



Git 간단히 알아보기

GIT의 역사와 사용법에 대한 간단한 소개

목차

- ▶ 1. Version Control System이란?
- ▶ 2. Git이란?
- ▶ 3. Git 설치
- ▶ 4. Git 써보기
- ▶ 5. Branch & Merge

Version Control System?

- ▶ 파일의 변화를 시간에 따라 기록하여, 과거 특정 시점의 버전을 다시 불러올 수 있는 시스템.

VCS의 장점?

- ▶ 개별 파일 혹은 프로젝트 전체를 이전 상태로 되돌릴 수 있다.
(즉, 파일을 잃어버리거나 잘못 고쳤을 때 쉽게 복구할 수 있다)
- ▶ 시간에 따른 수정 내용을 비교할 수 있다.
- ▶ 누가 문제를 일으켰는지, 언제 만들어진 이슈인지 추적할 수 있다.

VCS의 종류?

- ▶ 개별 관리
- ▶ Local VCS
- ▶ Centralized VCS - CVCS
- ▶ Distributed VCS - DVCS

CVCS vs. DVCS?

- ▶ CVCS : 모든 파일을 하나의 서버에 저장, 그 서버에서 파일들을 가져오고 수정 사항을 기록한다.
- ▶ DVCS : 저장소(repository)라는 개념을 사용, 저장소를 통째로 복제하고 수정된 저장소를 서버에 다시 복제한다.

Git?

- ▶ 2005년 리누스 토르발즈(Linus Torvalds)가 개발한 Open-source DVCS
- ▶ 리눅스 커널의 개발에 사용, 지금은 전세계적으로 널리 사용되고 있음

Git의 장점?

- ▶ 단순한 구조
- ▶ 매우 ~~마~~찬빠른 속도
- ▶ 분산 시스템 -> 비선형적 개발
- ▶ -> 초대형 프로젝트 개발에도 사용 가능

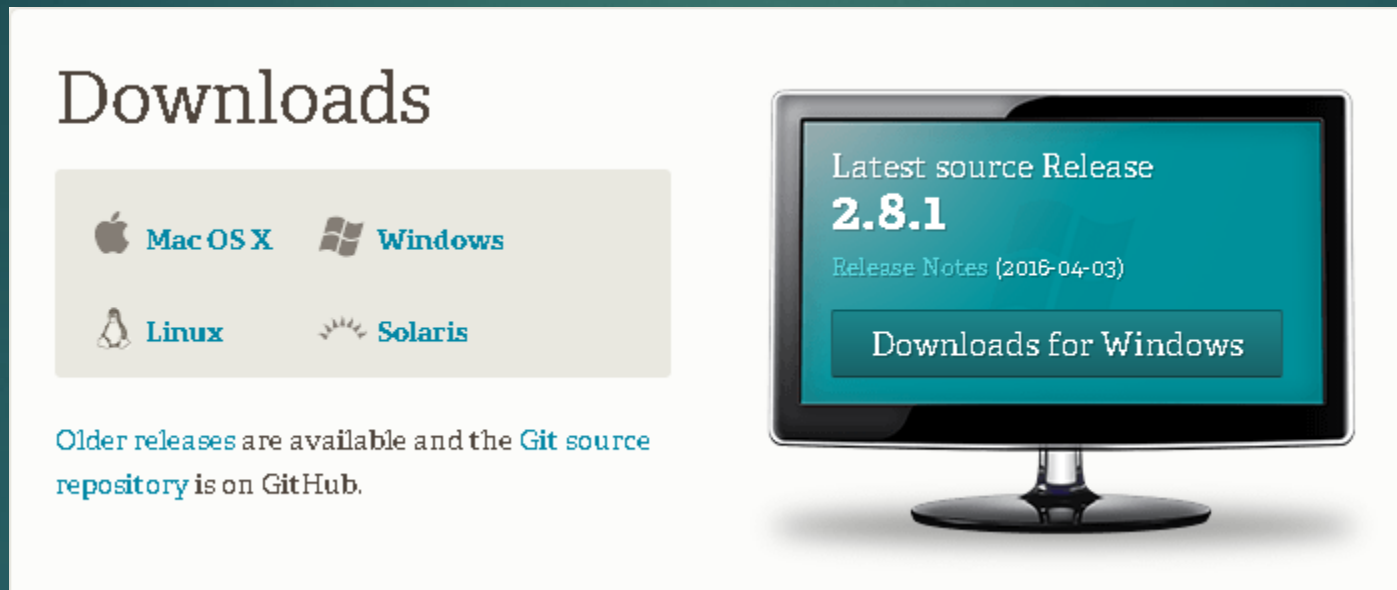
Git 용어 정리

- ▶ Revision : 하나 이상의 수정 혹은 개선을 가한 변화.
- ▶ Snapshot : 특정한 시점에서 시스템의 상태.
- ▶ Repository(저장소) : 파일들의 현재와 과거의 데이터가 저장되는 곳.
- ▶ Checkout : 저장소의 특정 Snapshot의 특정 파일을 작업 공간에 복사하는 행위 / 혹은 그 복사물.

Git 용어 정리

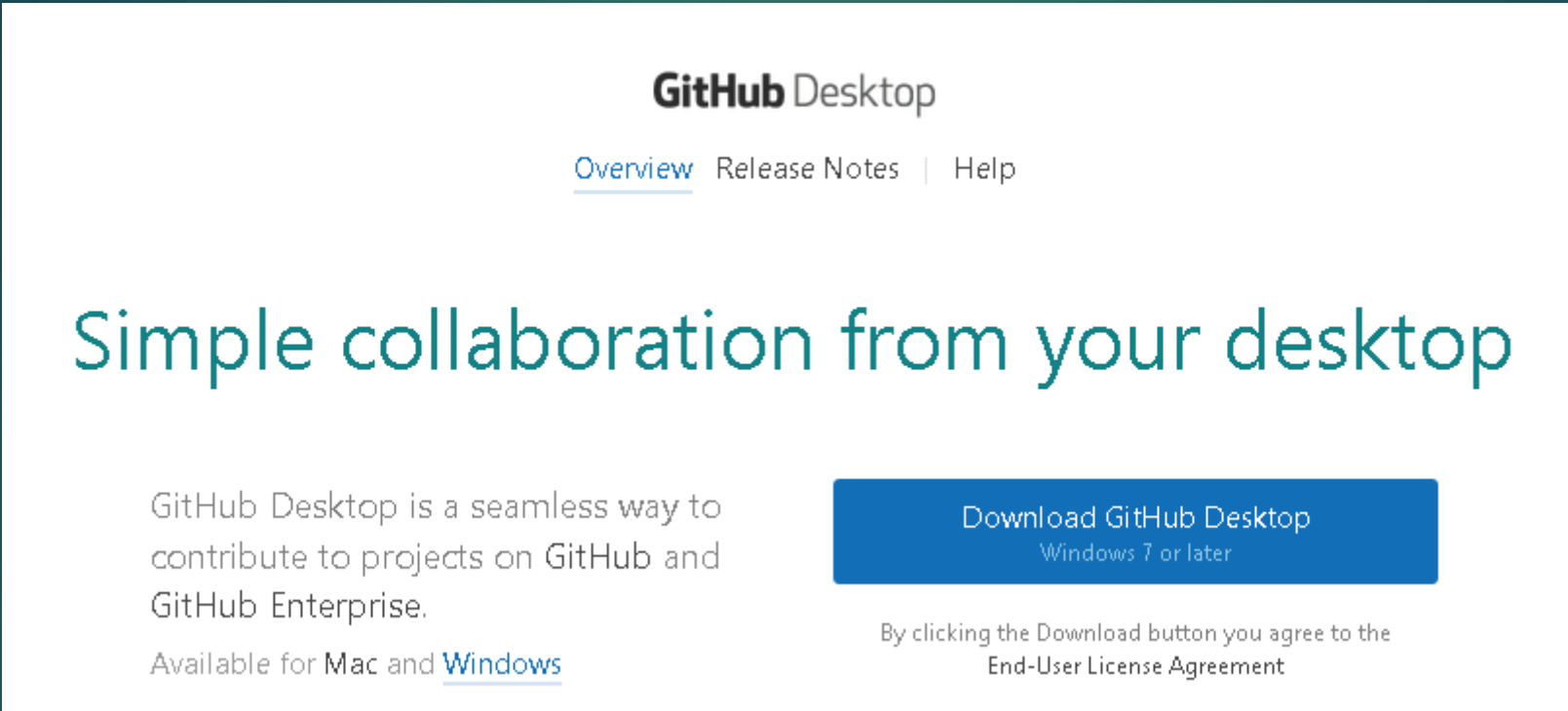
- ▶ Clone : 서버에서 저장소 자체를 복제하는 행위.
- ▶ Commit : 작업 공간에서 이뤄진 변경 사항을 저장소에 다시 기록하는 행위.
- ▶ Pull / Push : 하나의 저장소에서 다른 곳으로 Revision을 복사하는 행위.
Pull은 당겨오는 것, Push는 밀어넣는 것으로 이해할 수 있다.

Git 설치



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

Git 설치

A screenshot of the GitHub Desktop download page. The page has a white background with a dark teal header. At the top, it says "GitHub Desktop" in bold. Below that are links for "Overview", "Release Notes", and "Help". The main heading is "Simple collaboration from your desktop" in a large, teal font. Below this, there is a paragraph describing GitHub Desktop as a seamless way to contribute to projects on GitHub and GitHub Enterprise. To the right of this paragraph is a blue button that says "Download GitHub Desktop" with "Windows 7 or later" in smaller text below it. Below the button, there is a line of text stating that by clicking the download button, the user agrees to the End-User License Agreement. At the bottom left of the page, it says "Available for Mac and Windows" with "Windows" as a link.

GitHub Desktop

[Overview](#) | [Release Notes](#) | [Help](#)

Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

Available for Mac and [Windows](#)

Download GitHub Desktop
Windows 7 or later

By clicking the Download button you agree to the [End-User License Agreement](#)

<https://desktop.github.com/>

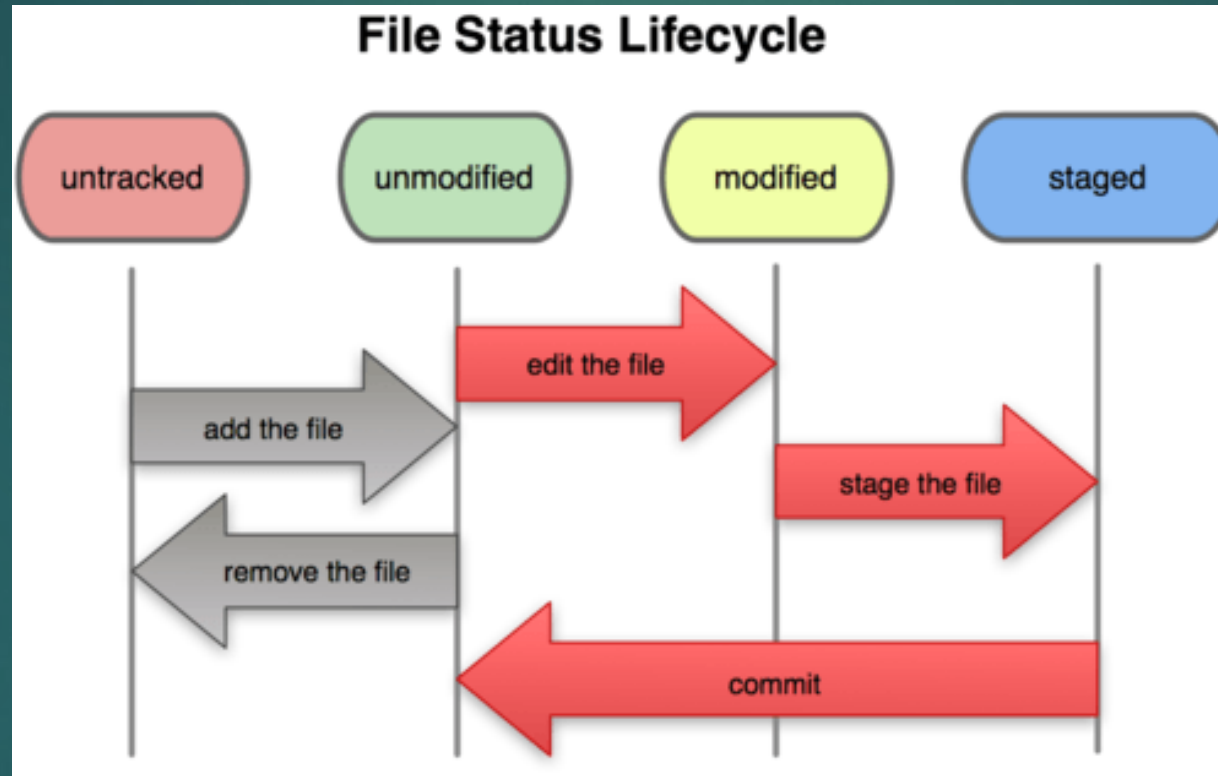
Git의 기본 실행 흐름

- ▶ 1. 저장소 초기화 / 복제
- ▶ 2. 파일 추가 / 수정
- ▶ 3. Commit!
- ▶ 4. 다시 2로

Git 사용해보기! - 1

- ▶ 1. 저장소 만들기 / 기존 저장소를 clone하기
- ▶ 2. 파일을 추가하고 commit하기
- ▶ 3. 히스토리 조회하기

파일 상태 수명 주기



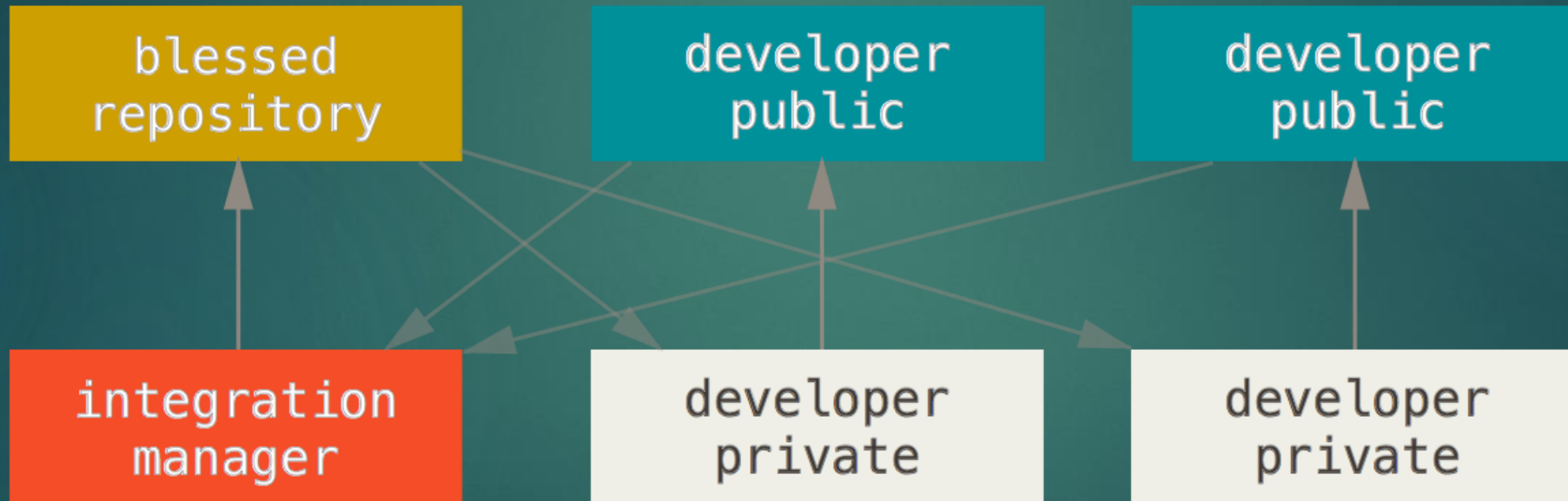
Git 사용해보기! - 2

- ▶ 1. 되돌리기
- ▶ 2. 리모트 저장소

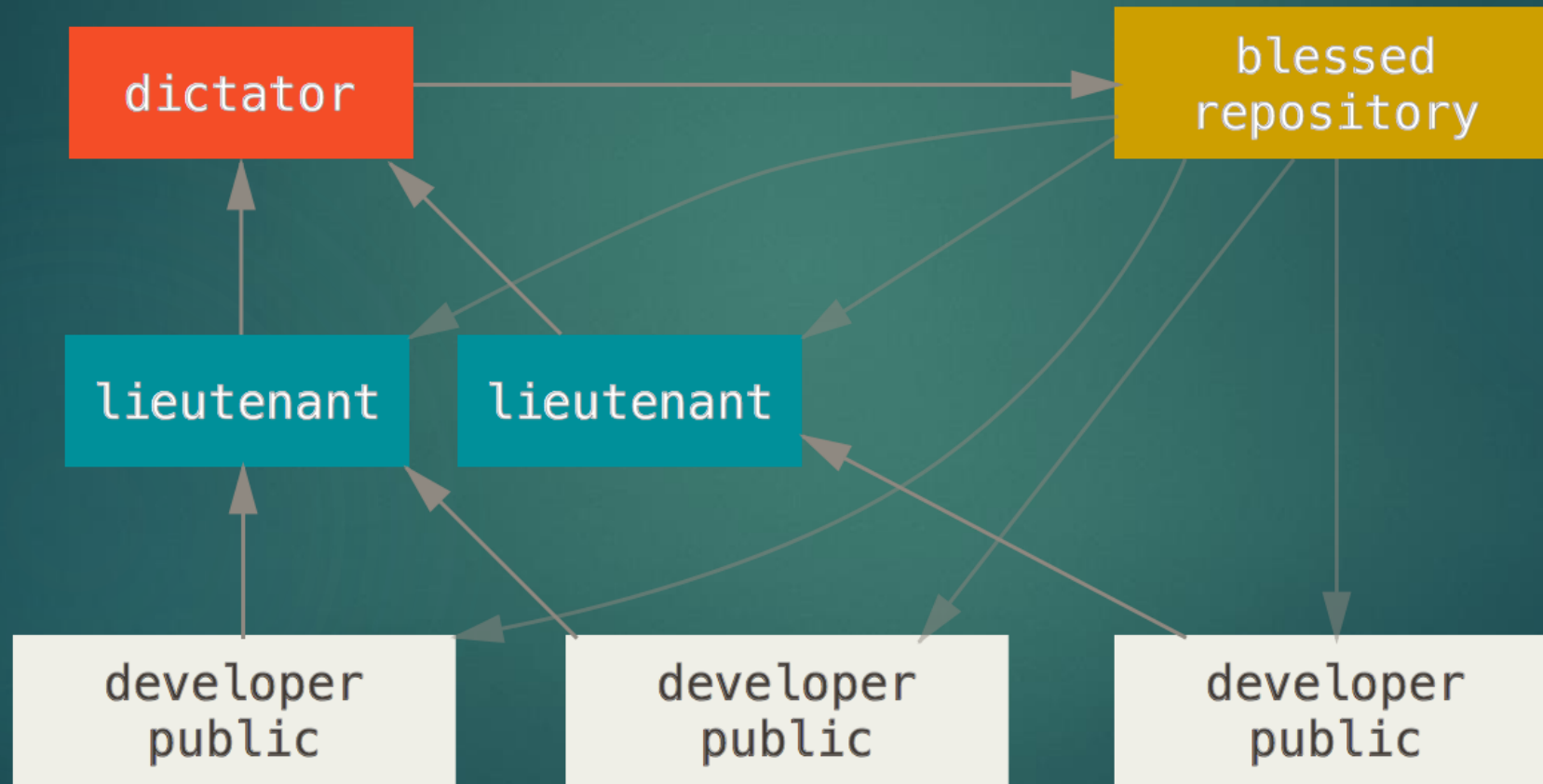
Branch & Merge(Rebase)

- ▶ 비선형적 개발의 핵심
- ▶ 거칠게 설명하자면, 환경을 여러 '가지'로 나누어 작업하고, 작업이 끝나면 나누어진 '가지'를 하나로 합쳐 결과물을 이끌어내는 과정

Integration-Manager Workflow



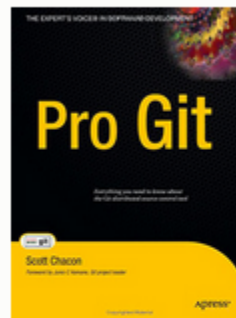
Dictator and Lieutenant Workflow



참고 / 더 많은 정보를 원한다면...

Book

The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress, is available here. All content is licensed under the [Creative Commons Attribution Non Commercial Share Alike 3.0 license](#). Print versions of the book are available on [Amazon.com](#).



1st Edition (2009)

[Switch to 2nd Edition](#)

1. 시작하기

- 1.1 버전 관리란?
- 1.2 짧게 보는 Git의 역사
- 1.3 Git 기초
- 1.4 Git 설치
- 1.5 Git 최초 설정
- 1.6 도움말 보기
- 1.7 요약

<https://git-scm.com/book/ko/v1>

<https://git-scm.com/book/ko/v2>