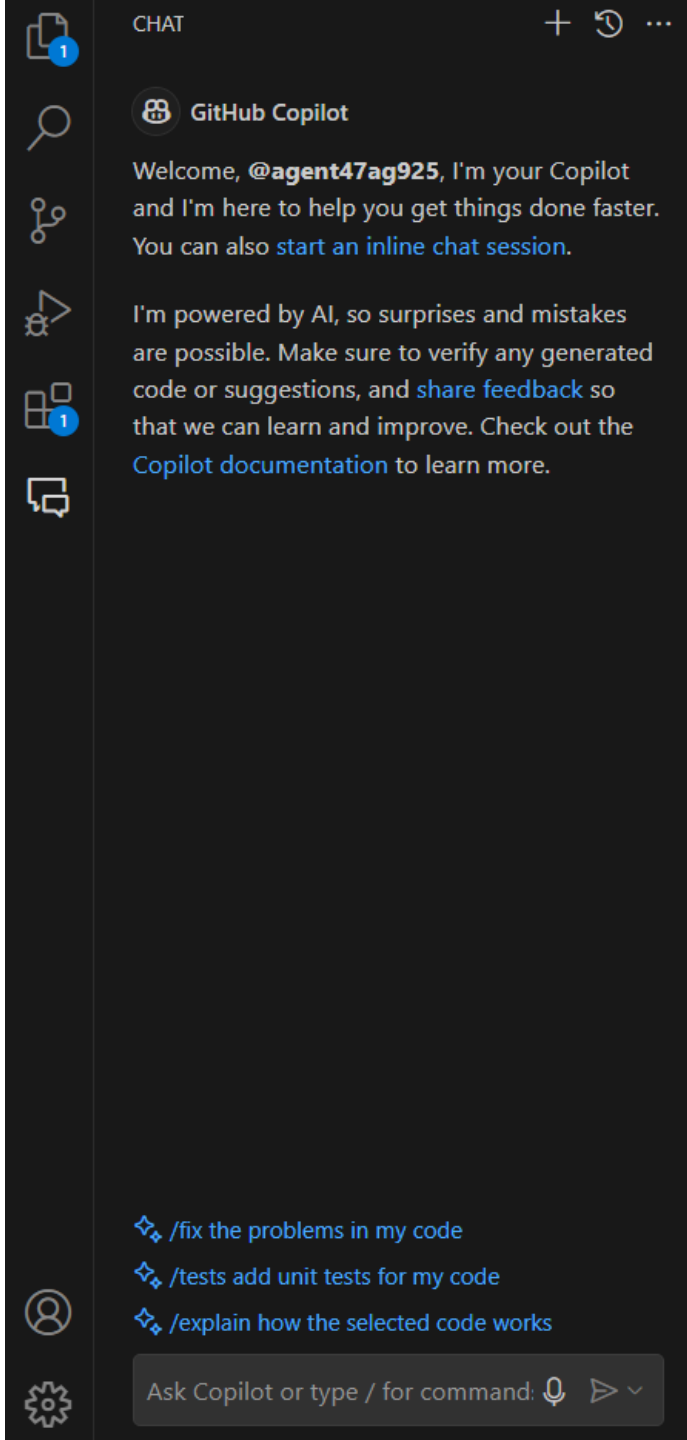
Cancel

Authorize Visual-Studio-Code

Authorizing will redirect to <https://vscode.dev>

Github Copilot

- Workspace : 코드 작성에 도움을 주는 코파일럿
- Terminal : 터미널의 명령어와 관련하여 도움을 주는 코파일럿
- Vscode : vscode 자체의 기능에 대하여 도움을 주는 코파일럿



Command	Usage	Chat window	Inline chat
/doc	<p>Add comments for specified or selected code.</p> <p>Examples:</p> <ul style="list-style-type: none"> - /doc DeleteBasketAsync method in BasketService.cs - select desired code and enter /doc 	Yes	Yes
/explain	<p>Get code explanations.</p> <p>Examples:</p> <ul style="list-style-type: none"> - /explain the AddItemToBasket method in BasketService.cs - select desired code and enter /explain 	Yes	Yes
/fix	<p>Propose a fix for problems in the selected code.</p> <p>Examples:</p> <ul style="list-style-type: none"> - /fix the SetQuantities method in BasketService.cs - select desired code and enter /fix 	Yes	Yes
/generate	<p>Generate code to answer specified question.</p> <p>Example: /generate code to add two numbers in Calculator.cs</p>	Yes	Yes
/help	<p>Get help on using Copilot Chat.</p> <p>Example: /help</p>	Yes	Yes
/optimize	<p>Analyze and improve running time of the selected code.</p> <p>Examples:</p> <ul style="list-style-type: none"> - /optimize the AddItemToBasket method in BasketService.cs - select desired code and enter /optimize 	Yes	Yes
/tests	<p>Create unit tests for the selected code.</p> <p>Example: select desired code and enter /tests</p>	Yes	Yes

끝! 감사합니다. 😊

이 저작물의 권리는 저작권자에게 있습니다.

해당 자료는 2024년 학습 목적으로 사용되었습니다.

저자의 허락이 없는 무단 복제, 전송(파일 업로드), 배포(출력)는 저작권법 위반 이오니 학습자료 관리에 유의해주시오.

내용 및 저작물 사용 문의 : jeongeunswd@gmail.com