

git GUI applications



About..

컴퓨터소프트웨어공학과
김 원 일



목 차



- TortoiseGit
 - 다운로드
 - 설치
 - 설정
 - 사용
- GitHubDesktop
 - 설치와 설정

TortoiseGit

- 윈도우 git GUI 제공 프로그램
 - 공개 소프트웨어로 제한 없이 사용가능
 - 기본 git bash 대신 사용할 수 있는 GUI를 제공
 - 약간의 명령어 차이만 있을 뿐 동일하게 사용 가능
 - <https://tortoisegit.org/>



The banner features the TortoiseGit logo (a blue turtle) and the text "TortoiseGit Windows Shell Interface to Git". Navigation links include "About", "Download", "Support" (with a dropdown arrow), and "Contribute". The main headline reads "The Power of Git – in a Windows Shell". Below this, it states: "TortoiseGit provides overlay icons showing the file status, a powerful context menu for Git and much more!". A link says "Learn more [about TortoiseGit.](#)". A large white button with a download icon and the text "Download" is at the bottom left. The background illustration shows a person's hands holding a document with a Git logo, surrounded by a coffee cup, a smartphone, a calculator, glasses, and a notepad.

 **TortoiseGit**
Windows Shell Interface to Git

About Download Support ▾ Contribute

The Power of Git – in a Windows Shell

TortoiseGit provides overlay icons showing the file status,
a powerful context menu for Git and much more!

Learn more [about TortoiseGit.](#)

 **Download**



- 다운로드 프로그램

- 현재 2.12.0 버전이 안정화된 버전
- 설치 시스템에 맞는 설치 프로그램을 다운로드 받아 실행

[TortoiseGit.org](#) » Download

Download

The current stable version is: **2.12.0**

For detailed info on what's new, read the [release notes](#).

FAQ: [System prerequisites and installation](#)

[Donate](#)

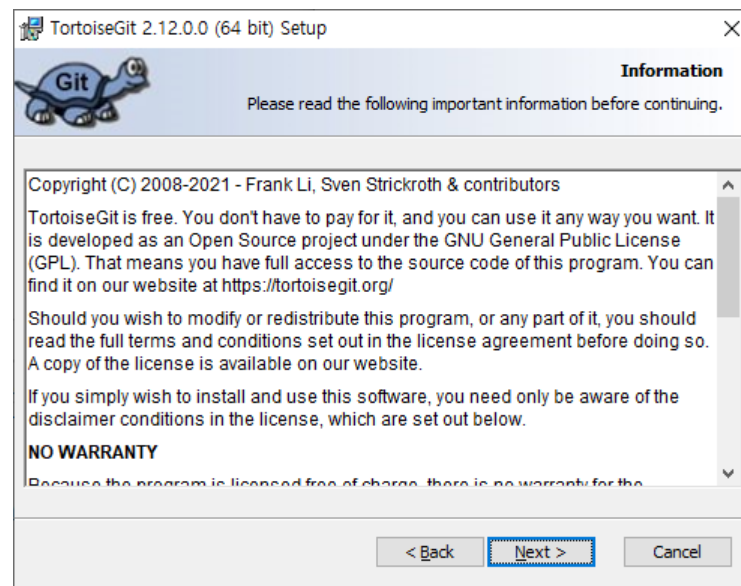
Please make sure that you choose the right installer for your PC, otherwise the setup will fail.

for 32-bit Windows	for 64-bit Windows
Download TortoiseGit 2.12.0 - 32-bit (~17.0 MiB)	Download TortoiseGit 2.12.0 - 64-bit (~19.6 MiB)



TortoiseGit 설치 - 1

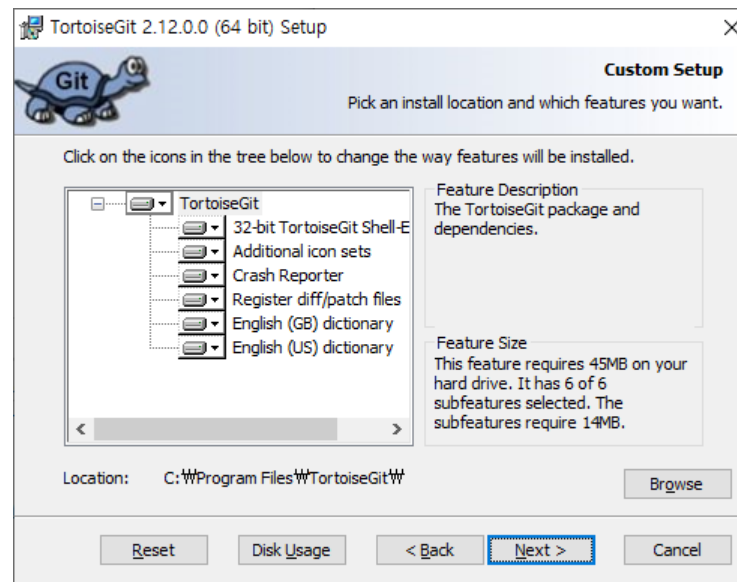
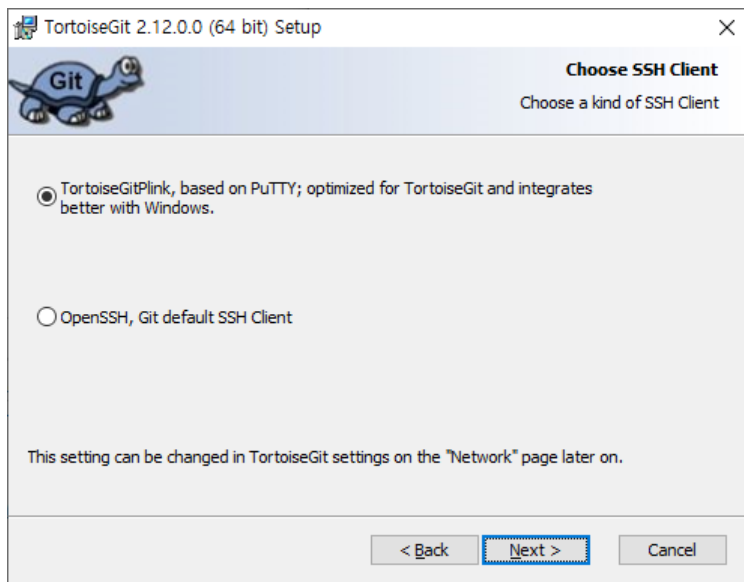
- 간단히 설치 가능
 - 라이선스 정보 명시





TortoiseGit 설치 – 2

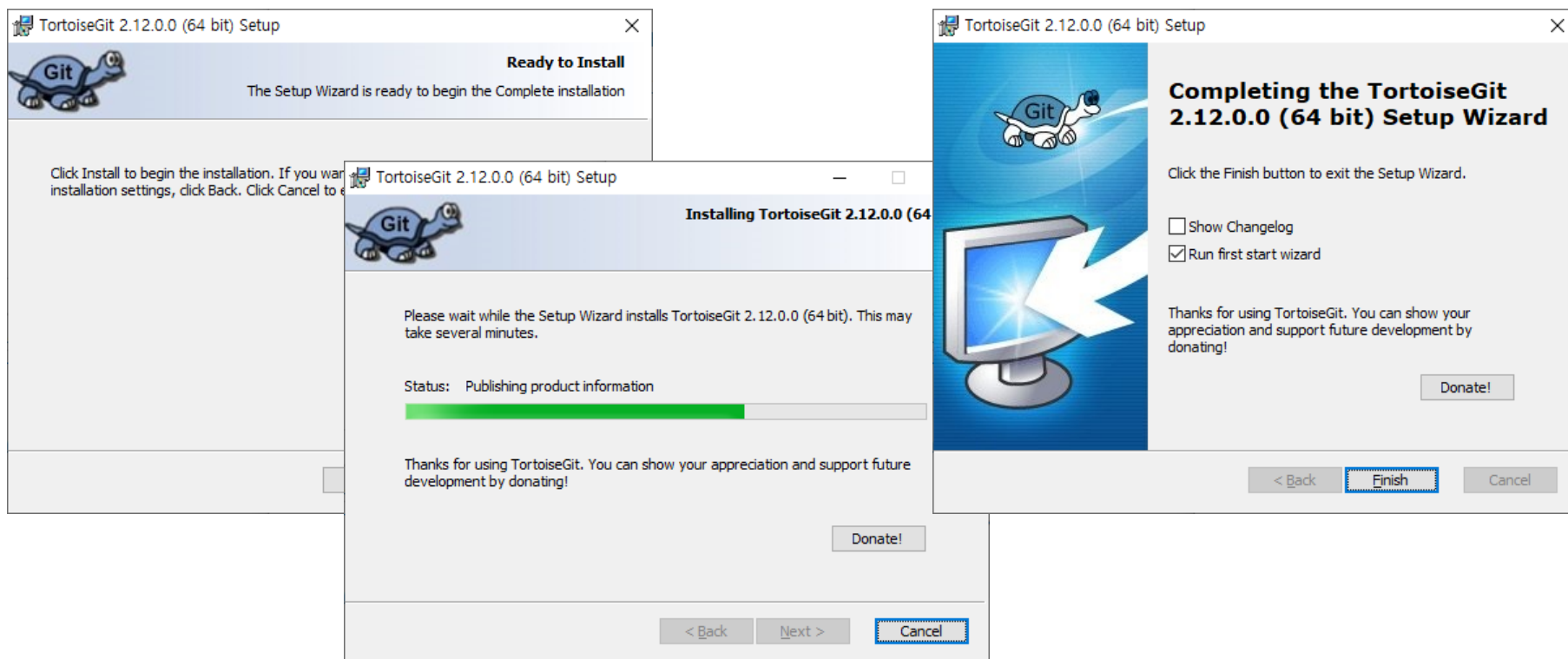
- SSH(Secure Shell) 클라이언트 선택
 - 보안 접속을 위한 방법을 선택
 - openSSH를 선택할 경우, openSSH가 별도로 설치되어 있어야 ga
 - 제공되는 SSH를 선택
 - 설치 구성요소 선택





TortoiseGit 설치 – 3

- 별도의 추가 선택 없이 설치
 - 설치가 매우 간편한 대시 설정 창이 복잡
 - 최초 실행 설정 마법사 실행



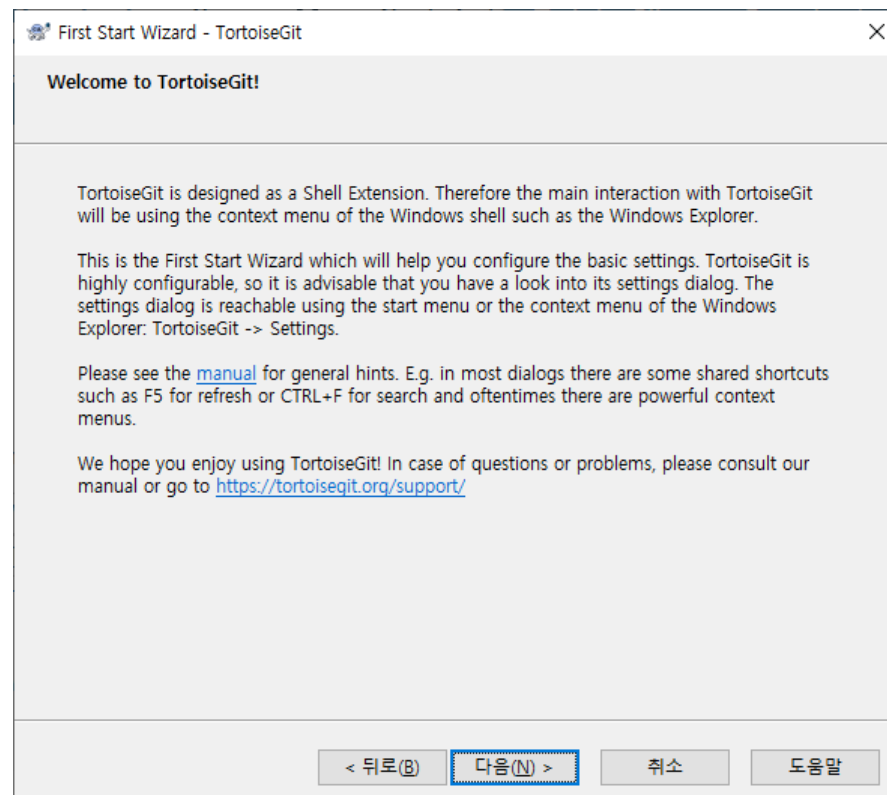
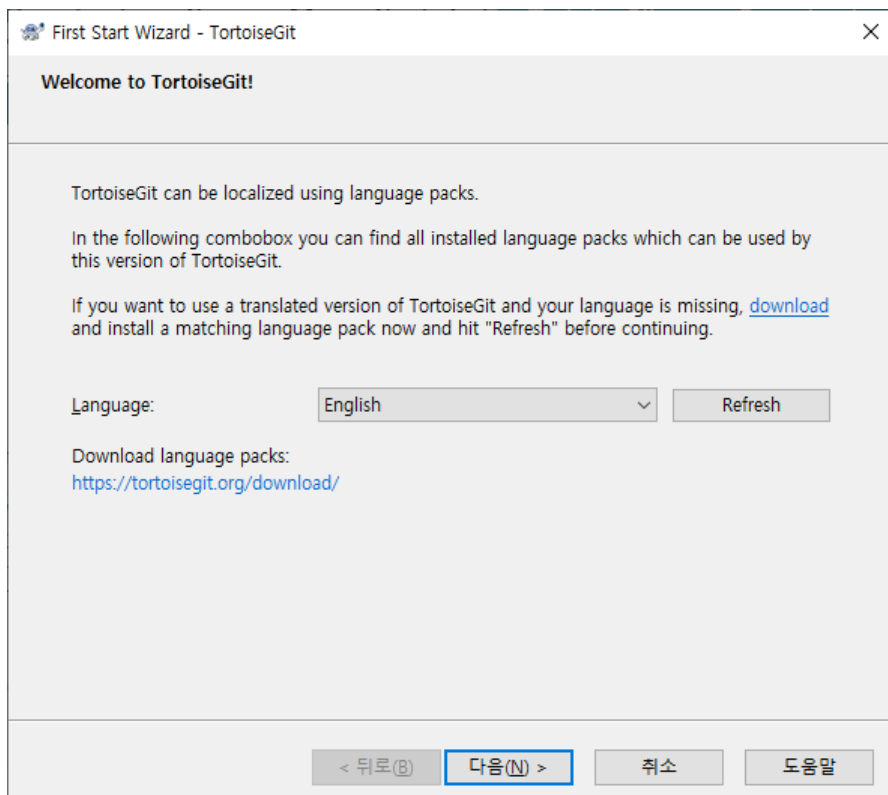


TortoiseGit 설정 - 1



• 최초 실행 마법사

- 언어 선택에 한국어를 기본으로 지원하지 않음
- 언어 선택 팩으로 설치도 가능





TortoiseGit 설정 - 2



• git 사용 환경 설정

- git 경로를 설정. 설치되어 있다면 경로가 바로 출력
- 설치되지 않은 경우는 직접 입력해야 함
- 사용자 정보를 입력. "git config --global"로 입력되는 정보 설정

First Start Wizard - TortoiseGit

Configure git.exe

TortoiseGit requires a git.exe for its operations. TortoiseGit tries to automatically detect a working git.exe, but if that doesn't work or you want to use a different one please specify the path manually!

Git.exe Path: ...

Extra PATH:

git version 2.33.0.windows.1

Recommended: Git for Windows
<https://git-for-windows.github.io/>

< 뒤로(B) 다음(N) > 취소 도움말

First Start Wizard - TortoiseGit

Configure user information

Git requires that you set up a user name and email address. Both are used as meta data for your commits (not for authentication).

Name:

Email:

These settings will be stored to your global git configuration (%HOME%/.gitconfig) and will be used for all your git repositories as a default.

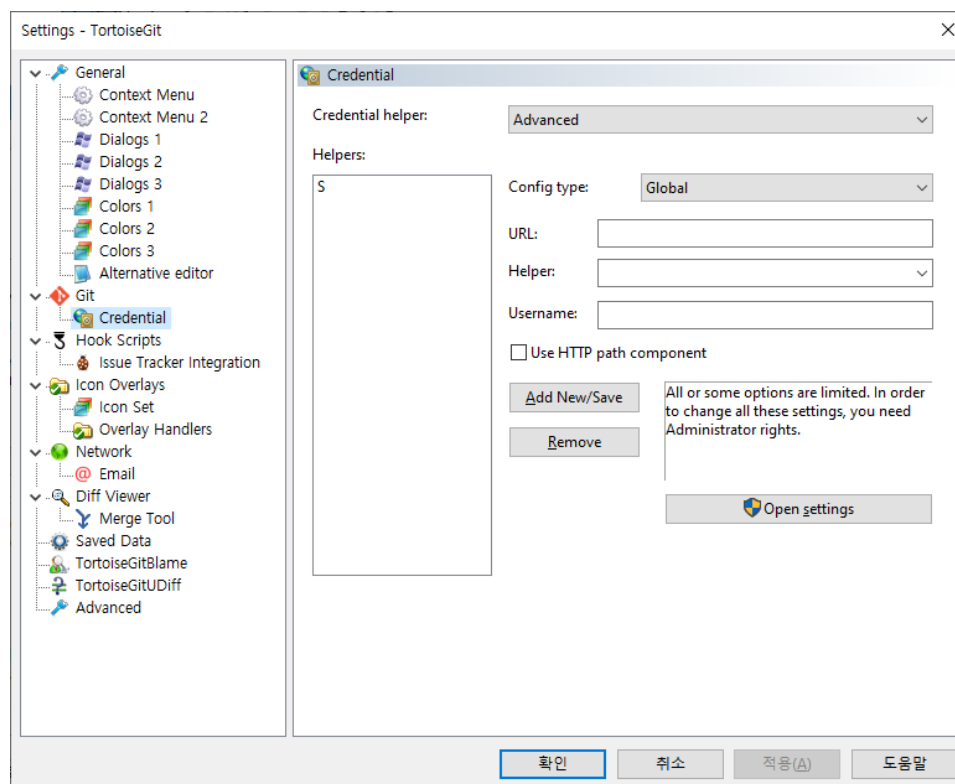
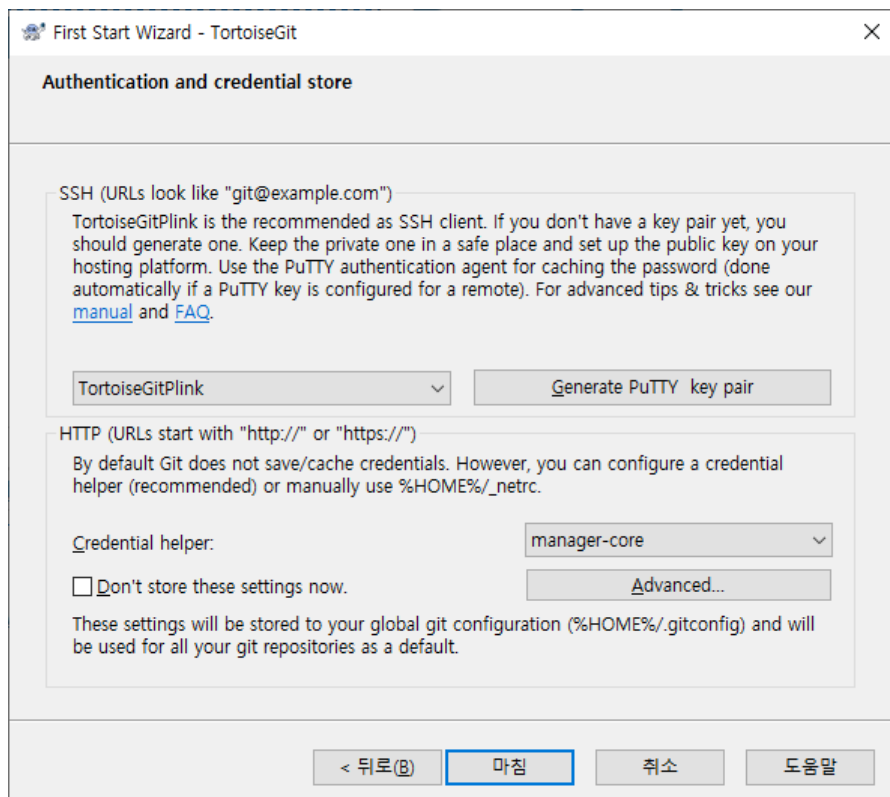
☐ Don't store these settings now.

< 뒤로(B) 다음(N) > 취소 도움말



TortoiseGit 설정 - 3

- SSH 설정
- TortoiseGit 세부 설정
 - CUI 설정을 GUI로 입력할 수 있도록 간단하게 지원

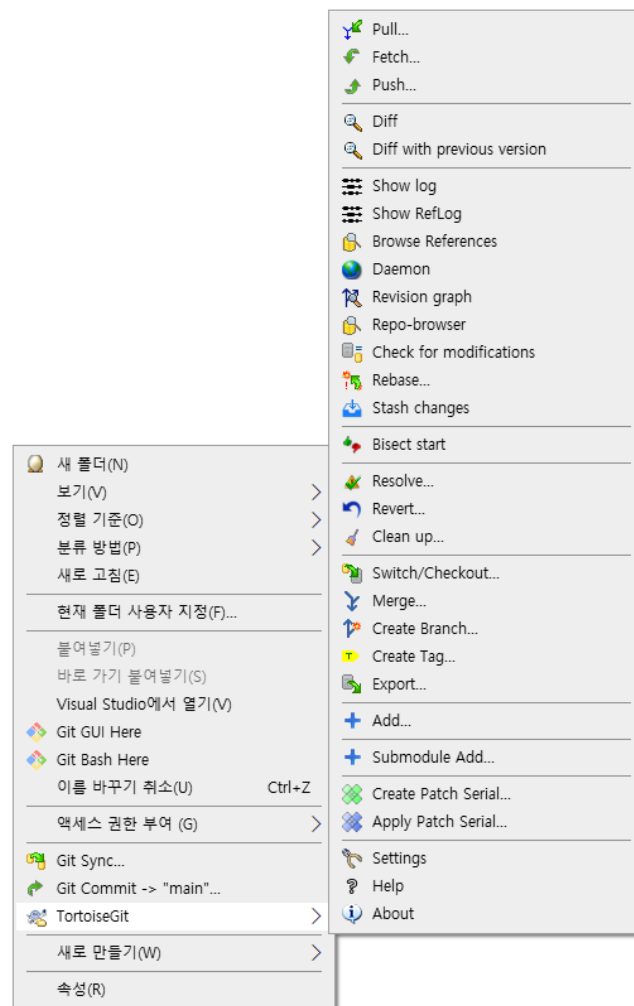
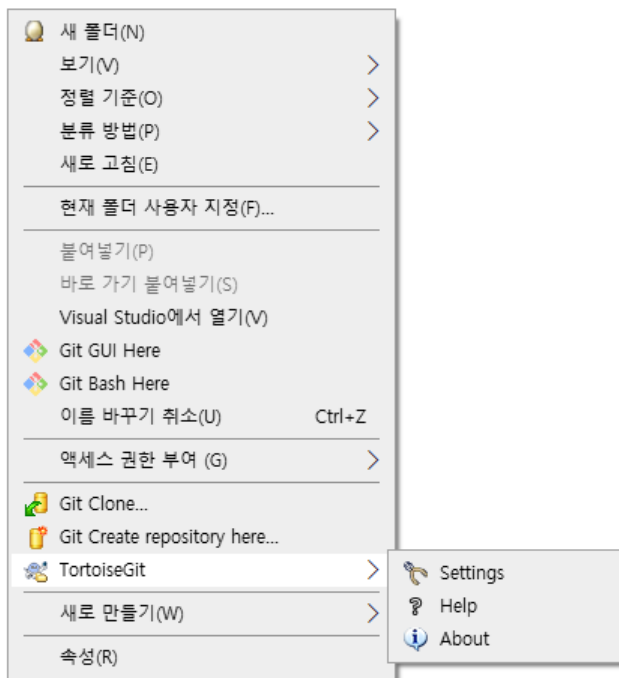




TortoiseGit 설정 - 4

• 탐색기에 추가 메뉴 등록 확인

- 탐색기와 결합되어 사용이 간편
- 저장소와 저장소가 아닌 경우 메뉴 확인
- 저장소에서 사용할 수 있는 기능 확인

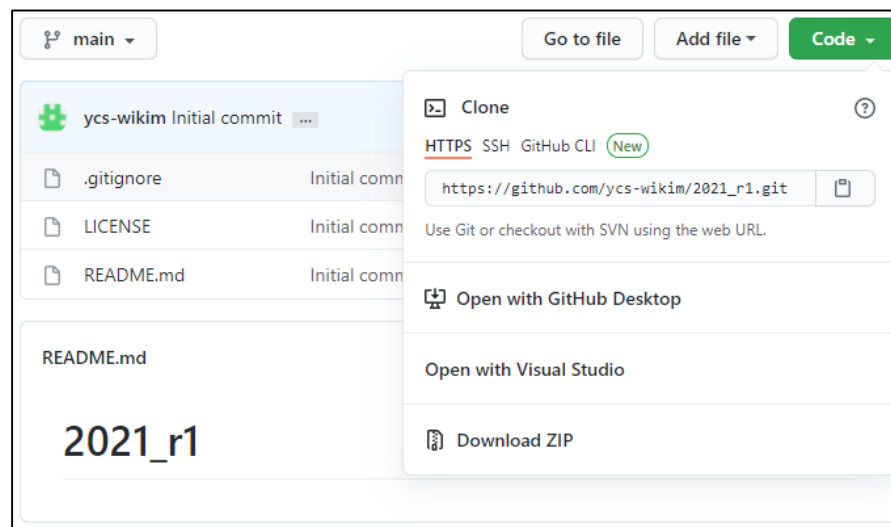
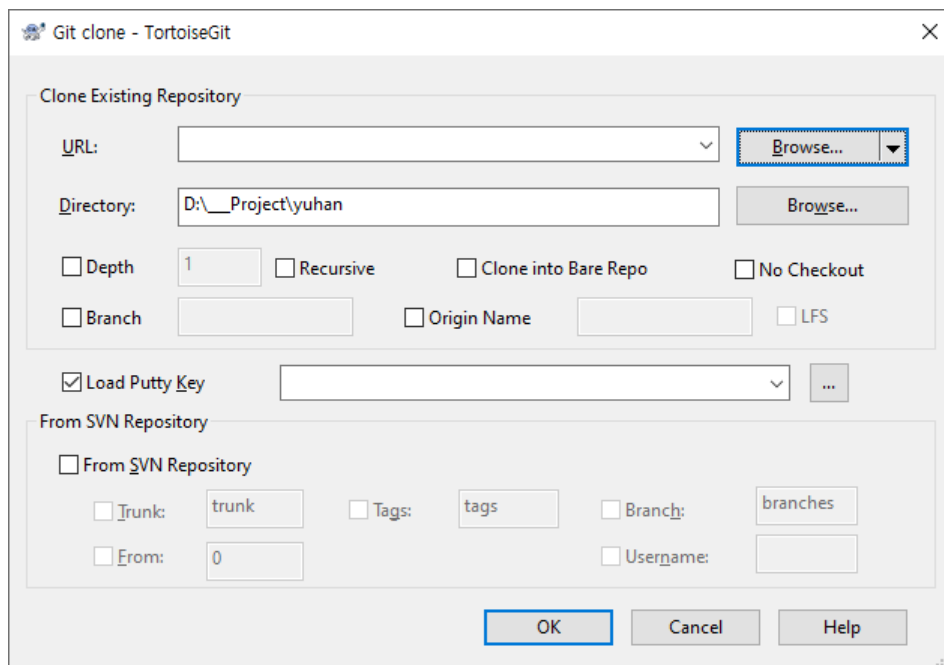




TortoiseGit 사용 - 1

• 저장소 복제 - 1

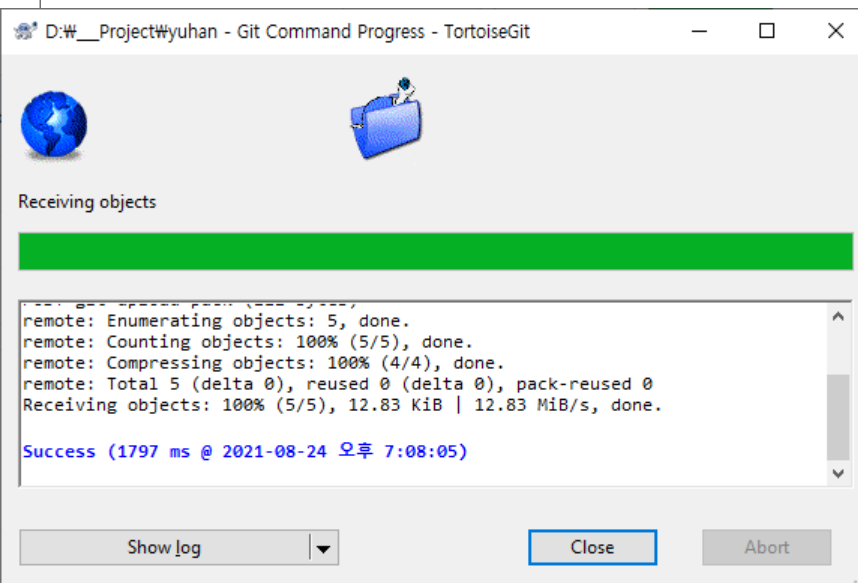
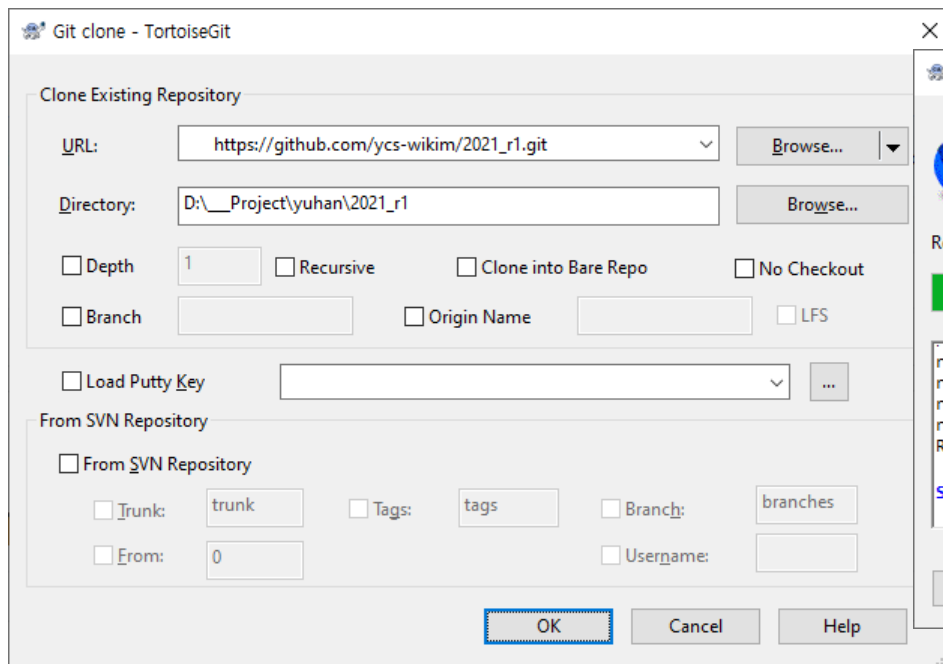
- URL 입력 등을 GUI로 간단히 선택하여 사용 가능
- github.com 저장소 메인 화면에서 필요한 주소 형식을 제공
- 필요에 따라 선택하여 사용





• 저장소 복제 - 2

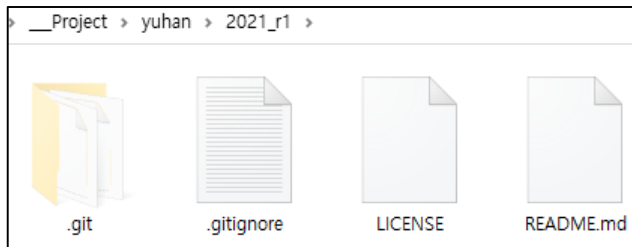
- 지정된 위치로 저장소를 복제
- 출력되는 정보도 Git Bash의 메시지와 동일하게 확인 가능
- 해당 화면에서 필요한 추가 작업을 항상 표시



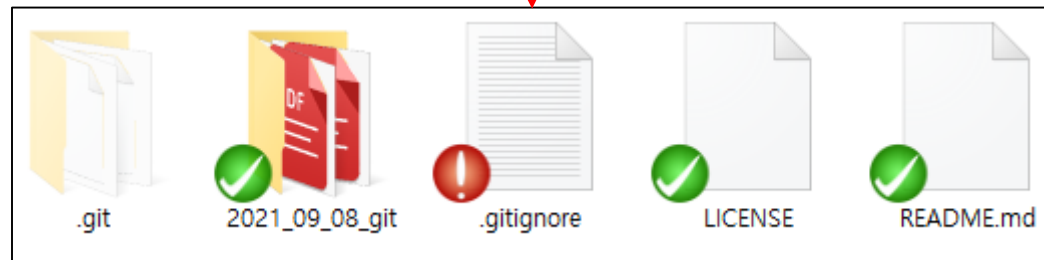
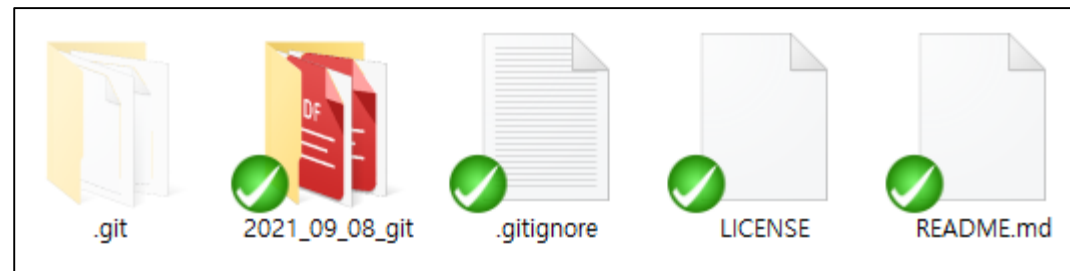


• 아이콘 오버레이 적용

- TortoiseGit 설치 시, 탐색기에서 직관적으로 저장소 여부 확인 가능
- 파일 변화를 아이콘으로 표시하여 직관적으로 확인
- commit 파일과 수정된 파일에 대해 바로 확인



Vs.

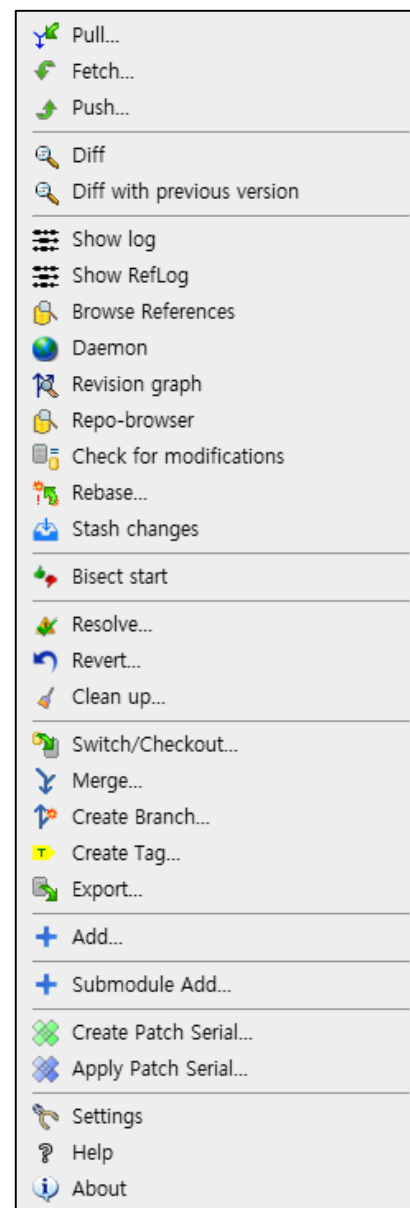




TortoiseGit 사용 - 3

• GUI 메뉴의 활용

- 별도의 프로그램 동작 없이 사용 가능
- 저장소에서 수행할 수 있는 기능을 모두 표시
- 모두 GUI로 표시하기 때문에 사용성이 증가



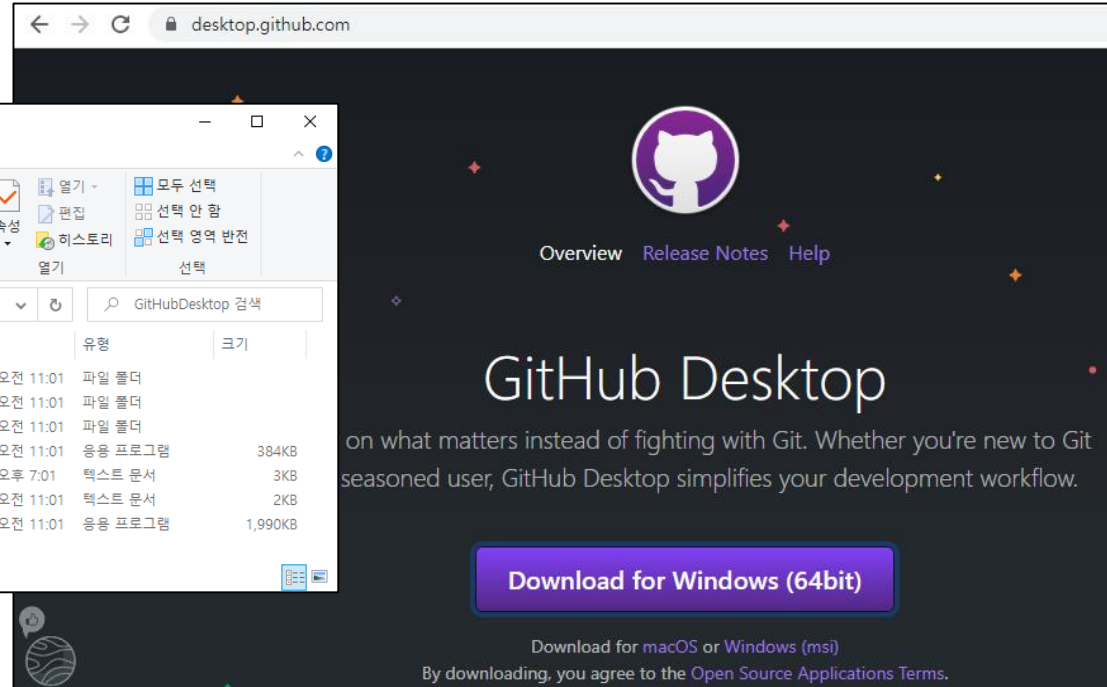
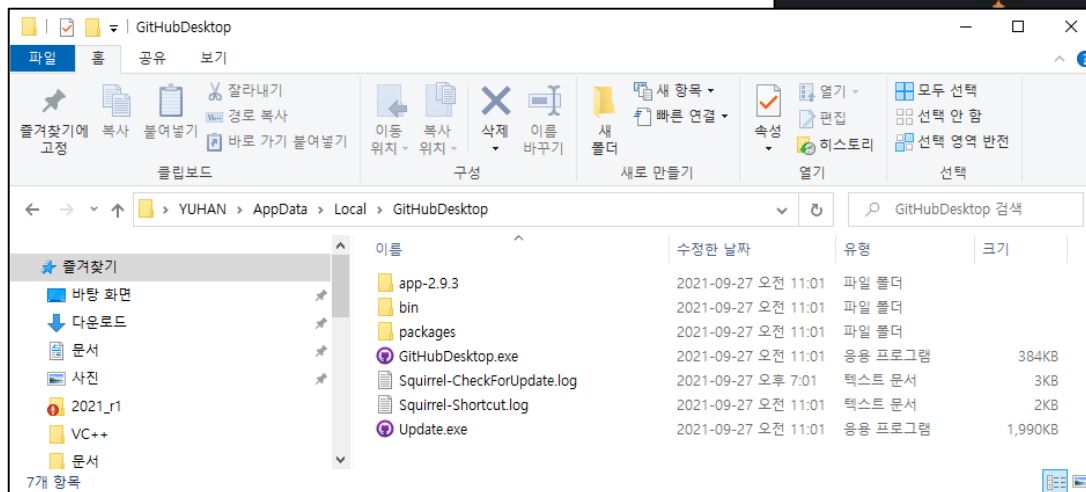
All things are difficult, before they are easy.



GitHub Desktop

- github 제공 GUI 프로그램

- github.com 연동과 관련된 git 기능 사용을 위해 제공
- 즉시 연동되는 GUI를 통해 간단히 사용 가능
- 일반적인 github.com 관련 기능들을 쉽게 사용할 수 있음
- 일반 프로그램 설치 위치와 다른 곳에 설치되므로 주의



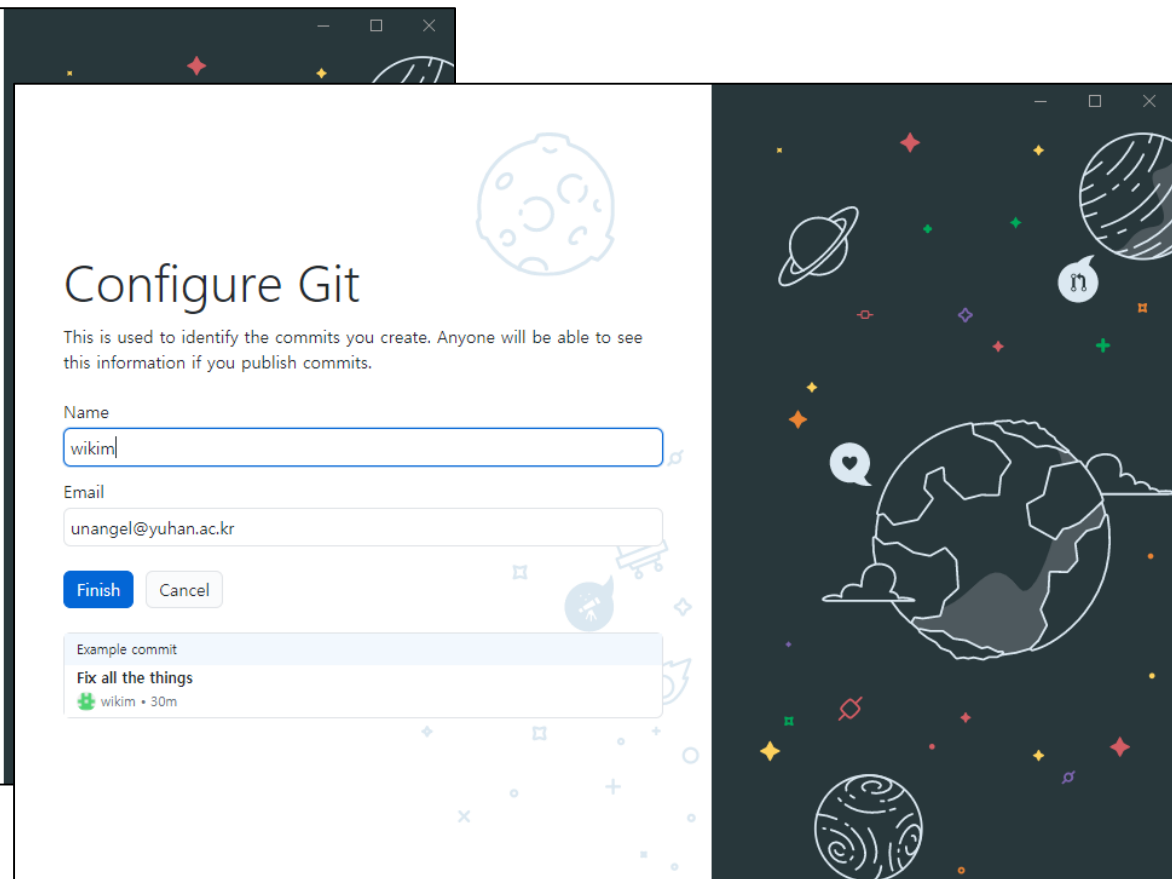
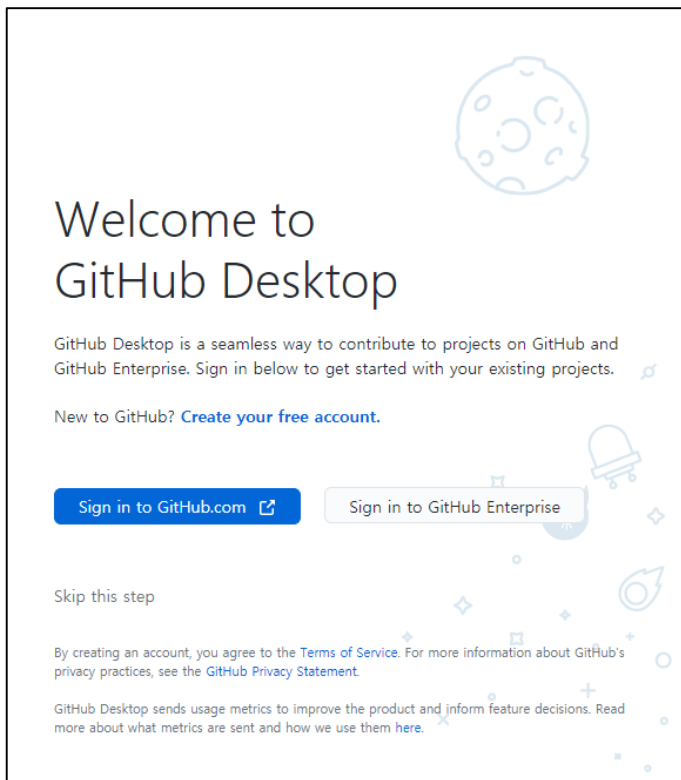


GitHub Desktop 설치, 설정 - 1



- 매우 간단한 설치

- 간단한 설치 후 즉시 사용 가능
- github.com 연동만 가능하므로, 필수적으로 인증을 받아야 함

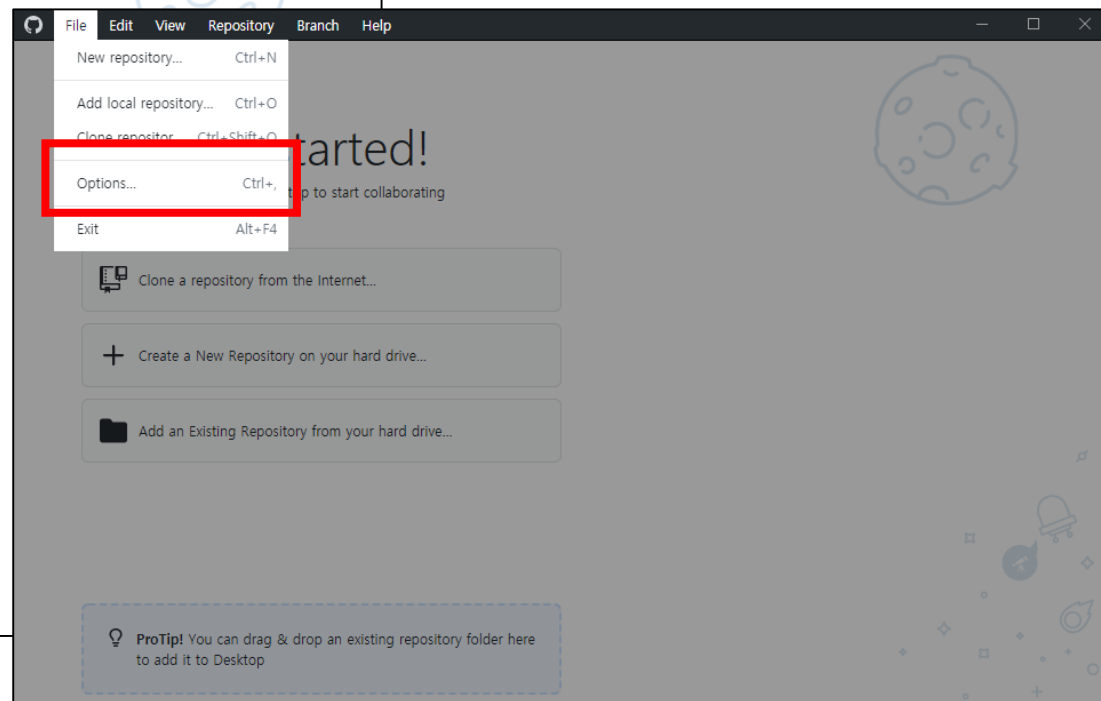
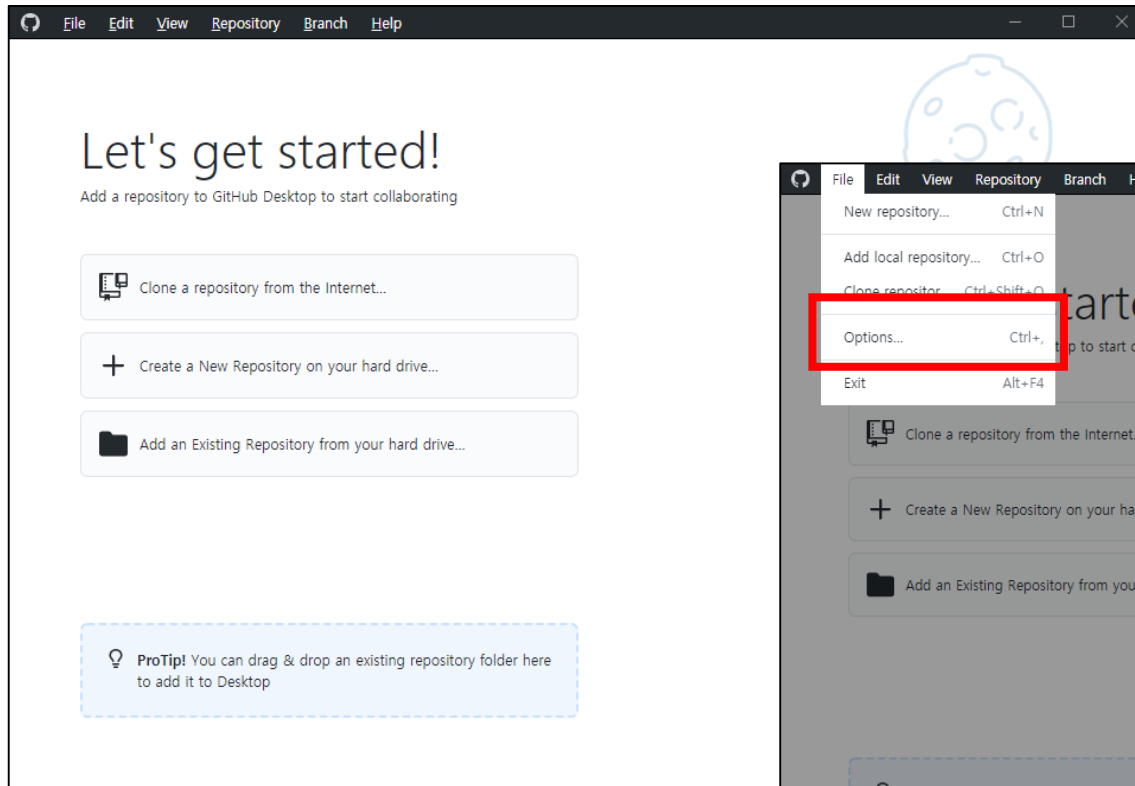




GitHub Desktop 설치, 설정 - 2

- 인증 없이 실행한 경우

- 기본적으로 git을 사용할 수 있는 환경을 제공
- File → Options 실행하여 github.com 인증을 수행해야 함

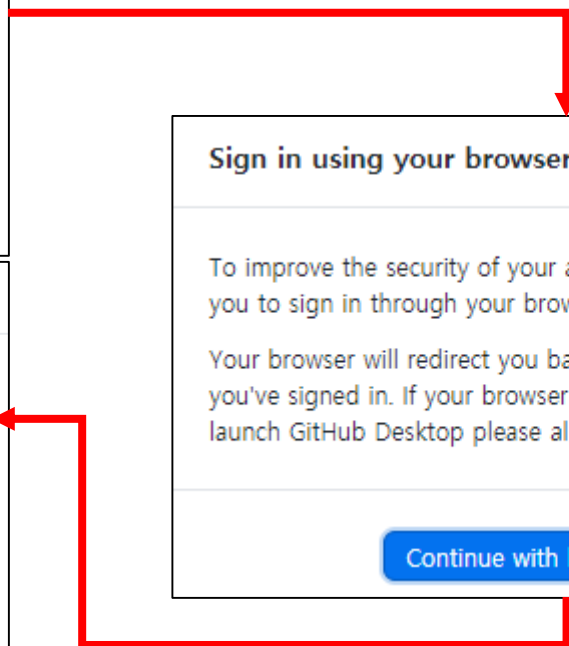
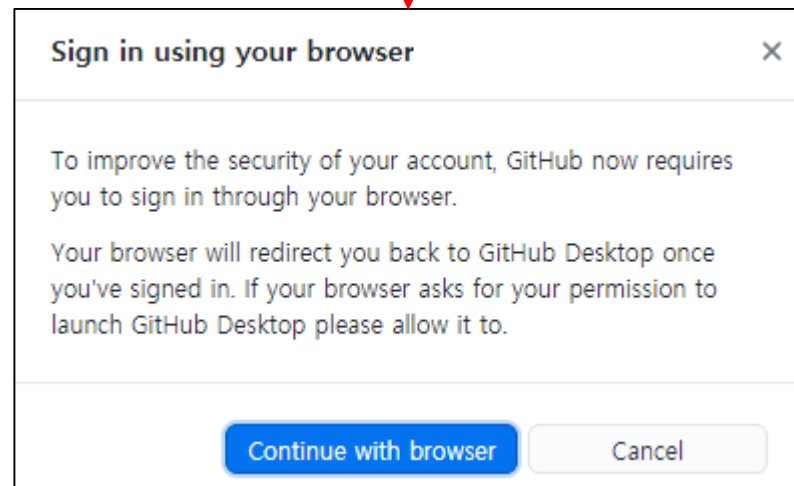
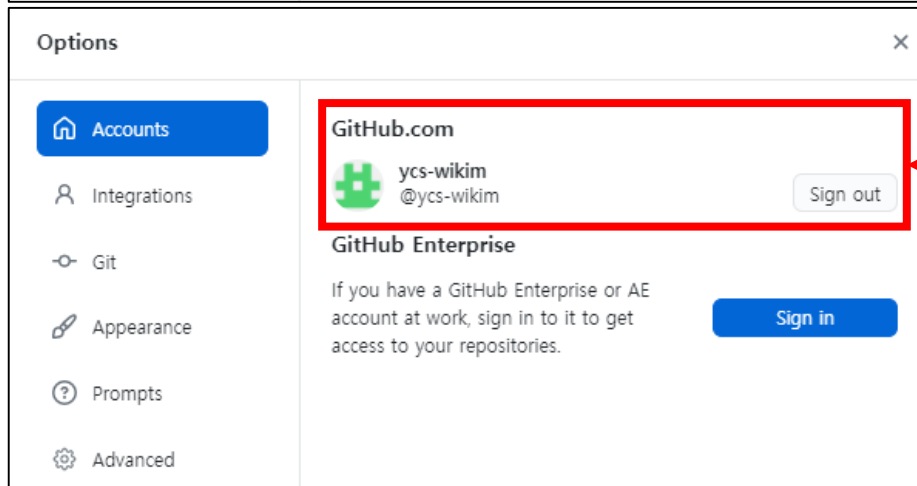
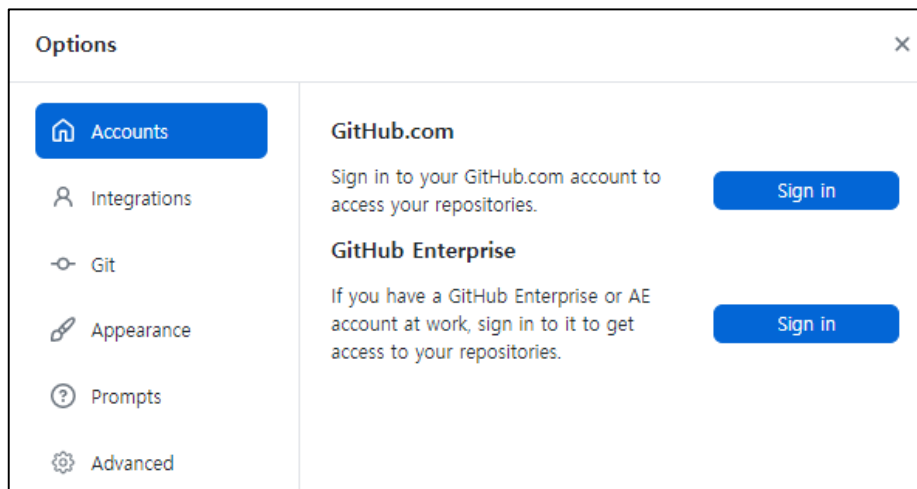




GitHub Desktop 설치, 설정 - 3



- GitHub.com 인증 전후의 계정 정보





GitHub Desktop 설치, 설정 - 4

- github.com 인증
 - 웹 브라우저를 통해 인증을 수행
 - 인증 후, 프로그램을 실행해야만 정상적으로 인증이 완료

Sign in to GitHub
to continue to GitHub Desktop

Username or email address

unangel@yuhan.ac.kr

Password

Forgot password

Sign in

GitHubDesktop.exe을(를) 여시겠습니까?

<https://github.com>에서 이 애플리케이션을 열려고 합니다.

☐ 항상 github.com에서 연결된 앱에 있는 이 유형의 링크를 열도록 허용

GitHubDesktop.exe 열기 취소

Authorize GitHub Desktop

GitHub Desktop by desktop wants to access your ycs-wikim account

Repositories
Public and private

Personal user data
Full access

low
GitHub Action Workflow files.

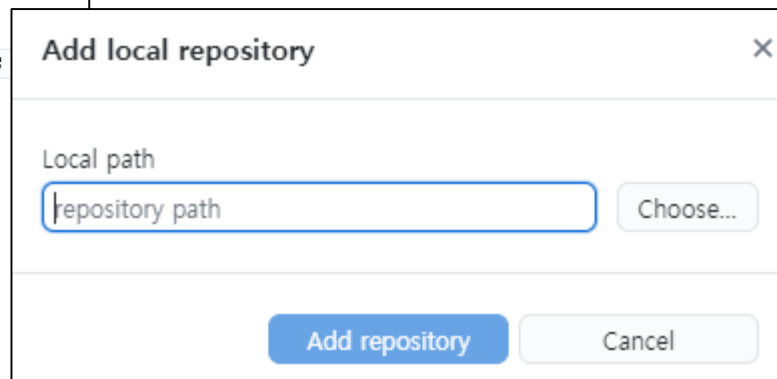
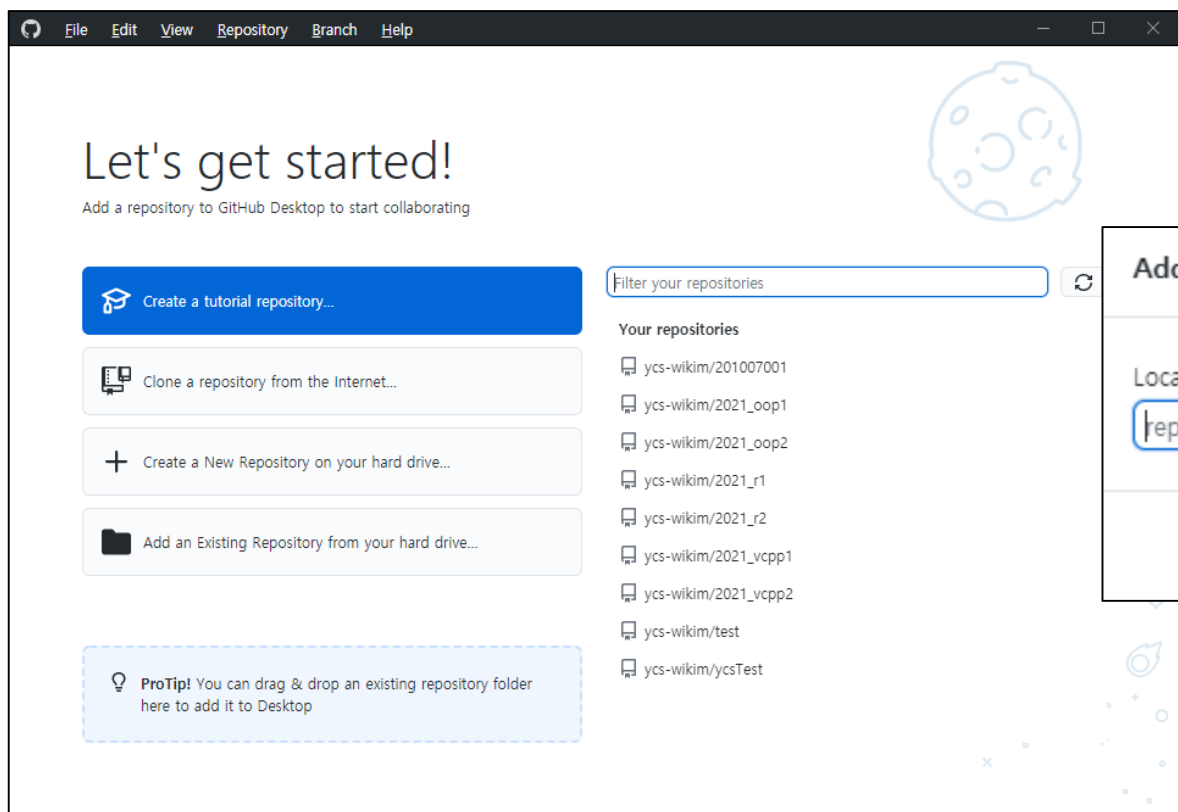
Cancel Authorize desktop

Authorizing will redirect to
x-github-desktop-auth://oauth



GitHub Desktop 설치, 설정 – 5

- github.com 계정 연동
 - 자신의 저장소 목록이 자동 다운로드되어 확인 가능
 - 로컬 저장소를 연결하여 바로 사용 또는 복제도 가능

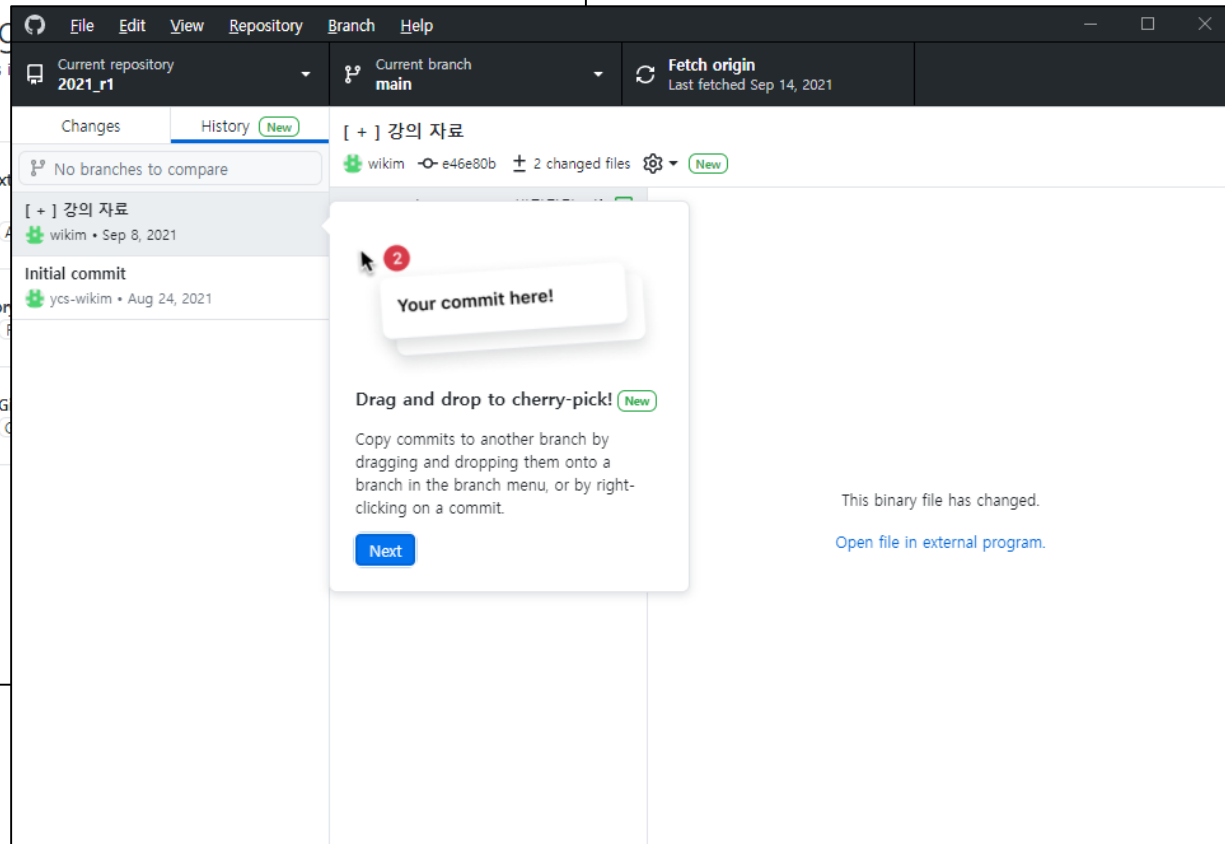
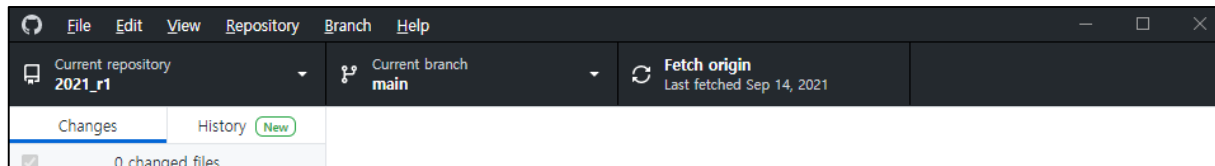




GitHub Desktop 설치, 설정 – 6



- 저장소 상태와 commit history 확인 가능



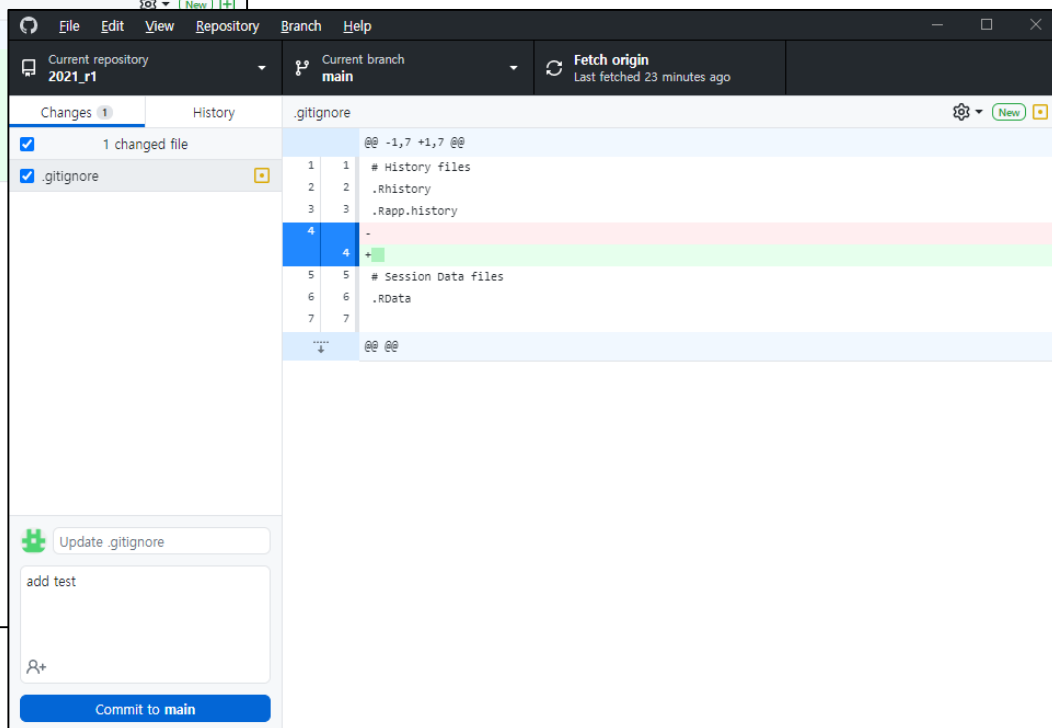
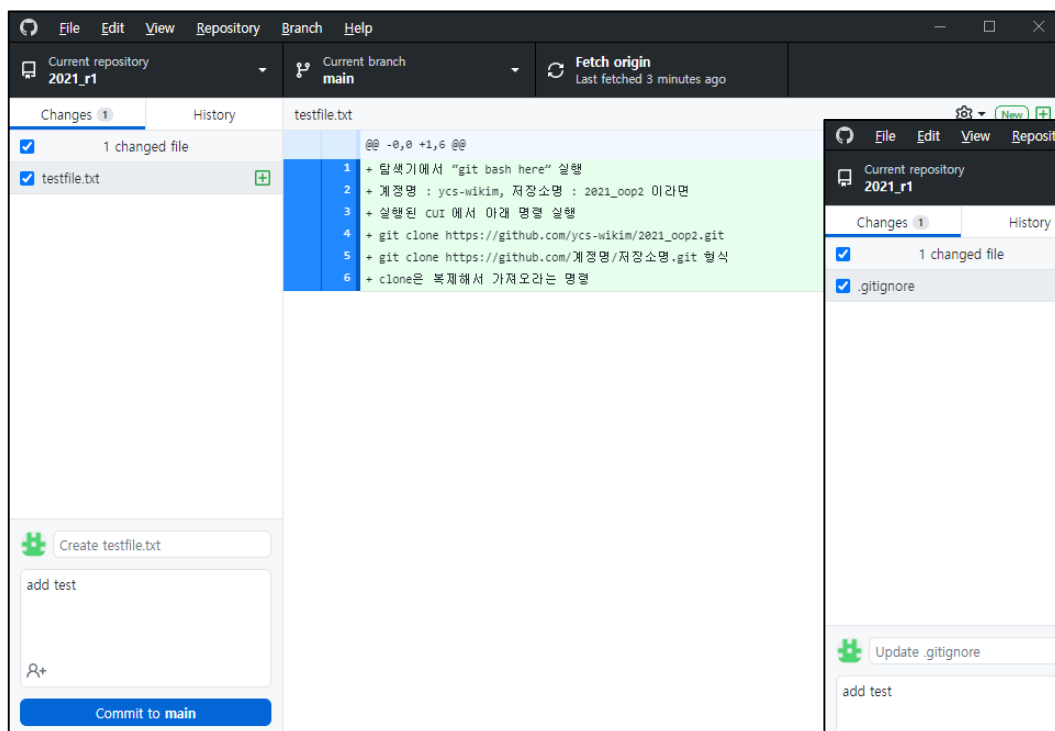


GitHub Desktop 설치, 설정 - 7



• 파일 추가/수정 확인 가능

- 추가 또는 수정된 파일과 내용 확인 가능
- 즉시 commit 가능하도록 인터페이스 지원
- 다수의 파일 수정과 선택을 통한 commit 등도 지원



All things are difficult, before they are easy.



TortoiseGit Vs. GitHubDesktop

부록 - git 협업



About..

컴퓨터소프트웨어공학과
김 원 일



목차



- 실습 조 구성
- 브랜치 생성
- 브랜치 병합
- 병합 내용 확인



• 실습 조 구성

- 조는 3인으로 구성
 - 조장 1명(C), 조원 2명(A, B)
- 조장이 github.com에 저장소를 생성
 - 저장소 명 : ycsTest
 - README.md와 .gitignore 파일을 추가하여 생성
- 생성한 저장소를 로컬로 복사
 - 모든 조원이 다음 명령으로 각각 저장소를 다운로드
git clone <https://github.com/조장ID/ycsTest.git>



- **협업할 파일들을 생성**

- 저장소로 이동
- 저장소에 협업할 파일들을 생성
- a.txt(조장의 파일), b.txt(조원 A의 파일), c.txt(조원 B의 파일) 생성
 - 각 파일에 해당 파일을 수정할 조원의 학번과 이름을 입력
- abc.txt 파일 생성
 - 각 파일에 해당 파일을 수정할 학생들의 학번과 이름을 입력
 - abc.txt 파일은 조장의 학번과 이름 그리고, 조원 A의 학번과 이름
 - bc.txt 파일은 조원 B의 학번과 이름 그리고, 조원 C의 학번과 이름
 - ca.txt 파일은 조원 C의 학번과 이름 그리고, 조원 A의 학번과 이름
- 생성한 파일들을 github로 업로드
 - git add . ; git commit -m "add files" ; git push



A, B, C 조원 수행



- **브랜치를 생성**

- git checkout -b 브랜치명
- 브랜치 이름은 본인 학번으로 생성
- 현재 브랜치가 생성한 브랜치인지 확인
git branch

- **저장소의 파일들을 수정**

- 본인에게 해당하는 파일들에 내용을 추가
- A 조원 : a.txt 만 수정
- B 조원 : b.txt 만 수정
- C 조원 : c.txt 만 수정
- 저장소를 commit 하여 github로 업로드
git add .
git commit -m "modify a.txt"
git push --set-upstream origin 브랜치명

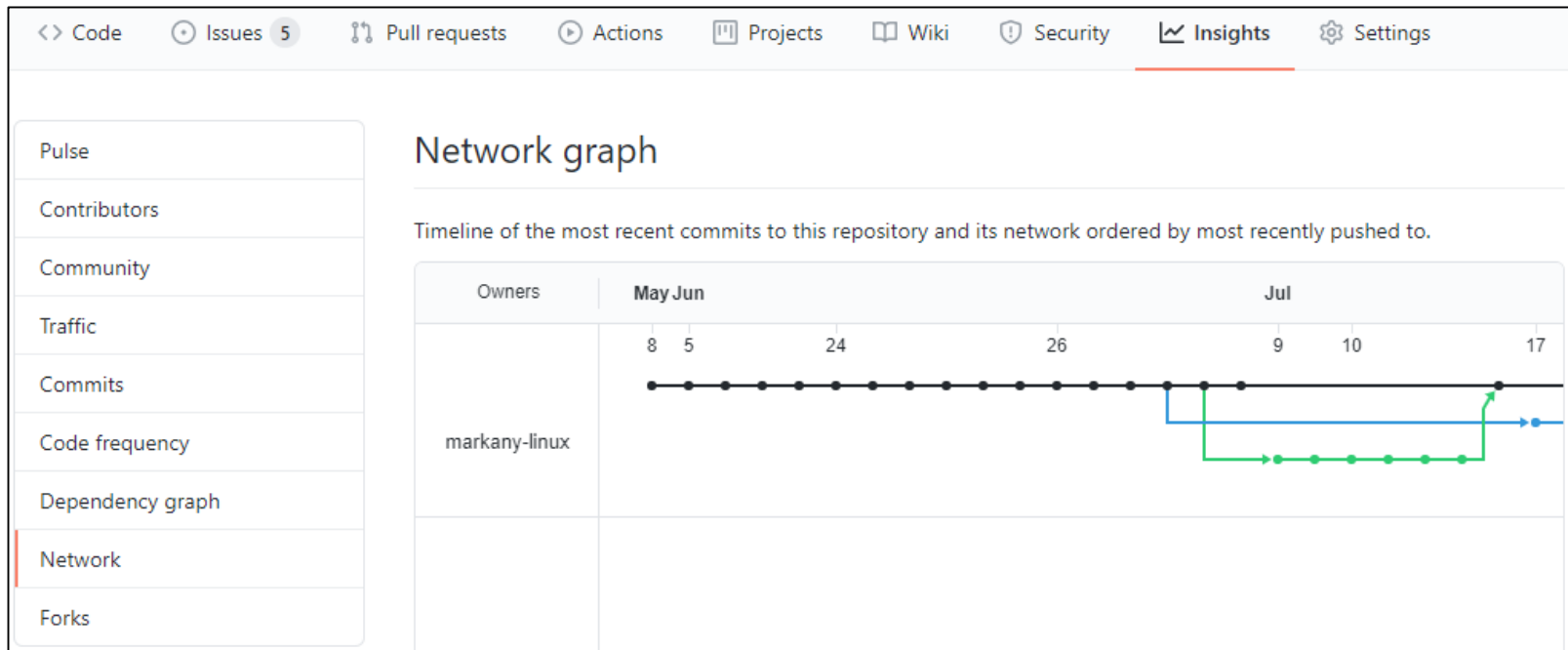


업로드 확인



- github 사이트 확인

- 상단 "Insights" 메뉴 선택
- 좌측 하단에서 "Network" 메뉴 선택
- 생성되어 업로드 된 브랜치 명을 확인





A, B, C 조원 수행



- **현재 브랜치에서 추가 작성 및 업로드**

- A, B, C 조원 모두 자신의 파일을 수정하여 업로드를 수회 실행
- 파일을 수정할 때 다음의 순서로 진행

파일을 수정

`git add .`

`git commit -m "a.txt 수정 내용"`

`/// github 확인`

파일을 수정

`git add .`

`git commit -m "a.txt 수정 내용"`

`/// github 확인`

파일을 수정

`git add .`

`git commit -m "a.txt 수정 내용"`

`/// github 확인`

... (다수 실행)

`git push`

`/// github 확인`

- 저장소 업로드 전까지 commit 된 내용은 로컬에서만 존재



C – 조장 수행



- C 조장은 자신의 브랜치를 병합하고 업로드

- main 저장소로 브랜치를 변경

git checkout main

- 본인의 저장소를 병합

git merge 본인브랜치명

- 병합된 파일명 확인

```
YUHAN@YP12624115 MINGW64 /d/_Development/github/testCode (main)
$ git merge test
Updating 4616214..1887947
Fast-forward
 test.c | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 test.c
```

- 현재 상태 확인

git status

- 자동으로 commit 추가 확인

- 병합 시 자동으로 발생

```
YUHAN@YP12624115 MINGW64 /d/_Development/github/testCode (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

- github로 업로드 : git push

```
YUHAN@YP12624115 MINGW64 /d/_Development/github/testCode (main)
```




B, C 조원 수행



- **업로드 된 저장소를 다운로드**

`git pull`

- **브랜치를 main 으로 변경하고 조장이 병합한 파일 내용을 확인**

`git checkout main`

- **다시 본인의 브랜치로 이동**

`git checkout 본인브랜치`

- **조장의 파일을 다시 확인**



C – 조장 수행



- **C 조장은 브랜치들을 병합하고 업로드**

- main 저장소로 브랜치를 변경
git checkout main
- 조원 A의 저장소를 병합
git merge 조원A브랜치명
- 현재 상태 확인
git status
- 조원 B의 저장소를 병합
git merge 조원B브랜치명
- github로 업로드 : git push



B, C 조원 수행



- **업로드 된 저장소를 다운로드**

`git pull`

- **브랜치를 main 으로 변경하고 조장이 병합한 파일 내용들을 확인**

`git checkout main`

- **다시 본인의 브랜치로 이동**

`git checkout 본인브랜치`

- **다른 파일들을 다시 확인**