Git & GitHub 入門

KYUNGSUNG UNIVERSITY SINCE 1955

형상관리 & 버전관리

형상관리 : 소프트웨어 개발 및 유지보수 과정에서 발생하는 소스코드, 문서 등 각종 결과물(형상)에 대한 변경사항을 체계적으로 관리하고 제어하기 위한 활동

버전관리 : 문서나 설계도, 소스 코드 등의 변경점을 관리

✓ 버전관리 : 동일한 정보에 대한 여러 버전을 관리하는 것
소프트웨어공학에서는 일반적으로 소스 코드만을 관리하는 것을 버전관리라고 정의

✓ 변경관리 : '소스코드'의 변경사항을 관리하는 것
✓ 버전관리 : 이러한 변경사항을 '버전'이라는 개념을 통해 관리한다는 점에서 그 차이가 있음.
✓ 형상관리 : 버전관리를 포함하여, 소프트웨어 프로젝트와 관련된 모든 변경사항을 관리
.

그러나 실무에서는 변경관리, 버전관리, 형상관리가 명확하게 구분되고 있지는 않고, 대체로 비슷한 의미로 사용된다고 한다

변경관리 ⊆ 버전관리 ⊆ 형상관리

버전관리 시스템

방식에 따른 버전관리 유형

버전 도구 유형	버전 관리 도구 설명
공유 폴더 방식 (RCS, SCCS)	• 매일 개발 완료 파일은 약속된 위치의 공유 폴더에 복사 • 담당자 한 명이 매일 공유 폴더의 파일을 자기 PC로 복사하고 컴파일하여 에러 확인과 정상 동작 여부 확인 • 정상 동작일 경우 다음날 각 개발자들이 동작 여부 확인
클라이언트/서버 방식 (CVS, SVN)	• 중앙에 버전 관리 시스템이 항시 동작 • 개발자들의 현재 작업 내용과 이전 작업내용 축적 용이 • 서로 다른 개발자가 같은 파일을 작업했을 때 경고 출력 • Trac이나 CVS view와 같은 GUI 툴을 이용하여 모니터링 가능
분산 저장소 방식 (Git, Bitkeeper, etc)	• 로컬 저장소와 원격 저장소 구조 • 중앙의 저장소에서 로컬에 복사(clone)한 순간 개발자 자신만의 로컬저장소에 생성 • 개발 완료한 파일 수정 이후 로컬 저장소에 커밋한 이후 다시 원격 저장소에 반영(push)하는 방식

버전관리 시스템

다양한 버전 관리 도구별 특징

버전 관리 도구	특징
CSV (Concurrent Versions System)	• 서버와 클라이언트로 구성되어 다수의 인원이 동식에 범용적인 운영체제로 접근하여 버전관리를 가능케 한다. • Client에 이클립스가 내장되어 있다.
SVN (Subversion)	• GNU의 버전 관리시스템으로 CVS의 장점은 이어받고 단점은 개선하여 2000년에 발표되었다. 사실상 업계 표준으로 사용되고 있으며 SVN 으로 불리고 있다.
RCS (Revision Control Systme)	• CVS와 달리 소스 파일의 수정을 한 사람만으로 제한하여 다수의 사람이 파일의 수정을 동시에 할 수 없도록 파일을 잠금하는 방식으로 버전 컨트롤을 수행한다.
Bitkeeper	• SVN과 비슷한 중앙 통제 방식의 비전 컨트롤 툴로서 대규모 프로젝트에서 빠른 속도를 내도록 개발되었다.
Git	• 기존 리눅스 커널의 버전 컨트롤을 하는 Bitkeeper를 대체하기 위해서 나온 새로운 버전 컨트롤로 현재의 리눅스는 이것을 통해 버전 컨트롤이 되고 있다. Git는 속도에 중점을 둔 분산형 버전 관리 시스템(DVCS)이며, 대형 프로젝트에서 효과적이고 실제로 유용하다. • Git는 SVN과 다르게 Commit은 로컬 저장소에서 이루어지고 push라는 동작으로 원격 저장소에 반영된다. • 또 받을 때도 Pull 또는 Fetch로 서버에서 변경된 내역을 받아 올 수 있다.
Clear Case	• IBM에서 제작되었다. • 복수 서버, 복수 클라이언트 구조이며 서버가 부족할 때 필요한 서버를 하나씩 추가하여 확장성을 기할 수 있다.

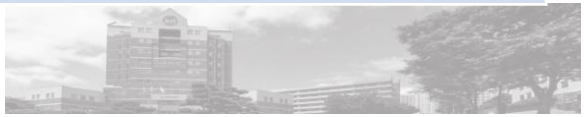
Git & Github

Git : **버전 관리 시스템**

Github : **Git으로 관리하는 프로젝트를 올려둘 수 있는 Git 호스팅 사이트**

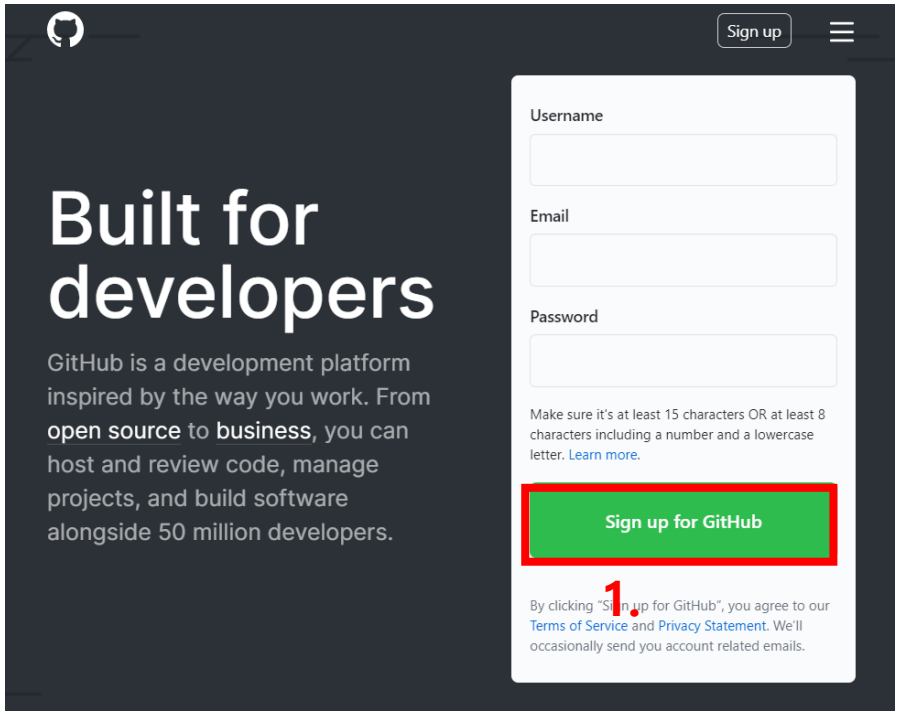
Git 호스팅 사이트	모기업	특징	가격정책
GitHub.com	GitHub Inc (MS에서 인수)	사용자 2800만명. 세계 최대 규모의 Git 호스팅 사이트	공개저장소 생성 무료. 비공개저장소는 작업자 3인 이하인 경우에는 무료. 설치형 버전인 Enterprise를 월 21달러에 사용할 수 있다.
GitLab.com	GitLab Inc	GitHub에 뒤지지 않는다. NASA, Sony 등 10만 개 이상의 조직이 사용하고 있다. GitLab 프로젝트 자체가 오픈소스여서 직접 서비스 발전에 기여할 수 있다.	공개저장소 및 비공개저장소 생성 무료. 소스코드 빌드에 유용한 도구 지원 성능에 따라 월 4달러에서 99달러 부담
BitBucket.org	Atlassian	사용자 600만명. 이슈 관리 시스템인 지라(Jira)를 만든 Atlassian이 모기업이어서 지라와 연동이 쉽다.	5명 이하 팀이면 공개저장소 및 비공개저장소 생성 무료. 그 이상이면 월 2달러에서 5달러 부담

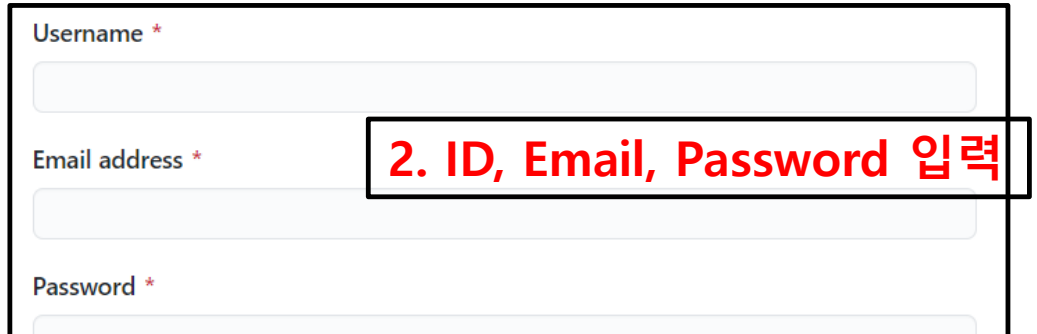
대표적인 Git 호스팅 사이트

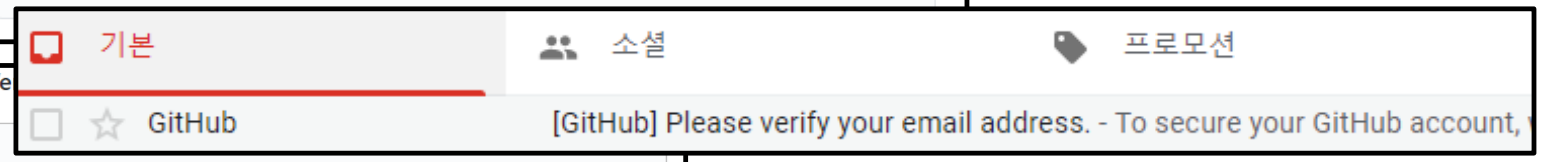


1. GitHub 가입하기

https://GitHub.com







5. Email click

Verify email address

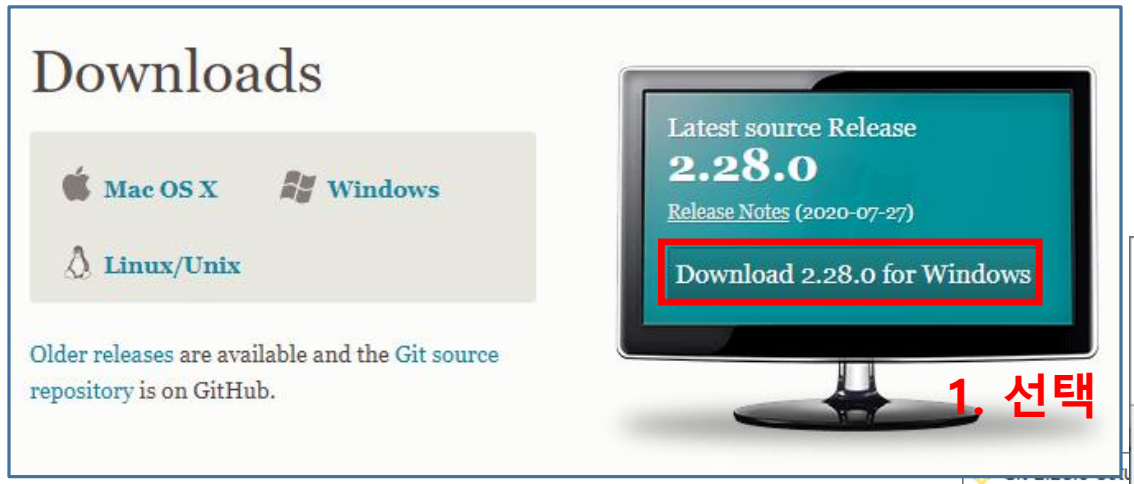
6. Button click

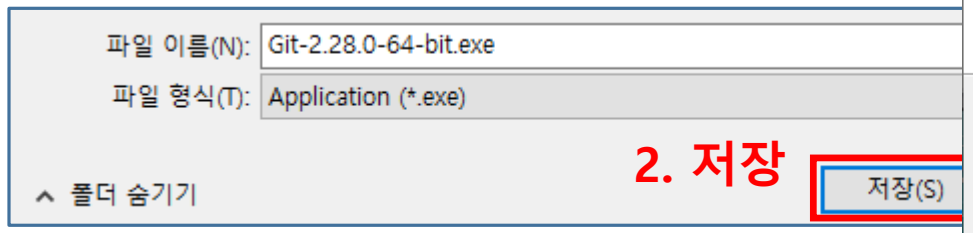
4. Button click

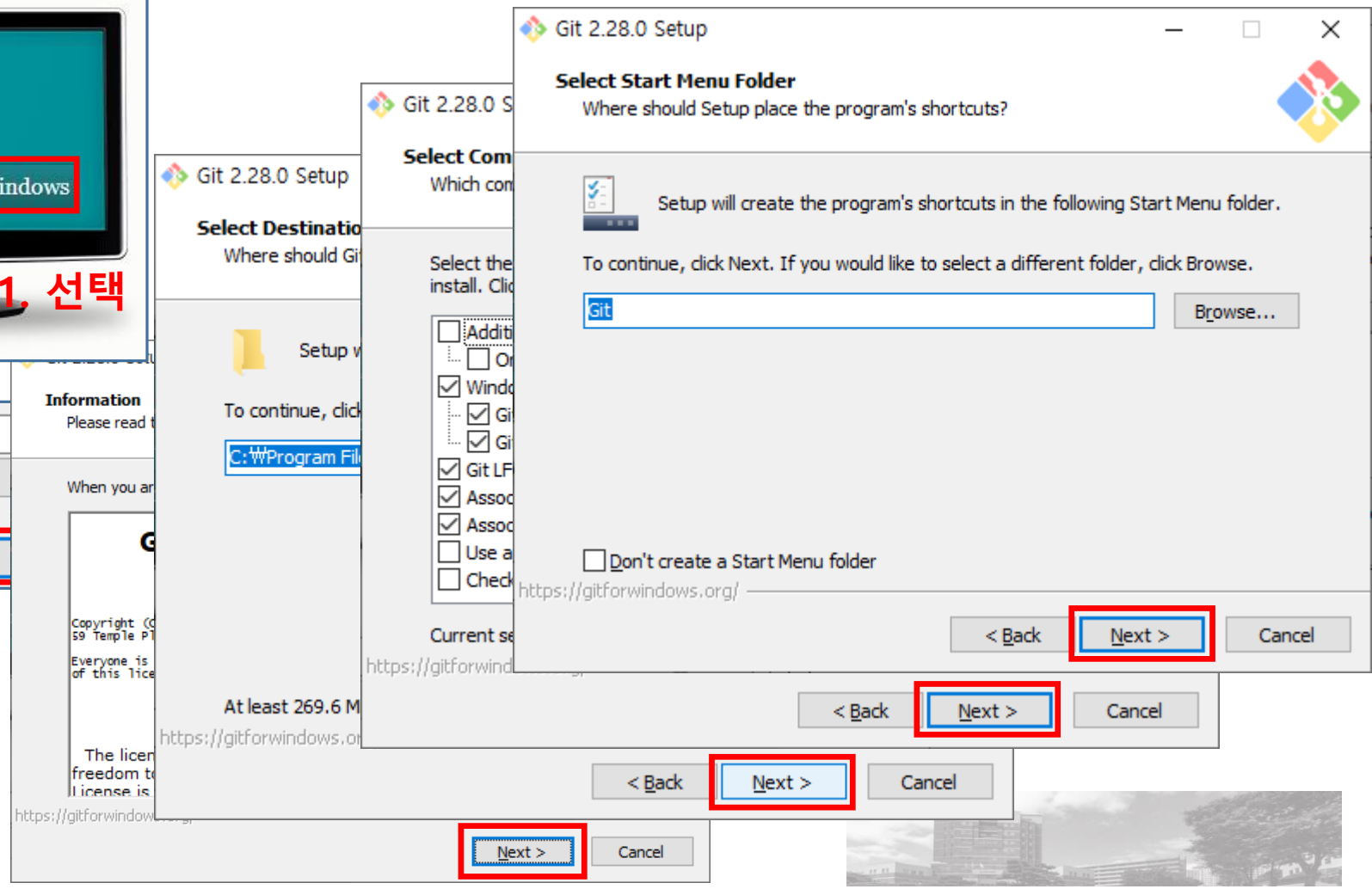
Create account

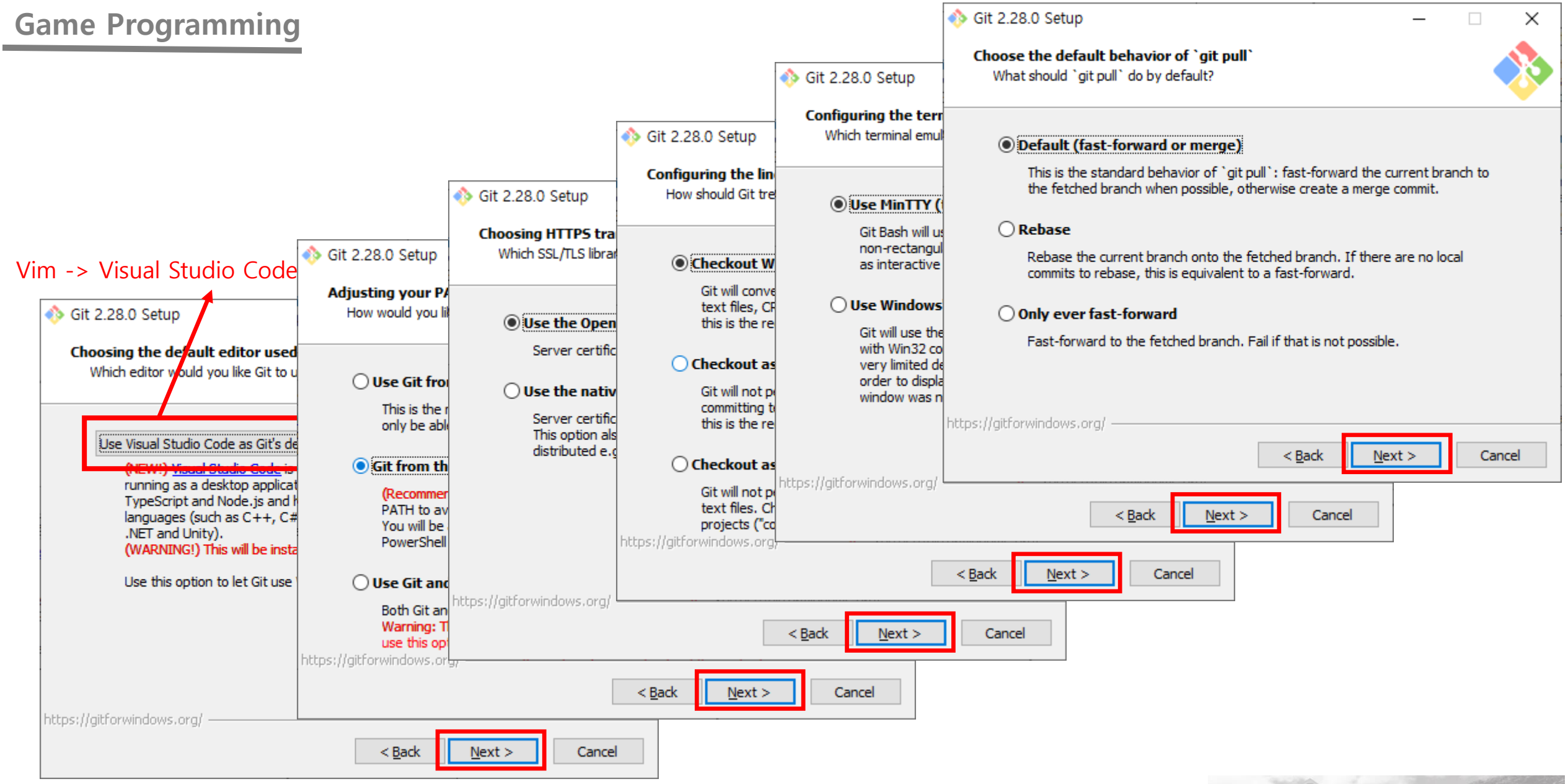
2. Git 설치하기

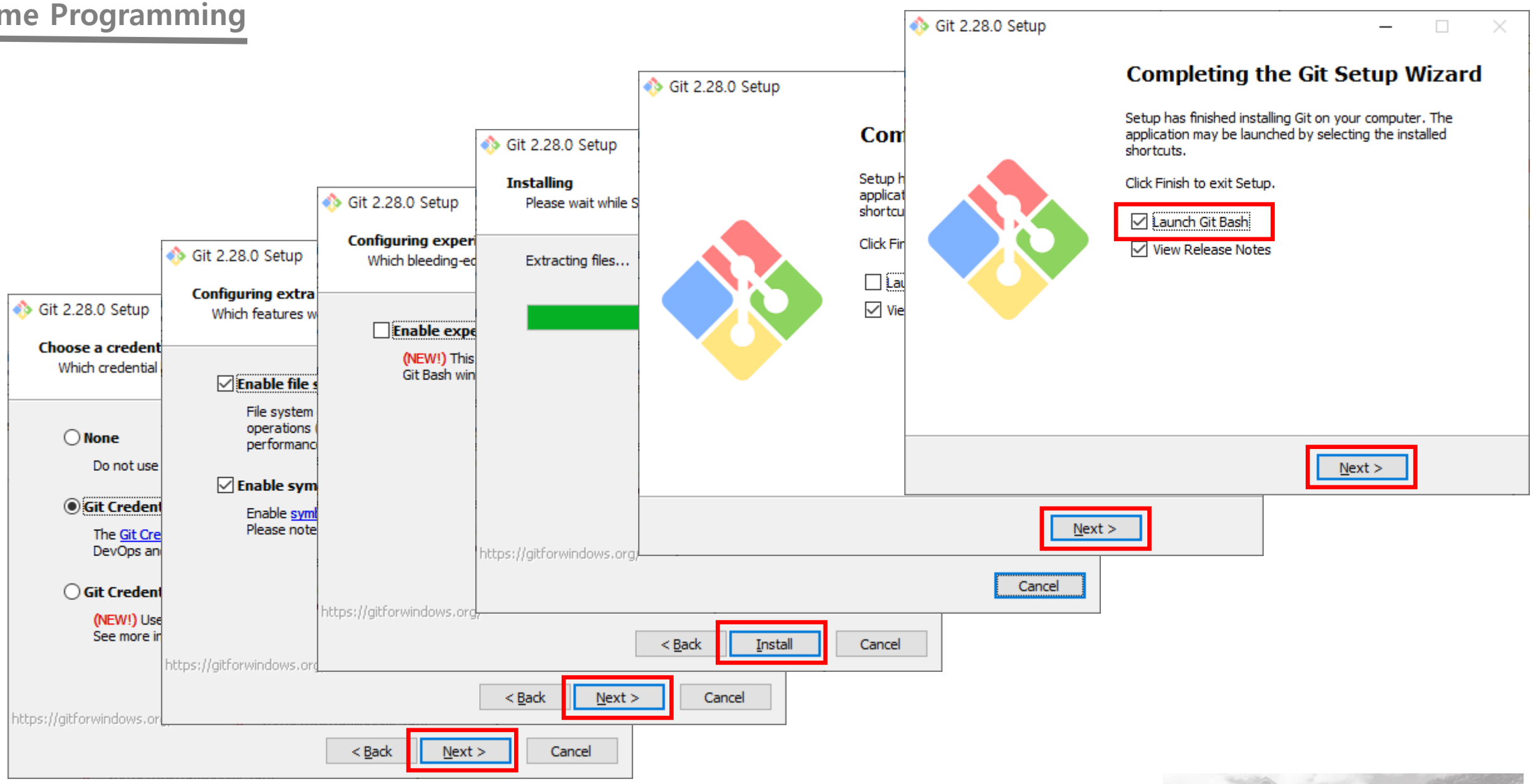
https://git-scm.com/download

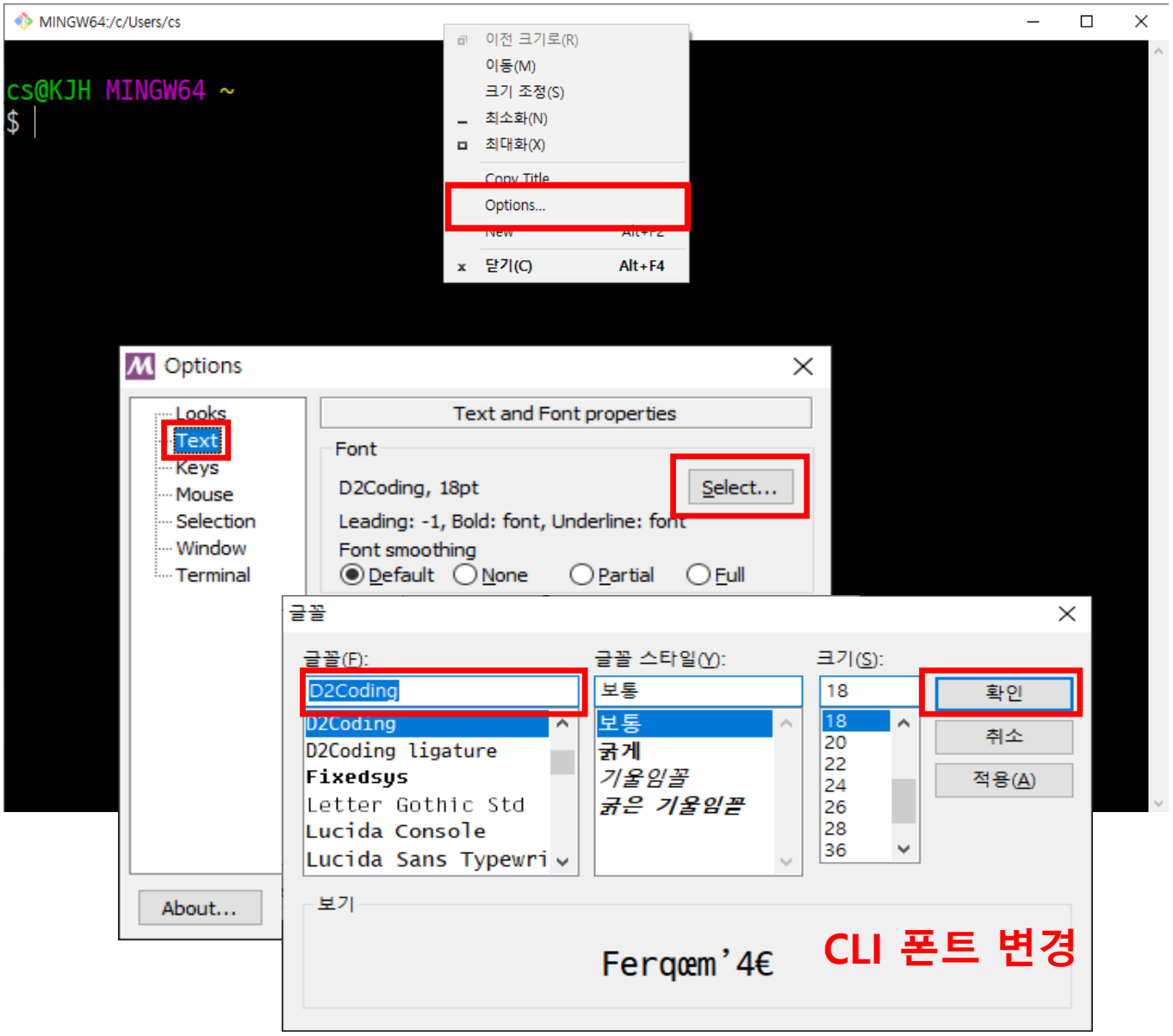


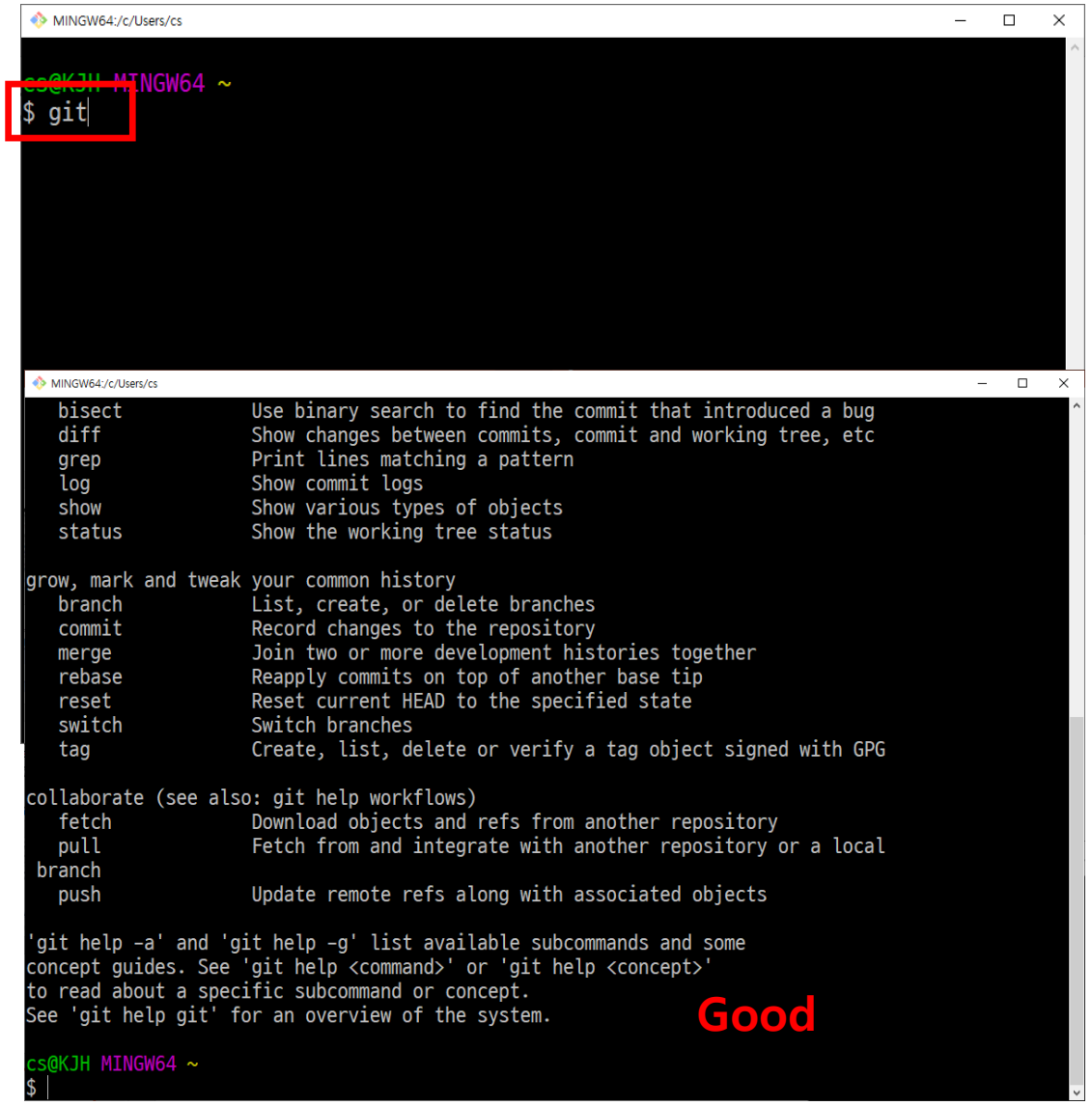





Vim -> Visual Studio Code
Git 2.28.0 Setup
Choosing the default editor used
Which editor would you like Git to u
Use Visual Studio Code as Git's de
(NEW!) Visual Studio Code is
running as a desktop applicat
TypeScript and Node.js and h
languages (such as C++, C#
.NET and Unity).
(WARNING!) This will be insta
Use this option to let Git use
https://gitforwindows.org/
< Back
Next >
Cancel
Git 2.28.0 Setup
Adjusting your PA
How would you li
Use Git fro
This is the
only be abl
Git from th
(Recommer
PATH to av
You will be
PowerShell
Use Git and
Both Git an
Warning: T
use this op
https://gitforwindows.org/
< Back
Next >
Cancel
Git 2.28.0 Setup
Choosing HTTPS tra
Which SSL/TLS librar
Use the Open
Server certific
Use the nativ
Server certific
This option als
distributed e.g
https://gitforwindows.org/
< Back
Next >
Cancel
Git 2.28.0 Setup
Configuring the lin
How should Git tre
Checkout W
Git will conve
text files, CR
this is the re
Checkout as
Git will not p
committing t
this is the re
Checkout as
Git will not p
text files. Ch
projects ("co
https://gitforwindows.org/
< Back
Next >
Cancel
Git 2.28.0 Setup
Configuring the terr
Which terminal emul
Use MinTTY (
Git Bash will us
non-rectangul
as interactive
Use Windows
Git will use the
with Win32 co
very limited de
order to displa
window was n
https://gitforwindows.org/
< Back
Next >
Cancel
Git 2.28.0 Setup
Choose the default behavior of `git pull`
What should `git pull` do by default?
Default (fast-forward or merge)
This is the standard behavior of `git pull`: fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.
Rebase
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.
Only ever fast-forward
Fast-forward to the fetched branch. Fail if that is not possible.
https://gitforwindows.org/
< Back
Next >
Cancel

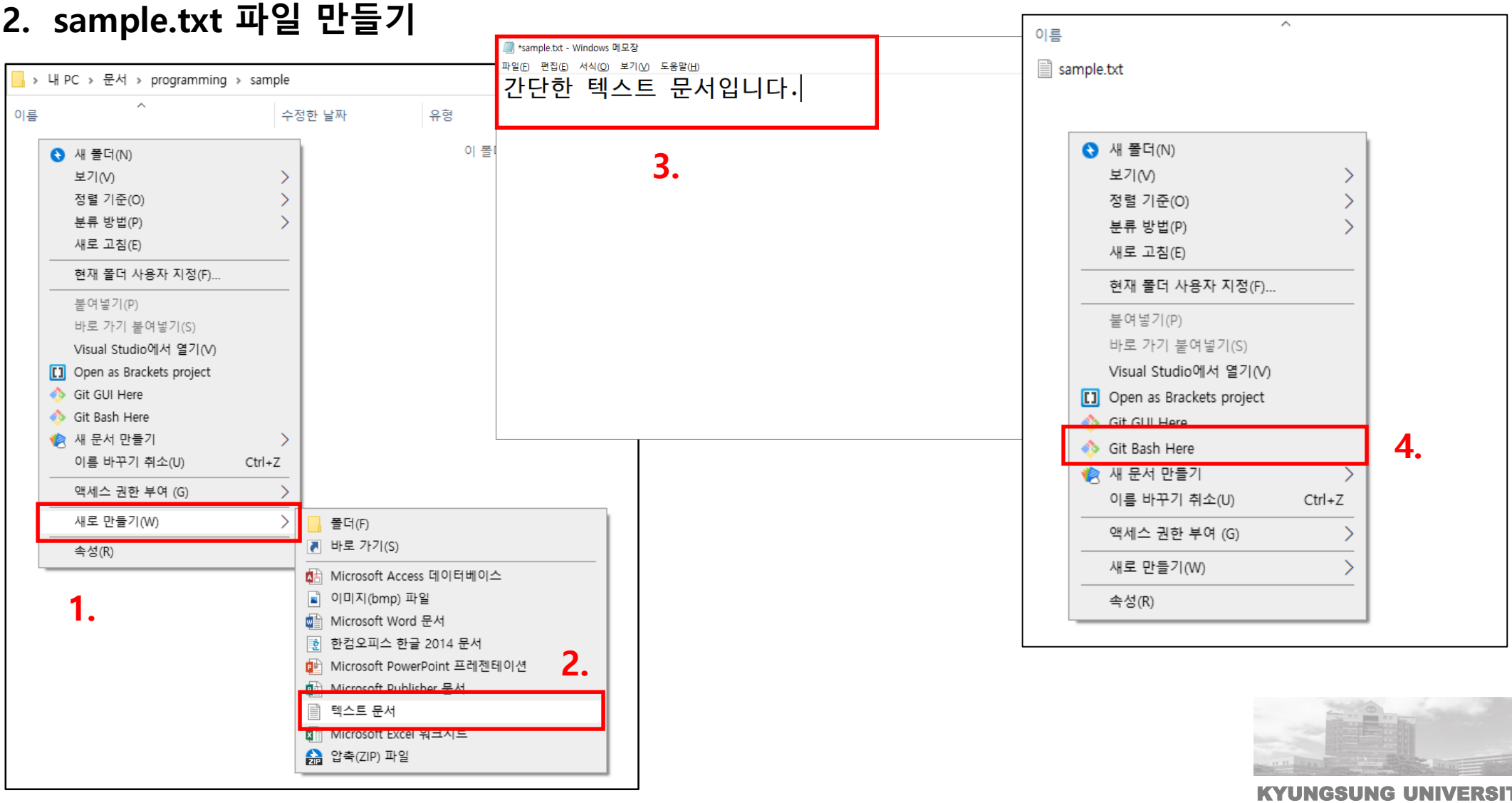

Git 2.28.0 Setup
Completing the Git Setup Wizard
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Git Bash
View Release Notes
Next >
Git 2.28.0 Setup
Installing
Please wait while S
Extracting files...
https://gitforwindows.org/
Cancel
Git 2.28.0 Setup
Configuring experi
Which bleeding-ed
Enable expe
(NEW!) This
Git Bash win
https://gitforwindows.org/
< Back
Install
Cancel
Git 2.28.0 Setup
Configuring extra
Which features w
Enable file s
File system
operations
performanc
Enable sym
Enable syml
Please note
https://gitforwindows.org/
< Back
Next >
Cancel
Git 2.28.0 Setup
Choose a credent
Which credential
None
Do not use
Git Credent
The Git Cre
DevOps an
Git Credent
(NEW!) Use
See more in
https://gitforwindows.org/
< Back
Next >
Cancel


MINGW64:/c/Users/cs
cs@KJH MINGW64 ~
$
이전 크기로(R)
이동(M)
크기 조정(S)
최소화(N)
최대화(X)
Copy Title
Options...
닫기(C) Alt+F4
Options
Looks
Text
Keys
Mouse
Selection
Window
Terminal
About...
Text and Font properties
Font
D2Coding, 18pt
Select...
Leading: -1, Bold: font, Underline: font
Font smoothing
Default None Partial Full
글꼴
글꼴(F):
D2Coding
D2Coding
D2Coding ligature
Fixedsys
Letter Gothic Std
Lucida Console
Lucida Sans Typewri
글꼴 스타일(Y):
보통
보통
굵게
기울임꼴
굵은 기울임꼴
크기(S):
18
18
20
22
24
26
28
36
확인
취소
적용(A)
보기
Ferqœm'4€
CLI 폰트 변경


MINGW64:/c/Users/cs
cs@KJH MINGW64 ~
$ git
MINGW64:/c/Users/cs
bisect Use binary search to find the commit that introduced a bug
diff Show changes between commits, commit and working tree, etc
grep Print lines matching a pattern
log Show commit logs
show Show various types of objects
status Show the working tree status
grow, mark and tweak your common history
branch List, create, or delete branches
commit Record changes to the repository
merge Join two or more development histories together
rebase Reapply commits on top of another base tip
reset Reset current HEAD to the specified state
switch Switch branches
tag Create, list, delete or verify a tag object signed with GPG
collaborate (see also: git help workflows)
fetch Download objects and refs from another repository
pull Fetch from and integrate with another repository or a local branch
push Update remote refs along with associated objects
'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
Good
cs@KJH MINGW64 ~
$

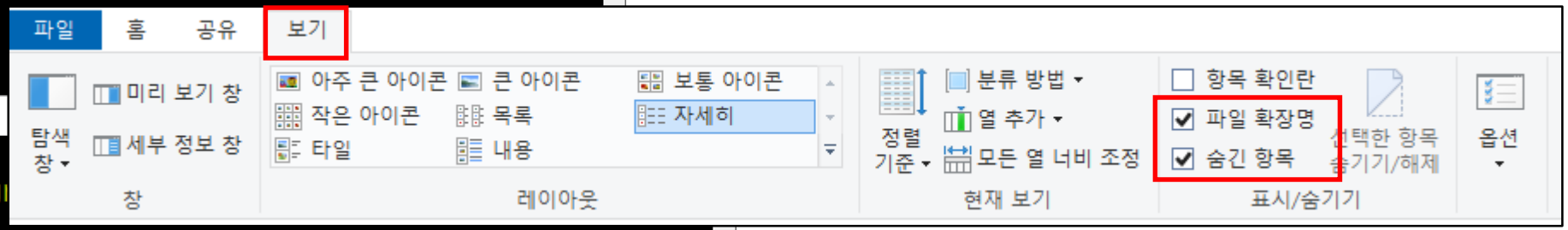
3. 로컬 저장소 만들기

1. 내 컴퓨터 – 문서 – programming – sample 폴더 생성
2. sample.txt 파일 만들기

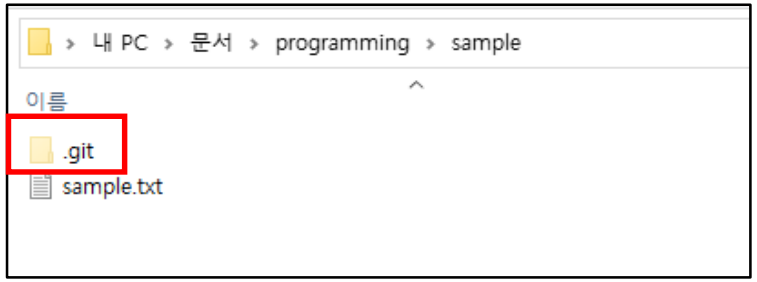


Git 초기화 과정 – 로컬저장소 생성

```
MINGW64:/c/Users/cs/Documents/programming/sample

cs@KJH MINGW64 ~/Documents/programming/sample
$ git init
Initialized empty Git repository in C:/Users/cs/Documents/programming/
sample/.git/

cs@KJH MINGW64 ~/Documents/programming/sample (master)
$
```




```
MINGW64:/c/Users/cs/Documents/programming/sample

cs@KJH MINGW64 ~/Documents/progr
$ ls -al
total 5
drwxr-xr-x 1 cs 197121  0 10월  6 17:18 ./
drwxr-xr-x 1 cs 197121  0 10월  6 17:07 ../
drwxr-xr-x 1 cs 197121  0 10월  6 17:18 .git/
-rw-r--r-- 1 cs 197121 36 10월  6 17:11 sample.txt

cs@KJH MINGW64 ~/Documents/programming/sample (master)
$
```

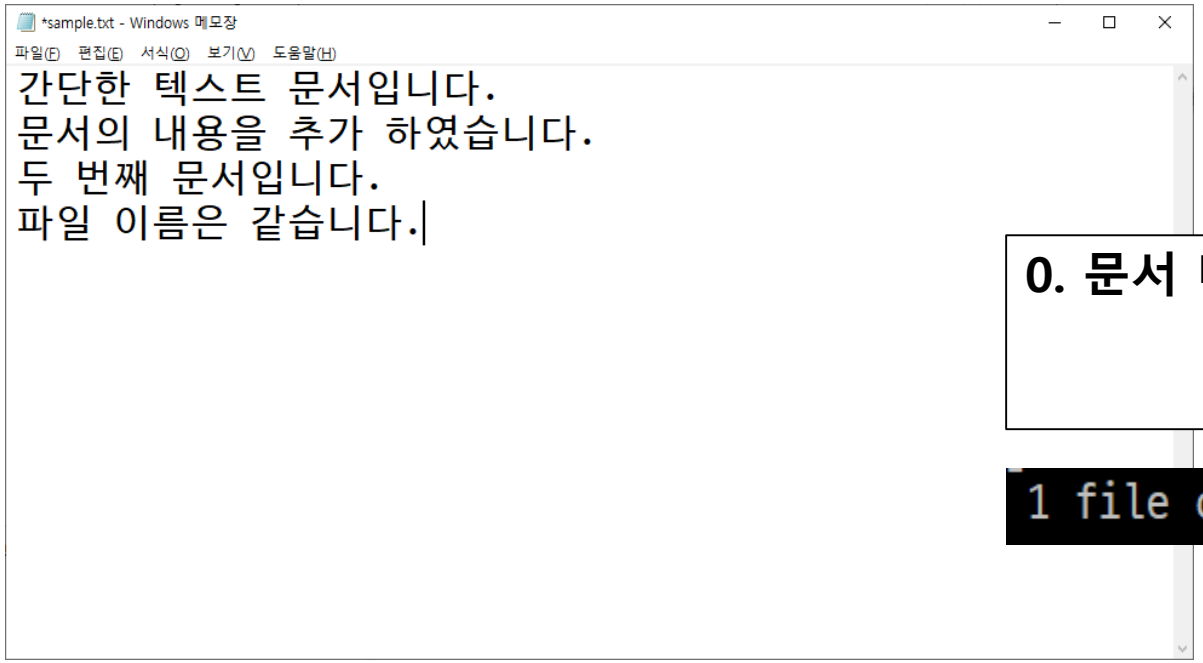


4. 첫 번째 commit 만들기

0. 윈도우 탐색기에서 Git Bash Here
   1. $ git config --global user.email “GitHub 회원가입 때 사용한 이메일“
   2. $ git config --global user.name “GitHub 회원가입 때 사용한 ID”
   3. $ git add sample.txt
   4. $ git commit -m “처음 만든 텍스트 파일 "

```
1 file changed, 1 insertion(+)
create mode 100644 sample.txt
```

5. 두 번째 commit 만들기



0. 문서 내용 변경

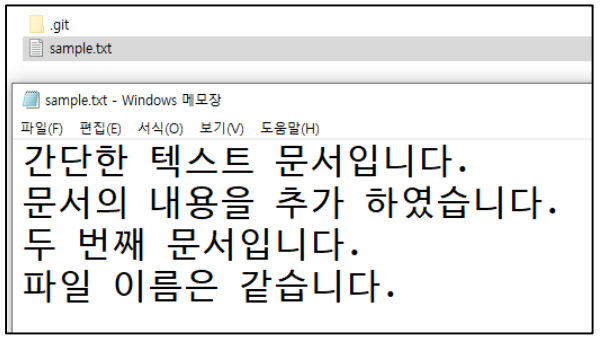
1. $ git add sample.txt

2. $ git commit –m “문서 내용 추가 "

```
1 file changed, 4 insertions(+), 1 deletion(-)
```

6. 다른 commit으로 이동하기

```
$ cat sample.txt
간단한 텍스트 문서입니다.
문서의 내용을 추가 하였습니다.
두 번째 문서입니다.
파일 이름은 같습니다.
```

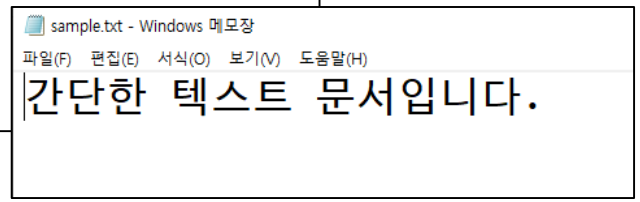


1. $ git log
   commit d7516eb0faadd89005f3e06352d0cfc98f2db727 (HEAD -> master)
   문서 내용 추가
   commit ae628ffbed01c585ab5a1cc19d1a69ac658692b1
   처음 만든 텍스트 파일

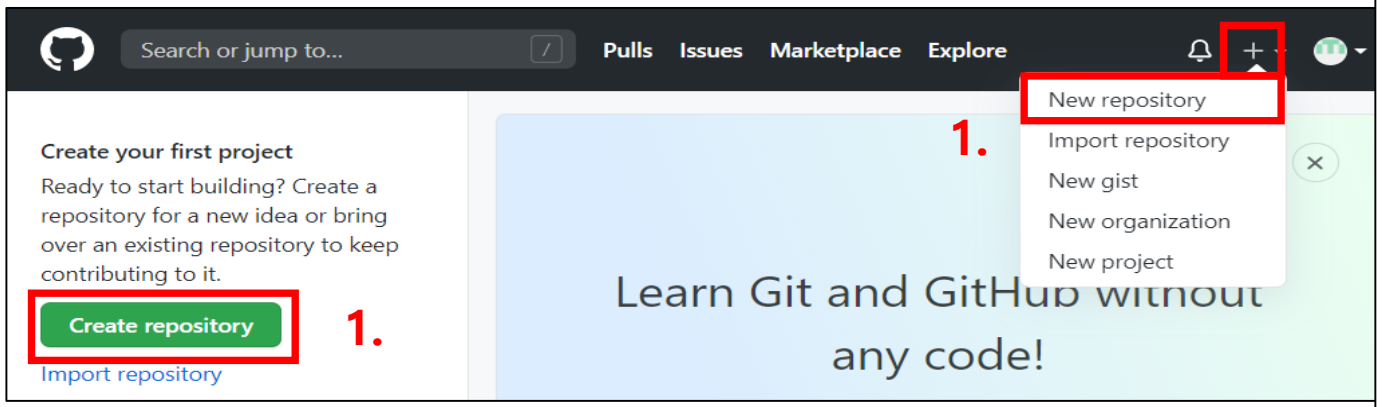
2. $ git checkout ae628ff 7자리 커밋 아이디
   Head is now at ae628ff 처음 만든 텍스트 파일

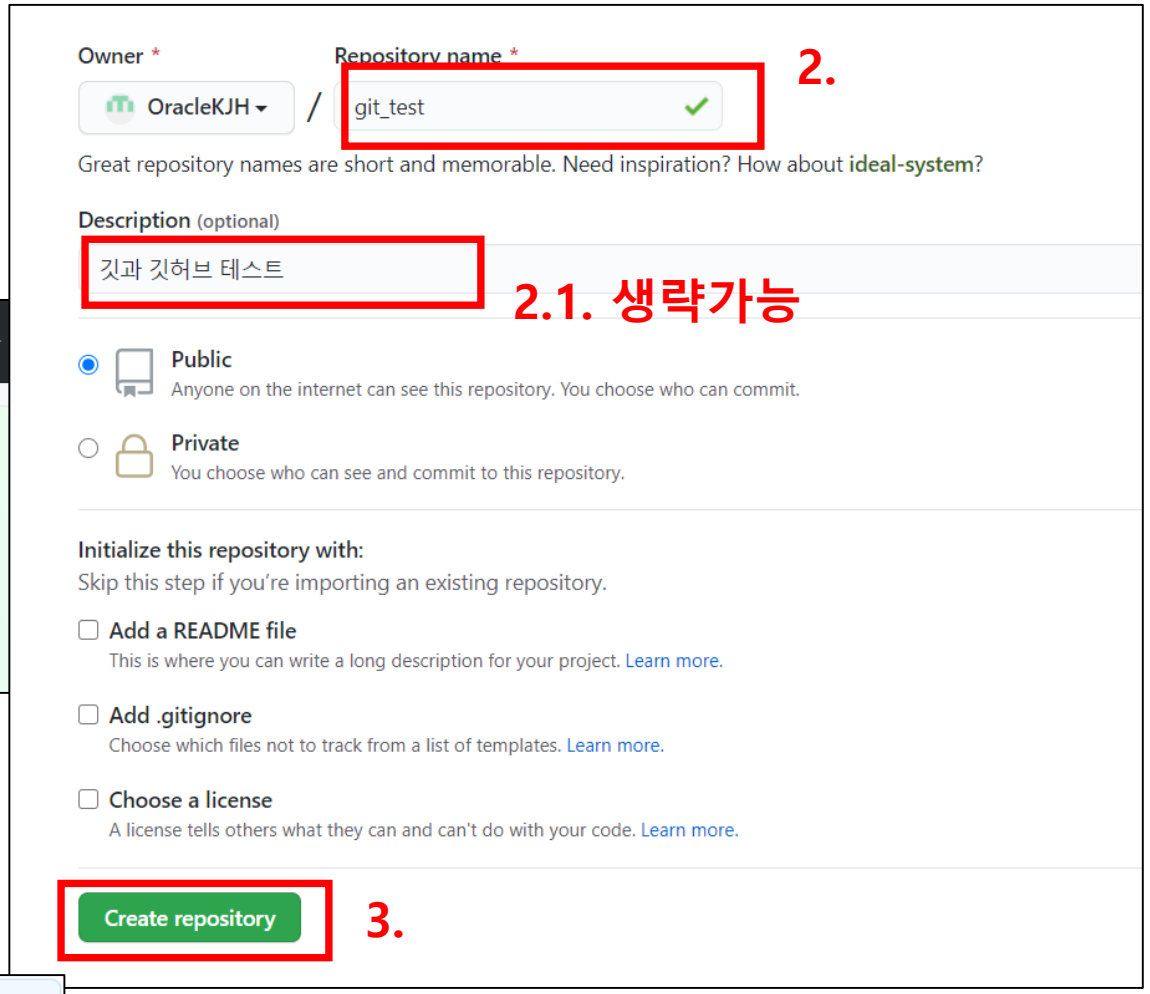
3. $ git checkout d7516eb 혹은 $git checkout -
   최신 커밋

```
$ cat sample.txt
간단한 텍스트 문서입니다.
```

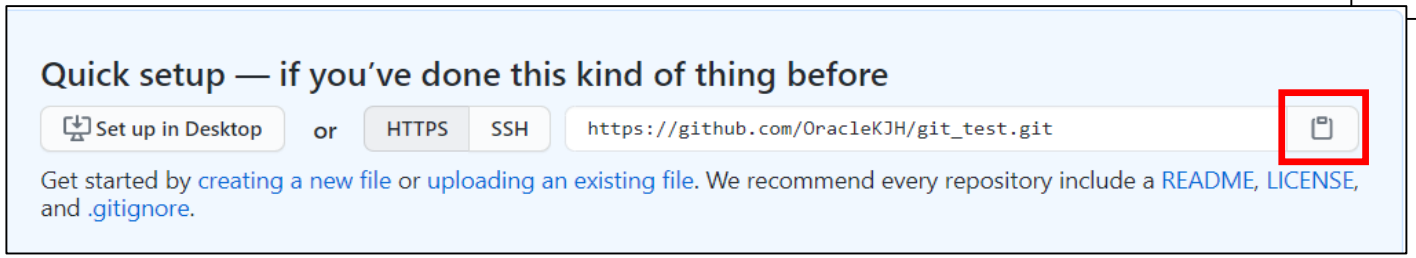


7. GitHub 원격저장소에 commit 올리기



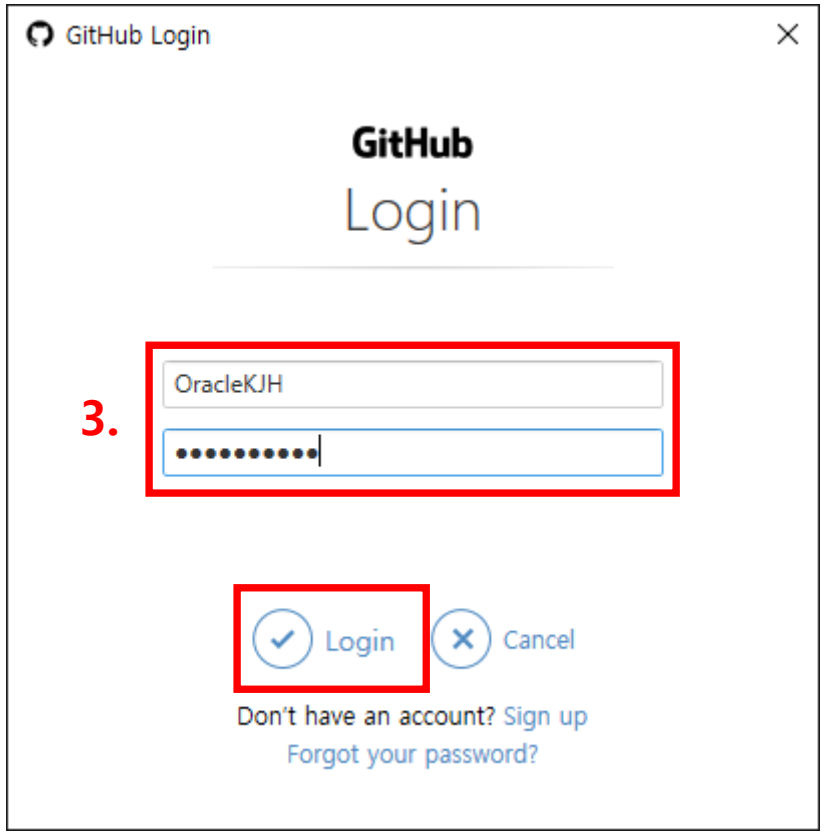


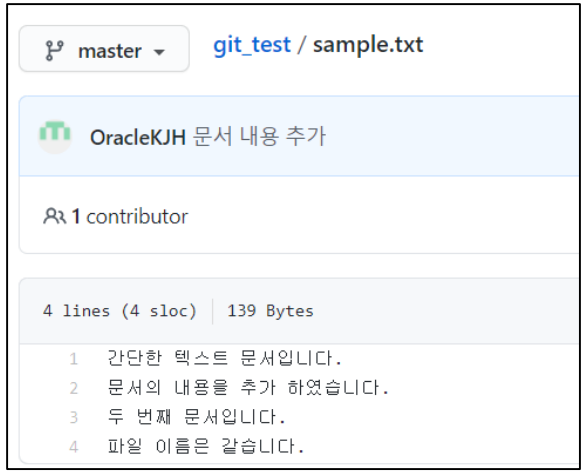
4. 원격저장소 주소 복사(다른 사람들과 공유)

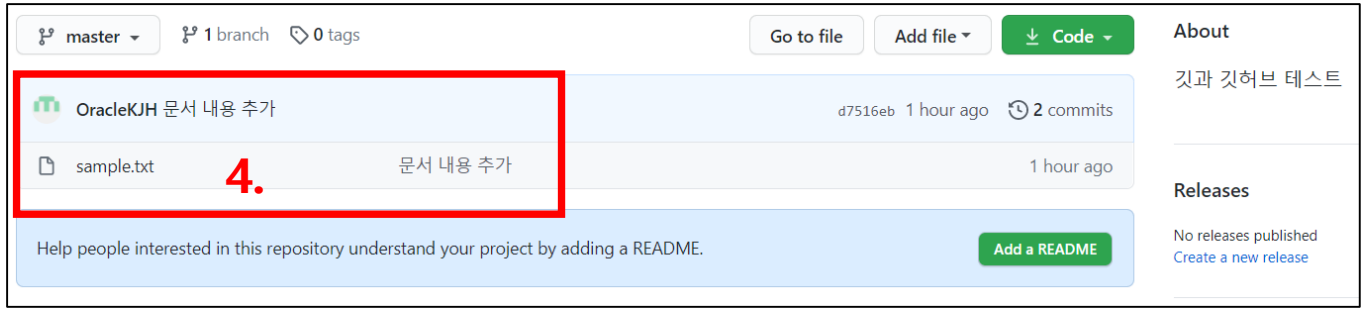


1.
```
cs@KJH MINGW64 ~/Documents/programming/sample ((d7516eb...))
$ git remote add origin https://github.com/OracleKJH/git_test.git
```

2.
```
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 641 bytes | 320.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/OracleKJH/git_test.git
 * [new branch]      master -> master

cs@KJH MINGW64 ~/Documents/programming/sample ((d7516eb...))
$
```







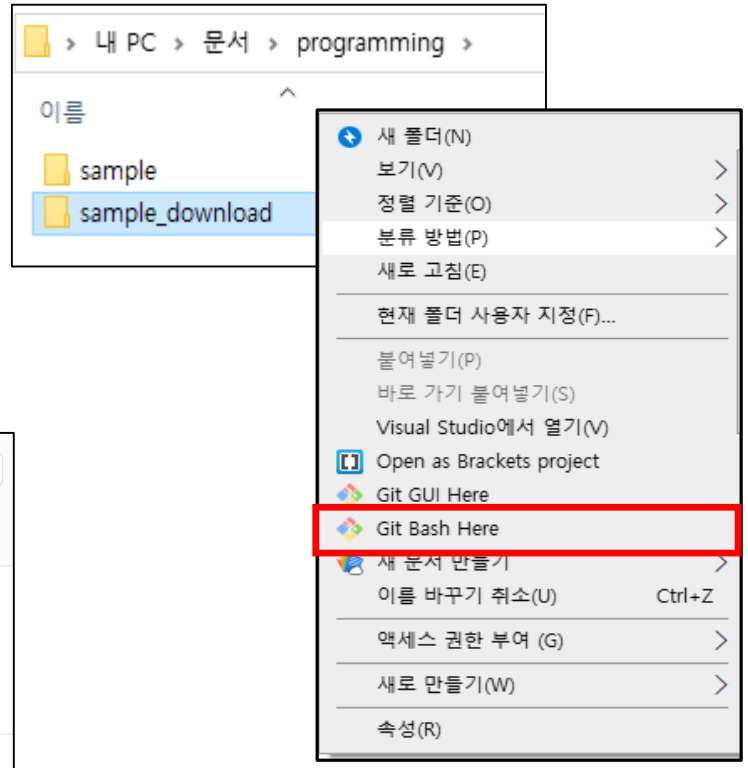
8. GitHub 원격저장소의 commit을 로컬 저장소에 다운로드

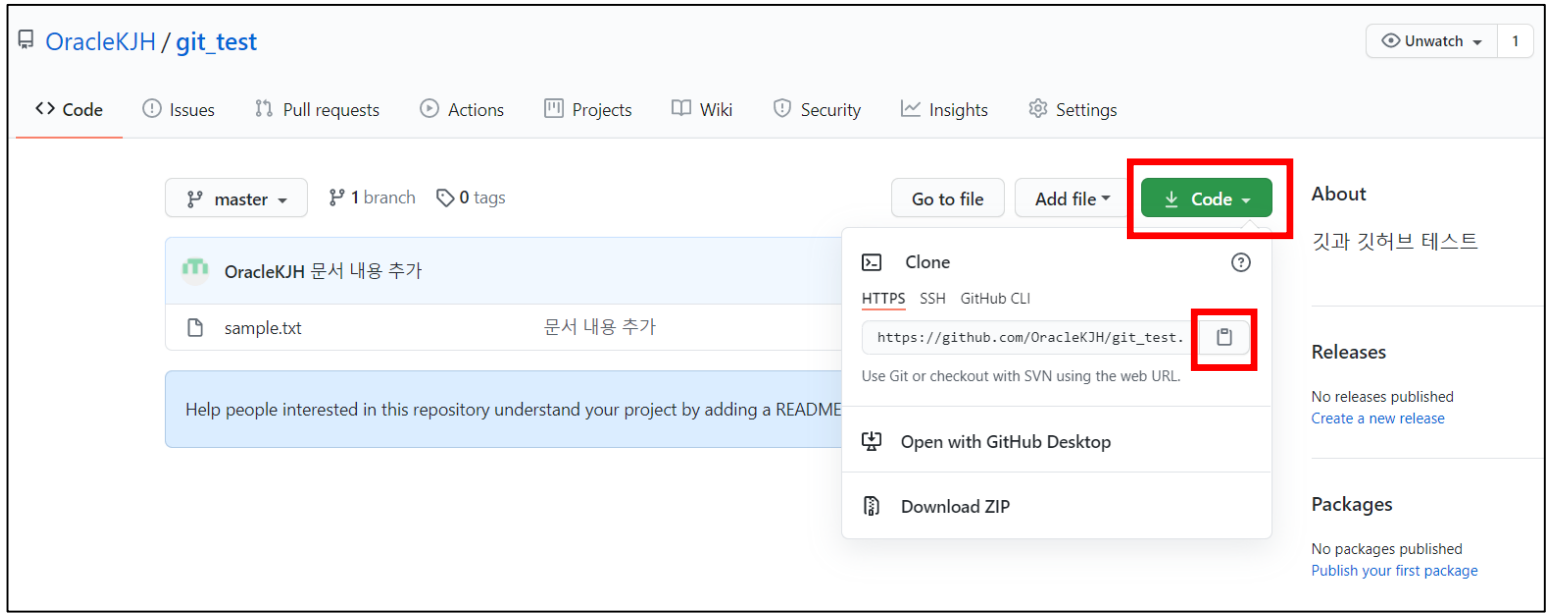
내 컴퓨터 – 문서 - programming – sample_download 폴더 생성

Sample_download 더블클릭

Git Bash Here

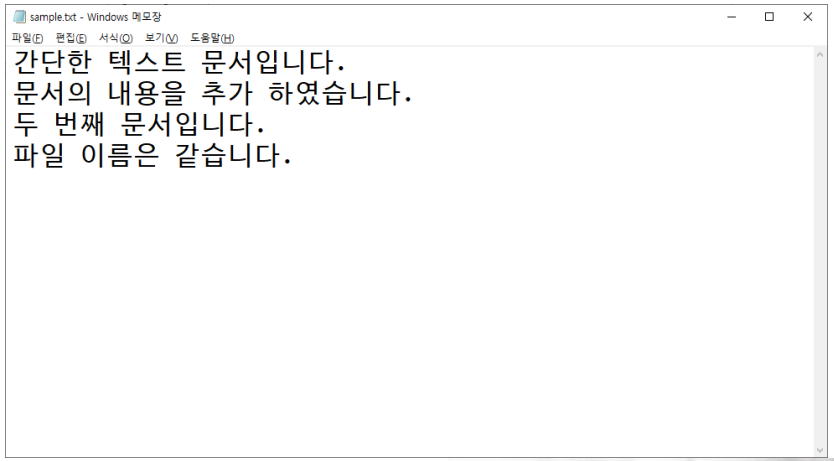
GitHub : Code – Clone – URL 복사



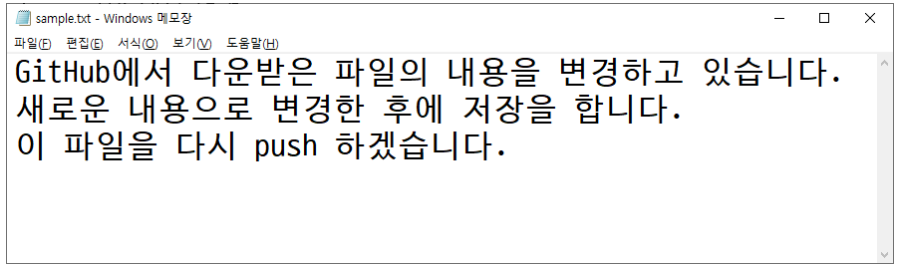


$ git clone https://github.com/OracleKJH/git_test.git .

```
cs@KJH MINGW64 ~/Documents/programming/sample_download
$ git clone https://github.com/OracleKJH/git_test.git .
Cloning into '.'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 621 bytes | 1024 bytes/s, done.

cs@KJH MINGW64 ~/Documents/programming/sample_download (master)
$
```



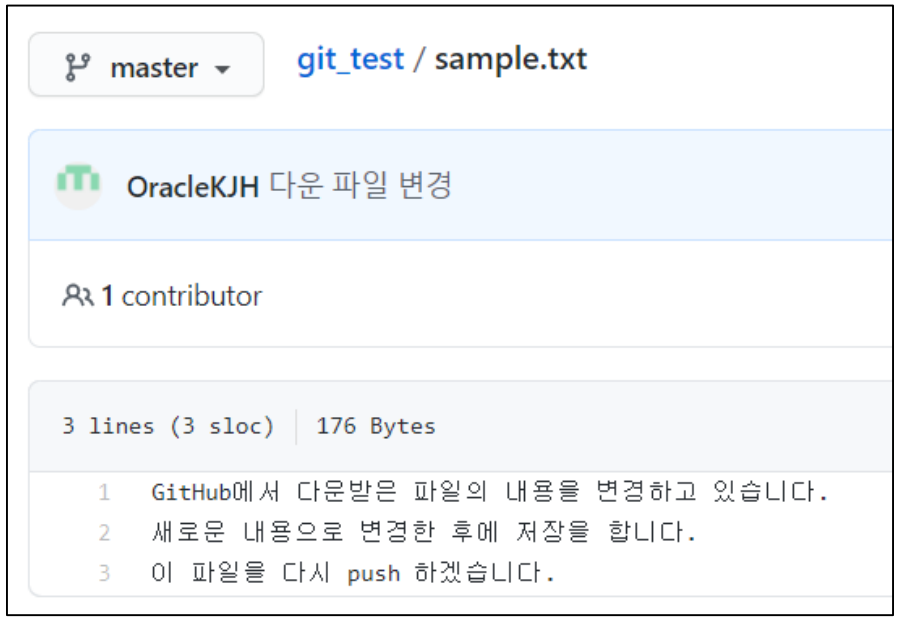
1. sample.txt 파일 내용 변경



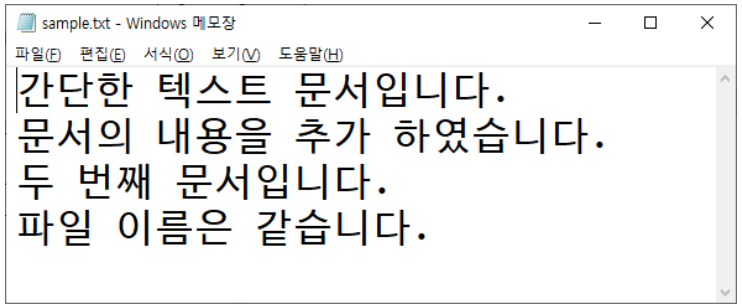
2. $ git add samplet.txt

3. $ git commit –m “다운 파일 변경“

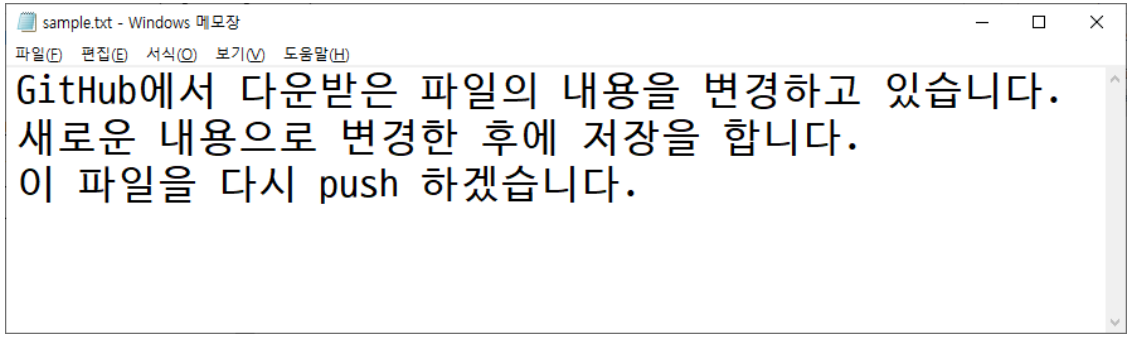
4. $ git push origin master



1. 내 컴퓨터 – 문서 – programming – sample 폴더로 이동
2. sample.txt 파일 내용 확인
3. Git Bash Here
4. $ git pull origin master
5. Sample.txt 파일 내용 확인




```
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), 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), 394 bytes | 3.00 KiB/s, done.
From https://github.com/OracleKJH/git_test
 * branch            master     -> FETCH_HEAD
   d7516eb..14ac64b  master     -> origin/master
Updating d7516eb..14ac64b
Fast-forward
 sample.txt | 7 +++----
 1 file changed, 3 insertions(+), 4 deletions(-)

cs@KJH MINGW64 ~/Documents/programming/sample ((14ac64b...))
$ cat sample.txt
GitHub에서 다운받은 파일의 내용을 변경하고 있습니다.
새로운 내용으로 변경한 후에 저장을 합니다.
이 파일을 다시 push 하겠습니다.
cs@KJH MINGW64 ~/Documents/programming/sample ((14ac64b...))
$
```



Reference

팀 개발을 위한 Git GitHub 시작하기, 정호영, 진유림, 한빛미디어