

# GitHub으로 학교 팀프로젝트 하기

25기 조성수 ( nexusz99 @ gmail.com ) 경희대학교



Hacker

~~팀플로 학점을 날로먹은 사람이 공유해주는~~

학교 팀 프로젝트를 그 누구보다 **멋있게** 하는 방법

학교 팀 프로젝트를 **성공적**으로 하는 방법

~~팀플로 학점을 날로먹은 사람이 공유해주는~~

학교 팀 프로젝트를 그 누구보다 **멋있게** 하는 방법

학교 팀 프로젝트를 **성공적**으로 하는 방법

을 Github으로 하는 방법

# 주 의 사 항

- 이 발표는 Github 을 사용하는 방법에 대한 가이드가 아닙니다.
- 그렇다고 Git 을 알려주는 발표도 아닙니다
- 그저, 팀 프로젝트에서 Github 을 어떻게 활용하는가?  
어떤 장점이 있는가에 대해 경험을 바탕으로 이야기합니다

# 소스 코드 버전 관리 시스템

개발자의 세계에는 없어서는 안되는  
두 개의 프로그램이 있습니다



# 소스 코드 버전 관리 시스템

이 두 프로그램을  
'소스코드 버전 관리 시스템'  
이라고 부릅니다



## 소스 코드 버전 관리

소스 코드의 변경 이력을 관리

소프트웨어 형상 관리의 일부분

개발 생산성 향상

# 소스 코드 버전 관리 시스템

이 발표에서는 현재 전 세계에서 가장 많이 쓰이는  
Git 을 선택하도록 하겠습니다





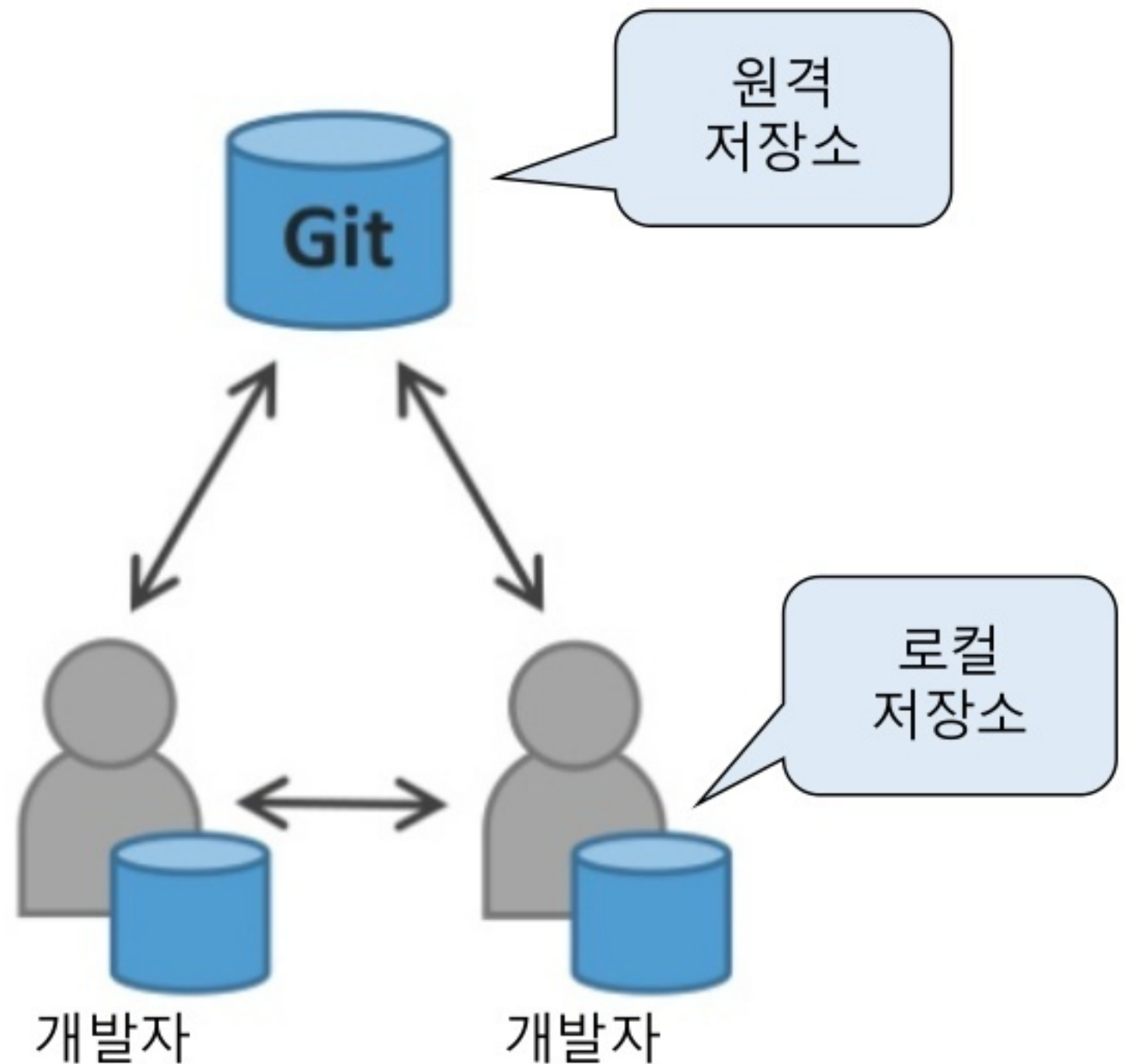
# 저장소 (Repository)

Git에는 2가지 저장소가 있습니다

**저장소** : 코드를 보관하는 곳

**원격 저장소** : 원격 서버에 존재

**로컬 저장소** : 개발자 PC에 존재



# Git 을 이용한 작업 흐름

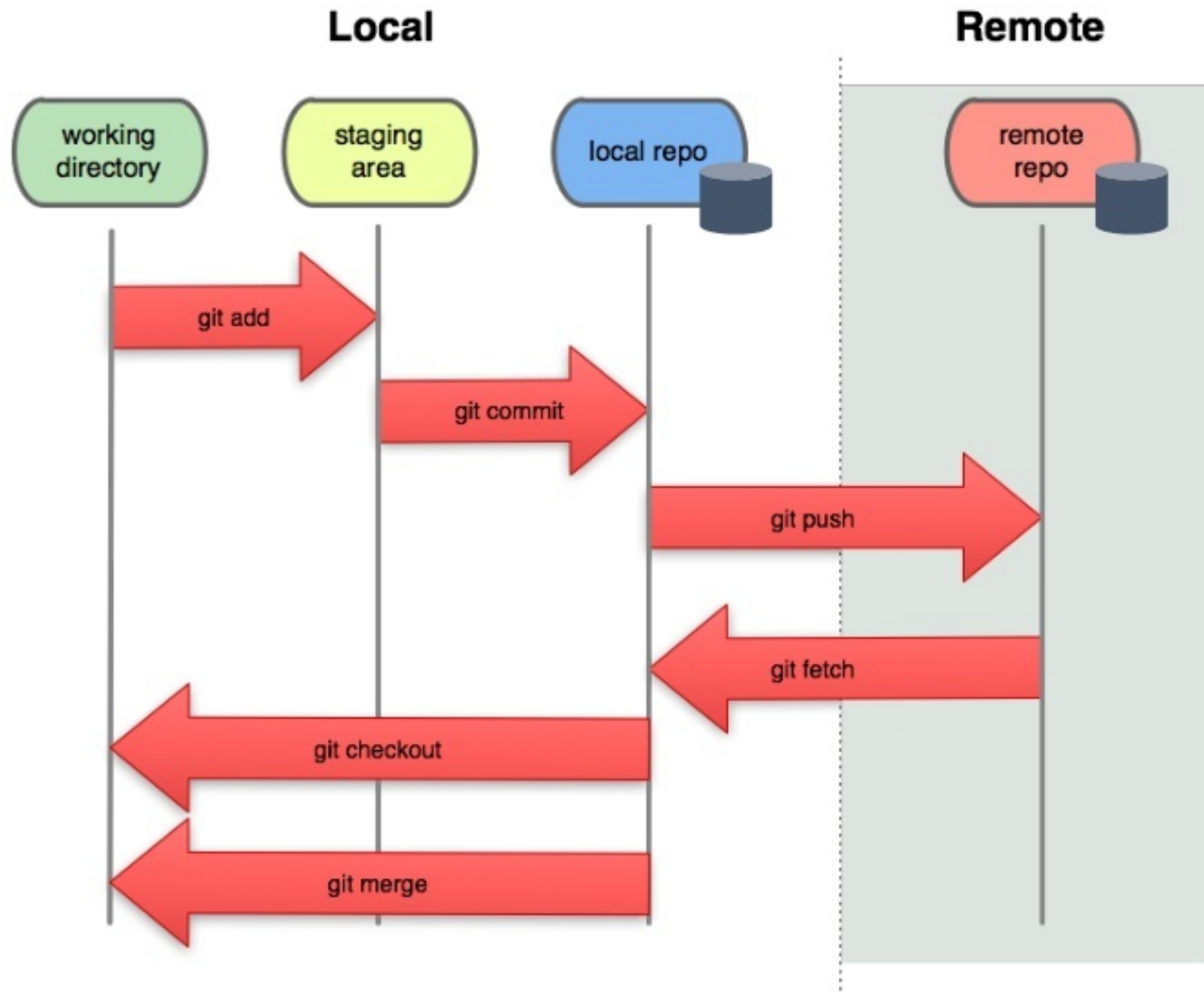


그림 출처 : <http://pismute.github.io/whygitisbetter/#everything-is-local>

# Git 을 이용한 작업 흐름

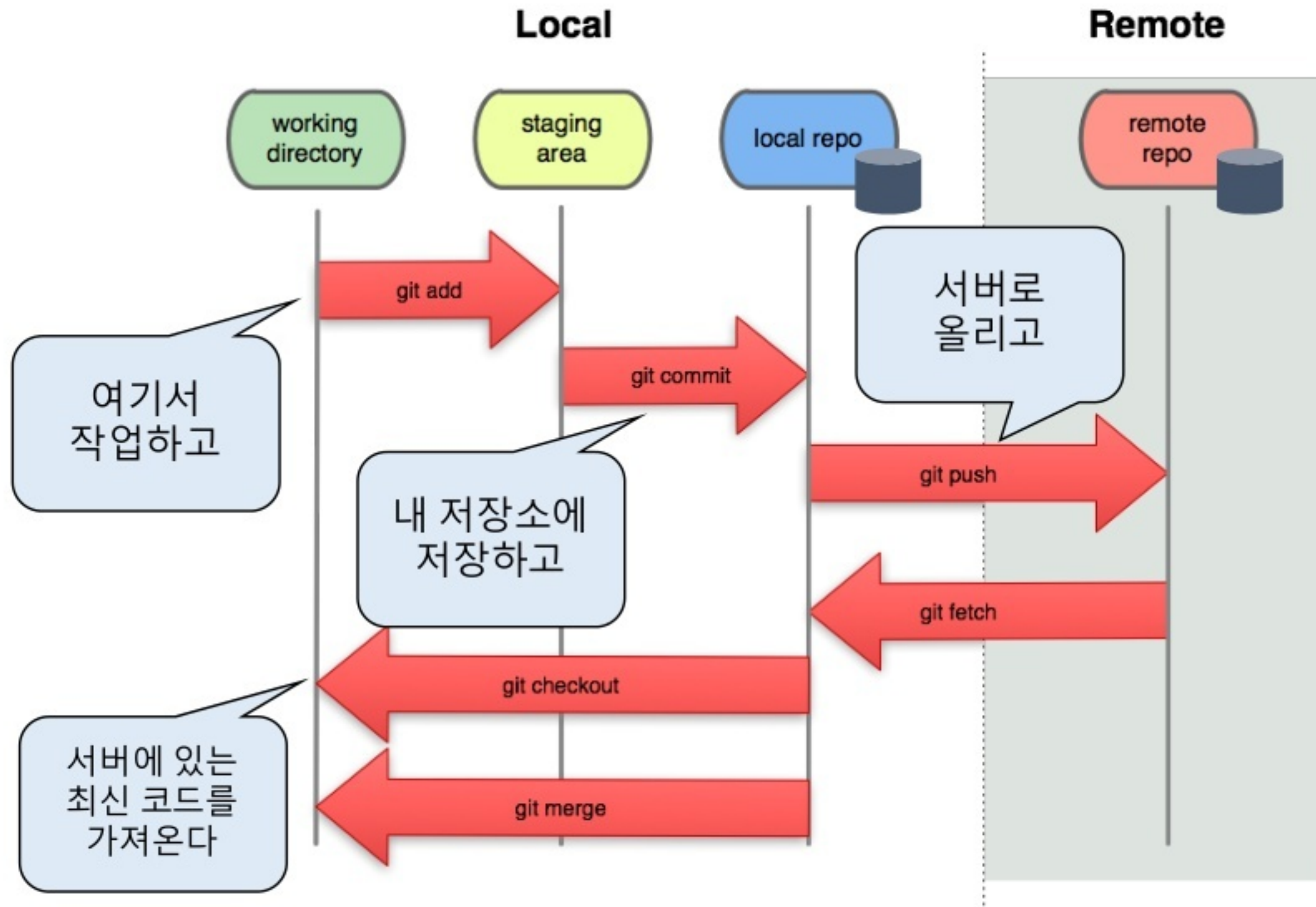


그림 출처 : <http://pismute.github.io/whygitisbetter/#everything-is-local>

Git 에 대해 더 자세한 내용은 아래 강의를 통해  
학습하면 좋습니다

[https://backlogtool.com/git-guide/kr/intro/intro1\\_1.html](https://backlogtool.com/git-guide/kr/intro/intro1_1.html)

## 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.







웹을 이용하여 Git 의 원격 저장소를 손쉽게 관리해주는 서비스

위키, 이슈관리, 머지 요청 관리, 팀원 관리 등  
프로젝트 진행에 필요한 대부분의 기능을 제공한다.

# Github 가입하기



Personal Open source Business Explore

Pricing Blog Support

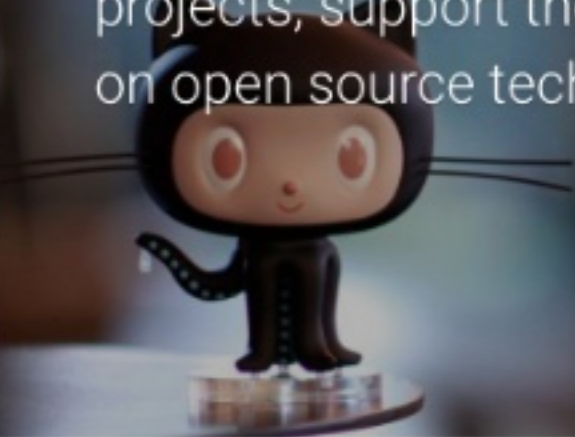
Search GitHub

Sign in

Sign up

## How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.



여기를  
통해 가입

Pick a username

Your email address

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.

Introducing unlimited private repositories

All of our paid plans on GitHub.com now include unlimited private repositories. [Sign up](#) to get started or [read more about this change on our blog](#).

Hacker

# Github 은 무료인가요?

## Priced for everyone

GitHub is free to use for public and open source projects.

Work together across unlimited private repositories with a paid plan.

Join GitHub for free

### Personal

\$7 / month

Build your own projects on GitHub.com and invite collaborators to join you in unlimited private repositories.

📦 Free for students as part of the Student Developer Pack.

Upgrade your account

### Organization

\$9 per user / month

Work with your team on GitHub.com in unlimited private repositories. Manage team and user level permissions.

👥 Starting at \$25 / month for your first 5 users.

Create an organization

### Enterprise

\$21 per user / month

Host your team's code on your own servers or in a private cloud with your existing security controls.

📦 Sold in packs of 10 users and billed annually.

Start a free Enterprise trial

# Github 의 Repository 종류

- Public Repository
  - 내가 작성한 소스코드가 다른 사람에게 공개되는 저장소
  - 오픈소스 프로젝트는 Public Repository 에 저장
- Private Repository
  - 소스코드가 다른 사람에게 공개되면 안될 경우 사용
  - 나 혹은 관련된 사람들에게만 보이는 저장소
  - Private Repository 를 쓰고 싶다면, 결제를 하면 됨



# Private Repository 를 무료로?

학교 프로젝트인데.. 내 코드를 공개하기 부끄럽다!

감추고 싶다!!!

하지만 돈은 .. 부담스럽다

학생이 이럴 때 좋습니다

## Student Developer Pack

The best developer tools, free for students



### Learn to ship software

There's no substitute for hands-on experience. The GitHub Student Developer Pack with some of our partner tools can be cost prohibitive for students. We've put together the GitHub Student Developer Pack with some of our partner tools in one place so they can learn by doing.

학생에게는 무제한 private  
저장소 제공 및 기타 다양한  
기능을 무료로 제공

**Get your pack**


THE TOOLS

# Github Repository

## Create a new repository

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

Owner

 nexusz99 ▾

Repository name

/

프로젝트 명

Great repository names are short and memorable. Need inspiration? How about [fantastic-octo-chainsaw](#).

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

공개 여부

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



라이선스  
설정


Create repository

# Github Repository

## Create a new repository

A repository contains all the files for your project.

Owner

 nexusz99 ▾

Repository name

Great repository names are short and unique.

Description (optional)

프로젝트에서  
사용할 언어로  
선택

☐  **Private**  
You choose who can see and who can push to your repository.

☐ **Initialize this repository with a README**  
This will let you immediately push new files to your repository.

Add .gitignore: **None** ▾

Create repository

Add .gitignore: **None** ▾

Add a license: **Apache License 2.0**

.gitignore

Filter ignores...

Android

AppEngine

AppceleratorTitanium

ArchLinuxPackages

**Autotools**

C

C++

CFWheels

CMake


CUDA




Hacker



# Github Repository





저장소를 생성하면, 다음과 같은 화면이 나옵니다

 nexusz99 / d2love Private


 Unwatch 1  Star 0  Fork 0


[Code](#) [Issues 0](#) [Pull requests 0](#) [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)


No description or website provided. — Edit

 1 commit  1 branch  0 releases  1 contributor

[Branch: master ▾](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▾](#)

 nexusz99 Initial commit Latest commit fb9b42b just now

 [.gitignore](#) Initial commit just now

 [LICENSE](#) Initial commit just now

Add a README with an overview of your project. [Add a README](#)

# Github Repository

저장소를 생성하면, 다음과 같은 화면이 나옵니다

nexusz99 / d2love Private

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

1 commit 1 branch 0 releases 1 contributor

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

nexusz99 Initial commit Latest commit fb9b42b just now

.gitignore	Initial commit	just now
LICENSE	Initial commit	just now

Add a README Add a README

다양한  
기능 메뉴


저장소에 있는  
파일 목록

Hacker

# 팀원 추가하기

Settings – Collaborators 에서 함께하는 팀원을 추가할 수 있습니다  
당연히, 팀원도 Github 에 가입이 되어있어야 합니다

 nexusz99 / deship

 Unwatch ▾

3

★ Star

<> Code

! Issues 0

🔑 Pull requests 0

📖 Wiki

📶 Pulse

📊 Graphs

⚙ Settings

Options

**Collaborators**

Branches

Webhooks & services

Deploy keys

Collaborators

Push access



JeongHoiun



Myoji

Search by username, full name or email address

 Hacker

# 프로젝트 시작하기

- Repository 를 각자의 컴퓨터에 Clone 하고, 개발 진행
- 팀장이 프로젝트의 윤곽을 먼저 개발한 뒤, 팀원들에게 작업을 분배
- 사전에 규칙을 정한다.
  - commit 전, 반드시 pull 을 받아서 코드를 최신으로 유지한다.
  - 최대한 1개의 파일을 2명 이상이 작업하지 않도록 역할을 나눈다.
  - IDE 가 생성하는 설정파일은 **절대로** 올리지 않는다
    - Visual Studio : Debug, Release, \*.sdf
    - PyCharm, IntelliJ : .idea
    - Eclipse : .classpath, .project



# 팀장의 작업 - 프로젝트 윤곽 개발

- 모든 팀원들이 동일한 코드 규칙을 유지하기 위해 필요
- 아래 내용을 팀장이 팀원과 상의하여 먼저 개발을 진행
  - 디렉토리 구조
  - 공통 모듈(common.cpp, util.py 와 같은 .. )
  - 기타 개발에 필요한 스크립트
- 윤곽이 나올 때 까지 팀원은 각자가 맡은 기능에 대해 연습용 코드를 만든다

# 보고서를 대비하자

- 학교 팀 프로젝트의 마지막은 최종보고서
- 보통 대부분의 시간은 개발에 투자하고, 보고서는 제출 1주일전부터 작성
- 개발이 다 끝난 다음 보고서를 쓰려면, 빠뜨리는 부분이 많이 발생하고, 귀찮다.
- 개발을 하면서 구현에 대한 문서를 동시에 작성한다

# 구현/설계 문서는 위키로

- Wiki 에다가 구현/설계 문서를 정리한다.

[Code](#) [Issues 0](#) [Pull requests 0](#) **Wiki** [Pulse](#) [Graphs](#) [Settings](#)

## [ Home Gateway] REST API Specification

SeongSoo Cho edited this page on Jun 11 · 10 revisions

[Edit](#) [New Page](#)

### 회원가입

/users/signup - POST

```
{'ID': 'test', 'user_name': '김개똥', "passwd": "1234", "address": "경기도 용인시 기흥구 뿌잉뿌잉", "phone": "010-1234-1234"}
```

### 로그인

/users/signin - POST

```
{'ID': "test", "passwd": "1234"}
```

### 기기 등록

/device - POST

```
{'SN': '12345'}
```

[기기 목록 보기](#)

**Pages 9**

- [Home](#)
- [\[ City Area \] DB Schema](#)
- [\[ City Area \] Distribution Manager REST API Specification](#)
- [\[ Cooperation \] consumable API](#)
- [\[ Cooperation \] transaction API](#)
- [\[ Home Gateway\] REST API Specification](#)
- [\[ Service Provider \] DB Schema](#)
- [\[ Service Provider \] REST API Specification](#)
- [Smart home service distributed database concept](#)

# 구현/설계 문서는 위키로

- 위키로 정리하면 어떤 점이 좋나요?
- 팀원들끼리 설계에 대한 내용을 빠르고 쉽게 공유할 수 있습니다.
- 내용이 바뀔경우 빠르게 문서로 반영할 수 있습니다.
- 문서 형식에 신경쓰지 않고 내용에 집중할 수 있습니다.

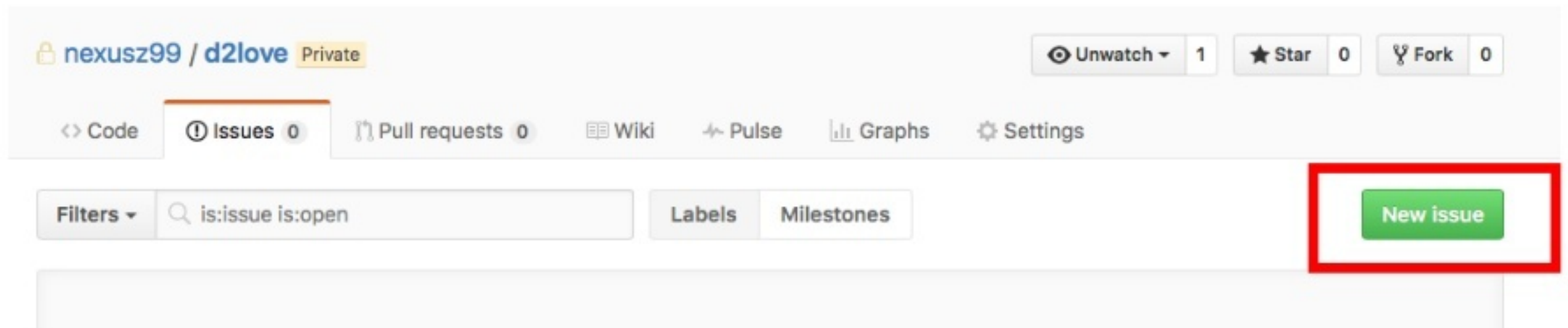
# 팀원에게 일을 주자

- 각자 맡은 기능을 개발하지만, 그 안에서 세세한 작업들이 나뉘어집니다.
  - 예) 로그인 기능 개발하기
    - 사용자 테이블 설계
    - DAO 구현
    - API 구현
- 세세한 작업들을 누가 언제까지 할 것인지 '구두'가 아닌 일감으로 만들어서 할당해줍니다
  - 나중에 가서 "까먹었어요", "들은 적이 없는데요?" 를 방지



# 팀원에게 일을 주자

- Github 에는 프로젝트의 작업(일감)을 관리하는 Issues 기능이 있습니다
- Issue 에서 새로운 일감을 생성합니다
- 일감은 본인이 생성하거나, 팀장이 할당해줍니다.



# 팀원에게 일을 주자

The screenshot shows a GitHub repository interface with the 'Issues' tab selected. The repository name is '[로그인] 사용자 테이블 설계'. The issue title is '[로그인] 사용자 테이블 설계', annotated with '일감 제목' (Task Title). The issue content is '로그인 기능 개발을 위해 데이터베이스에 사용자 테이블을 설계한다.', annotated with '일감 내용' (Task Content). The issue is assigned to 'nexusz99', annotated with '담당자' (Assignee). The issue has no labels, milestones, or comments. The 'Submit new issue' button is visible at the bottom right.

<> Code **Issues 0** Pull requests 0 Wiki Pulse Graphs Settings

[로그인] 사용자 테이블 설계

Write Preview

로그인 기능 개발을 위해 데이터베이스에 사용자 테이블을 설계한다.

Attach files by dragging & dropping or [selecting them](#).

Styling with Markdown is supported

Submit new issue

Labels: None yet

Milestone: No milestone

Assignees: nexusz99

일감 제목

일감 내용

일감 기한

담당자

# 마일스톤 생성하기

- 마일스톤 : 프로젝트에서의 중요한 시점
  - 예: 릴리즈 일정, 연구 마감 시한 등
- 팀 프로젝트에서 마일스톤은 팀 내에서 자유롭게 지정
- 보통 2~3주 단위로, 마일스톤을 생성하고 그 안에 해야할 작업을 마일스톤에 넣는다.
- 모든 팀원은 최대한 마일스톤 내에 일감을 완료하려 노력한다



# 마일스톤 생성하기

[Code](#)[Issues 1](#)[Pull requests 0](#)[Wiki](#)[Pulse](#)[Graphs](#)[Settings](#)

## New milestone

Create a new milestone to help organize your issues and pull requests. Learn more about [milestones and issues](#).

Title

마일스톤 1

마일스톤 제목

마감 기한

Due Date (optional) [clear](#)

◀ August ▶ 2016 ▶

Mon Tue Wed Thu Fri Sat Sun

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Description

A 기능

B 기능

위 두가지 기능을 우선적으로 개발하여, 프로그램 윤곽을 잡아가는 단계

이 마일스톤에 대한 설명

# 소스코드 제출 시

- 저장소 페이지에서 Download ZIP 을 통해 깔끔하게 다운로드

nexusz99 / deship

Unwatch 3 Star 1 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

Decentralized Smart Home IoT Platform — Edit

8 commits 5 branches 0 releases 1 contributor

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

nexusz99 Add lib folder

config/telehash	Add base code
deployment	Add logging
deship	Add lib folder
deship	Add deship distributed database architecture

Clone with SSH ? Use HTTPS

Use an SSH key and passphrase from account.

git@github.com:nexusz99/deship.git

Open in Desktop Download ZIP

# 질문

만약 한 팀 프로젝트에 여러개의 프로그램을  
만들어야한다면 어떻게 해야하나요?

# 하나의 팀플안에 여러개의 프로젝트

- 팀 프로젝트에서 여러개의 프로그램을 만들어야한다면  
어떻게 저장소를 구성해야할까요
  - 예) 안드로이드 앱, 서버, 홈페이지
- 방법 1 : 여러개의 저장소를 만든다
- 방법 2 : Organization 을 이용
- 방법 3 : 하나의 저장소에 폴더로 구분





# Organization 만들기

- 일종의 그룹을 만들고, 그 안에 저장소 생성

## Create an organization

✓ Completed  
Set up a personal account

 **Step 2:**  
Set up the organization

 **Step 3:**  
Invite members

### Set up the organization

**Organization name**

Organization name input field

The organization will live at [https://github.com/](https://github.com/dnsforever)

**Billing email**

Billing email input field


Receipts will be sent here

**Plan**

☒ Unlimited members and public repositories

☐ Unlimited private repositories at \$25/month for your first 5 users. \$9/month for each additional user.

Create organization

 dnsforever

News Feed Pull Requests Issues View dnsforever

You're an owner of the dnsforever organization! (on May 18, 2014)  
[Create a repository for dnsforever](#)

Subscribe to the dnsforever organization news feed


Learn more


### Repositories 3


New repository

Find a repository...

All Forks Sources Private Public

 [dnsforever/dnsforever-web](#)

 [dnsforever/dnsforever-bind](#)

 [dnsforever/dnsforever-dns](#)

무료로 쓰려면, public 만.

그룹 명/저장소 명  
으로 저장소 생성

# 하나의 팀플안에 여러개의 프로젝트

- 원래는 프로그램 별 저장소를 만들어야한다.
- 하지만 팀 프로젝트에서는 여기저기 저장소를 돌아다니면서 상황을 파악하기엔 **귀찮다**
- 저는 주로, **학교 팀플에서는** 하나의 저장소에 폴더로 구분해서 관리합니다

팀 프로젝트를 하면서, 이런 말들이  
사라졌습니다.

“팀 내에서, OO아 너가 만든거 코드좀 보내줘”

“어떤 파일이 그때 그 기능 만들었다는 거였더라?”

- 갑자기 잘 되던 기능이 안된다면 누가 어느 부분에서 잘못했는지 쉽게 추적이 가능했다.
  - 너가 범인이야 임마!
- 누군가가 버그를 못잡겠다고 한다면, 그 사람 컴퓨터로 직접 가지 않아도 된다.
  - 어? 그래? 그럼 너가 작업한거 branch 하나 만들어서 거기에 올려줘봐 작업해줄게
- 온라인으로도 충분히 팀 프로젝트 가능



# 부가적인 이득 – 개이득

- 요즘 취업할 때 github id를 기업이 요구한다.
  - 이 사람이 어떤 프로젝트를 진행했는지, 오픈소스 활동은 했는지 확인용 (?)
- 개발자들 사이에서도 github 계정이 새로운 명함
  - 이 사람이 만든 repository 는 별이 많아!
  - 이 사람은 유용한 라이브러리를 많이 만들어줘
- 포트폴리오로도 작용

# 결론

- 좋다
- 너무 좋다
- 편하다
- 체계적으로 하는 기분이다
- Github 을 쓰자.

