

Git이란 무엇인가?



- ✓ 형상 관리 도구
(Configuration Management Tool)
≡ 버전 관리 시스템
- ✓ 소스코드를 효과적으로
관리하게 할 수 있음
- ✓ 공개 소프트웨어(Free)

Git이란,

: 버전을 편리하게 관리할 수 있도록 도와주는 도구(Version Control System)
: 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다. (Distributed Version Control)

: Git은 명령어를 기본으로 한 명령어 프로그램이다. (ex) git 명령어 -option)
하지만, GUI를 제공한다.(공식사이트에서 확인가능)



ex) Github Desktop

SourceTree- Atlassian에서 만든 GUI 널리 쓰이고 있다 ,
GitKraken-화려한UI

: GUI는 터미널에서 명령어로 지원되는 모든 기능을 제공하지는 못한다.
또한, 버튼 클릭으로 이루어지는 기능이 어떠한 일을 하는지 알 수 없기 때문에
처음 git 연습할 때는 명령어기반으로 학습해보는 것을 권장한다.
: 각 운영체제에서 지원하는 터미널을 이용해도 되고 mac에서는 iterm2 , window
에서는 Cmder의 좀더 편리한 터미널을 설치해서 사용해도 된다.

그렇다면,

버전 관리란 무엇인가?

어떠한 의미 있는 변화 즉, 기능의 개선, 버그의 수정, 고객의 요구사항의 변경
같은 것들을 말한다. 이러한 변화들을 관리하는 체계, 시스템을 **버전관리 시스템**
이라고 한다.

Git이 필요한 이유는?

두명의 개발자가 하나의 서비스를 만든다고 생각해보자.

1. 각자 이름의 폴더를 만들고 각자 개발하다가 메일 or 카톡으로 보내서
합친다.
2. 혹시 에러 날 수 있으니 백업본을 만들어 둔다.
3. source_ver1.zip / source_ver2.zip ...
4. 상대방이 작업중인 파일을 고치고 싶으면 메일에 따로 적어준다.
5. 코드를 합칠 때 이를 확인해서 내 코드에도 반영한다.

따로 조금씩 작업하다 내가 원할 때 코드를 합칠 수 있는 방법이 없나?

백업도 좀 쉽게 하고...→**“있다”** 현재는 Git 대세!!!

VCS의 대표적인 흐름을 주도했던 제품들로는

- 매트맥컬이 개발한 Mercurial 2005년도
- 아파치사에서 개발한 SVN
- 리눅스 개발자인 리누스토르발스가 만든 Git

*** Git이 다른 분산시스템과 다르게 가지는 가장 대표적인 특징은**

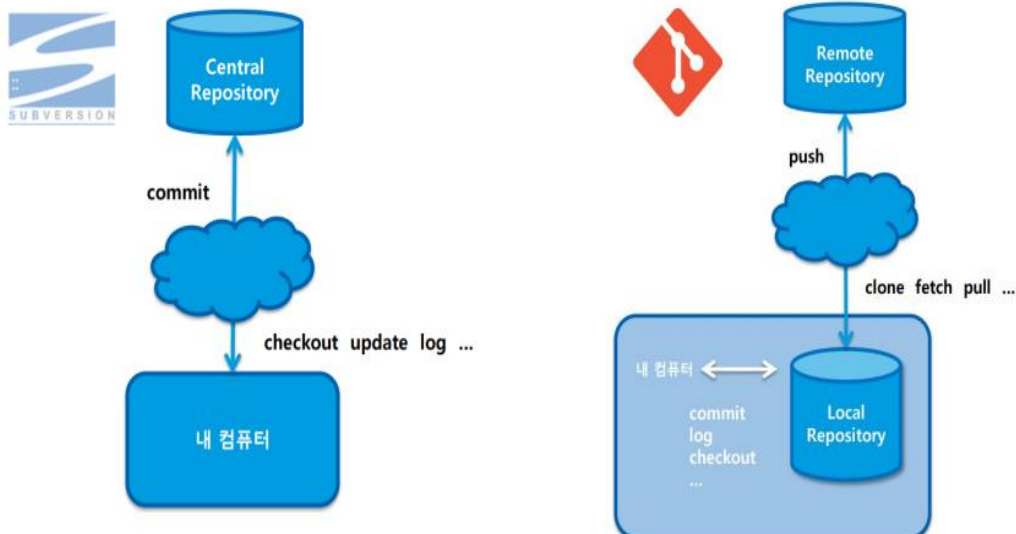
Master 저장소 서버(Remote Storage) 와 Master저장소의 완전한 사본을 가지고 있는 클라이언트 저장소(Local Storage)로 이뤄져 있다는 것이다.

SVN(subversion) vs Git



SVN	Git
하나의 repository에 여러 프로젝트 구성 가능	보통 하나의 repository에 하나의 프로젝트 구성 (빌드 단위)
중앙 집중형 (Centralized)	분산형 (Distributed)
	대부분 작업이 로컬에서 수행 (Svn 대비 속도가 빠름)
디렉토리/파일별 권한 부여 가능	개별 권한 부여 불가
리비전(revision) 사용	Object Hash 사용
사용이 쉬움 (적은 명령, 자유도 낮음)	학습 곡선(Learning curve)

Centralized vs Distributed



Github란 무엇인가?



- ✓ 형상 관리 도구 (Git)의 웹 호스팅(Web hosting) 서비스
- ✓ 협업하는 코드를 저장하는 서버
- ✓ 협업 이벤트들(push,pull,request 등)에 반응하여 자동으로 작업을 실행할 수 있게 함

GitHub은

깃허브(GitHub, 원래 이름-Logical Awesome LLC)는 분산 버전 관리 툴인 깃 (Git) 저장소 호스팅을 지원하는 웹 서비스이다.

Git으로 버전 관리한 코드를 올릴 수 있는 클라우드 서버

Git 호스팅 사이트	모기업	특징	가격
GitHub	GitHub Inc (Microsoft에서 인수)	사용자 2,800만의 세계 최대 규모의 Git 호스팅 사이트	<ul style="list-style-type: none"> • 공개저장소 무료, • 비공개 저장소는 3인 이하인 경우는 무료 • 설치형 버전인 엔터프라이즈는 월21달러
GitLab	GitLab Inc	NASA, Sony 등 10만개 이상의 조직이 사용. GitLab 자체가 오픈소스인 특징	<ul style="list-style-type: none"> • 공개, 비공개 저장소 생성 무료
BitBucket	Atlassian	사용자 600만명. 지라(Jira)와 연동 쉽다	<ul style="list-style-type: none"> • 5명 이하 팀이면 공개 및 비공개 저장소 생성 무료



Git의 장점



✓ 같은 파일을 여러 명이 동시에 작업하는
병렬 개발이 가능



✓ 분산 버전관리 방식
 > 인터넷 연결 유무와 상관없이 개발 진행 가능
 > 데이터 원상복구 용이

1) 코딩할 때 단순히 ctrl+z를 눌러 이전 상태로 되돌리는 것이 아니라, 원하는 시점마다 버전을 만들고 이들 간에 자유롭게 돌아다닐 수 있다.

2) 내가 만든 버전 뿐 아니라 동료가 만든 버전으로 이동할 수 있고, 동료와 내 버전을 비교해서 최신본으로 코드를 업데이트를 할 수 있다.

Git을 사용하는 방법 2가지

1) CLI 방식

```

MINGW64:/c
KOSTA@DESKTOP-7PLAKSP MINGW64 /c
$ git
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          [--config-env=<name>=<envvar>] <command> [<args>]

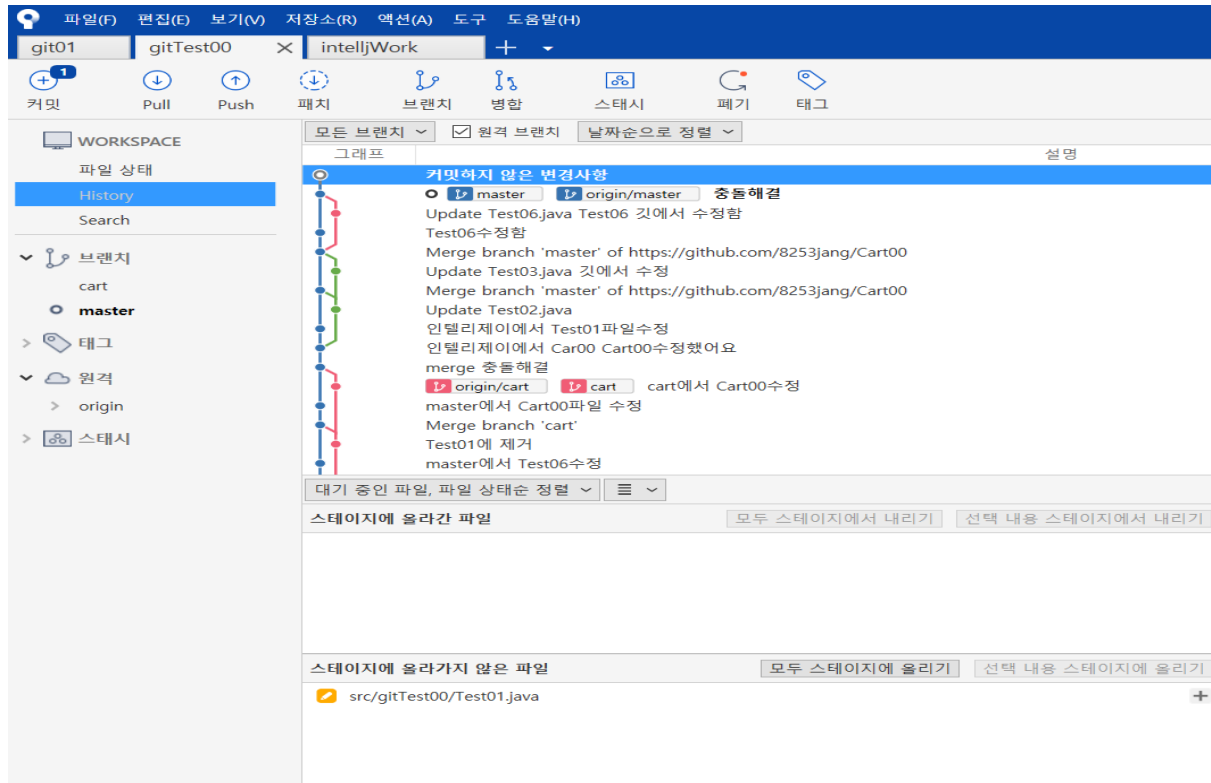
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  
```

2) GUI 방식



Git을 시작 해 보자.

1) 회원가입하기

<https://github.com/>

2) Git 설치하기

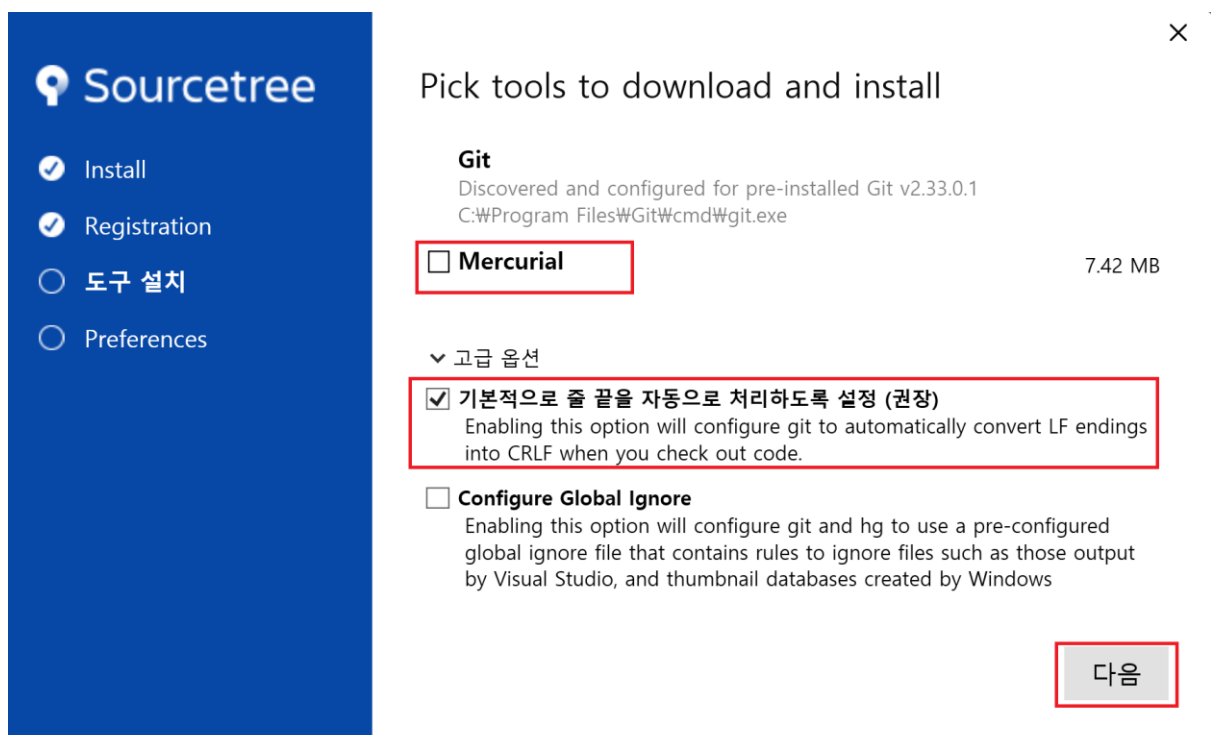
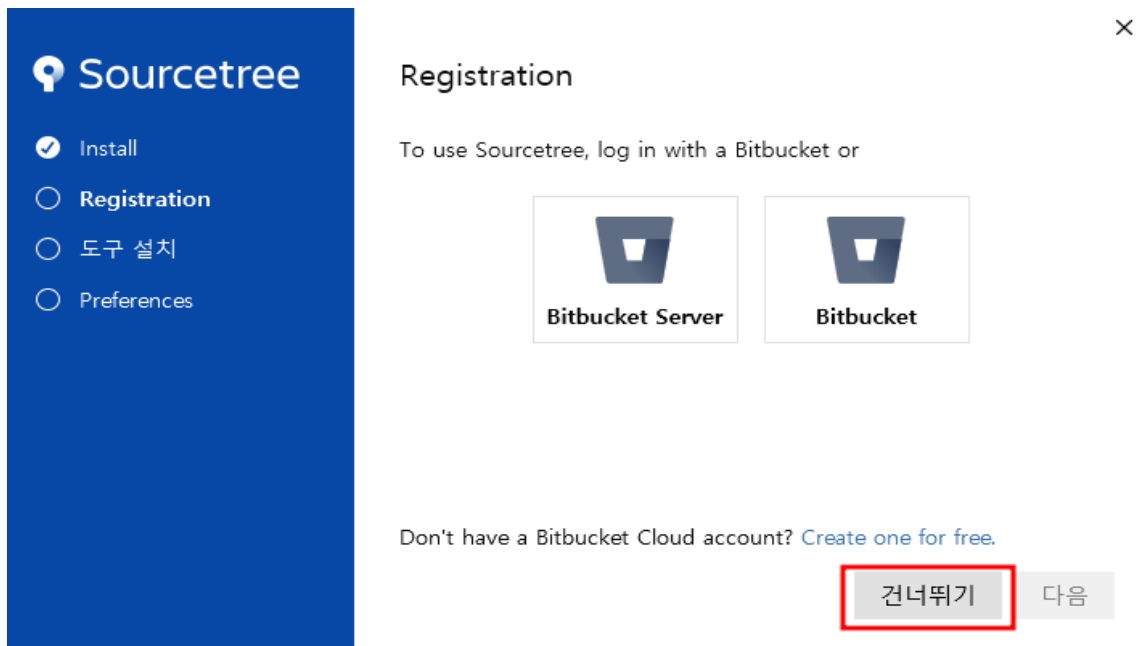
<https://git-scm.com/>

: 설치시 기본으로 설치하면 된다.

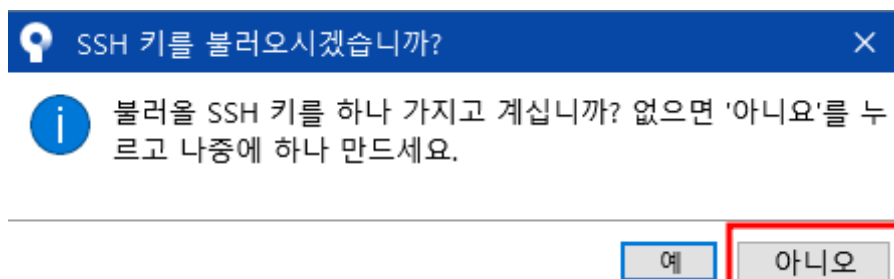
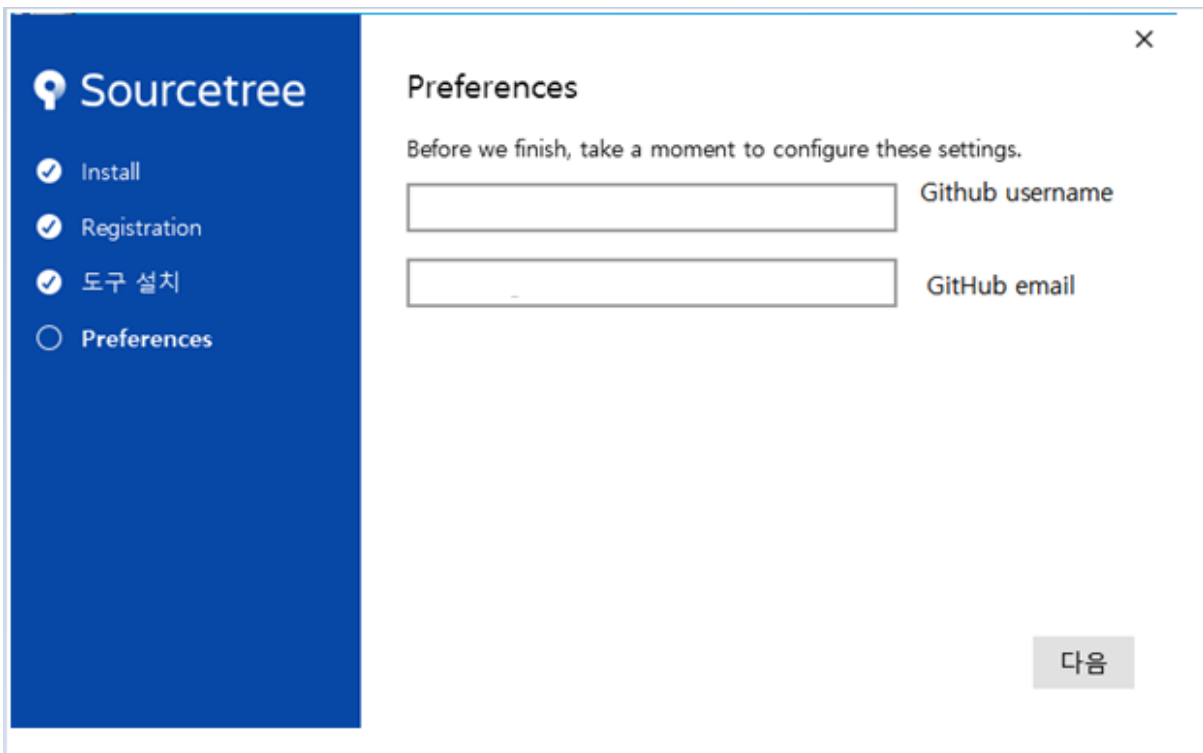
참고 사이트 - <https://taewow.tistory.com/13>

3) Source Tree 설치하기

<https://www.sourcetreeapp.com/>



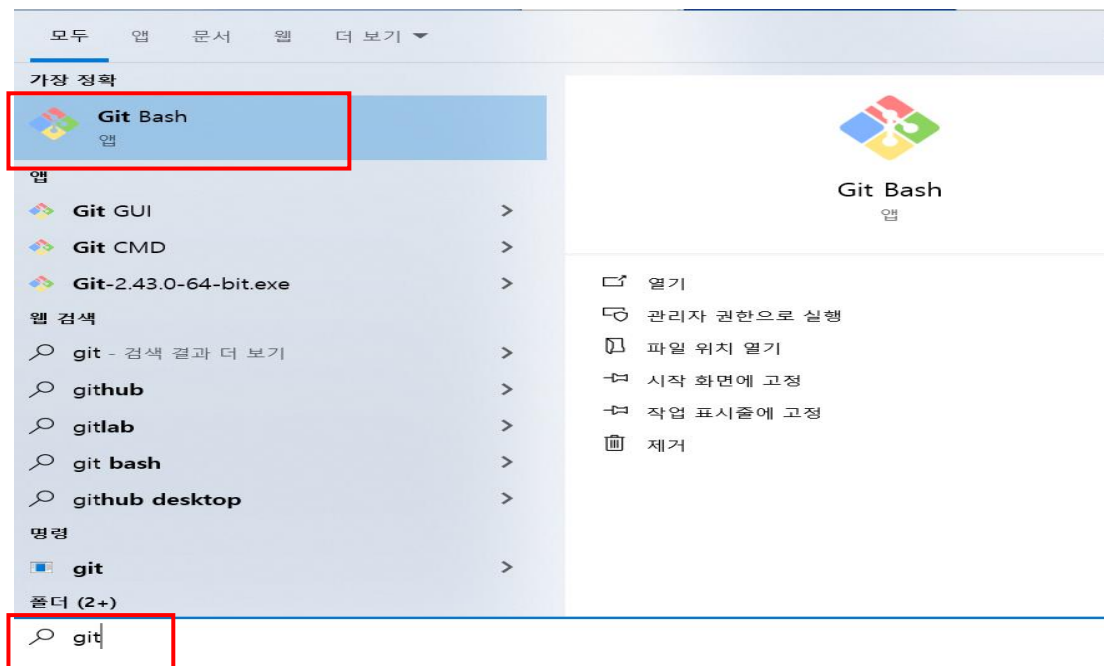
* Mercurial 선택하지 않는다.



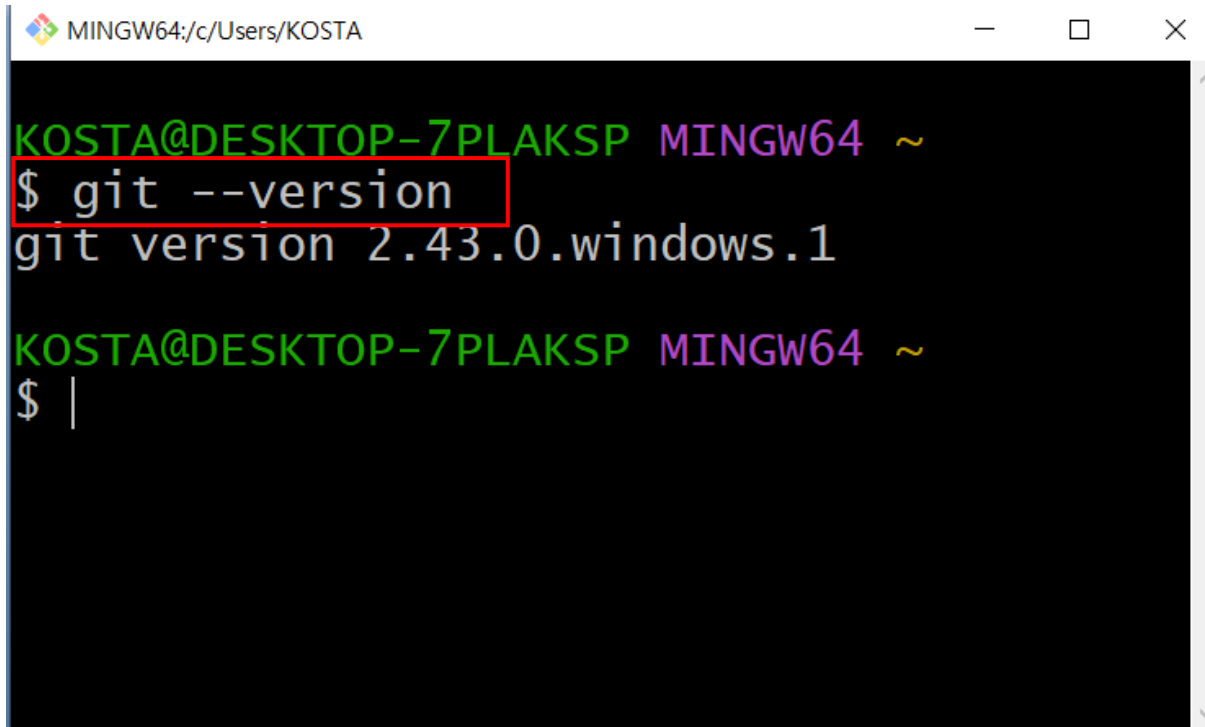
설치 완료 후 확인해보자!

1) 검색 창에서 git을 검색한다.

2) Git Bash를 선택한다.



3) git -version 입력하고 엔터

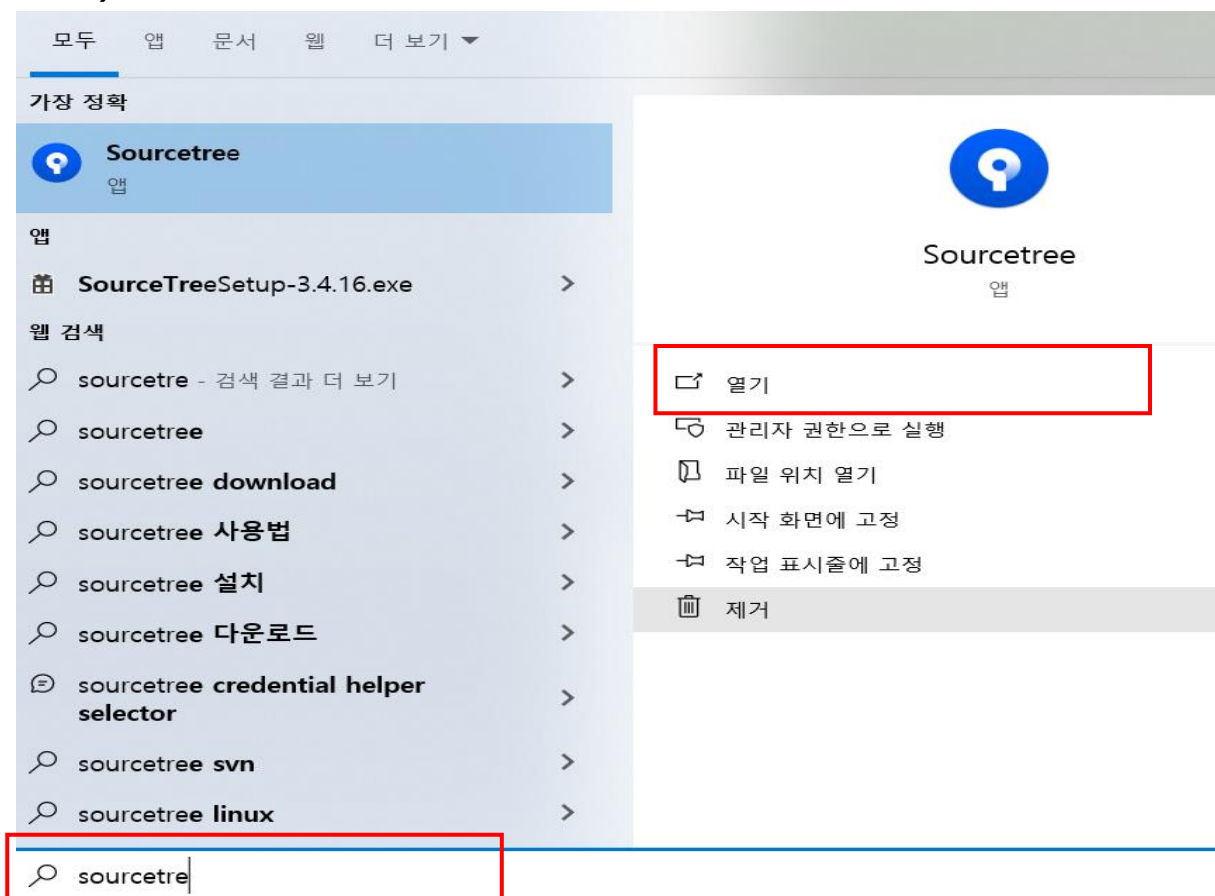


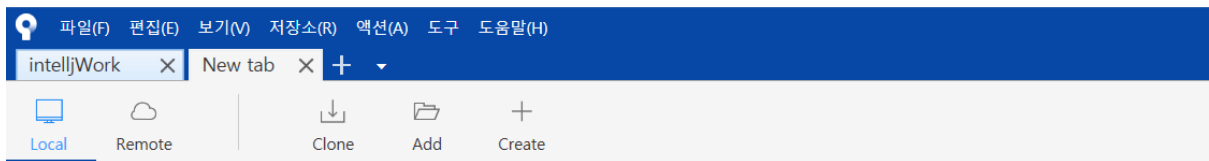
```
MINGW64:/c/Users/KOSTA

KOSTA@DESKTOP-7PLAKSP MINGW64 ~
$ git --version
git version 2.43.0.windows.1

KOSTA@DESKTOP-7PLAKSP MINGW64 ~
$ |
```

4) 검색 창에 sourcetree 검색 하고 열기를 클릭한다.





로컬 저장소

검색