

Open source Software

# 오픈소스 소프트웨어

## 03. 깃과 깃허브 개요



### 학습 개요

1. 깃 개요와 기능
2. 깃 설치: 두 개의 SW 제공 Git Bash, Git GUI
3. 깃의 내부 저장소 구조와 이동 명령
4. 브랜치 이해
5. 깃허브 이해



### 학습 목표

1. 깃을 이해하고 깃 설치하면 설치되는 소프트웨어를 파악할 수 있다.
2. 깃의 주요 영역과 이동 명령을 설명할 수 있다.
3. 브랜치를 이해할 수 있다.
4. 깃허브를 이해할 수 있다.

# LESSON 01

## 깃 개요



### 1 깃개요

#### 깃 Git

- ✓ 2005년, 리누스 토발즈가 개발
  - 주니오 하마노(Junio Hamano)가 소프트웨어의 유지보수
- ✓ 대표적인 오픈소스 소프트웨어 프로젝트
  - <https://github.com/git>
  - <https://git.kernel.org/pub/scm/git/git.git>



### 1 깃 개요

#### 깃 사용 장점

- ✓ 모든 개발자는 지역 시스템에 코드의 전체 사본을 소유
  - 소스 코드에 대한 모든 변경 사항은 다른 사용자가 추적 가능



<출처> : <https://www.simplilearn.com/tutorials/git-tutorial/what-is-git>

### 1 깃개요

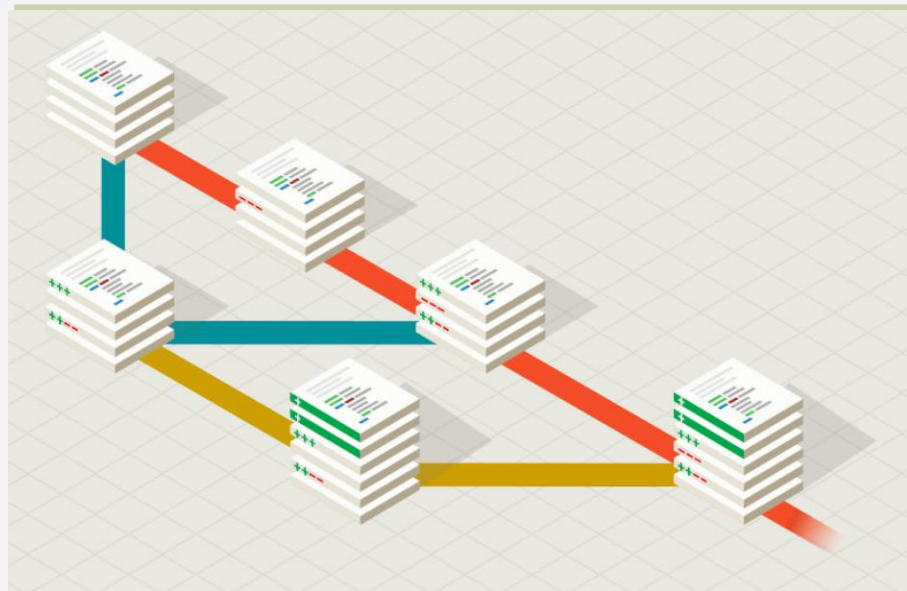
#### 깃 기능

##### 정의

- 컴퓨터 파일의 변경을 추적하는 데 사용되는 버전 관리 시스템

##### 기능

- 여러 개발자가 함께 작업
- 소스 코드의 변경 사항을 추적하는 데 사용
- 소스 코드 관리에 분산 버전 제어 도구가 사용
- 여러 개의 병행 분기를 통해 비선형 개발을 지원

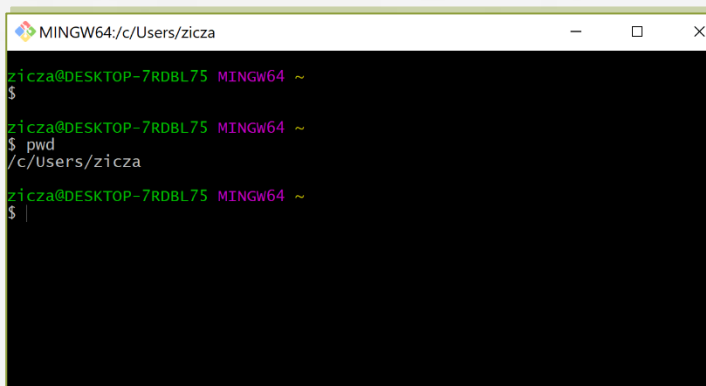


### 1 깃개요

## ⚙ 깃 설치: 두 개의 SW 제공 Git Bash, Git GUI

### ✓ 깃 Bash: CLI(Command Line Interface)

- 명령 행 인터페이스
  - 처음엔 어렵지만
  - CLI를 사용할 줄 알면 GUI도 사용할 수 있지만
    - ➡ 반대는 성립하지 않음
- Mac의 Terminal
- Windows의 CMD나 Powershell



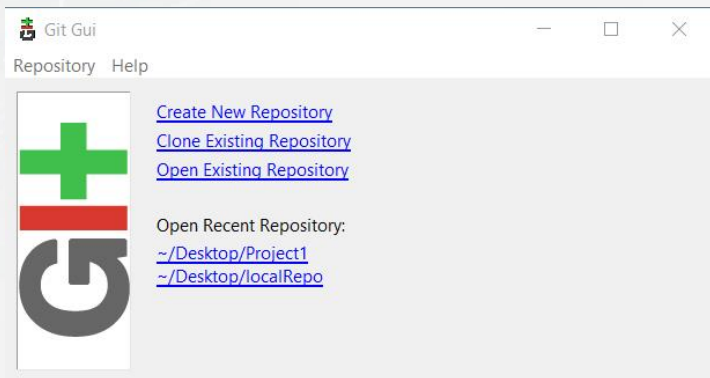
```
MINGW64: c:/Users/zicza
zicza@DESKTOP-7RDBL75 MINGW64 ~
$
zicza@DESKTOP-7RDBL75 MINGW64 ~
$ pwd
/c/Users/zicza
zicza@DESKTOP-7RDBL75 MINGW64 ~
$ |
```

### 1 깃개요

## 깃 설치: 두 개의 SW 제공 Git Bash, Git GUI

### ✓ Git GUI: GUI(Graphical user interface)

- GUI 프로그램의 대부분은 Git 기능 중 일부만 구현하기 때문에 비교적 단순





### 1 깃개요

## 깃의 내부 주요 영역 구조

### 깃 내부 저장소 상태

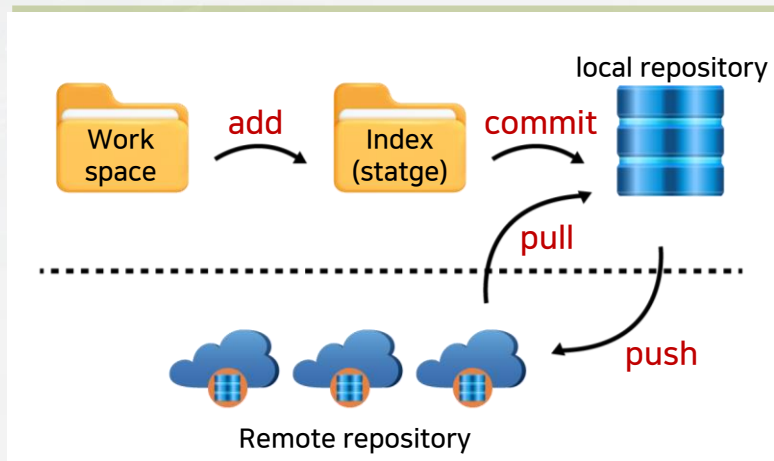
- 작업 디렉토리(working directory, working folder)
- 작업 공간(work space), 작업 트리(working tree)
  - modified, untracked
- 스테이징 영역(staging area, stage area, index)
  - staged, indexed
- 깃 저장소(git repository, repository, .git directory)
  - Committed
- 임시 저장소(stash)
  - stash

## 1 깃개요

### 깃에서 상태 간의 이동 명령

✓ add

✓ commit



### 1 깃개요

#### 🔧 브랜치

#### ☑ 나무 가지



#### ☑ 지점

#### ☑ 분기

## 1 깃개요

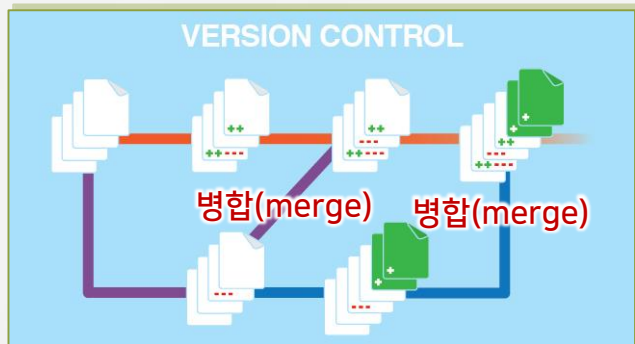
## 분기를 이용한 프로젝트의 비선형 개발 작업 흐름

## 우리의 실 생활

- 여러 파일을 제목의 변형으로 버전을 관리
- 파일 집합을 다시 버전 시작
  - 새로운 폴더에 복사해서 사용하는 것과 비슷

## 분기, 가지(branch) 개념

- 새로운 수정을 할 수 있는 또 다른 버전의 작업 흐름(workflow)
  - 버전 관리를 수행하던 일련의 파일 집합을 통째로 복사해 독립적으로 다시 개발을 진행하는 개념





## LESSON 02

# 깃허브 이해



### 2 깃허브 이해

#### 깃허브 소개

##### ✓ Github

- 버전 관리를 위한 서버 저장소 및 프로젝트 개발을 위한 협업 관리 서비스
  - 시스템 개발자와 운영을 담당하는 정보기술 전문가 사이의 소통, 협업, 통합 및 자동화 지원
  - 전 세계 개발자가 함께 개발 지원
- 프로젝트 소스 공유와 협업 소프트웨어 빌드 플랫폼
  - 개발자들의 소셜 네트워크 서비스 SNS



### 2 깃허브 이해

#### ⚙ 깃허브 이해

##### ✓ Git(깃)?

- 분산 버전 제어 SW

##### ✓ Github(깃허브)?

- Git을 위한 웹 저장소(Repository)
  - 커뮤니티 협의공간
  - 시스템 개발자와 운영을 담당하는 정보기술 전문가 사이의 소통, 협업, 통합 및 자동화 지원



### 2 깃허브 이해

#### 깃허브 서비스

**Github**

분산 환경 소스코드 버전 관리 도구인  
**Git**의 글로벌 웹 호스팅 서비스

#### 세계 최대의 오픈소스 공유 플랫폼

- 공개/비공개 저장소, 코드 리뷰, 문서화, 커뮤니티 등 제공
- 무료 서비스 사용 시 공개 저장소만 사용 가능
  - ➡ 모두에게 소스 코드 내용 오픈
  - ➡ 비공개 저장소 사용 시 유료 서비스 가입



### 2 깃허브 이해

#### 마이크로소프트 자회사

**Github**

분산 환경 소스코드 버전 관리 도구인  
**Git**의 글로벌 웹 호스팅 서비스

#### 세계 최대의 오픈소스 공유 플랫폼

- 2018년 8월 기준 9600만 개 이상의 프로젝트 호스팅  
3100만 명 이상의 개발자 참여
- 2018년 6월 4일 마이크로소프트사 75억 달러 인수 발표

# Summary

## » Git

- ◆ a version control system to keep track of changes to files and projects over time

## » Github

- ◆ A website that hosts Git repositories online, making it easier for developers to share code

## » Repositories(or repo)

- ◆ Folders which contain intentional snapshots of progress called commits

# Summary

## » 깃 설치

- ◆ Git Bash, Git GUI

## » 깃 저장소 구조

- ◆ Working directory
  - Working folder, Working tree, Work space
- ◆ Staging area
- ◆ Git Repository
- ◆ Stash

## » Add 명령

- ◆ Working directory ➡ Staging Area 이동

## » Commit 명령

- ◆ Staging Area ➡ Git repository 이동

# Summary

## » 브랜치

- ◆ 버전을 관리하던 일련의 파일 집합을 통째로 복사해 독립적으로 다시 개발을 진행하는 개념

## » Github

- ◆ 버전을 관리하기 위한 서버 저장소 및 프로젝트를 개발하기 위한 협업 관리 서비스

# Summary

## » 깃 + 깃허브 == Open Source Software || Software Development

- ◆ 깃과 깃허브를 통해 오픈소스 소프트웨어를 개발

