

Lecture #2: Outline of git

Software Engineering laboratory
Yeungnam University



SW 개발시 git의 필요성

■ (1) 변경 내역을 확인하기 어려움

● (1-1) 소스코드 수정시 하나의 파일에 저장하는 단순 방식

- 대개 파일을 단순히 저장하면 이전에 저장된 내용에서 현재 내용으로 덮어씀.
- 즉, 저장된 파일은 항상 최신 상태만 갖게 됨.
- 이런 방식으로는 현재 저장된 내용이 이전에 비해 무엇이 어떻게 달라졌는지 알기 어려움.
- 다시 말해, 변경 내역을 추적하기가 어려움.

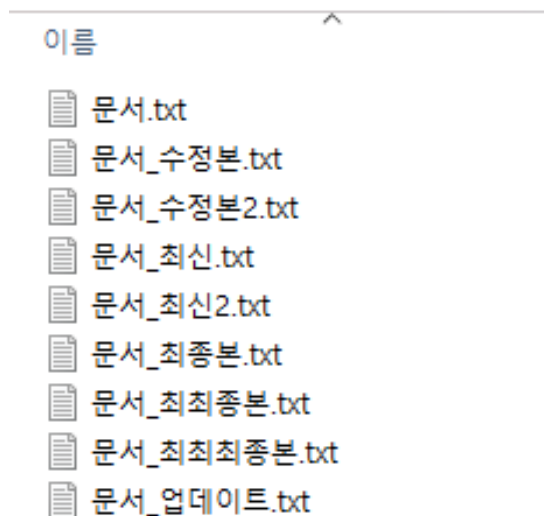


SW 개발시 git의 필요성

■ (1) 변경 내역을 확인하기 어려움

● (1-2) 소스코드 수정시 다른 이름으로 여러 개의 파일에 저장하는 방식

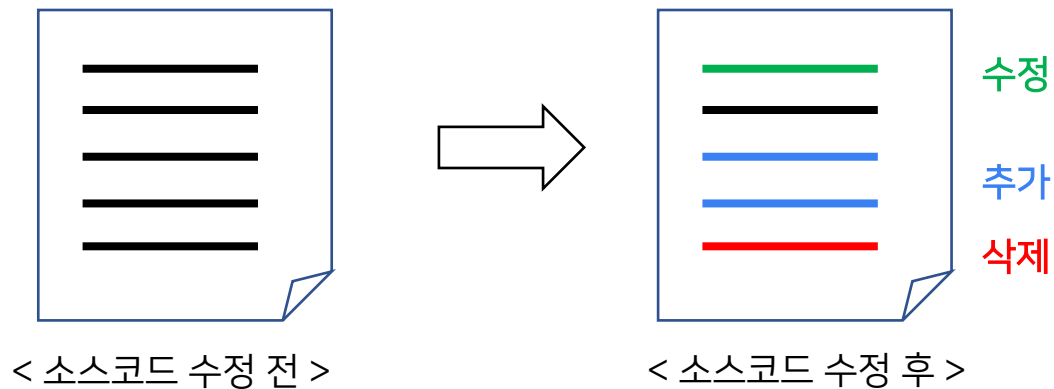
- 매번 다른 이름으로 따로 파일을 저장하여 관리하는 방법도 있지만, 이는 SW 개발 과정에서만큼은 권장할 만한 방법이 아님.
- 매번 파일을 다른 이름으로 새롭게 저장하여 변경 내역을 관리하는 것은 저장 공간을 낭비하는 일일 뿐 아니라 쉽게 실수할 수도 있기 때문.



SW 개발시 git의 필요성

■ (2) 버전(version)을 되돌리기 어려움

- SW 개발시에는 반드시 과거 버전으로 쉽게 회귀할 수 있도록 개발 환경을 갖추어야 함.
 - SW변경(디자인 변경 or 버그 수정 등) 수정한 코드에 문제가 생겼거나 사용자 반응이 영 좋지 않은 등 여러 문제로 이전의 모습으로 되돌려야 하는 상황이 생길 수 있음.
 - 만일 파일을 단순히 덮어썼거나 다른 이름으로 저장하는 방식으로 변경 내역을 관리했다면 파일의 어느 부분이 삭제됐고, 어느 부분을 어떻게 되돌려야 할지 파악하기 어려울 것.
 - 다시 말해, 이전으로 되돌리기가 어려워짐.



SW 개발시 git의 필요성

■ (3) 협력하기 어려움

- 대규모 SW는 대부분 여러 개발자가 협업하여 개발
 - 예) 로그인 기능, 결제 기능, 게시판 등등 각자 개발할 업무를 맡고, 추후 각자 만든 내용들을 통합.
- 코드를 합치는 과정에서 서로가 작업한 내용을 일일이 비교해야 한다면 시간이 많이 걸릴뿐더러 실수도 매우 빈번하게 발생
 - 만일 모두가 작업한 파일을 덮어쓰는 방식으로 저장했거나 다른 이름으로 파일을 저장하는 방식으로 파일을 관리했다면 서로의 작업 내역을 합칠 때 매우 어려워짐.
 - 웹 사이트를 이루는 파일이 여러 개이고 코드 양이 방대하다면 누가 어떤 파일에서 어떻게 코드를 수정했는지 파악하기 힘들기 때문임.

A 개발자: 1개 파일 삭제, 2개 파일 생성, 3개 파일 수정
B 개발자: 2개 파일 삭제, 1개 파일 생성, 10개 파일 수정
C 개발자: 0개 파일 삭제, 3개 파일 생성, 6개 파일 수정
...

형상관리 (Configuration management: CM or SCM)

■ 형상 관리 개념의 등장

- SW 개발 생명주기 전반에 걸쳐 생성되는 모든 산출물의 통합 및 변경 과정을 체계적으로 관리하고 유지하는 개발 관리 활동.
- 개발 중 발생하는 모든 산출물들이 변경됨으로써 점차 변해가는 SW형상을 체계적으로 관리하고 유지하는 기법.
- The task of tracking and controlling changes in the SW.
- Version management.
 - 필요하면 이전의 버전으로 언제든지 되돌림.
 - 여러 사용자에게 대한 버전 이력 추적관리.
 - 소스코드 충돌 처리 및 파일 변경 사항 확인 가능.

- 되던 코드를 고쳤는데 에러가 나.. 근데 되던 코드를 저장을 안해놨어..”
- “너가 A기능을 만들어 내가 B기능을 만들게! 근데 우리 각각의 코드를 어떻게 합치지..?”
- “컴퓨터가 고장이나서 내가 짠 코드들을 찾을 수가 없어 ㅠㅠ;;”
- “매번 코드를 공유할때마다 압축해서 메일로 보내고 받고.. 협업하기 너무 힘드네 ..”

■ 개요

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
 - Git is released under the GNU General Public License version 2.0 (GPL 2.0)
- 2005년 리눅스의 아버지 Linus Torvalds가 전 세계 수많은 개발자와 함께 오픈 소스 프로젝트(리눅스 커널)를 진행하다가 버전 관리에 어려움을 느껴 만든 도구

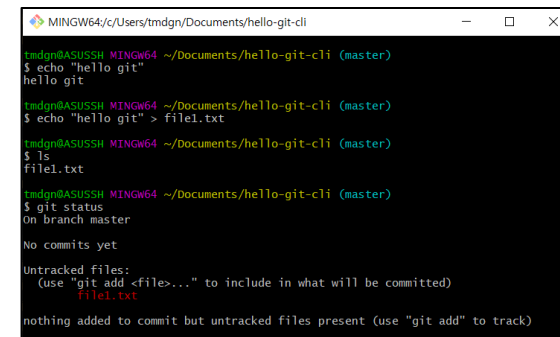


■ 특징

- 대표적인 버전 관리 시스템
 - 파일 변화를 시간에 따라 기록했다가 나중에 특정 시점의 버전을 다시 꺼내올 수 있는 시스템.
 - 소프트웨어 소스 코드만 보여주지만, 실제로 거의 모든 컴퓨터 파일의 버전을 관리할 수 있음.
- 각 파일을 이전 상태로 되돌릴 수 있음.
 - 프로젝트를 통째로 이전 상태로 되돌릴 수 있음.
 - 파일을 잃어버리거나 잘못 고쳤을 때도 쉽게 복구
- 시간에 따라 수정 내용을 비교 가능

■ 특징

- Version을 만들고 되돌리며, 다른 개발자들과 협업할 수 있음.
- git은 모든 변경사항과 파일들을 모든 시점에서 추적(tracking).
 - 무엇이, 어디에서, 언제, 누구에 의해 바뀌었는지 등을 알 수 있음.
 - 파일 수정 중에 실수로 웹 사이트가 망가졌다면, 이전 시점으로 되돌아갈 수 있게 함.
- git은 0,1로 이루어진 binary code로 파일을 읽기 때문에 원하는 것이 무엇이든 읽을 수 있음.
 - 오디오, 이미지, 엑셀파일, 텍스트파일 등도 가능
- git 또한 오픈 소스 프로젝트로, 모든 소스 코드가 공개되어 있음.
 - <https://github.com/git/git>
- 명령어로 이용하는 소프트웨어
 - git 제대로 활용하려면 git 명령어와 옵션을 숙지



```
MINGW64~/Users/tmdgn/Documents/hello-git-cli
tmdgn@BASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ echo "hello git"
hello git

tmdgn@BASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ echo "hello git" > file1.txt

tmdgn@BASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ ls
file1.txt

tmdgn@BASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt

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

■ 중요성

- 소스 코드 변경 이력 관리에 사용되는 사실상의 표준 (de facto standard)

Software Development DevOps ▾ Agile Microservices

> Learn Git

▾ **Beginner**

What is version control

Source Code Management

What is Git

Why Git for your organization

Install Git

Git SSH

Git archive

GitOps

Version control with Git

Git is the best choice for most software teams today. While every team is different and should do their own analysis, here are the main reasons why version control with Git is preferred over alternatives:

Git is good

Git has the functionality, performance, security and flexibility that most teams and individual developers need. These attributes of Git are detailed above. In side-by-side comparisons with most other alternatives, many teams find that Git is very favorable.

Git is a de facto standard

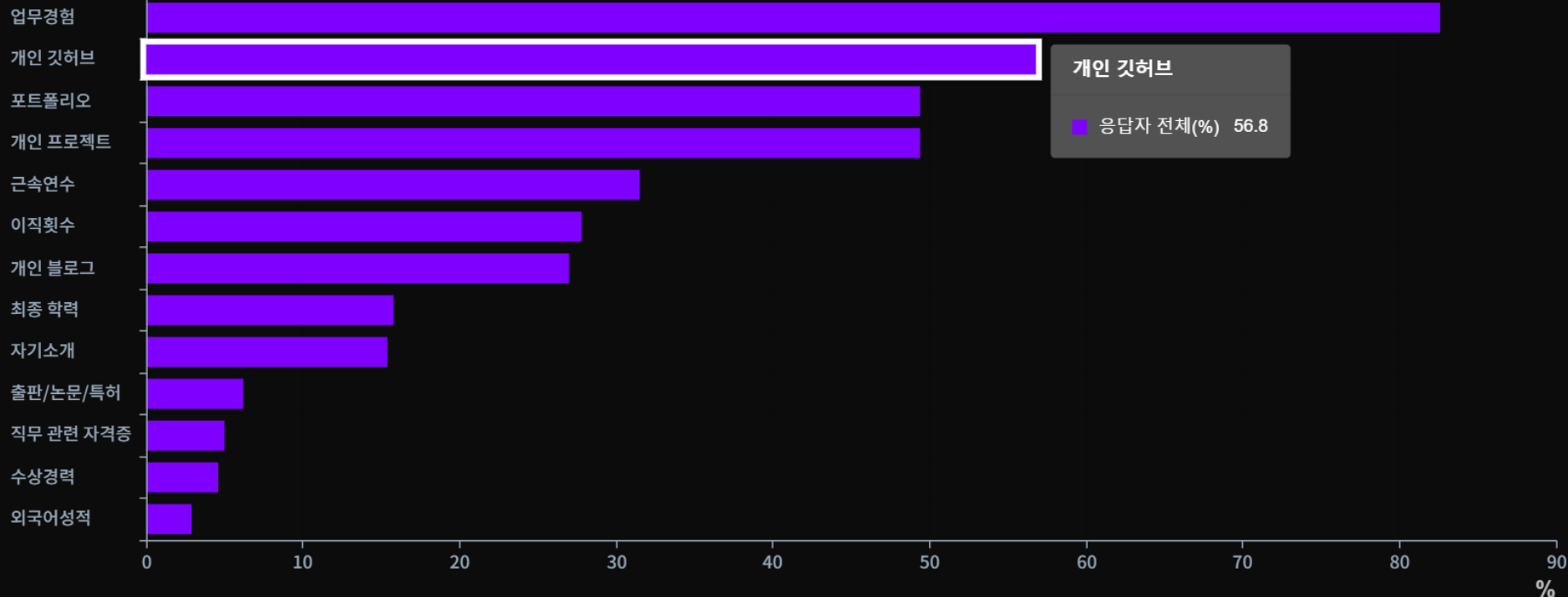
Git is the most broadly adopted tool of its kind. This makes Git attractive for the following reasons. At Atlassian, nearly all of our project source code is managed in Git.

■ 중요성

● IT 채용 및 교육 회사 프로그래머스의 채용담당자 대상 설문조사

개발자 이력서에서 눈여겨보는 것은 무엇인가요?
(모두 선택해주세요)

이력서 항목

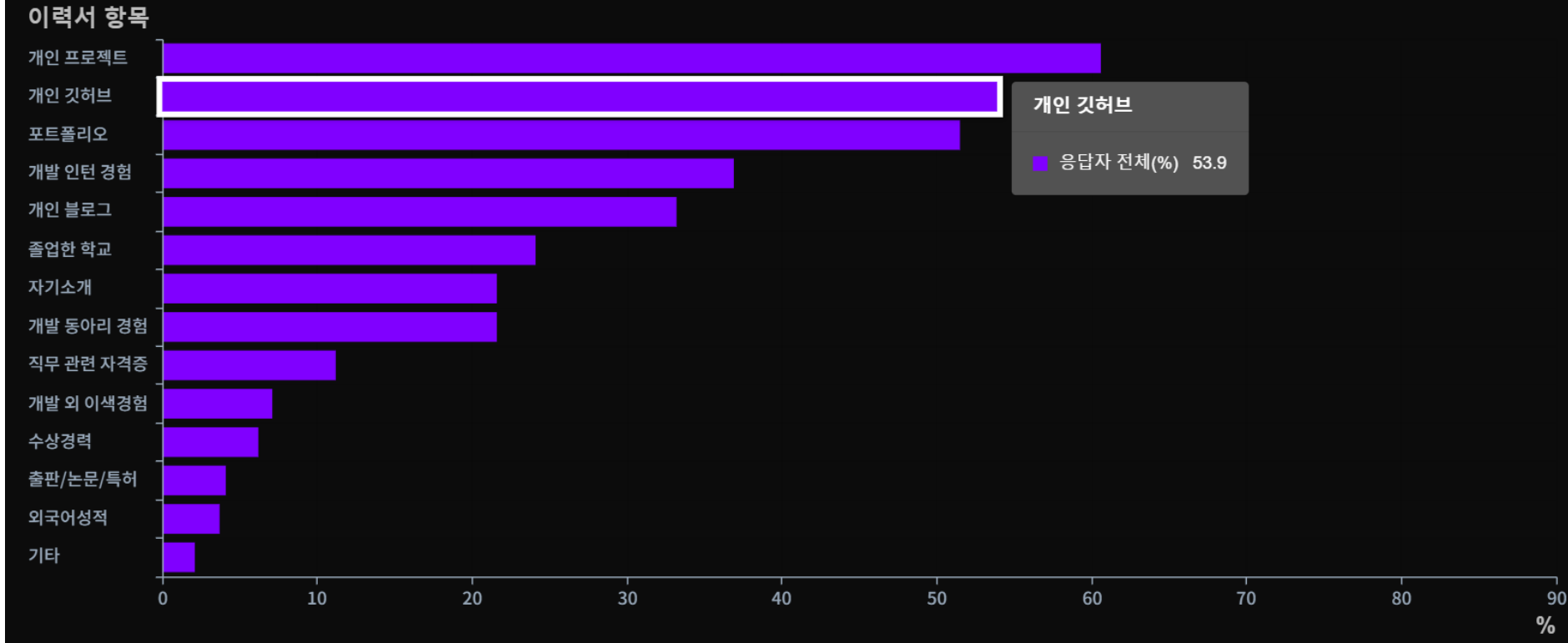


출처: <https://programmers.co.kr/pages/2023-recruiting-survey>

■ 중요성

● IT 채용 및 교육 회사 프로그래머스의 채용담당자 대상 설문조사

신입 개발자 이력서에서 추가적으로 눈여겨보는 것이 있다면, 무엇인가요?
(모두 선택해주세요)



출처: <https://programmers.co.kr/pages/2023-recruiting-survey>

■ 사용하기

● Git Bash

- CLI (Command Line Interface)
- 명령 행 인터페이스
- CLI 사용할 줄 알면 GUI도 사용할 수 있지만, 반대는 성립하지 않음
- Windows의 명령 프롬프트(cmd)
- Mac의 Terminal

```
MINGW64/c/Users/tmdgn/Documents/hello-git-cli
tmdgn@ASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ echo "hello git"
hello git

tmdgn@ASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ echo "hello git" > file1.txt

tmdgn@ASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ ls
file1.txt

tmdgn@ASUSSH MINGW64 ~/Documents/hello-git-cli (master)
$ git status
On branch master

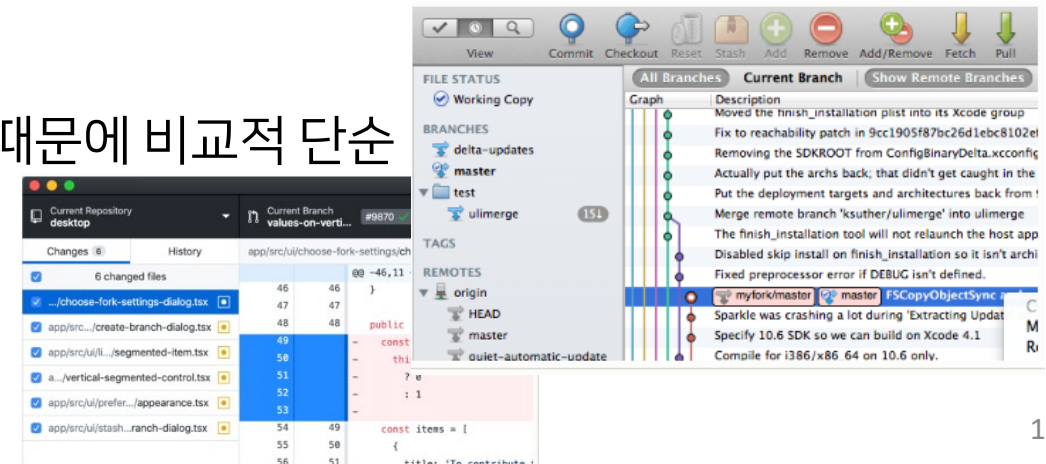
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
file1.txt

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

● Git GUI

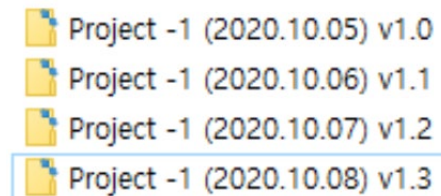
- GUI 프로그램의 대부분은 Git 기능 중 일부만 구현하기 때문에 비교적 단순
- SourceTree, GitHub Desktop 등



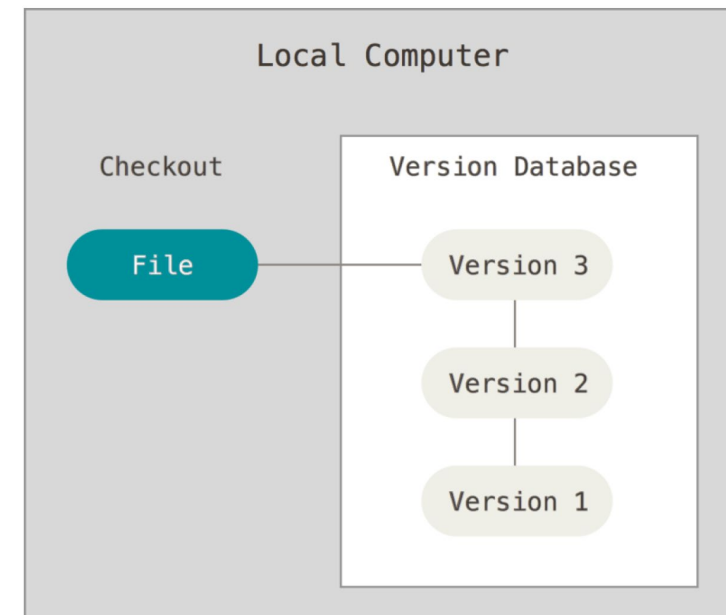
■ 버전 관리 시스템(VCS: Version Control System)의 종류 비교

● (1) Local VCS

- 간단한 DB에 파일의 변경정보를 담아 관리.
- 가장 쉽게 간편한 방법으로 파일을 각 시점마다 복사.
- 삭제 및 변경에 취약.
 - 작업하던 디렉토리를 지워버릴 수 있음.
 - 실수로 파일을 잘못 고칠수도 있고, 잘못 복사할수도 있음.
- 다른 컴퓨터에서는 해당 DB에 접근이 어려움.



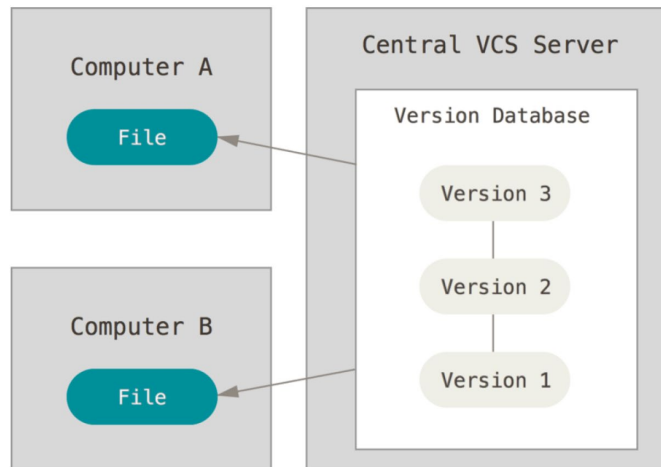
Project -1 (2020.10.05) v1.0
Project -1 (2020.10.06) v1.1
Project -1 (2020.10.07) v1.2
Project -1 (2020.10.08) v1.3



■ 버전 관리 시스템(VCS: Version Control System)의 종류 비교

● (2) 중앙 집중식 버전 관리 (CVCS: Centralized VCS)

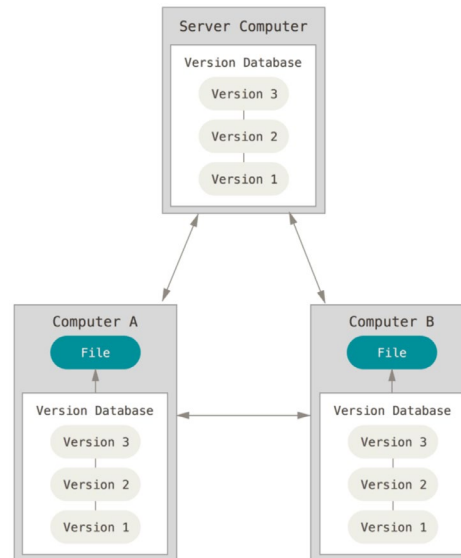
- 파일 관리를 하는 중앙 서버에 DB가 있어 각 로컬에서 해당 서버에서 파일을 받아서 사용.
- DB를 각 로컬에서 관리하지 않고 중앙에서 관리하기에 관리가 쉬움.
- 서버 기록을 보면 모든 사용자 활동 내역 확인 가능.
- 중앙 서버가 작동하지 않으면 그동안 아무도 다른 사람과 협업할 수 없음.
- 중앙 서버 하드디스크에 문제가 생기면 프로젝트의 모든 history를 잃어버림.



■ 버전 관리 시스템(VCS: Version Control System)의 종류 비교

● (3) 분산 버전 관리 (DVCS: Distributed VCS)

- 로컬에 중앙 DB 저장소를 히스토리와 더불어 전부 복제 (Clone).
- 로컬에서 버전 관리 DB를 만들어 서버에 올릴 수도 있고, 다른 컴퓨터 로컬에 복사해서 줄수도 있음.
- 서버에 문제가 생기면 이 복제물로 다시 작업 시작 가능.
- 로컬 클라이언트 중에서 아무 것이나 골라도 서버 복원 가능.

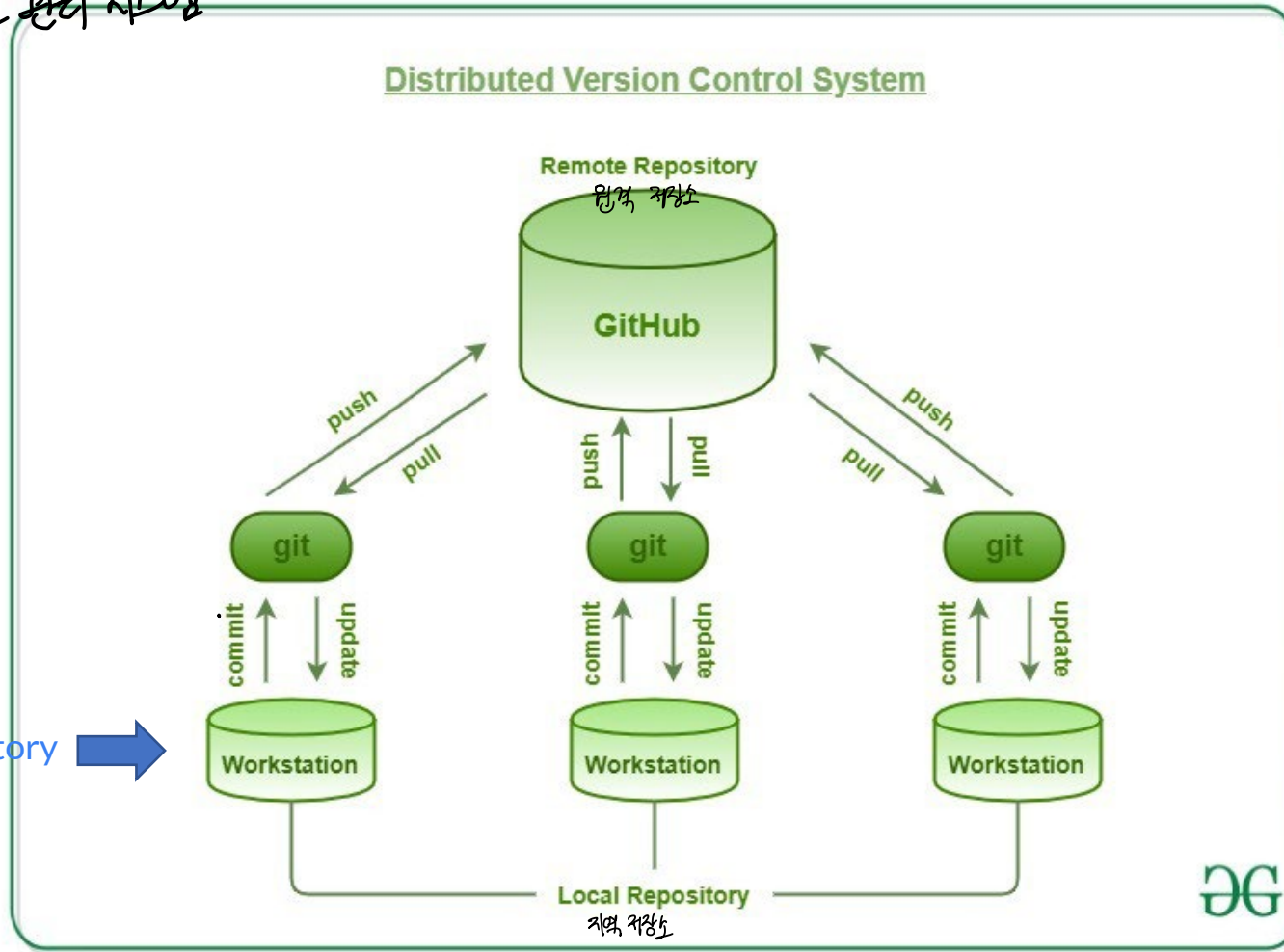


■ git은 DVCS에 해당

분산 버전 관리 시스템

Local clone of repository

지역 복제 저장소



■ 개요

※ 호스트: 물리적 or 가상의 환경

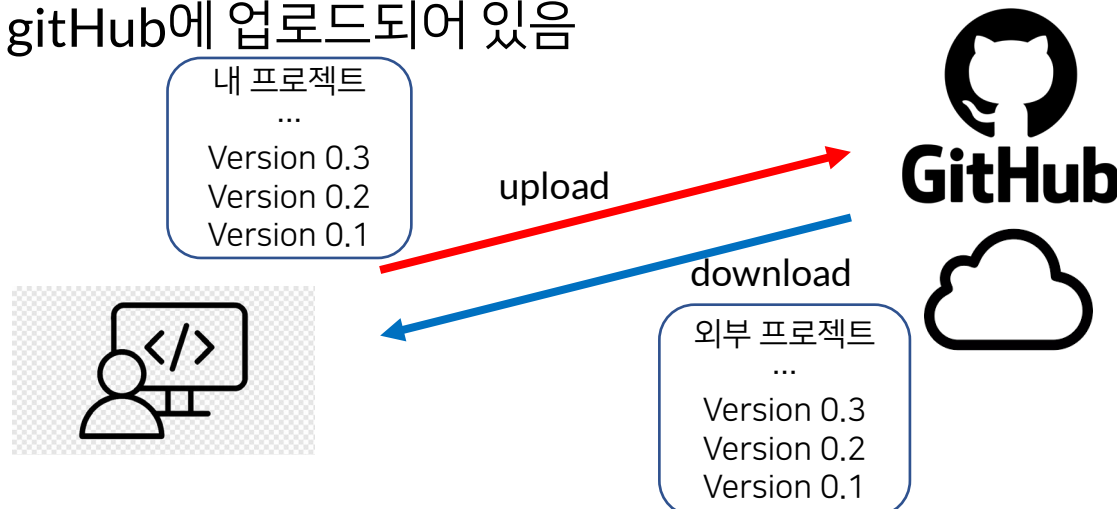
● 원격 저장소 호스팅 서비스

일정 저장소를 사용자들이 사용하도록 내어주는 것

- '원격 저장소'라는 말이 조금 생소하겠지만, 쉽게 '깃으로 버전을 관리하는 프로젝트들이 모여 있는 웹 사이트' 정도로 생각해도 무방함.

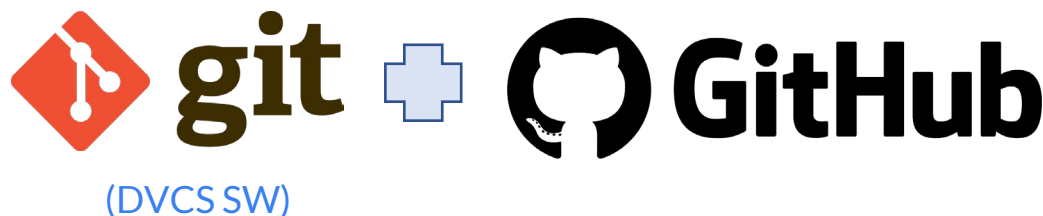
- git으로 버전 관리한 프로젝트를 GitHub에 업로드 가능. GitHub 업로드한 프로젝트에 새로운 버전을 추가 가능.
- 반대로 GitHub에 업로드된 전 세계 개발자들의 프로젝트를 로컬 컴퓨터로 다운로드 가능.

- 텐서플로(tensorflow), 쿠버네티스(Kubernetes), 리액트(react) 등 이름만 들어도 알 만한 유명한 프로젝트들이 이미 gitHub에 업로드되어 있음



■ 개요

- 전세계적으로 가장 대표적인 오픈소스 SW 프로젝트 플랫폼
 - 소스코드 공유와 협업 플랫폼.
 - GitHub에 업로드된 프로젝트에 코드를 기여하고, 다른 개발자들과 협업할 수도 있음.
 - 공개/비공개 저장소, 코드 리뷰, 문서화, 커뮤니티 등 제공.
 - 무료 서비스 사용시 공개 저장소만 사용 가능.
 - 모두에게 소스코드 내용 오픈.
 - 비공개 저장소 사용시 유료 서비스 가입.
- 2018년 6월 4일, Microsoft사 75억 달러(약 10조원) 인수 발표




Git을 위한 웹 저장소(Repository)
 + 커뮤니티 협의 공간
 + 개발자들 간에 소통, 협업, 통합 및 자동화 지원

git and GitHub

■ git과 GitHub의 개념

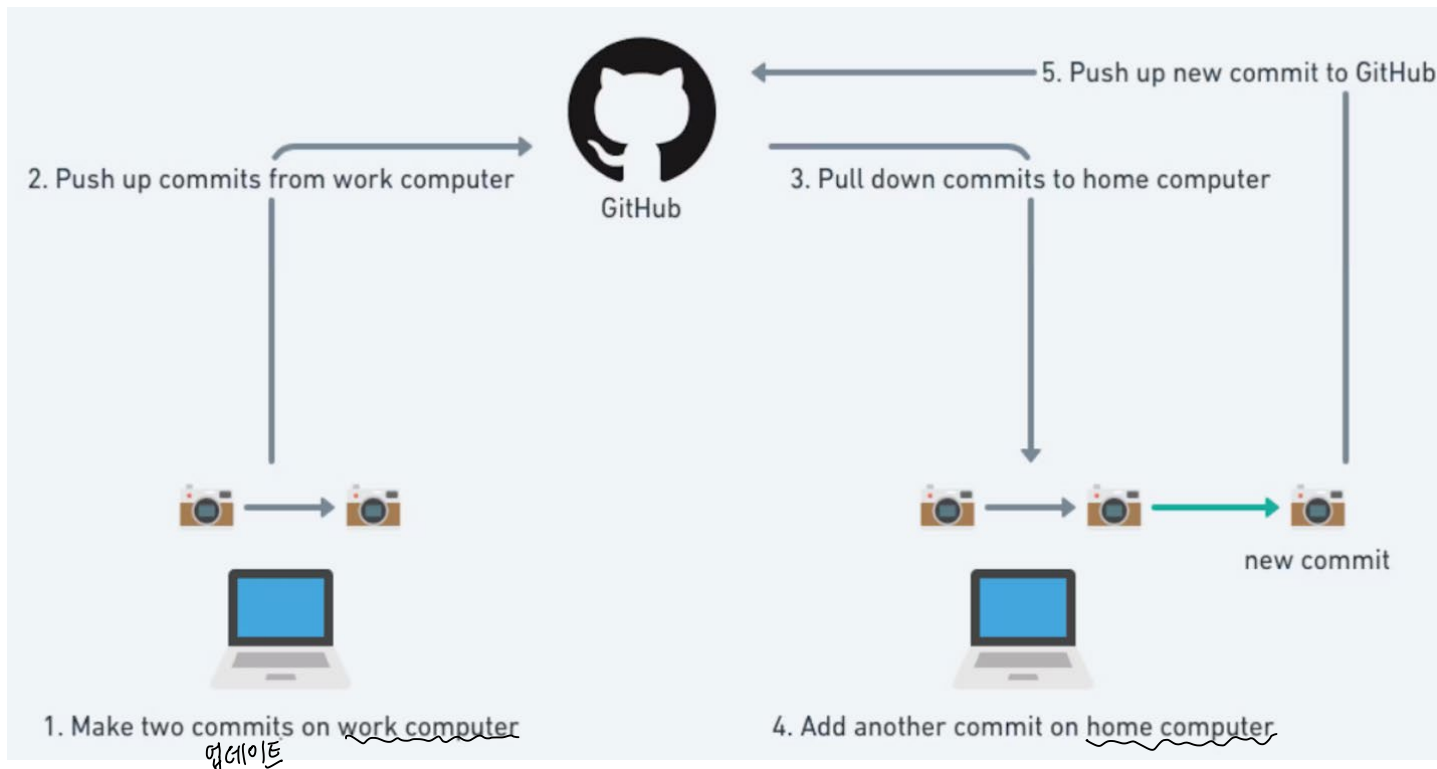
WHAT ARE GIT & GITHUB?



Git is a version control system to keep track of changes to files and projects over time.
버전 관리 시스템

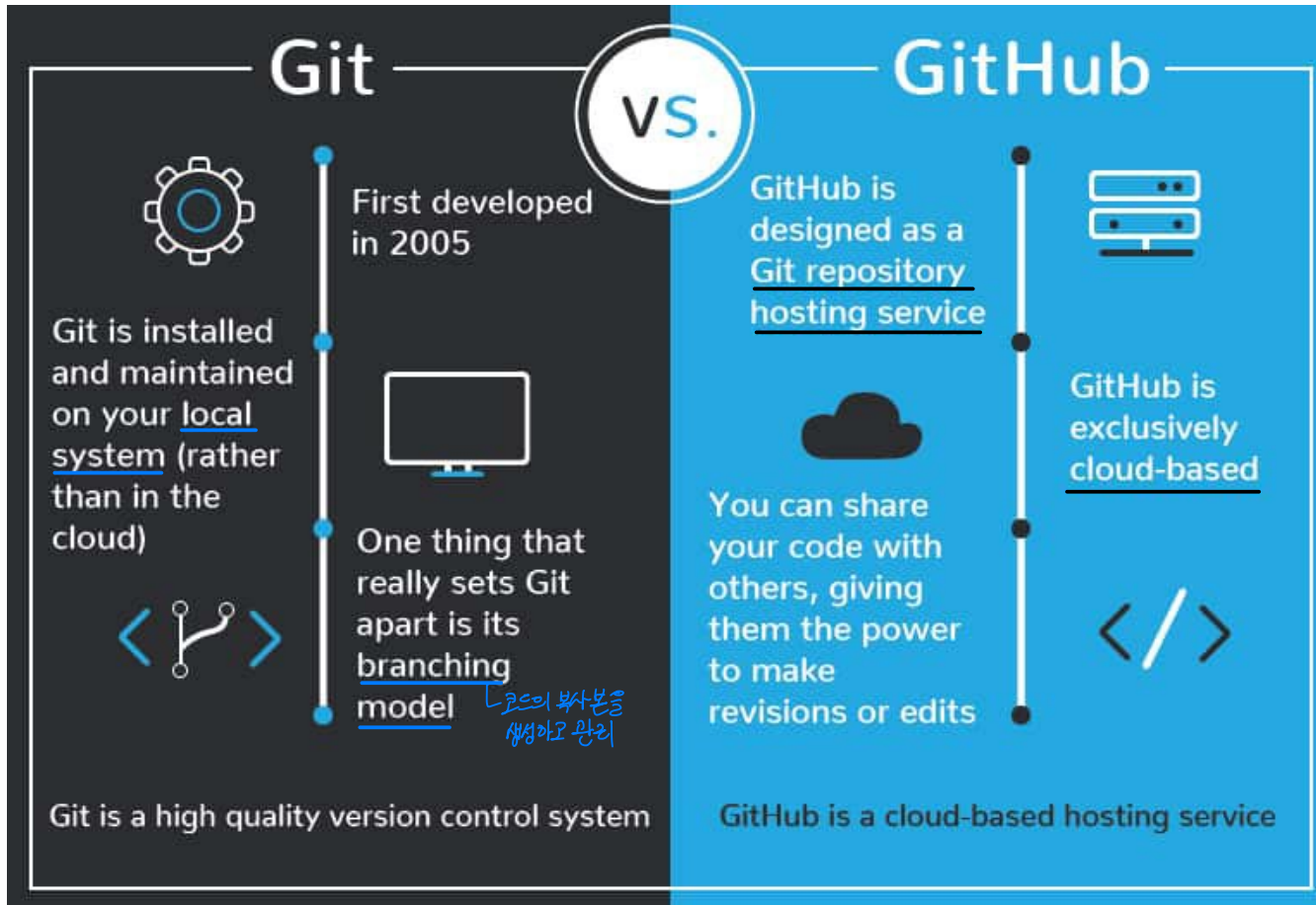
GitHub is a website that hosts Git repositories online, making it easier for developers to share code.
저장소

Repositories (or "repos") are folders which contain intentional snapshots of progress called commits.
의도적인 스냅샷 저장하는 폴더



git and GitHub

■ git과 GitHub의 비교



git and GitHub

■ git과 GitHub의 정리

- git은 소스코드 버전 관리 등 작업 파일들을 효과적으로 관리할 수 있게 해주는 SW.
- GitHub는 나의 git 파일을 업로드하는 곳.
 - git의 웹호스팅을 제공 (GitHub 외에 GitLab, Bitbucket 등이 있음)
- GitHub 변경사항을 회사, 친구와 공유할 수 있음.
 - 나의 git 파일 업로드, 다른 사람의 git 파일 다운로드 가능. (git ≠ github)
- SourceTree (or Github Desktop)은 가독성이 좋지 않은 CLI(Command Line Interface)의 git을 위해 보기 좋게 GUI(Graphic User Interface)를 제공하는 프로그램

git 사용 준비

■ git 설치하기

- git 홈페이지 접속

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

- 적절한 운영체제 선택 후 다운로드 및 실행하여 설치

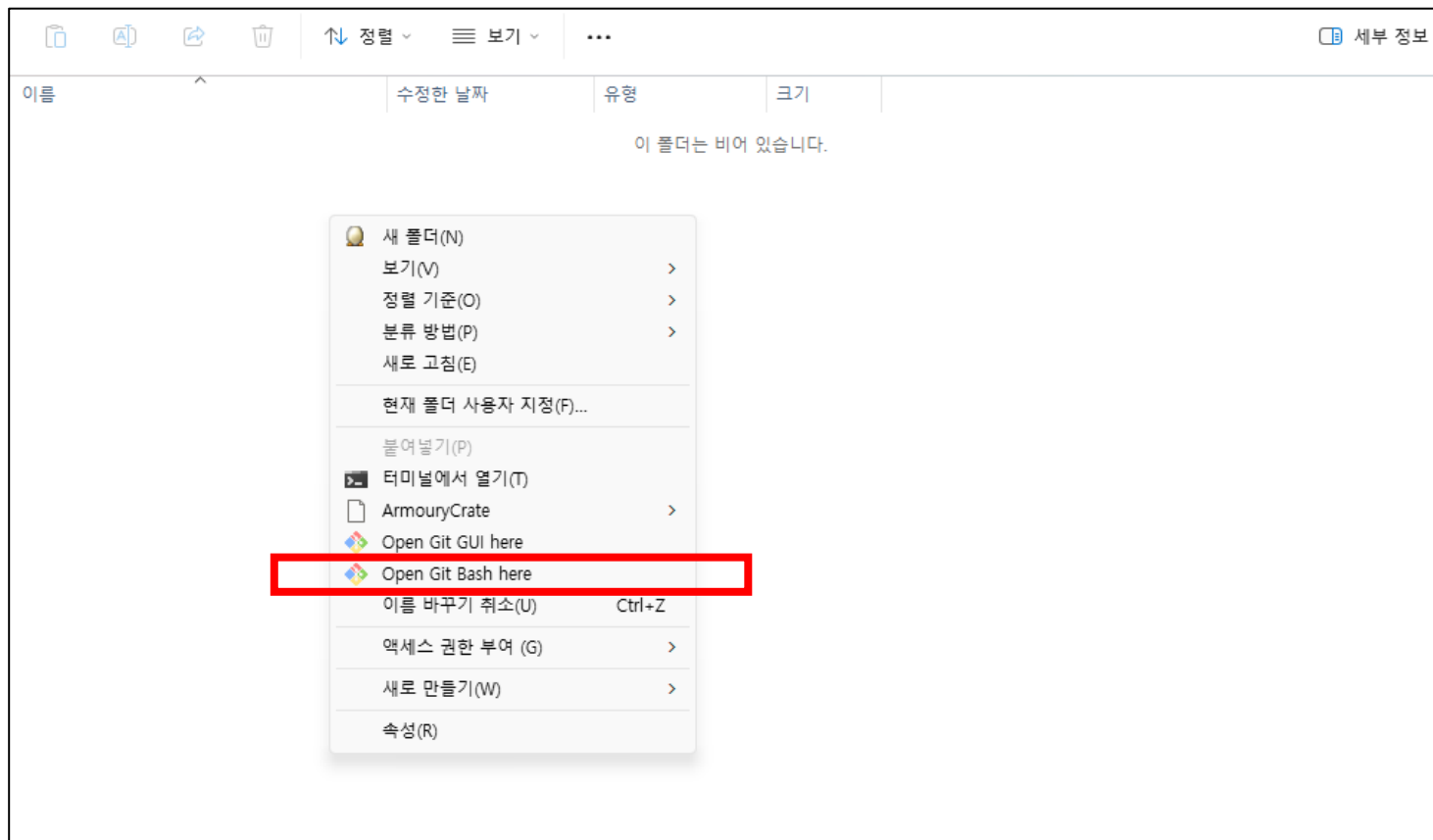
- Next 버튼 클릭하여 기본 설정대로 설치

- git에서 사용할 기본 문서 편집기(에디터)를 선택하는 창에서는 Use Vim (the ubiquitous text editor) as Git's default editor 선택 권장
 - Use bundled OpenSSH

git 사용 준비

■ git 설치 확인

- 임의의 폴더(ex. test) 생성 후 그 안에서 마우스 우측 버튼 클릭
- Open Git Bash here 클릭



git 사용 준비

■ git 명령어 입력해보기

● git bash 확인

마이크로소프트 windows로 포팅한
GNU 소프트웨어 도구 모음

현재 작업공간 확인
(명령어를 입력하고 있는 작업공간)

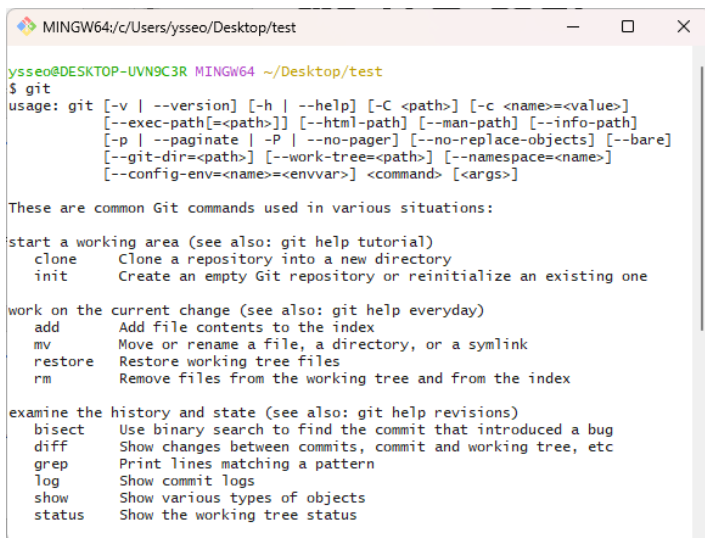


```
MINGW64:/c/Users/ysseo/Desktop/test
ysseo@DESKTOP-UVN9C3R MINGW64 ~/Desktop/test
$
```

● git 입력시 관련 명령어 목록 출력

■ git -v

■ git --version



```
MINGW64:/c/Users/ysseo/Desktop/test
ysseo@DESKTOP-UVN9C3R MINGW64 ~/Desktop/test
$ 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
grep       Print lines matching a pattern
log        Show commit logs
show       Show various types of objects
status     Show the working tree status
```

git 사용 준비

■ git 설정하기

- 사용자 컴퓨터에 사용자 이름과 이메일을 등록하는 간단한 초기 설정.

- `git config --global user.name "InChan Kim"`
- `git config --global user.email "ickim@yu.ac.kr"`

- 앞으로 깃을 이용해 만드는 모든 버전에는 '만든 사람', '지은이'와 같은 개념으로 지금부터 설정할 이름과 이메일이 함께 명시될 것임.
- 영어를 사용할 것을 권장.

- 설정한 이름과 이메일 확인

- `git config user.name`
- `git config user.email`

- `git config`로 설정한 값 확인

- `git config --list`

Sourcetree 사용 준비

■ Sourcetree 설치하기

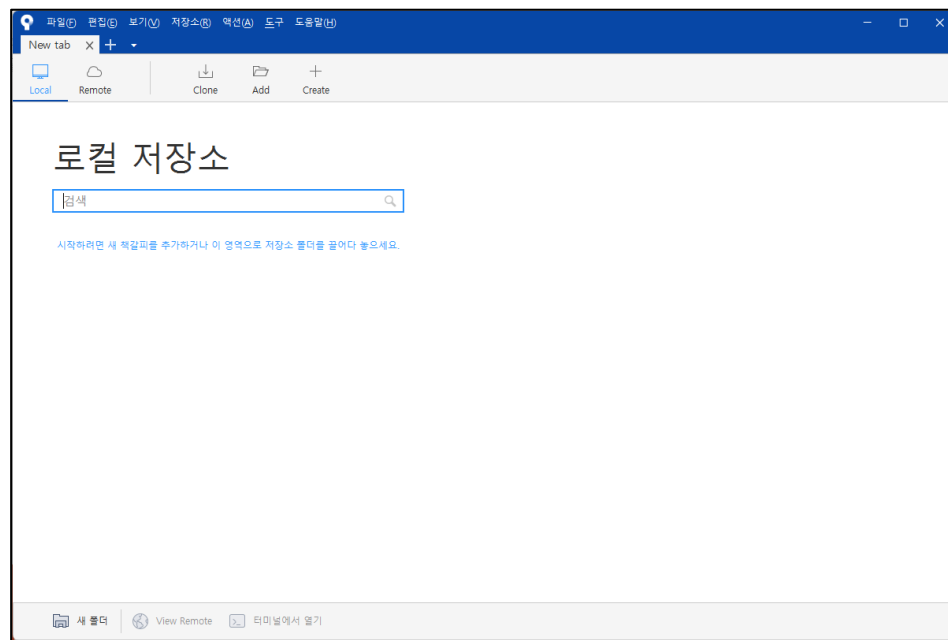
- Sourcetree 홈페이지 접속

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

- 적절한 운영체제 선택 후 다운로드 및 실행하여 설치

- Bitbucket 계정 등록은 하지 않고 “건너뛰기” 선택.
- Mercurial 체크 해제 후 “다음” 선택.
- 이름과 이메일은 앞서 설정한 값으로 자동 기입됨.
- “SSH 키를 불러오시겠습니까?” -> “아니오” 선택.

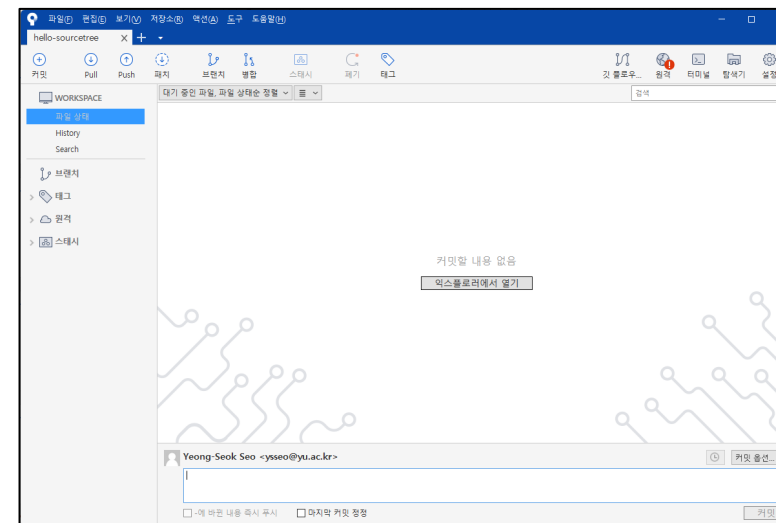
- SSH(Secure SHell)는 SSH 키를 이용해 안전하게 원격 컴퓨터와 연결하는 통신 방법



Sourcetree 사용 준비

■ Sourcetree에서 저장소(Local repository) 만들어보기

- Local repository: local 컴퓨터에서 버전들이 만들어지고 관리되는 공간.
- 상단의 Local - Create 클릭.
 - 목적지 경로: 내 컴퓨터 어느 곳에 저장소를 만들 것인지 결정.
 - 원하는 폴더 경로 입력 or 탐색 버튼을 클릭하여 폴더 지정.
 - C:\hello-sourcetree



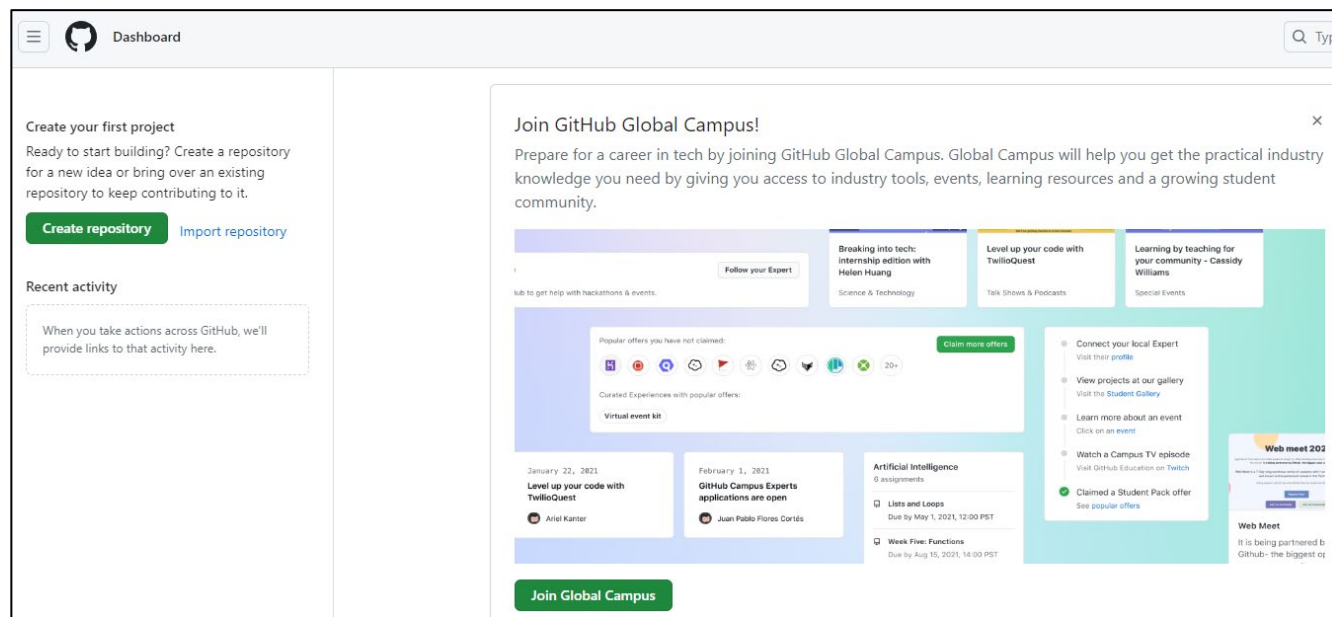
- C:\hello-sourcetree 에 그림과 같이 **.git 숨김 폴더**가 잘 만들어졌는지 확인 (숨겨진 파일 보이도록 설정).



GitHub 사용 준비

■ GitHub 회원 가입하기

- GitHub 홈페이지 접속.
 - <https://github.com/>
- Sign Up 클릭하여 다음 정보 입력.
 - email
 - password
 - username
- 간단한 퍼즐 풀기 후 계정 생성 완료.



Markdown

■ Markdown is...

- Lightweight markup language that you can use to add formatting elements to plaintext text documents.
 - Markup language: 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어의 한 가지
 - Markup language의 예: HTML, XML, json 등
- Created by John Gruber in 2004, Markdown is now one of the world's most popular markup languages.
- GitHub의 README.md 가 대표적인 Markdown 문서
- <https://www.markdownguide.org/> : 기본 문법 확인

Markdown

■ 장점

- 문법이 쉽고 간결함
 - It doesn't take long to learn the Markdown syntax 문법
- 지원 가능한 플랫폼과 프로그램이 다양함
 - Most people use Markdown to create content for the web, but Markdown is good for formatting everything from email messages to grocery lists.
- Text로 저장되기 때문에 용량이 적어 보관이 용이
- HTML 변환 가능

HyperText Markup Language

■ 단점

- 표준이 없어 도구에 따라 변환방식이나 생성물이 조금씩 다를 수 있음
- 모든 HTML 마크업을 대체하지 못함