

**git (1)**

**Fastcampus Programming SCHOOL**



**git**

# VCS (Version Control System)

== SCM (Source Code Management)

< SCM (Software Configuration Management: 형상관리)

# chronicle of git



## chronicle of git

- Linux Kernal을 만들기 위해 Subversion을 쓰다 화가 난 리누스 토발즈는 2주만에 git 이라는 버전관리 시스템을 만듦  
[git official repo](#)

# Characteristics of git

- 빠른속도, 단순한 구조
- 분산형 저장소 지원
- 비선형적 개발(수천개의 브랜치) 가능

## Pros of git

- 중간-발표자료\_최종\_진짜최종\_15-4(교수님이 맘에들어함)\_언제까지??\_이걸로갑시다.ppt
- 소스코드 주고받기 없이 동시작업이 가능해져 생산성이 증가
- 수정내용은 **commit** 단위로 관리, 배포 뿐 아니라 원하는 시점으로 **Checkout** 가능
- 새로운 기능 추가는 **Branch**로 개발하여 편한 실험이 가능하며, 성공적으로 개발이 완료되면 **Merge**하여 반영
- 인터넷이 연결되지 않아도 개발할 수 있음

## git GUI Clients

- git GUI
- sourcetree
- kraken
- smartGit



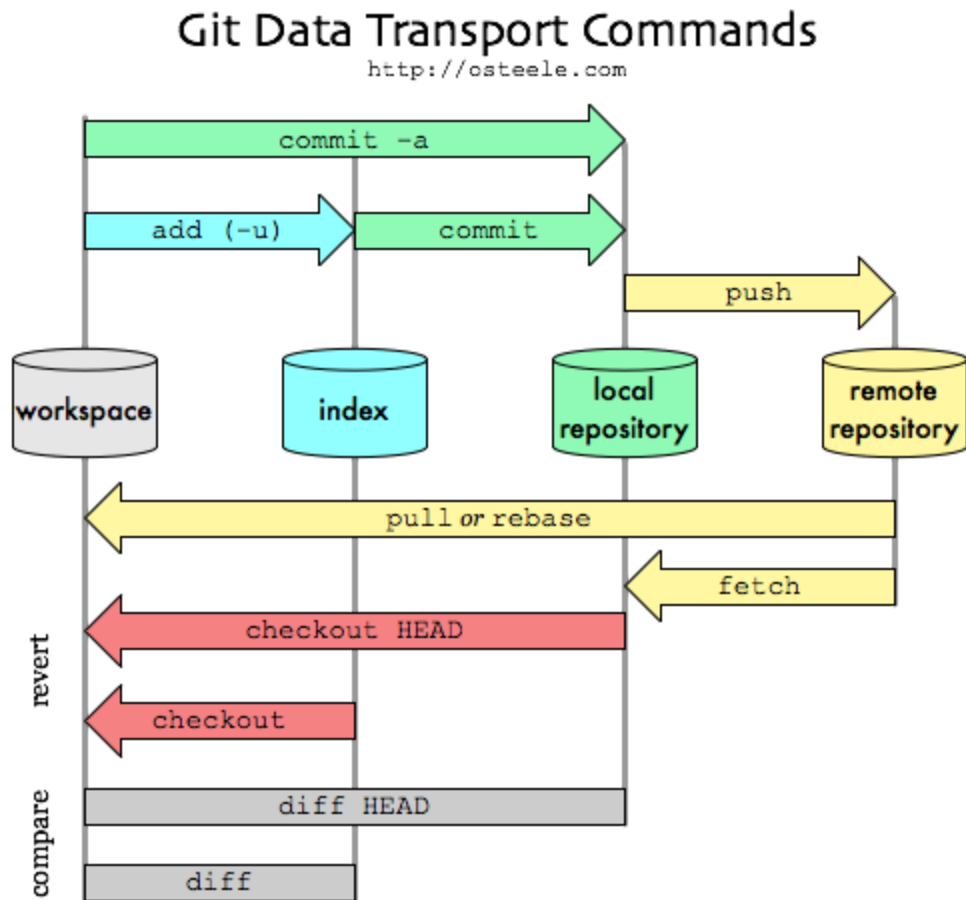
## CLI first

- Source code를 Cloud Platform에서 사용할 경우, CLI 커맨드로 버전관리를 수행해야 합니다.
- CLI 커맨드로 git을 사용할 줄 알면, GUI 도구가 제공하는 기능에 대한 이해가 빠릅니다.
- 확인용도로 GUI를 참고하는 것은 Good^^

## git objects

- Blob: 파일 하나의 내용에 대한 정보
- Tree: Blob이나 subtree의 메타데이터(디렉토리 위치, 속성, 이름 등)
- Commit: 커밋 순간의 스냅샷

# git Process and Command



# Useful package manager for mac

[http://brew.sh/index\\_ko.html](http://brew.sh/index_ko.html)

## install git

- for windows

<https://gitforwindows.org/>

- for MacOS

```
$ brew install git
```

- for Ubuntu

```
$ sudo apt-get install git
```

`$ git --version` 으로 정상적으로 설치되었는지를 확인

## Optional

- github CLI

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

# git is not equal to github



## sign up github

<https://github.com/>

## important!!

- 가입할 email 과 username 은 멋지게
- ~~private repo를 원한다면 \$7/month~~

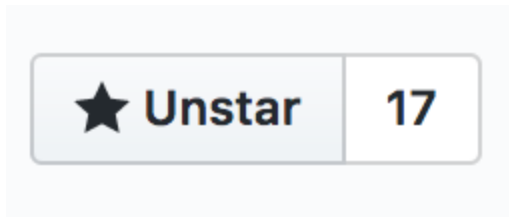
## Useful blog post

<https://ulgoon.github.io/2019/09/09/01-git-first/>

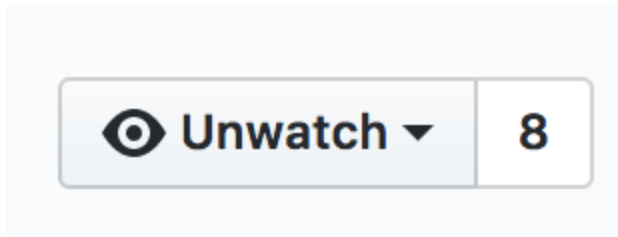


# Important github User Interface

## Star



## watch



# Set configuration

terminal

```
$ git config --global user.name "{github username}"  
$ git config --global user.email "{github email address}"  
$ git config --global core.editor "vim"  
$ git config --global core.pager "cat"  
$ git config --list
```

- optional

```
$ git config --global alias.lg "log --color --graph --  
pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr)  
%C(bold blue)<%an>%Creset' --abbrev-commit"
```

# My First Repo

Let's make your first repo with github

# My First Repo(1)

Let's Create New repo

## My First Repo(2)

```
$ mkdir first-repo && cd first-repo
$ git init
$ git remote add origin https://github.com/{username}/{reponame}.git
$ touch README.md
$ git add README.md
$ git commit -m "docs: Create README.md"
$ git push -u origin master
```

Caution: Do not `git init` on any other directories

## Second push to My First Repo

```
# make some change on README.md  
$ git add .  
$ git commit  
$ git push origin master
```



# Commit Convention

- 커밋 제목은 50자 이내로 요약하여 작성한다
- 제목과 내용사이 한 칸
- prefix를 사용하여 한 눈에 커밋의 용도를 알기 쉽게 한다

```
feat: features  
docs: documentations  
conf: configurations  
test: test  
fix: bug-fix  
refactor: refactoring  
ci: Continuous Integration  
build: Build  
perf: Performance
```

## Commit Convention - example

```
feat: Create server.py to start flask project  
docs: Create README.md  
conf: poetry init  
test: User model CRUD test complete
```

**Start Project with `git clone`**

## start project with clone

- github에서 repo를 생성합니다.

```
$ git clone {repo address}  
$ git add .  
$ git commit  
$ git push
```

# My First Github Pages

github 저장소를 활용해 정적인 사이트 호스팅이 가능

username.github.io

<http://tech.kakao.com/>

<https://spoqa.github.io/>

## sample index page

After create new repo through github,

```
$ git clone https://github.com/username/username.github.io.git
```

Create New file `index.html`

```
$ git add .
```

```
$ git commit -m "first page"
```

```
$ git push origin master
```

## sample index page

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My first gh page</title>
  </head>
  <body>
    <h1>Home</h1>
    <p>Hello, there!</p>
  </body>
</html>
```

# Static Site Generator

- [Jekyll](#): Ruby 기반 정적인 블로그 생성기
  - 설치와 사용이 쉬움
  - 사용자가 많았음
- [Hugo](#): Golang 기반 정적인 블로그 생성기
  - 빠른 속도로 사이트를 생성
  - 사용자 증가 중
- [Hexo](#): Node.js 기반 정적인 블로그 생성기
  - Node.js를 안다면 커스터마이징이 쉬움
  - 빠른 속도로 사용자 증가 중

## Recommand

Hexo > Jekyll > Hugo



## Let's use Hexo

[illegible]

## Requirements

1. git
2. node.js(<https://nodejs.org/en/>)

```
$ npm install -g hexo-cli
```

## Init hexo project

```
$ hexo init <folder>  
$ cd <folder>  
$ npm install
```

## clean && generate static files

```
$ hexo clean && hexo generate
```

## Run hexo server

```
$ hexo server
```

# deploy

```
$ npm install hexo-deployer-git --save
```

```
deploy:  
  type: git  
  repo: <repository url>  branch: [branch] #published  
  message:
```

# .gitignore and .gitattributes

**.gitignore:** 특정파일 추적을 하고 싶지 않을 경우

```
*.java  
*.py[cod]
```

**.gitattributes:** 파일단위, 디렉토리 별 다른 설정을 부여하고 싶을 경우

```
# Avoid conflicts in pbxproj files  
*.pbxproj binary merge=union  
  
# Always diff strings files as text  
*.strings text diff
```

- reference: <https://thoughtbot.com/blog/xcode-and-git-bridging-the-gap>