

상식적으로 배우는 git/github

윤성국

Modulabs Meani.mo

버전관리

처음 걸로 해주세요



```
4
5
6     #include "polynomial.h"
7     #include <iostream>
8
9     using std::cout;
10    using std::endl;
11
12
13    //using std::endl;
14
15    /* df:hkajasdlhjioeltheklhy
16    using std::endl;dkiklashj
17    dhkias
18    akjhjioenklasyhseldhglG */
19
20
21    Polynomial::Polynomial(int termCount): nTerm(termCount)
22
23
24
25        {polyno = new double[nTerm];for(int i = 0; i < nTerm; i++)
26            polyno[i] = 0;}
27
28    Polynomial::Polynomial(const double arr[], int nArr): nTerm(nArr)
29    {polyno = new double[nTerm];
30    for(int i = 0; i < nTerm; i++)
31        polyno[i] = arr[i];}
32
33
34
35
36
37
38    Polynomial::Polynomial(const Polynomial &poly): nTerm(poly.nTerm)
39    {polyno = new double[nTerm];
40    for (int i = 0; i < nTerm; i++)
41
42
43        polyno[i] = poly.polyno[i];}
44
45    Polynomial::~Polynomial(void)
46    {delete [] polyno;}
47
48    void Polynomial::reset(int termCount)
49    {nTerm = termCount; delete [] polyno;
50
51
52
53        polyno = new double[nTerm];
54
55    for(int i = 0; i < nTerm; i++)
56        polyno[i] = 0;}
57
58    void Polynomial::show(void)
59    {bool nlus = false;for(int i = nTerm - 1; i >= 0; i--)
```

어떻게 되돌리지?



?



?

버전관리



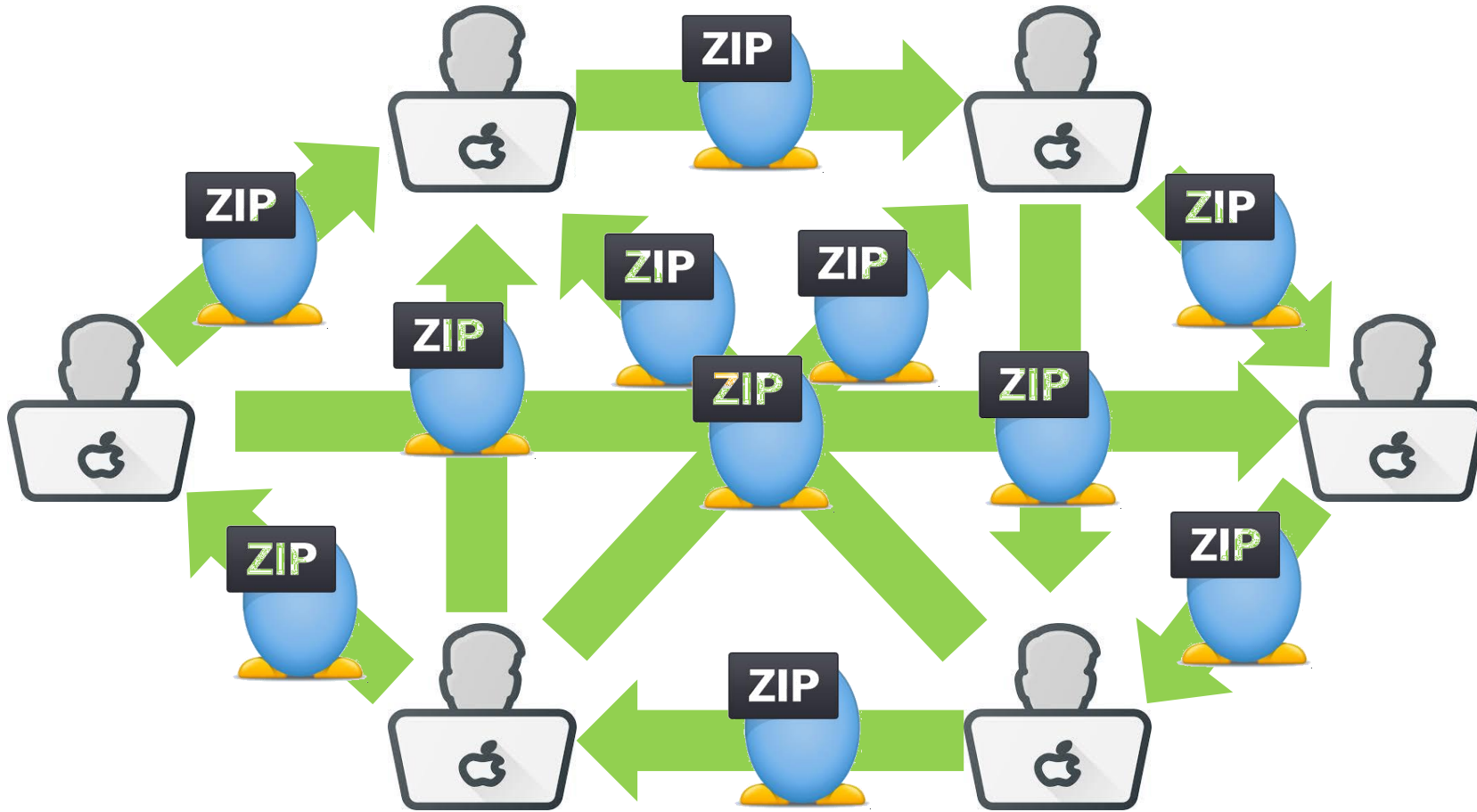
버전관리

최종_시안.png
최종_시안_(1).png
진짜최종.png
Final.png
이게_마지막_시아ㄴ.png
이게_진짜_끝.png
끝.png
최종(2).png
진짜진짜_진짜_최종.png
위에거_다아니고_이게_최종.png



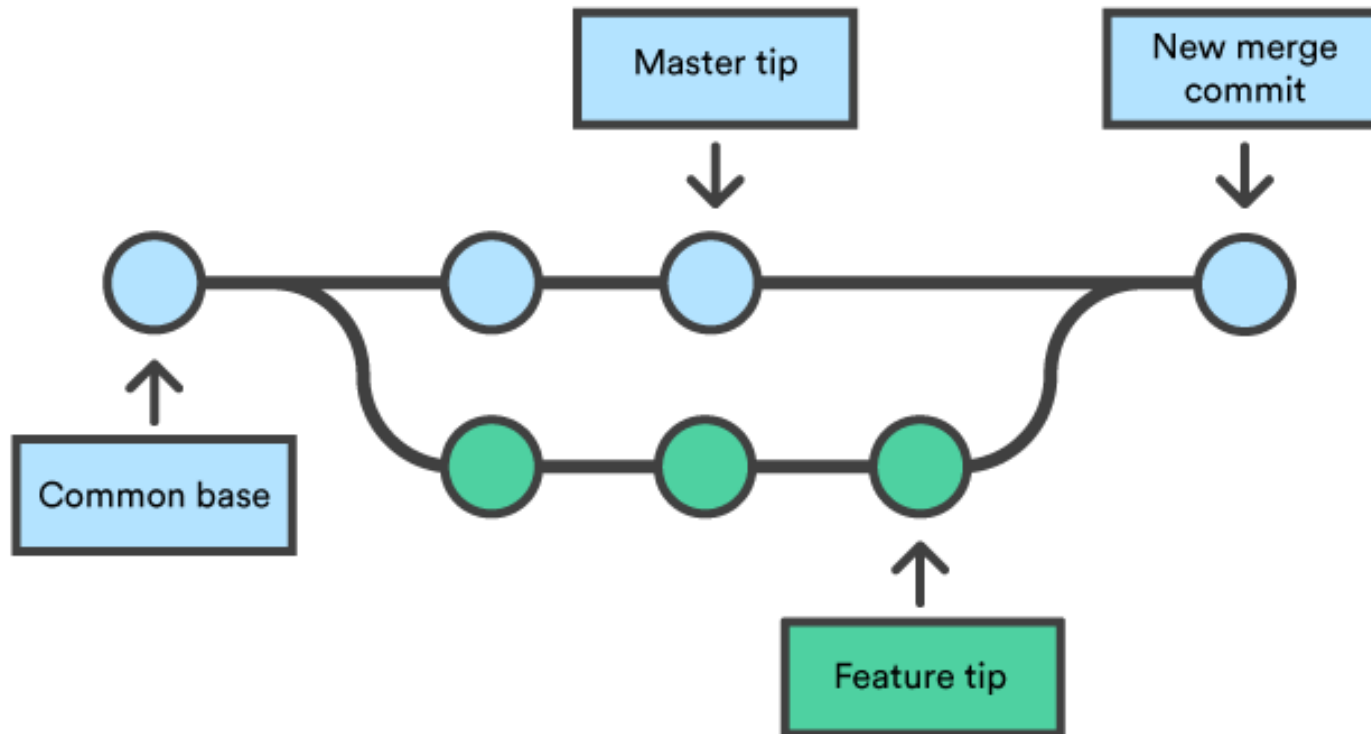
어떤 것이 진짜 최종인 것인가?
~~아름 제일 긴 게 제일 마지막이더라~~

소스코드 협업

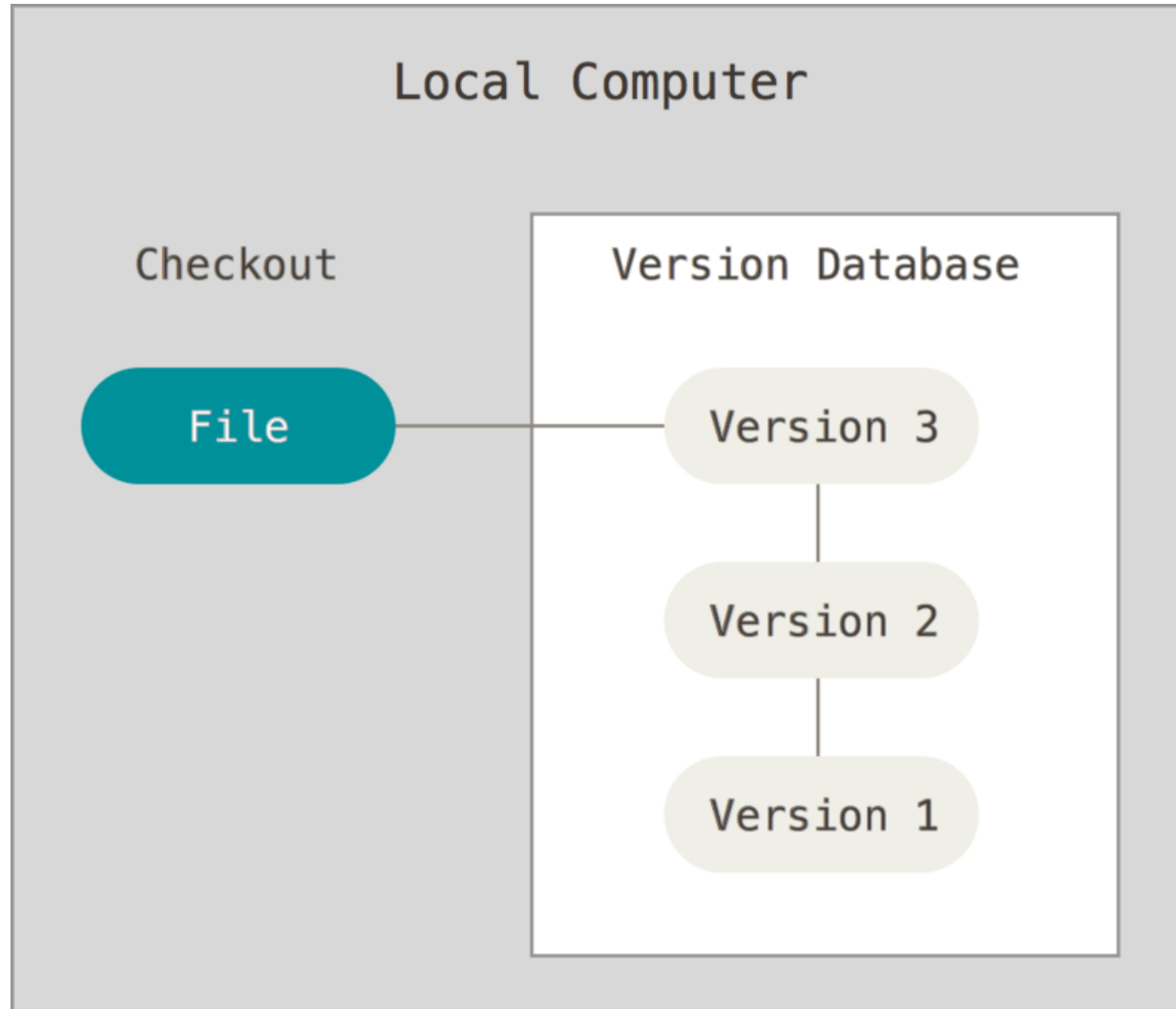


관리가 되지 않음 - 작업했던 게 없어지기도 함

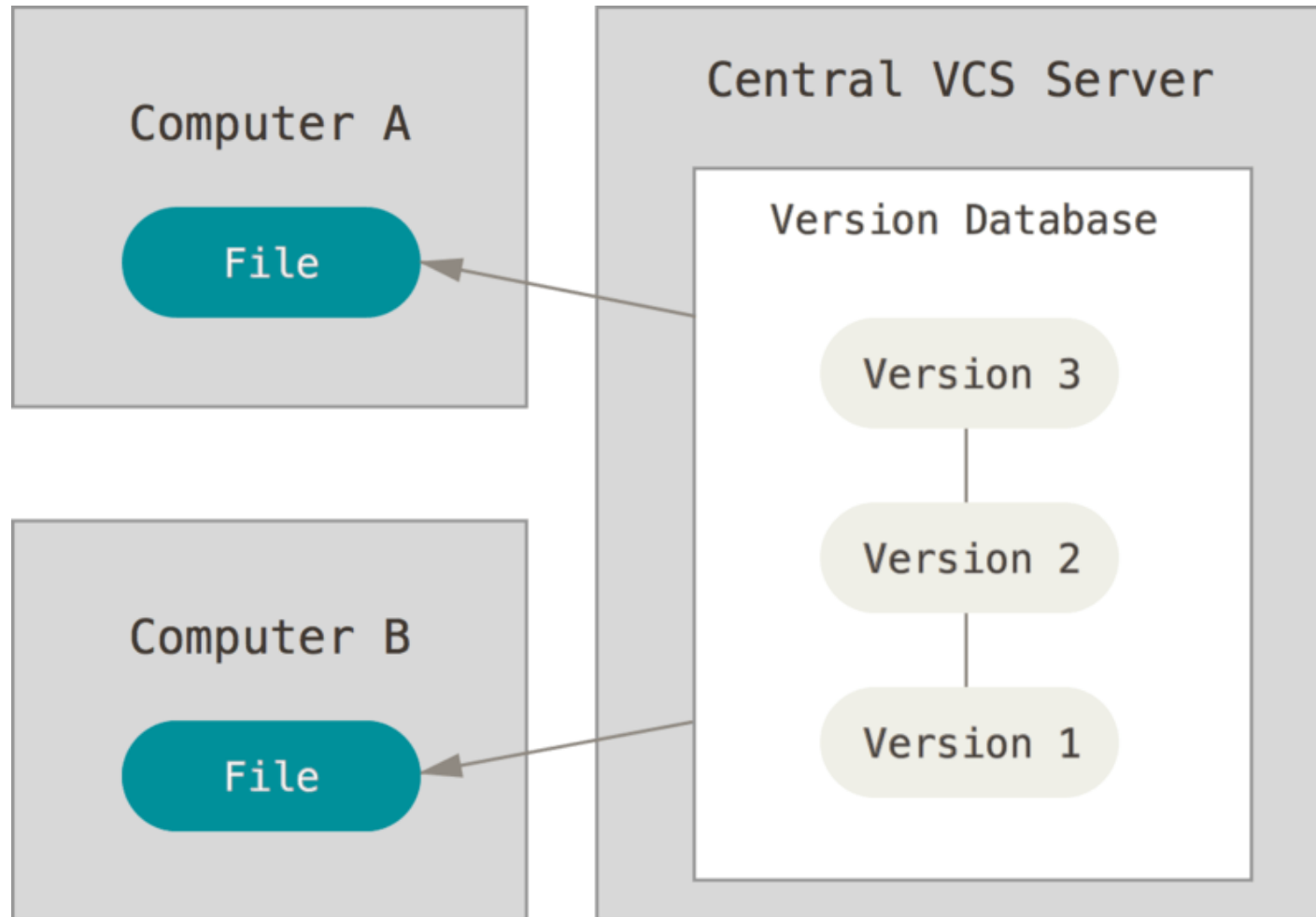
VCS : 버전관리 시스템



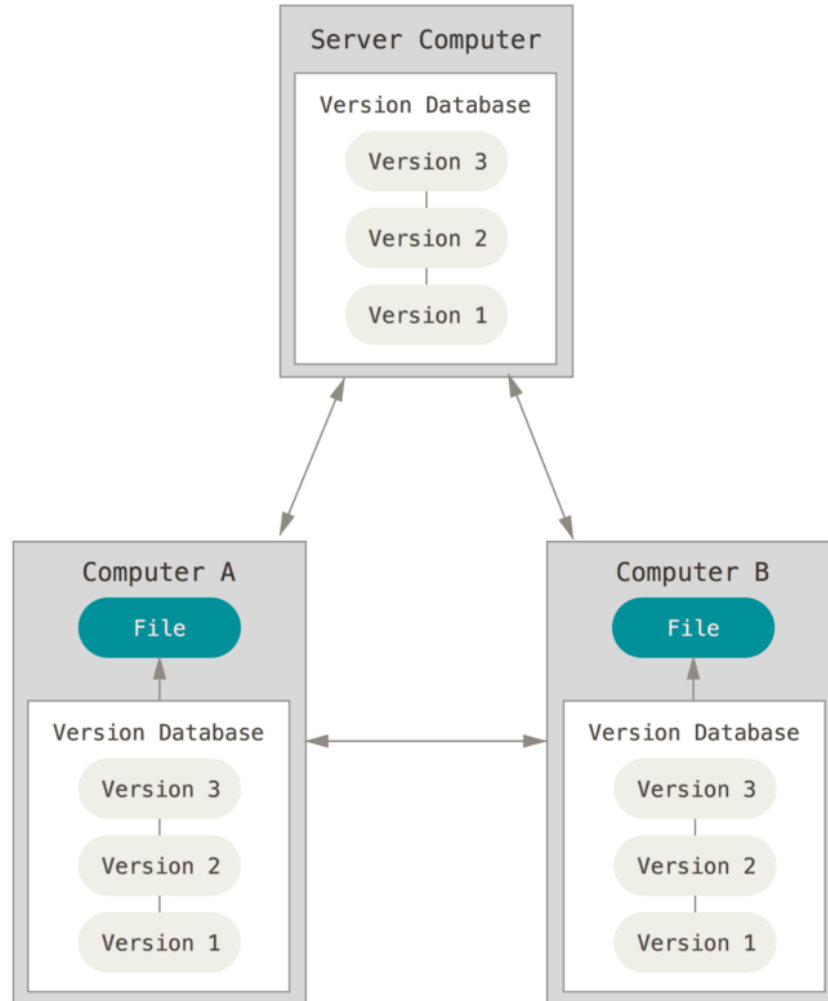
버전관리의 역사 – LVCS



버전관리의 역사 – CVCS



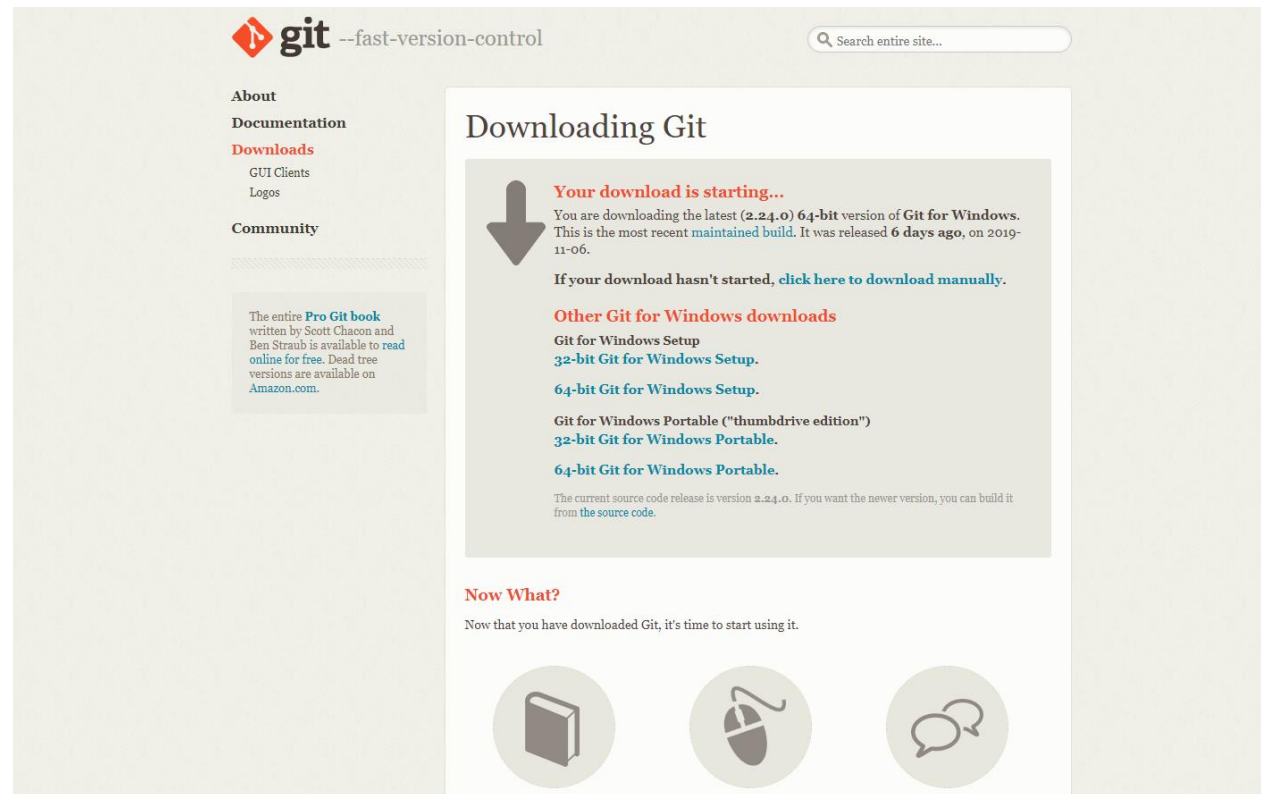
버전관리의 역사 - DVCS



git 설치

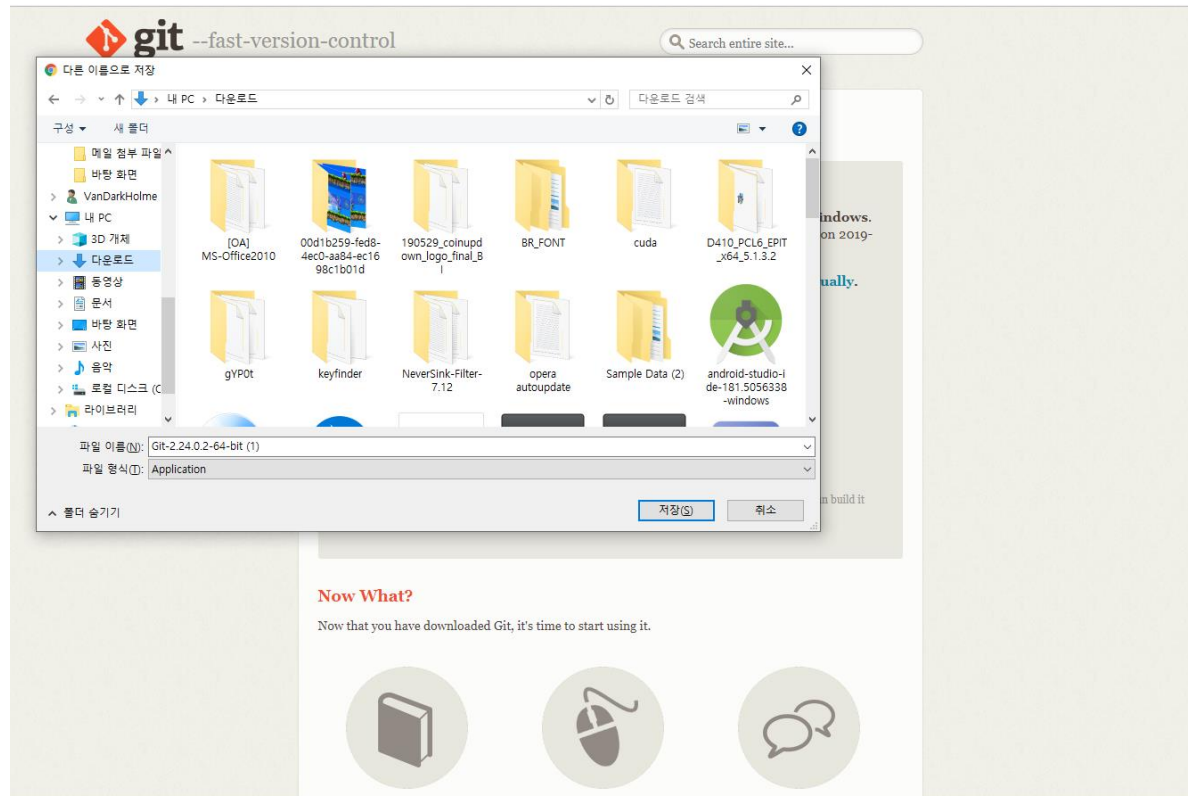
- windows

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



git 설치

- windows



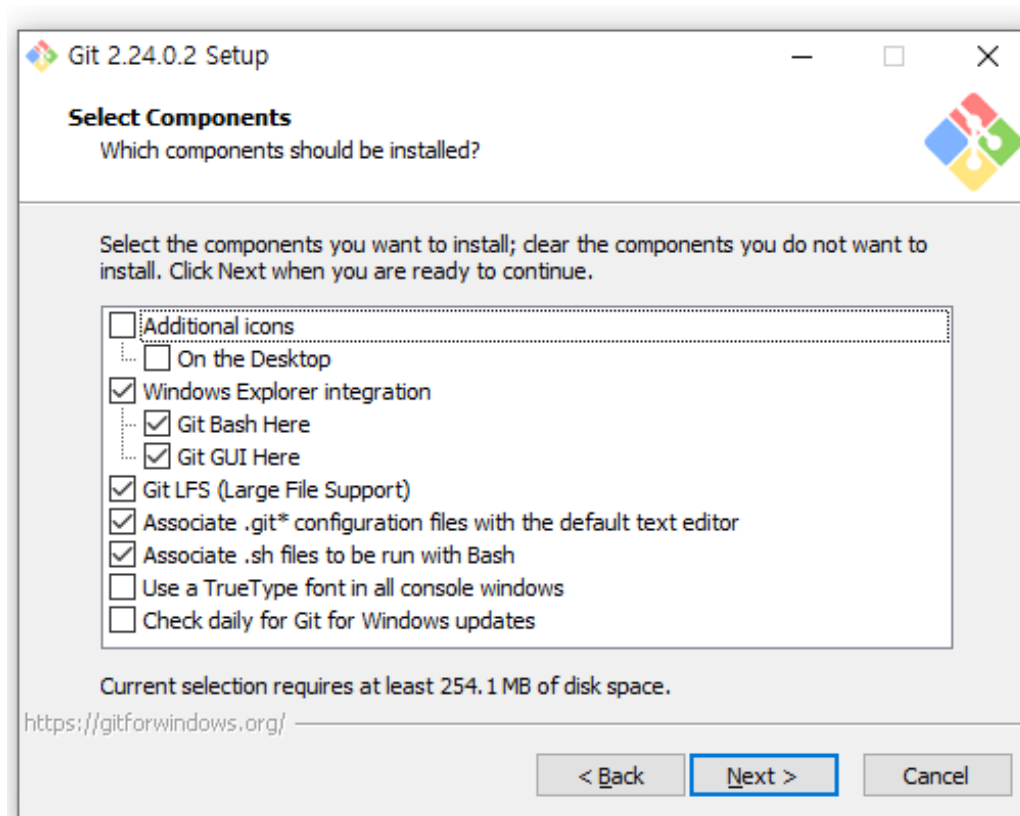
git 설치

- windows



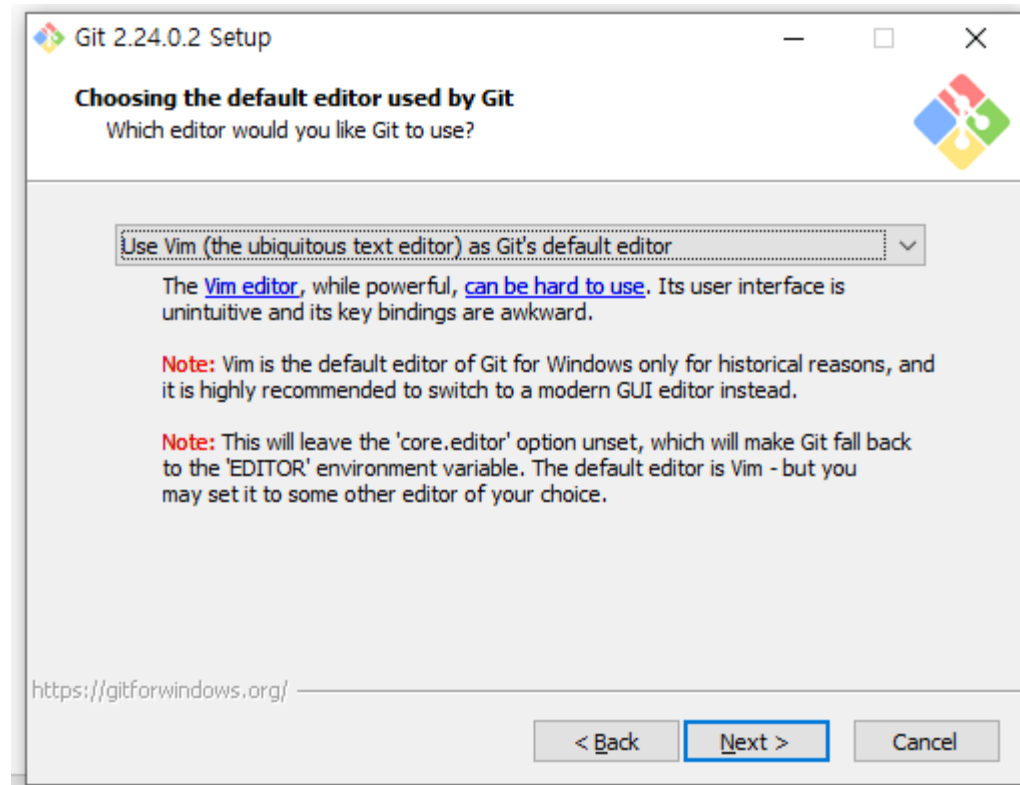
git 설치

- windows



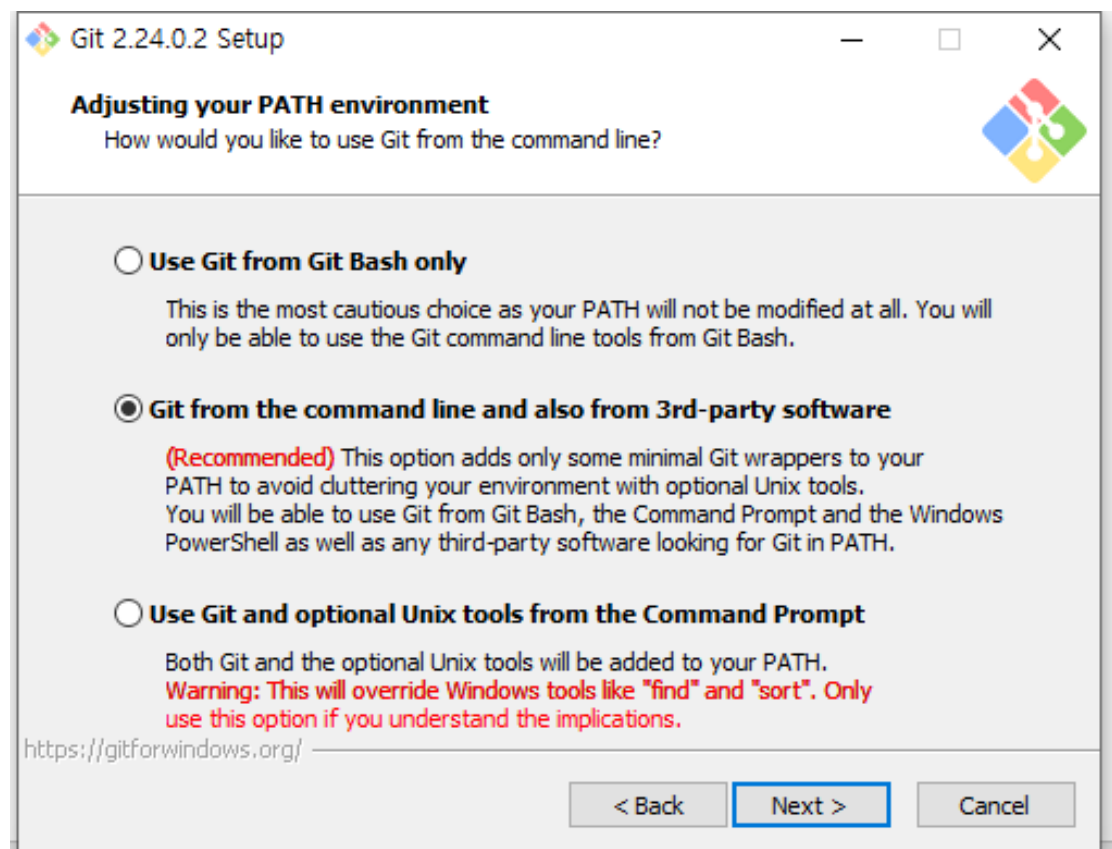
git 설치

- windows



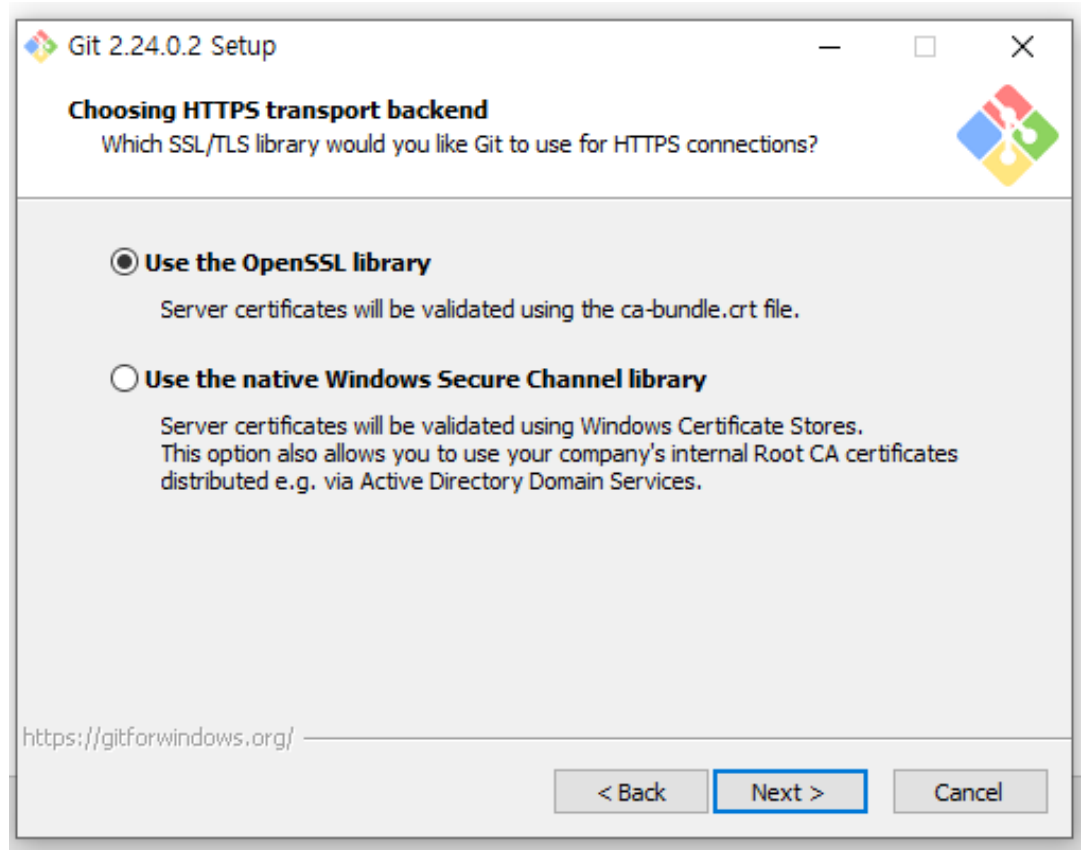
git 설치

- windows



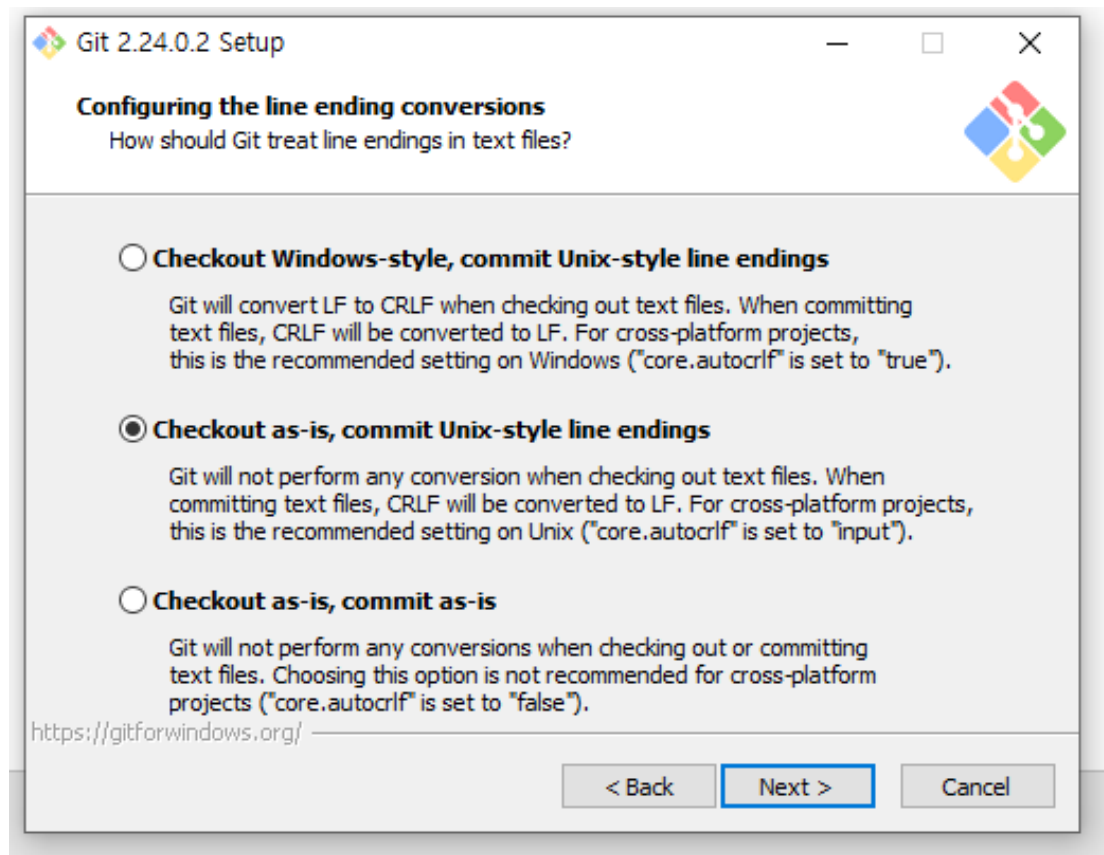
git 설치

- windows



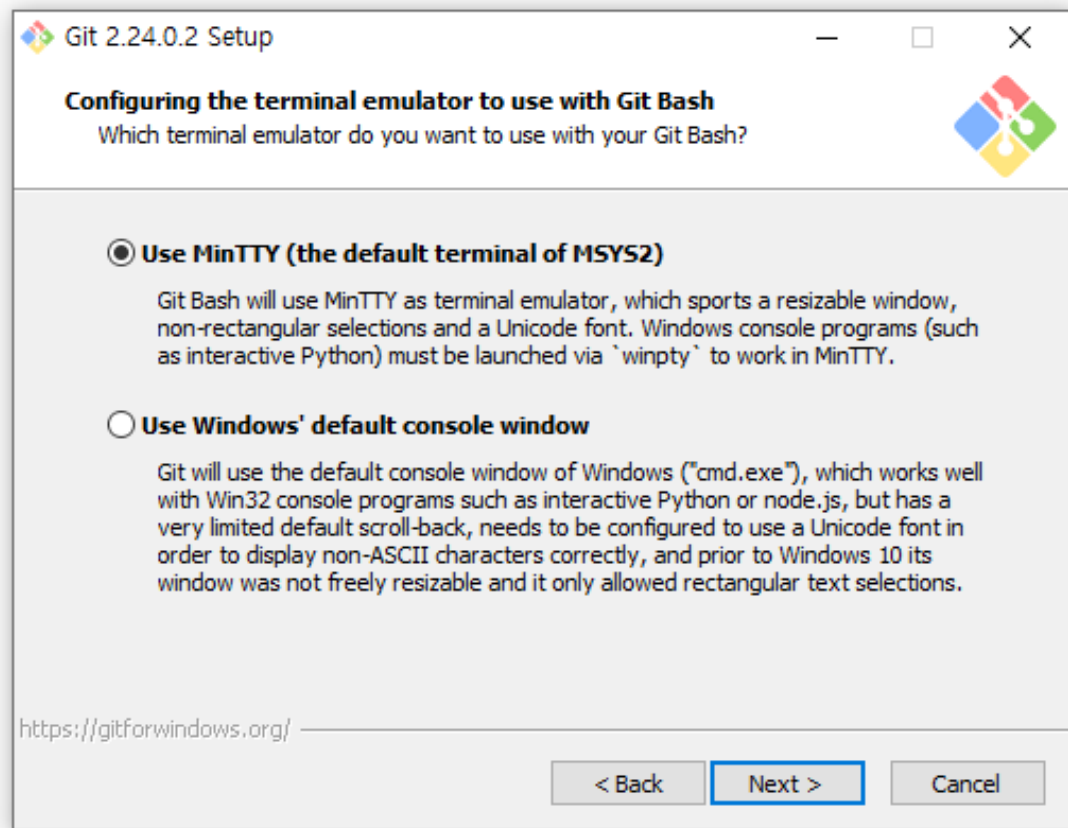
git 설치

- windows



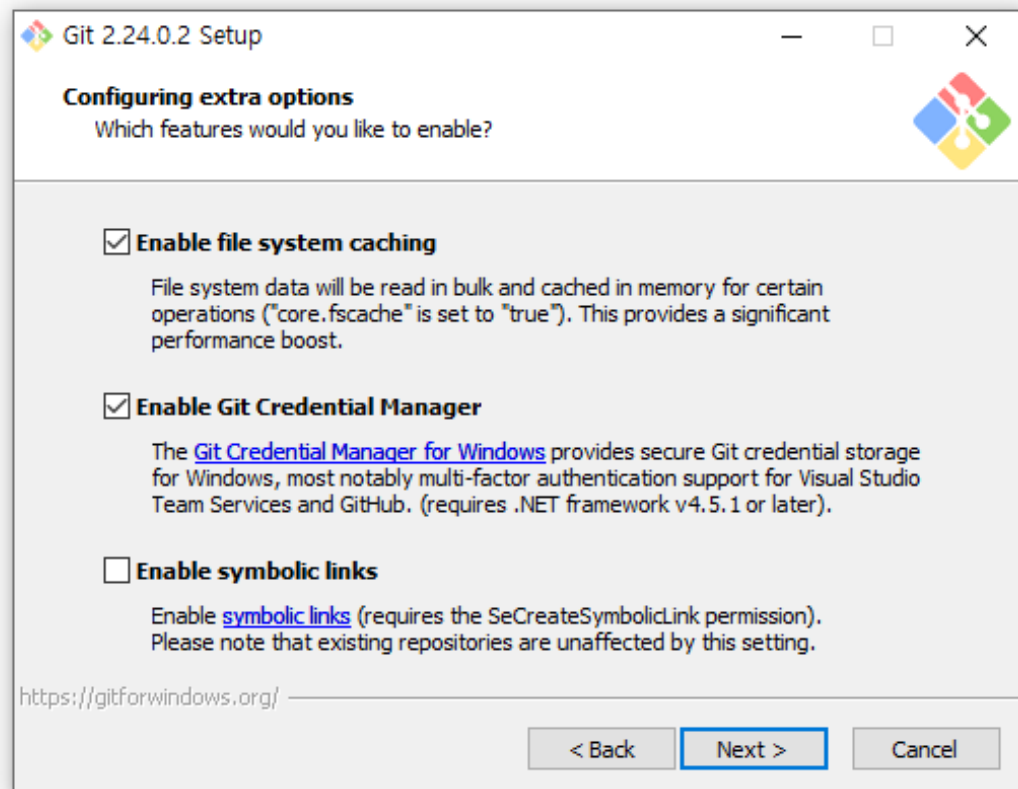
git 설치

- windows



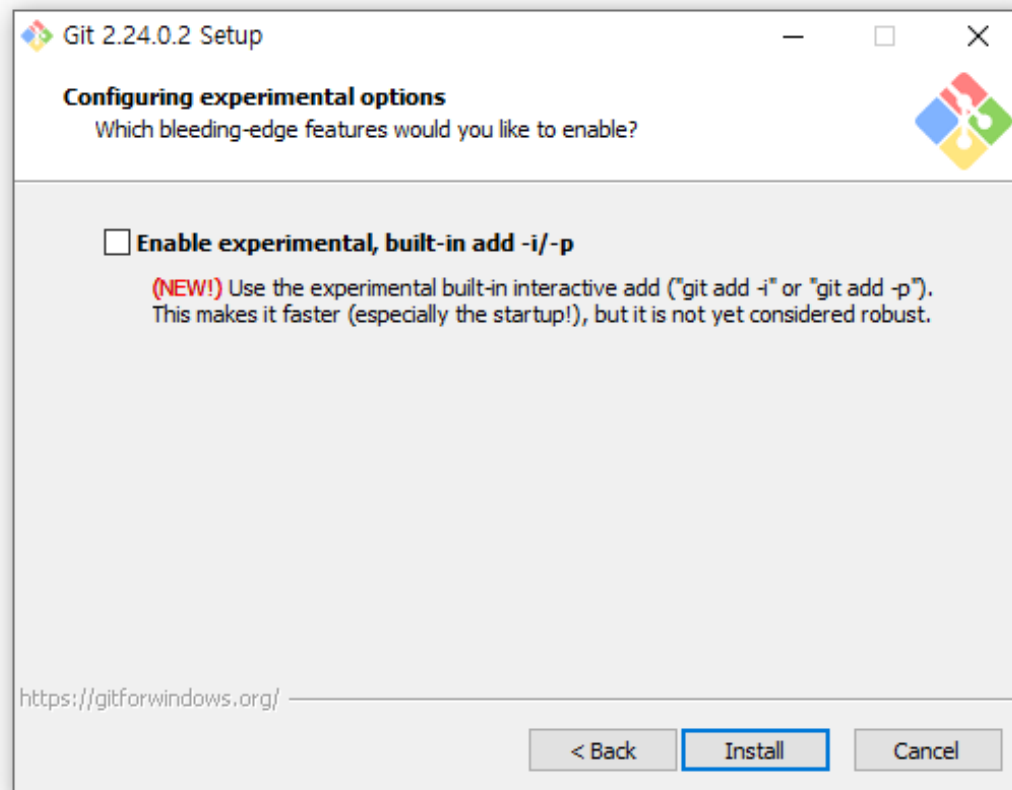
git 설치

- windows



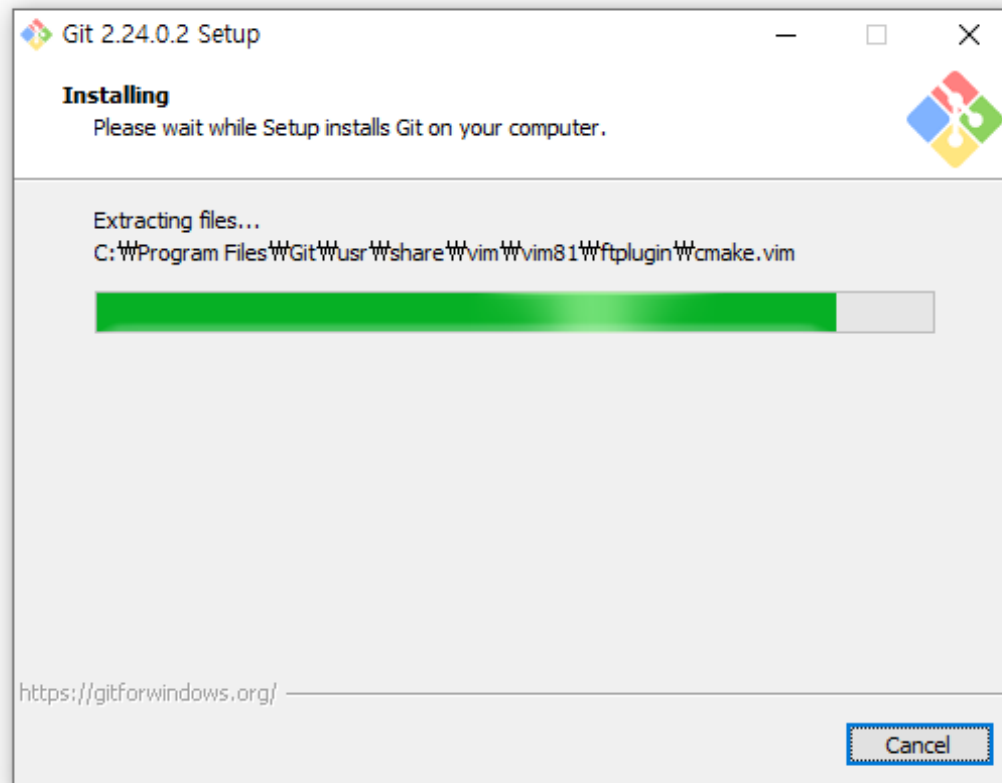
git 설치

- windows



git 설치

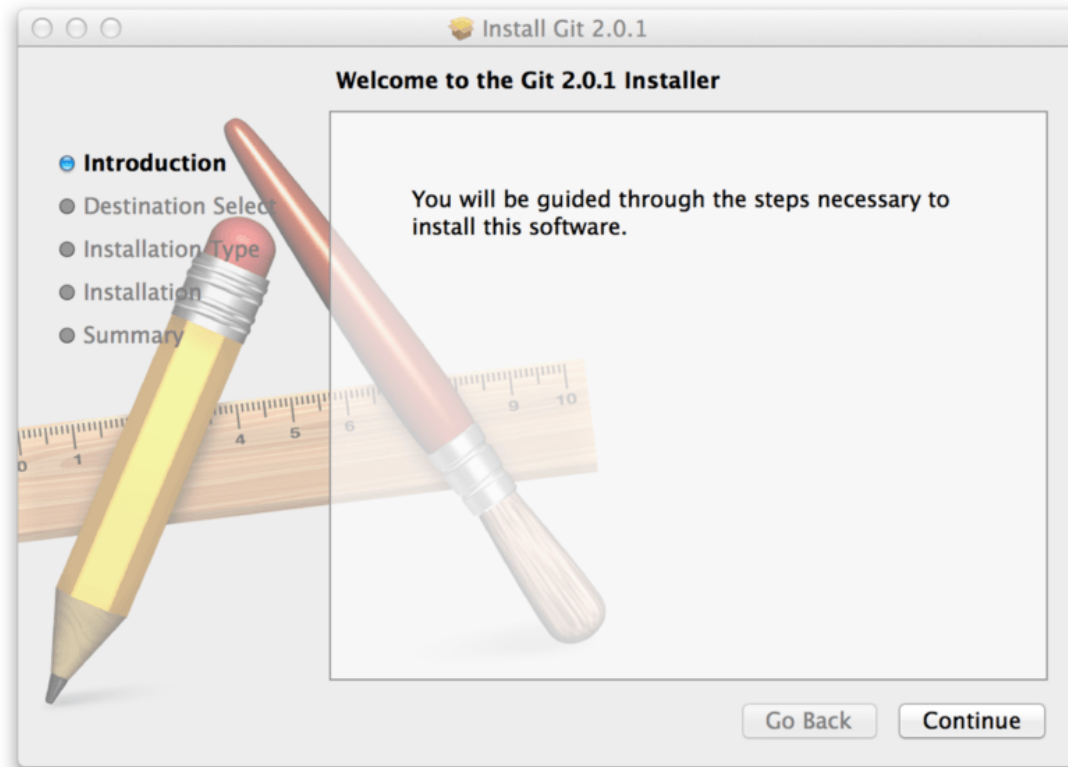
- windows



git 설치

- Mac OS

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



git 설치

- Linux

RedHat 계열

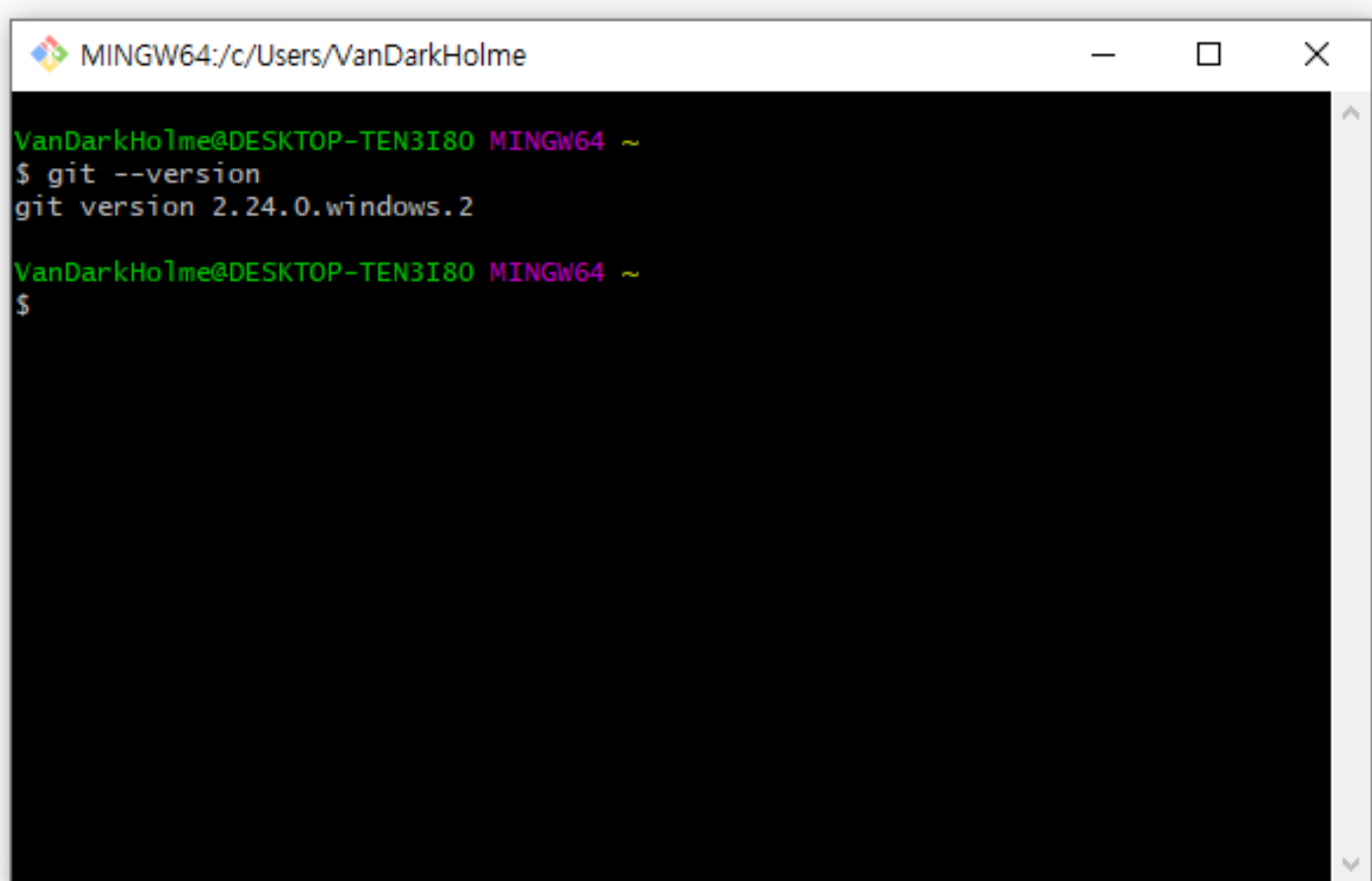
```
$ sudo dnf install git-all
```

Debian 계열

```
$ sudo apt install git-all
```

git 설치

- 설치확인



```
MINGW64:/c/Users/VanDarkHolme

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ git --version
git version 2.24.0.windows.2

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$
```


사용자 정보 지정

```
MINGW64:/c/Users/VanDarkHolme

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ git config --global user.name "brain-hack"

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ git config --global user.email deep-learning@kakao.com

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ git config --list
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
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
credential.helper=manager
core.autocrlf=input
core.fscache=true
core.symlinks=false
user.name=brain-hack
user.email=deep-learning@kakao.com

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$
```

git repository 만들기

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ mkdir learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ cd learnGit/

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit
$ git init
Initialized empty Git repository in C:/Users/VanDarkHolme/learnGit/.git/

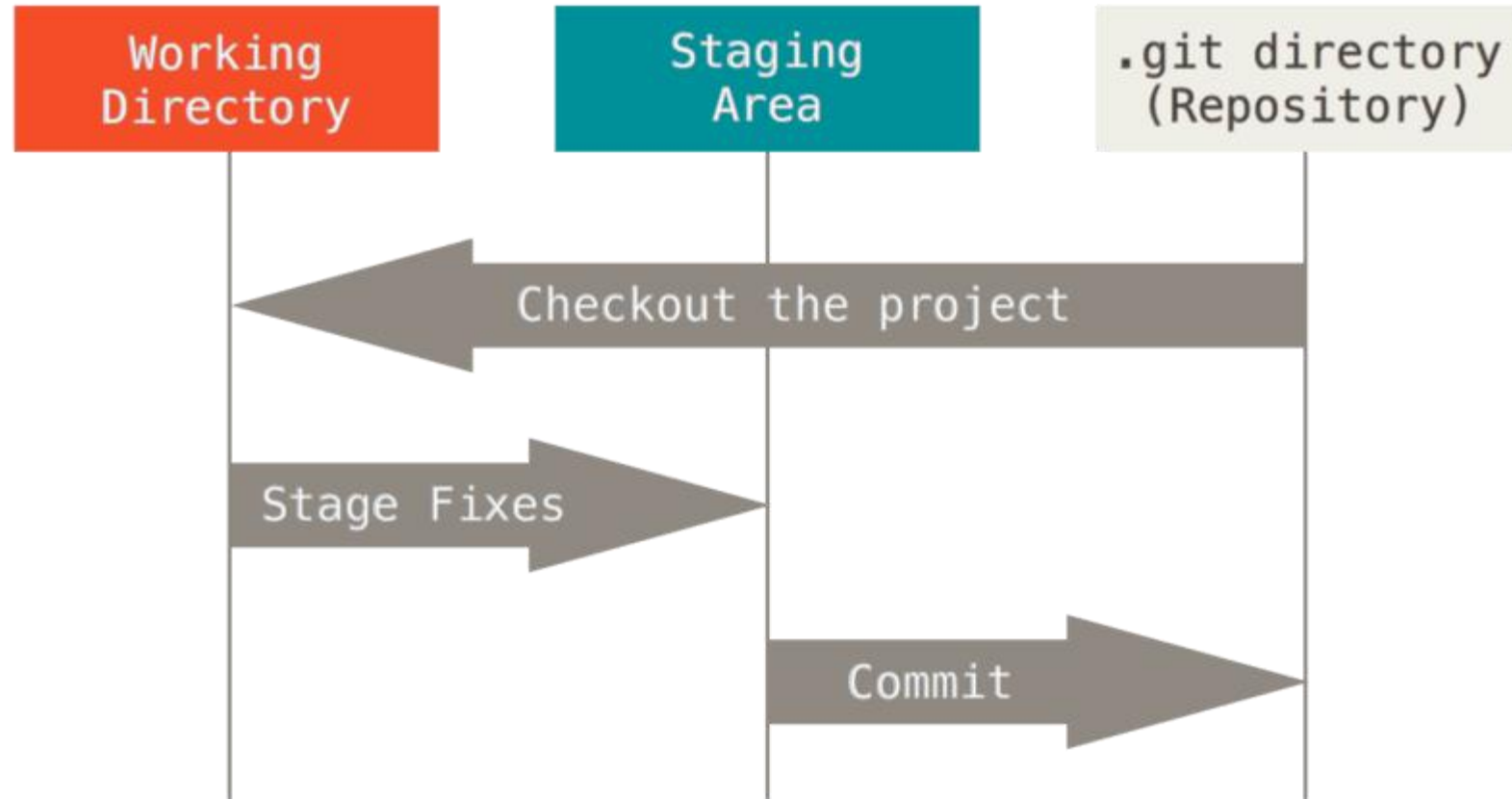
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master

No commits yet

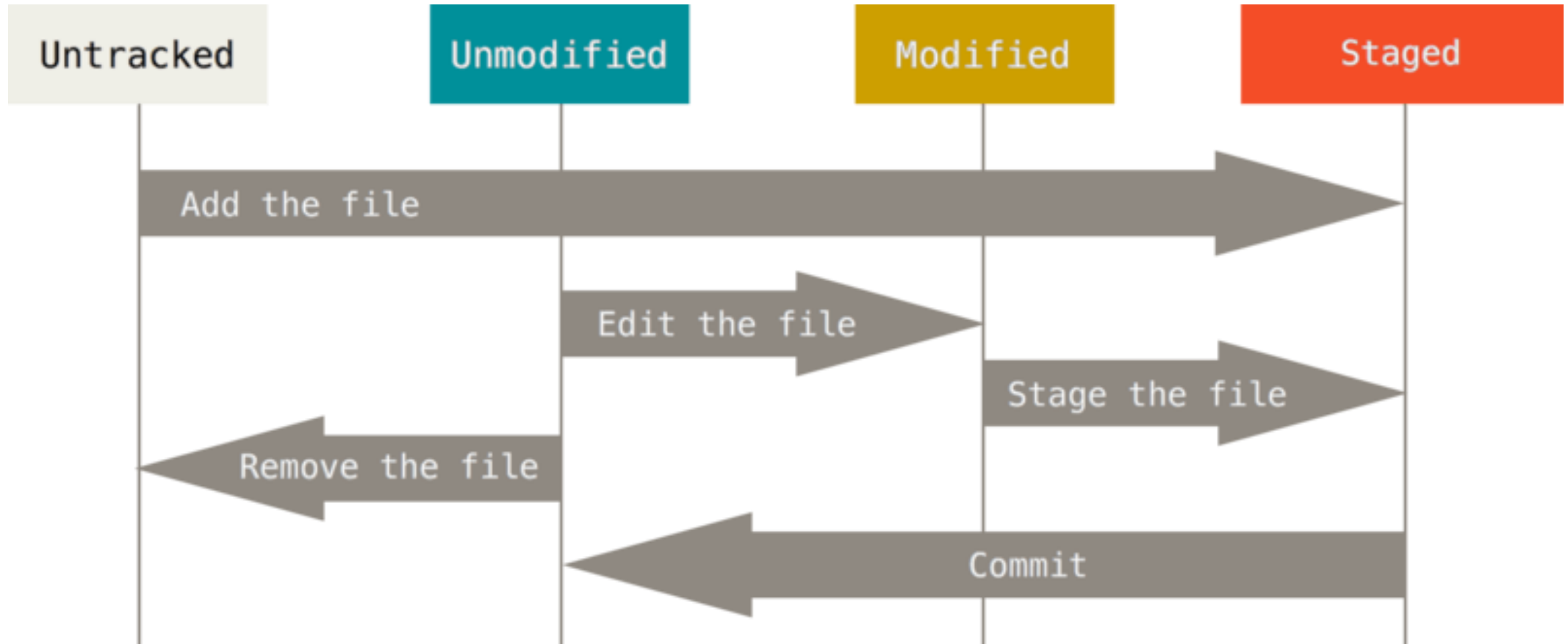
nothing to commit (create/copy files and use "git add" to track)

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$
```

git project의 세가지 단계



git repo에서 file의 life cycle



git repo에 파일 추가해보기

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ notepad README.md
```

```
파일(F) 편집(E) 서식(O) 보기(V) 도움말
# learn git

hello world!
```

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.md

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

file을 stage에 올리기

windows의 경우

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git add README.md
warning: CRLF will be replaced by LF in README.md.
The file will have its original line endings in your working directory

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git config --global core.autocrlf true

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git add README.md

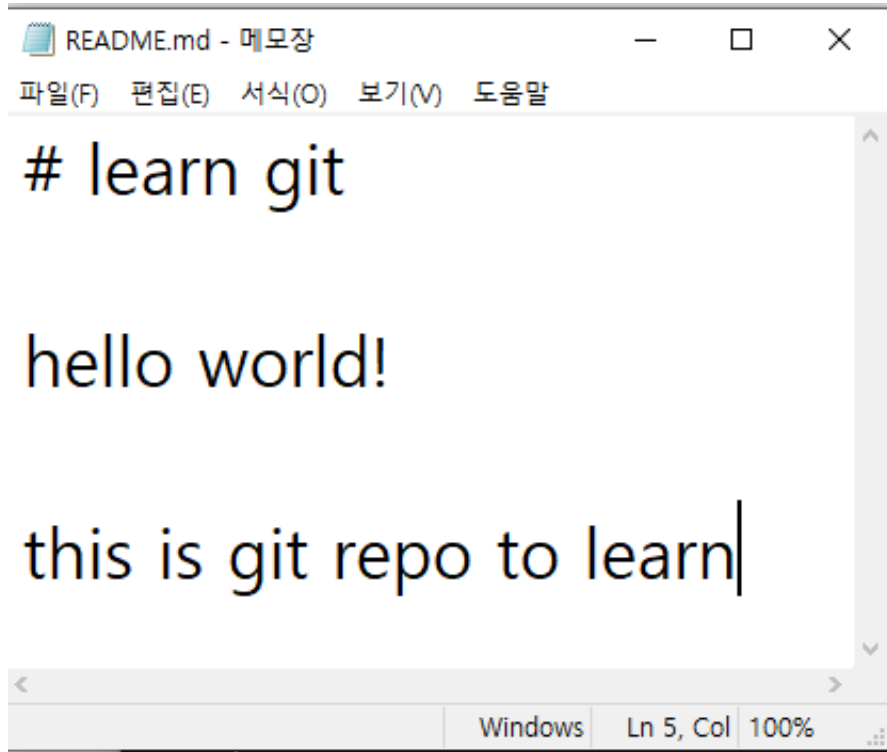
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$
```

file 내용을 추가해보기



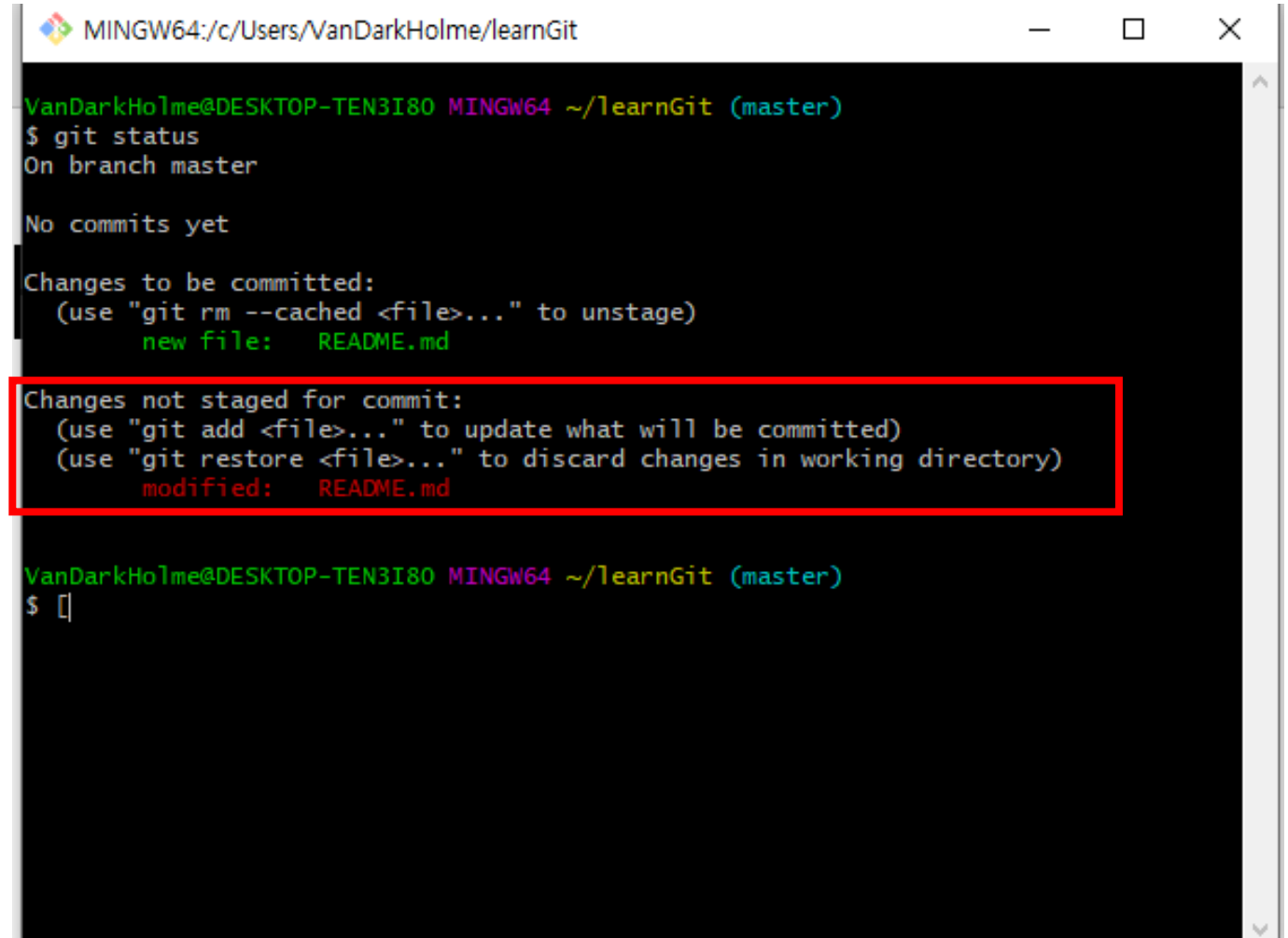
A screenshot of a text editor window titled "README.md - 메모장". The window contains the following text:

```
# learn git

hello world!

this is git repo to learn|
```

The status bar at the bottom indicates "Windows Ln 5, Col 100%".



A screenshot of a terminal window titled "MINGW64:/c/Users/VanDarkHolme/learnGit". The terminal shows the output of the 'git status' command:

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$
```

The section "Changes not staged for commit:" is highlighted with a red box.

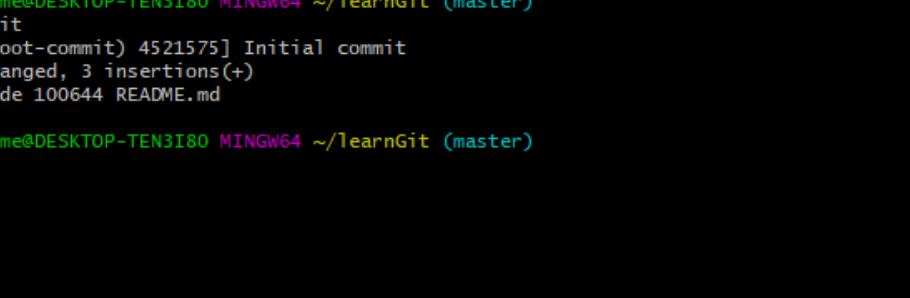
변경된 file 내용을 확인해보기

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git diff
diff --git a/README.md b/README.md
index 43b3580..8c8e330 100644
--- a/README.md
+++ b/README.md
@@ -1,3 +1,5 @@
 # learn git

-hello world!
\ No newline at end of file
+hello world!
+
+this is git repo to learn
\ No newline at end of file

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$
```


버전 commit 하기



The screenshot shows a Windows terminal window with the title bar "MINGW64:/c:/Users/VanDarkHolme/learnGit". The terminal content is as follows:

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git commit
[master (root-commit) 4521575] Initial commit
1 file changed, 3 insertions(+)
create mode 100644 README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$
```

```
MINGW64:/c/Users/VanDarkHolme/learnGit
```

```
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
#  
# On branch master  
#  
Initial commit  
#  
# Changes to be committed:  
#   new file:   README.md  
#  
# Changes not staged for commit:  
#   modified:   README.md  
#  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
  
C:/Users/VanDarkHolme/learnGit/.git/COMMIT_EDITMSG[+] [unix] (09:26 13/11/2019) 7,1 ㄴ ≡
```

commit 내역 확인하기

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log -p -1
commit 452157531a8beb00bb1b7b159278eed79db5de23 (HEAD -> master)
Author: brain-hack <deep-learning@kakao.com>
Date:   Wed Nov 13 09:26:29 2019 +0900

    Initial commit

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..43b3580
--- /dev/null
+++ b/README.md
@@ -0,0 +1,3 @@
+# learn git
+
+hello world!
\ No newline at end of file

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ |
```

좀 이상하지 않나요?

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log -p -1
commit 452157531a8beb00bb1b7b159278eed79db5de23 (HEAD -> master)
Author: brain-hack <deep-learning@kakao.com>
Date:   Wed Nov 13 09:26:29 2019 +0900

    Initial commit

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..43b3580
--- /dev/null
+++ b/README.md
@@ -0,0 +1,3 @@
+# learn git
+
+hello world!
\ No newline at end of file

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ |
```

```
README.md - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말

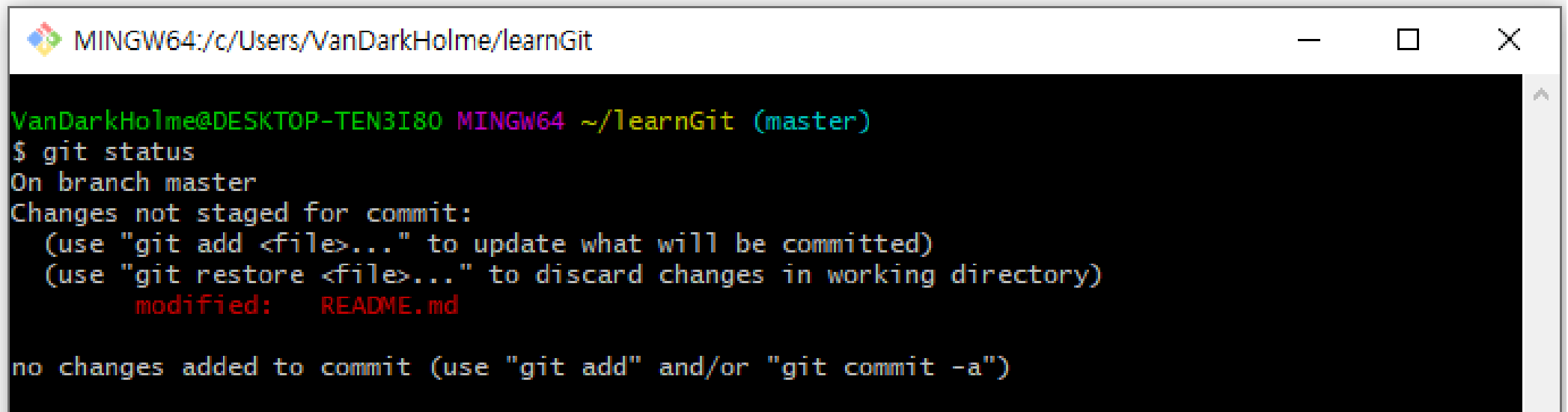
# learn git

hello world!

this is git repo to learn|

Windows Ln 5, Col 100%
```

현재 git 상태를 확인해보죠

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/VanDarkHolme/learnGit'. The window shows the output of the 'git status' command. The prompt is 'VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)'. The output indicates that the user is on the 'master' branch and that there are changes not staged for commit, specifically 'README.md' which has been modified. It provides instructions on how to stage the changes using 'git add' or discard them using 'git restore'.

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

commit을 수정해봅시다

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git add README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git commit --amend
[master abd1827] Initial commit
Date: Wed Nov 13 09:26:29 2019 +0900
1 file changed, 5 insertions(+)
create mode 100644 README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log
commit abd182712e7b732882d6f0b6b80a362b34d63613 (HEAD -> master)
Author: brain-hack <deep-learning@kakao.com>
Date:   Wed Nov 13 09:26:29 2019 +0900

    Initial commit
```

수정된 commit 내용을 볼까요

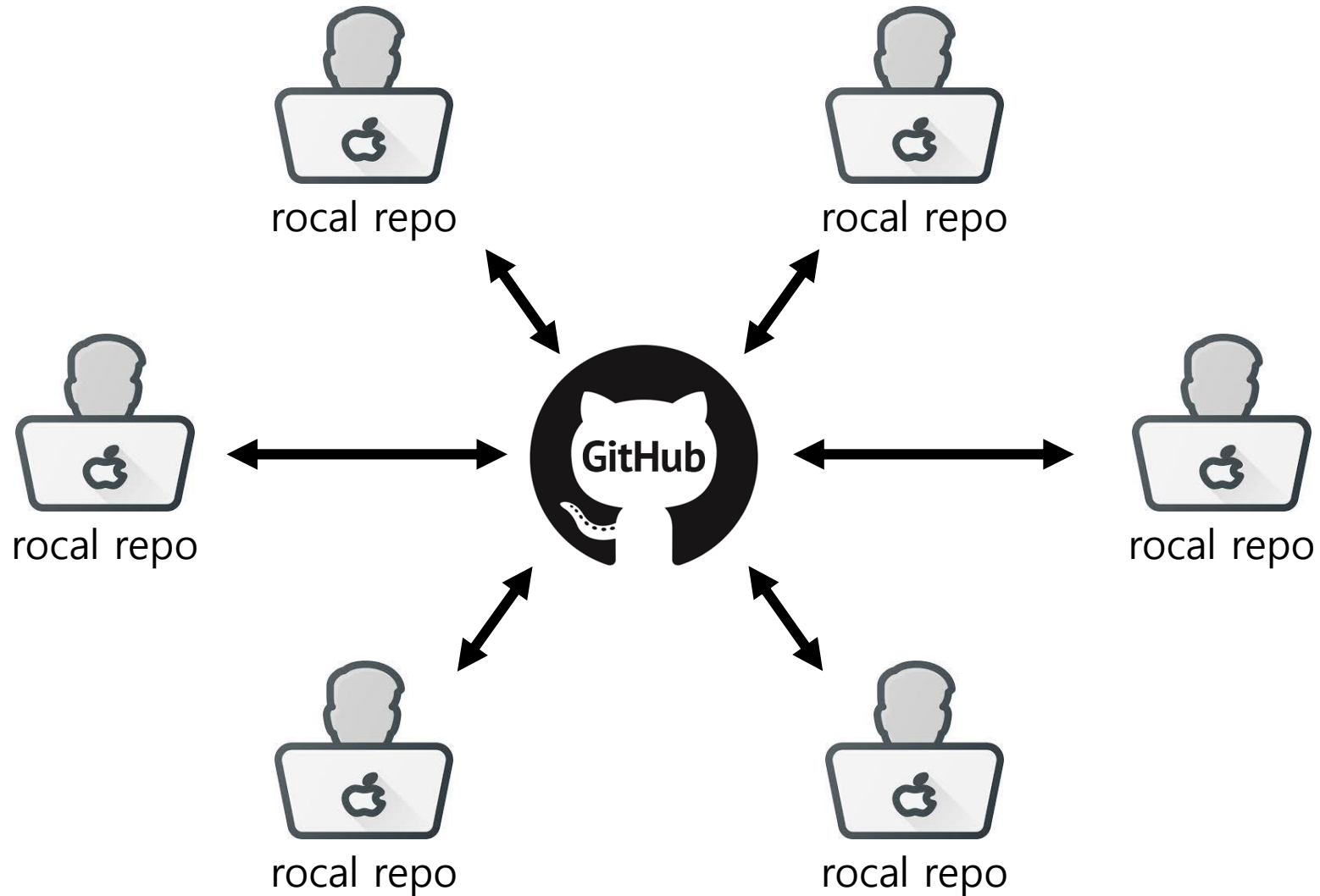
```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log -p -1
commit abd182712e7b732882d6f0b6b80a362b34d63613 (HEAD -> master)
Author: brain-hack <deep-learning@kakao.com>
Date:   Wed Nov 13 09:26:29 2019 +0900

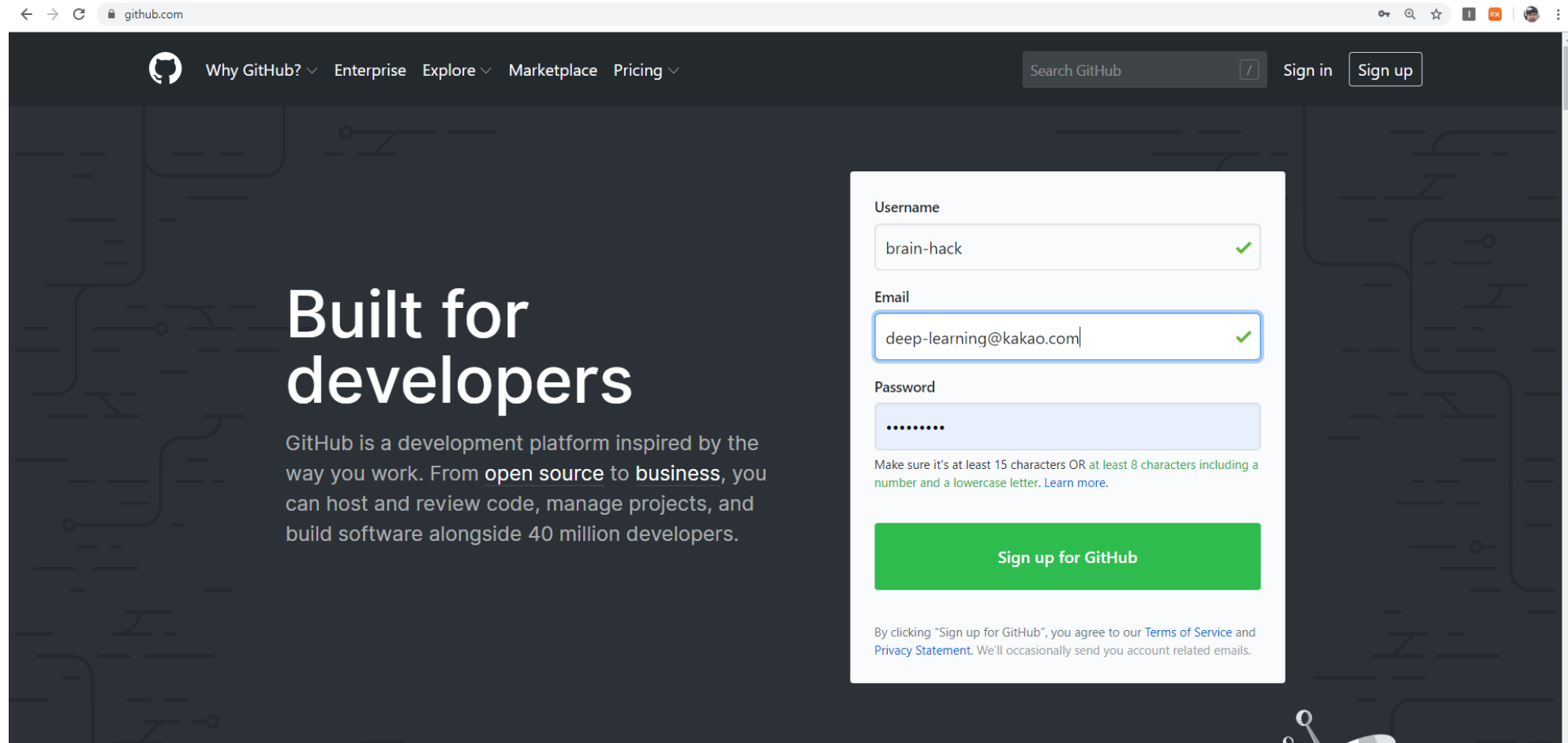
    Initial commit

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..8c8e330
--- /dev/null
+++ b/README.md
@@ -0,0 +1,5 @@
+# learn git
+
+hello world!
+
+this is git repo to learn
\ No newline at end of file
```

remote repository : github



github 가입

A screenshot of the GitHub website's sign-up page. The browser's address bar shows 'github.com'. The navigation bar includes links for 'Why GitHub?', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing', along with a search bar and 'Sign in' and 'Sign up' buttons. The main content area features the text 'Built for developers' and a description of GitHub as a development platform. On the right, a sign-up form is displayed with fields for 'Username' (filled with 'brain-hack'), 'Email' (filled with 'deep-learning@kakao.com'), and 'Password' (masked with dots). A green 'Sign up for GitHub' button is at the bottom of the form, followed by a disclaimer about terms of service and privacy policy.

← → ↻ github.com 🔍 ☆ 1 9%

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

brain-hack ✓

Email

deep-learning@kakao.com ✓

Password

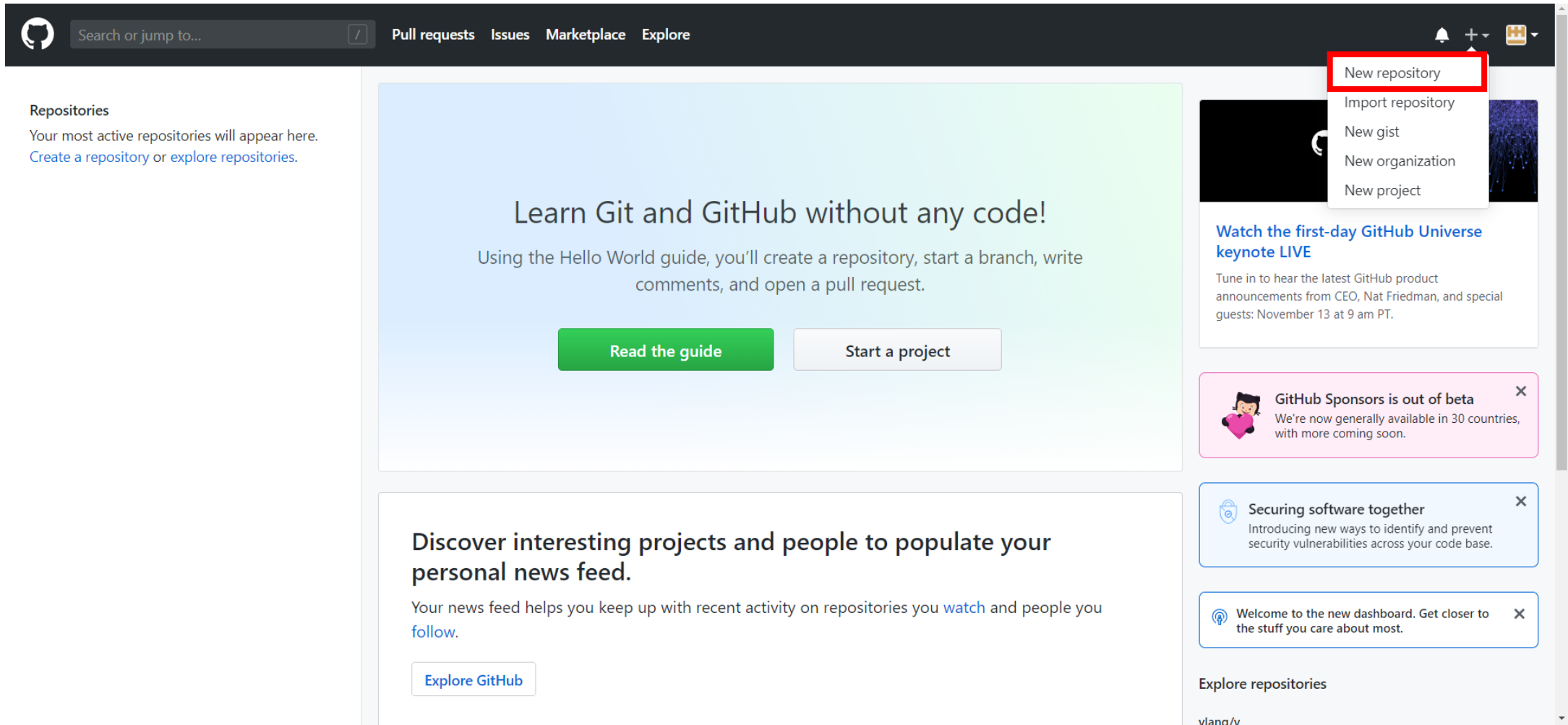
.....

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

새 github repo 만들기



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, Marketplace, and Explore. On the right side of the navigation bar, there are icons for notifications, a plus sign, and a user profile. A dropdown menu is open from the plus sign, showing options: New repository (highlighted with a red box), Import repository, New gist, New organization, and New project.

Repositories
Your most active repositories will appear here.
[Create a repository](#) or [explore repositories](#).

Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

Discover interesting projects and people to populate your personal news feed.
Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you [follow](#).

[Explore GitHub](#)

Watch the first-day GitHub Universe keynote LIVE
Tune in to hear the latest GitHub product announcements from CEO, Nat Friedman, and special guests: November 13 at 9 am PT.

GitHub Sponsors is out of beta
We're now generally available in 30 countries, with more coming soon.

Securing software together
Introducing new ways to identify and prevent security vulnerabilities across your code base.

Welcome to the new dashboard. Get closer to the stuff you care about most.


Explore repositories
vlang/v

새 github repo 만들기

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

 brain-hack ▾

Repository name *

/ learnGit1 ✓

Great repository names are short and memorable. Need inspiration? How about [bug-free-dollop](#)?

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

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

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.


Add .gitignore: **None** ▾


Add a license: **None** ▾

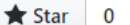


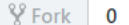
Create repository

새 github repo 만들기

 brain-hack / learnGit1

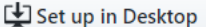
 Unwatch ▾ 1

 ★ Star 0

 Fork 0


[↩ Code](#) [! Issues 0](#) [🔗 Pull requests 0](#) [📁 Projects 0](#) [📖 Wiki](#) [🛡 Security](#) [📊 Insights](#) [⚙ Settings](#)

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 "# learnGit1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/brain-hack/learnGit1.git
git push -u origin master
```



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

```
git remote add origin https://github.com/brain-hack/learnGit1.git
git push -u origin master
```



새 github repo 만들기(2)


Create a new repository

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

[Import a repository.](#)

Owner

Repository name *

 brain-hack ▾

/ learnGit2 ✓

Great repository names are short and memorable. Need inspiration? How about **fuzzy-doodle**?

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

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

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: Python ▾

Add a license: Apache License 2.0 ▾



Create repository

새 github repo 만들기(2)

brain-hack / learnGit2

Unwatch

1

Star

0

Fork

0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

brain-hack Initial commit

Latest commit 752cb4c now

.gitignore	Initial commit	now
LICENSE	Initial commit	now
README.md	Initial commit	now

README.md

learnGit2

local과 github repo 연결



brain-hack / learnGit1

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

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

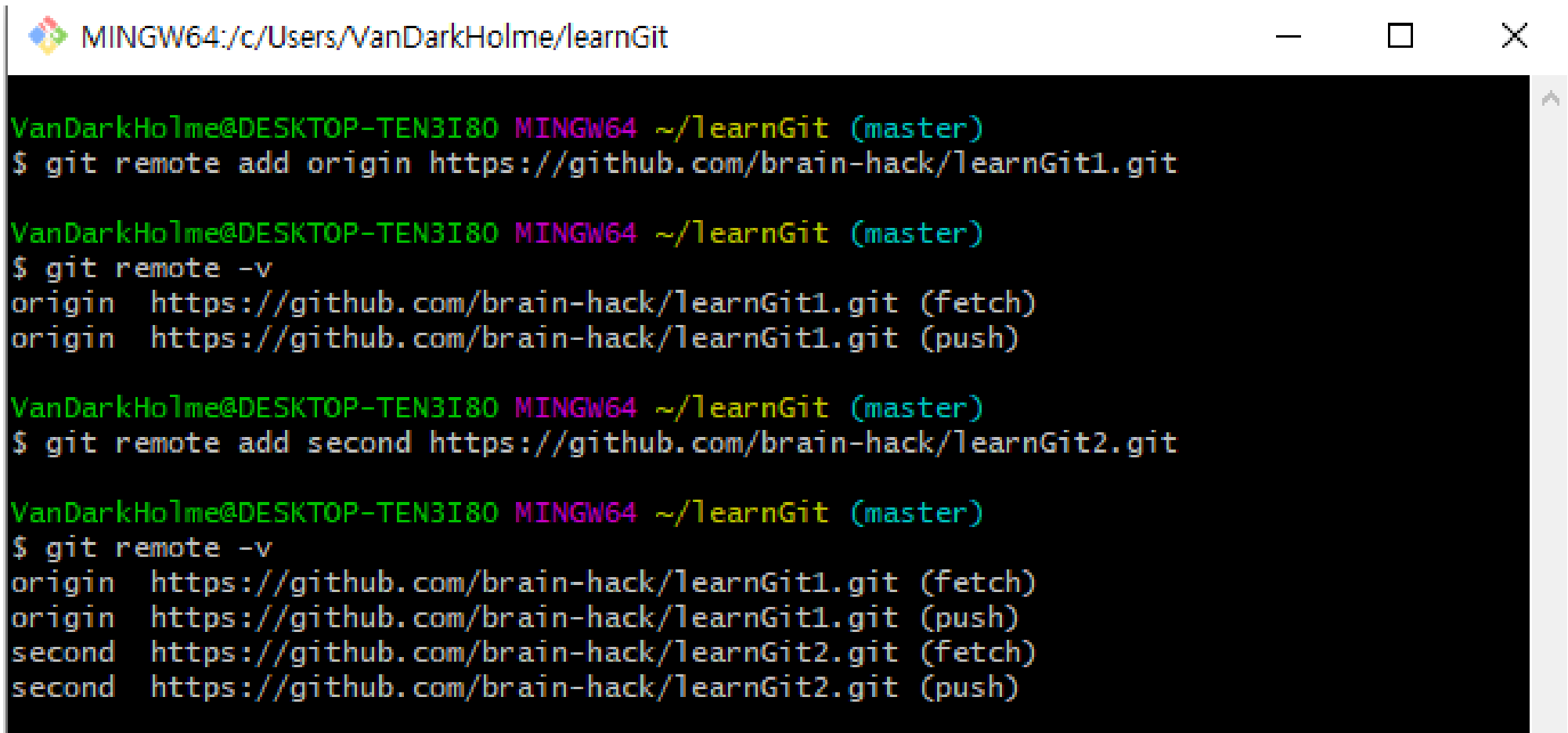
Set up in Desktop or **HTTPS** SSH `https://github.com/brain-hack/learnGit1.git`

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 "# learnGit1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/brain-hack/learnGit1.git
git push -u origin master
```

local과 github repo 연결



```
MINGW64:/c/Users/VanDarkHolme/learnGit

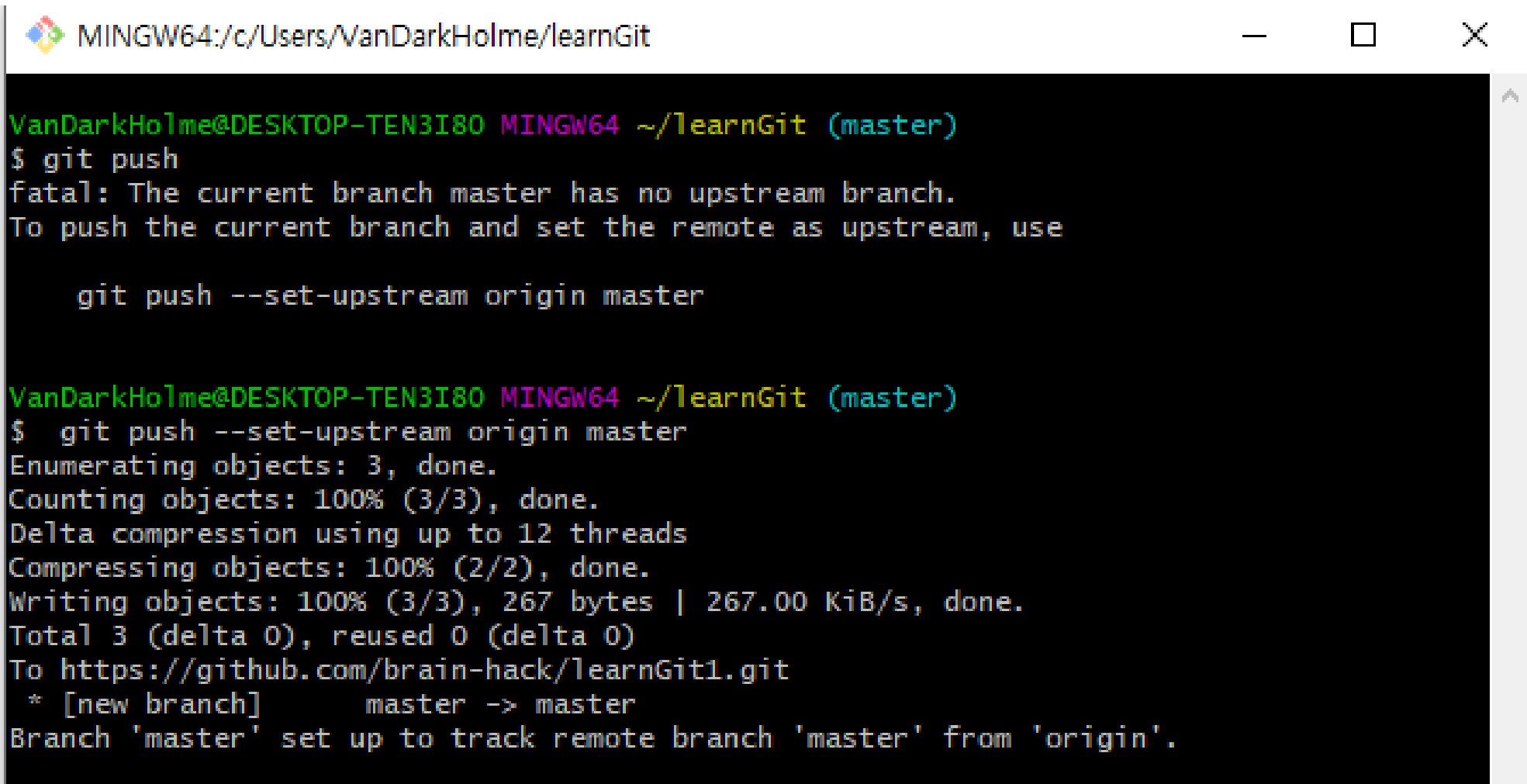
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git remote add origin https://github.com/brain-hack/learnGit1.git

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git remote -v
origin    https://github.com/brain-hack/learnGit1.git (fetch)
origin    https://github.com/brain-hack/learnGit1.git (push)

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git remote add second https://github.com/brain-hack/learnGit2.git

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git remote -v
origin    https://github.com/brain-hack/learnGit1.git (fetch)
origin    https://github.com/brain-hack/learnGit1.git (push)
second    https://github.com/brain-hack/learnGit2.git (fetch)
second    https://github.com/brain-hack/learnGit2.git (push)
```

github에 local commits 업로드



```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git push --set-upstream origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 267 bytes | 267.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/brain-hack/learnGit1.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```


여기서 문제

- 과연 push한 local commit은 어디로 업로드가 되었을까요?


Repo1

github.com/username/learnGit1

Repo2

github.com/username/learnGit2

github repo 확인

 brain-hack / learnGit1

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

No description, website, or topics provided.

[Manage topics](#)

1 commit

1 branch

0 releases

1 contributor

Branch: master


New pull request

Create new file


Upload files

Find file

Clone or download


 brain-hack Initial commit

Latest commit abd1827 7 hours ago

 README.md

Initial commit

6 hours ago

 README.md

learn git

hello world!

this is git repo to learn

local에 새 저장소를 만들어보겠습니다

근데 이제 clone을 곁들인

The screenshot shows a GitHub repository page for 'brain-hack / learnGit1'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted, and a modal is open showing the HTTPS clone URL: `https://github.com/brain-hack/learnGit1.` The modal also includes options to 'Open in Desktop' or 'Download ZIP'. The repository content shows a README.md file with the text 'learn git', 'hello world!', and 'this is git repo to learn'.

brain-hack / learnGit1

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

`https://github.com/brain-hack/learnGit1.`

Open in Desktop Download ZIP

brain-hack Initial commit

README.md Initial commit

README.md

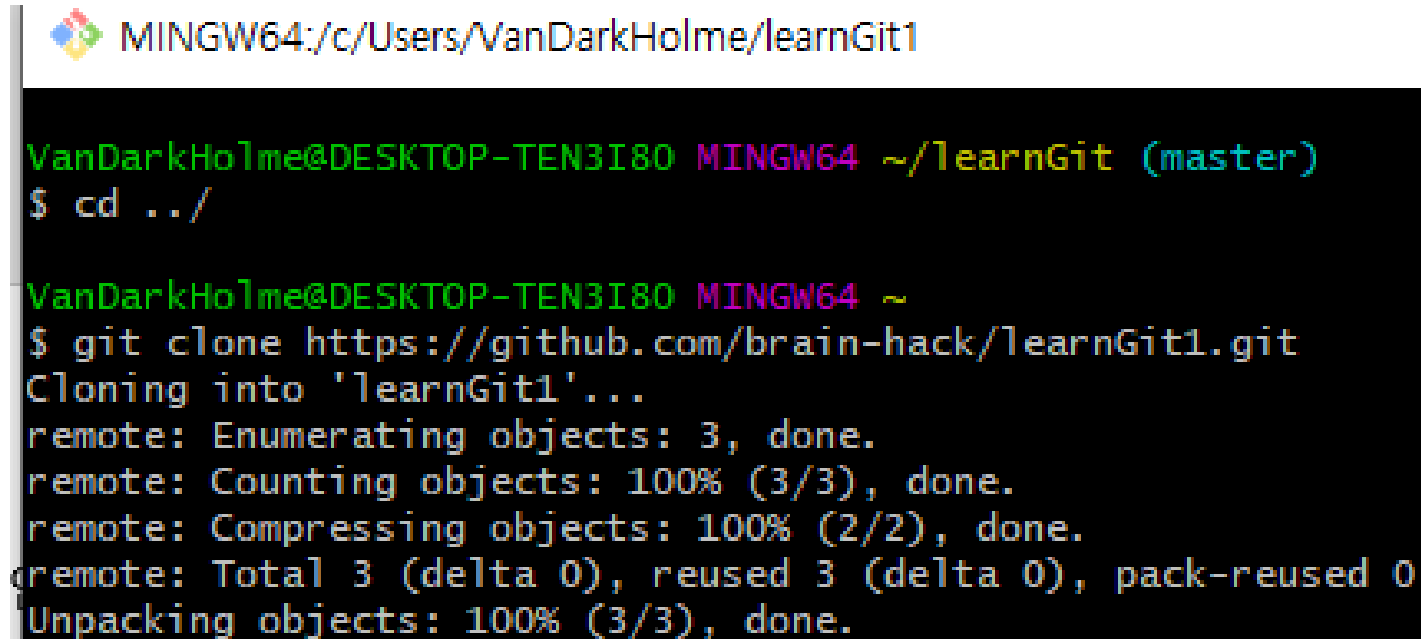
learn git

hello world!

this is git repo to learn

local에 새 저장소를 만들어보겠습니다

근데 이제 clone을 곁들인



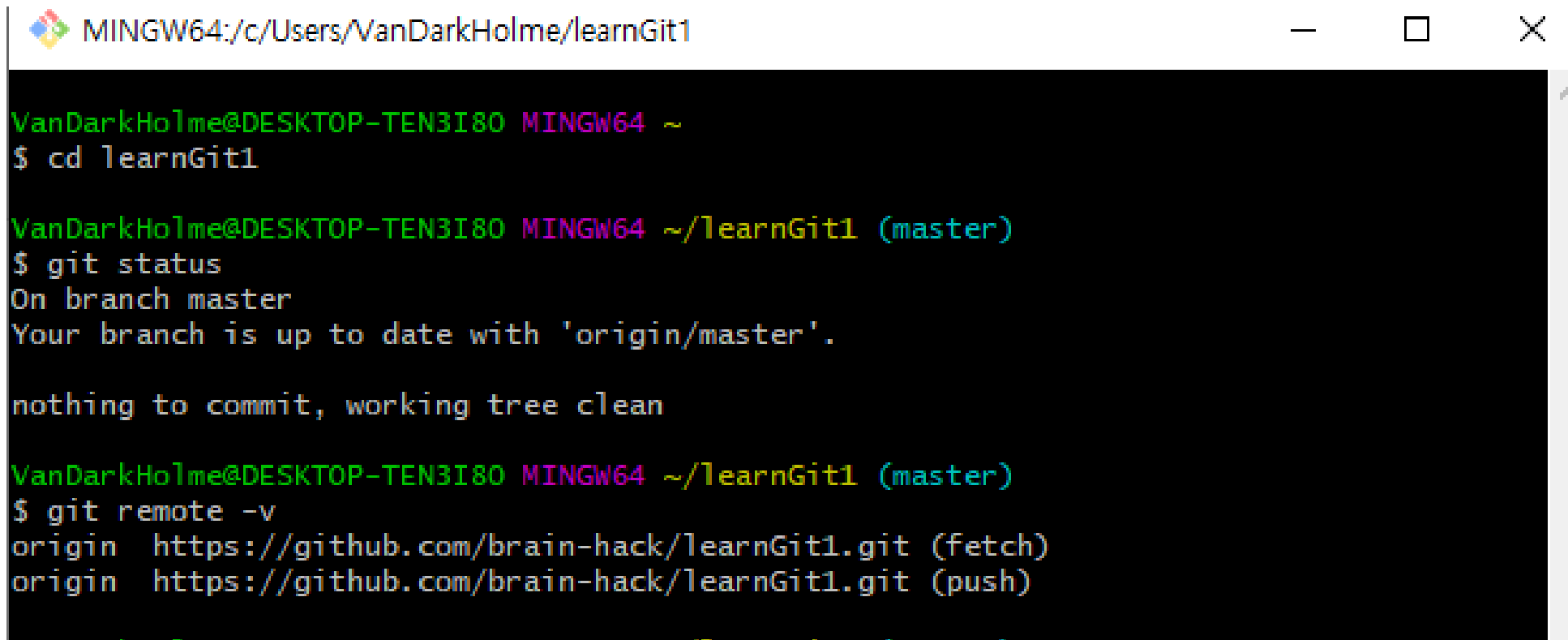
A screenshot of a Windows terminal window titled "MINGW64:/c/Users/VanDarkHolme/learnGit1". The window shows a user named VanDarkHolme@DESKTOP-TEN3I80 in a MINGW64 environment. The user is currently in the ~/learnGit (master) directory. They execute the command `$ cd ../` to move to the parent directory. Then, they execute `$ git clone https://github.com/brain-hack/learnGit1.git`. The terminal output shows the cloning process: "Cloning into 'learnGit1'...", "remote: Enumerating objects: 3, done.", "remote: Counting objects: 100% (3/3), done.", "remote: Compressing objects: 100% (2/2), done.", "remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0", and "Unpacking objects: 100% (3/3), done.".

```
MINGW64:/c/Users/VanDarkHolme/learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ cd ../

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ git clone https://github.com/brain-hack/learnGit1.git
Cloning into 'learnGit1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

clone된 repo확인



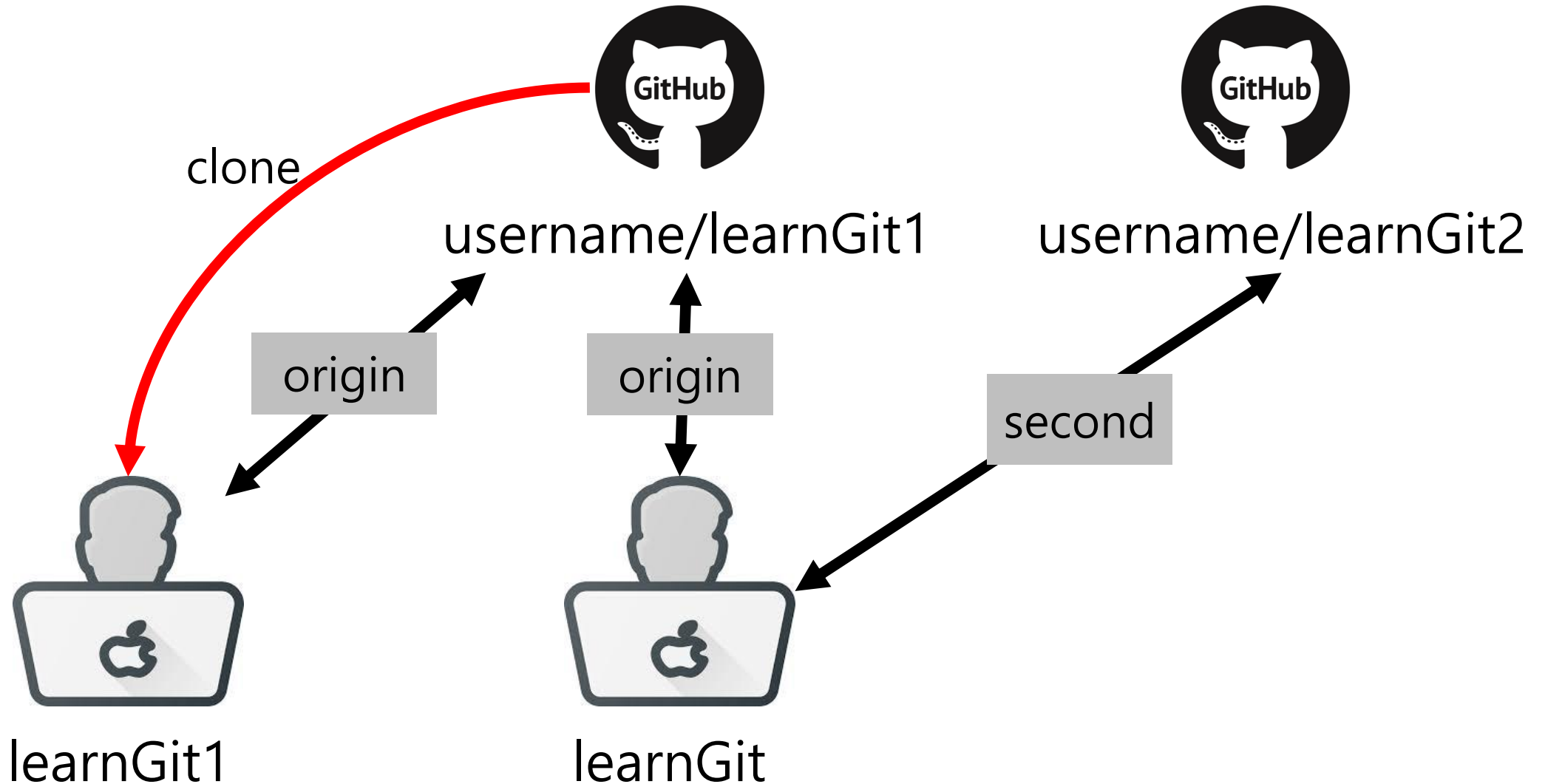
```
MINGW64:/c/Users/VanDarkHolme/learnGit1
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ cd learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git remote -v
origin  https://github.com/brain-hack/learnGit1.git (fetch)
origin  https://github.com/brain-hack/learnGit1.git (push)
```

지금 현재 local과 remote의 관계



local repo : learnGit1에서 작업하기

MINGW64:/c/Users/VanDarkHolme/learnGit1

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ notepad hello.py

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ ls
hello.py  README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.py

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

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git add hello.py

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git commit -m "add python code : hello world"
[master 6079006] add python code : hello world
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 hello.py
```

hello - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

```
def print_message():
    message = "world"
    print("hello {}".format(message))

if __name__ == "__main__":
    print_message()
```

Windows (CR) Ln 1, Col 1 100%

local repo : learnGit1 에서 push

MINGW64:/c/Users/VanDarkHolme/learnGit1

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 292 bytes | 292.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/brain-hack/learnGit1.git
abd1827..6079006 master -> master
```

No description, website, or topics provided.


[Manage topics](#)

🕒 2 commits

🌿 1 branch

Branch: master ▾

[New pull request](#)


 brain-hack add python code : hello world

 [README.md](#)

Initial commit

 [hello.py](#)

add python code : hello world

 [README.md](#)

learn git

hello world!

this is git repo to learn

local repo : learnGit에서 작업하기

MINGW64:/c/Users/VanDarkHolme/learnGit

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ notepad README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ notepad .gitignore

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git add README.md .gitignore

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git commit -m "update readme and add gitignore"
[master b990c79] update readme and add gitignore
2 files changed, 109 insertions(+), 1 deletion(-)
create mode 100644 .gitignore
```

README - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

hello world!

this is git repo to learn

this repo have these code:

- python hello world

Windows Ln 9, Col 100%

.gitignore - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

Rope project settings

.ropeproject

mkdocs documentation

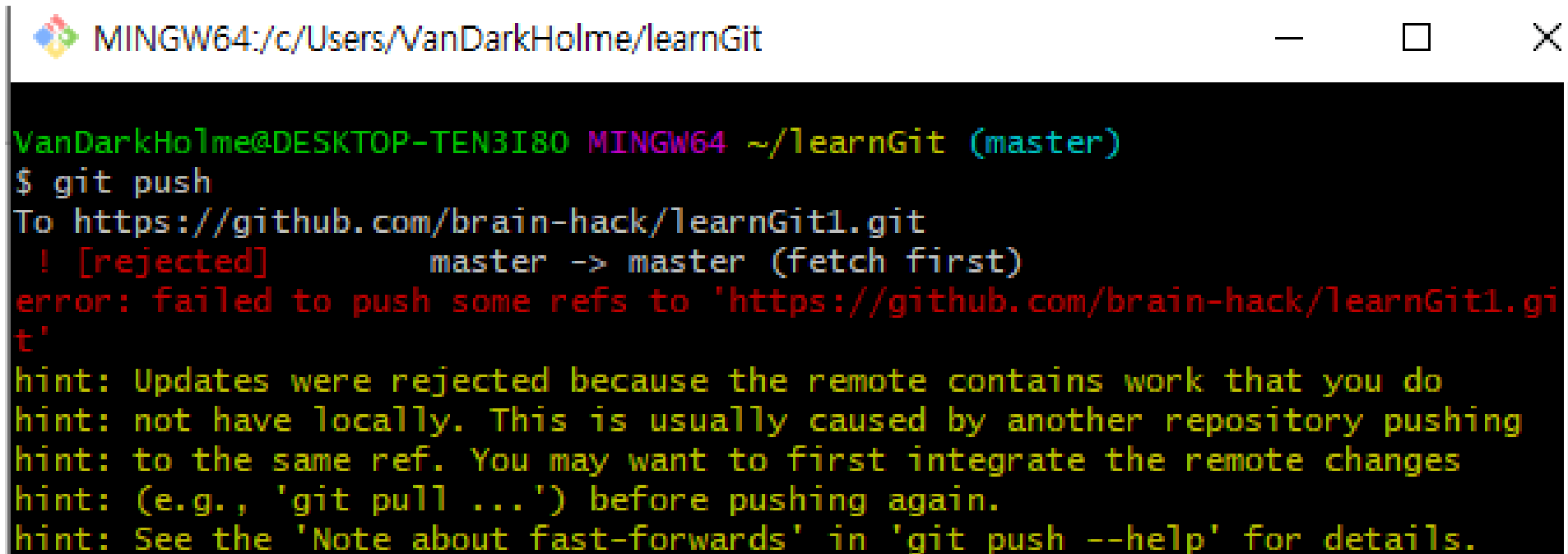
/site

mypy

.mypy_cache/

Windows Ln 98, Col 100%

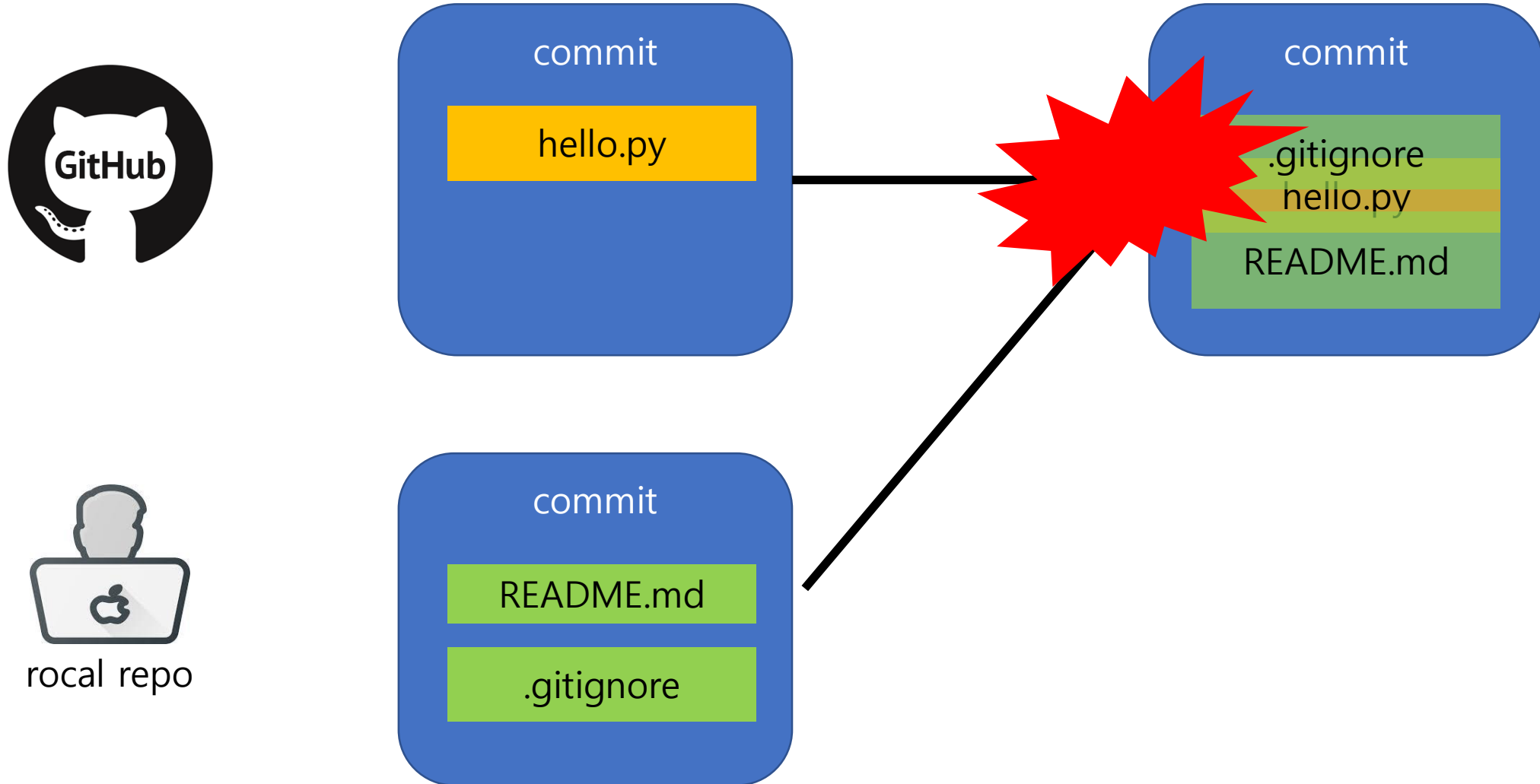
push 문제 발생

A screenshot of a Windows terminal window with a black background and colored text. The title bar at the top shows the MINGW64 logo and the path 'MINGW64:/c/Users/VanDarkHolme/learnGit'. The terminal content shows a user attempting to push to a remote repository, which results in a rejection because the remote has newer commits. The user is prompted to fetch the remote changes first.

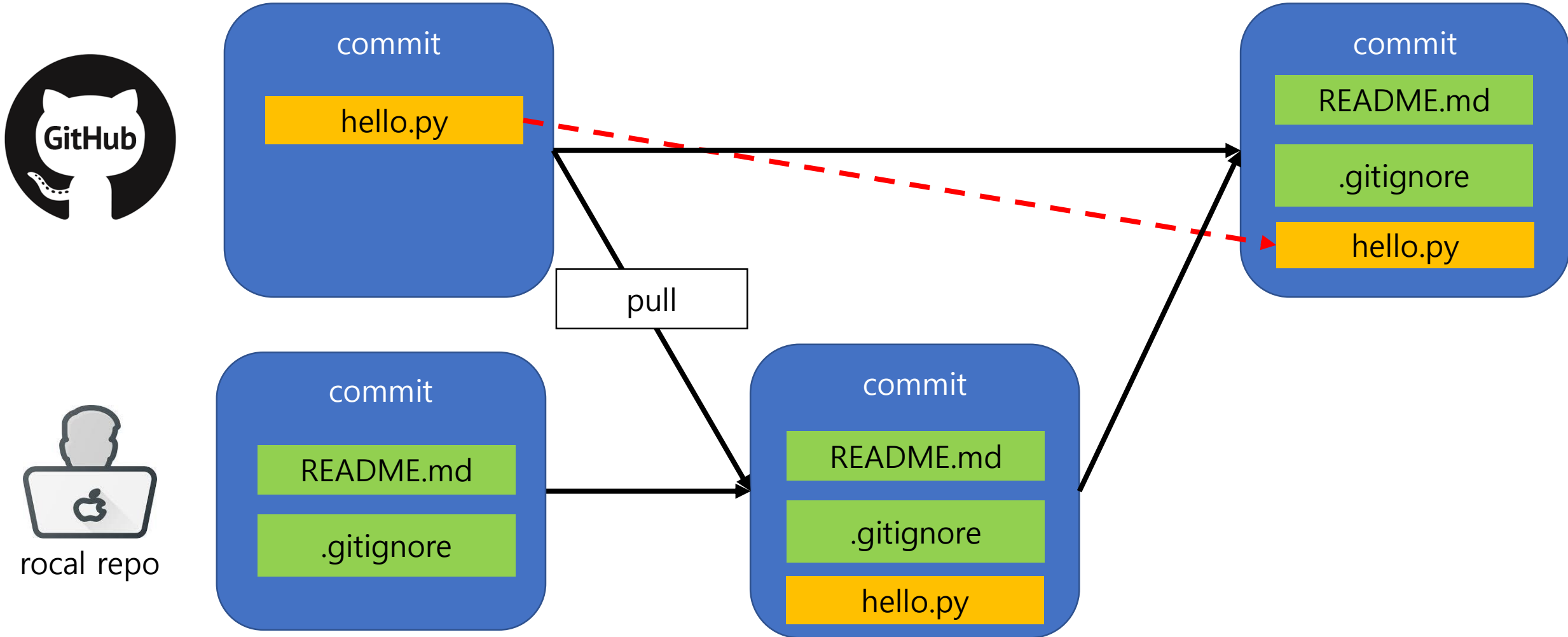
```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git push
To https://github.com/brain-hack/learnGit1.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/brain-hack/learnGit1.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

왜 문제가 발생했을까요?



어떻게 해결할까?



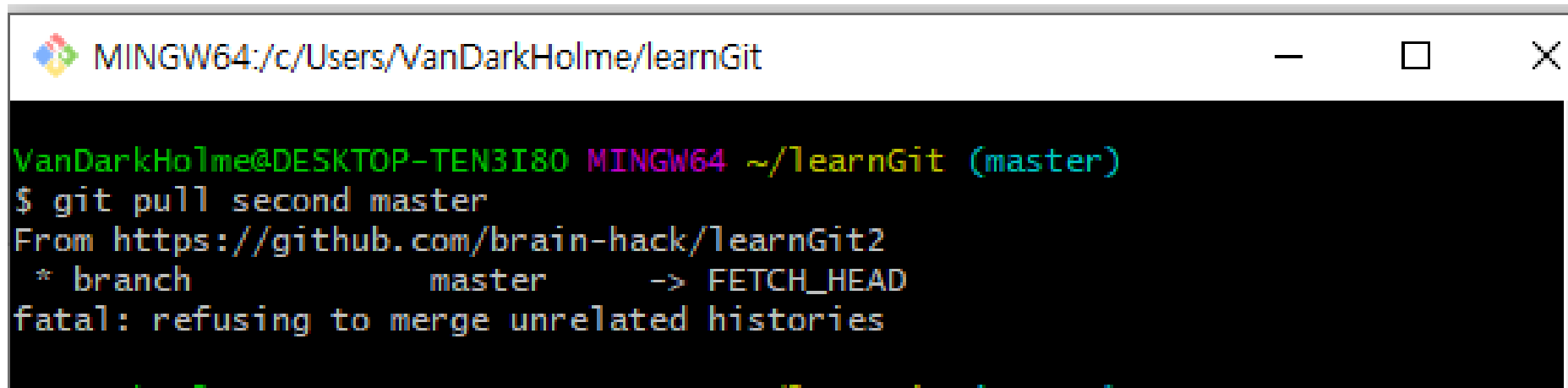
git pull을 통해 remote내용 download

MINGW64:/c/Users/VanDarkHolme/learnGit

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/brain-hack/learnGit1
   abd1827..6079006  master    -> origin/master
Merge made by the 'recursive' strategy.
 hello.py | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello.py

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ ls
hello.py README.md
```

하지만 일이 순탄치 않을 때가....



```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git pull second master
From https://github.com/brain-hack/learnGit2
 * branch          master      -> FETCH_HEAD
fatal: refusing to merge unrelated histories
```

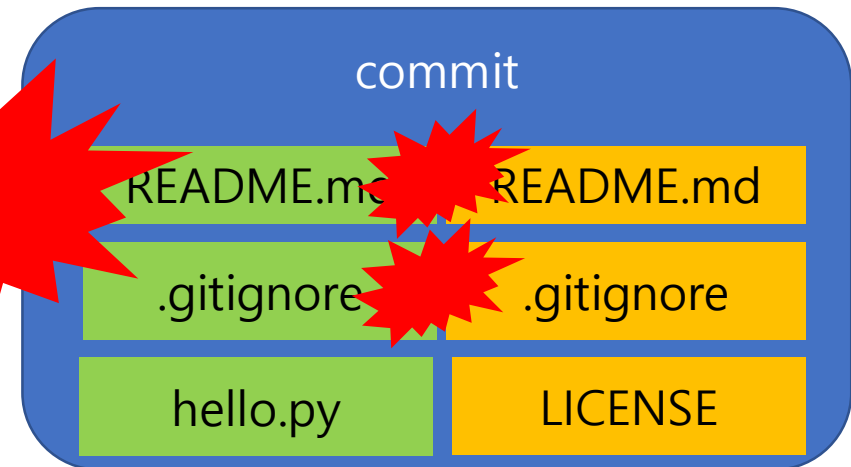
왜 문제가 발생했을까요?



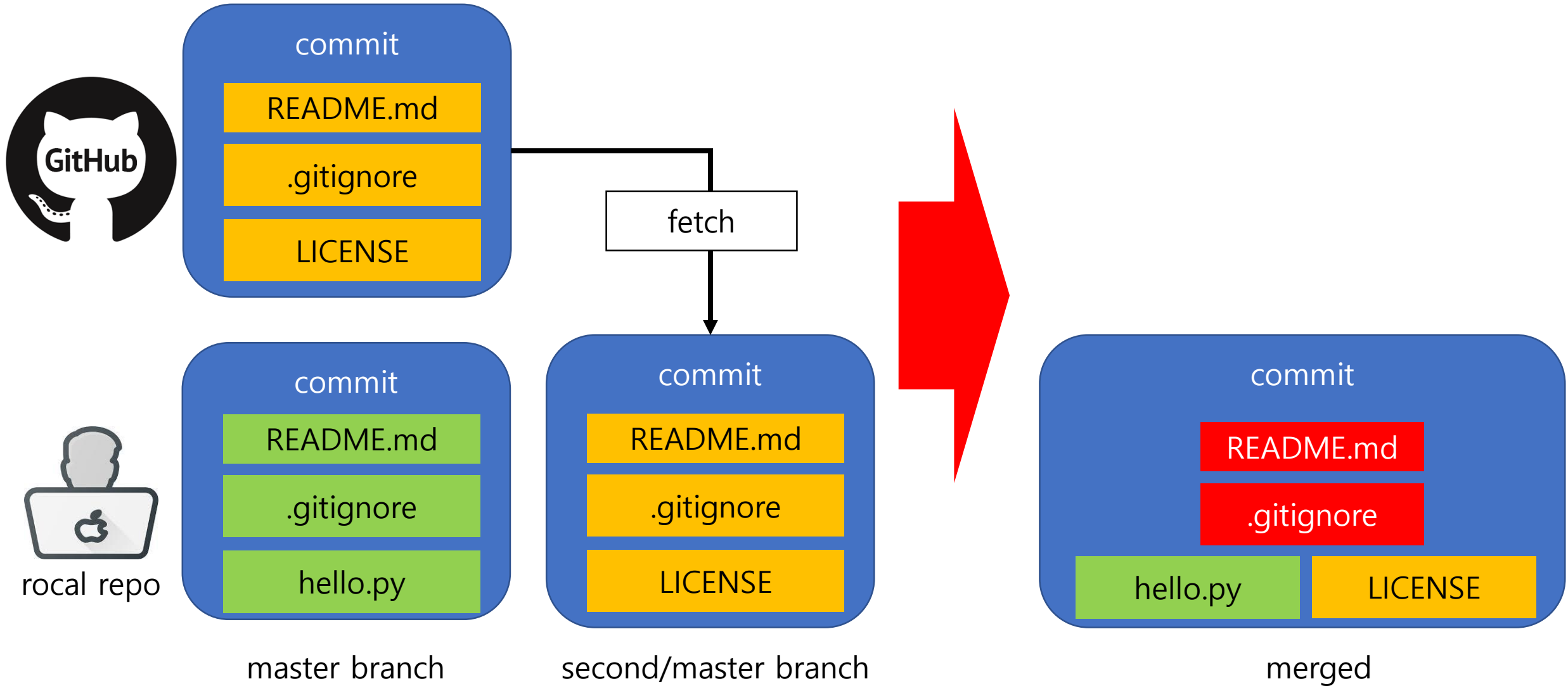
pull



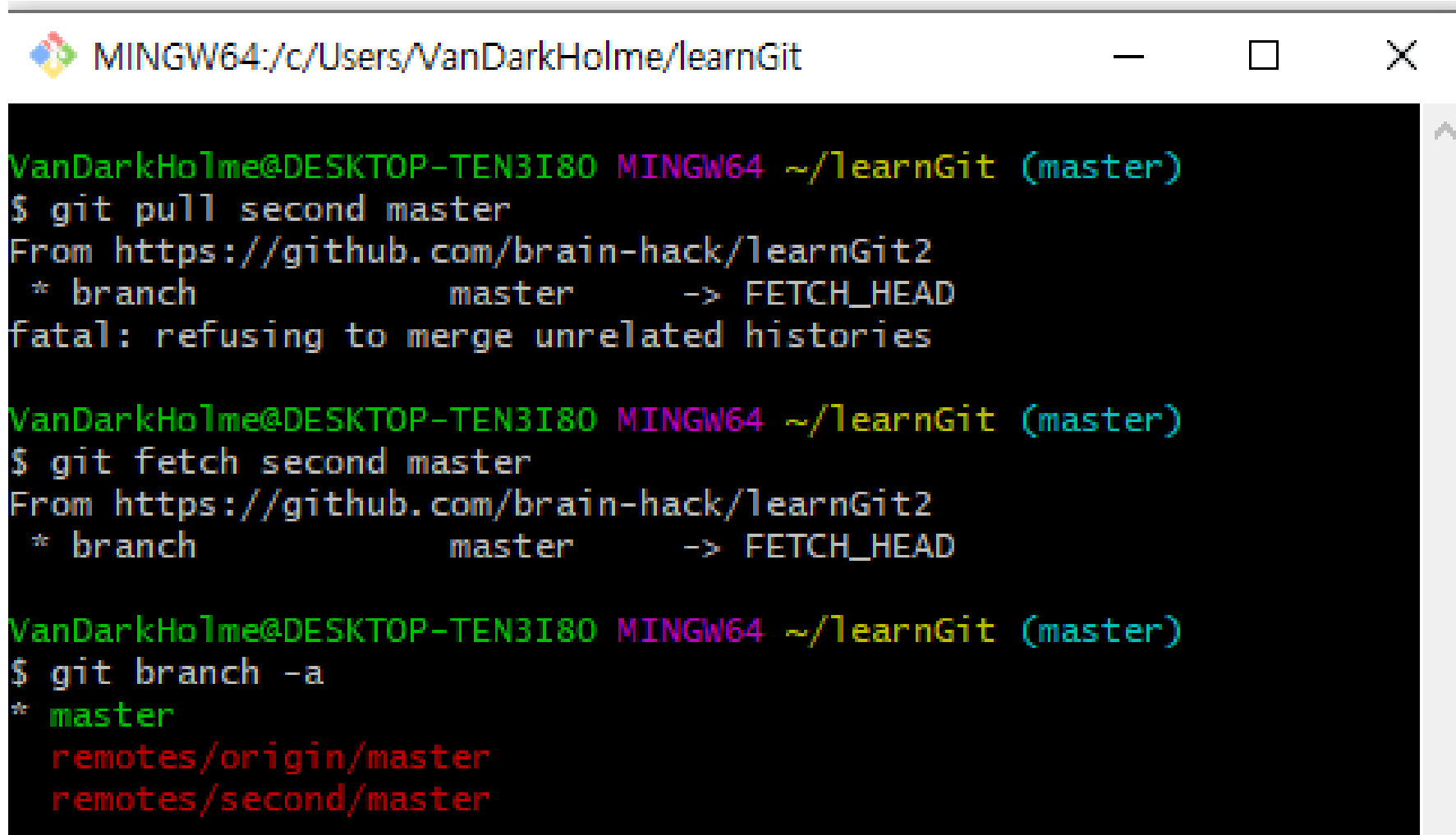
rocal repo



fetch후 수동으로 합친 후 merge



git fetch를 해봅시다



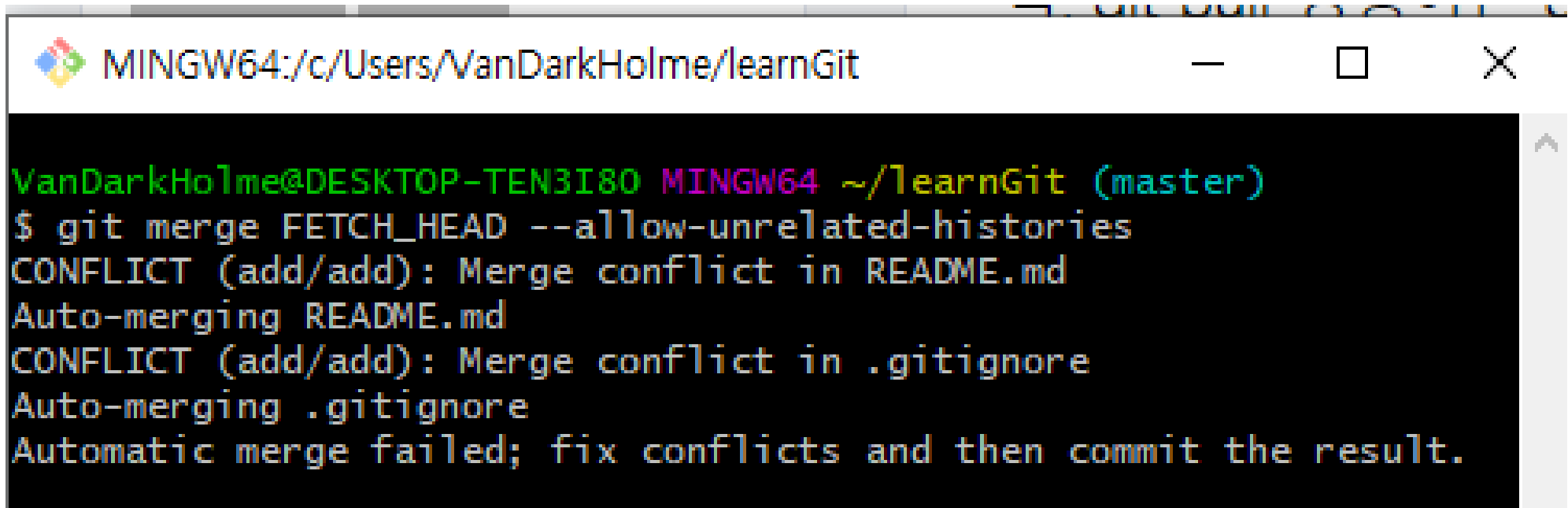
```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git pull second master
From https://github.com/brain-hack/learnGit2
 * branch          master      -> FETCH_HEAD
fatal: refusing to merge unrelated histories

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git fetch second master
From https://github.com/brain-hack/learnGit2
 * branch          master      -> FETCH_HEAD

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git branch -a
* master
  remotes/origin/master
  remotes/second/master
```

merge를 해볼까요(헬게이트 오픈)



```
MINGW64:/c/Users/VanDarkHolme/learnGit

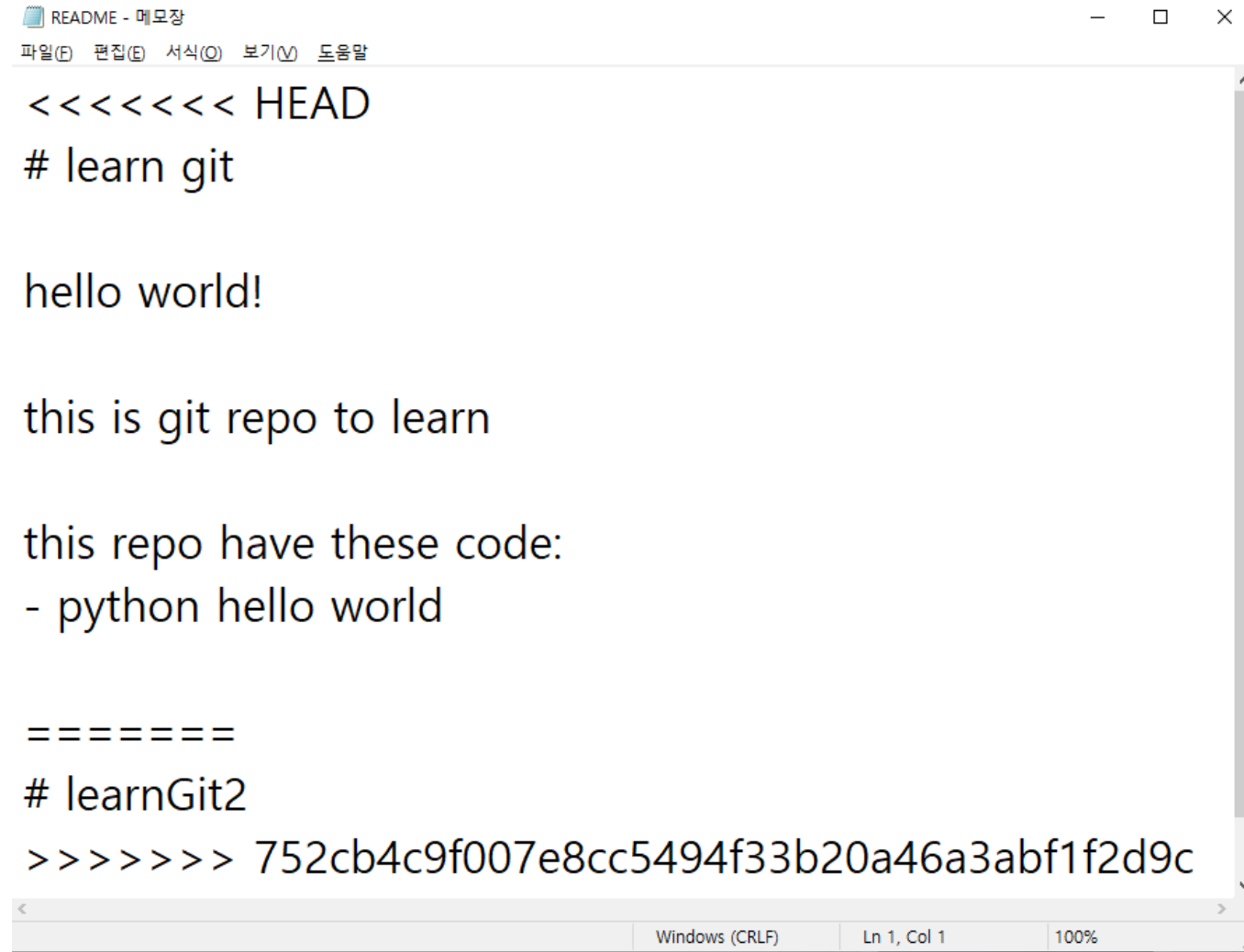
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git merge FETCH_HEAD --allow-unrelated-histories
CONFLICT (add/add): Merge conflict in README.md
Auto-merging README.md
CONFLICT (add/add): Merge conflict in .gitignore
Auto-merging .gitignore
Automatic merge failed; fix conflicts and then commit the result.
```

당연히 auto merge는 실패합니다

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master|MERGING)
$ git diff
diff --cc .gitignore
index e824463,894a44c..0000000
--- a/.gitignore
+++ b/.gitignore
@@@ -1,4 -1,4 +1,8 @@@
++<<<<<<< HEAD
+ # Byte-compiled / optimized / DLL files
++=====
+ # Byte-compiled / optimized / DLL files
++>>>>>>> 752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c
+ __pycache__/
+ *.py[cod]
+ *$py.class
@@@ -101,4 -101,4 +105,8 @@@ venv.bak
+ /site

+ # mypy
- .mypy_cache/
++<<<<<<< HEAD
+ .mypy_cache/
++=====
++ .mypy_cache/
++>>>>>>> 752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c
diff --cc README.md
```

수동으로 합쳐보죠 -README.md



```
README - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
<<<<<<< HEAD
# learn git

hello world!

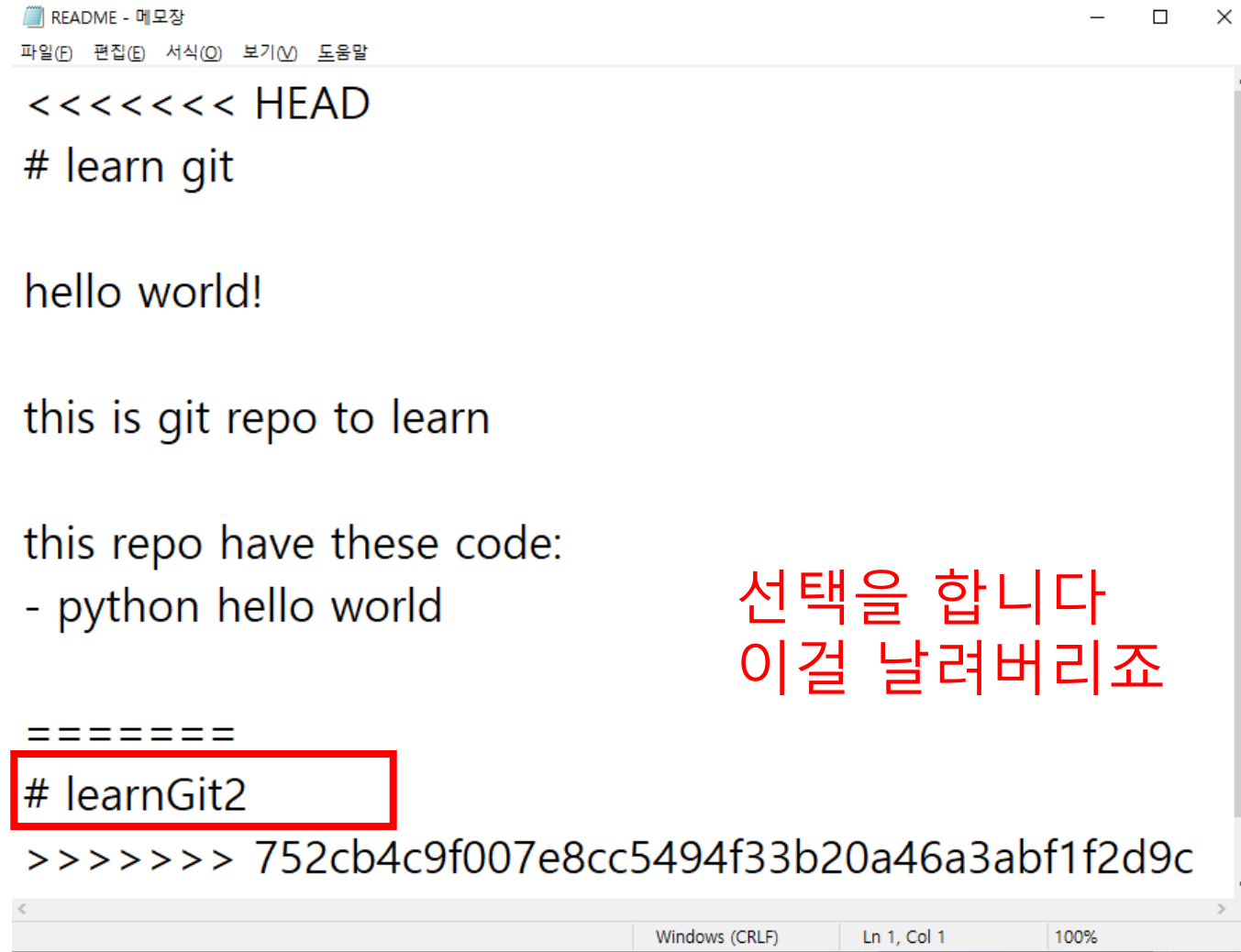
this is git repo to learn

this repo have these code:
- python hello world

=====
# learnGit2
>>>>>>> 752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c
```

Windows (CRLF) Ln 1, Col 1 100%

수동으로 합쳐보죠 -README.md



```
README - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
<<<<<<< HEAD
# learn git

hello world!

this is git repo to learn

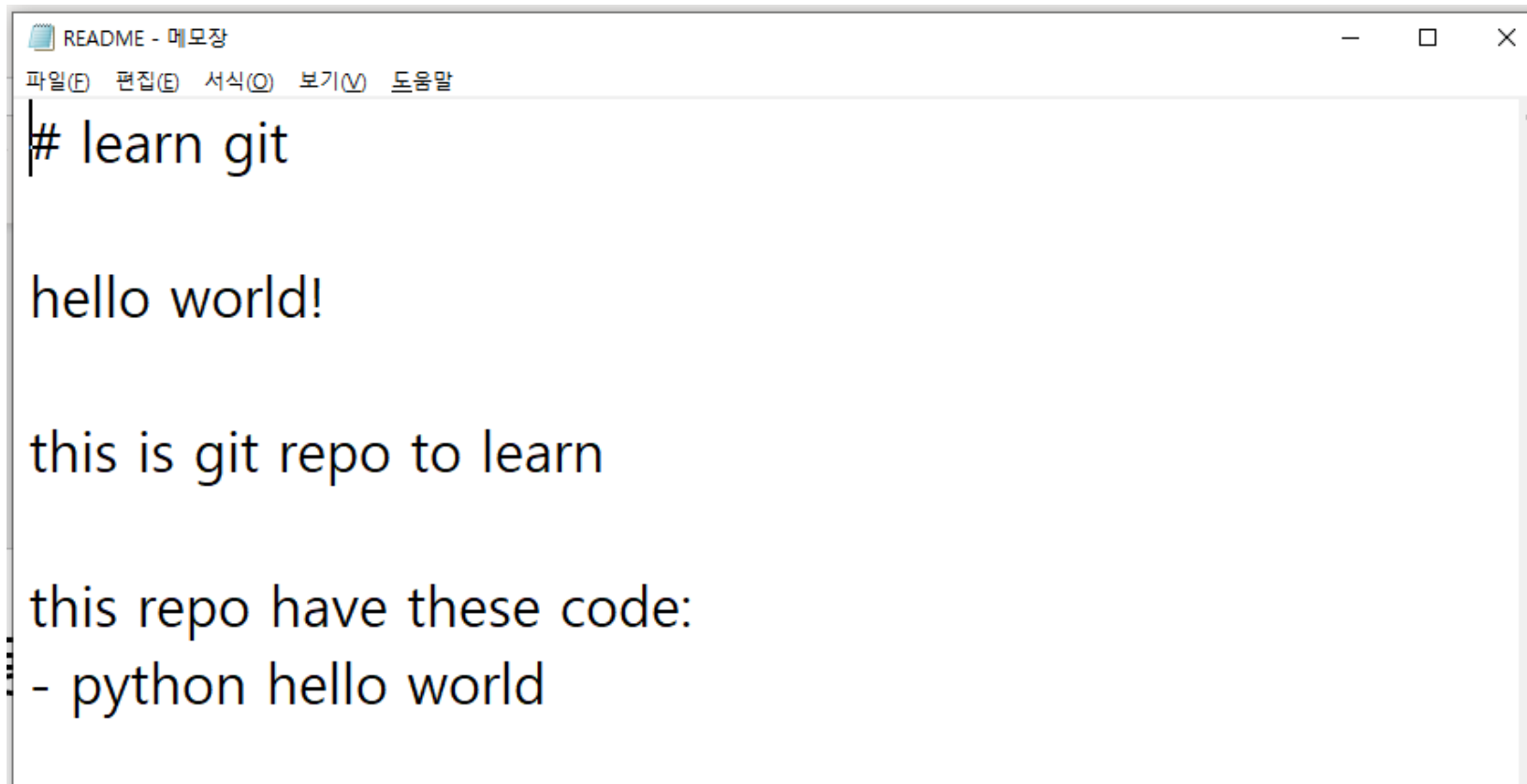
this repo have these code:
- python hello world

=====
# learnGit2
>>>>>> 752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c
```

선택을 합니다
이걸 날려버리죠

Windows (CRLF) Ln 1, Col 1 100%

수동으로 합쳐보죠 -README.md



A screenshot of a Notepad window titled "README - 메모장". The window has a menu bar with "파일(F)", "편집(E)", "서식(O)", "보기(V)", and "도움말". The text inside the window is as follows:

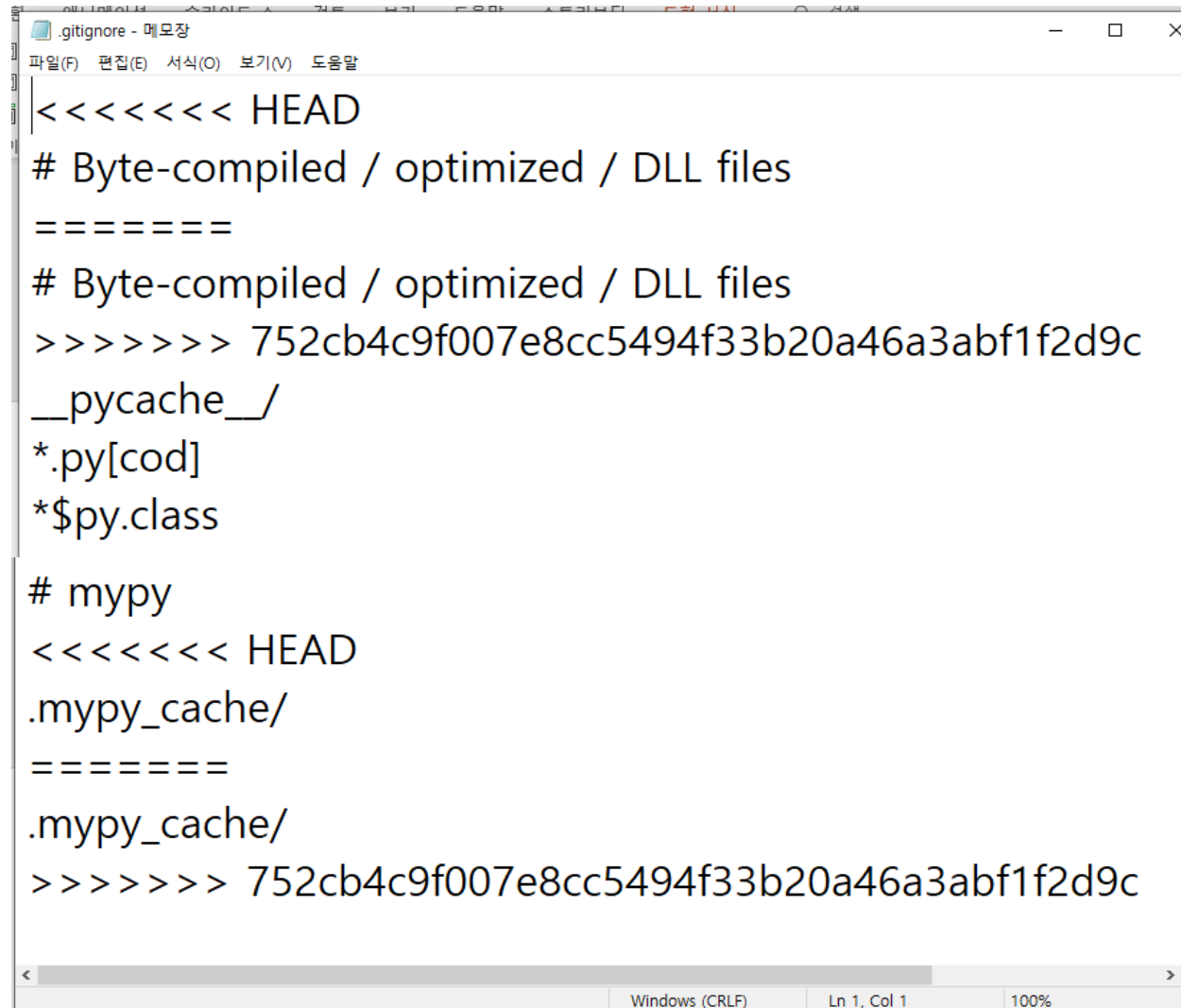
```
# learn git

hello world!

this is git repo to learn

this repo have these code:
- python hello world
```

수동으로 합쳐보죠 - .gitignore

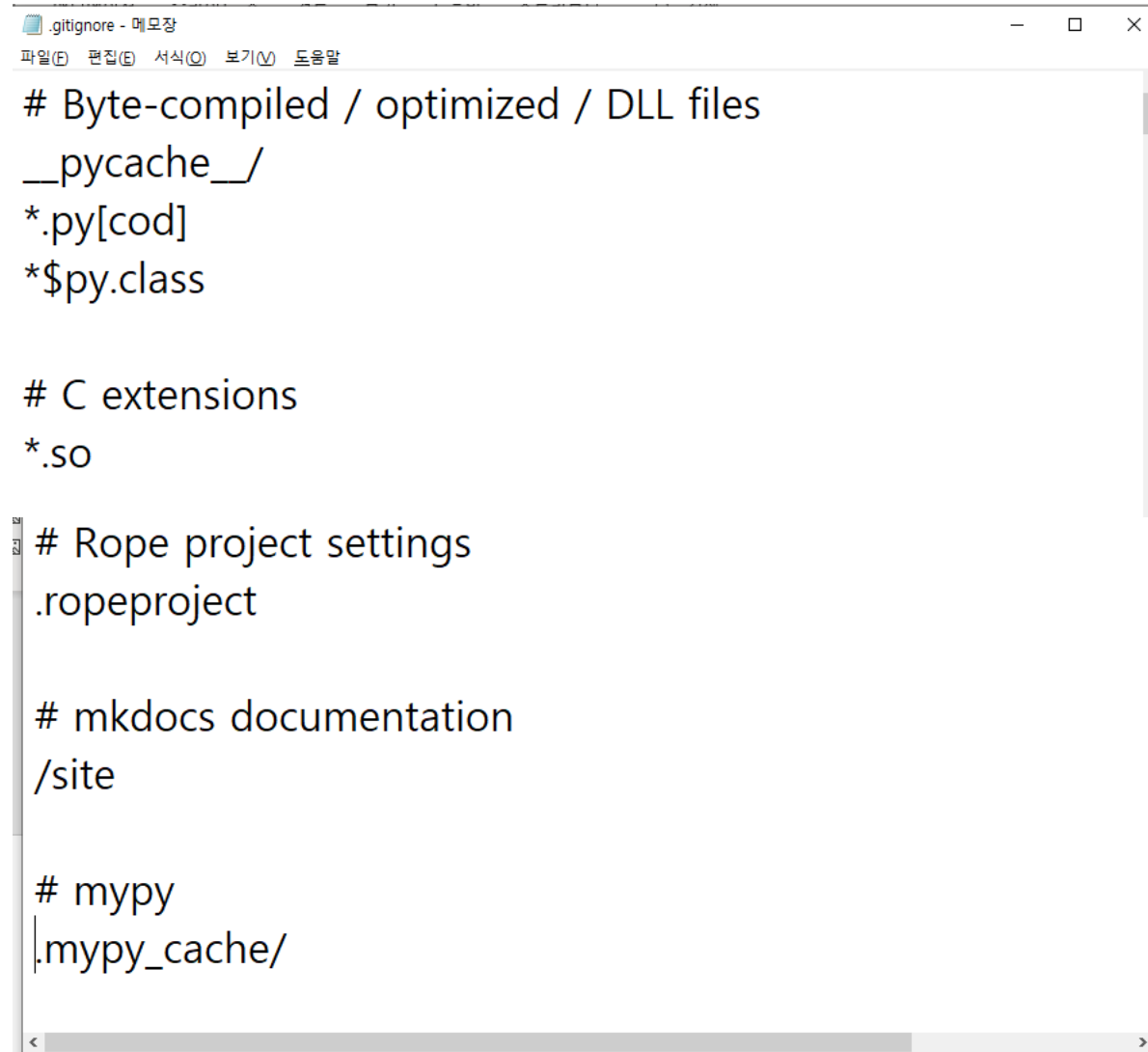
A screenshot of a text editor window titled ".gitignore - 메모장". The window contains a .gitignore file with the following content:

```
<<<<<<< HEAD
# Byte-compiled / optimized / DLL files
=====
# Byte-compiled / optimized / DLL files
>>>>>>> 752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c
__pycache__/
*.py[cod]
*$py.class

# mypy
<<<<<<< HEAD
.mypy_cache/
=====
.mypy_cache/
>>>>>>> 752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c
```

The editor has a menu bar with "파일(F)", "편집(E)", "서식(O)", "보기(V)", and "도움말". The status bar at the bottom shows "Windows (CRLF)", "Ln 1, Col 1", and "100%".

수동으로 합쳐보죠 - .gitignore

A screenshot of a text editor window titled ".gitignore - 메모장". The window contains a .gitignore file with the following content:

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Rope project settings
.ropeproject

# mkdocs documentation
/site

# mypy
.mypy_cache/
```


충돌 제거한 내용을 commit

```
MINGW64:/c/Users/VanDarkHolme/learnGit
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  new file:   LICENSE

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both added:   .gitignore
    both added:   README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master|MERGING)
$ git add .gitignore README.md

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master|MERGING)
$ git commit
[master 64345e4] Merge branch 'master' of https://github.com/brain-
hack/learnGit2
```

새로 merge한 내역 push

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git push origin
Enumerating objects: 19, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 6.52 KiB | 606.00 KiB/s, done.
Total 15 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/brain-hack/learnGit1.git
    6079006..64345e4  master -> master

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git push second
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 12 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (16/16), 1.78 KiB | 606.00 KiB/s, done.
Total 16 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), completed with 1 local object.
To https://github.com/brain-hack/learnGit2.git
    752cb4c..64345e4  master -> master
```

remote가 업데이트 됐으니 받아야겠죠?

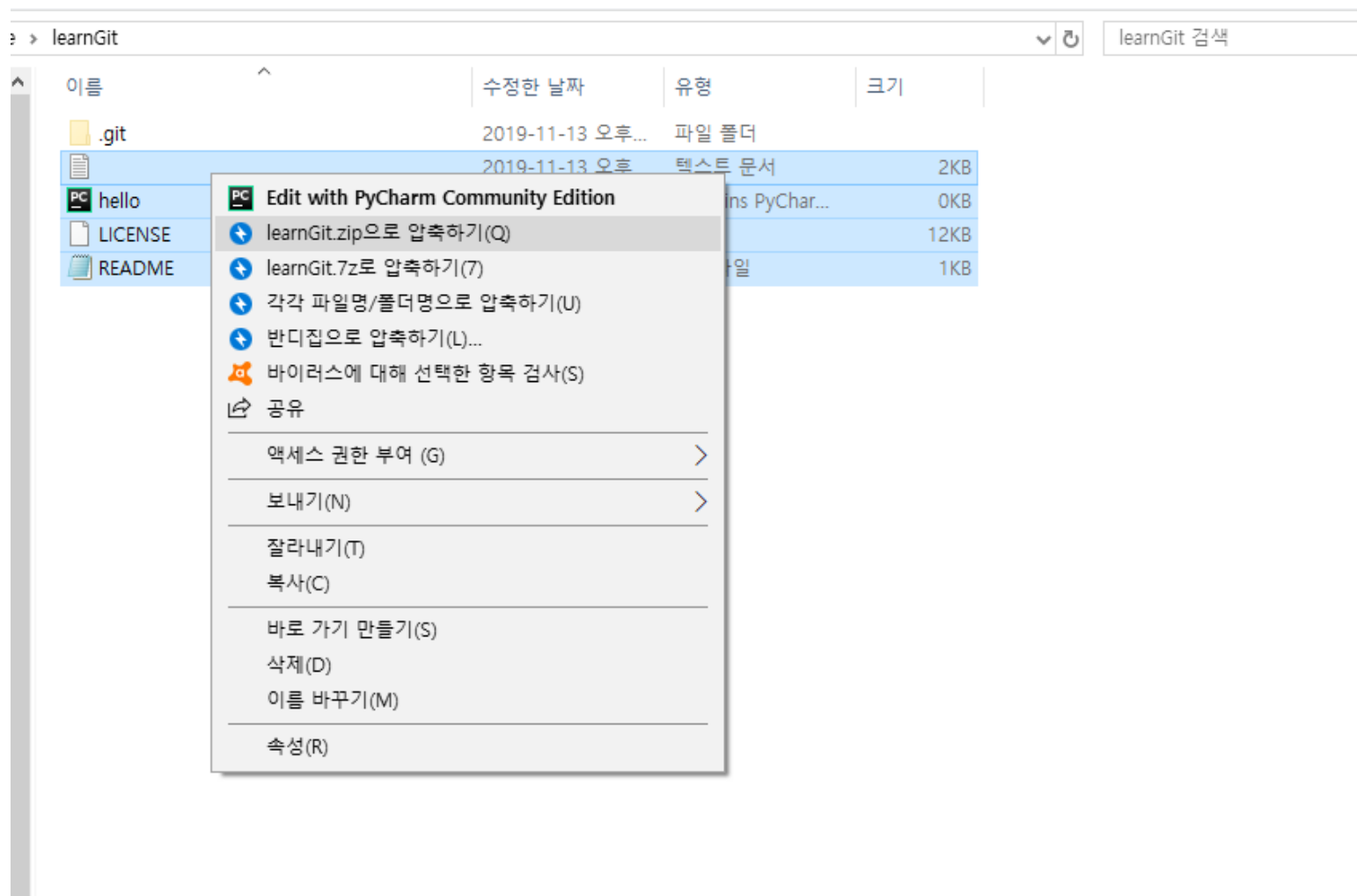
```
MINGW64:/c/Users/VanDarkHolme/learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ cd ../learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git pull
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 4), reused 15 (delta 4), pack-reused 0
Unpacking objects: 100% (15/15), done.
From https://github.com/brain-hack/learnGit1
   6079006..64345e4  master    -> origin/master
Updating 6079006..64345e4
Fast-forward
 .gitignore | 104 ++++++
 LICENSE    | 201 ++++++
+++++++
 README.md  |   8 ++-
3 files changed, 311 insertions(+), 2 deletions(-)
create mode 100644 .gitignore
create mode 100644 LICENSE
```

local learnGit : 우선 압축파일을 만들어볼까요

(이 파일은 커밋에 안 넣을 겁니다)



local learnGit : requirements 파일을 추가합니다.

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ cd ../learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ notepad
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$
```

```
requirements - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
numpy
tensorflow
jupyter == 0.9.5
Windows Ln 3. Col 100%
```

git add를 해볼까요

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git add .

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   learnGit.zip
    new file:   requirements.txt

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ |
```

오 이런 실수로 압축파일도 추가했네요

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   learnGit.zip
    new file:   requirements.txt

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset HEAD learnGit.zip

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   requirements.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    learnGit.zip
```

requirements의 주피터 버전을 1.0.0으로

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ notepad requirements.txt

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   requirements.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   requirements.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    learnGit.zip

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git diff
diff --git a/requirements.txt b/requirements.txt
index 2f98193..5bc6f67 100644
--- a/requirements.txt
+++ b/requirements.txt
@@ -1,3 +1,3 @@
 numpy
 tensorflow
-jupyter == 0.9.5
\ No newline at end of file
+jupyter == 1.0.0
\ No newline at end of file
```

```
requirements - 메모상
파일(F) 편집(E) 서식(O) 보기(V) 도움말

numpy
tensorflow
jupyter == 1.0.0|

Window Ln 3, Co 100%
```


앗 jupyter 1.0.0에서는 원하는 기능이 안됩니다.

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   requirements.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   requirements.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    learnGit.zip

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git checkout -- requirements.txt

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   requirements.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    learnGit.zip
```

커밋하기

```
MINGW64:/c/Users/VanDarkHolme/learnGit

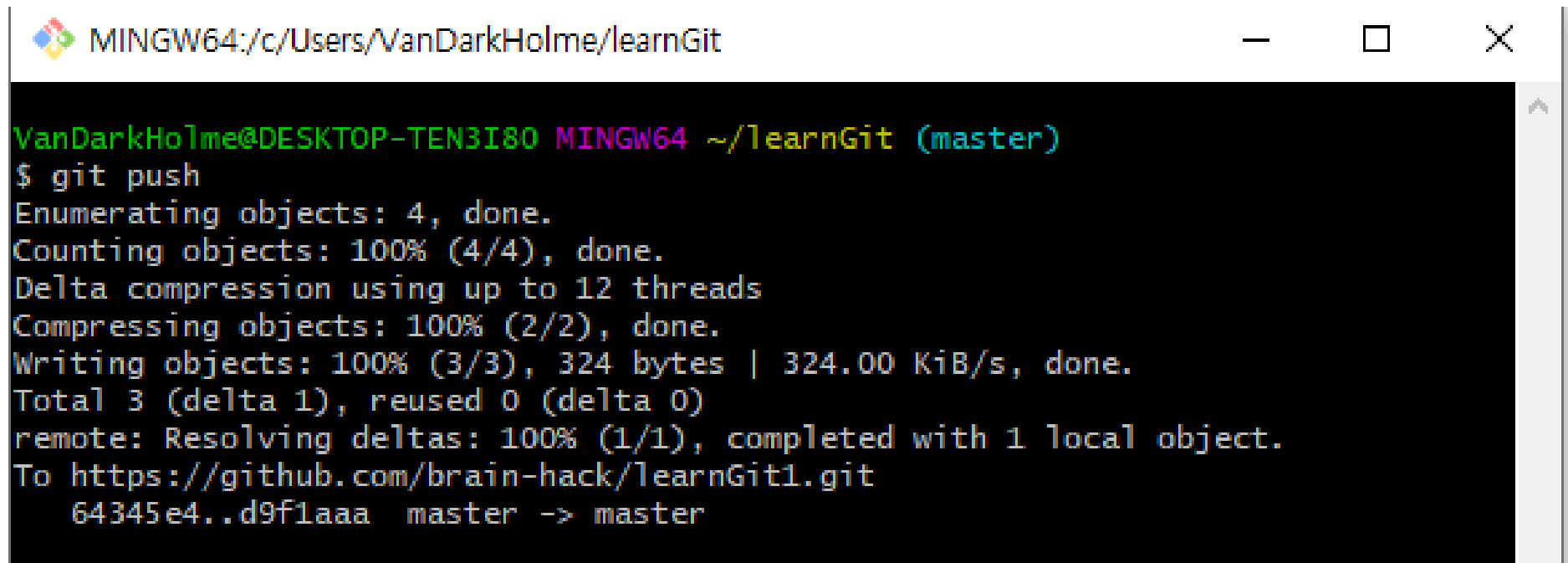
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git commit -m "add pip requirements file"
[master d9f1aaa] add pip requirements file
1 file changed, 3 insertions(+)
create mode 100644 requirements.txt

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log -p -1
commit d9f1aaa38738b10336b3ade45abb145e67f74c90 (HEAD -> master)
Author: brain-hack <deep-learning@kakao.com>
Date: Thu Nov 14 03:54:09 2019 +0900

    add pip requirements file

diff --git a/requirements.txt b/requirements.txt
new file mode 100644
index 0000000..2f98193
--- /dev/null
+++ b/requirements.txt
@@ -0,0 +1,3 @@
+numpy
+tensorflow
+jupyter == 0.9.5
\ No newline at end of file
```

push하기

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/VanDarkHolme/learnGit". The terminal shows the execution of the "git push" command. The output indicates that 4 objects were enumerated, 4 objects were counted (100%), and 2 objects were compressed (100%). A total of 3 objects (delta 1) were written to the remote repository, reusing 0 objects (delta 0). The remote repository is "https://github.com/brain-hack/learnGit1.git", and the push was successful, updating the master branch from 64345e4 to d9f1aaa.

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 324.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/brain-hack/learnGit1.git
   64345e4..d9f1aaa  master -> master
```

local repo : learnGit1 작업하기

```
MINGW64:/c/Users/VanDarkHolme/learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ cd ../learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/brain-hack/learnGit1
   64345e4..d9f1aaa  master    -> origin/master
Updating 64345e4..d9f1aaa
Fast-forward
 requirements.txt | 3 +++
1 file changed, 3 insertions(+)
create mode 100644 requirements.txt
```

local repo : learnGit1 작업하기

```
MINGW64:/c:/Users/VanDarkHolme/learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ notepad hello.py

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git add hello.py

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git commit -m "add func print_git"
[master 3486e28] add func print_git
1 file changed, 13 insertions(+)

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 397 bytes | 397.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/brain-hack/learnGit1.git
d9f1aaa..3486e28 master -> master
```

```
hello - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말

import sys

def print_message():
    message = "world"
    print("hello {}".format(message))

def print_git():
    message = "from the hell"
    sys.print("git {}".format(message))

if __name__ == "__main__":
    print_message()
    print_git()
```

Windows (CRLF) Ln 9, Col 9 100%

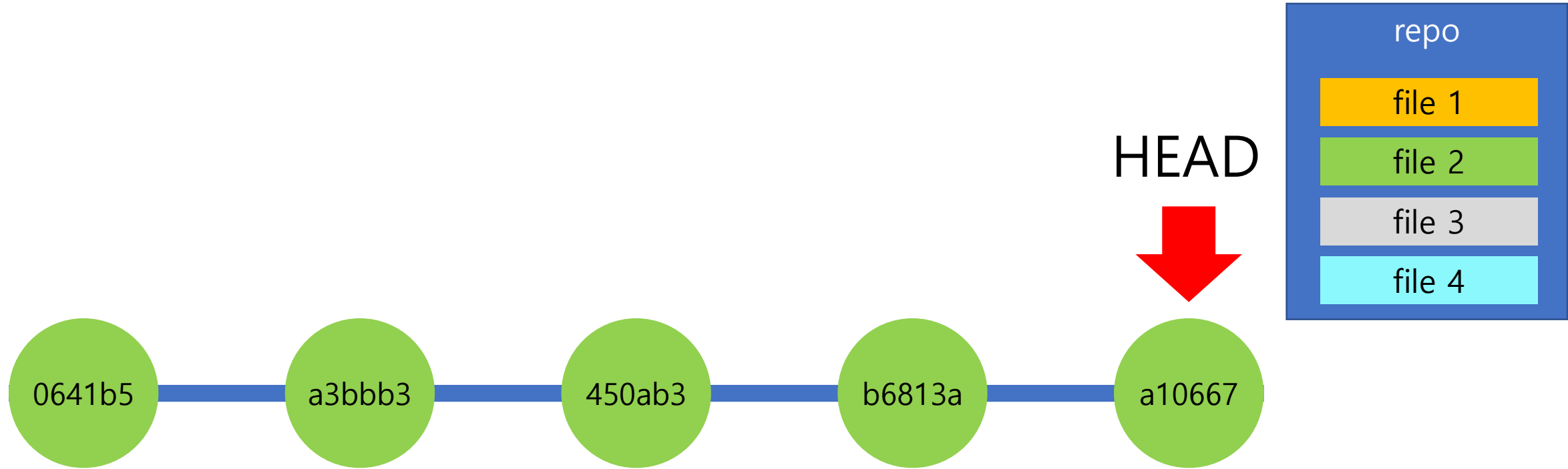
local repo : learnGit pull받기

```
MINGW64:/c/Users/VanDarkHolme/learnGit

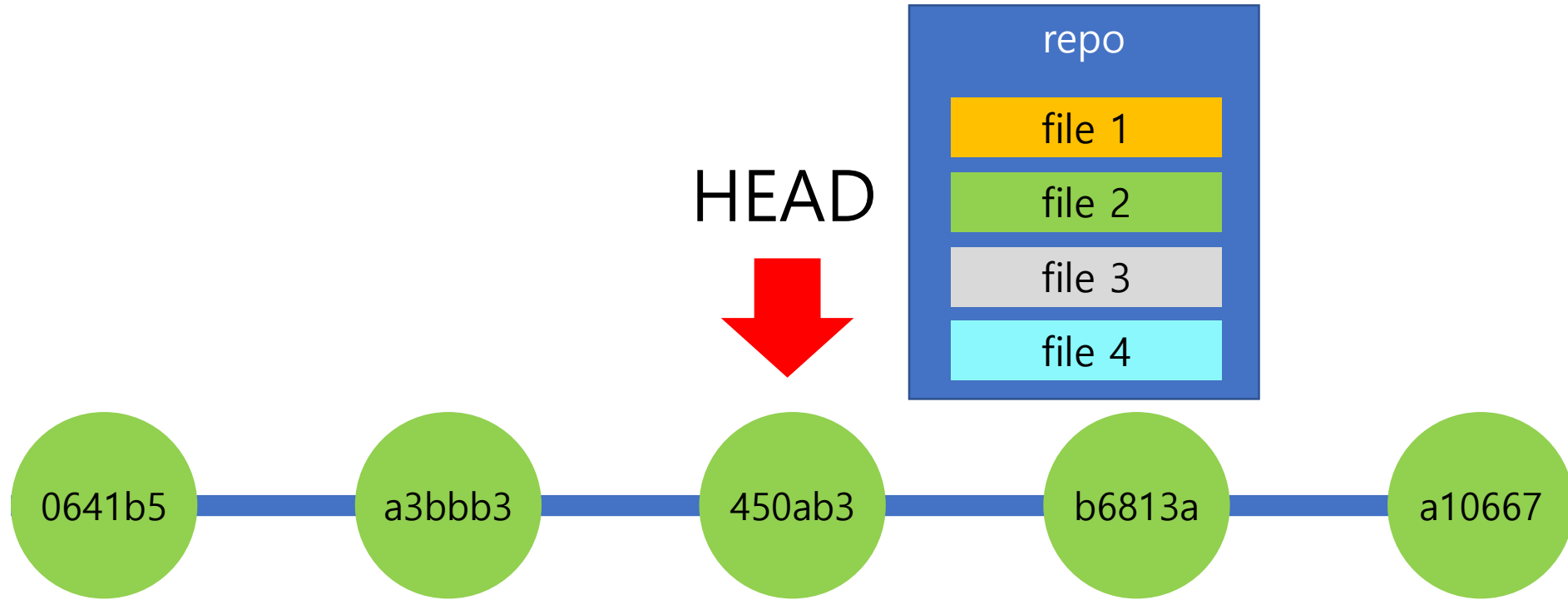
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit1 (master)
$ cd ../learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/brain-hack/learnGit1
   d9f1aaa..3486e28  master      -> origin/master
Updating d9f1aaa..3486e28
Fast-forward
 hello.py | 13 ++++++++
 1 file changed, 13 insertions(+)
```

커밋 되돌리기

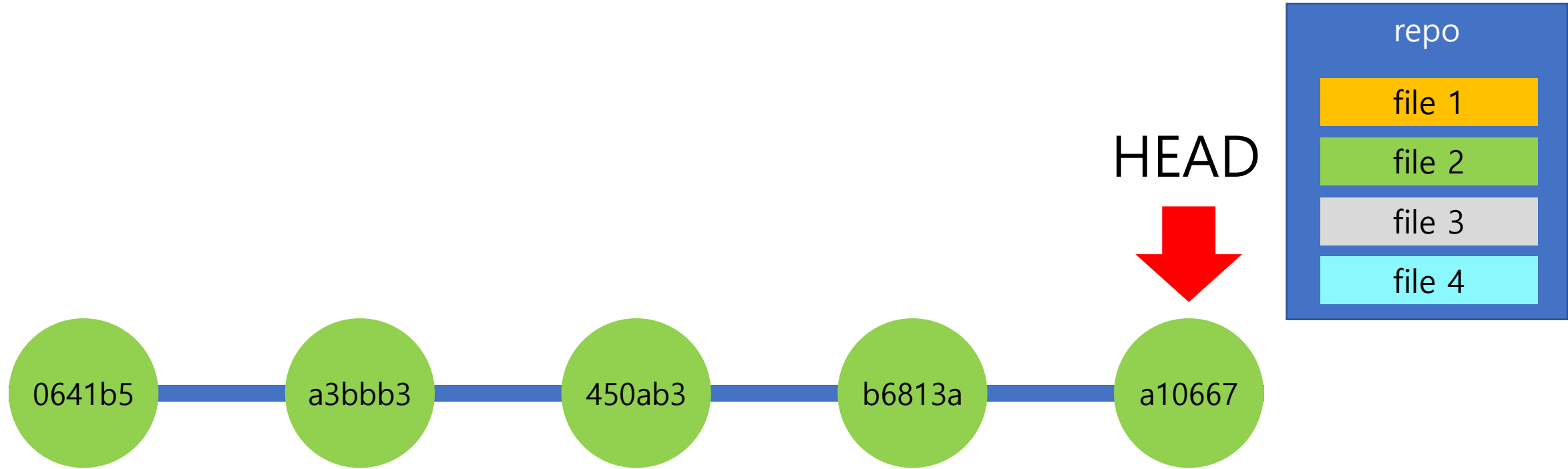


커밋 되돌리기

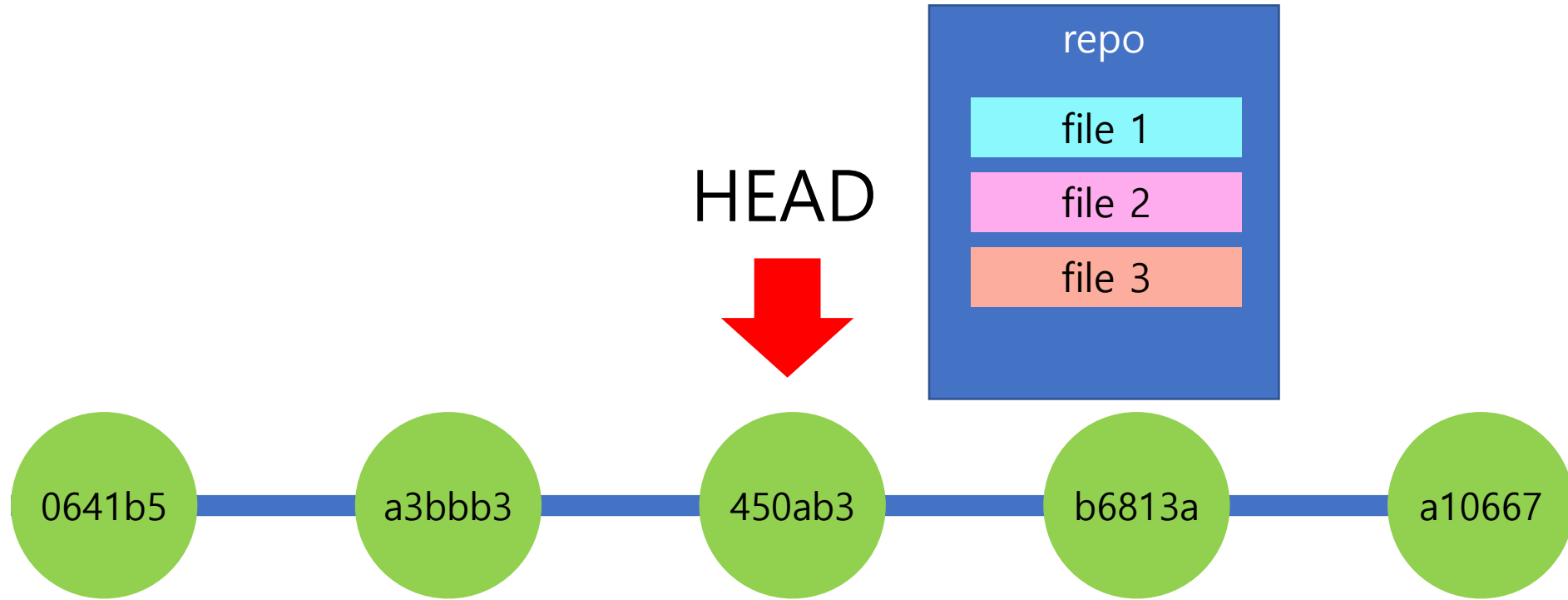


```
$ git reset --soft 450ab3
```


커밋 되돌리기

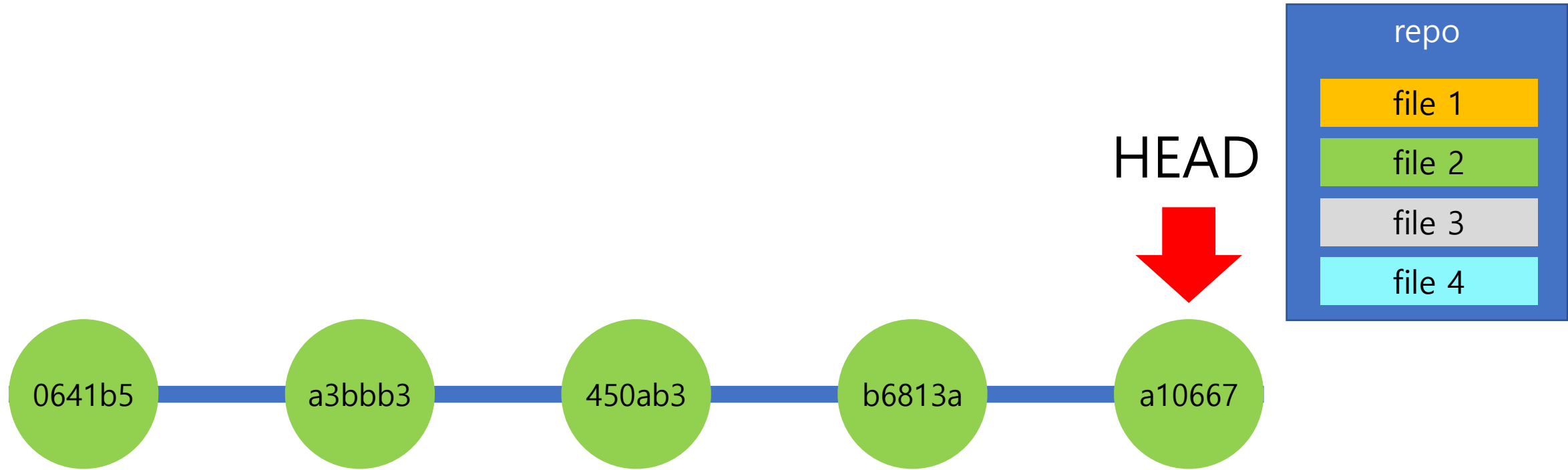


커밋 되돌리기

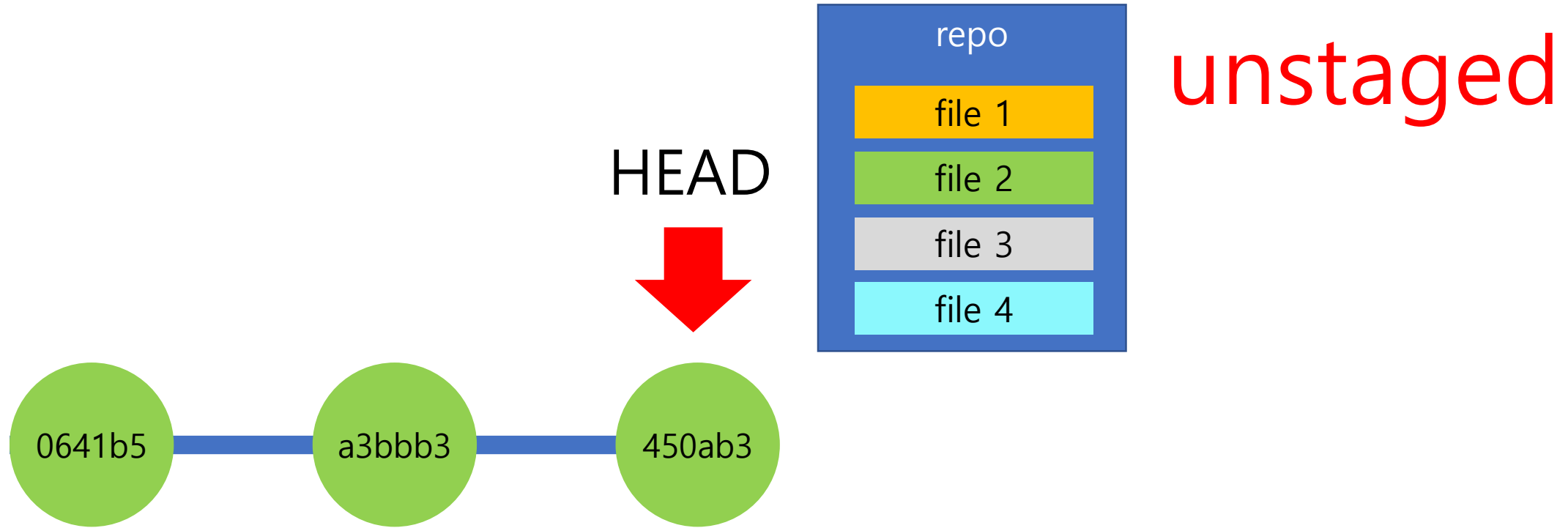


```
$ git reset --mixed 450ab3
```

커밋 되돌리기



커밋 되돌리기



```
$ git reset --hard 450ab3
```

커밋 되돌리기 reset --soft

```
MINGW64:/c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
3486e281c7a066d7b769f0c7d62d539afb8ca462 (HEAD -> master, origin/master) add func print_git
d9f1aaa38738b10336b3ade45abb145e67f74c90 add pip requirements file
64345e4765a6bdfc1fc952bfe069c47b3e78d87e (second/master) Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae15496e28cbb63a58491b56d536cb506e93 Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c Initial commit
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset --soft cdafe15496e28cbb63a58491b56d536cb506e93

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
cdafae15496e28cbb63a58491b56d536cb506e93 (HEAD -> master) Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        new file:   LICENSE
        modified:   README.md
        modified:   hello.py
        new file:   requirements.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        learnGit.zip

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git idff
git: 'idff' is not a git command. See 'git --help'.

The most similar command is
    diff
```

reset 취소

```
MINGW64:/c/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reflog
cdafae1 (HEAD -> master) HEAD@{0}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{1}: reset: moving to HEAD@{6}
3486e28 (origin/master) HEAD@{2}: reset: moving to HEAD@{5}
cdafae1 (HEAD -> master) HEAD@{3}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{4}: reset: moving to HEAD@{3}
cdafae1 (HEAD -> master) HEAD@{5}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{6}: reset: moving to HEAD@{1}
cdafae1 (HEAD -> master) HEAD@{7}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{8}: pull: Fast-forward
d9f1aaa HEAD@{9}: commit: add pip requirements file
64345e4 (second/master) HEAD@{10}: commit (merge): Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae1 (HEAD -> master) HEAD@{11}: pull: Merge made by the 'recursive' strategy.
b990c79 HEAD@{12}: commit: update readme and add gitignore
abd1827 HEAD@{13}: commit (amend): Initial commit
4521575 HEAD@{14}: commit (initial): Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset --hard HEAD@{6}
HEAD is now at 3486e28 add func print_git

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
3486e281c7a066d7b769f0c7d62d539afb8ca462 (HEAD -> master, origin/master) add func print_git
d9f1aaa38738b10336b3ade45abb145e67f74c90 add pip requirements file
64345e4765a6bdfc1fc952bfe069c47b3e78d87e (second/master) Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae15496e28cbb63a58491b56d536cb506e93 Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c Initial commit
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit
```

커밋 되돌리기 reset --mixed

```
MINGW64: /c/Users/VanDarkHolme/learnGit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
3486e281c7a066d7b769f0c7d62d539afb8ca462 (HEAD -> master, origin/master) add func print_git
d9f1aaa38738b10336b3ade45abb145e67f74c90 add pip requirements file
64345e4765a6bdfc1fc952bfe069c47b3e78d87e (second/master) Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae15496e28cbb63a58491b56d536cb506e93 Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c Initial commit
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset --mixed cdafae15496e28cbb63a58491b56d536cb506e93
Unstaged changes after reset:
M   .gitignore
M   README.md
M   hello.py

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
cdafae15496e28cbb63a58491b56d536cb506e93 (HEAD -> master) Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore
        modified:   README.md
        modified:   hello.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        LICENSE
        learnGit.zip
        requirements.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

reset 취소

```
MINGW64:/c:/Users/VanDarkHolme/learnGit
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reflog
cdafae1 (HEAD -> master) HEAD@{0}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{1}: reset: moving to HEAD@{6}
3486e28 (origin/master) HEAD@{2}: reset: moving to HEAD@{5}
cdafae1 (HEAD -> master) HEAD@{3}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{4}: reset: moving to HEAD@{3}
cdafae1 (HEAD -> master) HEAD@{5}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{6}: reset: moving to HEAD@{1}
cdafae1 (HEAD -> master) HEAD@{7}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{8}: pull: Fast-forward
d9f1aaa HEAD@{9}: commit: add pip requirements file
64345e4 (second/master) HEAD@{10}: commit (merge): Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae1 (HEAD -> master) HEAD@{11}: pull: Merge made by the 'recursive' strategy.
b990c79 HEAD@{12}: commit: update readme and add gitignore
abd1827 HEAD@{13}: commit (amend): Initial commit
4521575 HEAD@{14}: commit (initial): Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset --hard HEAD@{6}
HEAD is now at 3486e28 add func print_git

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
3486e281c7a066d7b769f0c7d62d539afb8ca462 (HEAD -> master, origin/master) add func print_git
d9f1aaa38738b10336b3ade45abb145e67f74c90 add pip requirements file
64345e4765a6bdfc1fc952bfe069c47b3e78d87e (second/master) Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae15496e28cbb63a58491b56d536cb506e93 Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c Initial commit
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit
```


사실 reset hard도 복구가 가능하다

MINGW64~/c/Users/VanDarkHolme/learnGit

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
3486e281c7a066d7b769f0c7d62d539afb8ca462 (HEAD -> master, origin/master) add func print_git
d9f1aaa38738b10336b3ade45abb145e67f74c90 add pip requirements file
64345e4765a6bdfc1fc952bfe069c47b3e78d87e (second/master) Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae15496e28cbb63a58491b56d536cb506e93 Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c Initial commit
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset --hard cdafae15496e28cbb63a58491b56d536cb506e93
HEAD is now at cdafae1 Merge branch 'master' of https://github.com/brain-hack/learnGit1

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    learnGit.zip

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

```
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
cdafae15496e28cbb63a58491b56d536cb506e93 (HEAD -> master) Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit

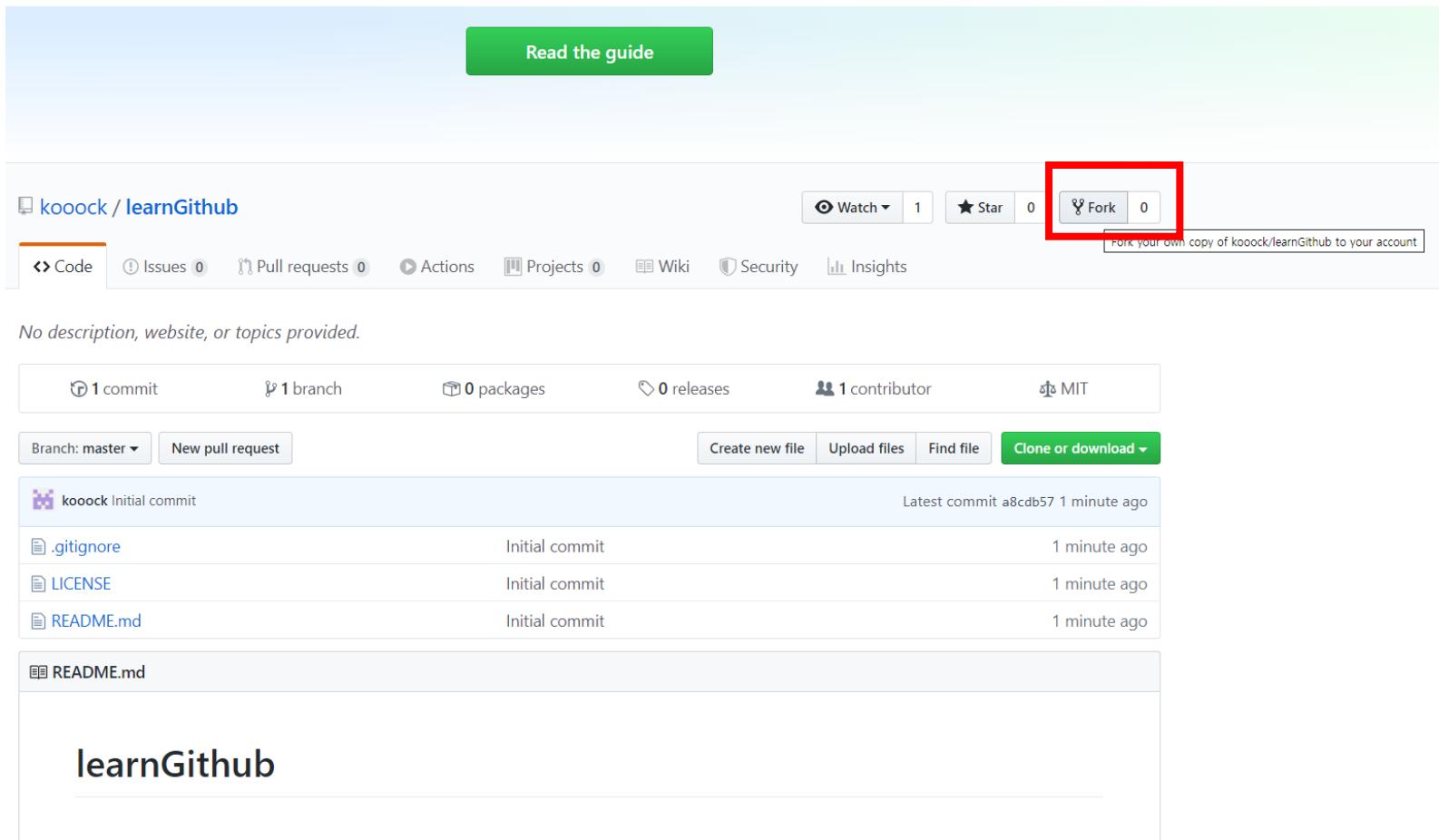
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reflog
cdafae1 (HEAD -> master) HEAD@{0}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{1}: reset: moving to HEAD@{6}
cdafae1 (HEAD -> master) HEAD@{2}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{3}: reset: moving to HEAD@{6}
3486e28 (origin/master) HEAD@{4}: reset: moving to HEAD@{5}
cdafae1 (HEAD -> master) HEAD@{5}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{6}: reset: moving to HEAD@{3}
cdafae1 (HEAD -> master) HEAD@{7}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{8}: reset: moving to HEAD@{1}
cdafae1 (HEAD -> master) HEAD@{9}: reset: moving to cdafae15496e28cbb63a58491b56d536cb506e93
3486e28 (origin/master) HEAD@{10}: pull: Fast-forward
d9f1aaa HEAD@{11}: commit: add pip requirements file
64345e4 (second/master) HEAD@{12}: commit (merge): Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae1 (HEAD -> master) HEAD@{13}: pull: Merge made by the 'recursive' strategy.
b990c79 HEAD@{14}: commit: update readme and add gitignore
abd1827 HEAD@{15}: commit (amend): Initial commit
4521575 HEAD@{16}: commit (initial): Initial commit

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git reset --hard HEAD@{10}
HEAD is now at 3486e28 add func print_git

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ git log --pretty=oneline
3486e281c7a066d7b769f0c7d62d539afb8ca462 (HEAD -> master, origin/master) add func print_git
d9f1aaa38738b10336b3ade45abb145e67f74c90 add pip requirements file
64345e4765a6bdfc1fc952bfe069c47b3e78d87e (second/master) Merge branch 'master' of https://github.com/brain-hack/learnGit2
cdafae15496e28cbb63a58491b56d536cb506e93 Merge branch 'master' of https://github.com/brain-hack/learnGit1
b990c79dfa3eaa956207e365c22c46b1344091ea update readme and add gitignore
60790065d89a06a27a53f8bce2dd004a80b10239 add python code : hello world
752cb4c9f007e8cc5494f33b20a46a3abf1f2d9c Initial commit
abd182712e7b732882d6f0b6b80a362b34d63613 Initial commit
```

fork를 해봅시다

<https://github.com/kooock/learnGithub>



Read the guide

kooock / learnGithub

Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights

No description, website, or topics provided.

1 commit 1 branch 0 packages 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

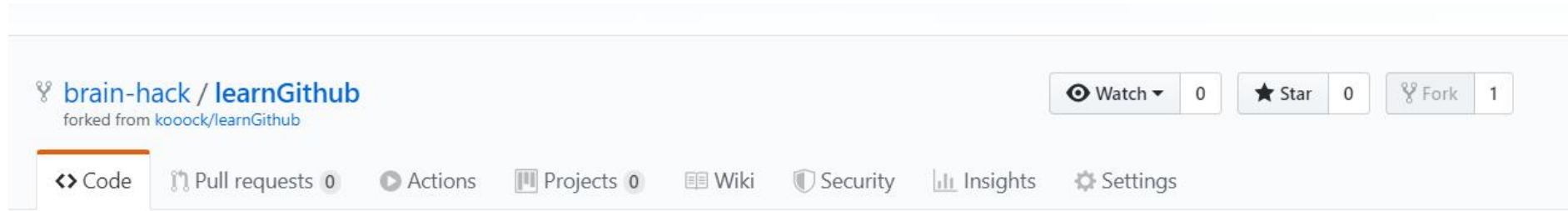
kooock Initial commit Latest commit a8cdb57 1 minute ago

.gitignore	Initial commit	1 minute ago
LICENSE	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago

README.md

learnGithub

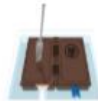
fork를 해봅시다




Forking kooock/learnGithub


It should only take a few seconds.


Refresh





fork를 해봅시다


 **brain-hack / learnGithub**
forked from kooock/learnGithub


 Watch 0


 Star 0


 Fork 1


 Code


 Pull requests 0


 Actions

 Projects 0

 Wiki


 Security


 Insights


 Settings


No description, website, or topics provided. [Edit](#)


[Manage topics](#)


 1 commit

 1 branch

 0 packages

 0 releases

 1 contributor

 MIT

Branch: master ▾

New pull request


Create new file




Upload files



Find file

Clone or download ▾

This branch is even with kooock:master. [Pull request](#) [Compare](#)

 **kooock** Initial commit Latest commit a8cdb57 31 minutes ago

 .gitignore	Initial commit	31 minutes ago
 LICENSE	Initial commit	31 minutes ago
 README.md	Initial commit	31 minutes ago

 [README.md](#) 

learnGithub

clone을 해봅시다



```
MINGW64:/c/Users/VanDarkHolme/learnGithub

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGit (master)
$ cd ../

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ git clone https://github.com/brain-hack/learnGithub.git
Cloning into 'learnGithub'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~
$ cd learnGithub/

VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```


local repo에서 작업을 해볼까요?

```
MINGW64:/c/Users/VanDarkHolme/learnGithub
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub (master)
$ cd brain-hack/
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub/brain-hack (master)
$ notepad
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub/brain-hack (master)
$ ls
introduce.md
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub/brain-hack (master)
$ git add .
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub/brain-hack (master)
$ cd ..
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub (master)
$ git commit -m "add brain-hack introduction"
[master 456b40a] add brain-hack introduction
1 file changed, 5 insertions(+)
create mode 100644 brain-hack/introduce.md
VanDarkHolme@DESKTOP-TEN3I80 MINGW64 ~/learnGithub (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 464 bytes | 464.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/brain-hack/learnGithub.git
a8cdb57..456b40a master -> master
```

```
introduce - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말
# 안녕하세요 brain-hack 입니다.


만나서 반갑습니다. <br/>
저는 모두의 연구소에서 meani.mo
연구원으로 활동중입니다.
Windows (CR) Ln 3, Col 17 100%
```

remote repository 확인


 **brain-hack / learnGithub**
forked from [kooock/learnGithub](#)


Watch ▾0


★ Star0


 Fork1


<> Code


 Pull requests0


 Actions

 Projects0

 Wiki

 Security


 Insights


 Settings


No description, website, or topics provided.


[Manage topics](#)


Edit


 2 commits

 1 branch

 0 packages

 0 releases

 2 contributors

 MIT

Branch: master ▾

New pull request


Create new file


Upload files


Find file

Clone or download ▾





This branch is 1 commit ahead of kooock:master.


 Pull request


 Compare

 **brain-hack** add brain-hack introduction

Latest commit 456b40a 3 minutes ago


 brain-hack	add brain-hack introduction	3 minutes ago
 .gitignore	Initial commit	1 hour ago
 LICENSE	Initial commit	1 hour ago
 README.md	Initial commit	1 hour ago




 README.md











learnGithub


pull request 만들기



 **brain-hack / learnGithub**
forked from [kooock/learnGithub](#)

 Watch 0  Star 0  Fork 1


 Code  Pull requests 0  Actions  Projects 0  Wiki  Security  Insights  Settings

Filters ▾

 is:pr is:open

 Labels 9  Milestones 0

New pull request





Welcome to Pull Requests!

Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).



pull request 만들기





Comparing changes


Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



 base repository: kooock/learnGithub ▾ base: master ▾  head repository: brain-hack/learnGithub ▾ compare: master ▾


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



 **Create pull request** Discuss and review the changes in this comparison with others. 

 **1** commit  **1** file changed  **0** commit comments  **1** contributor

 Commits on Nov 14, 2019

  **brain-hack** add brain-hack introduction 456b40a

 Showing **1** changed file with **5** additions and **0** deletions. Unified Split

▼ 5  brain-hack/introduce.md 

... ... @@ -0,0 +1,5 @@



1 + # 안녕하세요 brain-hack 입니다.

2 +

3 + 만나서 반갑습니다.

4 + 저는 모두의 연구소에서 meani.mo


5 + 연구원으로 활동중입니다.

  ...

pull request 만들기

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 repository: `kooock/learnGithub` ▼ base: `master` ▼ ◀ head repository: `brain-hack/learnGithub` ▼ compare: `master` ▼

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



add brain-hack introduction

Write

Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ 📌 ↶

안녕하세요 brain-hack 윤성국이라고 합니다.
제 소개파일을 추가했습니다.
PR받아주시면 감사하겠습니다.

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



☒ Allow edits from maintainers. [Learn more](#)

Create pull request ▼

pull request 확인

[kooock](#) / [learnGithub](#)

Watch1

Star0

Fork1

[Code](#) [Issues0](#) [Pull requests1](#) [Actions](#) [Projects0](#) [Wiki](#) [Security](#) [Insights](#)

add brain-hack introduction #1

Edit

Open

 brain-hack wants to merge 1 commit into [kooock:master](#) from [brain-hack:master](#)


Conversation0

Commits1


Checks0

Files changed1

+5-0



 brain-hack commented now

안녕하세요 brain-hack 윤성국이라고 합니다.
제 소개파일을 추가했습니다.
PR받아주시면 감사하겠습니다.

 add brain-hack introduction

456b40a

Add more commits by pushing to the **master** branch on [brain-hack/learnGithub](#).

  **This branch has no conflicts with the base branch**
Only those with [write access](#) to this repository can merge pull requests.

Reviewers

No reviews

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

