

형상관리 도구

Version Control Revision Control Tool

- 소프트웨어 버전 관리 툴이라고도 한다.

- 소스의 저장용도 또는 하나의 프로젝트를 여러 명이 수정 하는 용도로 쓴다.

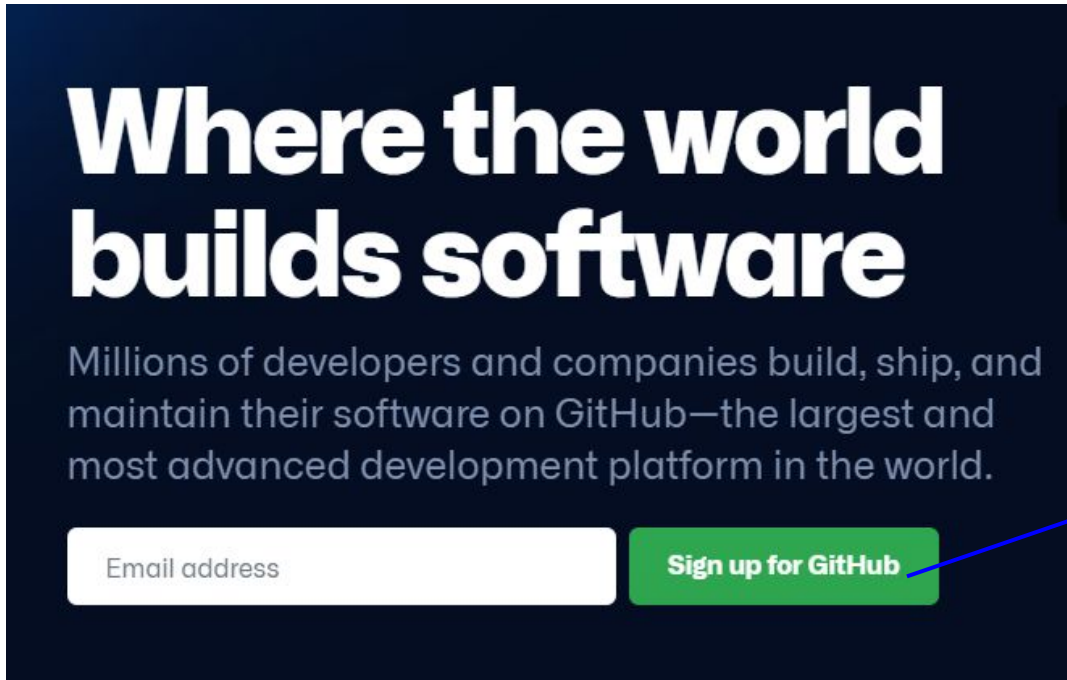
- 소스를 버전 별로 관리할 수 있어서 개발할 때 실수로 소스를 삭제하거나, 수정하기 이전으로 돌아가야되는 경우 유용하게 사용되는 툴.

- 또한 팀 프로젝트에서도 누가 무엇을 어떻게 수정했는지도 알 수 있기 때문에 코드를 병합하거나 수정된 소스를 추적하는 데에도 쓰인다.

형상관리 도구 **Git** 사용하기

1. GitHub 가입

<https://github.com/> 접속 또는 검색사이트에 깃허브 검색

A dark blue banner with white text. The main headline reads "Where the world builds software". Below it, a subtitle says "Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world." At the bottom, there is a white input field labeled "Email address" and a green button labeled "Sign up for GitHub".

Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

Sign up for GitHub

클릭하고 가입 진행

무료버전 선택

2.Git 설치

Git 다운로드 사이트 - Window

<https://git-scm.com/download/win>

프로그램 다운로드 후 설치 가이드(next만 누르면 됨)

<https://goddaehee.tistory.com/216?category=381481>

Git 우분투 설치 – windows는 해당 안됨

```
sudo apt-get install git
```

```
sudo apt install git
```

<https://coding-factory.tistory.com/502> 해당 사이트 확인

- Git Bash에서 깃 버전을 확인 해보자.

git --version



```
dedur@DESKTOP-9F838L4 MINGW64 /c
$ git --version
git version 2.25.1.windows.1
```

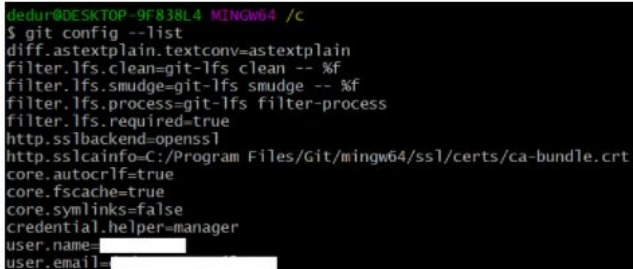
2.12 Git 사용자 등록 및 확인

1. 사용자 등록

```
git config --global user.name "name"
git config --global user.email "god@god.com"
```

2. 확인

```
git config --list
```



```
dedur@DESKTOP-9F838L4 MINGW64 /c
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
credential.helper=manager
user.name=[REDACTED]
user.email=[REDACTED]
```

<https://github.com/> 사이트에 접속하고 로그인을 한다.

new 또는 Create new repository를 눌러 새로운 repository를 생성한다.

The screenshot shows the GitHub homepage. At the top, there's a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Codespaces, Marketplace, and Explore. On the left sidebar, under 'Top Repositories', there's a 'New' button highlighted with a red box. Below it, a list of repositories is shown. Under 'Recent activity', there's a message about links to activity. The main content area has the heading 'The home for all developers — including you.' and a welcome message. Below this, there's a 'Start writing code' section with a 'Start a new repository' card. This card has a text input for the repository name, radio buttons for 'Public' and 'Private' (with 'Private' selected), and a 'Create a new repository' button highlighted with a red box. To the right, there's a 'Introduce yourself with a profile README' card with a 'Create' button.

Search or jump to...

Pull requests Issues Codespaces Marketplace Explore

Top Repositories

Find a repository...

New

RyuHongGeol/TeamProject

RyuHongGeol/gitTest

RyuHongGeol/ANU_SF

RyuHongGeol/TcpExample

RyuHongGeol/pythonProject8

RyuHongGeol/GitTestProject

RyuHongGeol/gitTest3

Show more

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

The home for all developers — including you.

Welcome to your personal dashboard, where you can find an introduction to how GitHub works, tools to help you build software, and help merging your first lines of code.

<> Start writing code

Start a new repository

A repository contains all of your project's files, revision history, and collaborator discussion.

RyuHongGeol /

☐ Public
Anyone on the internet can see this repository

☒ Private
You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

RyuHongGeol / README.md


```
1 - 👋 Hi, I'm @RyuHongGeol
2 - 🤖 I'm interested in ...
3 - 📖 I'm currently learning ...
4 - 🍷 I'm looking to collaborate on ...
5 - 📧 How to reach me ...
6
```


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *

 RyuHongGeol ▾

Repository name *

Great repository names are short and memorable. Need inspiration? How about [literate-waffle?](#)

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:

Skip this step if you're importing an existing repository.

☐ Add a README file

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

Add .gitignore

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

.gitignore template: None ▾

Choose a license

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

License: None ▾

 You are creating a public repository in your personal account.

Create repository

repository name을 작성하고

권한은 public으로 한다.

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

[Set up in Desktop](#)

or

[HTTPS](#)[SSH](#)<https://github.com/RyuHongGeol/gitTest3.git>

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 "# gitTest3" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/RyuHongGeol/gitTest3.git
git push -u origin main
```



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

```
git remote add origin https://github.com/RyuHongGeol/gitTest3.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)**복사!**

명령 프롬프트

cd \ 리눅스는 cd ~
cd 폴더 경로

git clone repository 경로

cd 생성된폴더

해당 폴더에서 메모장을 열어 새로운
document.txt 라는 파일을 만든다.

git add document.txt

git commit -m "create new file"

git push

c드라이브에 새폴더 생성하고 해당
폴더에서 명령 프롬프트 실행

경로는 github에서 복사해 온다.
repository가 복사되어 폴더가 생성된다.

.git이라는 숨김폴더가 있는 것을 확인한다.
.git은 메타데이터를 저장하는 폴더

staging area에 반영

Local Repository에 반영

github인 remote Repository에 반영
github에 해당 Repository에
반영되었는지 확인한다.

실행 예제 화면

```
C:\Users\AioT120>cd ₩
```

```
C:\₩>cd education
```

```
C:\₩education>git clone https://github.com/RyuHongGeol/gitTest3.git
Cloning into 'gitTest3'...
warning: You appear to have cloned an empty repository.
```

```
C:\₩education>cd gitTest3
```

```
C:\₩education\₩gitTest3>git add document.txt
```

```
C:\₩education\₩gitTest3>git commit -m "create new file"
[main (root-commit) 19b78fc] create new file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 document.txt
```

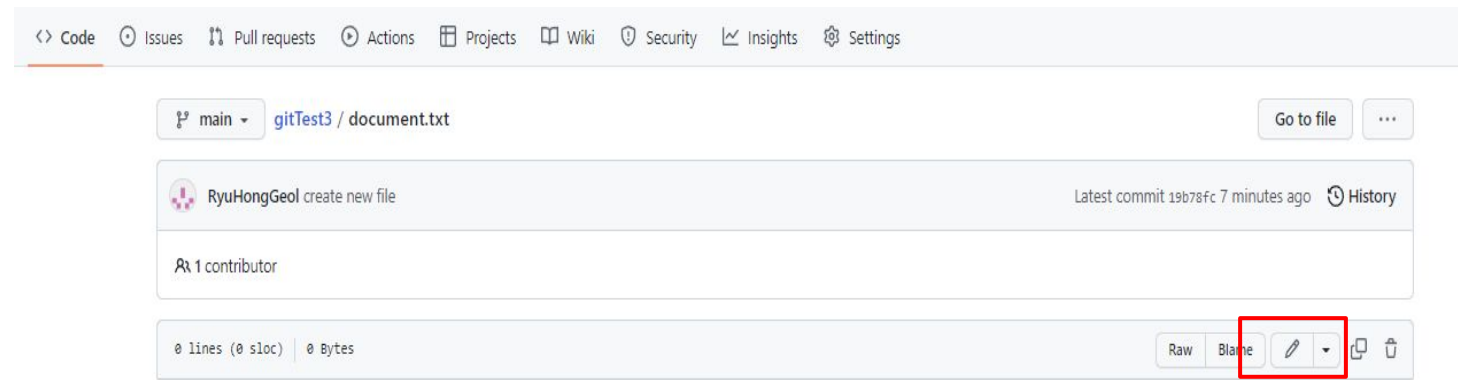
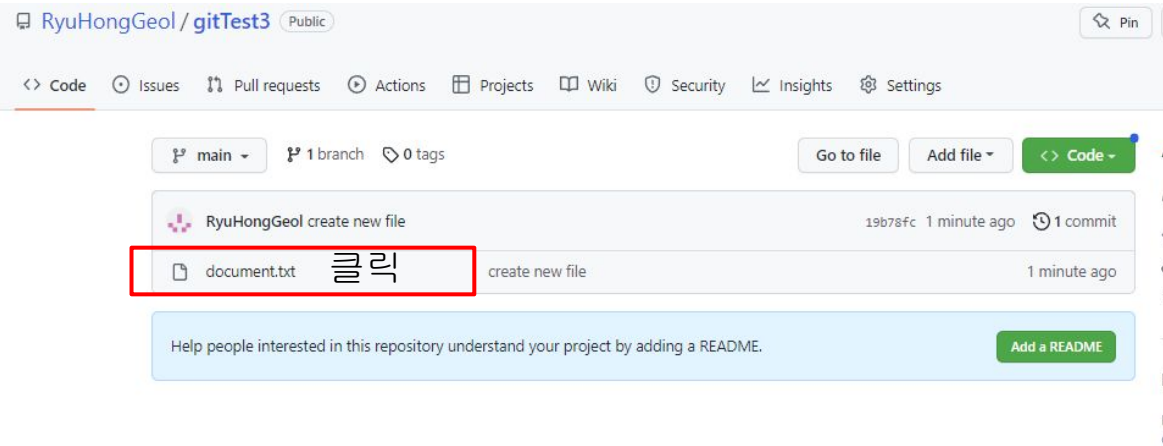
```
C:\₩education\₩gitTest3>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 220 bytes | 220.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/RyuHongGeol/gitTest3.git
* [new branch]      main -> main
```

참고

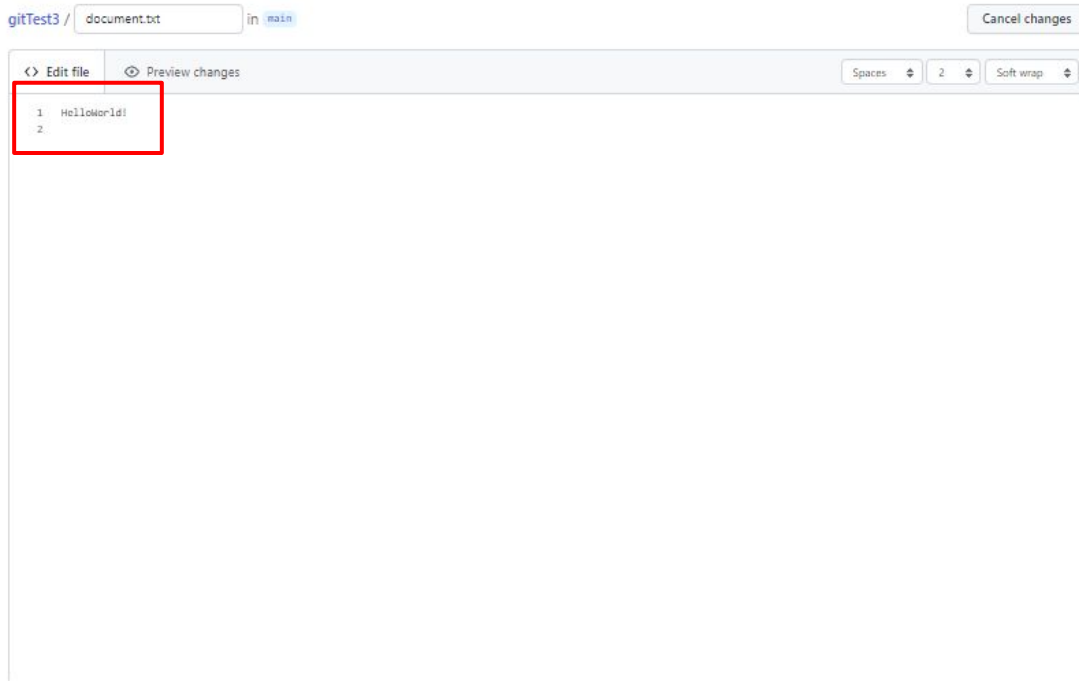
git status

명령어를 사용하면 상태를 확인할 수 있다.

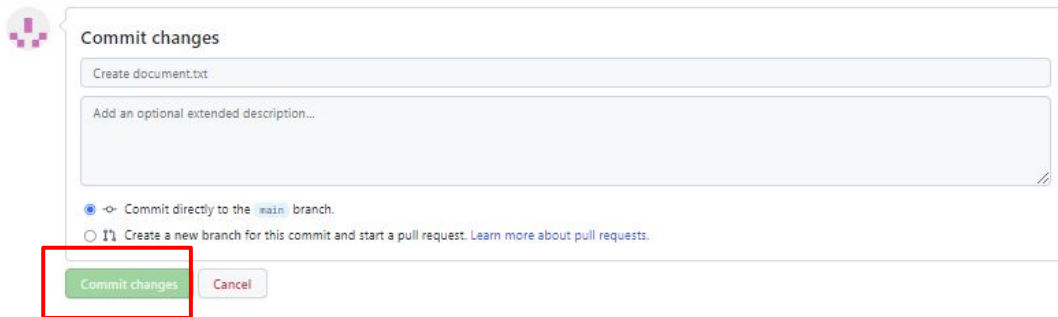
github에 반영되었다.



수정 표시 눌러서
수정해보자



수정하고 아래 **Commit** 버튼
눌러서 반영



```
Microsoft Windows [Version 10.0.19042.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AioT120>cd ₩

C:\₩>cd education

C:\₩education>cd gitTest3

C:\₩education\₩gitTest3>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 643 bytes | 42.00 KiB/s, done.
From https://github.com/RyuHongGeol/gitTest3
   19b78fc..70f709a  main       -> origin/main
Updating 19b78fc..70f709a
Fast-forward
 document.txt | 1 +
 1 file changed, 1 insertion(+)

C:\₩education\₩gitTest3>
```

명령 프롬프트

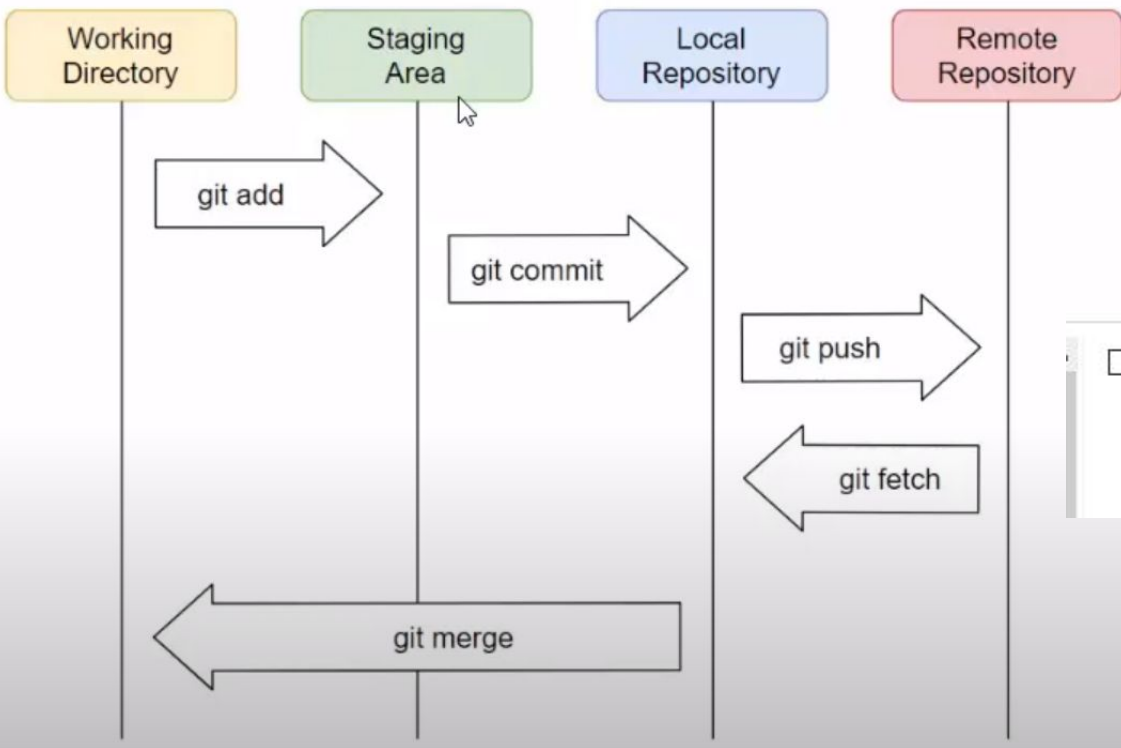
git pull

->github에서 수정된 내용이
LocalRepository에 반영된다.
pull은 fetch, merge 명령어를 합친 기능

작업한 파일

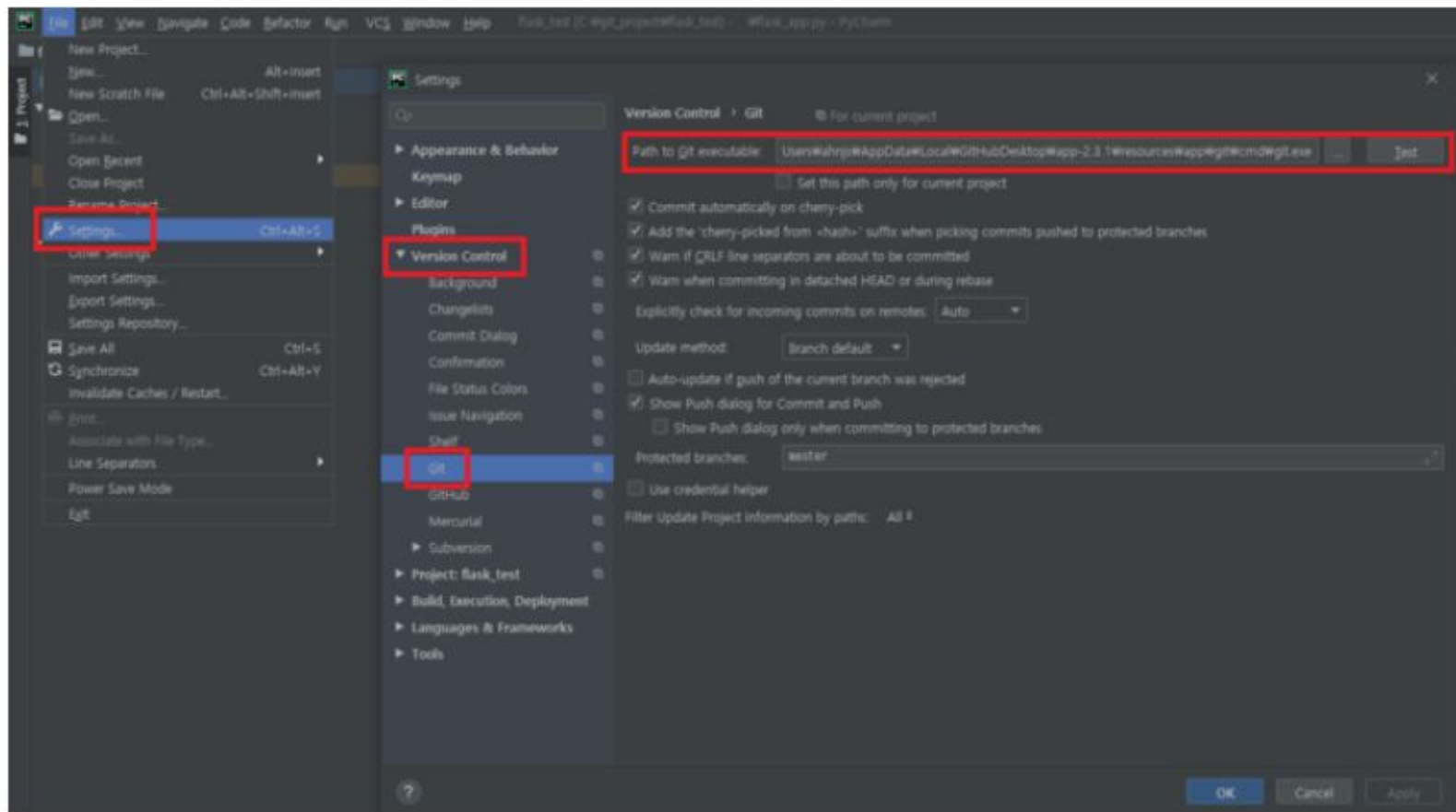
커밋을 수행할
파일이 있는 영역

Git 프로젝트의
메타 데이터와
데이터 정보 저장

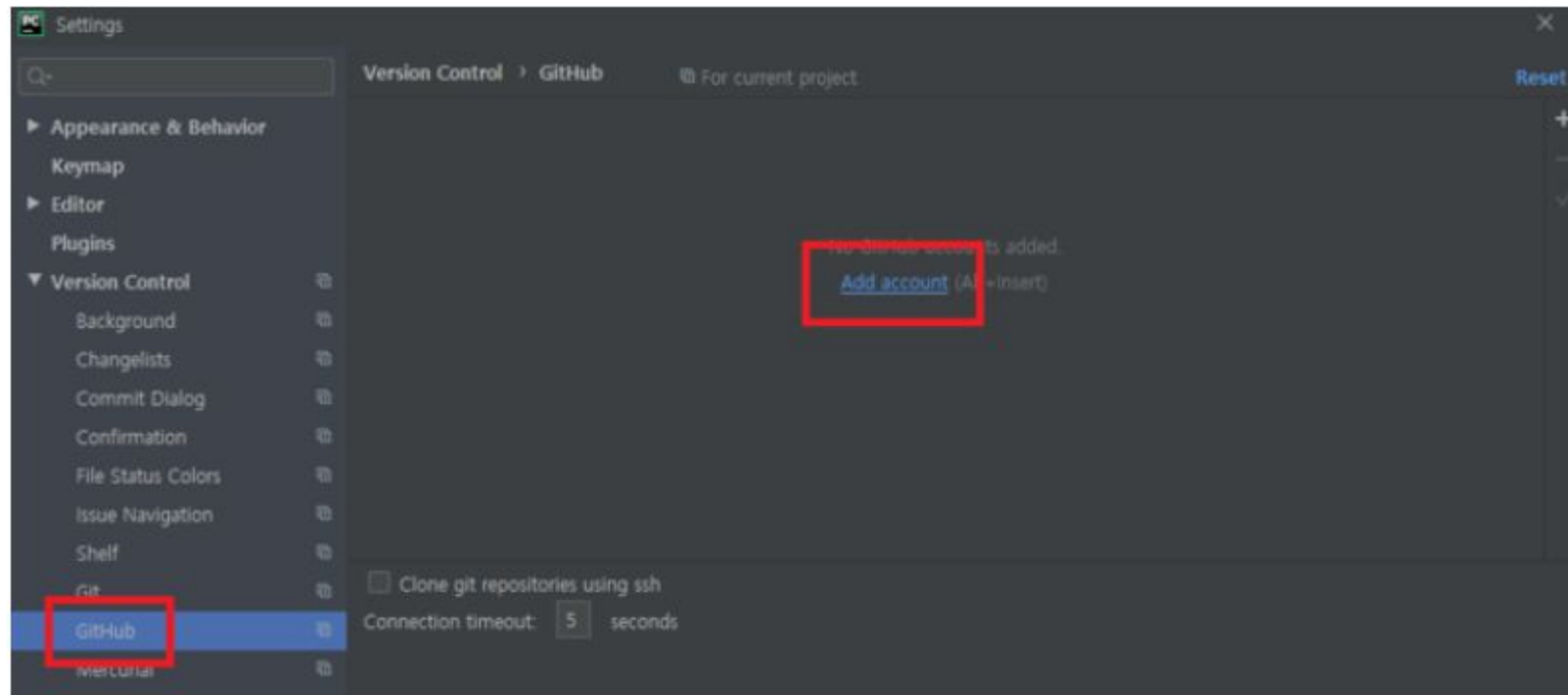


이름	수정한 날짜
.git	2022-06-08 오후 1:00
document.txt	2022-06-08 오후 1:00

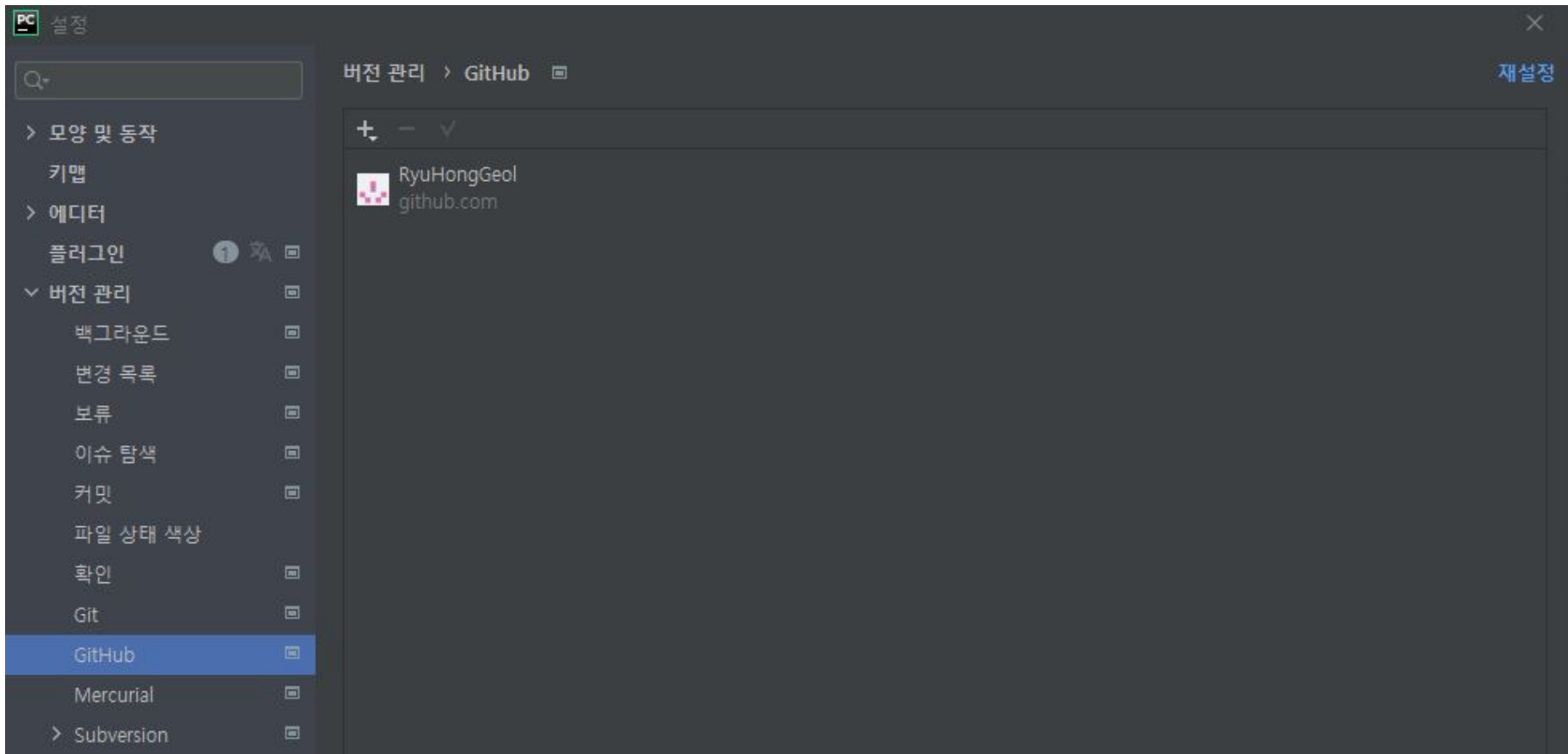
3. GitHub에 내 프로젝트 올리기 - Pycharm



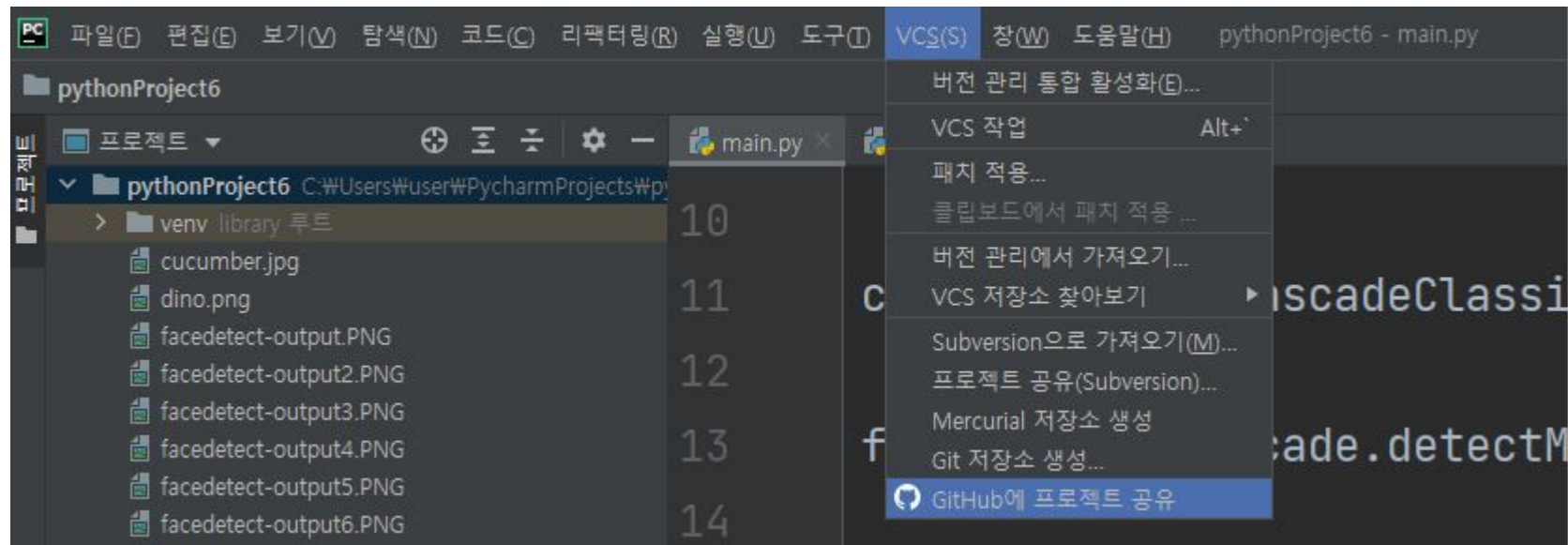
3. GitHub에 내 프로젝트 올리기 - Pycharm



3. GitHub에 내 프로젝트 올리기 - Pycharm



3. GitHub에 내 프로젝트 올리기 - Pycharm



3.GitHub에 내 프로젝트 올리기 - Pycharm

RyuHongGeol / pythonProject6 Public

Unwatch 1

Star 0

Fork

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights


Settings

master 1 branch 0 tags

Go to file

Add file

Code

 RyuHongGeol Initial commit e9e9893 21 minutes ago 1 commit

main.py	Initial commit	21 minutes ago
mosaic.py	Initial commit	21 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

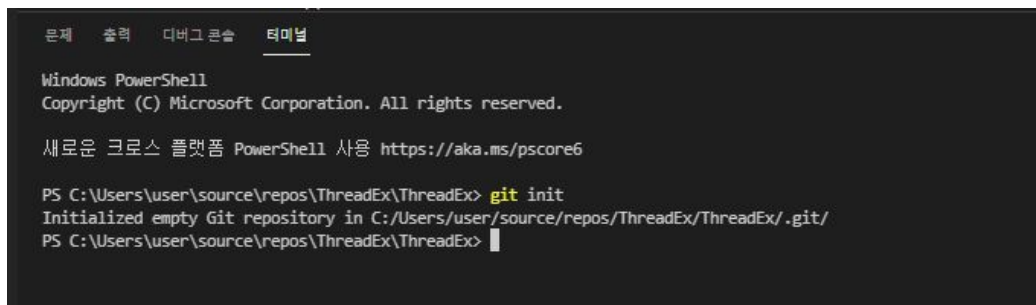
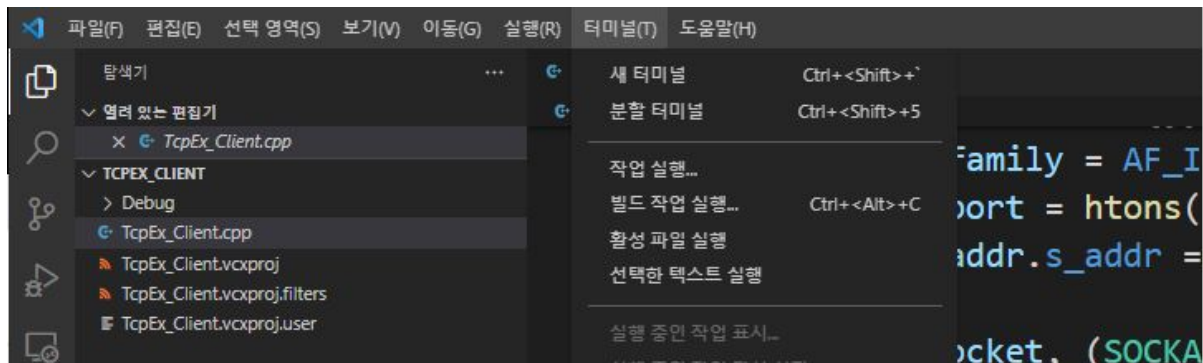
Python 100.0%

참고 자료



3. GitHub에 내 프로젝트 올리기 - VS Code , Terminal

VsCode 실행 후 터미널 창 실행



입력 << git init

3.GitHub에 내 프로젝트 올리기 - VS Code , Terminal

문제 출력 디버그 콘솔 터미널

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 <https://aka.ms/pscore6>

```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git init
Initialized empty Git repository in C:/Users/user/source/repos/ThreadEx/ThreadEx/.git/
```

```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git add .
```

```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git status
```

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

입력<<git add .

‘.’은 전부다 올리겠다는 뜻

‘.’대신 특정 파일 지정 가능

입력<<git status

상태 확인


```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git commit -m "first commit"
[master (root-commit) 4191a4f] first commit
19 files changed, 230 insertions(+)
create mode 100644 Debug/Main.obj
create mode 100644 Debug/ThreadEx.Build.CppClean.log
create mode 100644 Debug/ThreadEx.exe.recipe
create mode 100644 Debug/ThreadEx.ilc
create mode 100644 Debug/ThreadEx.log
create mode 100644 Debug/ThreadEx.tlog/CL.command.1.tlog
create mode 100644 Debug/ThreadEx.tlog/CL.read.1.tlog
create mode 100644 Debug/ThreadEx.tlog/CL.write.1.tlog
create mode 100644 Debug/ThreadEx.tlog/ThreadEx.lastbuildstate
create mode 100644 Debug/ThreadEx.tlog/link.command.1.tlog
create mode 100644 Debug/ThreadEx.tlog/link.read.1.tlog
create mode 100644 Debug/ThreadEx.tlog/link.write.1.tlog
create mode 100644 Debug/ThreadEx.vcxproj.FileListAbsolute.txt
create mode 100644 Debug/vc142.idb
create mode 100644 Debug/vc142.pdb
create mode 100644 Main.cpp
create mode 100644 ThreadEx.vcxproj
create mode 100644 ThreadEx.vcxproj.filters
create mode 100644 ThreadEx.vcxproj.user
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> █
```

입력 <<git commit -m "first commit"

"first commit"이라는 이름으로
이 파일의 버전을 만듦

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

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/RyuHongGeol/TeamProject.git>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 "# TeamProject" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/RyuHongGeol/TeamProject.git
git push -u origin main
```

새로 Repository
생성한 후 생기는
Command line 복사 후
붙여넣기

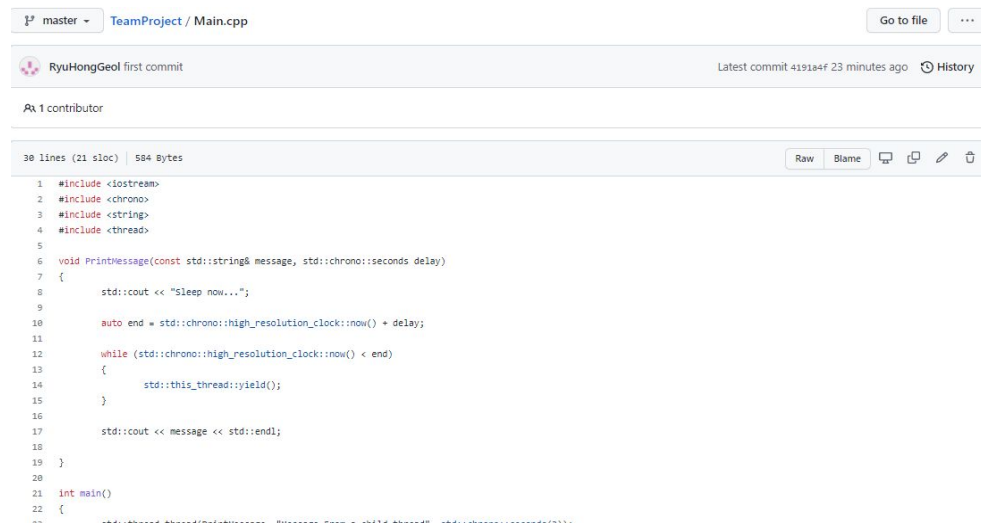
```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git remote add origin https://github.com/RyuHongGeol/TeamProject.git
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git remote -v
origin https://github.com/RyuHongGeol/TeamProject.git (fetch)
origin https://github.com/RyuHongGeol/TeamProject.git (push)
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> |
```

입력 <<git remote -v
연결된 repository
주소 확인

3. GitHub에 내 프로젝트 올리기 - VS Code , Terminal

```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git push origin master
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (23/23), 287.98 KiB | 3.13 MiB/s, done.
Total 23 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/RyuHongGeol/TeamProject.git
* [new branch]      master -> master
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> 
```

입력 <<git push origin master
mater에 내 프로젝트 push(밀어넣기)
하기



master TeamProject / Main.cpp

RyuHongGeol first commit Latest commit 4191a4f 23 minutes ago History

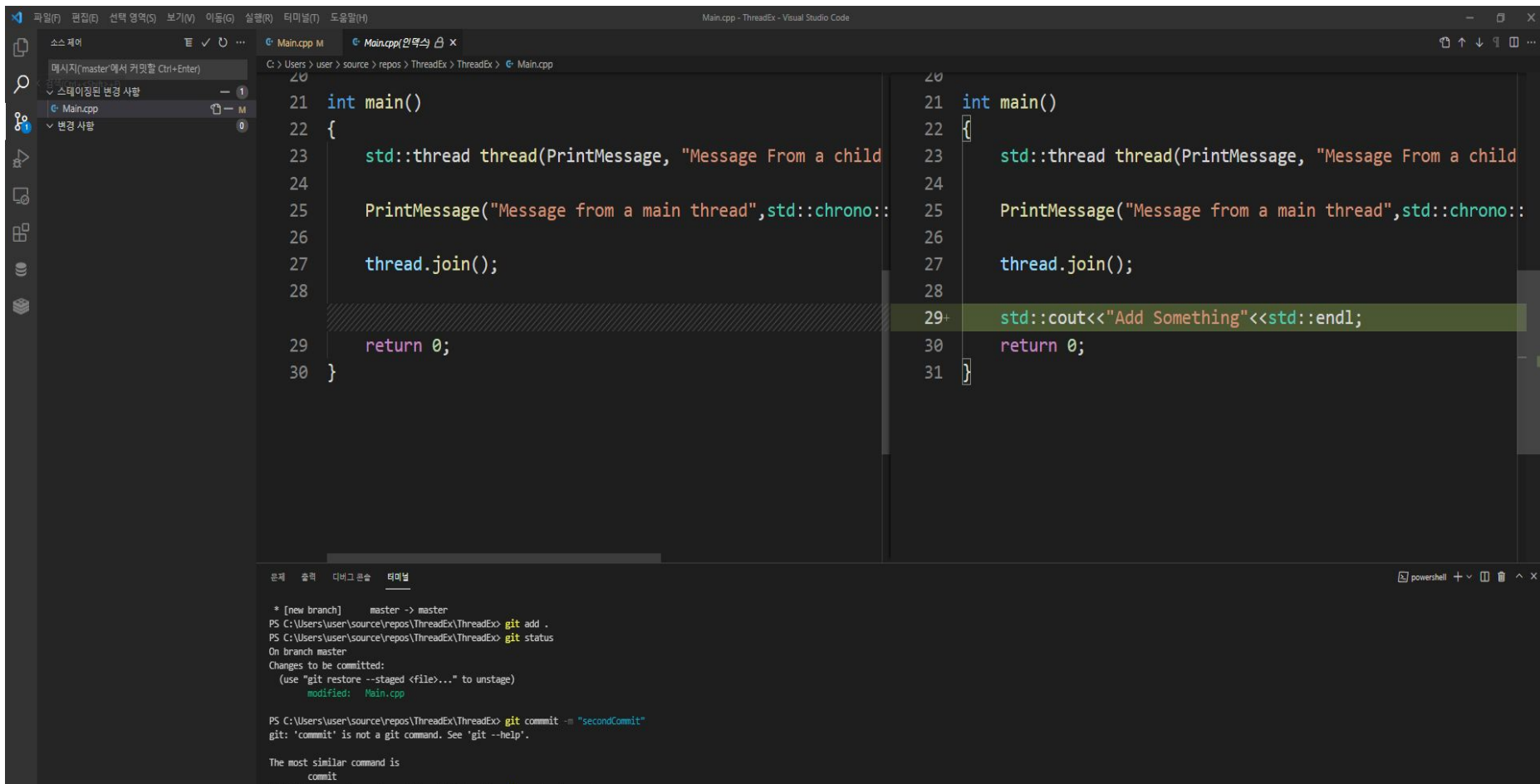
1 contributor

30 lines (21 sloc) | 584 Bytes

```
1 #include <iostream>
2 #include <chrono>
3 #include <string>
4 #include <thread>
5
6 void PrintMessage(const std::string& message, std::chrono::seconds delay)
7 {
8     std::cout << "Sleep now...";
9
10    auto end = std::chrono::high_resolution_clock::now() + delay;
11
12    while (std::chrono::high_resolution_clock::now() < end)
13    {
14        std::this_thread::yield();
15    }
16
17    std::cout << message << std::endl;
18 }
19
20 int main()
21 {
22     std::thread thread(PrintMessage, "Sleep now", std::chrono::seconds(5));
23 }
```

Git hub에서 올라간 코드 확인

3. GitHub에 내 프로젝트 올리기 - VS Code , Terminal

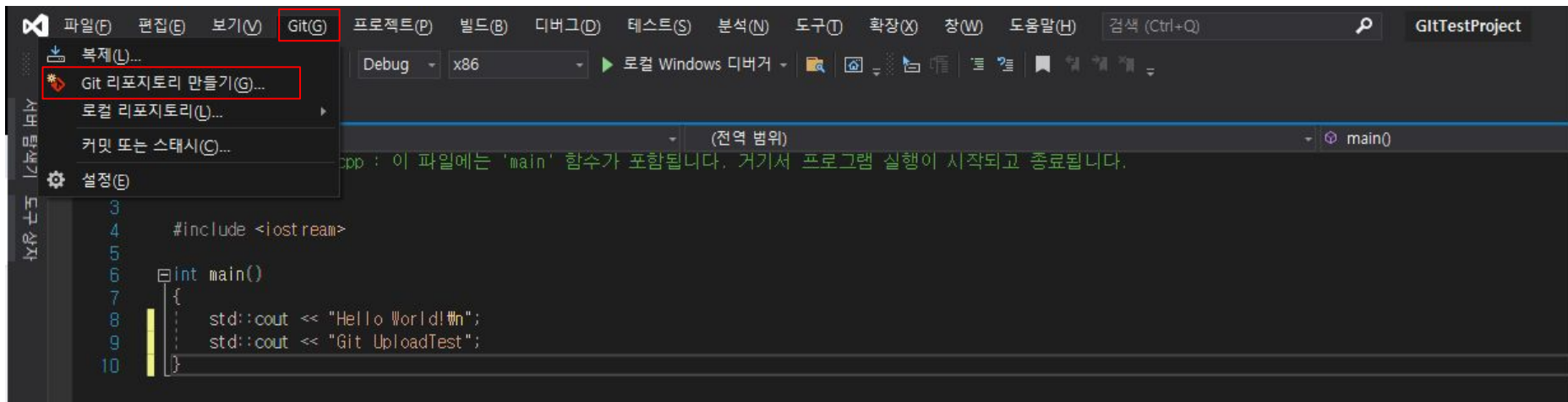


3. GitHub에 내 프로젝트 올리기 - VS Code , Terminal

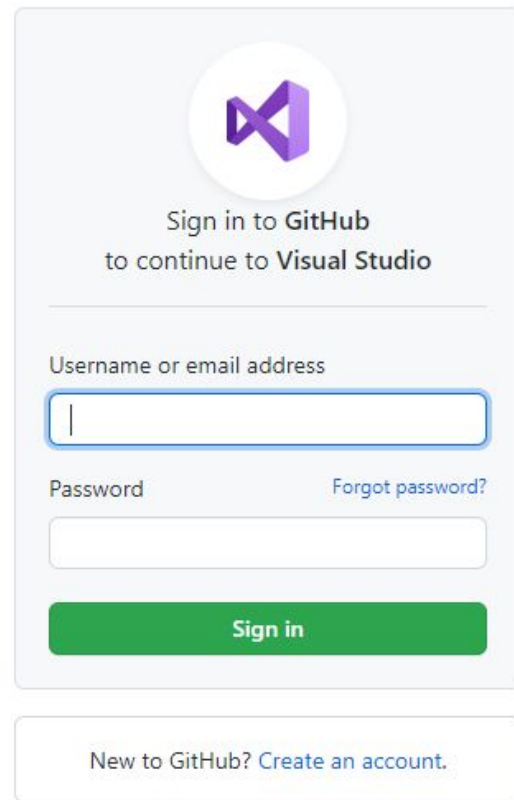
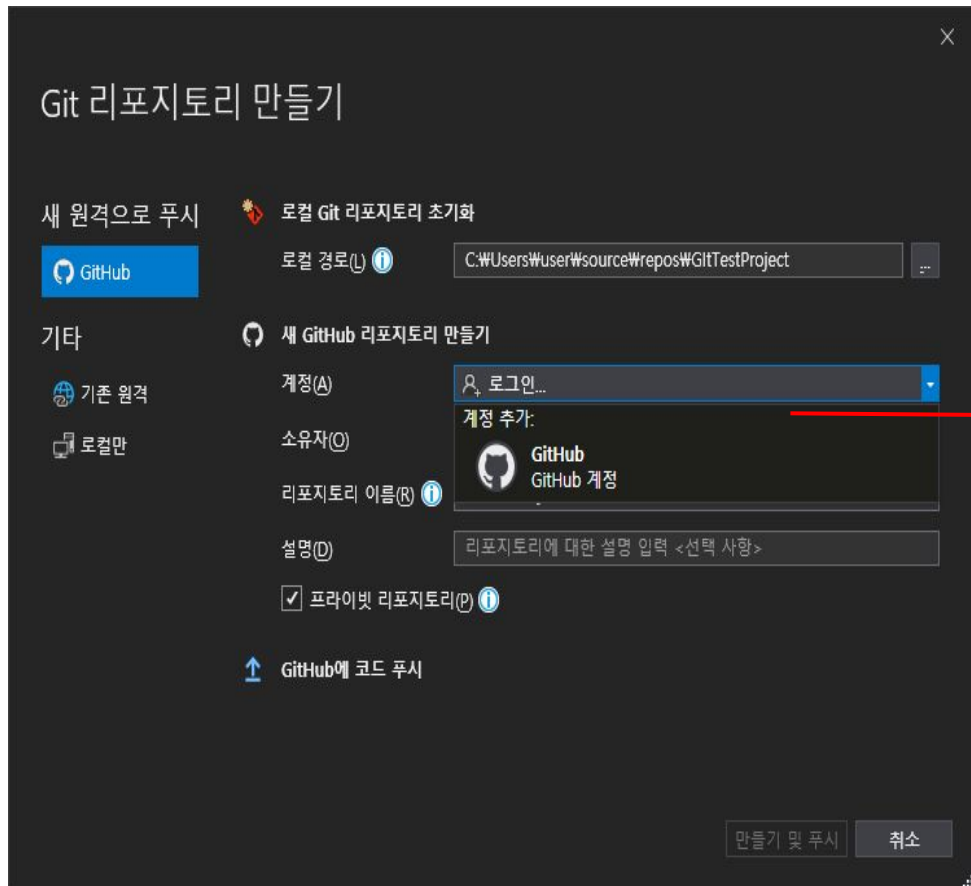
```
* [new branch]      master -> master
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git add .
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   Main.cpp
```

```
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git commit -m "secondCommit"
[master 251c669] secondCommit
 1 file changed, 1 insertion(+)
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 335 bytes | 335.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/RyuHongGeol/TeamProject.git
 4191a4f..251c669  master -> master
PS C:\Users\user\source\repos\ThreadEx\ThreadEx> 
```

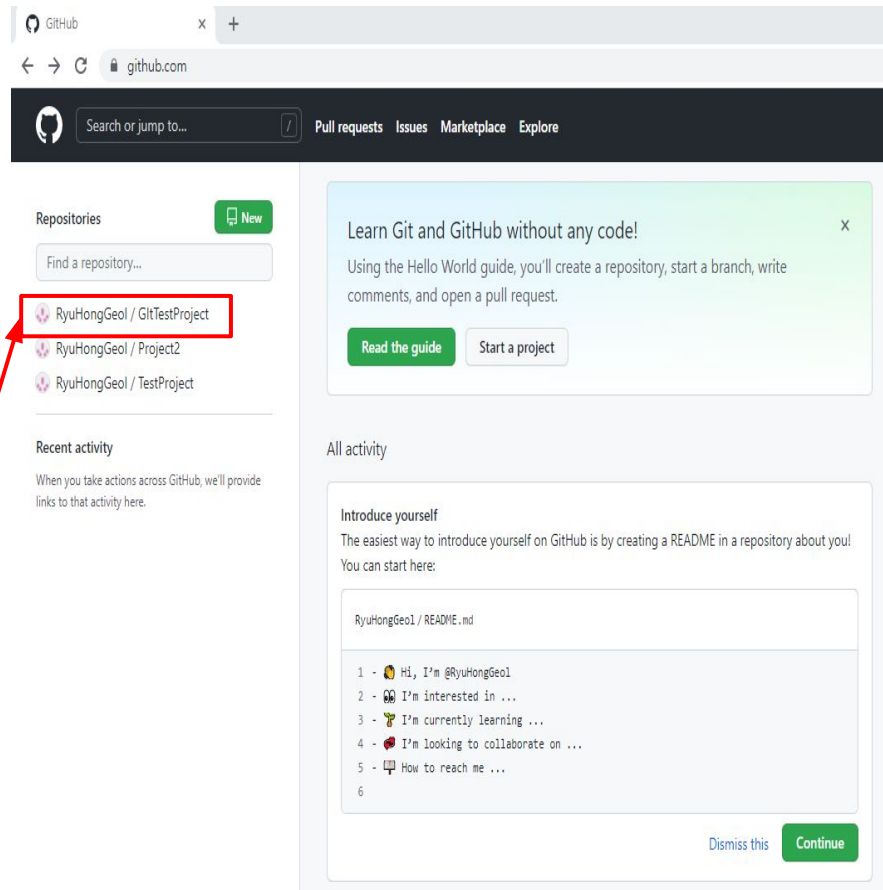
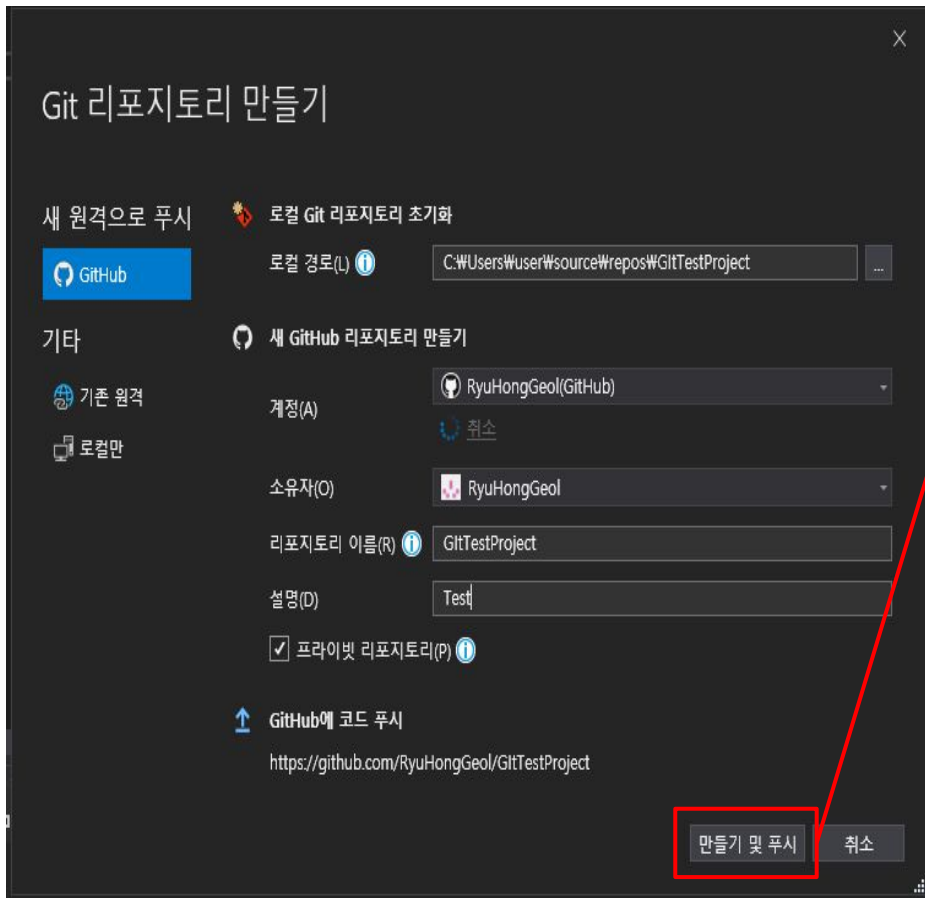
3. GitHub에 내 프로젝트 올리기 - VisualStudio



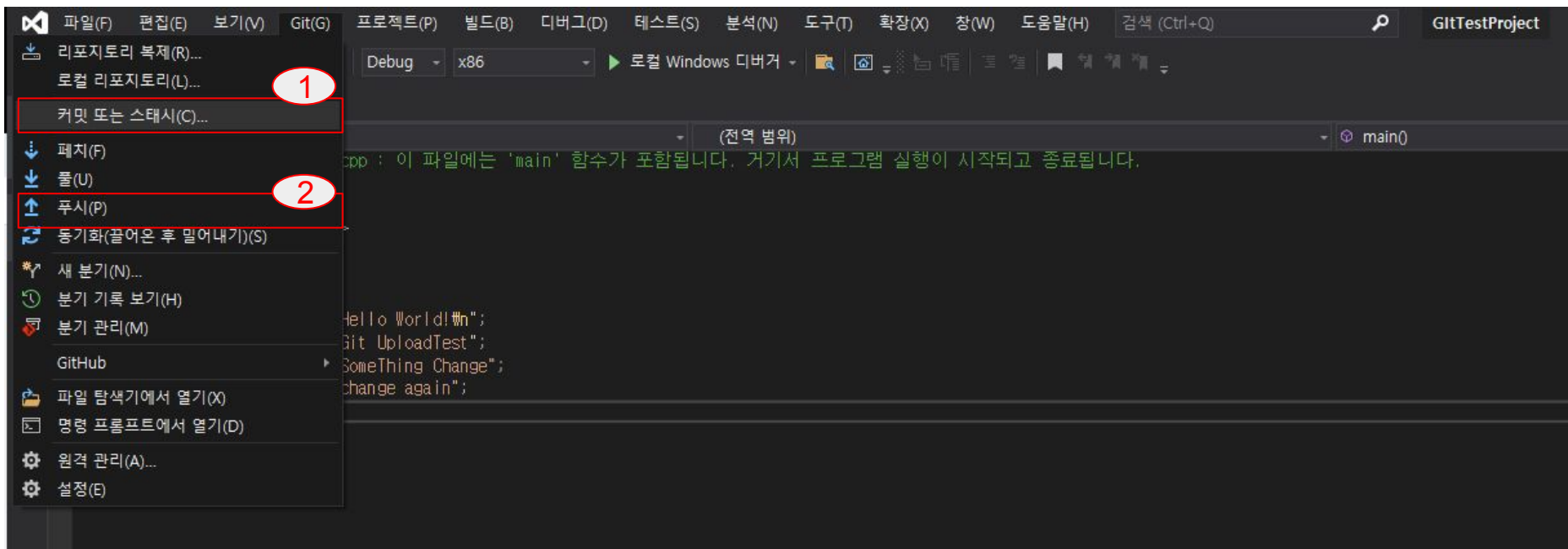
3. GitHub에 내 프로젝트 올리기 - VisualStudio



3. GitHub에 올라 간 것 확인



4. GitHub에 내 프로젝트 업데이트하기 - VisualStudio



1. 커밋해서 로컬에 업데이트 후


2. GitHub의 Repository에 Push하여 업데이트

second commit

[Browse files](#) master RyuHongGeol committed 11 minutes ago



1 parent a6fbaef

commit b3c823eb4b195826c5b2fbc877053a3a05e63f77

 Showing 1 **changed file** with 1 **addition** and 0 **deletions**.

Split

Unified

✓ 1  GItTestProject/GItTestProject.cpp 

↑	@@ -8,4 +8,5 @@ int main()
8	8 + std::cout << "Hello World!\n";
9	9 std::cout << "Git UploadTest";
10	10 std::cout << "SomeThing Change";
11	11 + std::cout << "change again";
11	12 } -

Update 된 부분 확인 가능

3. GitHub에 내 프로젝트 올리기 - VS Code , Terminal

secondCommit

master

RyuHongGeol committed 3 minutes ago

1 parent 4191a4f commit 251c669287ba557e2de9e9185c40c5659d5a4f21

Showing 1 changed file with 1 addition and 0 deletions.

Split

Unified

1 Main.cpp

↑	@@ -26,5 +26,6 @@ int main()
26	26
27	27 thread.join();
28	28
29	+ std::cout<<"Add Something"<<std::endl;
29	30 return 0;
30	31 }

0 comments on commit 251c669

Lock conversation

4.Git 소스코드 공유하기

The screenshot shows the GitHub interface for the repository 'RyuHongGeol / pythonProject8'. The 'Settings' tab is selected in the top navigation bar. On the left sidebar, the 'Manage access' option is highlighted. The main content area is titled 'Who has access' and shows two sections: 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. The 'PUBLIC REPOSITORY' section indicates that the repository is public and visible to anyone, with a 'Manage' link. The 'DIRECT ACCESS' section shows that 0 collaborators have access to the repository, with a note that only the user can contribute. Below these sections, the 'Manage access' section is displayed, featuring a lock icon and the message 'You haven't invited any collaborators yet'. A green 'Add people' button is visible in this section.

Search or jump to... Pull requests Issues Marketplace Explore

RyuHongGeol / pythonProject8 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Pages

Moderation settings

Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

[Manage](#)

DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

[Add people](#)

4.Git 소스코드 공유하기

1) Github에서 소스코드 다운로드

`git clone` 주소 폴더이름

- 주소는 깃허브에서 들고와야함
- 폴더이름은 선택사항이다 (즉 없어도됨) 폴더이름을 줄 경우에는 그 폴더가 새로 생성이 되면서 그 안에 코드들이 다운로드가 되고, 폴더이름을 안줄 경우엔 깃허브 프로젝트 이름으로 폴더가 자동으로 생기고 그안에 코드들이 다운로드된다.

2) Github에서 내 브랜치(branch)만들기

`git checkout -b` 브랜치이름

newBranch had recent pushes 1 minute ago

[Compare & pull request](#)

newBranch ▾

2 branches ▾ 0 tags

[Go to file](#)[Add file ▾](#)[Code ▾](#)

This branch is 1 commit ahead of master.

[Contribute ▾](#)

RyuHongGeol first commit

a19c561 2 minutes ago 3 commits



.vscode

first commit

2 minutes ago



Debug

first commit

2 minutes ago



Main.cpp

first commit

2 minutes ago



ThreadEx.vcxproj

first commit

3 hours ago



ThreadEx.vcxproj.filters

first commit

3 hours ago



ThreadEx.vcxproj.user

first commit

3 hours ago

Help people interested in this repository understand your project by adding a README.

[Add a README](#)

3)내 브랜치에 소스코드 업데이트하기

```
git add .
```

```
git commit -m "first commit"
```

```
git push origin 브랜치이름
```

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master

←







compare: newBranch

✓ Able to merge. These branches can be automatically merged.


first commit

Write

Preview

H B I  < >     @  ↩

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

Create pull request

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects


None yet

Milestone

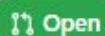
No milestone

Linked issues

Use [Closing keywords](#) in the description to automatically close issues

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

first commit #1



Open

RyuHongGeol wants to merge 1 commit into `master` from `newBranch`



Conversation 0



Commits 1



Checks 0



Files changed 6



RyuHongGeol commented now

Owner



No description provided.



first commit

a19c561

Add more commits by pushing to the `newBranch` branch on `RyuHongGeol/TeamProject`.



Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

first commit #1



Merged

RyuHongGeol merged 1 commit into `master` from `newBranch` now



Conversation 0



Commits 1



Checks 0



Files changed 6



RyuHongGeol commented 43 seconds ago

Owner



No description provided.



first commit

a19c561



RyuHongGeol merged commit 5f22c9d into `master` now

Revert



Pull request successfully merged and closed

Delete branch

You're all set—the `newBranch` branch can be safely deleted.



Write

Preview



Leave a comment

Attach files by dragging & dropping, selecting or pasting them.



Comment