


1

Version Control System - git



- Version Control System(VCS) 라고 함.
- 사본 생성, 보존, 복원 등을 할 수 있는 도구.
- 다른 사람들과 협업할 때 반드시 필요함.
- CVS(Concurrent Version System), SubVersion, Mercurial, git 등

Project 팀

소스 관리 도구

배포와 저장소

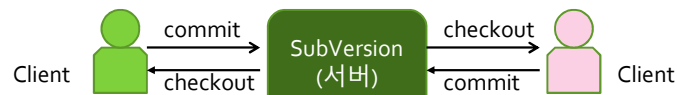
2

VCS 의 종류



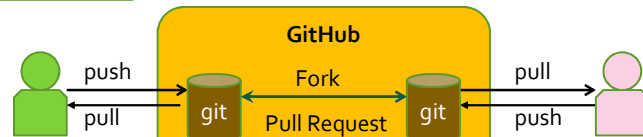
집중형

서버에 하나의 Repository로 집중하여 배치



분산형

다수의 Repository에서 작업하는 형태



3

Git이란?



- 2005년, 리눅스 커널 개발자 리누스 토르발스에 의해 개발
- 완벽한 분산 환경, 수천 개의 브랜치(branch) 작업을 동시에 수행 가능
- 대형 프로젝트의 버전 관리가 가능함.
- 전 세계의 수많은 개발자가 사용하므로 수많은 프로젝트가 존재함.
- Local은 Git을 사용하여 Version control, GitHub 웹 사이트를 이용하여 소스 공유 및 프로젝트 진행

Github :현재 진행되고 있는 대형 프로젝트 : <https://github.com/collections>

4

Git vs Github



Git

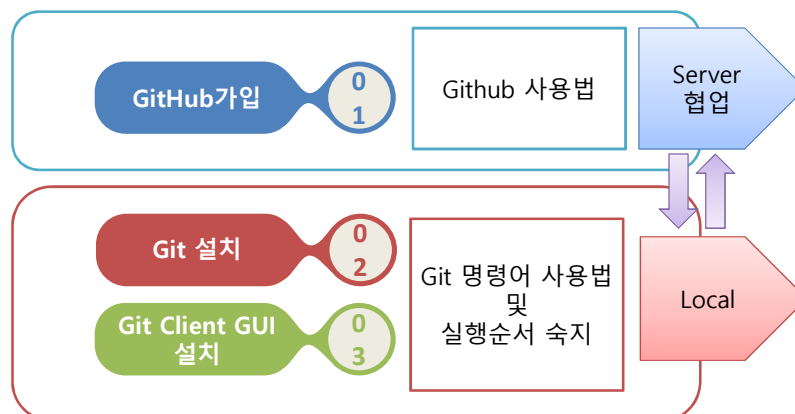
- ✓분산 버전 관리 시스템 - 소스 변경 내역 기록
- ✓Git Repository(저장소) 라고 부르는 데이터 저장소에 소스 코드 등을 넣어서 이용하는 것이다.

GitHub

- ✓Git Repository를 인터넷상에서 제공하는 서비스.
- ✓GitHub에서 공개되는 소프트웨어 소스는 모두 Git으로 관리됨.

5

Git 설치



6

https://git-scm.com/

Download Git

--distributed-is-the-new-centralized

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 **easy to learn** and has a tiny footprint with **lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.

 **Learn Git in your browser for free with Try Git.**

About

The advantages of Git compared to other source control systems.

Documentation

Command reference pages, Pro Git book content, videos and other material.

Downloads

GUI clients and binary releases for all major platforms.

Community

Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.17.0
 Release Notes (2018-04-09)
 Download 2.16.2 for Windows

Pro Git by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Windows GUIs Tarballs
 Mac Build Source Code

7

GUI Clients

GUI Clients

Git comes with built-in GUI tools for committing (**git-gui**) and browsing (**gitk**), but there are several third-party GUIs available for platform-specific experience.

<https://git-scm.com/downloads/guis> tool to this list, just follow the instructions.

All Windows Mac Linux Android iOS

SourceTree
 Platforms: Mac, Windows
 Price: Free
 License: Proprietary

GitHub Desktop
 Platforms: Mac, Windows
 Price: Free
 License: MIT

TortoiseGit
 Platforms: Windows
 Price: Free
 License: GNU GPL

Git Extensions
 Platforms: Linux, Mac, Windows
 Price: Free
 License: GNU GPL

8

<https://backlog.com/git-tutorial/kr/>

누구나 쉽게 이해할 수 있는 Git 입문
버전 관리를 완벽하게 이용해보자

입문 편 | 발전 편 | 찾아보기

누구나 쉽게 이해할 수 있는 Git 입문

버전 관리를 완벽하게 이용해보자!

누구나 쉽게 이해할 수 있는 Git 에 입문하신 것을 환영합니다. 지금부터 Git을 사용한 버전 관리 기능을 함께 공부해 보자구요! 총 3가지의 코스가 준비되어 있습니다. Git 초보자 분들은 '입문편'부터 시작해주세요. Git을 사용한 적이 있으신 분은 '발전편'을 추천 합니다. '어? 뭐였지...?' 싶을 때는 '찾아보기'를 확인하세요.

Git 초보자

Git을 사용한 적이 있는 분

어, 뭐였지? 싶을 때

입문 편
Click ➡

발전 편
Click ➡

찾아보기
Click ➡

9

Git 의 연동

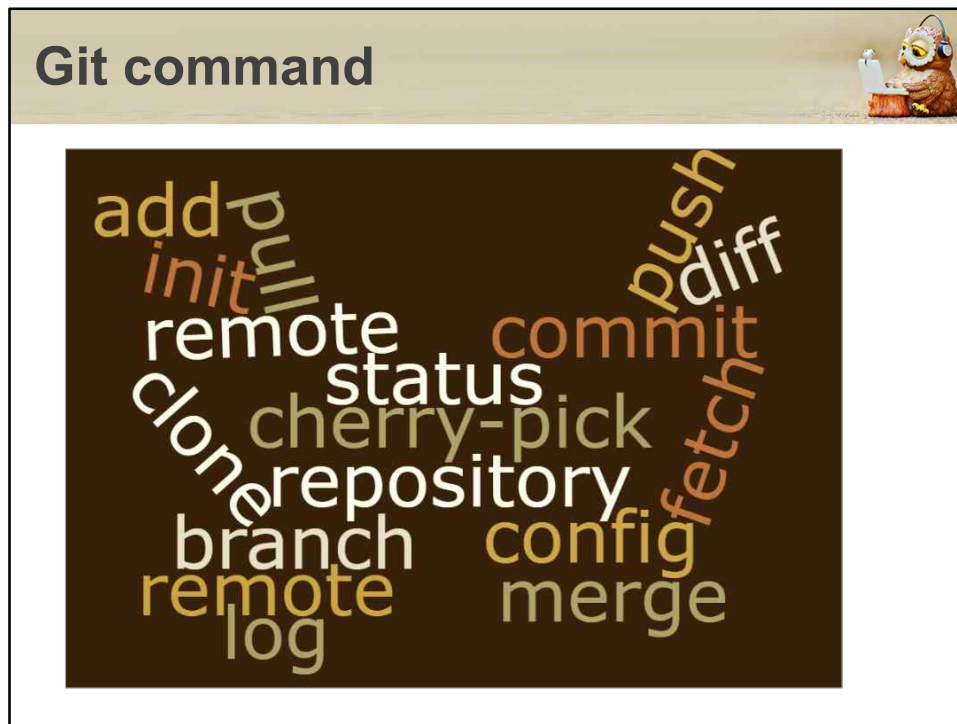
Eclipse & Git

Android Studio

다수의 개발툴에서 Git 과의 연동 기능 제공

다수의 개발툴에서 Git 과의 연동 기능 제공

10



11

Atom + Github 연동(1/2)

1. New Repository 생성 (in github) : ex) testRepo

Create a new repository

A repository contains all project files, including the revision history.

Owner: jerry10004 Repository name: testRepo ✓
2. Copy Repository URL

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/jerry10004/testRepo.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include:
3. Atom 실행 : Packages > Github > Toggle Git Tab

Welcome Guide — E:\CProject — Atom

File Edit View Selection Find Packages Help

Project: CProject

 - Bracket Matcher
 - Command Palette
 - Dev Live Reload
 - Git Diff
 - GitHub
 - Keybinding Resolver

test3.c | Teleme

Toggle Git Tab Ctrl+Shift+9

Toggle GitHub Tab Ctrl+Shift+B

12

Atom + Github 연동(2/2)



- 소스를 저장하기 위한 폴더 생성

```
Initialize git repository in directory
E:\CProject\testRepo
```

- git config 설정 (user.name, user.email)

```
E:\CProject\testRepo>git config user.name 이름
E:\CProject\testRepo>git config user.email 이메일@long.edu
```

- Github 원격 저장소 연결

```
E:\Cproject\testRepo>git remote add origin 'Repository URL'
```

- 파일 저장 후, github Tab에서 add > commit > push 실행

```
master Fetch GitHub Git (0)
```

13

git commands(1/4)



- git 설정하기

```
git config --global user.name "사용자 이름"
git config --global user.email "사용자 이메일"
git config --list
```

- 저장소 만들기

```
git clone URL
git init
```

- 작업 트리(Working Directory) > 인덱스 (Stage) > Git Repository



```
> touch newfile.txt
> git add newfile.txt
> git commit -m "create newfile.txt"
```

14

git commands(2/4)



- git 상태

```
git status
```

```
jerry1004@peace:~/project/gitRepo:> git status
On branch master
nothing to commit, working directory clean
```

- log

```
jerry1004@peace:~/project/gitRepo:> git log
commit c2f31288b885fe60ebaf5edd0a46e81fb24d02ab
Author: jerry1004 <jerry1004@handong.edu>
Date: Mon Apr 8 16:33:03 2019 +0900

    create newfile.txt
```

고유 SHA1 값 : 160bit = 20bytes = 40Hexadecimals = commit id
commit id (앞 7글자)

```
git log
git log --pretty=oneline
git log -graph
git log -p -1
```

15

git commands(3/4)



- **Clone** : Local Repo. <- Remote Repo. (or github repo)

```
git clone "원격저장소 repo URL"
```

- **Remote Repo info**

```
git remote -v
git remote -h
git remote add origin "repo's URL "
git remote remove origin
git remote add apple "repo's URL "
git remote -v
```

- **Push** : Local Repo. -> Remote Repo. (or github repo)

```
git push origin master
```

16

git commands(4/4)



- **pull** : Local Repo. <- Remote Repo. (or github repo)

```
git pull origin master
```

- **fetch** : Local Repo. <- Remote Repo. (or github repo)

```
git fetch
```

- **branch**

Branch image1

```
git branch -h
git branch
git branch apple
git branch
git branch -d apple
git branch banana
git checkout banana
git checkout -b apple
```