

Git

Git 이란?!

- DVCS(Distributed Version Control System): 분산 버전 관리 시스템
- 소프트웨어 개발 과정에서 소스 코드의 변경 사항을 추적하고 관리하는 데 사용
- 많은 개발사에서 사용 중
- 만약... 내가 짠 소스를 어제로 돌리고 싶다면?
- 만약... 내가 테스트로 기능을 몇가지 추가 하고 싶다면?
- 나의 소스코드를 협업을 하고 싶다면?

Git 사용 방법

- CLI (Command-Line Interface) : 명령어 기반 인터페이스, 터미널, 콘솔, 명령 프롬프트(cmd)
- GUI (Graphical User Interface) : 그래픽 기반 인터페이스, GitHub Desktop, SourceTree

Git 설치

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

Download for Windows

[Click here to download](#) the latest (**2.47.0**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **8 days ago**, on 2024-10-08.

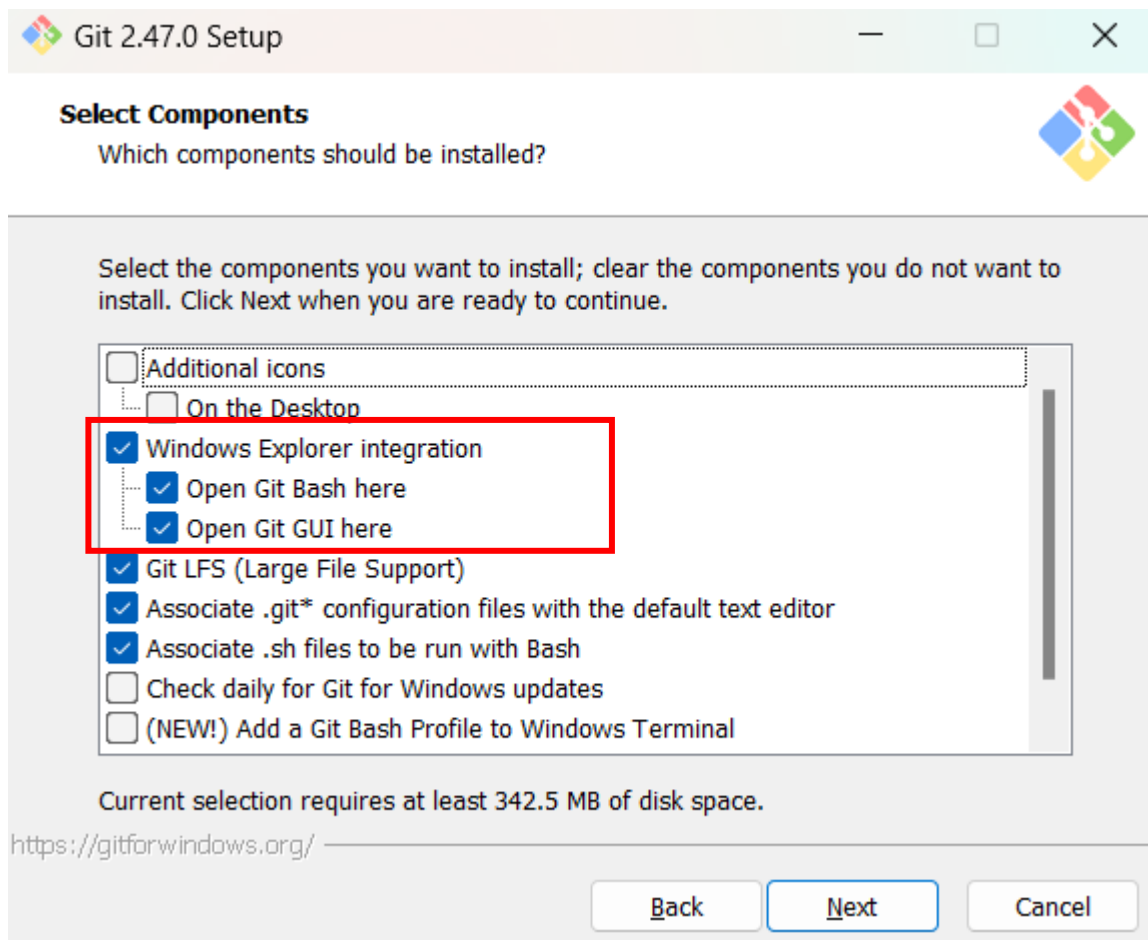
Other Git for Windows downloads

Standalone Installer

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

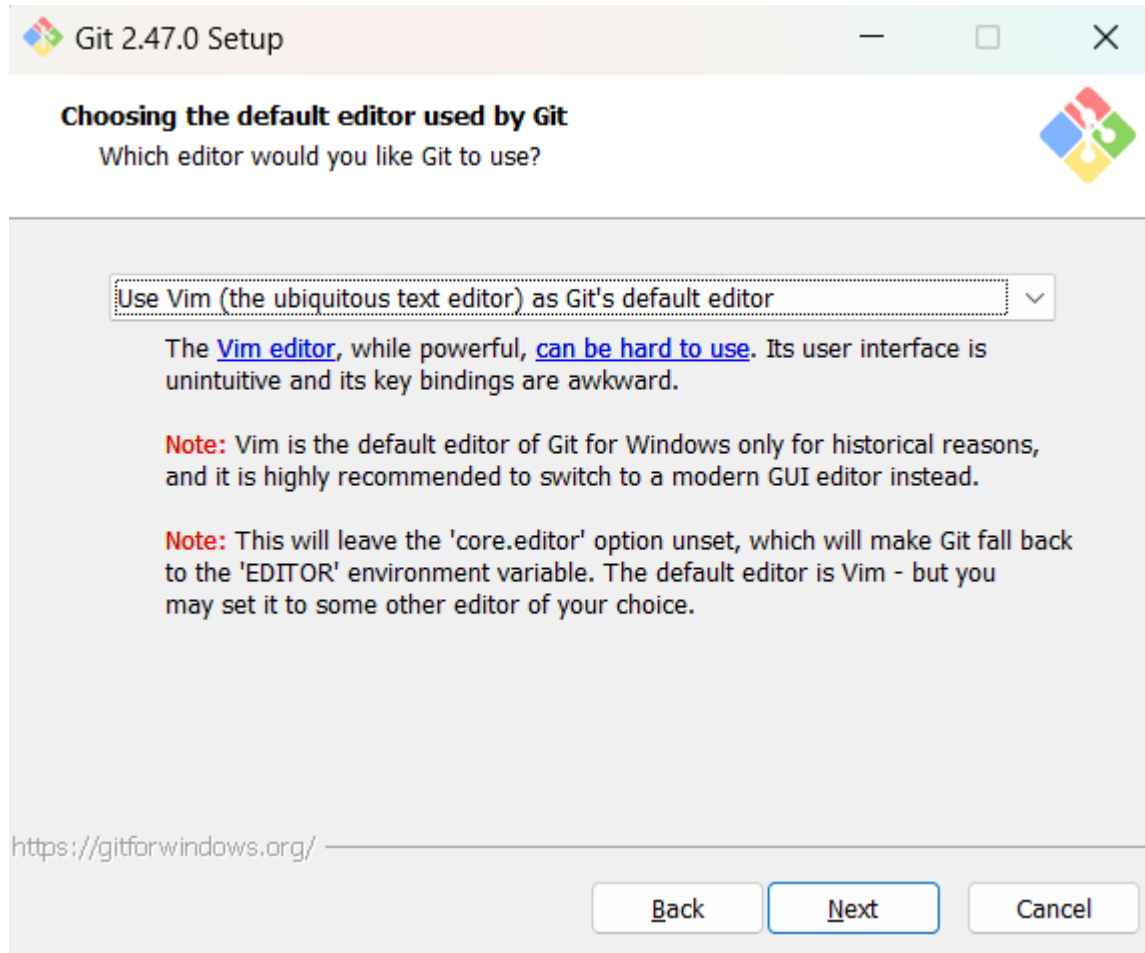
[64-bit Git for Windows Setup.](#)

Git 설치



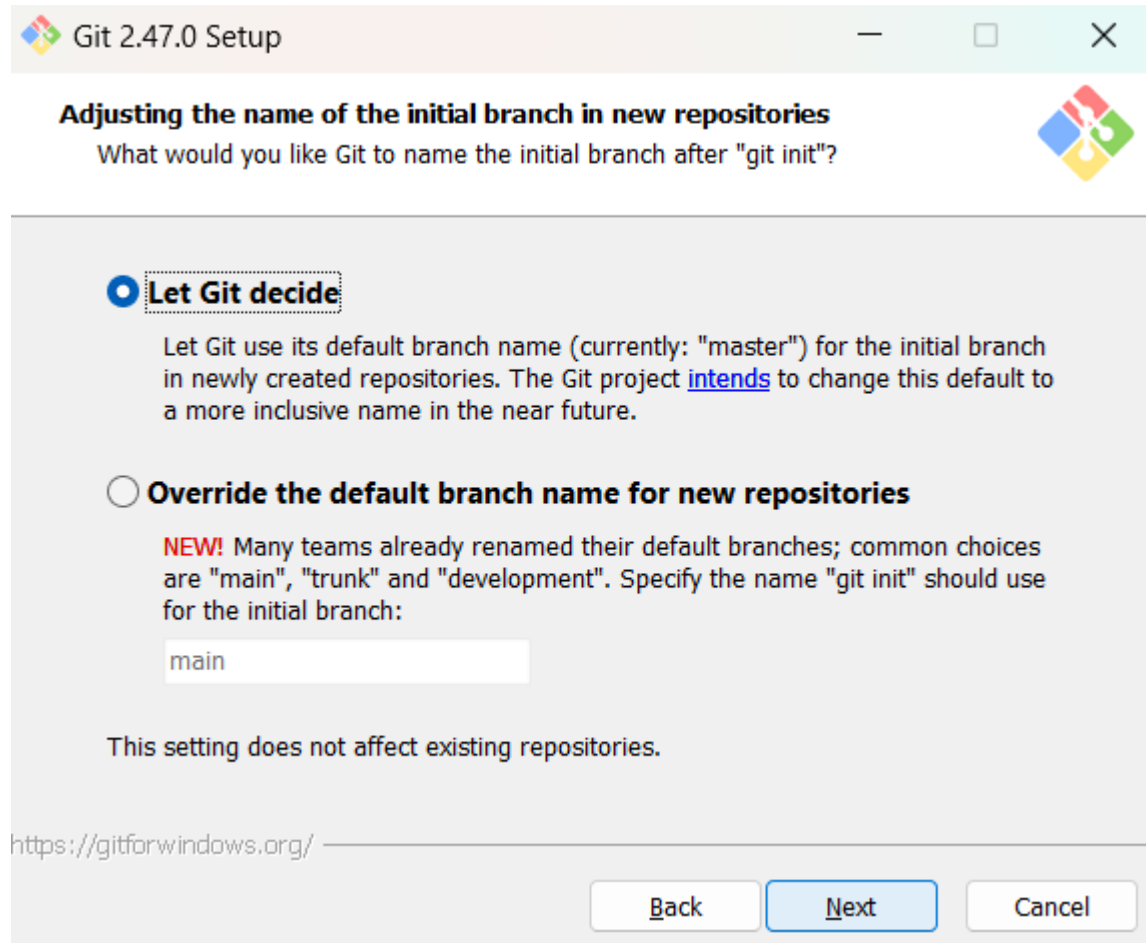
꼭! Git bash 포함되었는지 확인

Git 설치



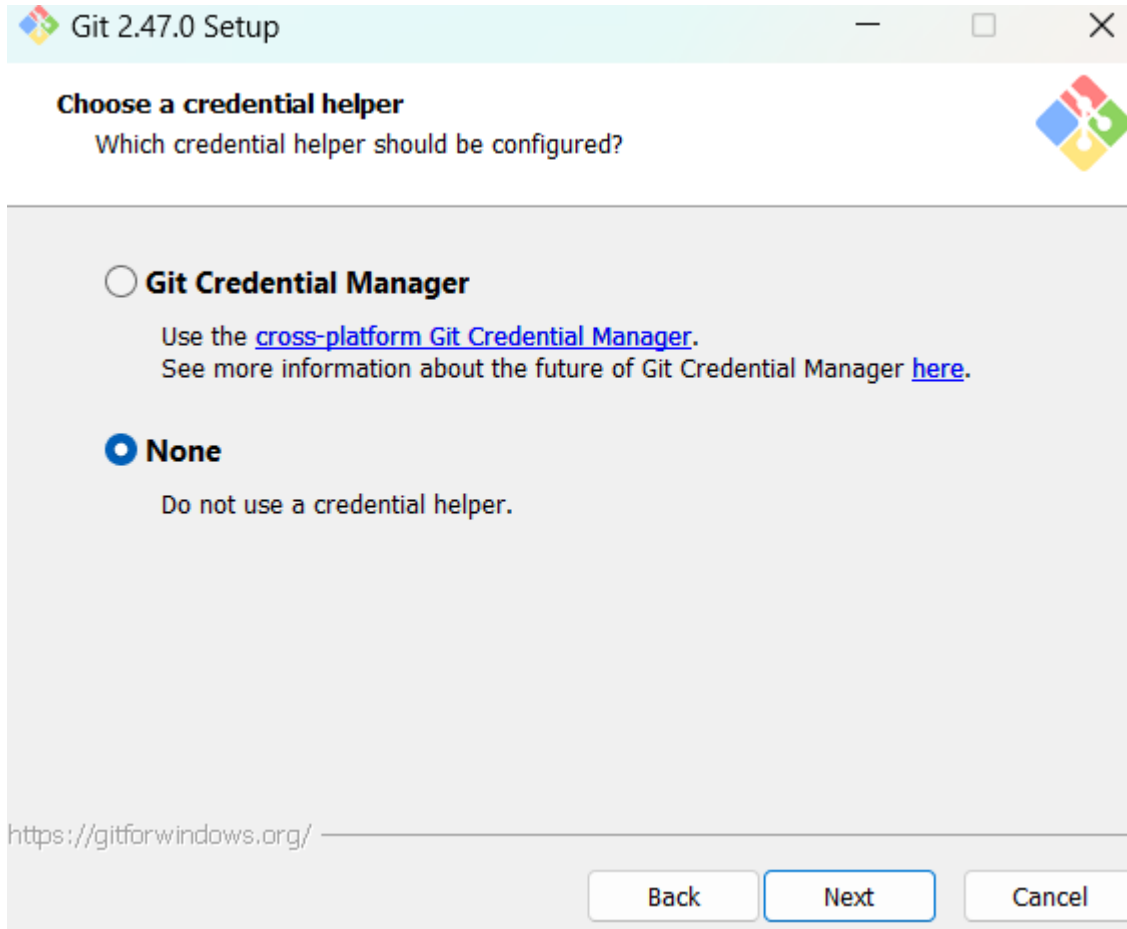
Git을 사용할 에디터 선택!
자신에게 익숙한 에디터 선택!

Git 설치



Git의 기본 분기 이름 사용
(Master)

Git 설치

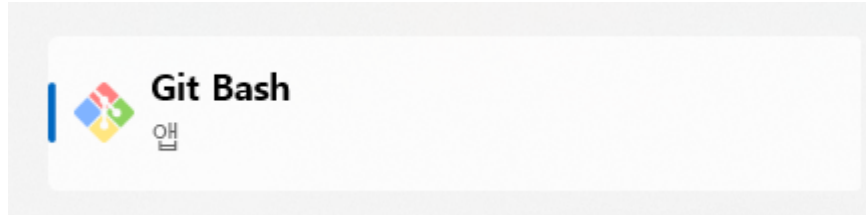


**None : Git에서 인증정보 입력마다
매번 사용자 이름과 비밀번호 입력**

**개인 PC의 경우,
Git Credential Manager 체크**

Git 설치 확인

- git bash 열기



- git --version : git 버전 확인
- git config --global core.autocrlf true : 윈도우와 맥의 엔터방식의 인한 오류 방지

```
MINGW64:/c/Users/gram15


admin@DESKTOP-15RVNPD MINGW64 ~
$ git --version
git version 2.47.0.windows.1

admin@DESKTOP-15RVNPD MINGW64 ~
$ git config --global core.autocrlf true

admin@DESKTOP-15RVNPD MINGW64 ~
$
```

SourceTree 설치

- <https://www.sourcetreeapp.com/> 다운로드 후 설치, BitBucket 계정 관련은 건너뛰기, 설치 중 Mercurial 체크 해제

 **SourceTree**

☒ Install


☒ Registration

☒ 도구 설치

☐ **Preferences**

Preferences

Before we finish, take a moment to configure these settings.

 SSH 키를 불러오시겠습니까?

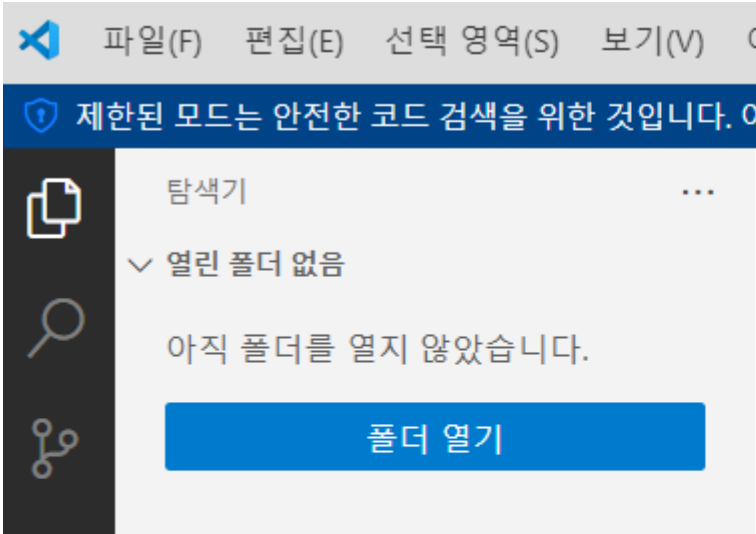
불러올 SSH 키를 하나 가지고 계십니까? 없으면 '아니요'를 누르고 나중에 하나 만드세요.

예

아니오

다음

VSC 세팅



탐색기

열린 폴더 없음

아직 폴더를 열지 않았습니다.

폴더 열기

Ctrl + Shift + P

Select Default Profile 검색하여 선택

Git Bash 선택

>select de

- 디버그: 디버그 세션 시작
Debug: **Select** Debug Session
- 디버그: 디버그 콘솔 선택
Debug: **Select** Debug Console
- 디버그: 디버깅 선택 및 시작
Debug: **Select** and Start **Debugging**
- 터미널: 기본 프로필 선택
Terminal: **Select** Default Profile
- Quick Open에서 다음 선택
Select Next in Quick Open
- Quick Open에서 이전 선택
Select Previous in Quick Open

기본 터미널 프로필 선택

| | | |
|--------------------|---|-----|
| Command Prompt | C:\Windows\System32\cmd.exe | 프로필 |
| Git Bash | C:\Program Files\Git\bin\bash.exe --login -i | ⚙️ |
| PowerShell | C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe | |
| Windows PowerShell | C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe | 검색됨 |

git 계정 정보 세팅

- Git 전역으로 사용자 이름과 이메일 주소 설정

#사용 방법

```
git config --global user.name " (본인 이름) "
```

```
git config --global user.email " (본인 이메일) "
```

#예

```
git config --global user.name " anhr "
```

```
git config --global user.email " anhr@shingu.ac.kr "
```

#확인

```
git config --global user.name
```

```
git config --global user.email
```

Git 기본 branch 세팅

- `git config --global init.defaultBranch main`
 - 기본 브랜치 명이 main
 - 사이트마다 상이하나 Master로 쓰는 경우도 있음

프로젝트 생성 및 git 관리 시작

- Git에서 사용할 디렉토리 생성 예) D:\git_it_a
- git bash에서 해당 디렉토리로 이동 예) cd /d/git_it_a

문제 출력 디버그 콘솔 터미널 포트

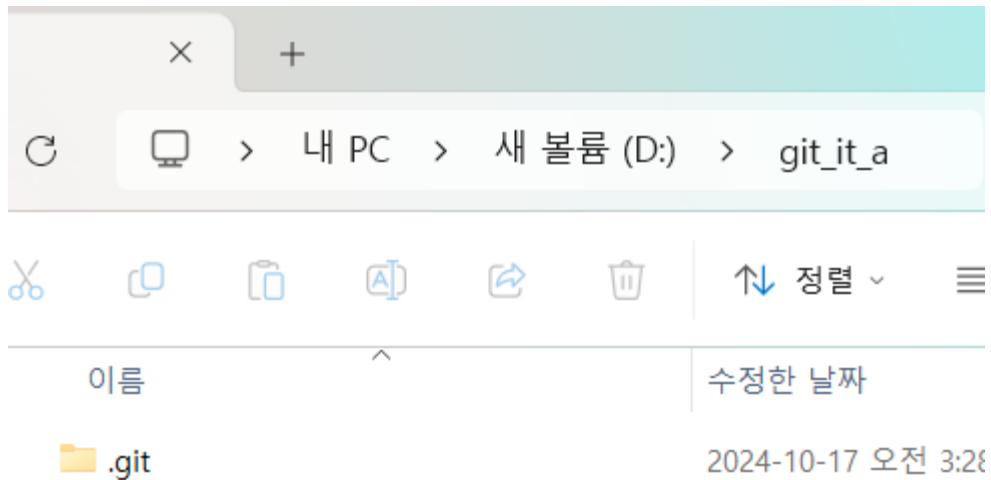
```
admin@DESKTOP-15RVNPD MINGW64 /  
$ cd /d/git_it_a
```

- 디렉토리를 git 리포지토리로 변환
git init

```
admin@DESKTOP-15RVNPD MINGW64 /d/git_it_a  
$ git init  
Initialized empty Git repository in D:/git_it_a/.git/
```

프로젝트 생성 및 git 관리 시작

- D:\Wgit_it_a 디렉토리로 이동하여 .git폴더가 생성됐는지 확인

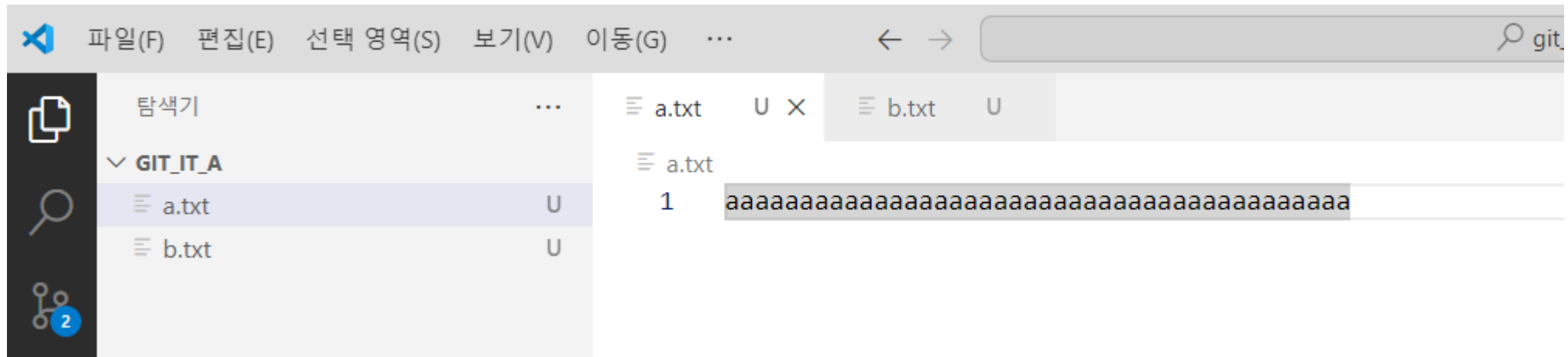


.git : git으로 관리하는 내역들이 저장되는 곳

.git 지울 시, 복구 불가!

git 관리

- D:\git_it_a 디렉토리 밑에 a.txt와 b.txt 생성!



git 관리

- VSC 터미널 : git status

문제 출력 디버그 콘솔 터미널 포트

```
admin@DESKTOP-15RVNPD MINGW64 /d/git_it_a (main)
```

```
$ git status
```

```
On branch main
```

```
No commits yet
```

```
Untracked files:
```

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

```
  a.txt
```

```
  b.txt
```

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

git 관리

- git status

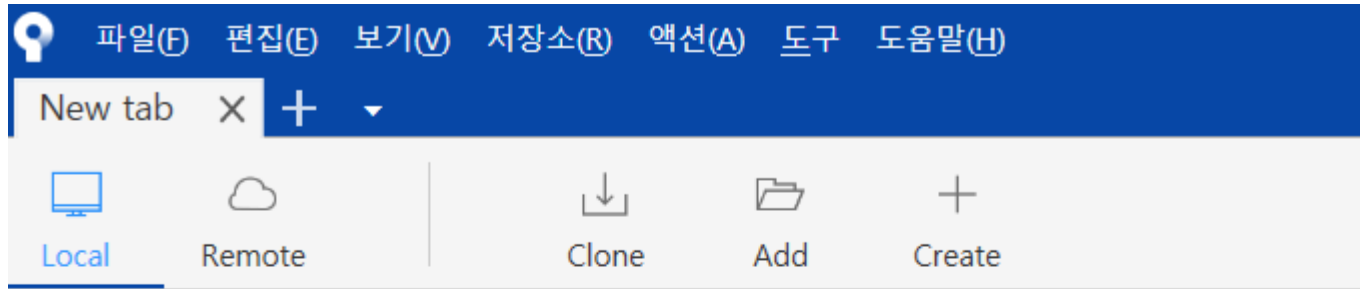
→ 추적되지 않은 파일들: Git이 아직 관리하지 않는 파일들을 보여줌

→ 변경된 파일들: Git이 추적하고 있는 파일 중 수정되었으나 아직 스테이징되지 않은 파일들을 보여줌

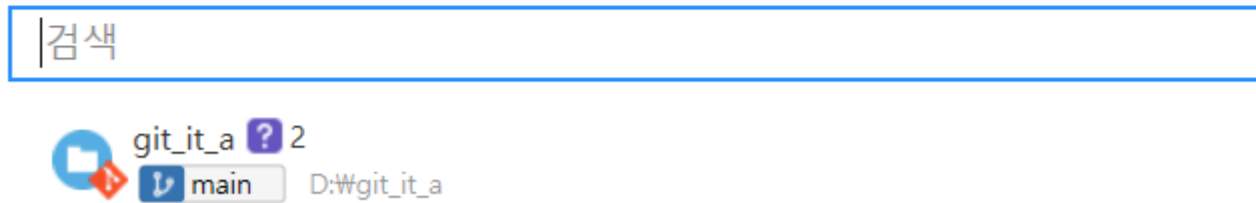
→ 스테이징된 파일들: 다음 커밋에 포함될 준비가 된, 즉 git add를 통해 스테이징된 파일 목록을 보여 줌.

git GUI로 보기

- 소스트리를 열어 해당 폴더 드레그 하여 넣기
- 해당 폴더 확인



로컬 저장소

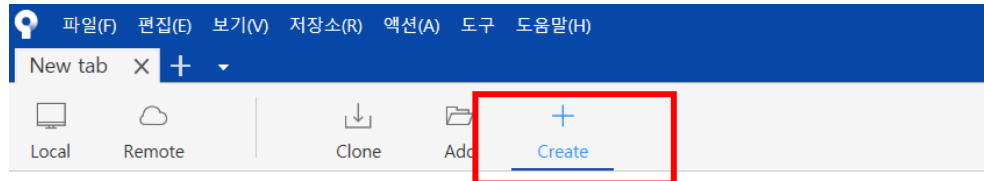


source tree를 통하여 로컬저장소 생성

- D:\git_it_a\ 디렉토리에서 .git 삭제
- git bash에서 확인

```
admin@DESKTOP-15RVNPD MINGW64 /d/git_it_a (main)
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

- source tree → create



Create a repository

목적지 경로:

이름:

Git

☐ 계정에 저장소 생성하기

source tree를 통하여 로컬저장소 생성

Create a repository

Git

▼

☐ 계정에 저장소 생성하기

생성

git_it_a

×

+

←

→

↑

↻

🖨

>

내 PC

>

새 볼륨 (D:)

>

git_it_a

⊕ 새로 만들기 ▼

✂

📄

📁

🔗

🗑

↕ 정렬 ▼

☰ 보

🖼 사진

🎵 음악

🎬 동영상

📁 스크린샷

📁

이름

📁 .git

📄 a.txt

📄 b.txt

수정한 날짜

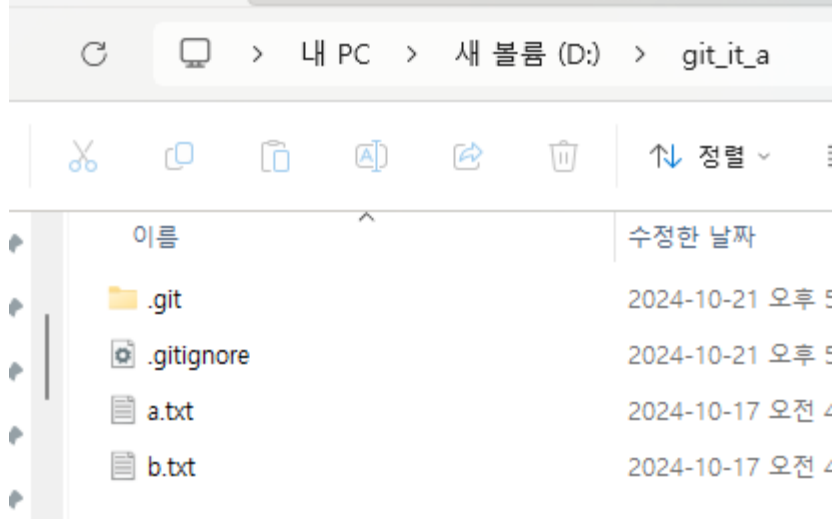
2024-10-17 오전 4:46

2024-10-17 오전 4:03

2024-10-17 오전 4:03

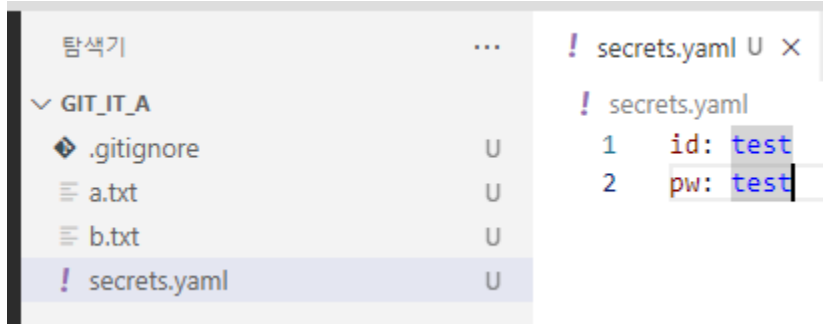
git 예외처리 방법 .gitignore

- .gitignore 파일을 사용하여 배제 가능
- 예) 로그파일, 빌드결과물, 보안상 민감한 파일, .DS_store 등..
- .gitignore 파일 생성



실습).gitignore 파일로 예외처리 하기

- secrets.yaml 파일 생성



- secrets.yaml 파일 확인

```
admin@DESKTOP-15RVNPD MINGW64 /d/git_it_a (main)
$ git status
On branch main

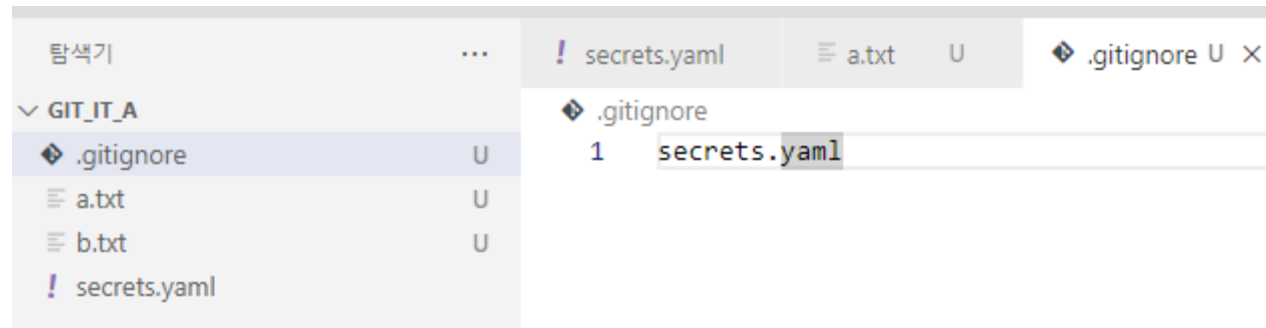
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    a.txt
    b.txt
    secrets.yaml

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

실습).gitignore 파일로 예외처리 하기

- .gitignore 파일에 secrets.yaml파일 추가하기



```
admin@DESKTOP-15RVNPD MINGW64 /d/git_it_a (main)
$ git status
On branch main

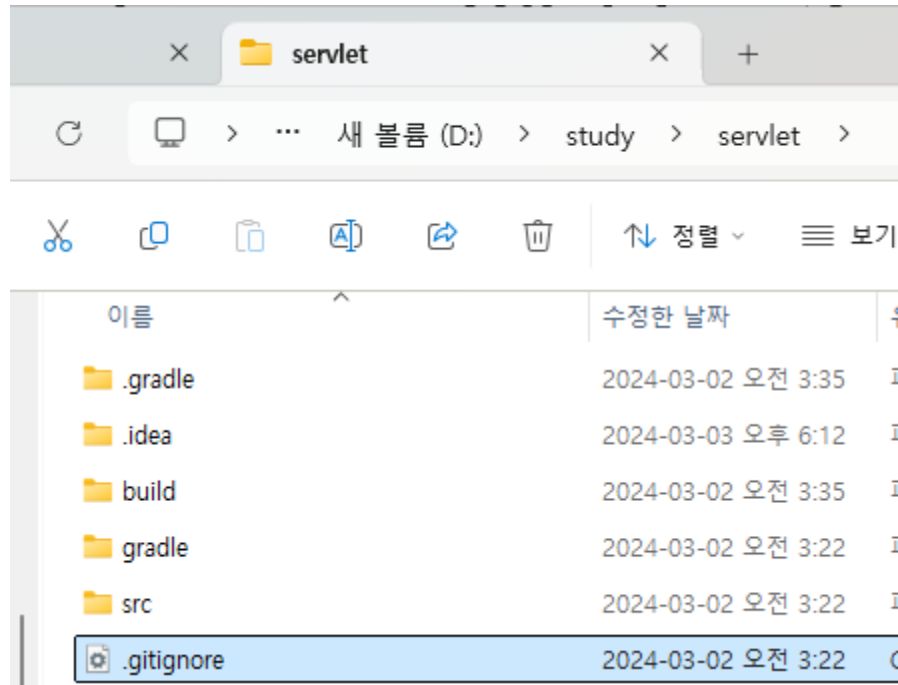
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    a.txt
    b.txt

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

.gitignore에 있는 파일은
예외처리 되는 것으로 확인!

Spring boots 에서 .gitignore 확인



```
.gitignore
1  HELP.md
2  .gradle
3  build/
4  !gradle/wrapper/gradle-wrapper.jar
5  !**/src/main/**/build/
6  !**/src/test/**/build/
7
8  ### STS ###
9  .apt_generated
10 .classpath
11 .factorypath
12 .project
13 .settings
14 .springBeans
15 .sts4-cache
16 bin/
17 !**/src/main/**/bin/
18 !**/src/test/**/bin/
19
20 ### IntelliJ IDEA ###
21 .idea
22 *.iws
23 *.iml
24 *.ipr
25 out/
26 !**/src/main/**/out/
27 !**/src/test/**/out/
28
29 ### NetBeans ###
30 /nbproject/private/
31 /nbbuild/
32 /dist/
33 /nbdist/
34 /.nb-gradle/
35
36 ### VS Code ###
37 .vscode/
```

.gitignore

- <https://git-scm.com/docs/gitignore> 에서도 사용법 확인 가능
- # : 주석
- 파일 이름을 지정하여 예외 : secrets.yaml
- 최상의 폴더의 파일 : /secrests.yaml
- 확장자로 지정하여 예외 : *.yaml
- yaml 확장자지만 무시하지 않을 파일 : !main.yaml
- logs란 이름의 파일, 폴더 그 내용들 : logs
- logs란 이름의 폴더와 그 내용들 : logs/
- logs 폴더 안에 txt파일 들 : logs/*.txt

실습 종료 후, 계정 초기화

Git bash에서 아래 git명령어 입력: 설정정보 초기화

```
git config --global --unset user.name
```

```
git config --global --unset user.email
```

git config --global --list 로 확인!

```
admin@DESKTOP-15RVNPD MINGW64 ~  
$ git config --global --list  
core.autocrlf=true  
difftool.sourcetree.cmd=''  
mergetool.sourcetree.cmd=''  
mergetool.sourcetree.trustexitcode=true  
init.defaultbranch=main
```