

# Git and Github

Riot Games

정 종 오(rcheong@riotgames.com)



# 강의 개요

- 소스이력관리의 필요성(협업시스템 그리고 CI(지속적인 통합))
- Git 그리고 GitHub?
- 오픈 소스 트렌드
- Git 설치
- Git Flow 이해
- Git의 사용법
- Git을 편하게 사용할 수 있는 툴 소개(Source Tree)
- 실습
- 과제



# 소스 이력 관리의 필요성



실전코딩과제  
최종

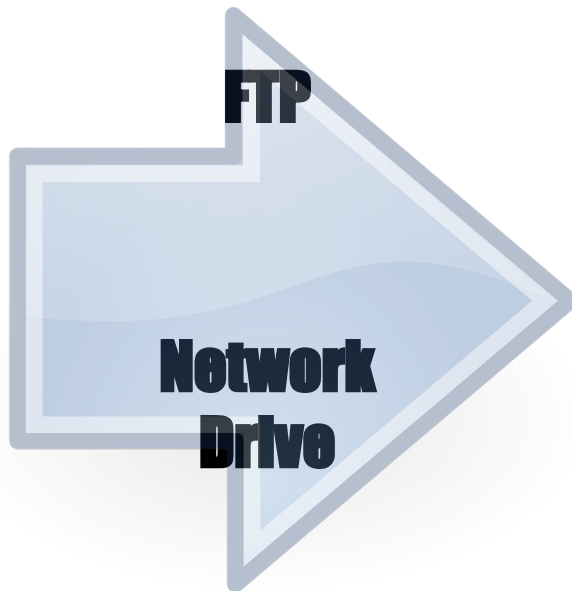


실전코딩1학기과제  
\_진짜최종




실전코딩1학기과제  
\_진짜진짜리얼최종

# 혼자가 아니라 함께 서버를 이용한다면?







누가  
뒤통수쳐야?



내 파일  
없어졌는데?



3일전꺼  
갖고 있는 사람?



좋은 말로  
어떻게?  
자수해라

누가? 언제? 어떤 파일을?

어떻게?      변경했는지를  
알 수 있다.

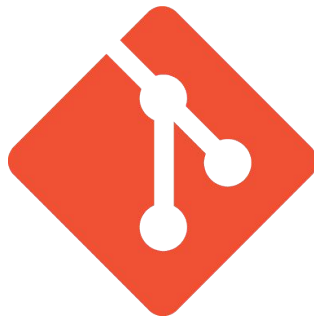


# 현재 사용되고 있는 대표적인 소스이력관리 시스템



중앙집중식  
버전관리

VS

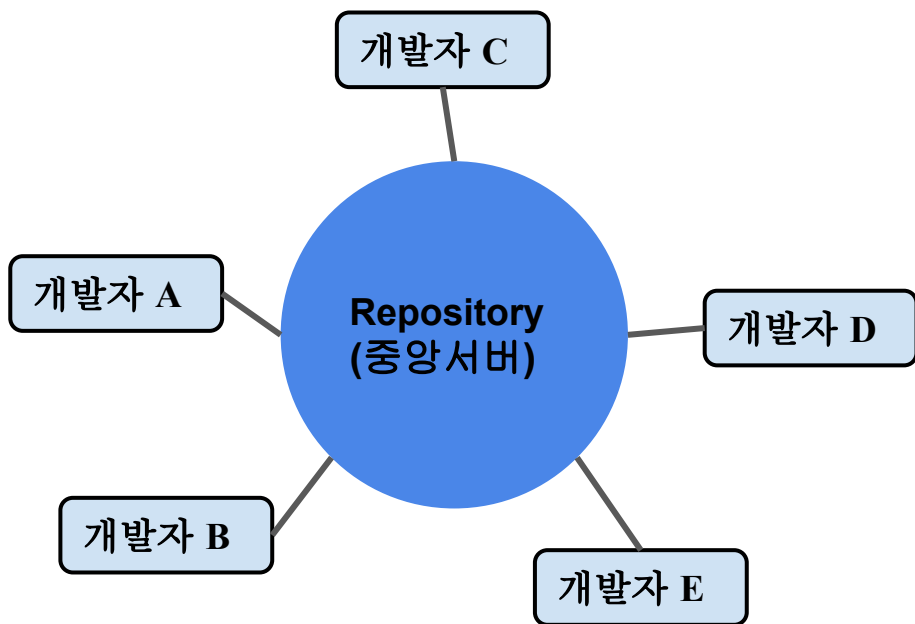


git

분산  
버전관리



# 중앙집중식 버전관리(CVS, SVN)

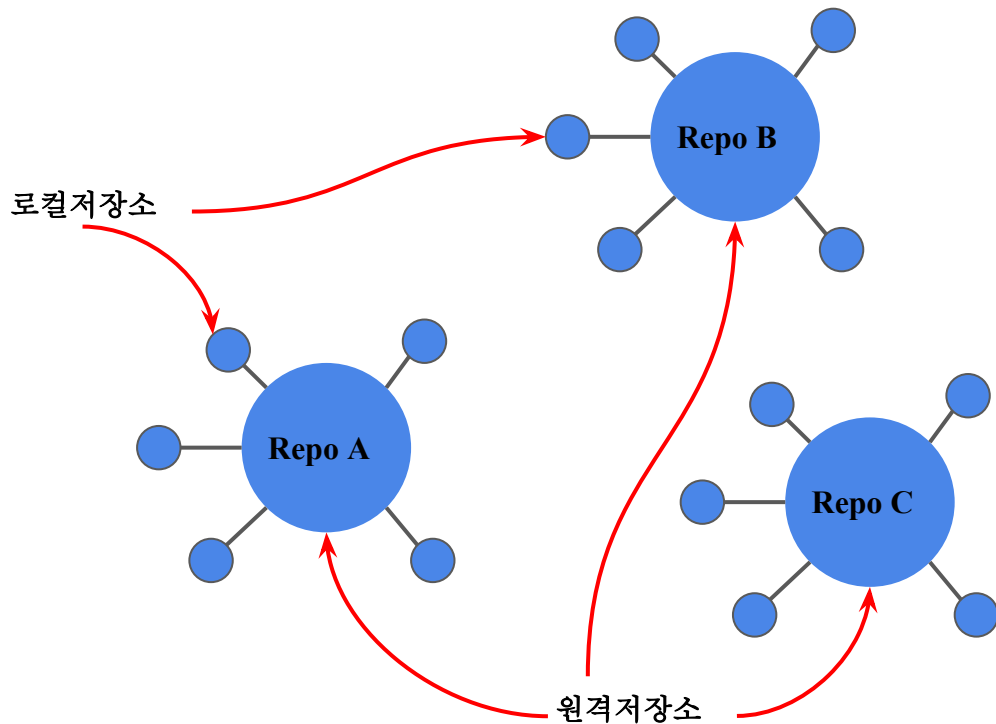


- 중앙서버를 통해 버전 관리
- 서버와의 통신이 필수
- 개발을 시작하기 전에는  
항상 최신 소스를 내려받는  
습관이 필요





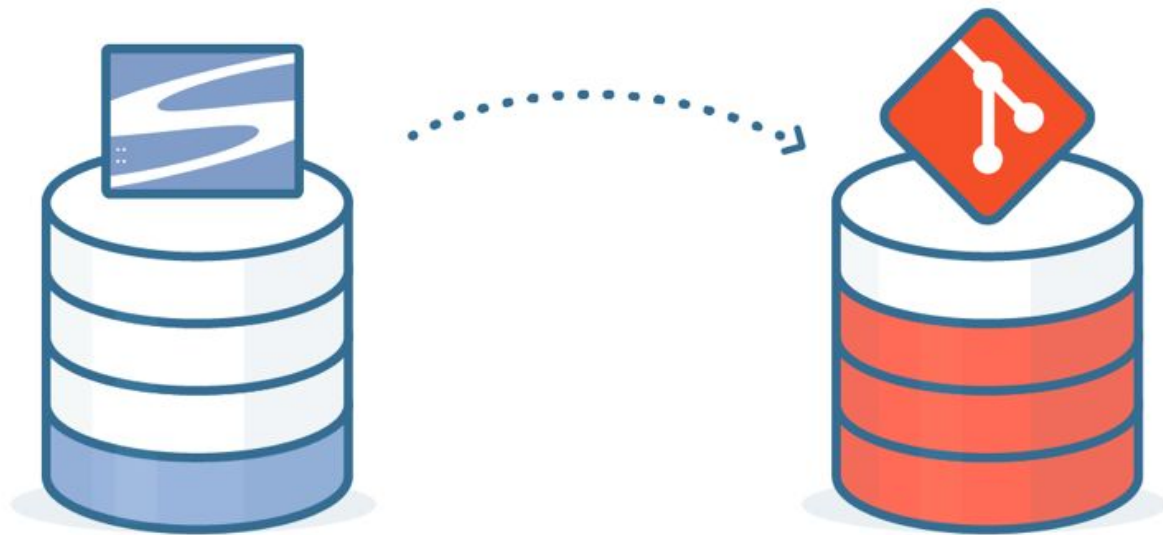
# 분산 버전 관리(Git, Mercurial, Bazaar)



- 소스가 저장 관리되는 곳이 여러 곳에 위치
- 로컬 저장소와 원격저장소로 구분
- 로컬 저장소는 다수의 원격 저장소와 연결할 수가 있음



요즘 추세는?



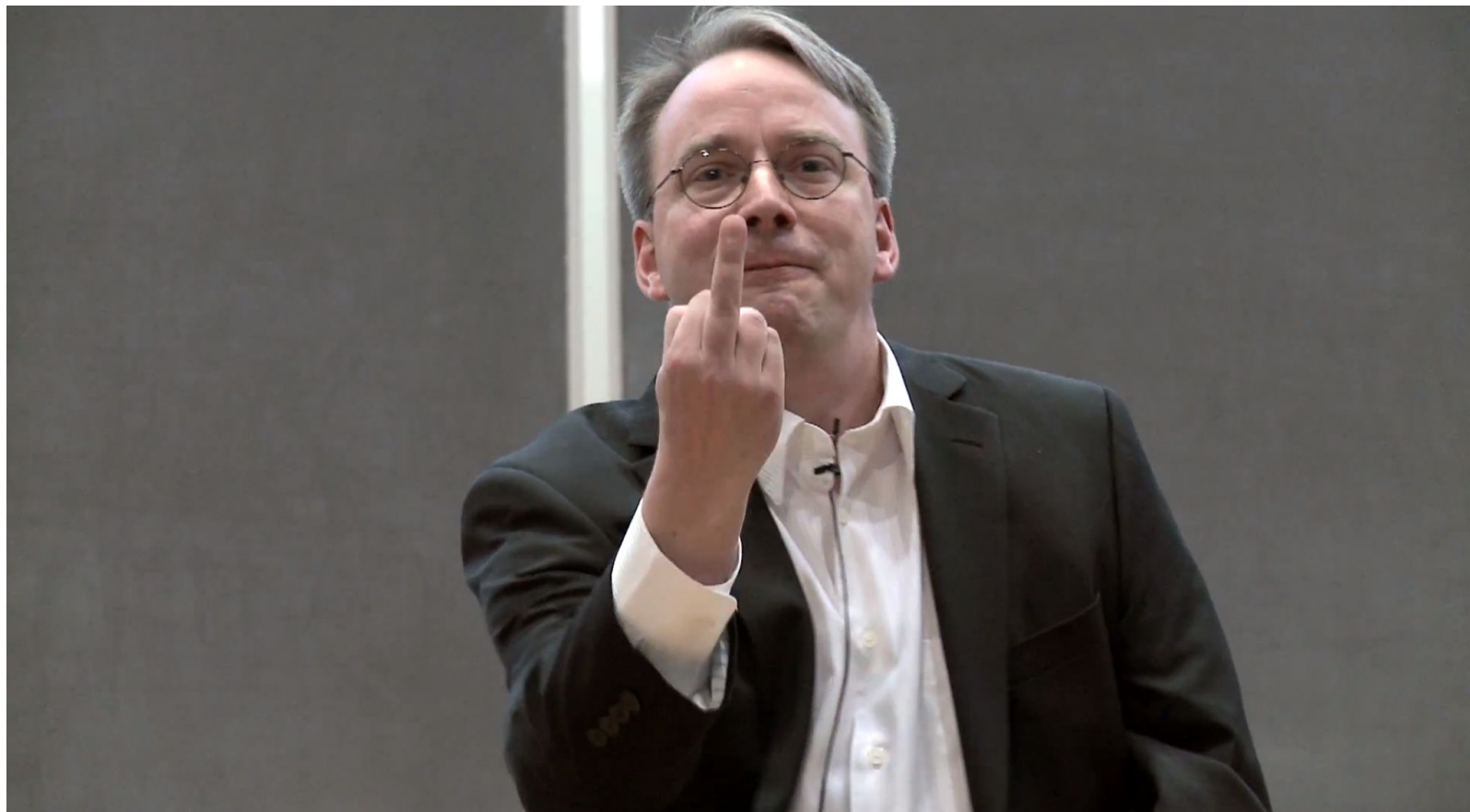
# git



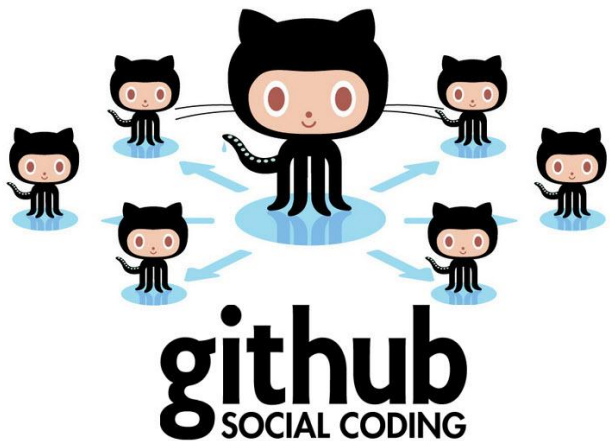
리누스토발즈가 커널 소스 이력관리를 위해 BitKeeper라는 상용툴을 쓰다가 관계가 틀어진 후 리눅스 2주만에 완성.

- 빠른 속도와 단순한 구조
- 비선형적 개발
- 완벽한 분산 시스템
- 대규모 프로젝트에 적합





# GitHub



- Git은 분산이력관리 시스템 그 자체
- GitHub은 git을 호스팅해주는 웹서비스
- 웹에서 Git 기반의 소스 이력관리를 직접 하거나 모니터링이 가능
- 수많은 오픈 소스가 등록되어 있음
- 소스코드의 위키디피아
- 공개 Repository는 무료 사용
- 이력서에 GitHub 주소 한줄의 효과는 기대 이상



# 오픈소스

- 오픈소스 하면 떠오르는 것?
- 공짜는 허접한 거, 비싸게 좋은 거?
- 혼자보다는 다수의 개발자가 함께 개발하는 것이 완성도가 높다
- 소스를 공개하고 다수의 개발자가 프로젝트에 참여가 가능
- GitHub은 오픈 소스의 바다
- 만약 컨트리뷰터나 커미터가 된다면 경력에 상당한 도움



# Git 사용하기 위한 환경 구성

## Windows, OSX, Linux 용 클라이언트

클라이언트 다운로드 및 설치(OSX, Windows)

[\*\*https://git-scm.com/download\*\*](https://git-scm.com/download)



[About](#)

[Documentation](#)

[Blog](#)

**Downloads**

[GUI Clients](#)

[Logos](#)

[Community](#)

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## Downloads



Older releases are available and the [Git source repository](#) is on GitHub.



### GUI Clients

Git comes with built-in GUI tools ([git-gui](#), [gitk](#)), but there are several third-party tools for users looking for built : 짓다, 쌓아올리다, 세우다, 건축하다, 부설하다 experience.

[View GUI Clients](#) →

### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in and print projects.

[View Logos](#) →

### Git via Git

If you already have Git installed, you can get the latest development version via Git itself:

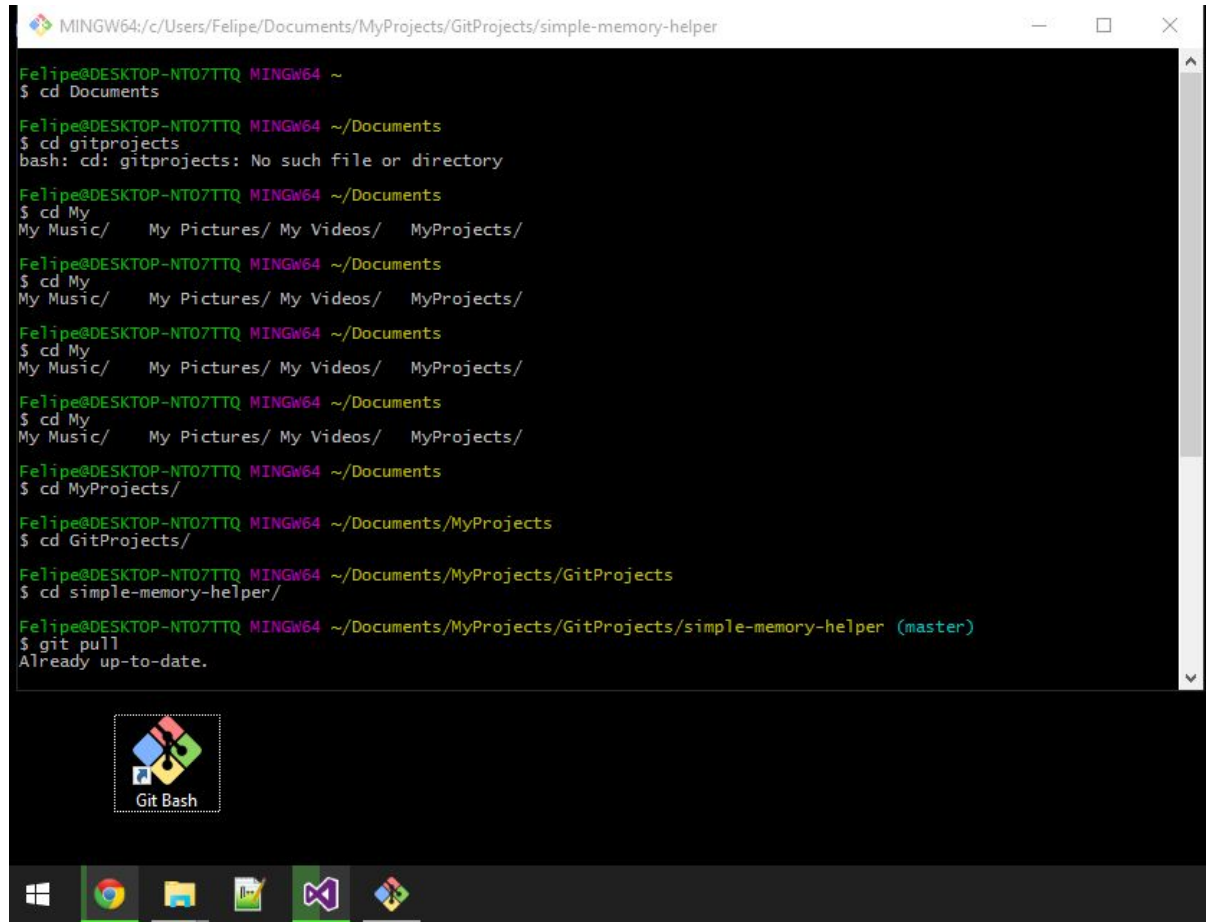
```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).





# Windows



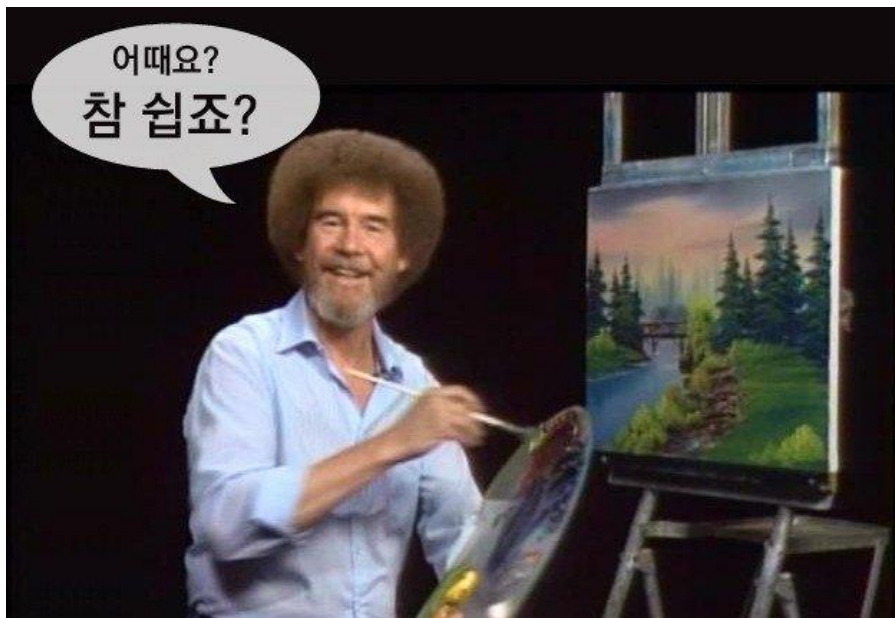
The screenshot shows a Windows desktop environment. A Git Bash terminal window is open, displaying the following commands and output:

```
MINGW64/c:/Users/Felipe/Documents/MyProjects/GitProjects/simple-memory-helper
Felipe@DESKTOP-NT07TTQ MINGW64 ~
$ cd Documents
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd gitprojects
bash: cd: gitprojects: No such file or directory
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/  My Pictures/ My Videos/  MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/  My Pictures/ My Videos/  MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/  My Pictures/ My Videos/  MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/  My Pictures/ My Videos/  MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents/MyProjects
$ cd GitProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents/MyProjects/GitProjects
$ cd simple-memory-helper/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents/MyProjects/GitProjects/simple-memory-helper (master)
$ git pull
Already up-to-date.
```

Below the terminal window, there is a taskbar with several icons: the Windows Start button, Google Chrome, File Explorer, a document icon, Visual Studio Code, and the Git Bash icon. The Git Bash icon is a square with a diamond shape inside, divided into four colored triangles (blue, red, green, yellow) with a black 'X' in the center.

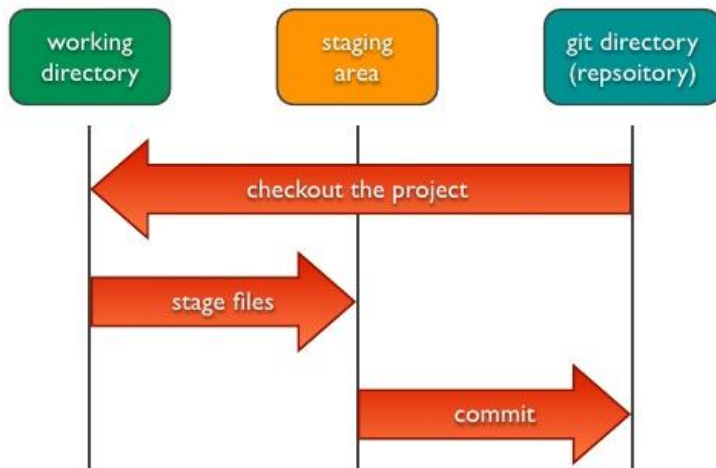


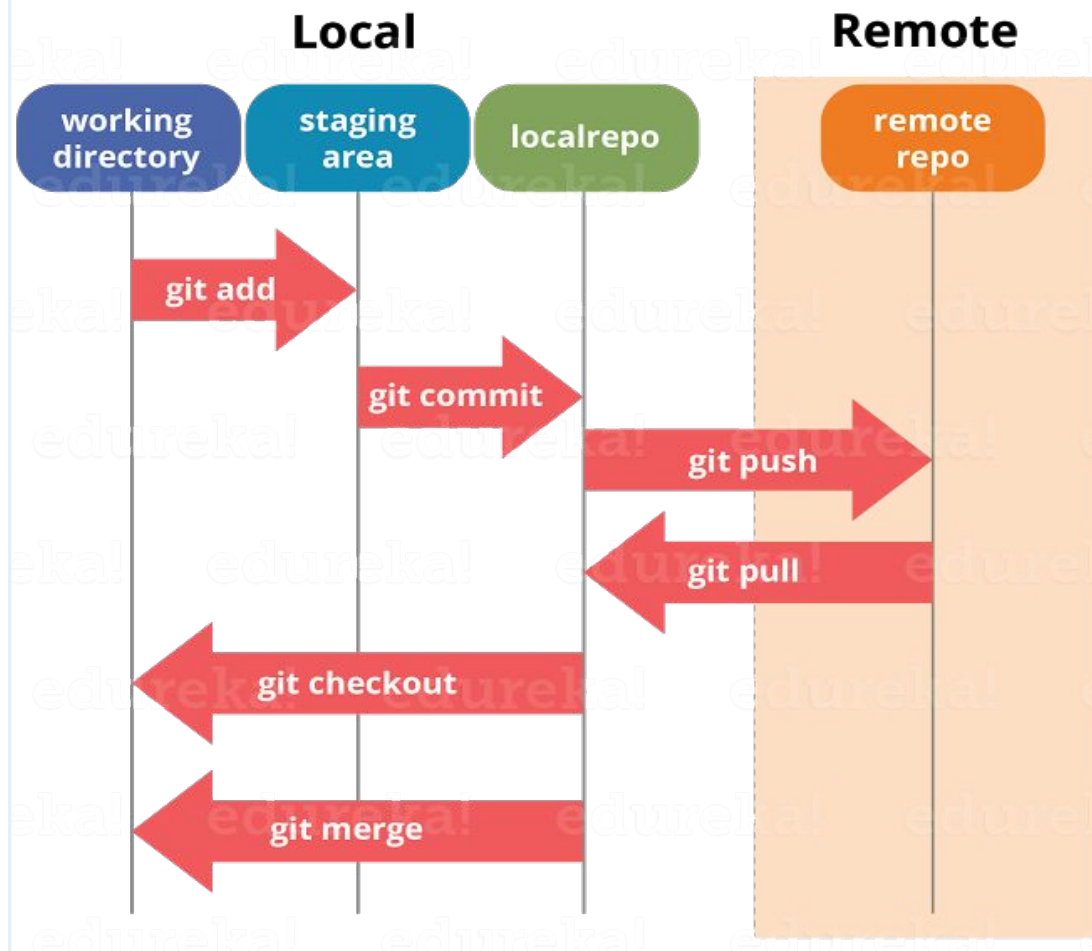
# Git 이 설치되었다면 기본적인 작업 준비는 끝



# Git Flow 이해하기

## local operations





# 브랜치(Branch)와 머지(Merge)



Branch와 Merge는  
하나의 프로젝트를  
여러 명이 함께  
개발해야 할 때 꼭 필요한  
존재이며, 소스이력관리의  
핵심 구성 요소



내가 이 부분을  
작업할게

나중엔 하나로  
다 합쳐야 한다고  
그러니까 잘  
분배해야 해

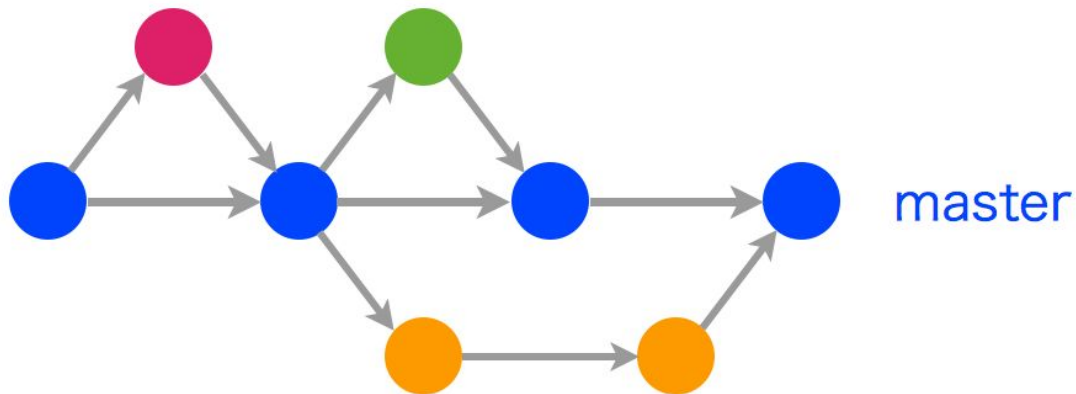
난 이쁘니까  
이름만 올려줘

내가 제일  
잘하니까  
중요한 부분은  
내가 할게



# 브랜치(Branch)와 머지(Merge)

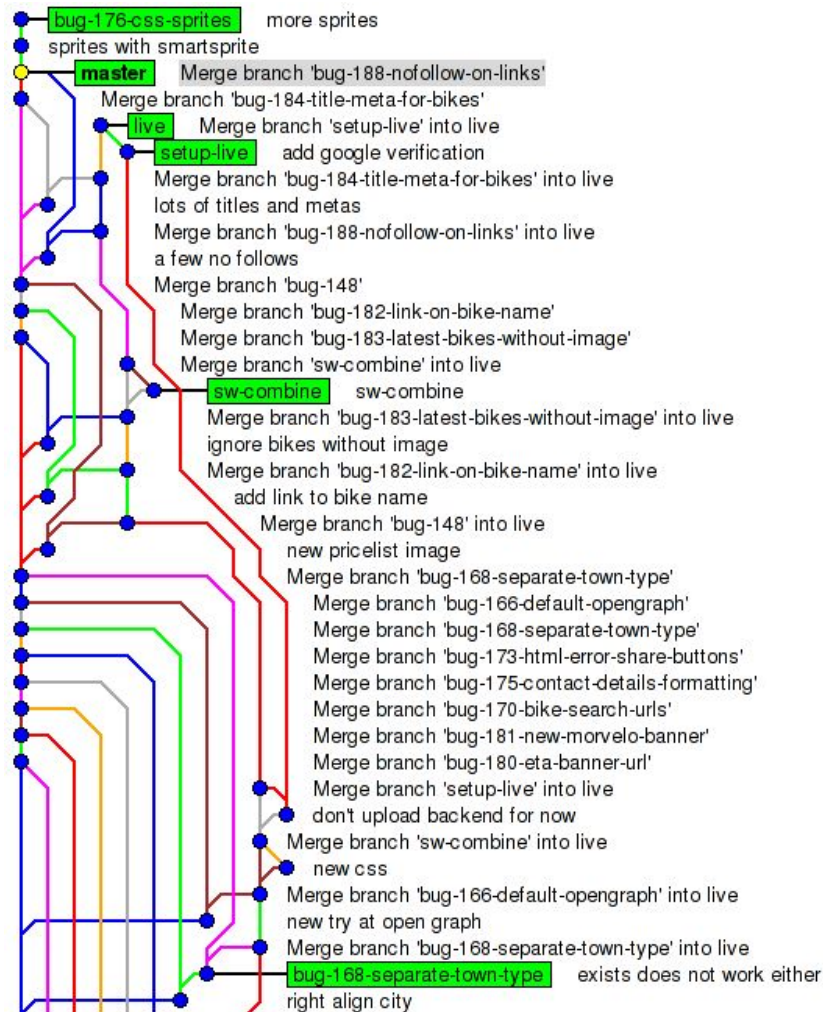
## GitHub flow

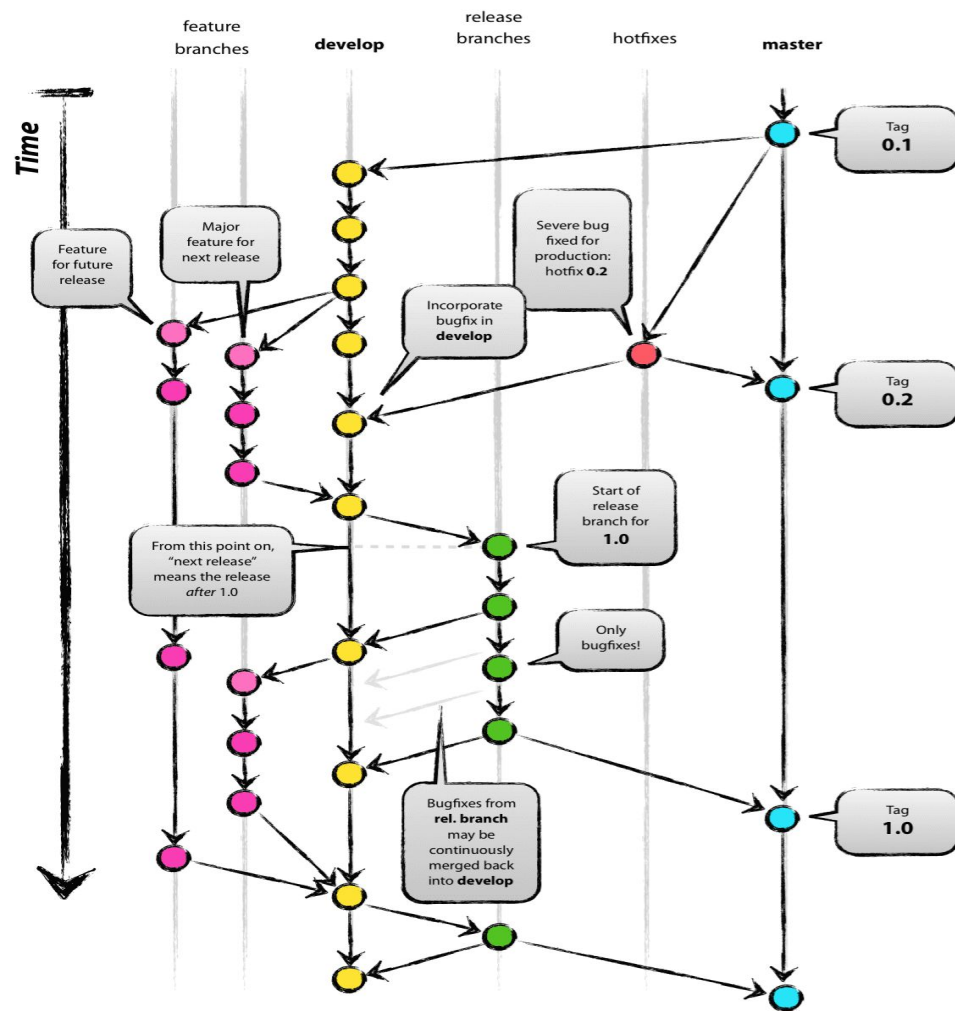


Git flow 데모 : <https://guides.github.com/introduction/flow/>

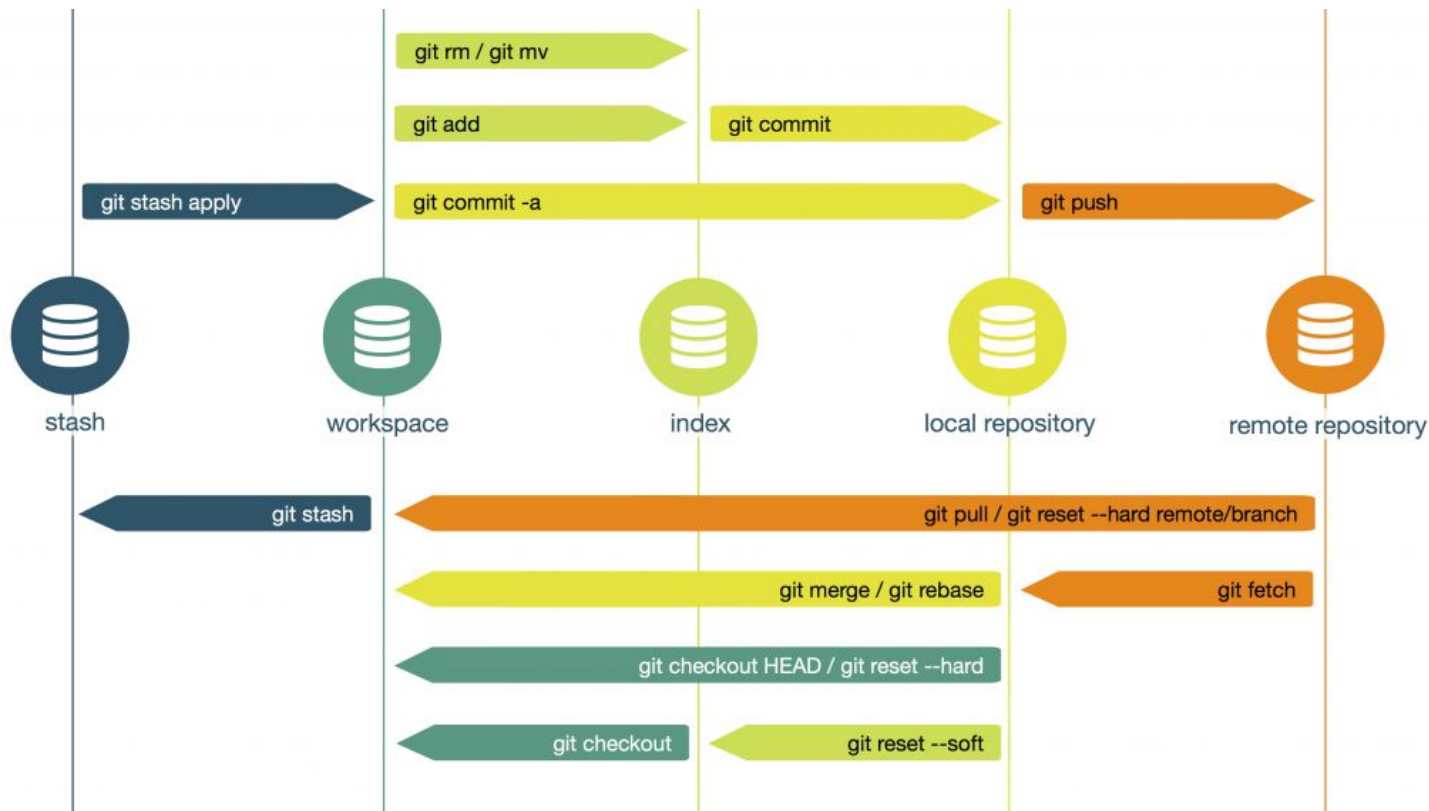




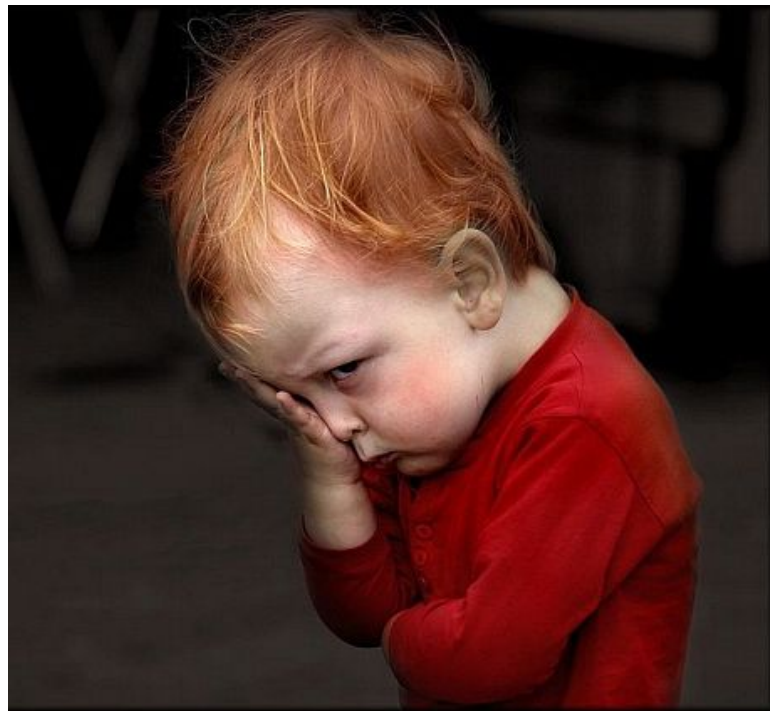




# Git 주요 명령들(이것만 익히면 충분합니다)



음! 뭔지 점점  
복잡해지는 듯 하다.



하나씩 해보면 전혀  
어렵지 않습니다.  
자 실습을 시작할게요.



## 사용자정보 config 설정

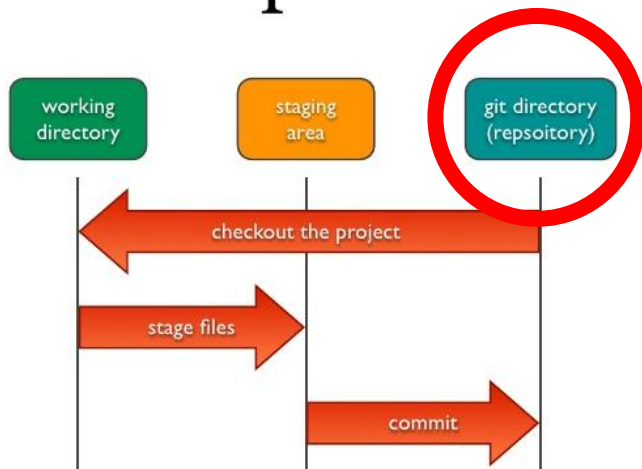
```
$ git config --global user.name "John Doe"
```

```
$ git config --global user.email johndoe@example.com
```



# git 저장소 생성

local operations

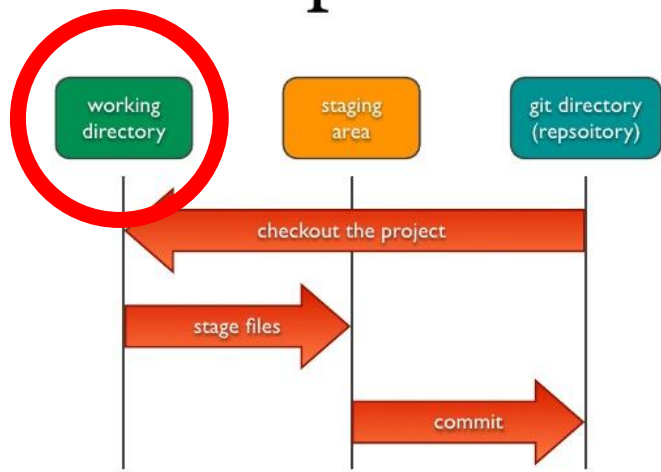


**\$ git init**



# git으로 관리할 파일 생성

## local operations



**\$ touch README.md**

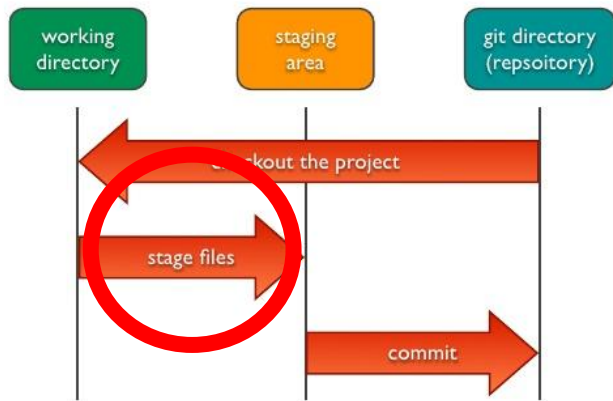
일단 빈 꺾데기 파일을 생성합니다.





# git Stage공간에 등록

local operations



**\$ git add README.md**



git 상태 확인

# \$ git status

친절하게 새로운 파일이 등록되었다고 알려줍니다.

```
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   README.md
```



**Tip! - Windows 버전에서 한글이 깨질 경우**

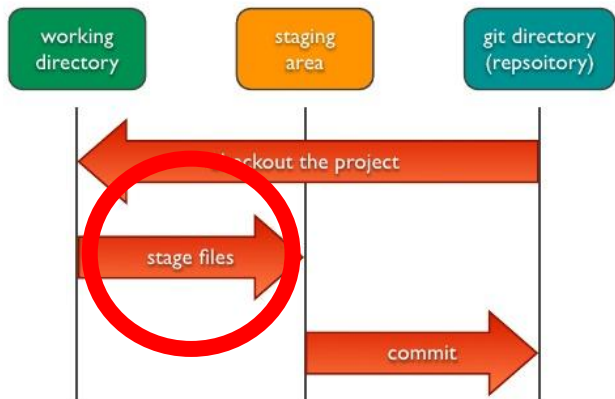
**\$ git config --global core.quotePath false**

셸에서는 한글을 특수문자로 인식해서 `espace` 처리를 하기 때문에, 이를 사용하지 않겠다는 옵션 설정을 사용합니다.



# 다수의 파일을 등록

local operations



```
$git add *
```

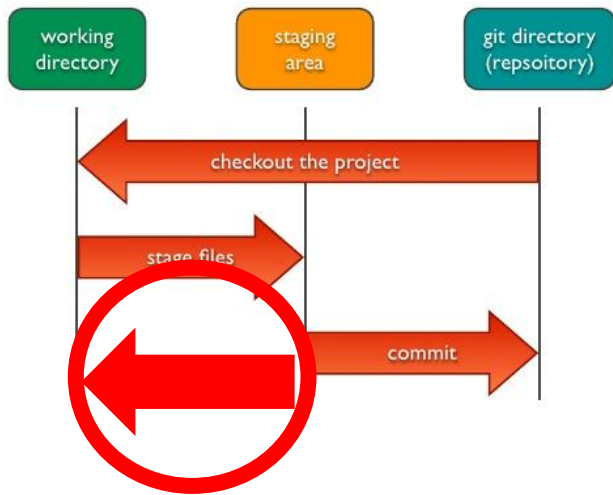
```
$git add *.java
```

```
$git add folder_name
```



# git add를 취소하고 싶을 때

## local operations



**\$git rm --cached hello.txt**  
**\$git reset hello.txt**



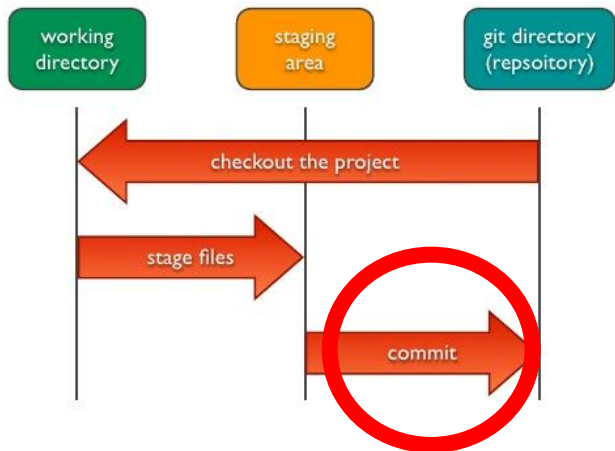
## 그럼 미션!!

- 자신이 좋아하는 음식 이름 파일 5개 생성  
예) 짜장면.txt 부대찌개.txt
- git add 명령을 통해 stage에 등록
- git status 명령으로 최종 등록 상태 확인
- [rcheong@riotgames.com](mailto:rcheong@riotgames.com) 으로 파일을 첨부하여 메일 전송



# git 저장소에 등록하기

## local operations

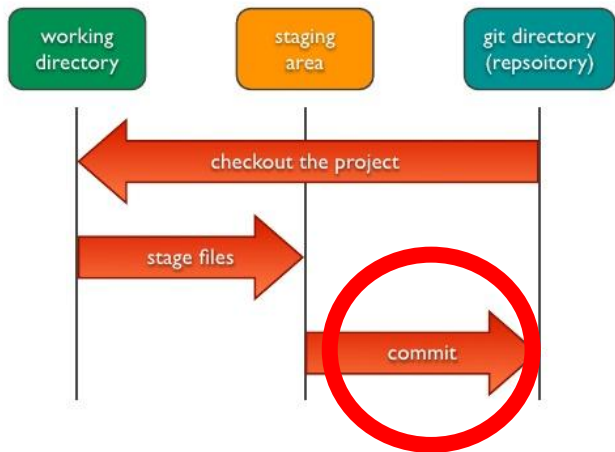


**\$ git commit**



# 메시지와 함께 바로 commit 하기

## local operations



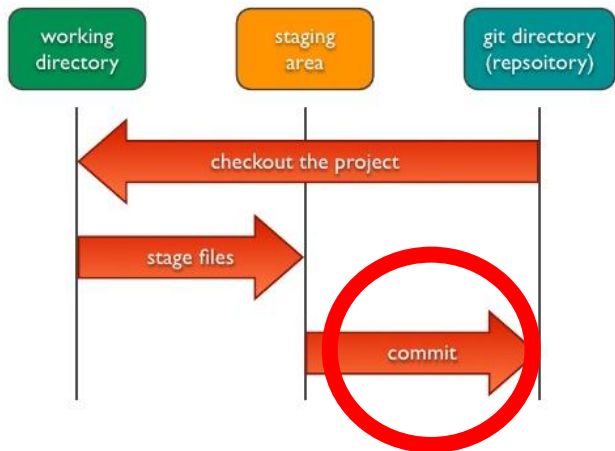
**\$git commit -m “First Commit”**





# add 하면서 메시지와 함께 바로 commit 하기

## local operations



**\$ git commit -a -m "First Commit"**  
**\$ git commit -am 'First Commit'**



# commit 할 때 알아두면 좋을 몇 가지

- **commit**은 가능한 작은 단위로 자주 해두자.
- **commit** 메시지는 **50**자 이내로 간결하게 적용하며, 첫 글자는 대문자를 사용한다.  
(물론 영어로 작성시)
- 긴 내용이 필요할 경우에는 제목이후 한 줄을 비우고 내용을 작성한다.
- 메시지 마지막에는 마침표(.)를 넣지 않는다.
- 메시지 제목은 명령문 스타일의 간결한 문장을 사용한다.

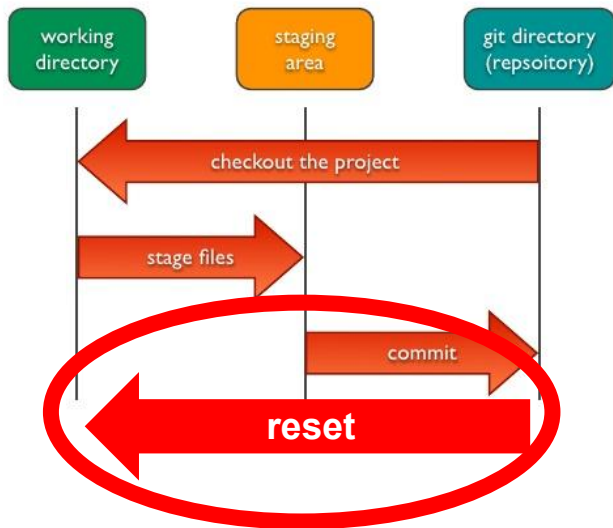
예) **Add file helloWorld.java, Remove deprecated methods, Refactor sub class for member**

- 어떻게 보다는 무엇과 왜를 설명한다.



# reset 명령을 통해 commit 취소하기

## local operations



옵션	HEAD 위치	Stage	작업 파일
soft	변경	변경 안 함	변경 안함
mixed	변경	변경	변경 안함
hard	변경	변경	변경

**\$ git reset --hard HEAD~1**

**\$ git reset --soft HEAD~1**

**\$ git reset --hard HEAD^**



# 저장소 버전과 현재 작업 중인 파일의 비교



**\$git diff**

**\$git diff filename**

**\$git diff --color-words**

**\$git diff --word-diff**

# commit log 확인

```
MINGW64:/d/dev/workspace2/git_lecture
$ git log -p -2
commit 71e512379bf42e5603c853ae7ae870c52dbcc50e
Author: Ryan Cheong <lastboy0@gmail.com>
Date: Sat Mar 18 21:17:01 2017 +0900

    commit test files

diff --git a/test1.txt b/test1.txt
deleted file mode 100644
index f2e0aac..0000000
--- a/test1.txt
+++ /dev/null
@@ -1,4 +0,0 @@
-hello
-world
-good
-news

commit c39e1a02abaf17b46d55310f93380e681a20c91
Author: Ryan Cheong <lastboy0@gmail.com>
Date: Sat Mar 18 20:55:56 2017 +0900

    commit test

diff --git a/test1.txt b/test1.txt
index e2f9b23..f2e0aac 100644
--- a/test1.txt
+++ b/test1.txt
@@ -1,3 +1,4 @@
 hello
 world
 good
+news
```

**\$git log**

**\$git log -p**

**\$git log -p -2**



# 여기서 잠깐!!! .git 폴더의 정 체

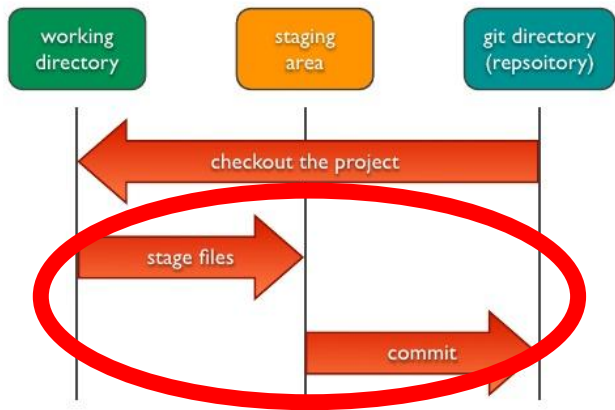
```
$ ls  
HEAD  
branches/  
config  
description  
hooks/  
index  
info/  
objects/  
refs/
```

- git으로 관리되고 있는 root 폴더에 위치
- git으로 관리하기 위한 모든 정보들을 담고 있다.
- 본인이 생성하지 않은 모르는 폴더가 생겼다면 무심코 지웠다가는 지옥을 맛보게 됨.



# git 상에서 파일 삭제

## local operations



```
$git rm hello.txt
```

```
$git commit -m 'delete hello.txt file'
```



작업 디렉토리에서 파일을 삭제했을 때 복구

**\$rm git\_study.txt** ← 삭제

**\$git checkout git\_study.txt** ← 복구

**\$ls** ← 확인



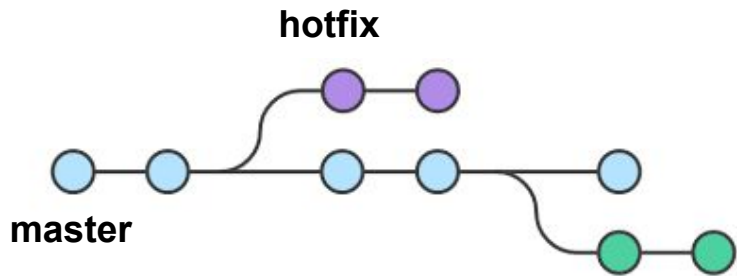


## 파일명 변경하기 - git mv

```
$git mv hello.txt hello_new.txt  
$git commit -m 'Rename hello.txt file'
```



# 브랜치(Branch) 만들기



**\$git branch hotfix**

**\$git checkout hotfix**

**\$git checkout -b hotfix**

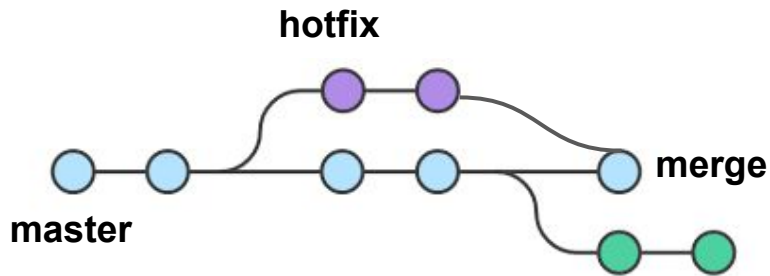
**\$git branch**

**\$git branch -r**

**\$git branch -a**



# 머지(Merge) 그리고 브랜치 삭제



**\$git checkout master**  
**\$git merge hotfix**  
**\$git branch -d hotfix**



# 충돌 해결하기

<<<<<<< HEAD

환영합니다.

=====

안녕하세요.

충남대 학생 여러분  
만나서 반갑습니다.

>>>>>>> hotfix



# Git branch, merge demo

<http://learnbranch.urigit.com/?demo>



## .gitignore 사용하여 git 관리대상에서 제외하기

- Git 최상위 디렉토리로 이동(`git init` 으로 지정한 폴더)
- .gitignore 파일이 있다면 편집기로 열고  
없다면 새로 만듦
- `$vi .gitignore`



# .gitignore 샘플

```
# built application files
*.apk
*.ap_
# files for the dex VM
*.dex
# Java class files
*.class
# generated files
bin/
gen/
# Local configuration file (sdk path, etc) local.properties
# Eclipse project files
.classpath
.project
# Proguard folder generated by Eclipse proguard/
# IntelliJ project files
*.iml
*.ipr
*.iws
.idea/
```



이제 GitHub을 살펴보도록  
하겠습니다.

지금까지 실습한 부분에서  
빠진 중요한 부분들이 있습니다.





# GitHub의 모습을 한번 살펴봅시다

The screenshot shows the GitHub repository page for 'git / git'. A red circle highlights the repository statistics: Watch (1,546), Star (16,598), Fork (9,369). A blue circle highlights the repository metadata: 46,198 commits, 5 branches, 631 releases, 1,084 contributors, and GPL-2.0 license. The page also shows the repository description, navigation tabs (Code, Pull requests, Projects, Pulse, Graphs), and a list of recent commits.

git / git

Watch 1,546 Star 16,598 Fork 9,369

Code Pull requests 66 Projects 0 Pulse Graphs

Git Source Code Mirror - This is a publish-only repository and all pull requests are ignored. Please follow Documentation/SubmittingPatches procedure for any of your improvements.

c shell

46,198 commits 5 branches 631 releases 1,084 contributors GPL-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

commit	message	time
gitster	Third batch after 2.12	9 hours ago
	Documentation	9 hours ago
	block-sha1	sha1: provide another level of indirection for the SHA-1 functions
	builtin	Merge branch 'js/realpath-pathdup-fix'
	ci	travis-ci: build documentation
	compat	Merge branch 'js/mingw-isatty'
	contrib	Merge branch 'ss/remote-bzr-hg-placeholder-wo-python'
	ewah	Merge branch 'jk/ewah-use-right-type-in-sizeof'
	git-gui	Merge tag 'gitgui-0.21.0' of git://repo.or.cz/git-gui
	gitk-git	Merge git://ozlabs.org/~paulus/gitk
	gitweb	Merge branch 'ab/gitweb-abbrev-links'
	mergetools	Merge branch 'da/mergetool-xxdiff-hotkey' into maint
	perl	Merge branch 'svn-escape-backslash' of git://bazaar.mins.org/git-svn



알림서비스 사용자

Fork한 수



즐거찾기나 별점 부여



Commit된 횟수



배포  
횟수



라이센스



46,198 commits

5 branches

631 releases

1,084 contributors

GPL-2.0

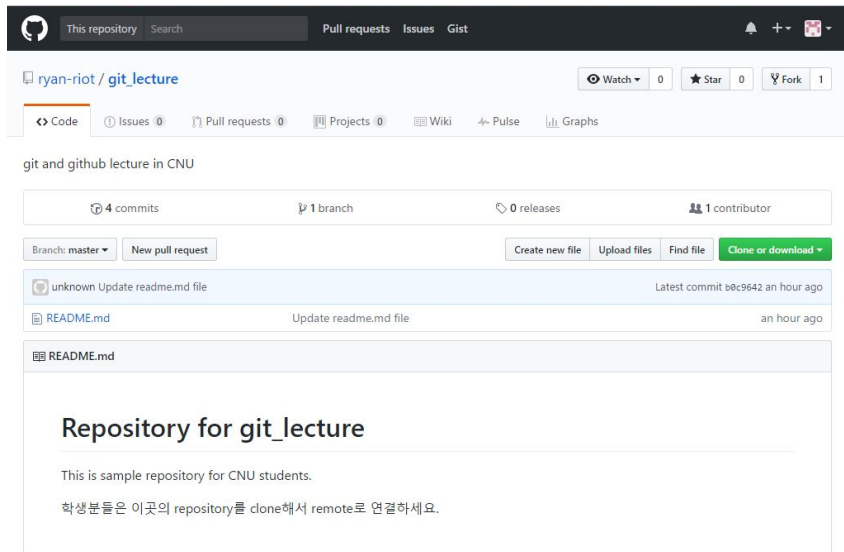
갖고 있는  
branch 수



기여자  
들



# remote 저장소를 fork 하기

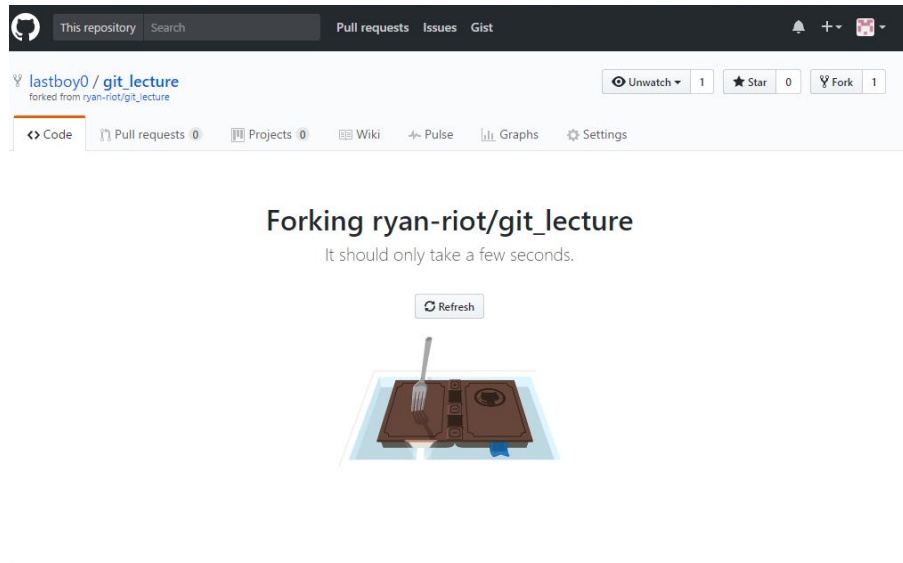


This screenshot shows the GitHub repository page for `ryan-riot/git_lecture`. The repository is described as "git and github lecture in CNU". It has 4 commits, 1 branch, 0 releases, and 1 contributor. The current branch is `master`. A recent commit by "unknown" updated the `readme.md` file. The README content is as follows:

```
Repository for git_lecture
```

This is sample repository for CNU students.

학생분들은 이곳의 repository를 clone해서 remote로 연결하세요.



This screenshot shows the GitHub repository page for `lastboy0/git_lecture`, which is a fork of `ryan-riot/git_lecture`. The page displays the title "Forking ryan-riot/git\_lecture" and the message "It should only take a few seconds." Below this, there is a "Refresh" button and an illustration of a fork being used to create a new repository. The footer of the page includes the GitHub logo and the text "© 2017 GitHub, Inc." followed by links to Terms, Privacy, Security, Status, and Help.



# 내 remote 저장소로 local 저장소 만들기 #1

1. 이곳을 클릭합니다.

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. A red arrow points to the 'Clone or download' button. Below this, there are links for 'Pull request' and 'Compare'. The latest commit is shown as 'b0c9642' from 'an hour ago'. A dropdown menu is open from the 'Clone or download' button, showing 'Clone with HTTPS' as the selected option. The URL 'https://github.com/lastboy0/git\_lecture.git' is displayed with a copy icon. Below the URL are buttons for 'Open in Desktop' and 'Download ZIP'. A second red arrow points to the copy icon next to the URL.

0 releases 1 contributor

Create new file Upload files Find file Clone or download

Pull request Compare

Latest commit b0c9642 an hour ago

md file a

Create new file Upload files Find file Clone or download

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

`https://github.com/lastboy0/git_lecture.git`

Open in Desktop Download ZIP

2. 이곳을 클릭하면 URL이 복사됩니다.



## 내 remote 저장소로 local 저장소 만들기 #2

**\$git clone remote-repository-url**

예) git clone https://github.com/lastboy0/git\_lecture.git

**\$git clone remote-repository-url git\_study**

뒤에 폴더명을 넣어주면 해당 폴더명으로 생성되고, 생략하면 remote 저장소 이름으로 자동생성됨



remote 저장소 확인, 추가

**\$git remote**

**\$git remote -v**

**\$git remote add name remote-repository-url**

예) git remote add t-origin https://github.com/ryan-riot/git\_lecture



remote 저장소로 push 하기

**\$git push origin branch\_name**

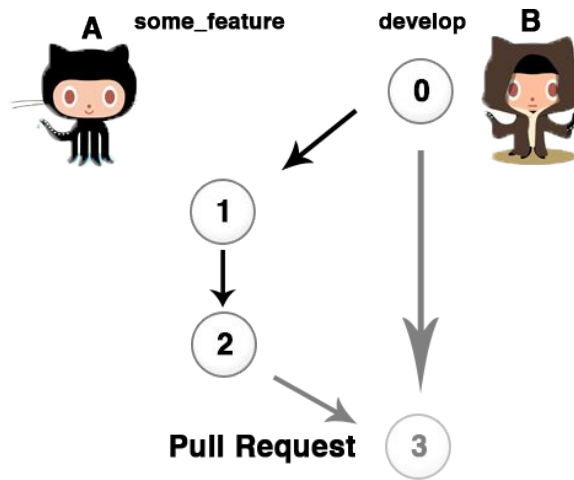
예) git push origin ryan\_branch





# PR(Pull Request) 보내기

- remote 저장소에 검토 요청 보내는 것을 PR이라 한다.
- 일반적으로 branch를 만들어 push한 후 github에서 요청작업을 진행한다



git and github lecture in CNU

Edit

Add topics

4 commits

2 branches

0 releases

1 contributor

Your recently pushed branches:

정종오 (less than a minute ago)

Compare & pull request

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is even with ryan-riot/master.

Pull request

Compare

unknown Update readme.md file

Latest commit b0c9642 2 hours ago

README.md

Update readme.md file

2 hours ago

README.md

## Repository for git\_lecture

This is sample repository for CNU students.

학생분들은 이곳의 repository를 clone해서 remote로 연결하세요.



## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: ryan-riot/git\_lecture base: master ... head fork: lastboy0/git\_lecture compare: update-girlgroup

✓ **Able to merge.** These branches can be automatically merged.



Update girlgroup

Write Preview

AA B i “ < > 🔗 ⋮ ≡ ≡ ↶ @ 📌

Leave a comment



Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

6 commits

2 files changed

0 commit comments

2 contributors

Commits on Mar 19, 2017

lastboy0

Add contents for my favorite girl group

1a



lastboy0 / git\_lecture  
forked from ryan-riot/git\_lecture

Unwatch 1 Star 0 Fork 1

Code Pull requests 1 Projects 0 Wiki Pulse Graphs Settings

## Add new girlgroup team #3

Edit

Open lastboy0 wants to merge 1 commit into master from update-girlgroup

Conversation 0 Commits 1 Files changed 1

+2 -0



lastboy0 commented just now

Owner + 🗑️ ✎️

No description provided.

🔗 Add new girlgroup team

0f25377

Add more commits by pushing to the **update-girlgroup** branch on **lastboy0/git\_lecture**.



✓ This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Write Preview

AA B i “ < > ↺ ⋮ ⋮ ⋮ ⏪ @ 📌

Leave a comment

Reviewers



No reviews—request one

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Notifications

Unsubscribe

You're receiving notifications because



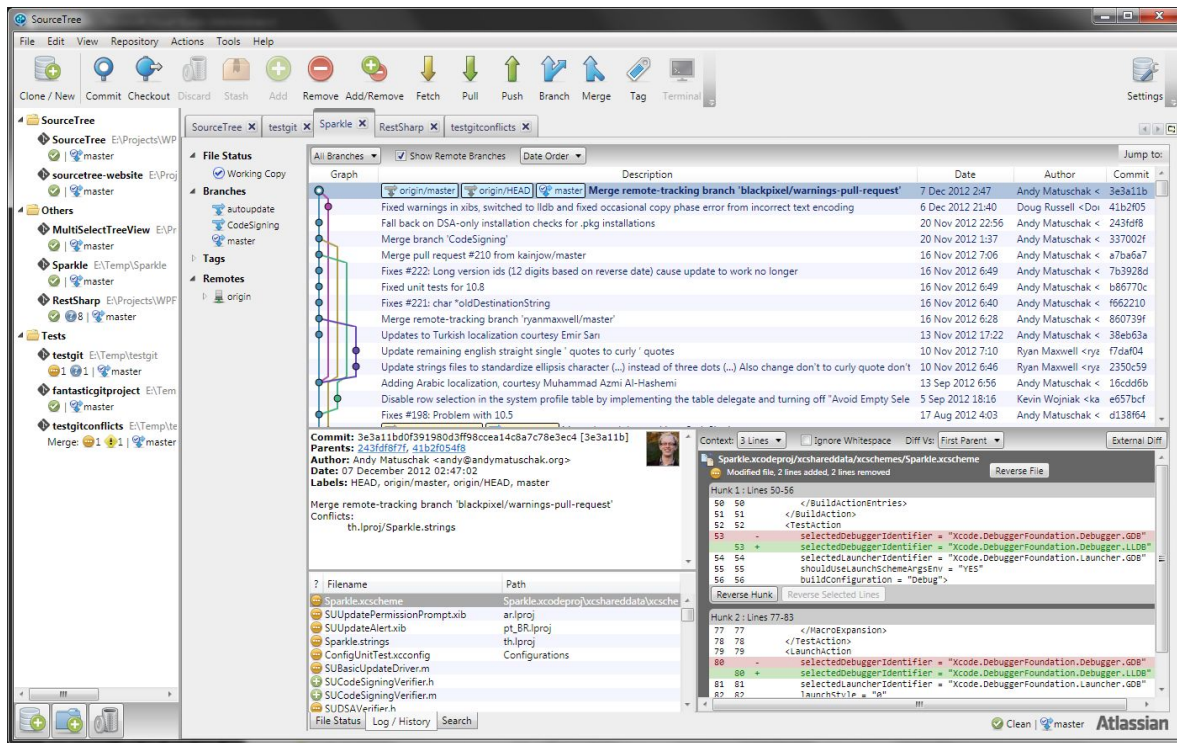
Fetch, Pull 로 remote 소스 가져오기

**\$git fetch origin**

**\$git pull origin**



# SourceTree 소개



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



# Git 학습을 위한 좋은 사이트들 소개

Branch 개념이해 학습 데모

<http://learnbranch.urigit.com/?demo>

누구나 쉽게 이해할 수 있는 git 입문

<http://backlogtool.com/git-guide/kr/>

Git 간편안내서

<http://rogerdudler.github.io/git-guide/index.ko.html>

Github 가이드 - 샘플 연습

<https://guides.github.com/activities/hello-world/>



# 과제

## Remote Repository를 통해 협업 진행하기

1. Team leader는 아래의 repository를 fork(clone)해서 팀용 public repository 생성  
[https://github.com/ryan-riot/cnu\\_homework](https://github.com/ryan-riot/cnu_homework)
2. 각 개인은 팀용 repository를 clone해서 본인 PC에 remote로 등록
3. 개인별 branch(본인이름)를 만들고 README.md 파일에 본인 이름을 추가
4. 팀 remote repository에 push후 PR 생성
5. Team leader는 충돌을 해결하고 하나의 파일로 병합
6. Team repository를 Trello에 등록

## id/pwd없이 ssh로 접속할 수 있도록 셋업하기

1. Github 인증샷 첨부





# 차주 강의 준비

## 1. Jenkins 설치해서 오기

<https://jenkins.io/> 에서 LTS 버전을 버전을 다운 후 설치하고 반드시 **실행**시켜볼 것.

## 2. JDK8 (JRE아님) 설치해오기

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

반드시 8 버전이어야 합니다.

## 3. Maven 설치하기

다운로드: <https://maven.apache.org/download.cgi>

설치가이드: <https://maven.apache.org/install.html>

## 4. Maven 설치 후 다음 git repository를 개인별로 fork & clone

<https://github.com/tony-riot/ci-example-1>

Console에서 해당 프로젝트 디렉토리로 이동 후

```
$ mvn clean package <enter>
```

빌드가 성공적으로 실행되는지 확인

