

소스코드 관리 도구 Git을 사용하여, GitHub와 연동하면 변경된 코드를 쉽게 확인할 수 있습니다. Github는 개발자 포트폴리오의 용도로 많이 사용됩니다.

많이 쓰는 에디터 VSCode를 이용해서 Git 설치 후 Github 계정에 연동 및 커밋하는 과정을 모두 담아 작성하였습니다. 아직 Git 사용법을 모르신다면, 천천히 따라해보세요!

앞서 확인하기

해당 조건들을 모두 만족하는지 확인 후, 다음 단계로 넘어가주세요.

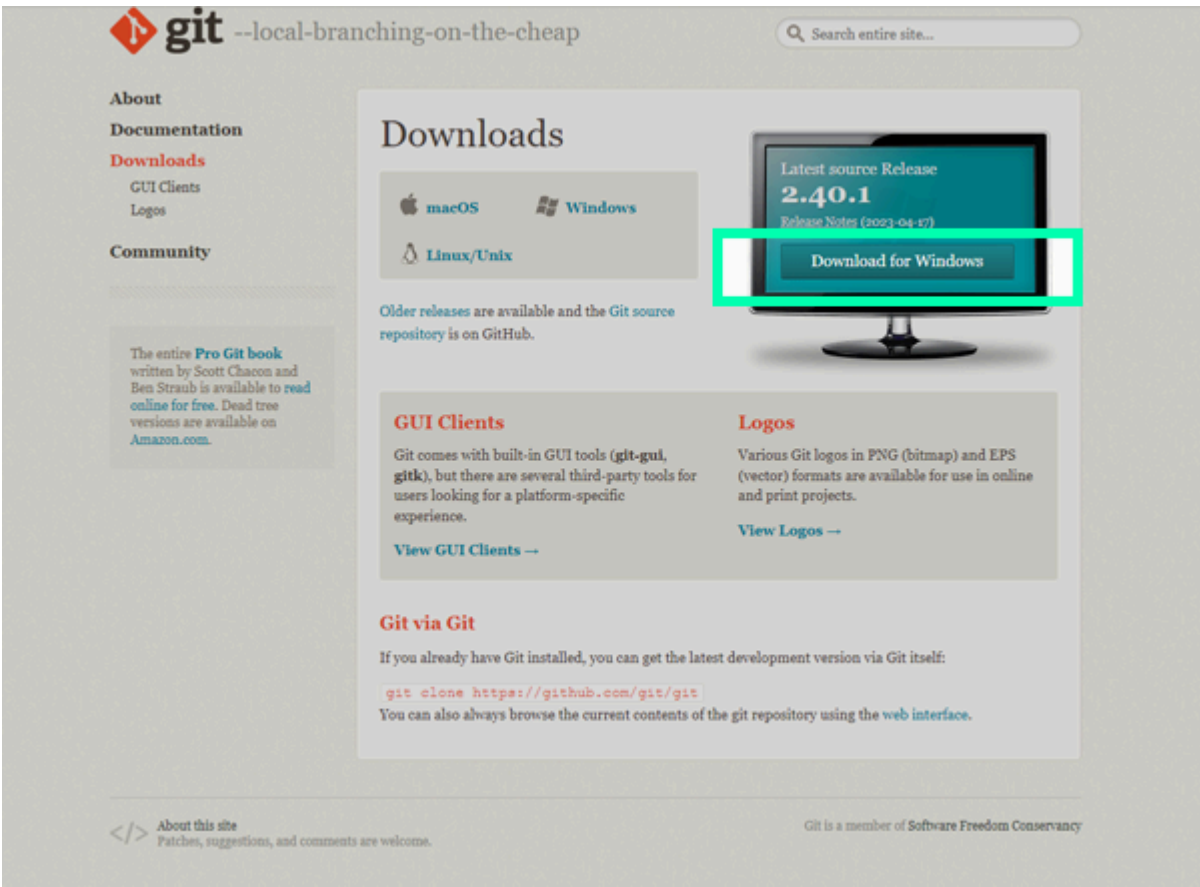
—

- 1. Visual Studio Code가 설치되어 있습니다.
☞ 아니오 >> **VSCode 다운로드하기 (Stable Build)**
- 2. 사용중인 GitHub 계정이 있습니다.
☞ 계정이 아직 없어요 >> **GitHub 계정 만들기**

—

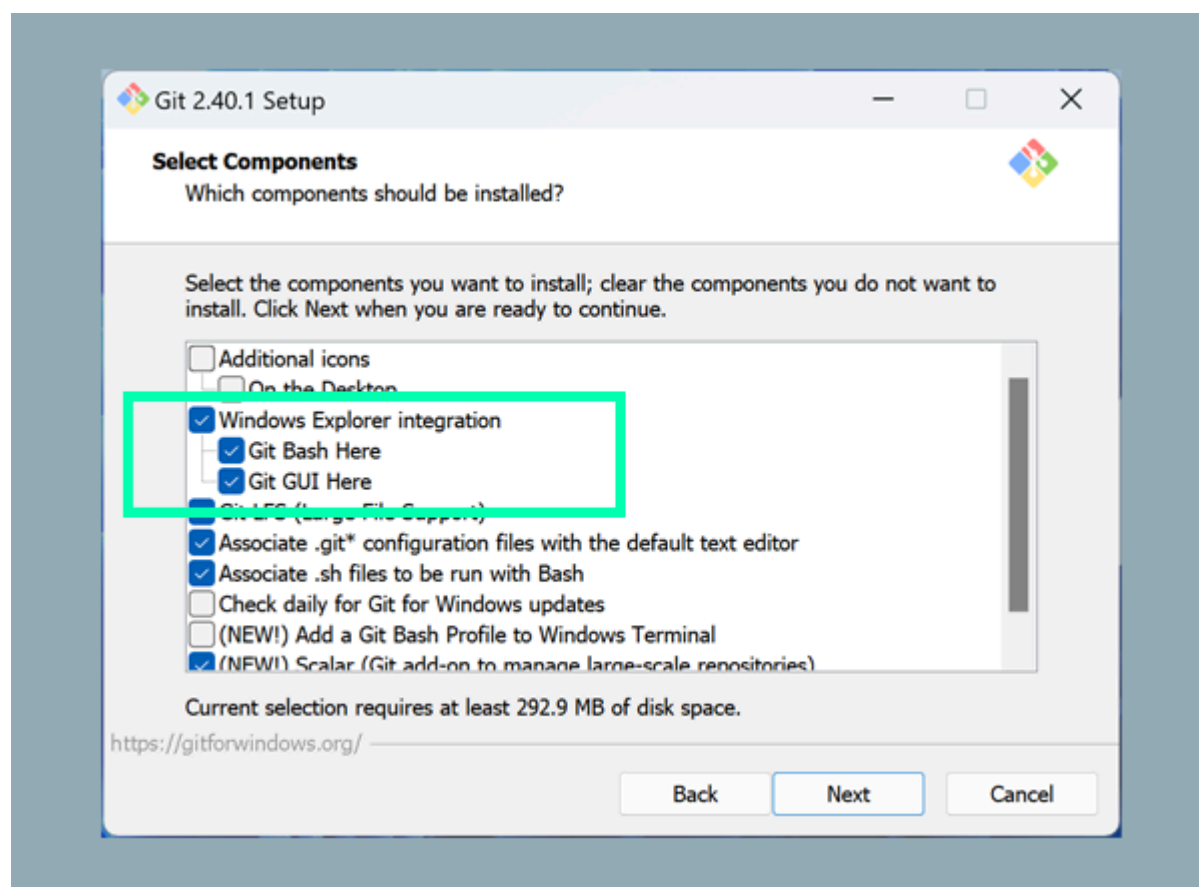
Git 설치하기

이미 설치되어 있다면, 넘어가셔도 좋습니다.

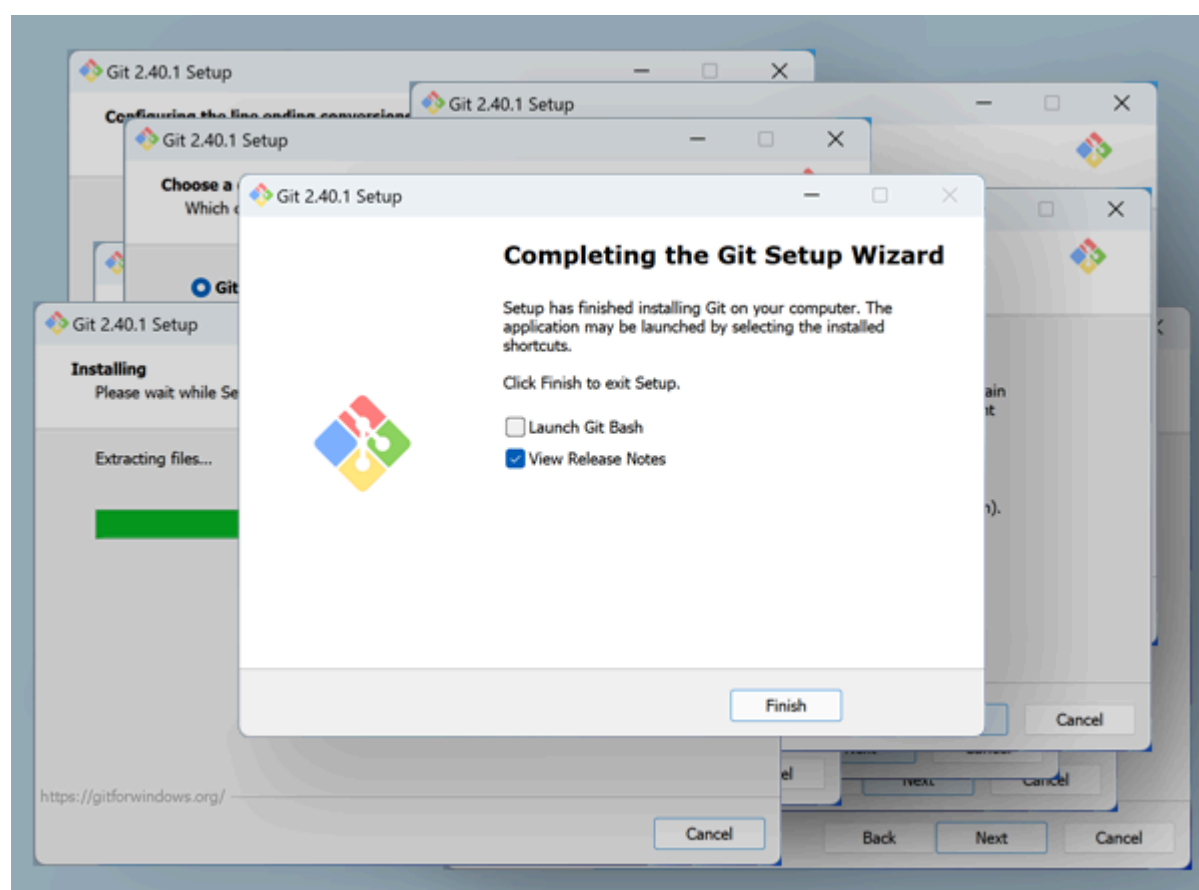


Git 홈페이지에서 운영체제에 맞는 릴리즈를 다운로드합니다. >> **Git 다운로드**

권한 동의 후 원하는 경로를 선택하고, 설치할 구성요소를 선택한 후 넘어갑니다.



Windows Explorer integration - 'Git Bash Here'을 선택하면, Windows 탐색기 우클릭 컨텍스트 메뉴에 'Git Bash here' 옵션이 나타납니다. 기본적으로 선택되어 있으니 넘어가셔도 좋습니다.



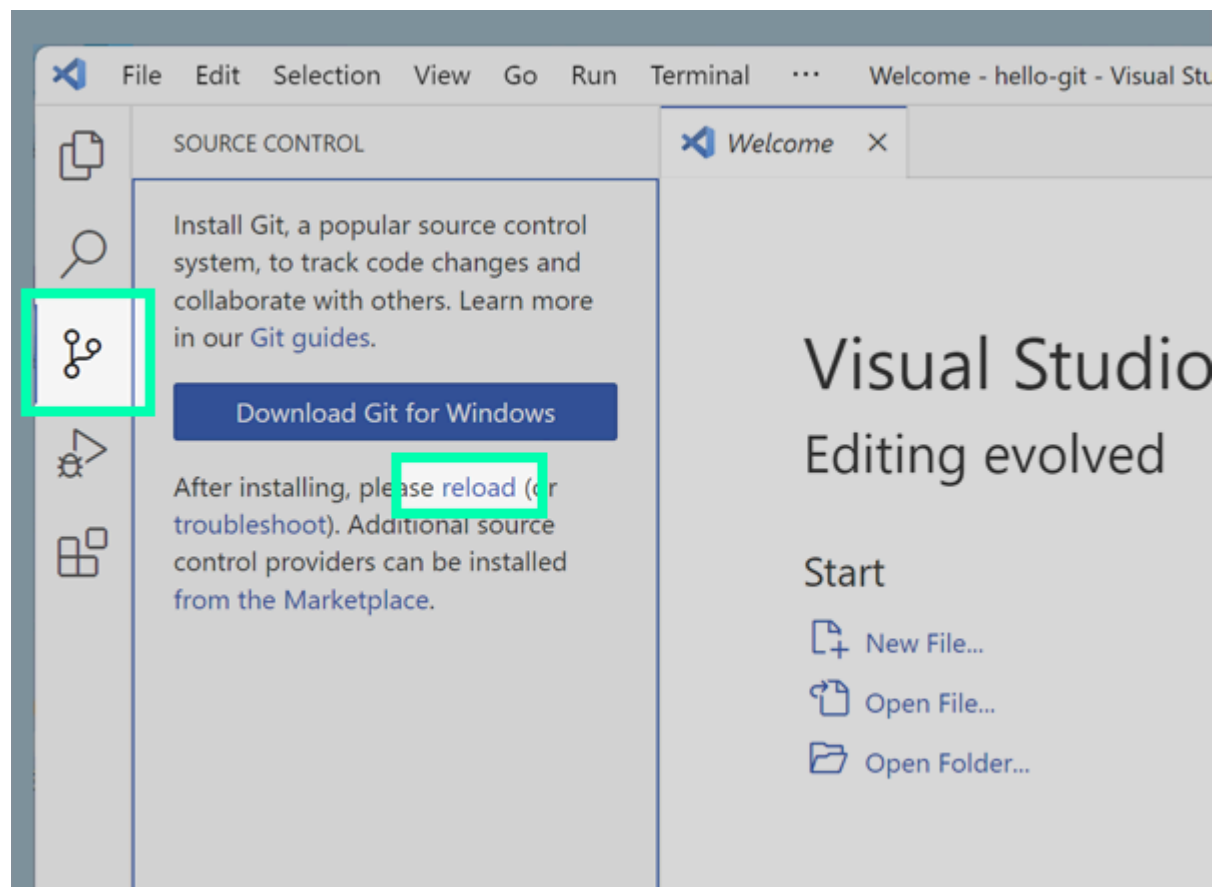
이후 일반적인 Git 사용을 위해서는 별도로 설정을 건드릴 필요가 없으므로, **모두 기본 선택** 사항으로 'Next' 선택 후 빠르게 설치를 마칩니다. (기본 에디터 선택 메뉴에서 Vim이 기본으로 선택되어 있으나, 필요에 따라 VSCode를 선택하셔도 됩니다.)

💬 조금 더 알아보고 설치하고 싶어요.

과정을 상세히 알려주는 좋은 글이 있어 아래 링크를 첨부해 드립니다. <https://xangmin.tistory.com/102>

VSCode에서 Git 설치 확인하기

Git 설치가 완료되었다면, 이제 VSCode와의 연동을 시작하겠습니다.



Visual Studio Code에서 좌측 메뉴 **세번째 탭**(소스 제어)을 열어줍니다. 앞으로 Git을 통한 커밋/동기화는 모두 '소스 제어' 탭으로 가능하게 됩니다.

만약 VSCode를 Git 설치 전에 열어두었다면, 새로고침 되지 않아 설치하라는 안내 문구만 나옵니다. **reload**(새로고침)을 누르거나 에디터를 다시 열어주세요.

GitHub 리포지토리와 연결하기

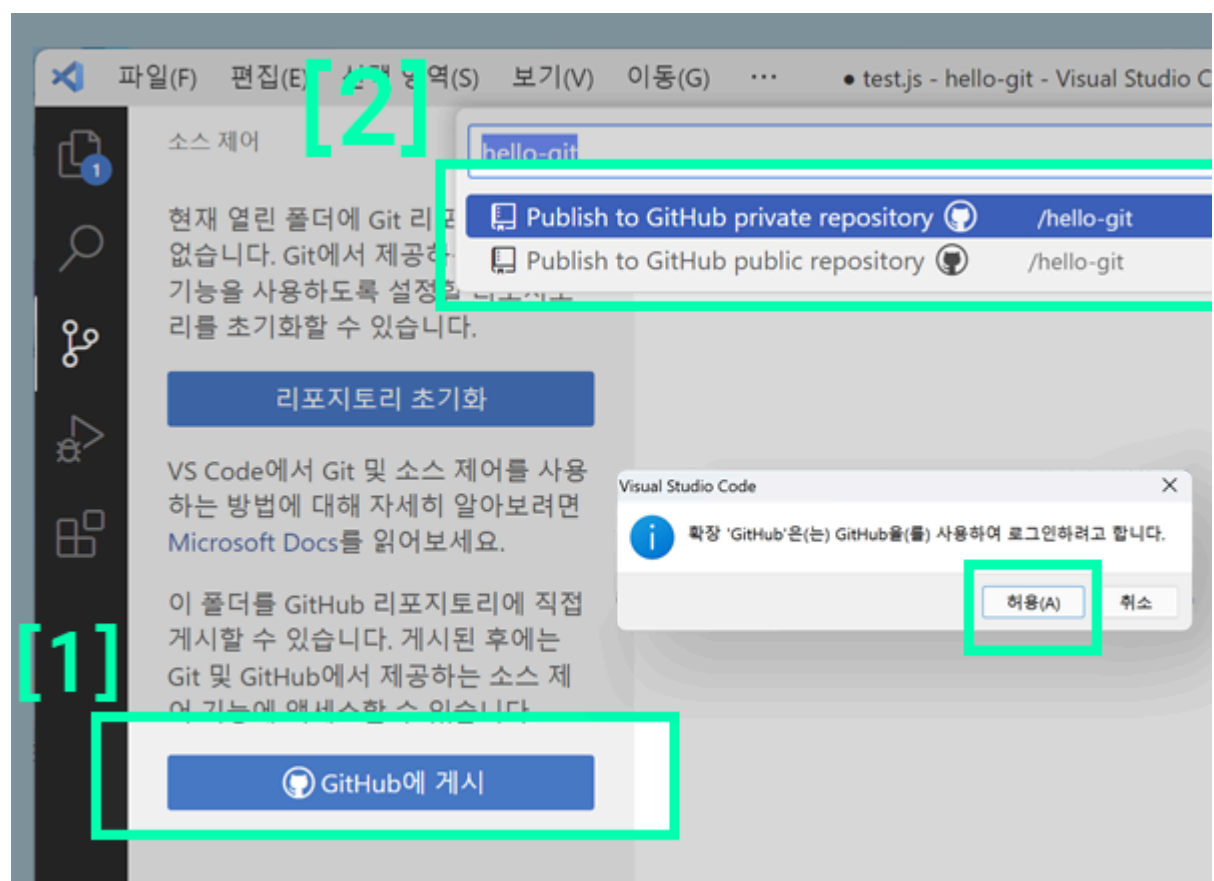
연결하기에 앞서, 두 가지 경우로 나눌 수 있습니다.

1. GitHub에 새롭게 업로드합니다.
2. 이미 GitHub에 직접 업로드한(생성한) 리포지토리가 있습니다.

먼저 '리포지토리(repository)'란, 하나의 프로젝트 또는 저장소와 같은 기능을 하는 단위입니다. 즉, 하나의 프로그램을 구성하는 모든 소스파일 및 구성요소들은 하나의 리포지토리에 들어가게 됩니다.

앞서 말씀드린 두가지 경우를 순서대로 알려드리겠습니다.

1. GitHub에 게시하기



GitHub에 업로드할 프로젝트의 **폴더를 열고**, 세번째 '소스 제어' 탭에서 '**GitHub에 게시**'를 선택합니다. 만약 GitHub 로그인이 필요하다면, '허용'을 눌러 GitHub 계정으로 로그인을 진행합니다. 상단의 팝업 창에서 Public과 Private를 선택 후 게시합니다.

- Public - 리포지토리에 포함된 소스와 파일이 모두 전체공개로 업로드됩니다. 원하는 사람은 모두 내 소스코드를 확인하고 내려받을 수 있습니다.
- Private - 리포지토리를 나만 확인할 수 있는 비공개로 게시합니다.

-

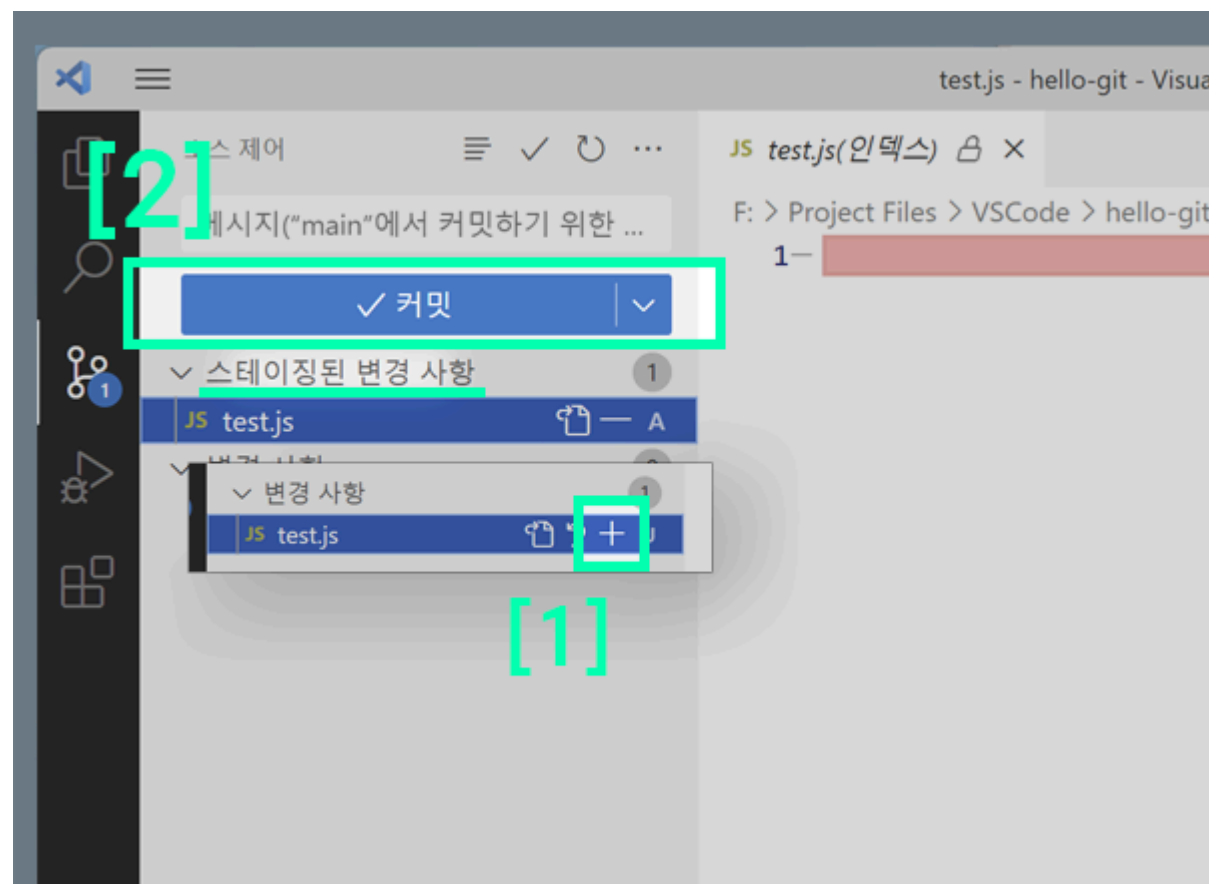
1-1. 소스코드 변경 후 커밋하기

프로그램을 변경하거나 소스코드를 추가했다면, 변경 사항을 커밋합니다.
이 과정을 거쳐야 GitHub에 업로드한 파일이 최신화됩니다.

💬 커밋(Commit)이 뭔지 모르겠어요!

커밋, 스테이징, 풀/푸시 등의 용어를 자세히 알려주는 좋은 글을 알려드립니다.

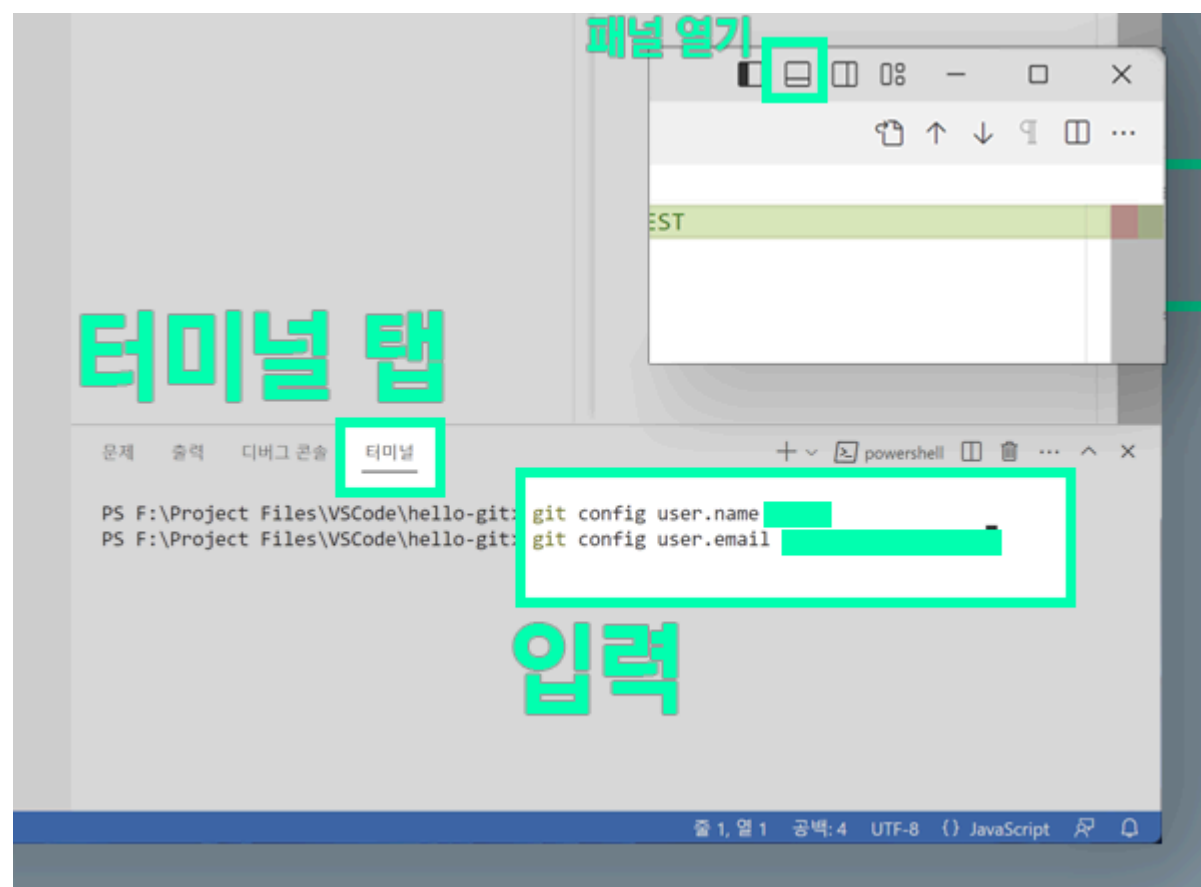
<https://parksb.github.io/article/28.html>



새로운 파일을 추가했거나, 내용을 덧붙였다면 '**변경 사항**' 목록에 표시됩니다.

'+' 버튼을 눌러 '스테이징된 변경 사항' 목록에 올립니다.

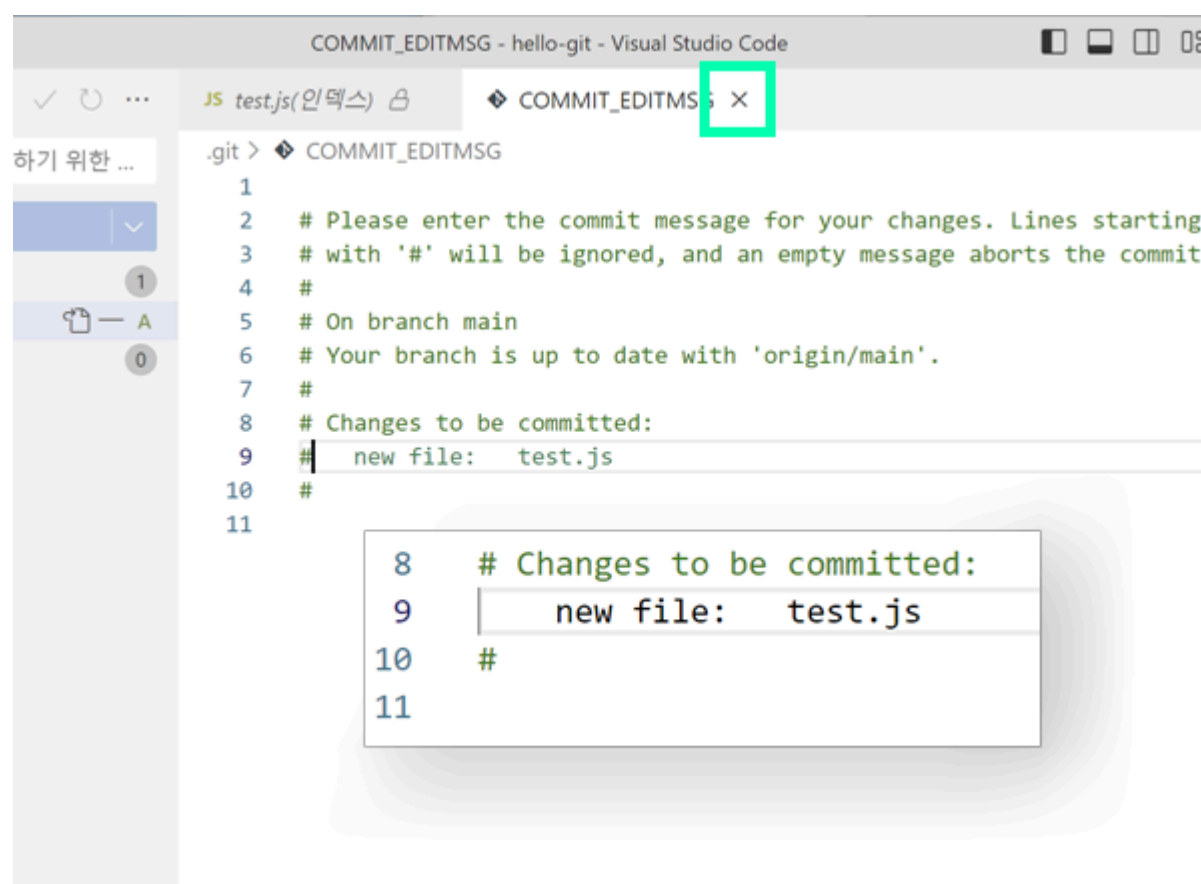
'커밋' 버튼을 눌러 변경 사항을 확정합니다. 스테이징하여 **인덱스 목록에 올린 파일만 커밋**됩니다.



커밋을 했을 때 최초 1회에 한해 id와 email을 입력하라는 문구가 출력될 수 있습니다.
우측 상단 2번째 버튼을 눌러 패널을 열고 터미널 탭에서 **아래 명령어를 입력**해주세요.

```
git config user.name (github 이름)
git config user.email (github 이메일)
```

터미널은 Ctrl+' 단축키를 통해서도 열 수 있습니다.



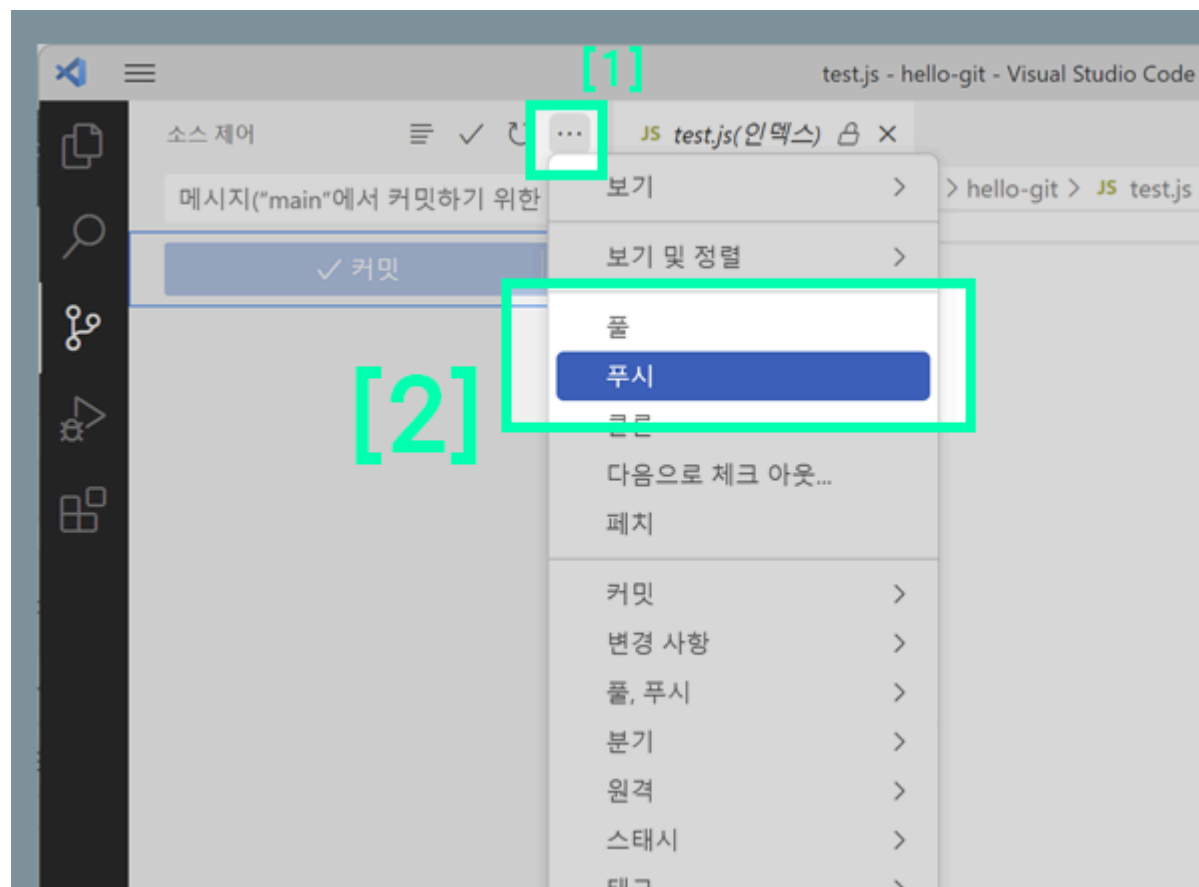
커밋 버튼을 누르게 되면, COMMIT_EDITMSG가 열리게 됩니다.
여러 파일을 동시에 스테이징했다면 이곳에 여러 파일의 목록이 표시됩니다.
커밋을 원하는 모든 파일의 앞에 붙은 '#'을 지워줍니다.
검은색 글씨가 되었다면, 그 파일은 커밋됩니다.

과정이 완료되었으면, Ctrl+S로 저장 후 창을 닫습니다.

1-2. 커밋이 끝났다면, 푸시(Push)하기

정말 마지막입니다.

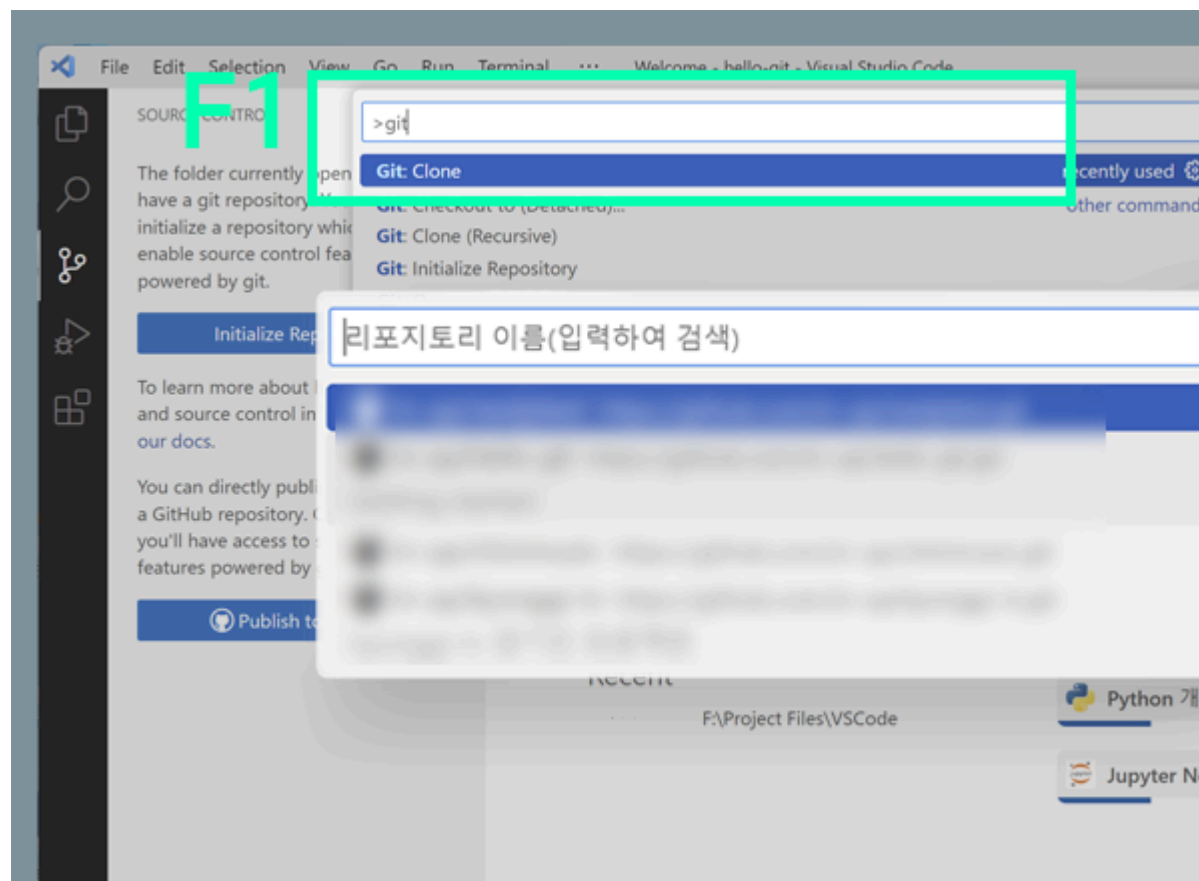
커밋을 완료해도, GitHub에서 리포지토리를 확인해보면 커밋 전과 똑같습니다. 커밋은 로컬 저장소에 등록하는 일이므로, 푸시(Push)까지 완료하여야 변경 사항이 원격 저장소에 반영됩니다.



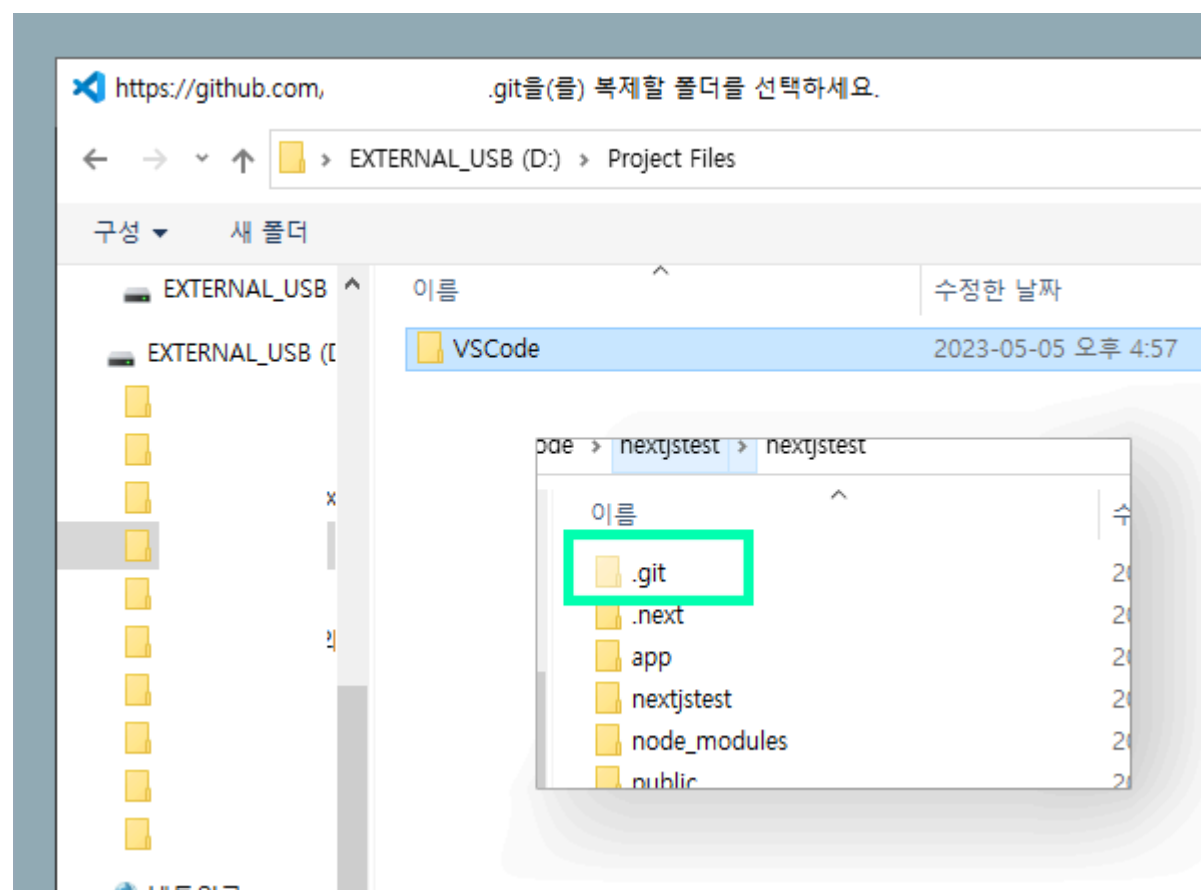
소스 제어 창의 '더보기(...)'를 눌러, 푸시를 선택해주세요.
GitHub에 다시 들어가 확인해보면, 최신화가 완료되었습니다.

2. 리포지토리 가져오기

다른 컴퓨터에서 작성한 소스파일을 가져오고 싶을 때, 리포지토리를 생성하였을 때 사용합니다.



'F1'을 눌러 명령 팔레트를 열고, **git clone**을 검색하여 선택합니다.
로그인한 계정의 리포지토리 목록이 나오면, 내 컴퓨터로 가져오고 싶은 리포지토리를 선택합니다.



적절한 위치를 선택 후 리포지토리를 로컬 저장소에 저장하면, .git 폴더가 생성되며 연동이 완료됩니다.