

# 누구나 간단하게 git 입문

# Git이란?

Git은 분산 버전 관리 시스템으로, 소스 코드의 변경 사항을 효율적으로 관리할 수 있게 도와줍니다

## 버전 관리

변경 이력을 기록하고  
언제든 이전 상태로 복원 가능



## 분산 구조

모든 개발자가 자신의  
로컬 저장소를 가지고 작업 가능



## 브랜치 관리

여러 버전의 개발을  
동시에 진행하고 쉽게 병합 가능



## 협업 효율성

여러 명이 동시에  
작업해도 충돌 최소화

# Git이란?

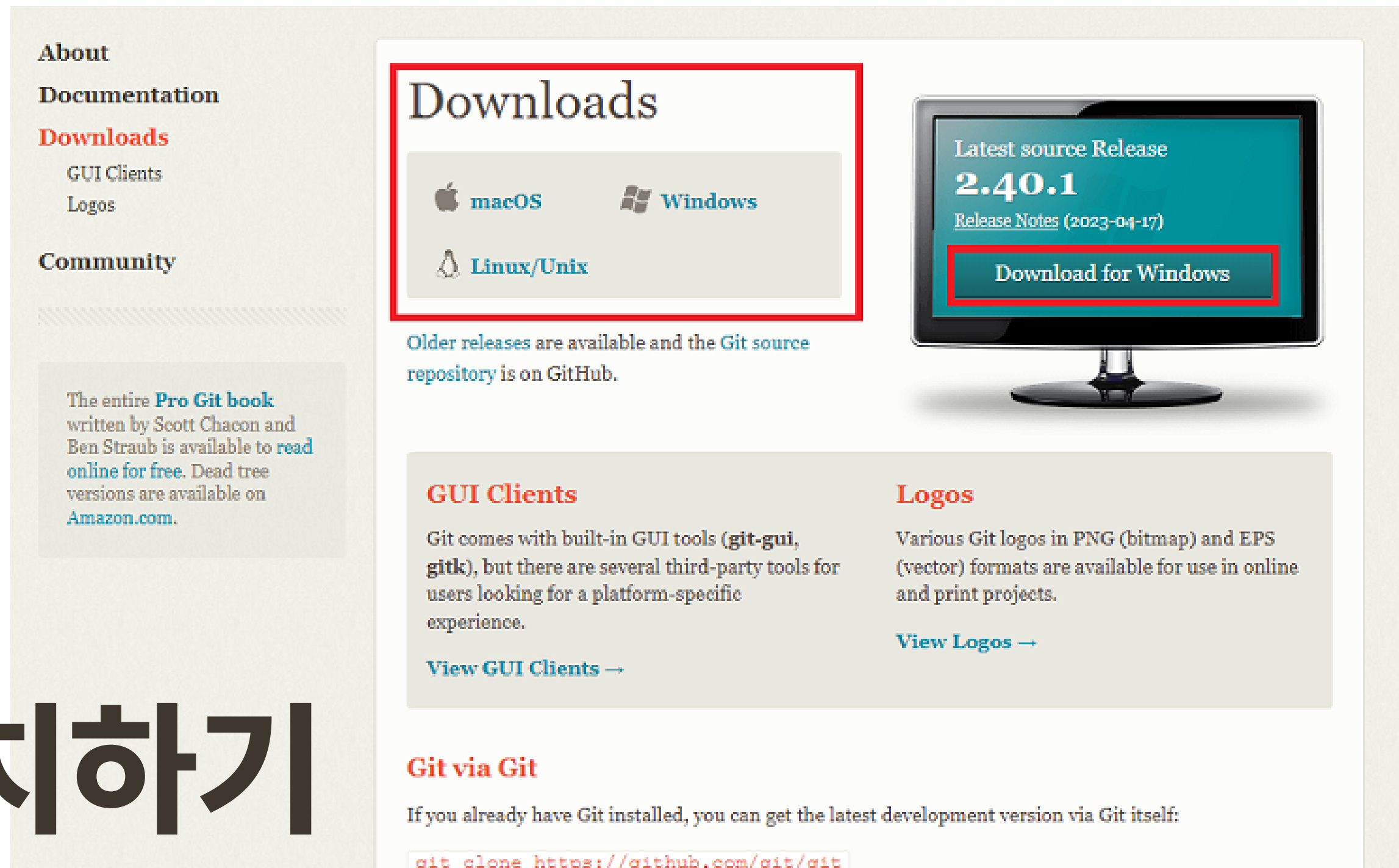
---

## 왜 필요한가?

실수로 코드를 잘못 바꾸었을 때 되돌리기 쉽고,  
여러 사람이 동시에 프로젝트를 개발할 때 충돌을 방지하며,  
코드 변경 이력을 남겨 누가 무엇을 언제 바꿨는지 추적 가능하게 해줍니다.

Git은 프로젝트의 과거와 현재를 모두 기억하며,  
팀원들과 안전하게 협업할 수 있도록 도와주는 도구입니다.

# 1. git 설치하기



The screenshot shows the Git website's navigation menu on the left with links for About, Documentation, Downloads, GUI Clients, Logos, and Community. The main content area features a 'Downloads' section with a red border, containing links for macOS, Windows, and Linux/Unix. To the right, a monitor graphic displays the 'Latest source Release 2.40.1' and a 'Download for Windows' button, both highlighted with red boxes. Below the Downloads section, there are sections for 'GUI Clients' and 'Logos'. At the bottom, the 'Git via Git' section includes a terminal command to clone the repository.

About  
Documentation  
**Downloads**  
GUI Clients  
Logos  
Community

The entire **Pro Git** book written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads

macOS Windows  
Linux/Unix

Older releases are available and the Git source repository is on GitHub.

### GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

### Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

About

Documentation

**Downloads**

GUI Clients

Logos

Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read [online for free](#). Dead tree versions are available on [Amazon.com](#).

## 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 **about 13 hours 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

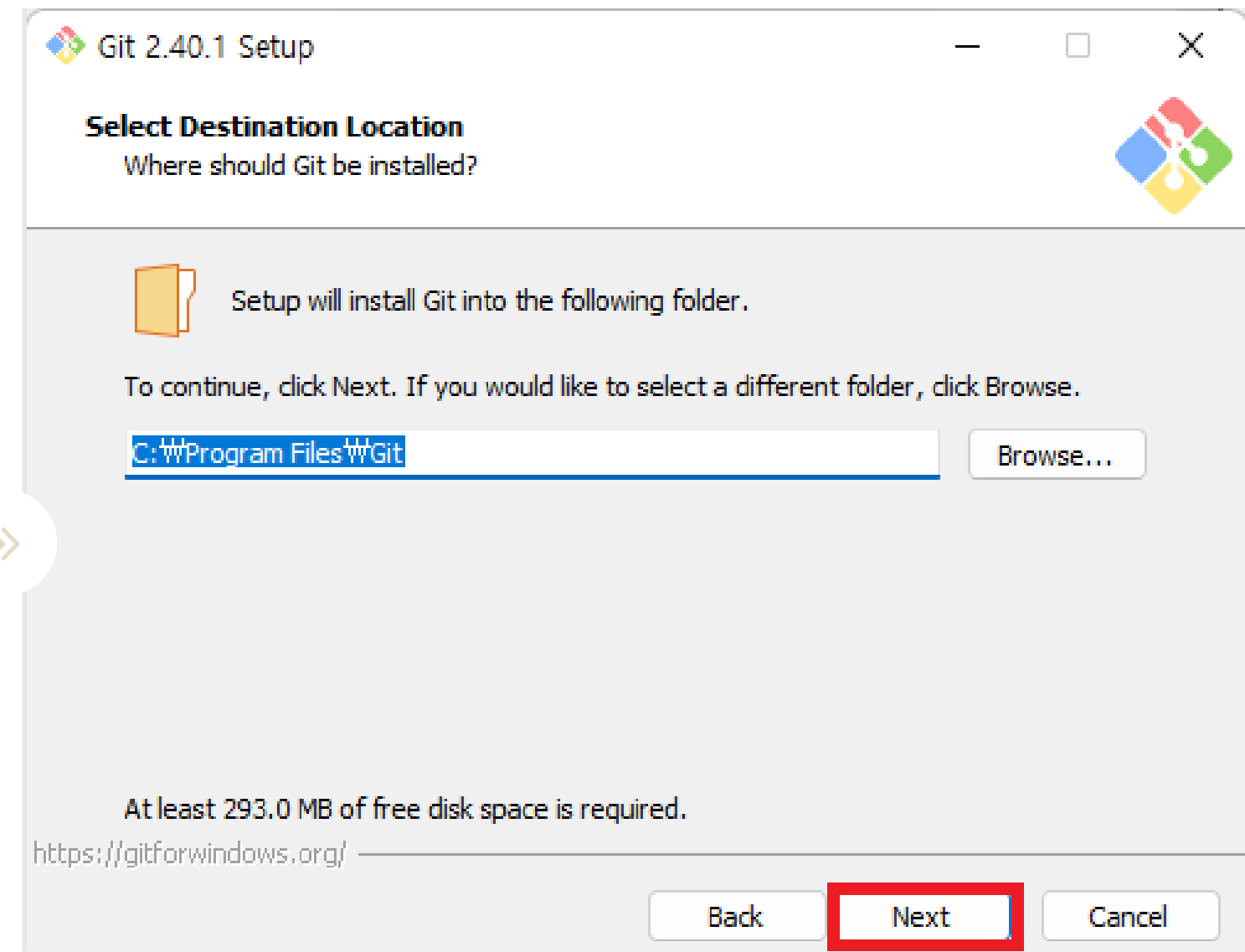
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

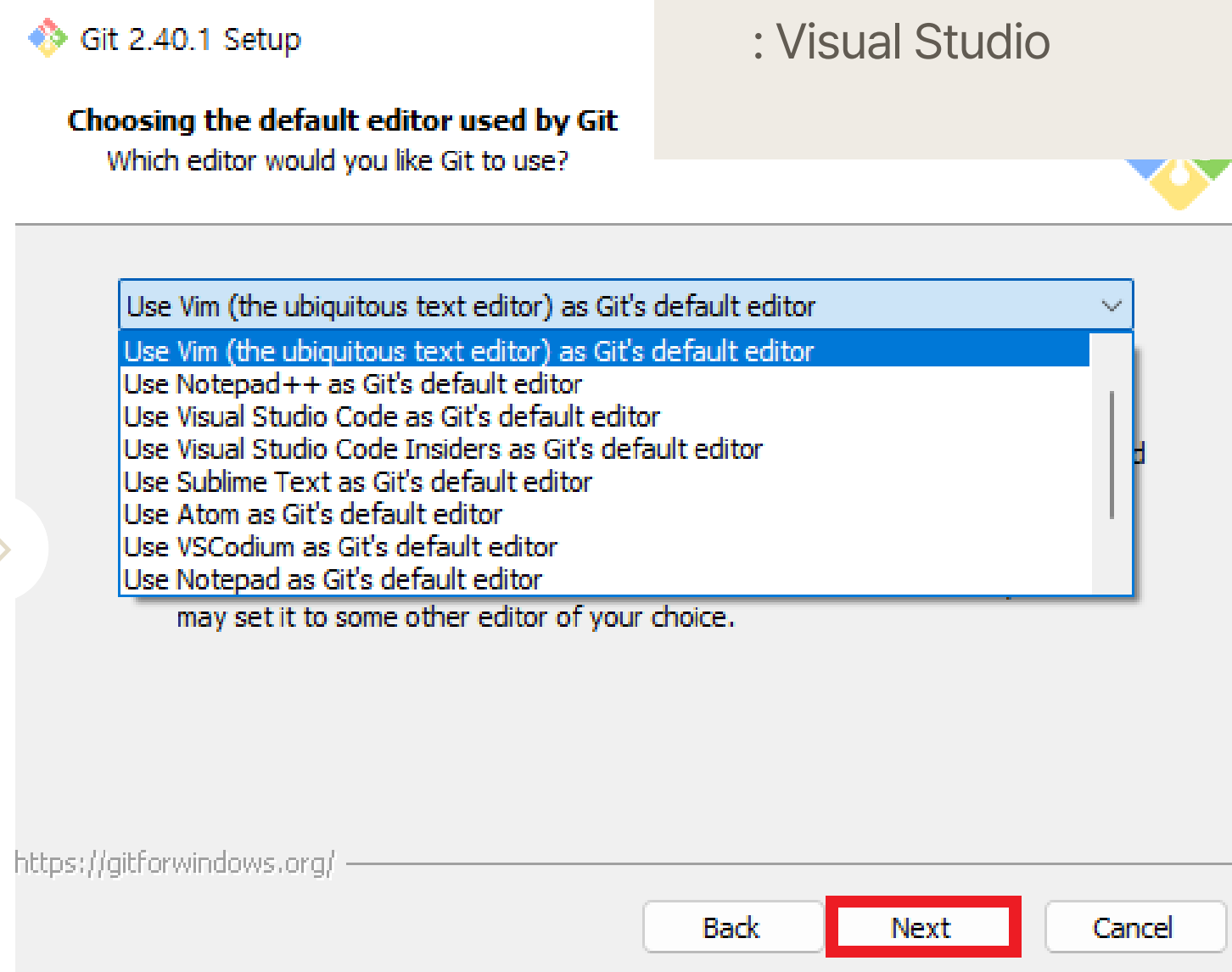
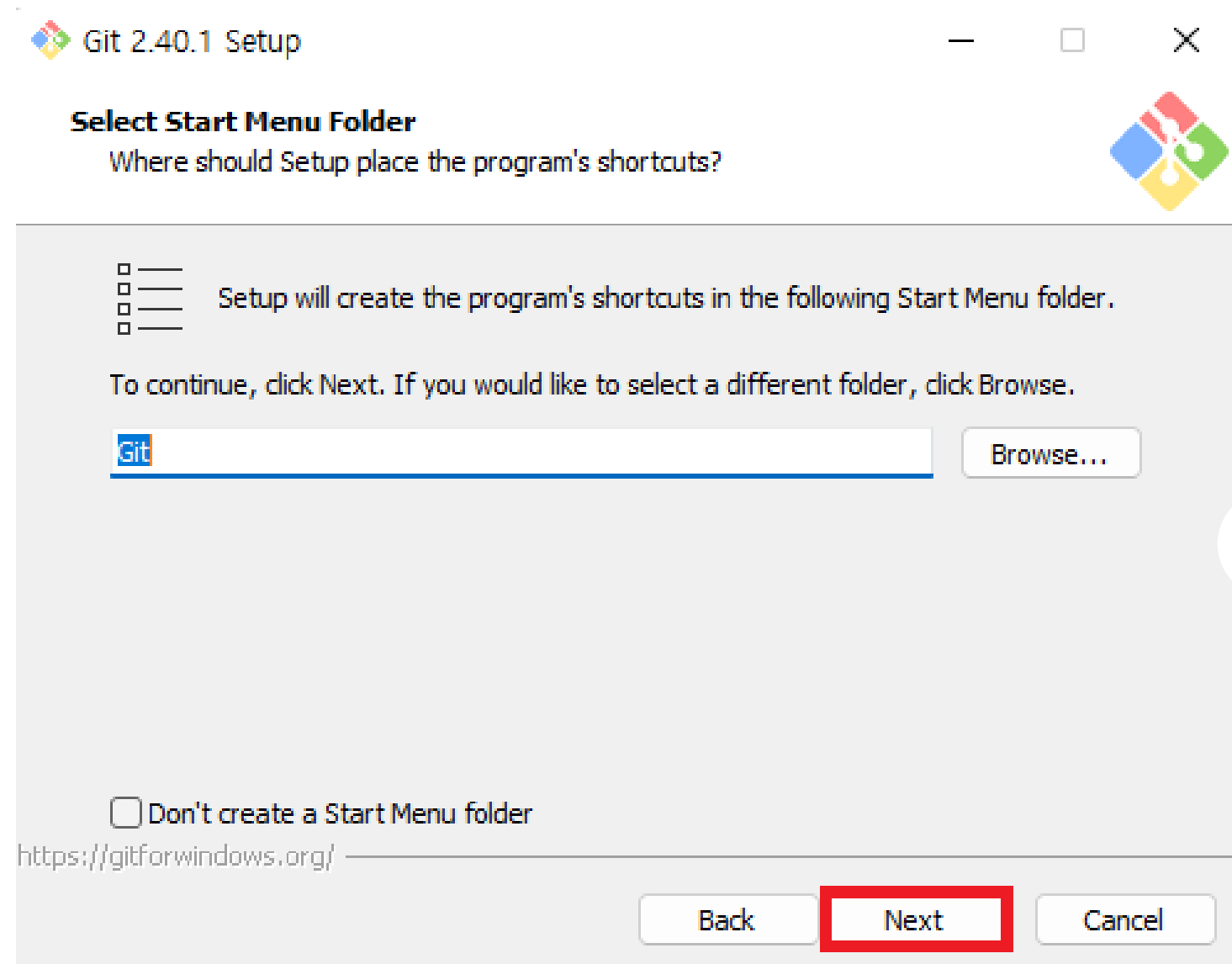
The current source code release is version **2.40.1**. If you want the newer version, you can build it from [the source code](#).

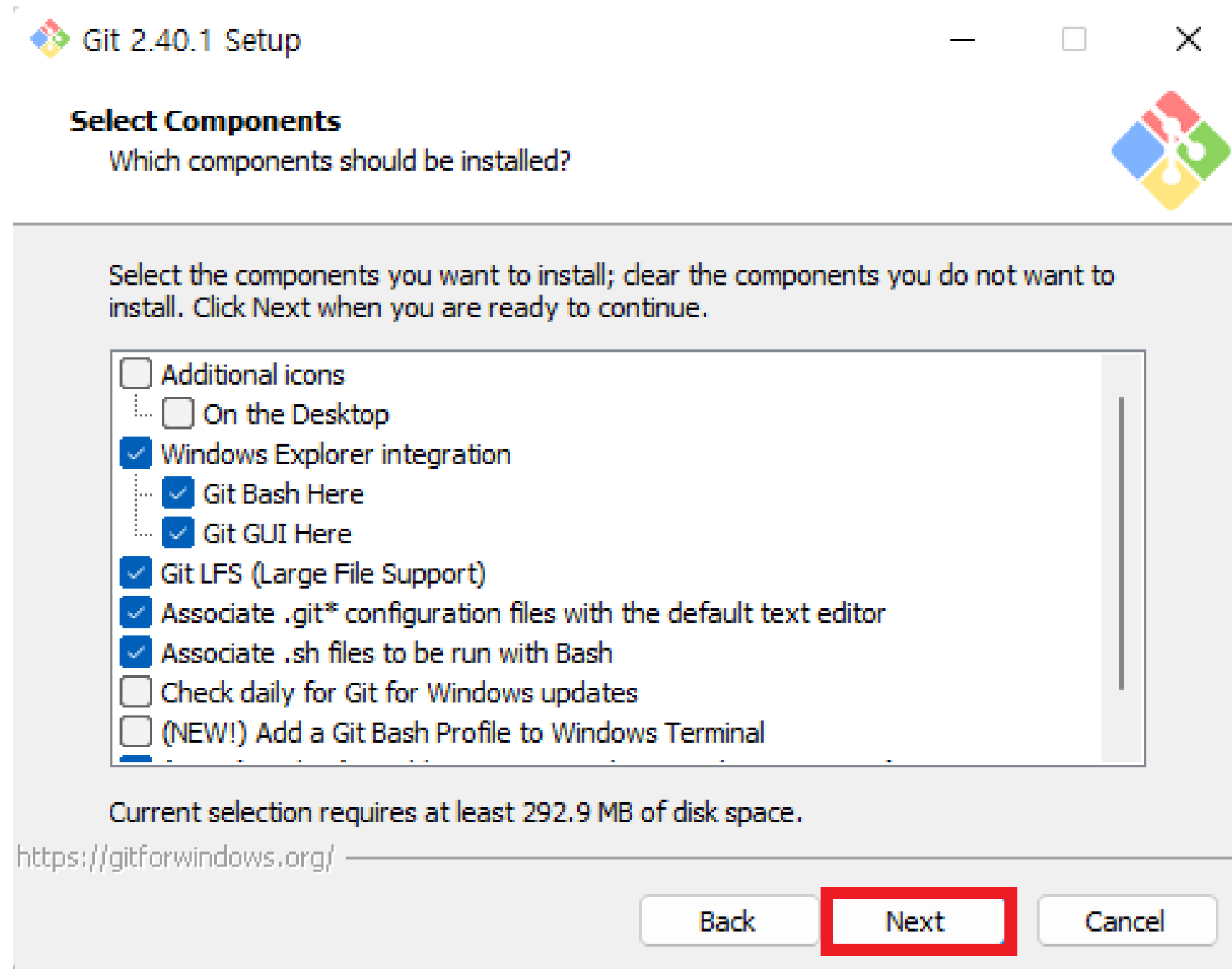
Standalone Installer과 Portable 두 가지가 있습니다.

Standalone은 실행파일을 다운 받고, Portable은 압축파일이 설치됩니다.



Git을 사용할 기본 에디터를 선택  
: Visual Studio





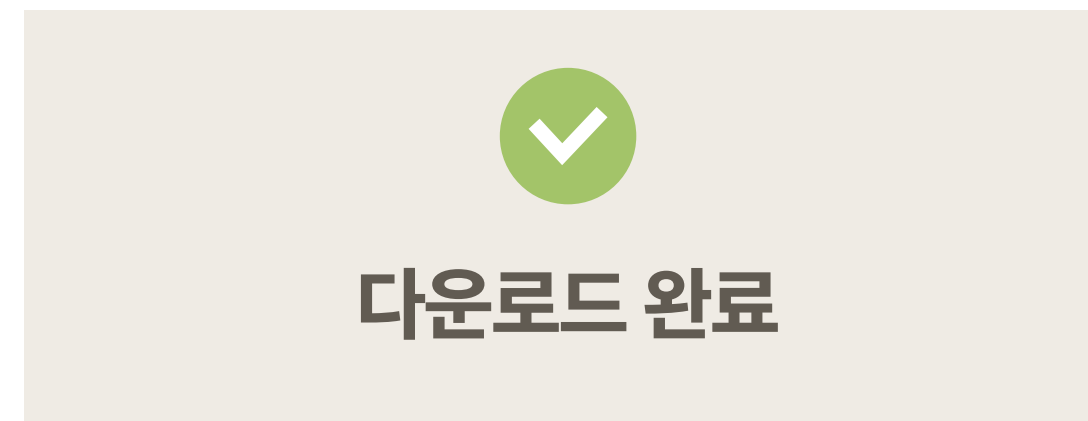
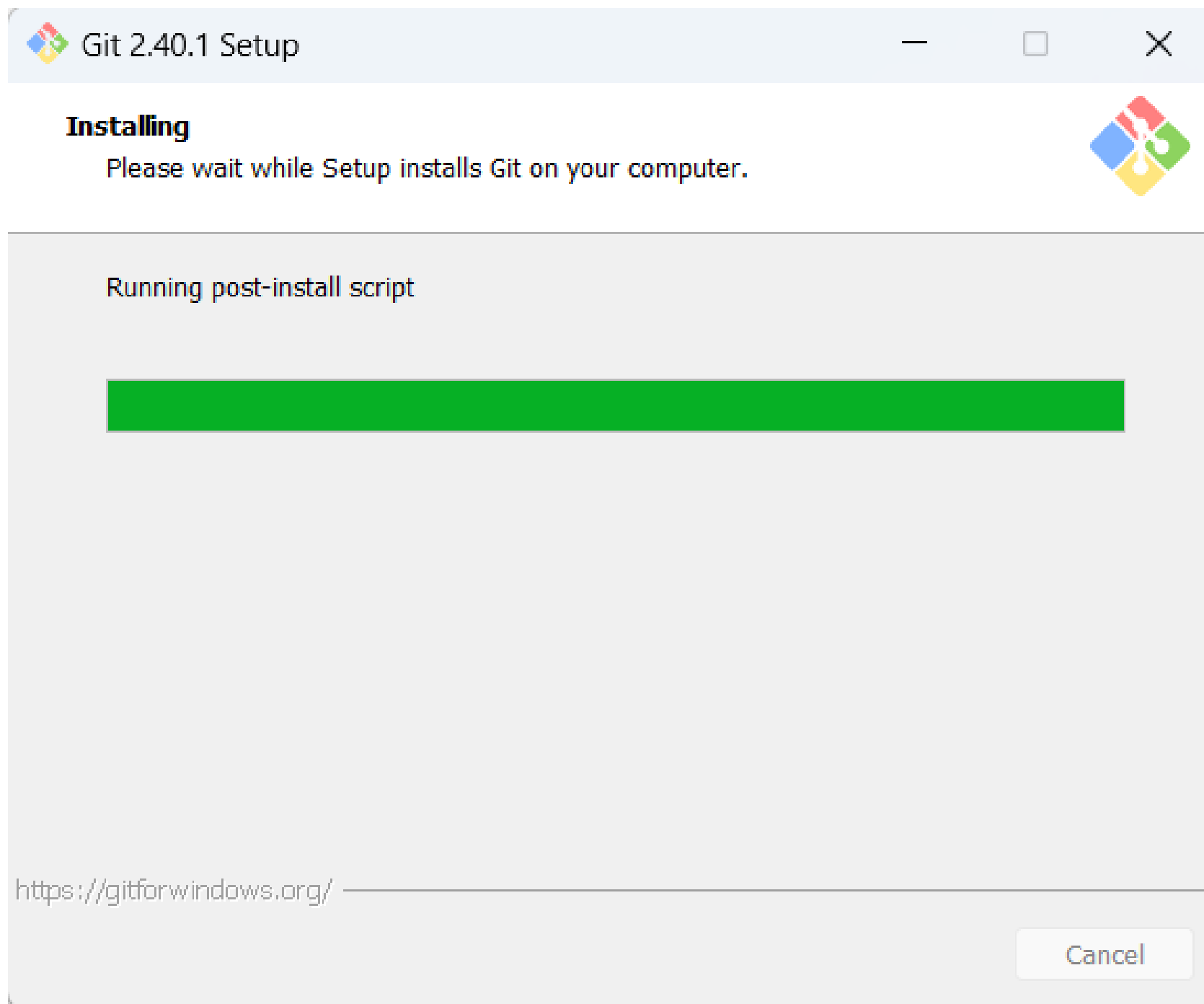
추가적으로 설치할 Component를  
선택하는 과정

: 기본적인 것만 설치합니다.

그 뒤로 모든 과정은 Next 해줍니다.







바탕 화면 > 새 폴더



새 폴더 검색



이름

수정한 날짜

유형

크기

이 폴더는 비어 있습니다.

보기(V) >  
정렬 기준(O) >  
분류 방법(P) >  
새로 고침(E)

현재 폴더 사용자 지정(F)...

붙여넣기(P)  
새로 만들기 취소(U) Ctrl+Z

터미널에서 열기(T)

Git GUI Here

Git Bash Here

Git을 사용하고자 하는 폴더에 들어가  
오른쪽 마우스를 클릭합니다.

Git Bash Here을 클릭하면 해당 폴더에서  
Git을 실행할 수 있습니다.



## 2. git 시작하기

### 1. 기본 환경설정

```
# Git 사용자 정보 설정 (터미널에서)
git config --global user.name "홍길동"
git config --global user.email "email@example.com"
```

```
# 설정정보 확인
git config --list
```

```
# 새 프로젝트 디렉토리 생성 및 초기화
mkdir my-project
cd my-project
```

```
# 현재 폴더를 Git 저장소로 초기화
git init
```

(1) git 의 핵심 폴더인 .git 폴더가 생성

(2) master 라는 기본 브랜치가 생성


.git 폴더: git 은 .git 폴더를 통해 해당 프로젝트에서 변경되는 파일들에 대한 추적/관리를 할 수 있음

## 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 \*      Repository name \*

 kimguppy /

Great repository names are short and memorable. Need inspiration? How about **urban-meme** ?

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.](#)

### Add .gitignore

.gitignore template: **None** ▾

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

### Choose a license

License: **None** ▾

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

 You are creating a public repository in your personal account.

Create repository

## 2. 원격저장소 생성



### Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot





### Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

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

 Set up in Desktop or **HTTPS** **SSH**  

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



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

```
git remote add origin https://github.com/kimguppy/gitnewbie.git
git branch -M main
git push -u origin main
```



### 3. 로컬저장소 생성 및 원격저장소 등록

```
# 원격 저장소 등록  
git remote add origin [저장소주소]
```

```
# 현재 등록된 원격저장소  
git remote -v
```

[저장소주소] : 깃허브에서 생성했던 레포지토리의 주소 삽입

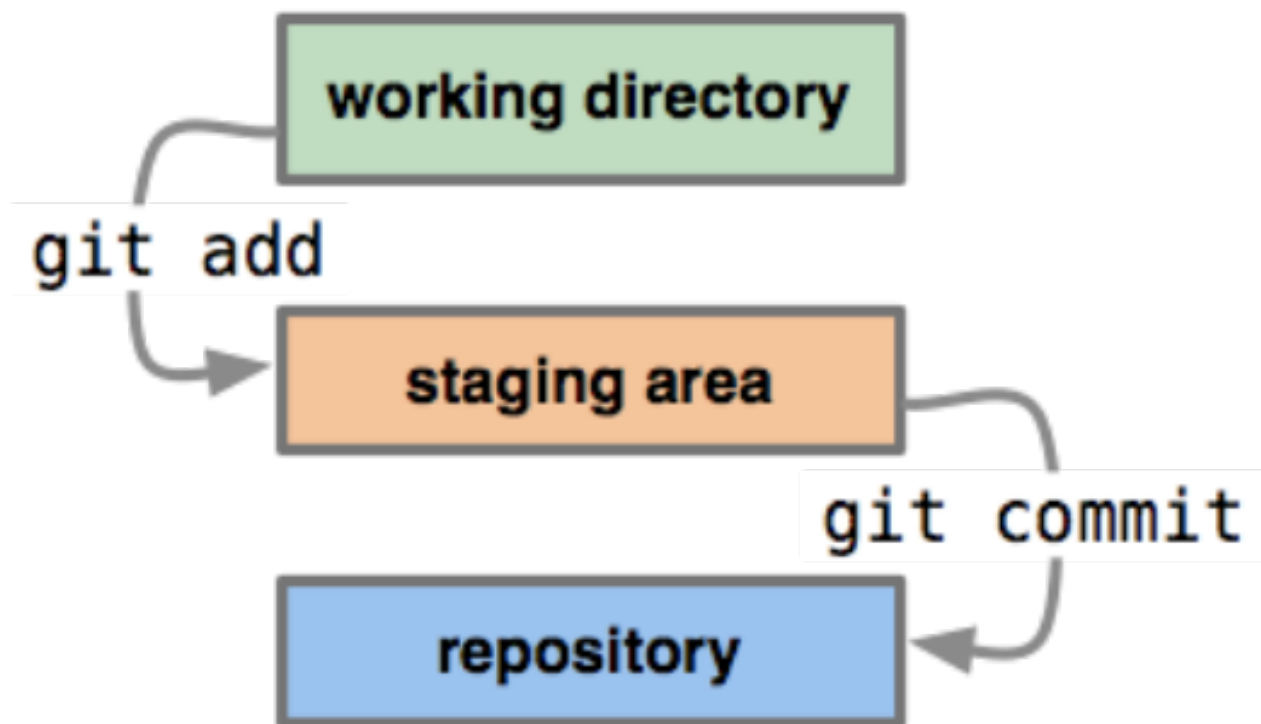
```
# 원격 저장소와 연결 시  
origin은 저장소에 쉽게 접근하기 위한 변수 (default, origin)
```

```
# 변경된 파일을 스테이징  
git add 파일명
```

```
# 변경 사항을 로컬 저장소에 저장  
git commit -m "커밋 메시지"
```

```
# 변경 사항을 원격 저장소에 저장  
git push -u origin master
```

```
# 원격 저장소의 변경사항을 로컬에 저장  
git pull
```



## git add vs git commit

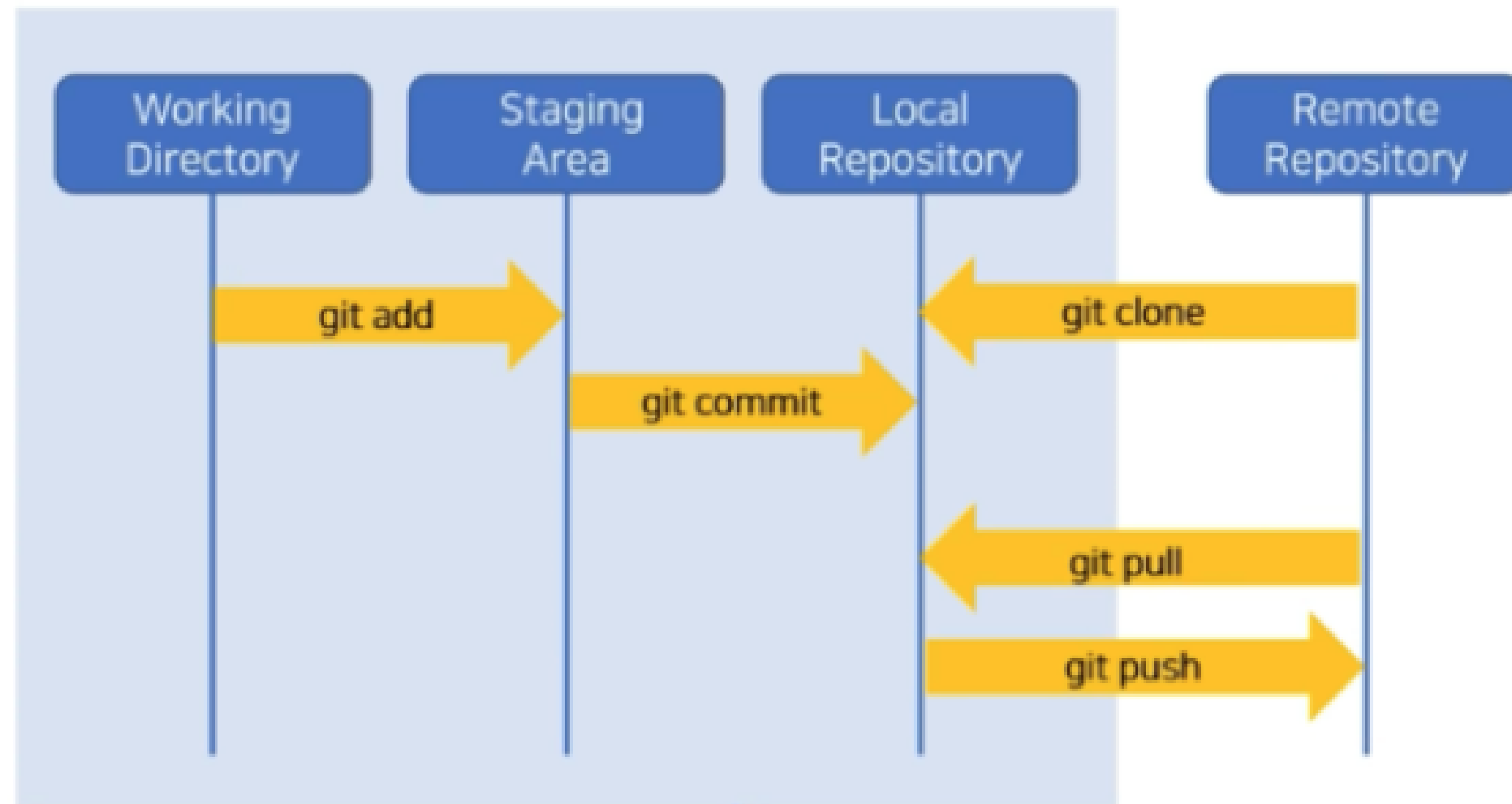
git add: 파일들을 다음 커밋에 포함(스테이징)

git commit: 스테이징된 파일들을 하나의 버전(스냅샷)으로 저장

git add = "상자에 파일을 담는 것"

git commit = "그 상자를 테이프로 포장하고 라벨 붙여서 기록으로 남기는 것"

## 4. 기본 플로우 실습



**git 기초 완료**

저장소를 생성하는 두가지 방법

1. clone 하기
2. 로컬저장소 생성 → 초기화(git init)  
→ 원격저장소 연결(git remote add)

## 5. 그외 자주 쓰는 Git 명령어 요약

명령어	설명
git status	현재 상태 확인
git log	커밋 내역 보기
git diff	변경사항 비교
git clone	원격 저장소 복제
git pull	원격 저장소 내용 가져오



# 감사합니다