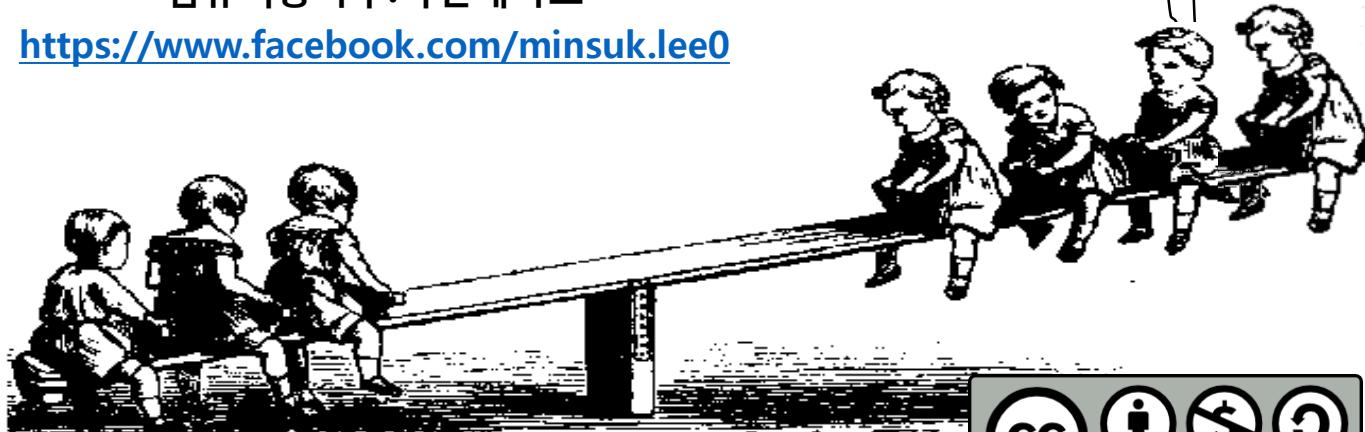


Open Source Software and **GITHUB**

Minsuk Lee

컴퓨터공학부.국민대학교

<https://www.facebook.com/minsuk.lee0>



<http://ko.wikipedia.org/wiki/%EB%8B%A8%EC%9E%91%ED%85%8C>



목차

- OSS 소개
- OSS License
- OSS Projects
- OSS 참여
- ALM 소개
- GITHUB
- GIT
- GITHUB 활용

최종본 슬라이드와 모든 링크가
정리되어 있는 곳

<http://hl1itj.tistory.com/118>

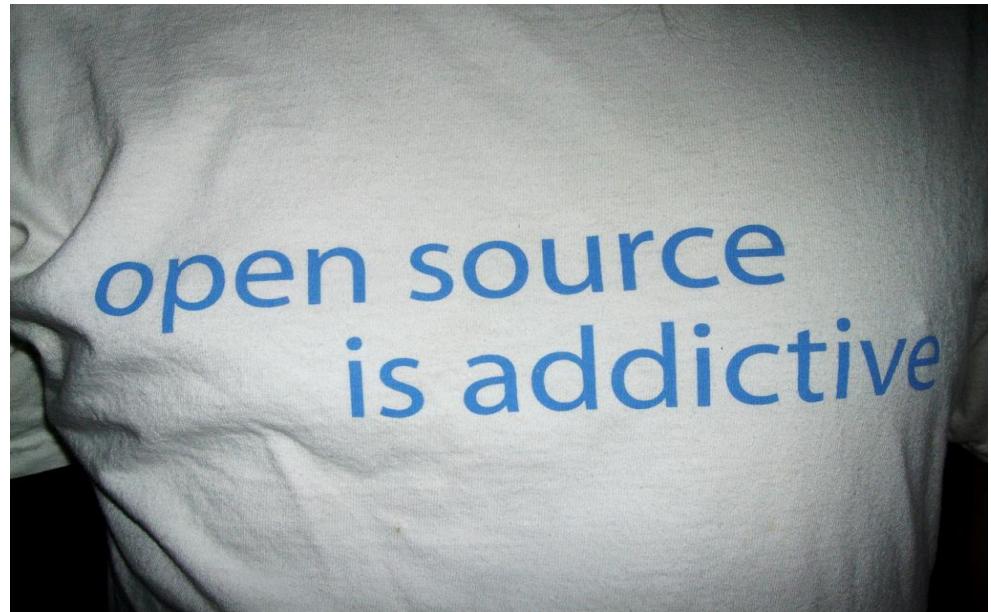


<http://opensource.com>

Open Source:

A proven business paradigm

to accommodate various requirements most efficiently.



<https://www.flickr.com/photos/mager/2299296256/>

Open Source:

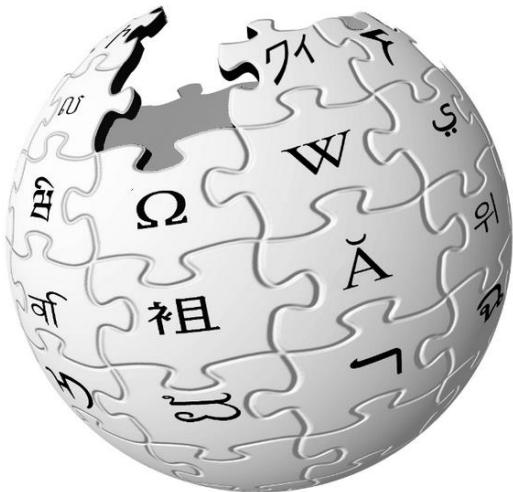
A cultural phenomenon that
evolves by participating, contributing and sharing.

Good Practice – reader community

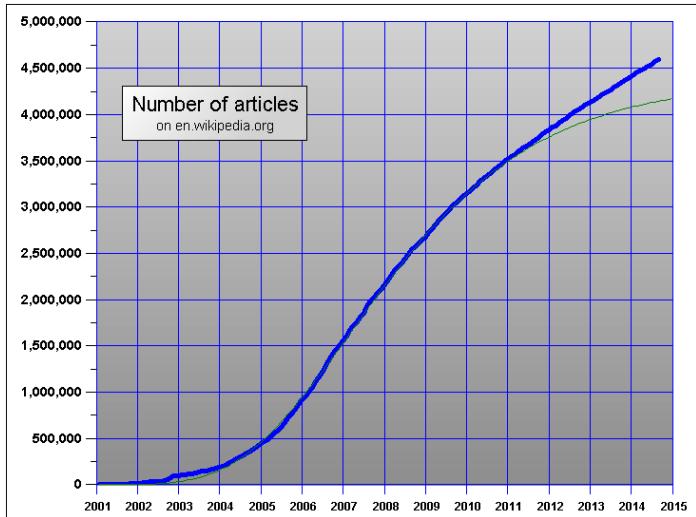


<http://tcdesigns84.deviantart.com/art/Distressed-Harley-Davidson-50734585>

Best Practice – writer community



WIKIPEDIA



<http://en.wikipedia.org/wiki/Wikipedia>

OSS의 정의

- 정의1: 자유롭게 열람, 사용, 수정하고,
다른 사람에게 배포할 수 있는 소프트웨어
 - ✓ 강한 규칙: 라이선스(저작권, 특허권)
 - <http://opensource.org/> - GNU, BSD, Apache, MIT 등 70+ 종
- 정의2: Community에서 Open Project로 집단 협업으로 만든 소프트웨어
 - ✓ 강한 절차(process)
 - 동료 검토 (Peer Review)
 - Maintainer / Committer 중심
 - 효율적 의사결정
 - 문서화

Open Source Software, now

- 78%의 회사가 Open Source 기반으로 운영됨
 - 3%의 이하만 Open Source를 사용하지 않음
- 소프트웨어가 필요할 때,
66% 이상의 회사가 Open Source가 있는지 먼저 확인
- 자사 엔지니어의 반 이상이 Open Source 관련된 일을 하고 있는 회사가 50% 이상
- 39%의 회사가 Open Source 프로젝트를 계획 중
- 47%의 회사가 자사의 툴과 프로젝트를 Open Source로 공개
- 55%가 Open Source가 보안상 확실히 안전하다고 응답

(참고, 2015 Research by North Bridge, Black Duck)

<http://www.slideshare.net/blackducksoftware/2015-future-of-open-source-survey-results>



ubuntu®

yobi



mozilla
Firefox®

ANDROID



Tadpole



TIZEN™



eGovframe 표준프레임워크



write less, do more.



왜 오픈소스 인가?

- No Vendor Lock-in
- OSS 자체적인 Business 모델로서의 가치
 - 이미 성공적인 큰 OSS 프로젝트, 커뮤니티에만 해당
- 핵심 역량을 발휘할 수 있는 부분에 집중
 - 다른 참여자/커뮤니티와 함께 역할 분담, 동반 성장
 - 세상의 모든 OSS는 내 친구, 내 재산
 - 나의 개발 역량 향상을 위한 발판
- 나 스스로 또는 외부에 대한 긍정적인 메시지
 - 좋은 사람들과의 협력 가능성
 - 내 역량을 세계 만방에 알리는 방법
 - 새로운 도전과 사업의 기회
- 인력 양성/ 리크루팅의 기회
 - 선수들이 인정한 선수들이 커뮤니티에 있음
 - 취업 전에 역량을 인정받을 거의 유일한 방법



오픈 소스에 대한 중요한 오해

1. 소스가 공개되어 있기 때문에 보안에 취약하다.
2. 오픈소스를 쓰면 회사의 기술 자산이 외부에 노출된다.
3. 오픈소스를 쓰고 수정한 부분은 공개하지 않는 것이 좋다.



이런 조직이 오픈 소스 소프트웨어 조직

1. 뭔가가 땅길 때, 오픈 소스를 먼저 생각합니다.
2. 개발자들이 커뮤니티에 나가 우리 기술과 로드맵을 설명합니다.
3. 우리 개발자들을 기다리는 커뮤니티가 있습니다.
4. 초보 커뮤니티 개발자들의 참여를 위해 노력합니다.
5. 커뮤니티 개발자들의 기여를 언제나 존중합니다.
6. 오픈소스 소프트웨어 저작권을 존중합니다.
7. 진짜 오픈을 합니다.



OSS 내 맘대로 분류

- Community OSS

- 커뮤니티에서 만들기 시작한 프로젝트
 - 대부분 OSS 프로젝트
 - 나중에 회사로 넘어가는 경우가 많음 (MySQL 등)
- Linux(www.linuxfoundation.org), 배포판 (Ubuntu, ...)
- Apache(www.apache.org), KDE(www.kde.org)

- Commercial OSS

- 특정 회사가 만든 (상용) SW를 전략적 목적에 의해 공개
 - Commercial version과 community version을 분리하기도 함
 - Commercial version : Long Term Stable Version (subscription based support)
 - 최근에는 나중에 공개하는 것이 아니라 처음부터 Community 방식으로 개발하는 것이 보통
- Solaris(SUN), Android(Google), Fedora(Red Hat), OpenSuse,
- Cubrid(Naver), ...

- 애매한 OSS (늘 그렇듯이)

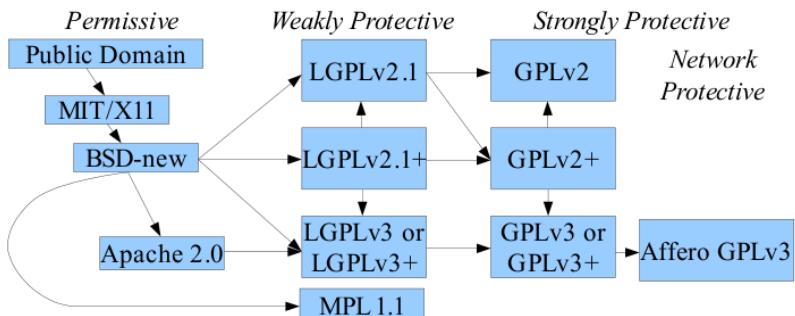
- 국가 프로젝트로 개발하고 잊혀진, 또는 Capstone 프로젝트

But...

License is not easy

OSS License

- 복잡한 법적인 이슈는 변호사에게..
- OSS 라이선스는 기본적으로
 - ✓ 소프트웨어의 사용을 장려하기 위한 문서
 - ✓ 참여자들의 (경제적 또는 비 경제적) 이익이 최대가 되는 합의
 - 코드 자산, 특히, 상표권
 - ✓ 공유와 헌신에 대한 예의
 - Reputation, 원본 소스의 존중
 - ✓ 저작자의 오픈 소스에 관한 철학의 존중
 - 파생물의 공개 조건

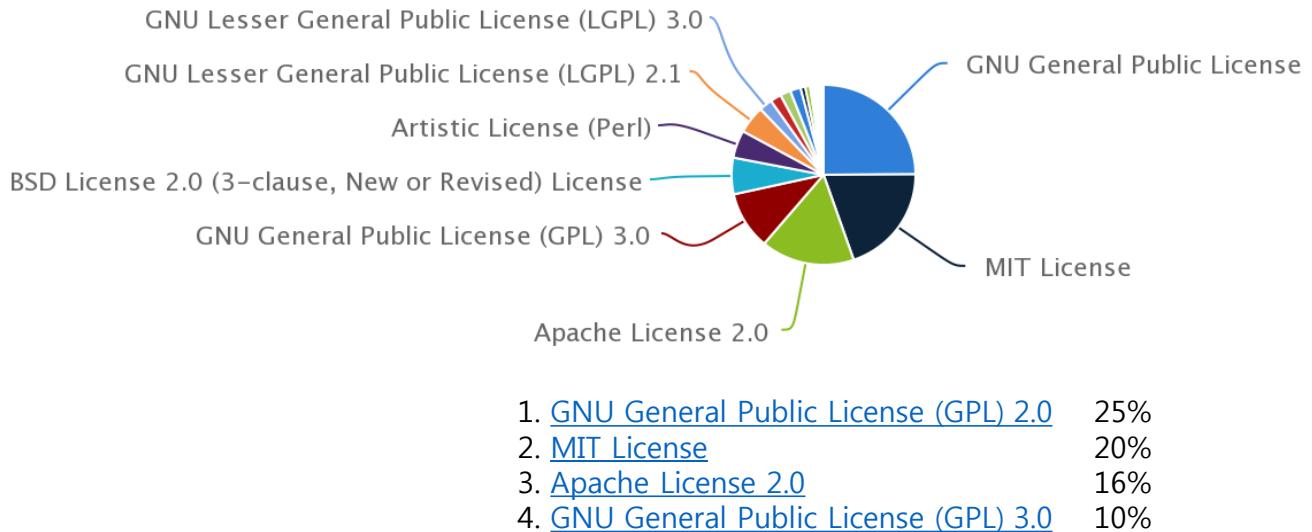


유명한 (aka. 큰 커뮤니티의) 라이선스

- Apache License, 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License
- GNU Library or "Lesser" General Public License
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

<http://opensource.org/licenses/category>

Open Source 라이선스 통계



소스 사용에 관한
자유를 더 부여하여
사용을 장려하고

Source에 대한 Control을 강화
(공개된 소스의 Integrity, Competency 유지)

<https://www.blackducksoftware.com/resources/data/top-20-open-source-licenses> (2015)

주요 Open Source 라이선스

	GPL	LGPL	MIT (BSD, MPL)	Apache
저작권 보호	O	O	O	O
상용 SW 혼용 가능	O	O	O	O
파생물 공개 의무	O	O	X	X
같은 라이선스 유지	O	O	X	X
특허권행사 가능	X	X	X	O
비공개SW와 혼용	X	O	O	O



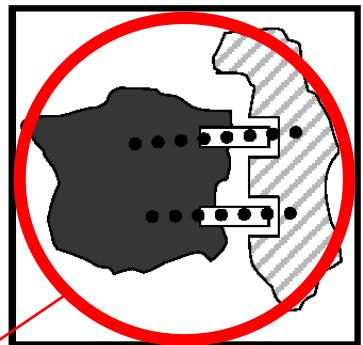
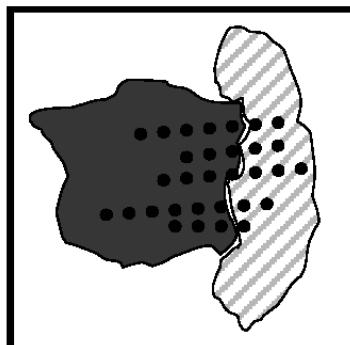
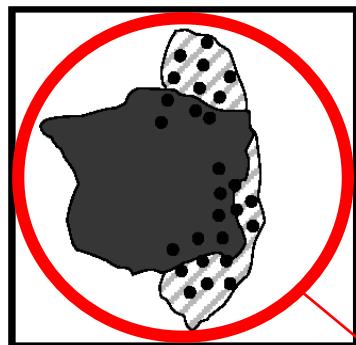
“파생물 (Derivative Works)”의 정의

- 파생물의 사전적 정의

“하나 이상의 기존 저작물을, 번역, 편곡, 희곡화, 소설화, 영화화, 녹음, 요약, 또는 기타 방법으로 제작, 개작, 적용한 저작물”

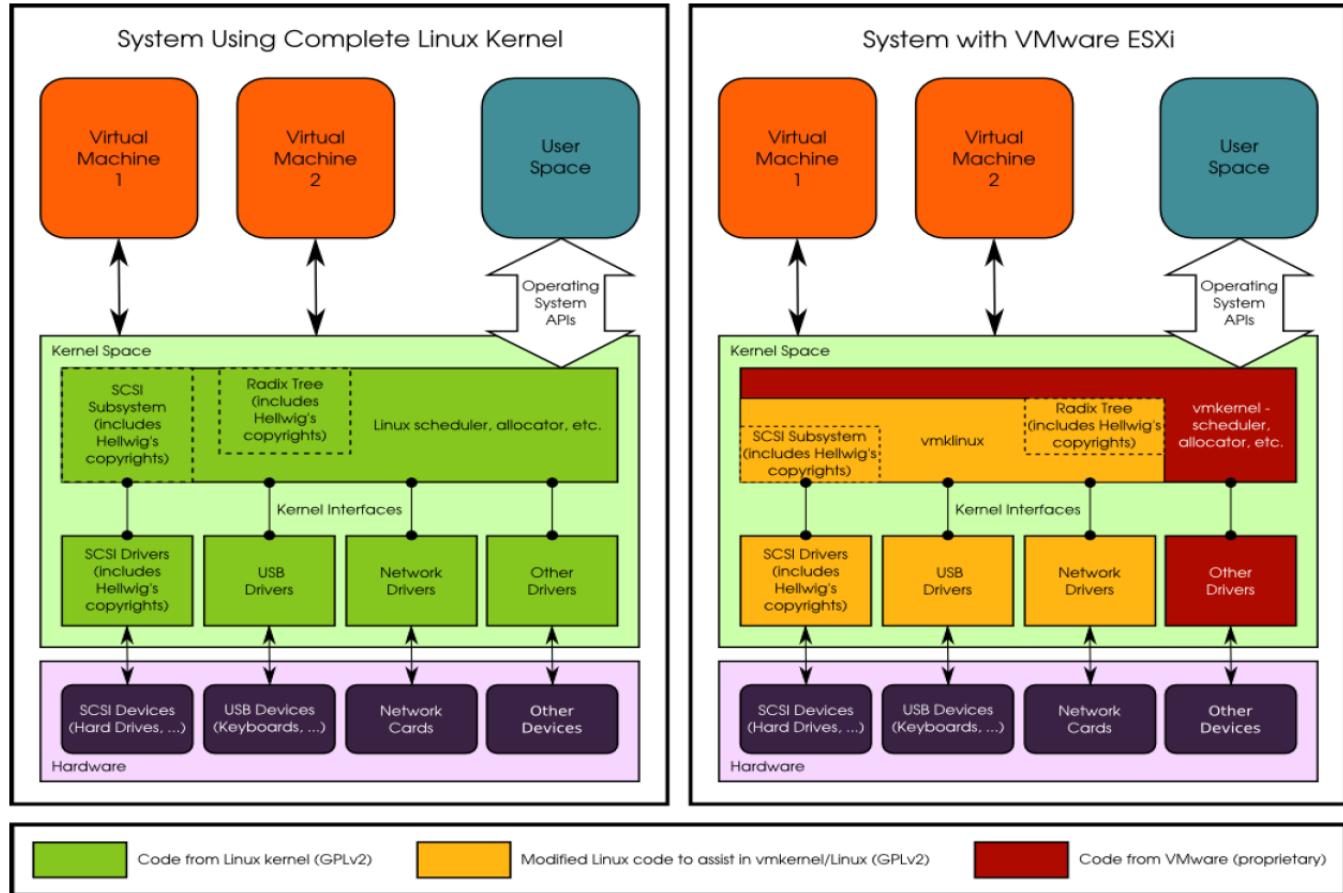
- Open Source Software 관점에서의 파생물이 아닌 경우:

이미 존재하는 API 만을 이용하여 **분리 빌드 / 실행되는 Module**



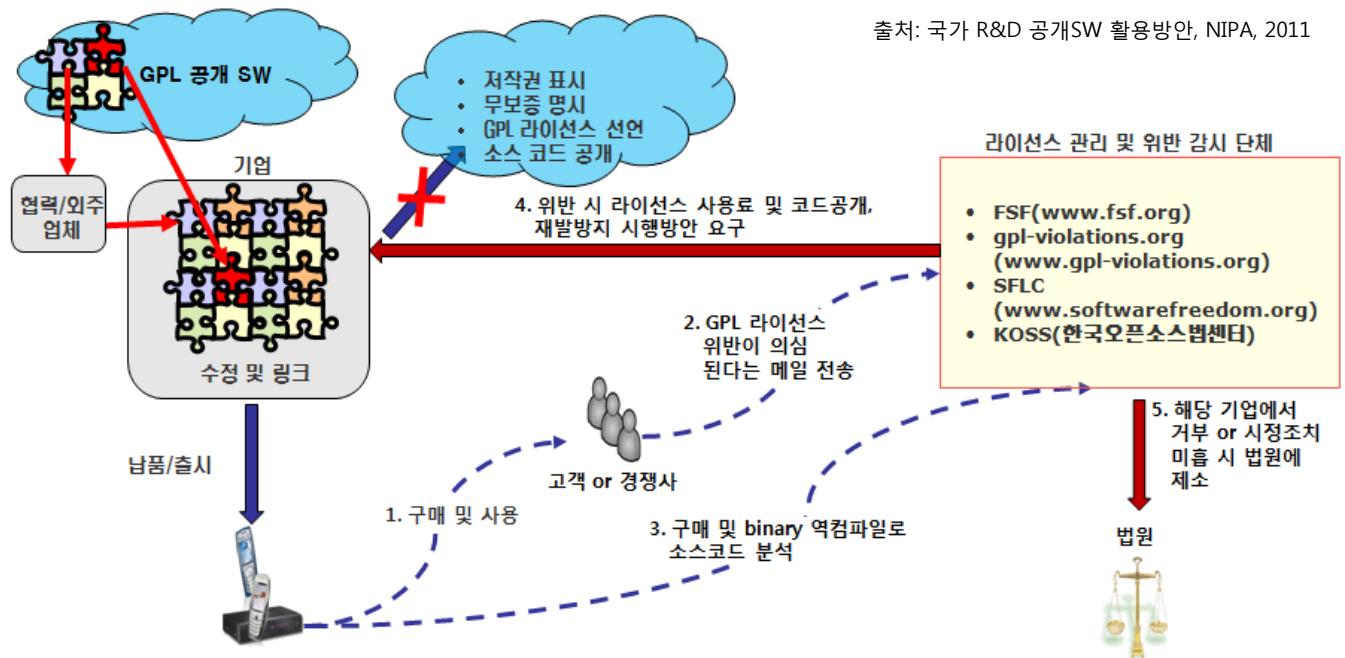
파생물!!
(소스 공개)

Hellwig vs. VMware case



(ref, Till Jaeger, BlackDuck Korea Conference 2015)

라이선스 위반 확인 모니터링 체계

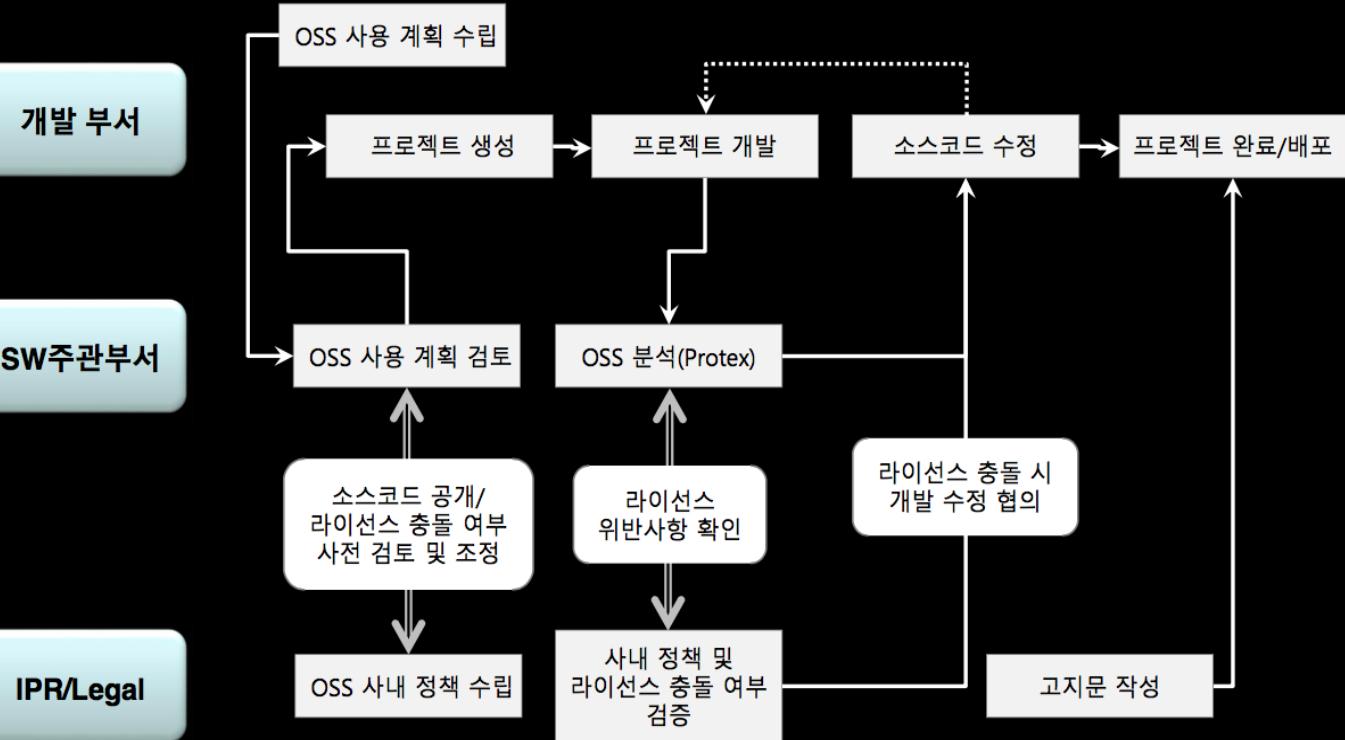


실제 업체간 소송은, 저작권+상표권+특허권 등이 조합된 종합선물세트 형태로 제기됨.

소스없이 위반을 어떻게 발견할 수 있는가? : <http://www.binaryanalysis.org>

<http://www.fossology.org/projects/fossology>

OSS Governance



OSS 어디에 있을까?

github.com
code.google.com
bitbucket.org
sourceforge.net
savannah.gnu.org
dev.naver.com
www.codeplex.com
...

2016.1.25 일 종료
(github export 가능)

FSF

Microsoft

List of open source code repositories

http://en.wikipedia.org/wiki/Comparison_of_open-source_software_hosting_facilities

freeopensourcesoftware.org/index.php?title=Main_Page Google

[page](#) [discussion](#) [view source](#) [history](#)

[Log in / create account](#)

FOSS

Main Page

"Because open source software features open code, more programmers are able to view the code, create new functionality, and fix bugs. This follows the same natural way that science has developed over time."

– Taoism of Open Source ↗; Chen Nan Yang; September 29, 2007.

This is a public Wiki about Free Open Source Software (FOSS), sometimes also called Free/Libre Open Source Software, Free Software, or just Open Source.

All FOSS licenses require that the software remain available to use, modify, and distribute at no cost. Most FOSS licenses also include kind of a lawyer's version of the [Golden Rule](#), requiring that all software modifications, such as bug fixes and enhancements, must also be made available under the same license - with permission to use, modify, and distribute at no cost. This naturally creates living software, continually growing the value for all. These licenses are good at generating the trust that leads to large and sustainable [communities](#). The great majority of FOSS is released under licenses including this give-back condition.

FOSS is increasingly the go-to standard for operating systems to user applications, for individuals to large enterprises. It helps reduce costs, avoid lock-in, increase productivity, enhance security, and improve standards compliance. With the best long-term investment protection, FOSS is the lowest risk choice for software systems today. More information can be found in the following sections.

Information	Software
<ul style="list-style-type: none">■ History■ Organizations■ Licenses■ People■ Mailing Lists■ News	<ul style="list-style-type: none">■ Primary Sources■ Secondary Sources■ Directories■ System Software – more than 100 resources■ Application Software – more than 200 resources■ Cloud Computing – more than 30 resources

 **GNU FDL**
FREE DOC
LICENSE

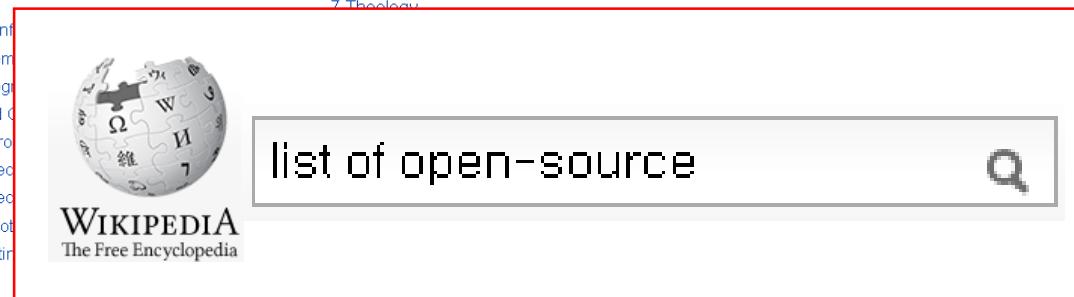
Wikipedia: List of Open Source Software

http://en.wikipedia.org/wiki/List_of_free_and_open_source_software_packages

- 모든 오픈소스 SW가 있는지는 않지만,
응용 분야별로 대표적인 공개 SW가 나열

1 Applied fields

- 1.1 Artificial intelligence
- 1.2 CAD
 - 1.2.1 Electronic design automation (EDA)
- 1.3 Computer simulation
- 1.4 Finance
- 1.5 Integrated Library System
- 1.6 Mathematics
- 1.7 Reference management software
- 1.8 Science
 - 1.8.1 Bioinf.
 - 1.8.2 Chem
 - 1.8.3 Geog.
 - 1.8.4 Grid C
 - 1.8.5 Micro
 - 1.8.6 Molec
 - 1.8.7 Molec
 - 1.8.8 Nanot
 - 1.8.9 Plottin
- 1.9 Statistics



2 Assistive technology

- 2.1 Speech (synthesis and recognition)
- 2.2 Other assistive technology

11.1 Content management systems

11.2 Wiki software

12 Healthcare software

4.15 Webgrabber

4.16 Web-related

4.17 Other networking programs

<https://www.openhub.net>

https://www.openhub.net

Join Now

Claim your contributions

Manage your project's data

Highlight your use of FOSS

Join Now

What's New

Ohloh has changed
It's now Open HUB

OH

Most Popular Projects

	Mozilla Firefox	13090 users
	Apache HTTP Server	9345 users
	MySQL	9073 users
	Apache Subversion	8698 users
	PHP	7674 users
	Linux Kernel	6982 users

Most Active Projects

	MediaWiki	22422 commits
	Google V8 JavaScript Engine	21984 commits
	Simulation of Urban MObility	8774 commits
	Systems Biology Markup Language (SBML)	8382 commits
	DPorts	6153 commits
	Mozilla Firefox	

Most Active Contributors

	Translation updater bot	1379 commits
	Scaramanga	1366 commits
	Andrea Scarpino	1010 commits
	Peter Simons	889 commits
	patacongo	752 commits
	xaizek	736 commits

openhub.net의 검색 결과

BLACKDUCK | Open HUB

PROJECTS

예, Virtual Machine 검색

Projects

Showing page 1 of 118

Search / Filter on: virtual machine

Follow @ OH SIGN IN JOIN NOW Projects virtual machine

Browse Projects By Tags Add a New Project

Sort by: Relevance

Oracle VM VirtualBox



Analyzed about 15 hours ago

Oracle VM VirtualBox is a family of powerful x86 virtualization products for enterprise as well as home use. VirtualBox is an extremely feature rich, high performance product for enterprise customers. VirtualBox runs on Windows, Linux, Mac OS X and Solaris hosts in both 32-bit and 64-bit modes. It supports a large number of guest op... [More]

5.45M lines of code

2 current contributors

4 days since last commit

956 users on Open Hub

Compare



Very High Activity



5 Reviews

Mostly written in C

Licenses: GPL-2.0+

x86 virtualization virtual osx machine virtualbox_ose virtual_machine emulation virtualisation oracle sun hypervisor

more...

Virtual-Linux



No analysis available

"Virtual-Linux" is a project undertaken to isolate groups of processes running on a single LINUX machine.

Mostly written in language not available

Licenses: No declared licenses

virtual_machines

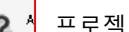
0 lines of code

0 current contributors

0 since last commit

0 users on Open Hub

Compare



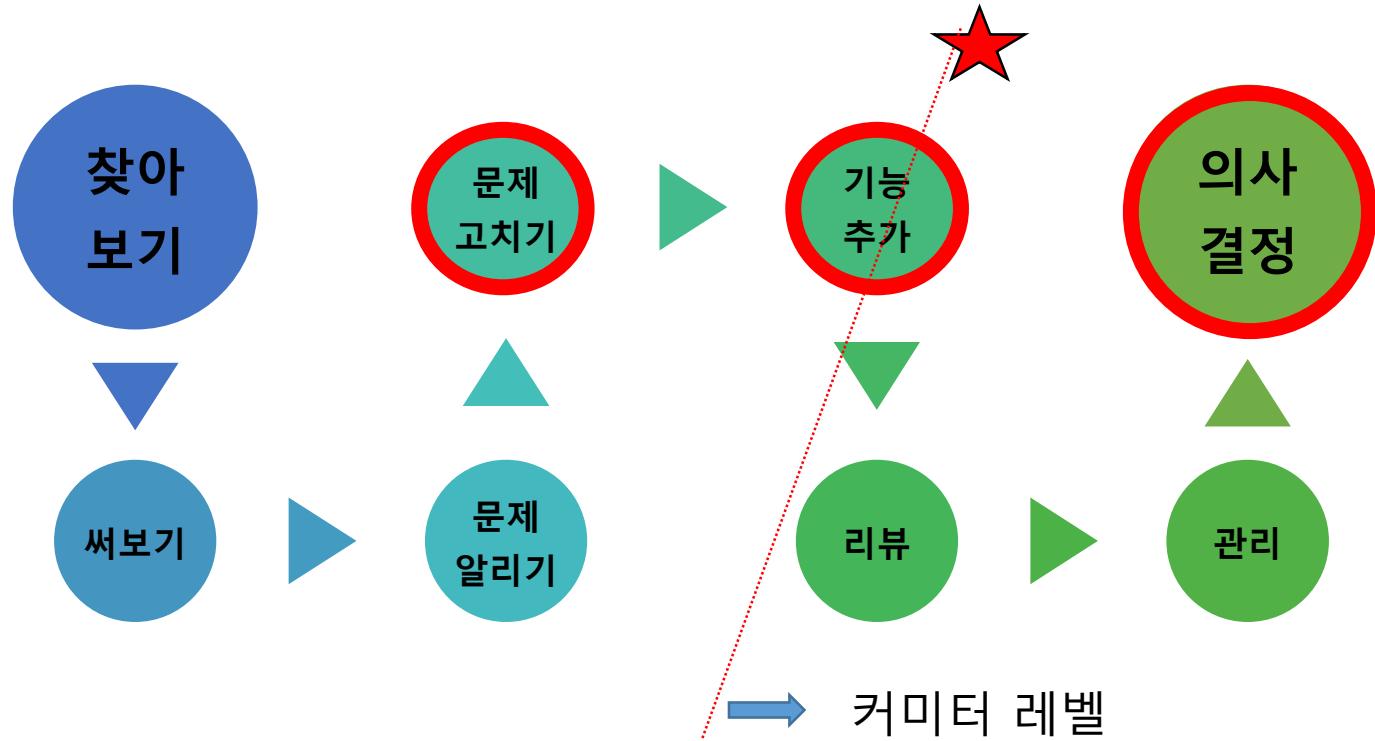
Ask

프로젝트 별로

- 개요 (+라이선스, 언어, ...)
- 현재 line 수
- 개발자 수
- 활동 상황

등을 보여줌

OSS 참여자의 입장에서..



나중엔 내 프로젝트로 큰 커뮤니티 만들기...

구글 오픈소스 라운드 테이블 (12.2.28, 14.9.11)

“오픈 소스 프로젝트를 즐기는데 있어서

필수 조건: ‘고민하지 말고 무조건 저질러라’”

“그 어떤 프로젝트든 자신이 특화 해서 기여할 부분은

조금씩 남아 있으니, 우선 참여해볼 것을 권한다”

“‘이걸 올려도 정말 될까’같은 고민은 하지 않길 바란다”

“영어 실력이 있으면 좋지만, 영어 실력이 문제되지는 않는다”

“공유를 통해서 기쁨을 느끼는 개발자가 늘었으면 좋겠다”

“기업들이 오픈 소스 프로젝트에 참여했던 개발자를 우대하는 분위기이기 때문에

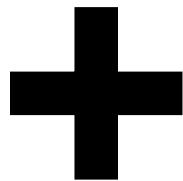
오픈 소스 프로젝트 참여로 얻는 것이 분명 있다”



허준회, 허태준, 김국진, 김남형

선수의 반열에 오르려면..

**열정
진정성**



Contributor가 되려면 ...

- 세 가지 작전

1. 관심 있는 (자주 쓰는) 프로젝트를 선정

- 열심히 써본다.
- 매뉴얼, 사용 팁을 만든다.
- 크고 작은 문제를 발견하고 보고한다.
- 해당 부분 소스를 수정하여 고친다.
- 새로운 기능을 추가한다.

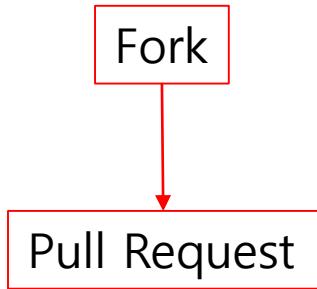
2. 발전 가능성이 있는 프로젝트를 선정

- 한글을 지원한다.
- 다른 프로젝트와 엮어본다.
- 새로운 기능을 제안하고 추가한다.

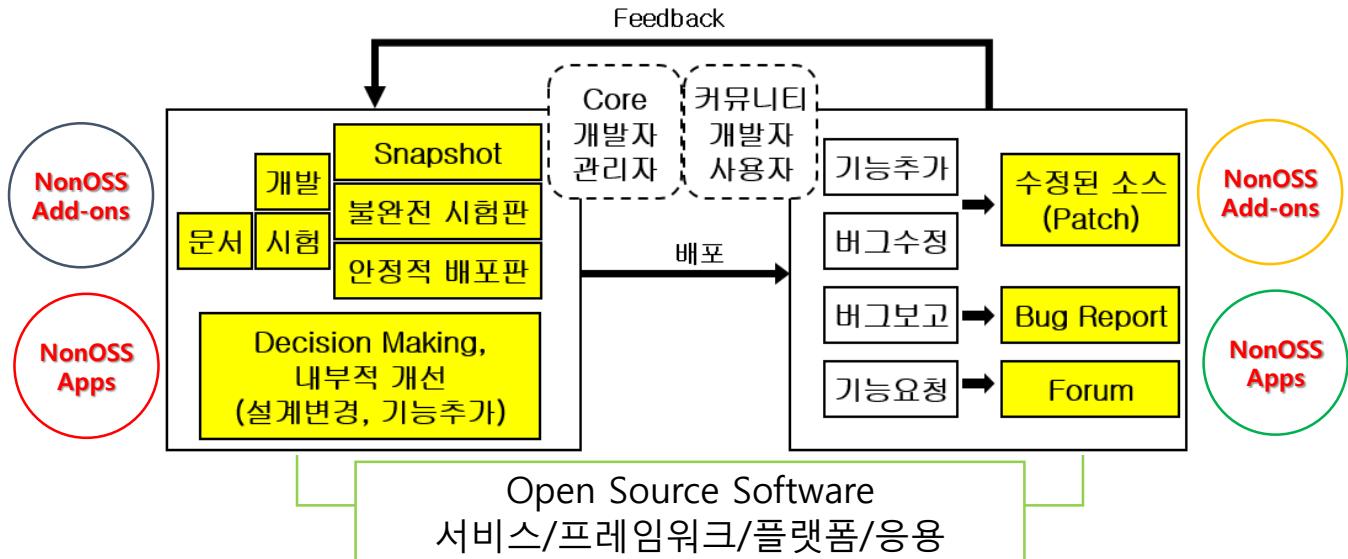
3. 마구잡이로 선정 (큰 프로젝트를 선정)

- Document, Comment에 집중한다.
- Spelling, 비문, ... 수정한다.

* 3번도 좋지만, Wikipedia (한글, 영문) article 쓰기



공개는 언제나 적극적인 공개를 의미



Open Source의 장점을 살리기 위해서는
처음부터 적극적 공개 (의사 결정 과정, 문서, 소스)
즉, 내가 하고 있는 모든 것을 공개해야 한다.

Worth Reading

- [오픈소스 개발 방법론 - Mozilla 사례 중심 \(2010\)](#) by 윤석찬 (Daum)
- <https://code.google.com/p/open-source-class/> by 윤석찬 (Daum)
- [The Origins and Future of Open Source Software](#)
- [오픈소스 소프트웨어의 기술혁신 특성:리뷰](#)
- [Netty 개발자 이희승\(트위터\) 커미터, 성지순례 사이트](#)
- [레이어 별 오픈소스](#) by 최지웅 (오픈 소스 컨설팅)
- [공개SW 전환방법 및 전략](#) by 김형태 (아비도스)
- [한국 오픈 소스 커미터 목록 \(KLDP\)](#)
- [Open Source iOS Apps – Real iOS Source Code Examples](#)
- [성당과 시장 \(한빛출판사 무료 ebook\)](#) – Eric Raymond
- [오픈소스로 개발 실력 쌓기](#) by 진성주 (아파치 usergrid 커미터)
- [Head of Open Source Software \(OSS\)](#) – OSS 헤드 구인 광고
- [오픈 소스, 왜 해야 하나?](#) – by 김수보 (eBrain 아카데미)
- [왜 공개소스 소프트웨어여야 하나?](#) – by 이민석

Q&A

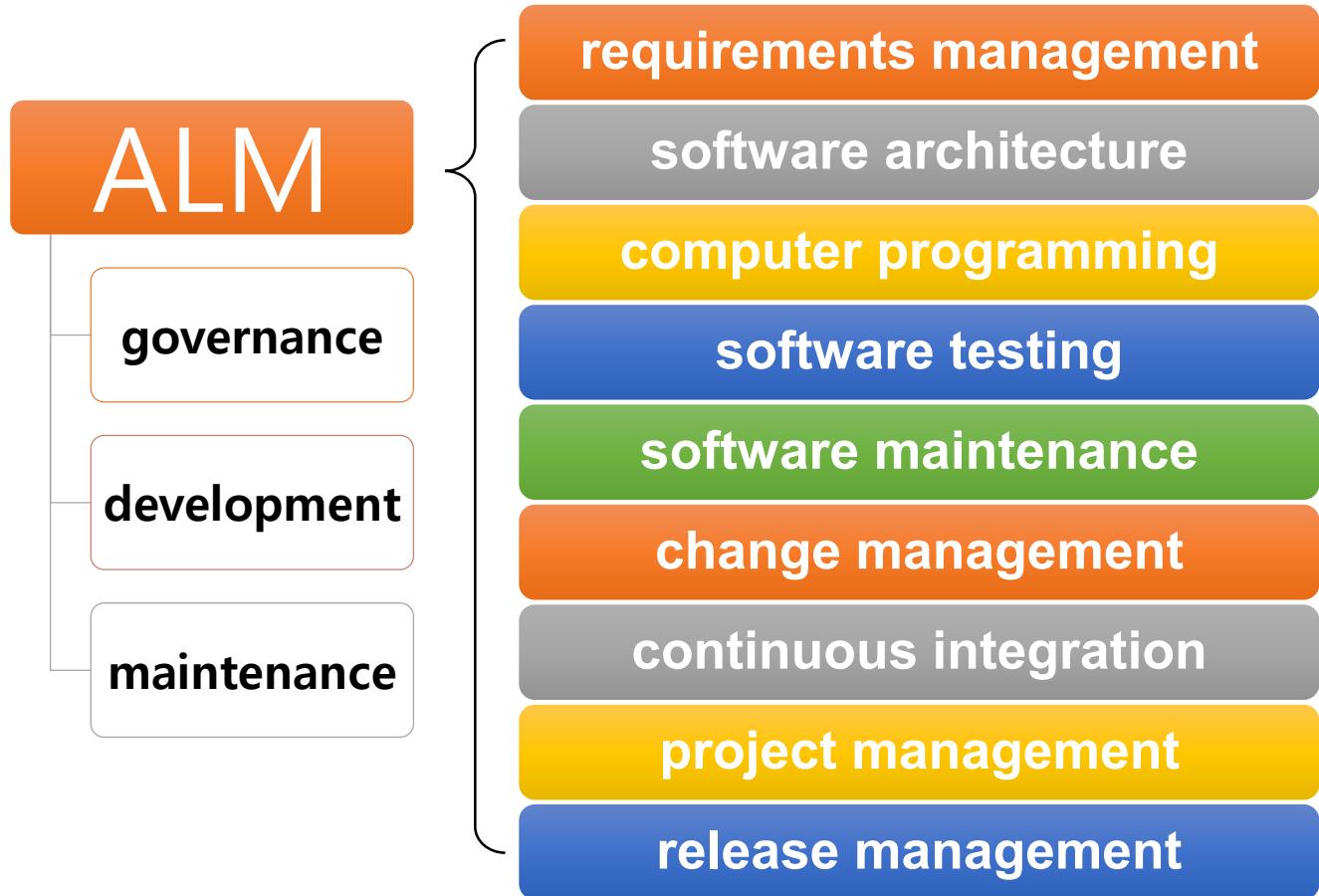
(중간 정산)

Minsuk Lee

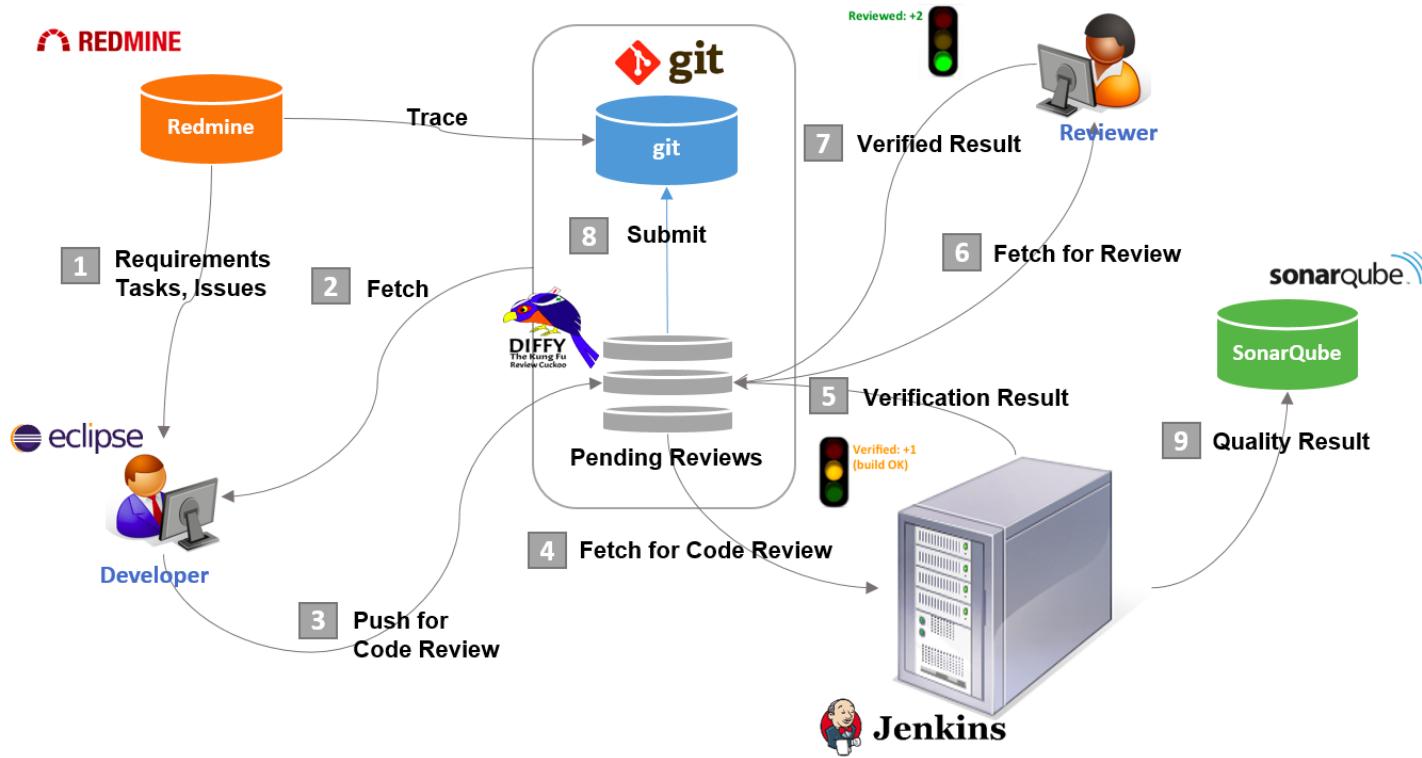
컴퓨터공학부.국민대학교

<https://www.facebook.com/minsuk.lee0>

ALM (Application Lifecycle Management)



Typical ALM with Open Source



Typical ALM Workflow

1. 개발자는 Redmine에 작성되어 있는 요구사항, 업무, 이슈들을 이클립스의 작업 리스트에서 확인
2. 개발자는 자신의 업무와 관련 있는 소스 코드를 Git + Gerrit 으로부터 Fetch 받음
3. 개발자가 코딩과 Local Test를 마친 자신의 소스 코드를 리뷰 요청을 위해 Gerrit 에 Push 함
4. Jenkins는 빌드, 정적분석, 단위 테스트를 위해 Gerrit에 Push된 소스 코드를 Fetch 함
5. Jenkins는 확인 결과를 Gerrit에 등록
6. 리뷰어들은 개발자가 개발한 소스 코드의 리뷰를 수행
7. 리뷰어들은 검증 결과를 Gerrit에 등록
8. Gerrit은 코드 리뷰 결과와 빌드 검증 결과를 바탕으로 소스를 Git에 submit

Github

Open Source 계에서는 github이 이제 갑

Virtual Machine 검색

Search virtual machine

We've found 2,029 repository results

Sort: Best match ▾

mitchellh/virtualbox
Ruby ★ 233 ⚡ 46
Create and modify virtual machines in VirtualBox using pure ruby.
Updated on 23 Dec 2011

railsbridge/railsbridge-virtual-machine
Ruby ★ 13 ⚡ 12
Railsbridge Virtual Machine
Updated on 4 Jan

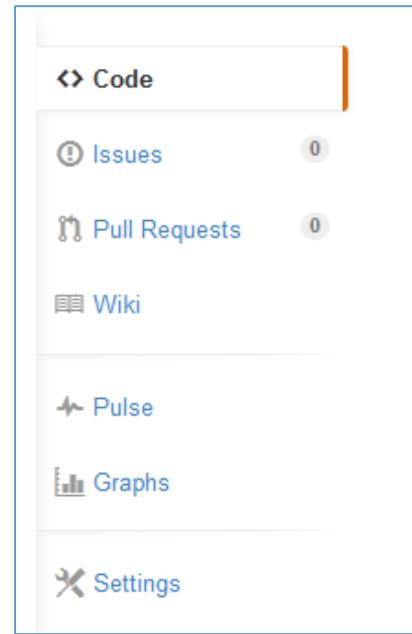
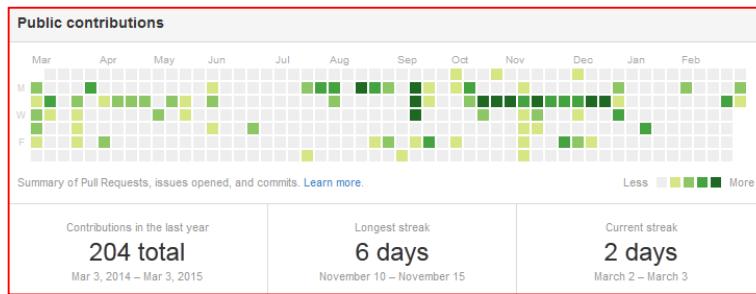
parrot/parrot
Parrot Virtual Machine
Updated 3 days ago

parrt/simple-virtual-machine
Java ★ 13 ⚡ 9
A simple VM for a talk on building VMs
Updated 4 days ago

**Github이 최근
신규 (학생) 개발자 유입을 위한
Promotion을 시작함
<https://education.github.com/pack>**

Github?

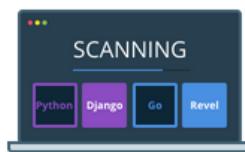
- Git을 기반으로 하는 소프트웨어 프로젝트 관리 사이트
- GITHUB과 git 관계?
 - Git이라는 Source Control 방법을 github이 사용할 뿐
- Github을 쓴다는 것은
 - Source Control
 - Issue Tracking/Control
 - 협업 도구 (fork/pull request)
 - Statistics 등을 쓴다는 것



왜 오픈소스 인가?

source{d}

How did we find you?



Code Analysis

It all starts in the code. We have analyzed over 14 million open source repositories, extracting features along the process.

More on our Analysis

Matching with jobs

We automatically match the profiles obtained in our analysis with our open job offers.

More on Matching

Contact you

We send highly relevant job proposals to the suggested profiles and inform them about all the aspects of the position.

More on our Communication

<http://sourced.tech/>

Github은 꽤 중요해서..

- <http://rankedin.kr/>
- <http://git.io/top>
- <https://code.google.com/p/gitinspector/>
- <http://resume.github.io/>
- <http://git.io/top>
- <http://ghv.artzub.com/>
- <https://www.openhub.net/people>

Github 시작하기 - github.com

The screenshot shows the GitHub homepage with a dark background featuring various development-related icons. At the top left is the GitHub logo. To its right are navigation links: Explore, Features, Enterprise, and Blog. On the far right is a blue "Sign in" button. The main headline reads "Build software better, together." Below it, a sub-headline says "Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo." To the right, there's a registration form with a placeholder for a username and a password field. A large green "Sign up for GitHub" button is at the bottom. A red oval highlights this button. A small note below the button states: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally."

GitHub

Explore Features Enterprise Blog

Sign in

Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo.

Use at least one lowercase 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 will send you account related emails occasionally.

Repository 만들기



The screenshot shows the GitHub homepage. A red oval highlights the 'New repository' button in the top right corner of the header bar. Below the header, there's a 'GitHub Bootcamp' modal window with four numbered steps: 1. Set up Git, 2. Create repositories, 3. Fork repositories, and 4. Work together.

Step	Icon	Description
1		Set up Git A quick guide to help you get started with Git.
2		Create repositories Repositories are where you'll work and collaborate on projects.
3		Fork repositories Forking creates a new, unique project from an existing one.
4		Work together Send pull requests, follow friends. Star and watch projects.

또는... 다른 원본 소스를 fork 해오면
같은 복사된 repository가 내게도 생김 !!

Github Repository 생성

Owner Repository name



hl1itj

Repository name

test-repo



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

Description (optional)

저장소 이름과 설명

Test Repository

Public

남들도 볼 수 있게, private은 유료

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

Private

You choose who can see and commit to this repository.

Readme를 만들어 줌

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: C ▾

Add a license: MIT License ▾

라이선스 선택

Create repository

버전 관리 예외 파일 등록
- 언어를 선택

Github Repository 화면

Unwatch 1

Star 0

Fork 0

Test Repository — Edit

Master Branch 표시

살아있는
issue, pull request 수

1 commit

1 branch

0 releases

1 contributor



branch: master

test-repo / +

Initial commit

hl1itj authored just now

.gitignore

LICENSE

README.md

이 Repository에서 유지하고 있는
소스 Branch 수

README.md

.gitignore, license 선택 결과

test-repo

Test Repository

자동으로 만들어진 '제목만 있는'
Readme.md (Markdown 언어로 편집 가능)

Issues

Pull requests

Wiki

Pulse

Graphs

Settings

HTTPS clone URL

<https://github.com/l>

You can clone with HTTPS, SSH,
or Subversion.

Clone in Desktop

Download ZIP

Github Repository 화면

Contributor
이 프로젝트 참여자수

Click 해서 Issue제기

Release 하면
이름 붙은 Tar Ball을
Download 할 수 있게 만들어 줌

클릭해서
파일들을 직접 수정 가능
Commit를 하게 됨

로컬 환경에서
이 github repository를 보는 URL

Contributor
1 contributor

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

Settings

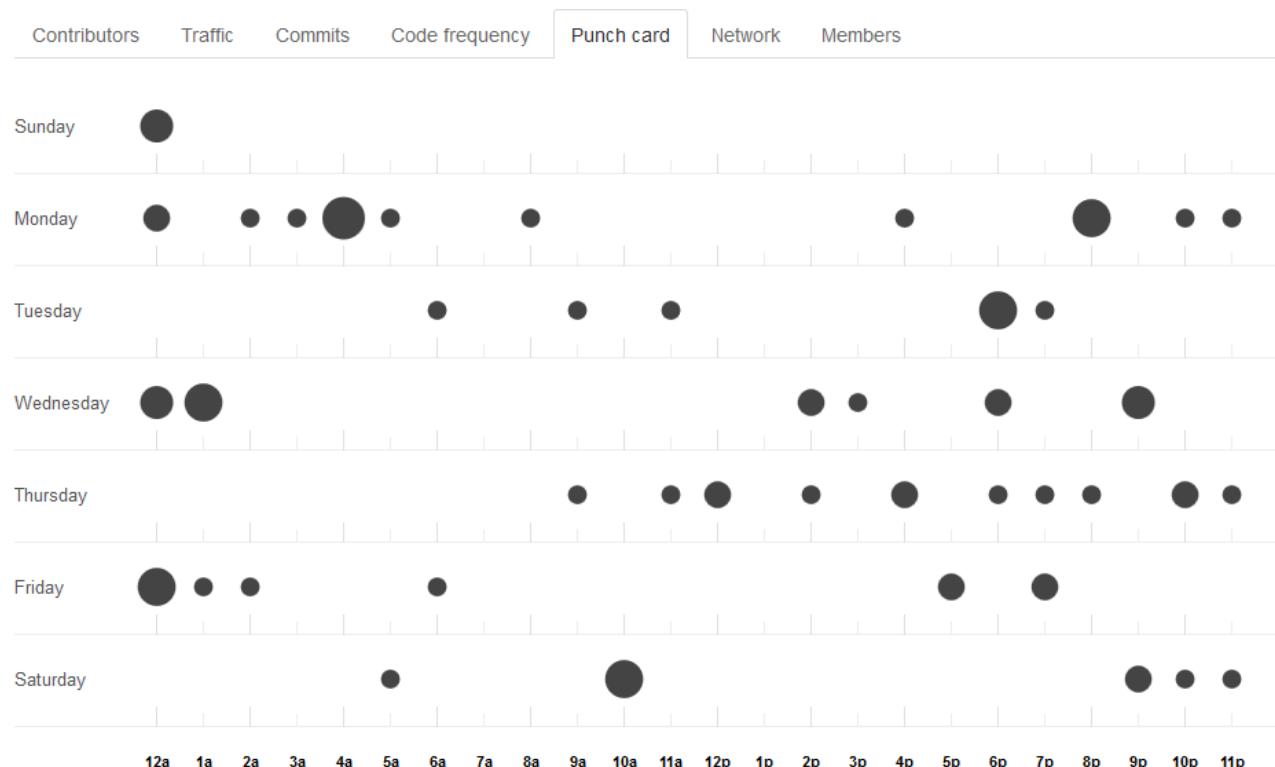
HTTPS clone URL
<https://github.com/>

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Repository Punch Card (contributor 화면에서)





hl1itj / test-repo

branch: master ▾

test-repo / .gitignore



hl1itj 17 minutes ago Initial commit

1 contributor

33 lines (27 sloc) | 0.246 kB

```
1 # Object files
2 *.o
3 *.ko
4 *.obj
5 *.elf
6
7 # Precompiled Headers
8 *.gch
9 *.pch
10
11 # Libraries
12 *.lib
13 *.a
14 *.la
15 *.lo
16
17 # Shared objects (inc. Windows DLLs)
18 *.dll
19 *.so
20 *.so.*
21 *.dylib
22
```

.gitignore 내용

Object file,
Precompiled Header,
Shared library 등

Build해서 만들어지는 것은
모두 version 관리에서 제외
(편집 가능)

LICENSE 내용

branch: master [test-repo / LICENSE](#)

hl1itj 22 minutes ago Initial commit

1 contributor

23 lines (17 sloc) | 1.068 kB

[Raw](#) [Blame](#) [History](#)   

1 The MIT License (MIT)

2

3 Copyright (c) 2015

4

[Raw](#)[Blame](#)[History](#)9 ~~Copies of the software, and to permit persons to whom the software is~~

10 furnished to do so, subject to the following conditions:

11

12 The above copyright notice and this permission notice shall be included in all
13 copies or substantial portions of the Software.

14

15 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
16 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
17 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
18 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
19 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
20 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
21 OF THE SOFTWARE.

편집, 삭제가 가능

Revision History

[Edit file](#)[Preview changes](#)

Spaces

2

No wrap



1 The MIT License (MIT)

2

3 Copyright (c) 2015, Minsuk Lee, Kookmin University|

4

5 Permission is hereby granted, free of charge, to any person obtaining a copy
6 of this software and associated documentation files (the "Software"), to deal
7 in the Software without restriction, including without limitation the rights
8 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell20 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
21 SOFTWARE.

22

23

수정한 부분

Commit 메시지

원본(master)는 그대로 두고
새로운 Branch를 만들 수도 있음

Commit changes

add author, etc

Add an optional extended description...

 Commit directly to the master branch Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)[Commit changes](#)[Cancel](#)

LICENSE 수정 후 변화

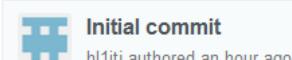
History for test-repo / LICENSE

Commits on Jun 15, 2015



hl1itj authored 2 minutes ago

Commits on Jun 14, 2015



hl1itj authored an hour ago

b3f1f03



Commit 메시지와 함께
History에 추가됨

Commit 시점의 파일 보기

test-repo / LICENSE



Blame 화면

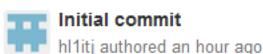
Newer Older

100644 | 23 lines (17 sloc) | 1.099 kB

Raw

Normal view

History



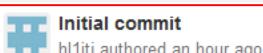
071c261

The MIT License (MIT)



b3f1f03

Copyright (c) 2015, Minsuk Lee, Kookmin University



071c261

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights

History for test-repo / LICENSE

모든 변화에는 ID가 할당됨

Commits on Jun 15, 2015



add author, etc

hl1itj authored 2 minutes ago



b3f1f03



Commits on Jun 14, 2015

add author, etc

master



hl1itj authored 17 minutes ago

1 parent 071c261 commit b3f1f03615fe5aeb59be4b3ae7cd54cf992f9323

Browse files



Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

2 3 4 5 6 LICENSE

View

... ... @@ -1,6 +1,6 @@

3

-Copyright (c) 2015

3

+Copyright (c) 2015, Minsuk Lee, Kookmin University

6 6 of this software and associated documentation files (the "Software"), to deal



Write

Preview

Markdown supported

Edit in fullscreen

Leave a comment

Issue.. 는 Issue다.

The screenshot shows a GitHub repository named 'hl1itj / test-repo' with the 'Issues' tab selected. A red box highlights the explanatory text about what issues are, and a red circle highlights the 'New issue' button.

Issues are used to... 어떤 이슈든
이 프로젝트와 상관이 있는
'해결되어야 하는' 문제/comment를 기록하고
해결되는 과정을 Track

New issue

Welcome to Issues!

Issues are used to... 어떤 이슈든
이 프로젝트와 상관이 있는
'해결되어야 하는' 문제/comment를 기록하고
해결되는 과정을 Track

1. Bug Report (재현 방법까지)
2. 새로운 기능 요구
3. Review / Comment
4. ...

Issue 제기하기

hl1itj / test-repo

Issues Pull requests Labels Milestones

Add Makefile

We need a Makefile to easy build!

Write Preview Markdown supported Edit in fullscreen

Attach images by dragging & dropping or selecting them.

제기된 이슈는 모든 사람이 볼 수 있음

Submit new issue

전체 Issue 화면

The screenshot shows a GitHub repository interface for 'hl1itj/test-repo'. The top navigation bar includes 'Issues' (selected), 'Pull requests', 'Labels', and 'Milestones'. A search bar displays the query 'is:issue is:open'. On the right, there are buttons for 'Unwatch', 'Star', 'Fork', and 'New issue'.

The main area shows a summary: 1 Open issue and 0 Closed issues. Below this is a list of issues, starting with '#1 Add Makefile' (opened 8 minutes ago).

Two specific UI elements are highlighted with red boxes:

- A dropdown menu labeled 'Author ▾ Labels ▾ Milestones ▾ Assignee ▾ Sort ▾' located at the top right of the issue list.
- An 'Apply labels' modal window with a 'Filter labels' input field and a list of available labels: bug, duplicate, enhancement, help wanted, invalid, question, and wontfix.

Red arrows point from the 'Labels' dropdown in the main list to the 'Labels' dropdown in the header, and from the 'Labels' dropdown in the header to the 'Apply labels' modal.

Apply labels

Filter labels

- bug
- duplicate
- enhancement
- help wanted
- invalid
- question
- wontfix

각 Issue 화면..

hl1itj / test-repo Unwatch 1 Star 0 Fork 0

Add Makefile #1

Open hl1itj opened this issue just now · 0 comments

hl1itj commented just now

We need a Makefile to easy build

Owner

Write Preview Markdown supported Edit in fullscreen

여기에서.. 이 이슈에 대하여 comment를 달 수도 있고

Close issue Comment

Labels None yet

Milestone No milestone

Assignee No one—assign yourself

Notifications

Unsubscribe You're receiving notifications because you authored the thread.

1 participant

Issue를 close하고 나면

The screenshot shows a GitHub repository page for 'hl1itj / test-repo'. The 'Issues' tab is selected, showing 0 open issues and 1 closed issue. A red arrow points from the text '해결된 Issue로 이전' (Move to solved Issue) to the '1 Closed' link. The 'closed' issue is not visible in the list. Below the list, there is a message: 'There aren't any c' followed by 'Use the links above to find what you're looking for, or try a new search...'. A large red box highlights the text '모든 사소한, 중대한 쟁겨야 할 것들을 이슈로 등록하고' and '하나씩 해결하면서 개발'.

해결된 Issue로 이전

Issues Pull requests Labels Milestones Filters is:issue is:open New issue

0 Open 1 Closed

Author Labels Milestones Assignee Sort

There aren't any c

Use the links above to find what you're looking for, or try a new search... for quickly finding issues m

모든 사소한, 중대한
쟁겨야 할 것들을
이슈로 등록하고

하나씩 해결하면서 개발

Markdown Language

```
6 An h1 header
7 -----
8
9 Paragraphs are separated by a blank line.
10
11 2nd paragraph. *Italic*, **bold**, and `monospace`. Itemized lists
12 look like:
13
14     * this one
15     * that one
16     * the other one
17
18 Note that --- not considering the asterisk ---
19 content starts at 4-columns in.
20
21 > Block quotes are
22 > written like so.
23 >
24 > They can span multiple paragraphs,
25 > if you like.
26
27 Use 3 dashes for an em-dash. Use 2 dashes for
28 in chapters 12--14"). Three dots ... will be
29 Unicode is supported. ☺
30
31
32
33 An h2 header
34 -----
```

README.md



An h1 header

Paragraphs are separated by a blank line.

2nd paragraph. *Italic*, **bold**, and `monospace`. Itemized lists look like:

- this one
- that one
- the other one

Note that --- not considering the asterisk --- the actual text content starts at 4-columns in.

Block quotes are written like so.

They can span multiple paragraphs, if you like.

Use 3 dashes for an em-dash. Use 2 dashes for ranges (ex., "it's all in chapters 12–14"). The will be converted to an ellipsis. Unicode is supported. ☺

An h2 header

다른 프로젝트 fork 해오기

This repository

Search

Pull requests Issues Gist

Watch 1 Star 0 Fork 0

hl1itj / test-repo

Test Repository

4 commits 1 branch 0 releases 1 contributor

branch: master test-repo / +

minor change

hl1itj authored 27 minutes ago

.gitignore

LICENSE

README.md

README.md

QUICK MARKDOWN EXAM

copied from <http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html>

An h1 header

Code

Issues 0

Pull requests 0

Wiki

Pulse

Graphs

HTTPS clone URL

<https://github.com/l1itj/test-repo>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop

Download ZIP

이 화면은

'minsuklee' account에서 본
hl1itj/test-repo 저장소

Fork 결과

This repository Search Pull requests Issues Gist

Unwatch 1 Star 0 Fork 1

minsuklee / test-repo forked from hl1tij/test-repo

4 commits 1 branch 0 releases 1 contributor

branch: master test-repo / +

This branch is even with hl1tij:master

minor change

hl1tij authored 31 minutes ago

.gitignore

LICENSE

README.md

README.md

QUICK MARKDOWN EXAMPLE

copied from <http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html>

Code

Pull requests 0

Wiki

Pulse

Graphs

Settings

HTTPS clone URL
<https://github.com/l>

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

이 화면은
(fork 해온)
minsuklee/test-repo 저장소

Pull request 연습 (1/10)

minsuklee / test-repo
forked from h1it/test-repo

Unwatch 1 Star 0 Fork 1

test-repo / README.md or cancel

Edit file Preview changes Spaces 1 Soft wrap

```
1 QUICK MARKDOWN EXAMPLE - touched by minsuklee|  
2  
3 copied from http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html  
4  
5  
26  
27 Use 3 dashes for an em-dash. Use 2 dashes for ranges (ex., "it's all
```

Commit changes

Update README.md

for pull request practice

Commit directly to the master branch

Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

수정 후 log와 함께 commit

Commit changes Cancel

Pull request 연습 (2/10)

minsuklee / test-repo
forked from hl1itj/test-repo

branch: master test-repo / README.md

minsuklee a minute ago Update README.md

2 contributors

163 lines (111 sloc) 3.536 kB

Raw Blame History

QUICK MARKDOWN EXAMPLE - touched by minsuklee

copied from <http://hl1itj/test-repo>

An h1 header

원본 repo 아직 그대로..

hl1itj 44 minutes ago minor change

1 contributor

163 lines (111 sloc) 3.513 kB

Raw Blame History

QUICK MARKDOWN EXAMPLE

copied from <http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html>

An h1 header

Pull request 연습 (3/10)

This repository Search

Pull requests Issues Gist

Unwatch 1 Star 0 Fork 1

minsuklee / test-repo
forked from hi1itj/test-repo

Test Repository — Edit

5 commits 1 branch 0 releases 1 contributor

branch: master + test-repo / +

This branch is 1 commit ahead of hi1itj:master

Update README.md ...

minsuklee authored 6 minutes ago latest commit c5f7a9f961

.gitignore Initial commit 11 hours ago

LICENSE add author, etc 10 hours ago

README.md Update README.md 6 minutes ago

README.md

QUICK MARKDOWN EXAMPLE - touched by minsuklee

copied from <http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html>

Code Pull requests Wiki Pulse Graphs Settings

HTTPS clone URL <https://github.com/>

You can clone with HTTPS, SSH, or Subversion. ⚙

Clone in Desktop Download ZIP

An h1 header

Pull request 연습 (4/10)

hl1itj / test-repo → 원래 repository와 비교

Watch 1 Star 0 Fork 1

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base fork: hl1itj/test-repo base: master ... head fork: minsuklee/test-repo compare: master ✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others. ②

1 commit

Commits on Jun 15, 2015

minsuklee

정말 PULL-REQUEST?
누르면... hl1itj/test-repo 에
pull request 가 전달됨

commit comments 1 contributor c5f7a9f

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

README.md

2	1	@@ -1,4 +1,4 @@
...	...	-QUICK MARKDOWN EXAMPLE
1	1	+QUICK MARKDOWN EXAMPLE - touched by minsuklee
2	2	
3	3	copied from http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html
4	4	

No commit comments for this range

Pull request 연습 (5/10)

hl1itj / test-repo 아직도 'minsuklee' 화면

Update README.md #2

Open minsuklee wants to merge 1 commit into `hl1itj:master` from `minsuklee:master`

Conversation 0 Commits 1 Files changed 1 +1 -1

minsuklee commented 5 minutes ago
for pull request practice

Update README.md ... c5f7a9f

Add more commits by pushing to the `master` branch on [minsuklee/test-repo](#).

This pull request can be automatically merged by project collaborators.
Only those with [write access](#) to this repository can merge pull requests.

Write Preview Markdown supported Edit in fullscreen

This is additional 코멘트

추가적인 comment

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

Comment

Pull request 연습 (6/10)

hl1itj / test-repo

이 화면은
hl1itj 가 보는 test-repo
(아직 반영 안됨)

Unwatch 1 Star 0 Fork 1

Test Repository — Edit

4 commits 1 branch 0 releases 1 contributor

branch: master + test-repo / +

minor change

hl1itj authored an hour ago latest commit 4ba506d24f

.gitignore Initial commit 11 hours ago

LICENSE add author, etc 11 hours ago

README.md minor change an hour ago

README.md

QUICK MARKDOWN EXAMPLE

copied from <http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html>

Code Issues Pull requests 1

Wiki Pulse Graphs Settings

HTTPS clone URL <https://github.com/t>

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop Download ZIP

An h1 header

Pull request 연습 (7/10)

[test-repo] Update README.md (#2) Minsuk Lee 오전
보내 사람 Minsuk Lee <notifications@github.com>★

답장 주소 hl1itj/test-repo <reply+0018d7fe42e42884254c62e04894f05965d05f0636d20f7a92cf000000011195ef4d92a169ce05431fb... ★
받는 사람 hl1itj/test-repo <test-repo@noreply.github.com>★

제목 [test-repo] Update README.md (#2)

for pull request practice

You can view, comment on, or merge this pull request online at:

<https://github.com/hl1itj/test-repo/pull/2>

Commit Summary

- Update README.md

누군가 프로젝트에
Pull Request를 하면
hl1itj 에게 메일이 도착

File Changes

- M [README.md](#) (2)

Patch Links:

- <https://github.com/hl1itj/test-repo/pull/2.patch>
- <https://github.com/hl1itj/test-repo/pull/2.diff>

Reply to this email directly or [view it on GitHub](#).

Pull request 연습 (8/10)

hl1itj / test-repo

Issues Pull requests Labels Milestones Filters is:pr is:open New pull request

1 Open 0 Closed

Update README.md (opened 12 minutes ago) by mnsuklee

Author Labels Milestones Assignee Sort

Author Labels Milestones Assignee Sort

PULL REQUEST 역시
ISSUE와 같은 수준의 처리

Pull request 연습 (9/10)

hl1itj / test-repo

Unwatch 1 Star 0 Fork 1

Update README.md #2

Open minsuklee wants to merge 1 commit into hl1itj:master from minsuklee:master

Conversation 1 Commits 1 Files changed 1

minsuklee commented 13 minutes ago
for pull request practice

Update README.md ... c5f7a9f

minsuklee commented 7 minutes ago
This is additional 코멘트

This pull request can be automatically merged.
You can also merge branches on the command line.

Merge pull request

Write Preview Markdown

Leave a comment

Attach images by dragging & dropping or selecting them.

Close pull request Comment

Labels None yet

Milestone No milestone

Assignee No one—assign yourself

Notifications Unsubscribe

You're receiving notifications because you're subscribed to this repository.

정말 MERGE?
누르면... minsuklee의 수정 내용을
master branch와 merge 함
(confirm/comment 화면 경우)

Pull request 연습 (10/10)

The screenshot shows a GitHub repository page for `hl1itj/test-repo`. The repository has 6 commits, 1 branch (master), 0 releases, and 1 contributor (minsuklee). A red box highlights the 'Pull requests' link in the sidebar. Another red box highlights the commit message 'QUICK MARKDOWN EXAMPLE - touched by minsuklee' in the commit list.

Test Repository — Edit

6 commits 1 branch 0 releases 1 contributor

branch: master test-repo / +

Merge pull request #2 from minsuklee/master ...

hl1itj authored 3 minutes ago latest commit 310f2/c7bb

.gitignore Initial commit 12 hours ago

LICENSE add author, etc 11 hours ago

README.md Update README.md 44 minutes ago

README.md

QUICK MARKDOWN EXAMPLE - touched by minsuklee

copied from <http://www.unexpected-vortices.com/>

Issues Pull requests 0

Wiki

Pulse

Graphs

Settings

HTTPS clone URL

https://github.com/

hl1itj / test-repo

Unwatch 1 Star 0 Fork 1

Pull requests Labels Milestones Filters is:pr is:open New pull request

0 Open ✓ 1 Closed

Author Labels Milestones Assignee Sort

There aren't any open pull requests.

minsuklee의 수정 내용 반영됨

git

Local에서 ... git download

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

The screenshot shows the official Git website's download page. On the left, there's a sidebar with links for About, Documentation, Blog, Downloads (which is currently selected), and Community. The main content area has a large "Downloads" heading. Below it, there are four download links: Mac OS X, Windows, Linux, and Solaris. To the right, a monitor icon displays the latest source release version 2.3.1 and a "Downloads for Windows" button. At the bottom left, there's a note about the Pro Git book and a section for GUI Clients. A red box highlights the "GUI Clients" section and the terminal commands for apt-get.

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

[Mac OS X](#) [Windows](#)
[Linux](#) [Solaris](#)

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

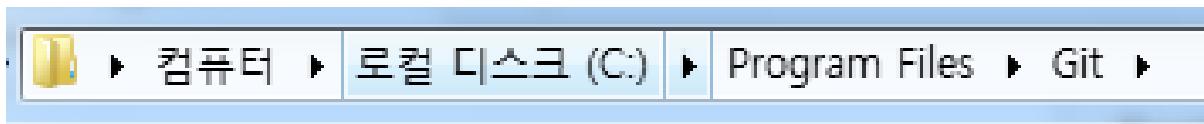
GUI Clients

Git comes with `gitk`, but there are other clients for users looking for a graphical experience.

[View GUI Clients](#)

OR

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



C:\Program Files\Git



📁 bin	2015-03-03 오후...	파일 폴더
📁 cmd	2015-03-03 오후...	파일 폴더
📁 doc	2015-03-03 오후...	파일 폴더
📁 etc	2015-03-03 오후...	파일 폴더
📁 git-cheetah	2015-03-03 오후...	파일 폴더
📁 lib	2015-03-03 오후...	파일 폴더
📁 libexec	2015-03-03 오후...	파일 폴더
📁 share	2015-03-03 오후...	파일 폴더
📁 ssl	2015-03-03 오후...	파일 폴더
Git Bash.vbs	2014-12-17 오후...	VBScript 스크립... 1KB
ReleaseNotes.rtf	2014-12-17 오후...	서식 있는 텍스트 46KB
unins000.dat	2015-03-03 오후...	DAT 파일 578KB
unins000.exe	2015-03-03 오후...	응용 프로그램 1,216KB

MINGW32:/

Welcome to Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.

Run 'git help <command>' to display help for specific commands.

user@CS2-PC-42 ~

\$

Mingw는 처음이지?

\$ cd /c

하면 C:\> 위치로 감

설정 후 clone 해 옴

clone : remote repository를 복사해옴

```
MINGW32:/c/temp
user@CS2-PC-42 /c/temp
$ cd

user@CS2-PC-42 ~
$ cd /c/temp

user@CS2-PC-42 /c/temp
$ git config --global user.name "hl1itj"

user@CS2-PC-42 /c/temp
$ git clone https://github.com/hl1itj/test-repo.git
Cloning into 'test-repo'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.

user@CS2-PC-42 /c/temp
$ ls
HncDownload test-repo

user@CS2-PC-42 /c/temp
$ 

user@CS2-PC-42 /c/temp
```

당연히 GUI version도 있음

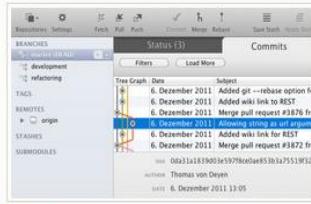
<http://git-scm.com/downloads/guis>



GitHub for Mac

Platforms: Mac

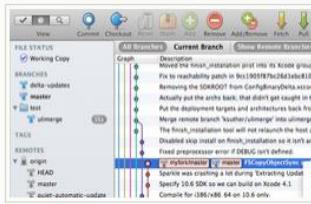
Price: Free



Tower

Platforms: Mac

Price: \$59/user (Free 30 day trial)



SourceTree

Platforms: Mac, Windows

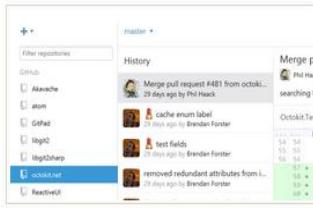
Price: Free



git-cola

Platforms: Windows, Mac, Linux

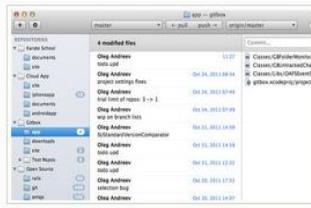
Price: Free



GitHub for Windows

Platforms: Windows

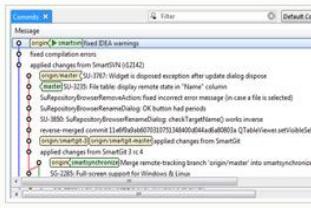
Price: Free



Gitbox

Platforms: Mac

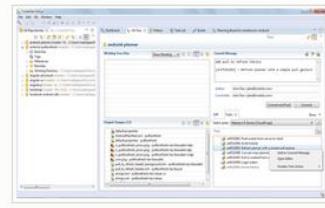
Price: \$14.99



SmartGit

Platforms: Windows, Mac, Linux

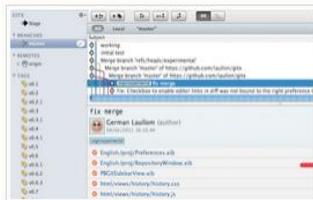
Price: \$79/user / Free for non-commercial use



GitEye

Platforms: Windows, Mac, Linux

Price: Free



GitX-dev

Platforms: Mac

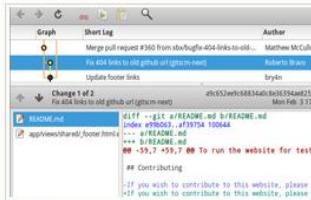
Price: Free



Git Extensions

Platforms: Windows

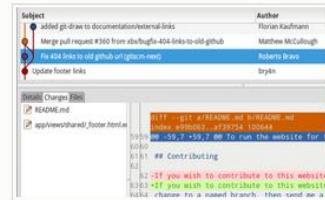
Price: Free



giggle

Platforms: Linux

Price: Free



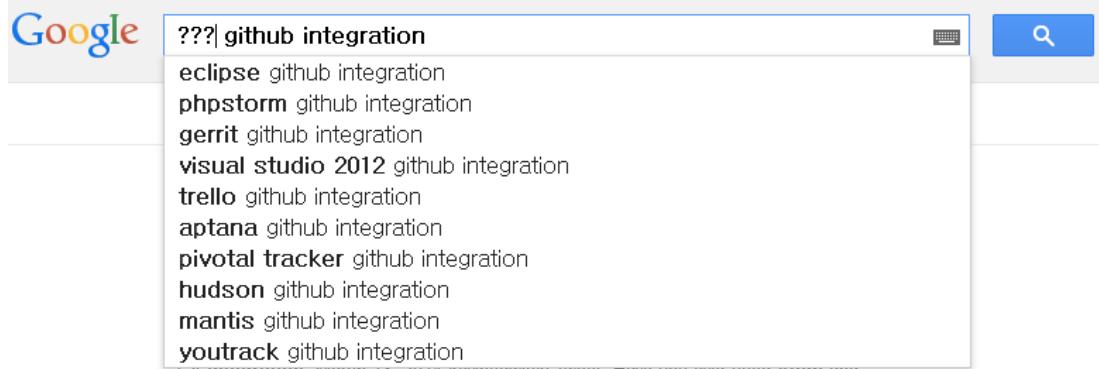
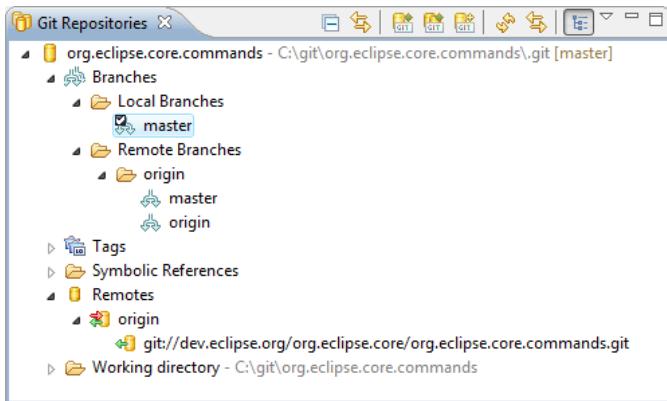
git

Platforms: Linux

Price: Free

그리고 ...

- 모든 IDE가 git을 지원
 - Eclipse
 - IntelliJ
 - Visual Studio Code
- 모든 뜰뚝한 Editor도
 - Sublime Text
 - Notepad++
 - Atom



'git' survival guide (1/2)

```
$ git config --global user.email "name@domain.com"
```

```
$ git config --global user.name "your_name"
```

```
$ git clone https://github.com/<id>/<test-repo.git
```

→ github의 test-repo 를 local work space로 복사해옴 (.git 생성)

```
$ cd test-repo      → 이 위치가 작업 Directory
```

// 이 Directory에서 작업

Clone 대신 Local 쪽에서 먼저 시작할 수도 있음

→ **비추** : github에서 빈 repository를 만들고 clone 하는 것을 권장 !!

```
$ mkdir test-repo
```

```
$ git init → .git 폴더가 만들어짐 (이 안에 모든 local version history가 저장됨)
```

// 작업 후

```
$ curl 명령으로 ... 복잡함 (github command line API 이용)
```

'git' survival guide (2/2)

// clone으로 생성된 'test-repo' directory에서 작업

// create, edit, compile, test, ...

\$ git add * → '*' 또는 추가/수정한 file들, 버전 관리 대상으로 지정

→ Source, Makefile, Configuration File, Sample Data 등,

→ NO BINARY please : .gitignore에 file extension 등록

\$ git commit -m "my commit messages"

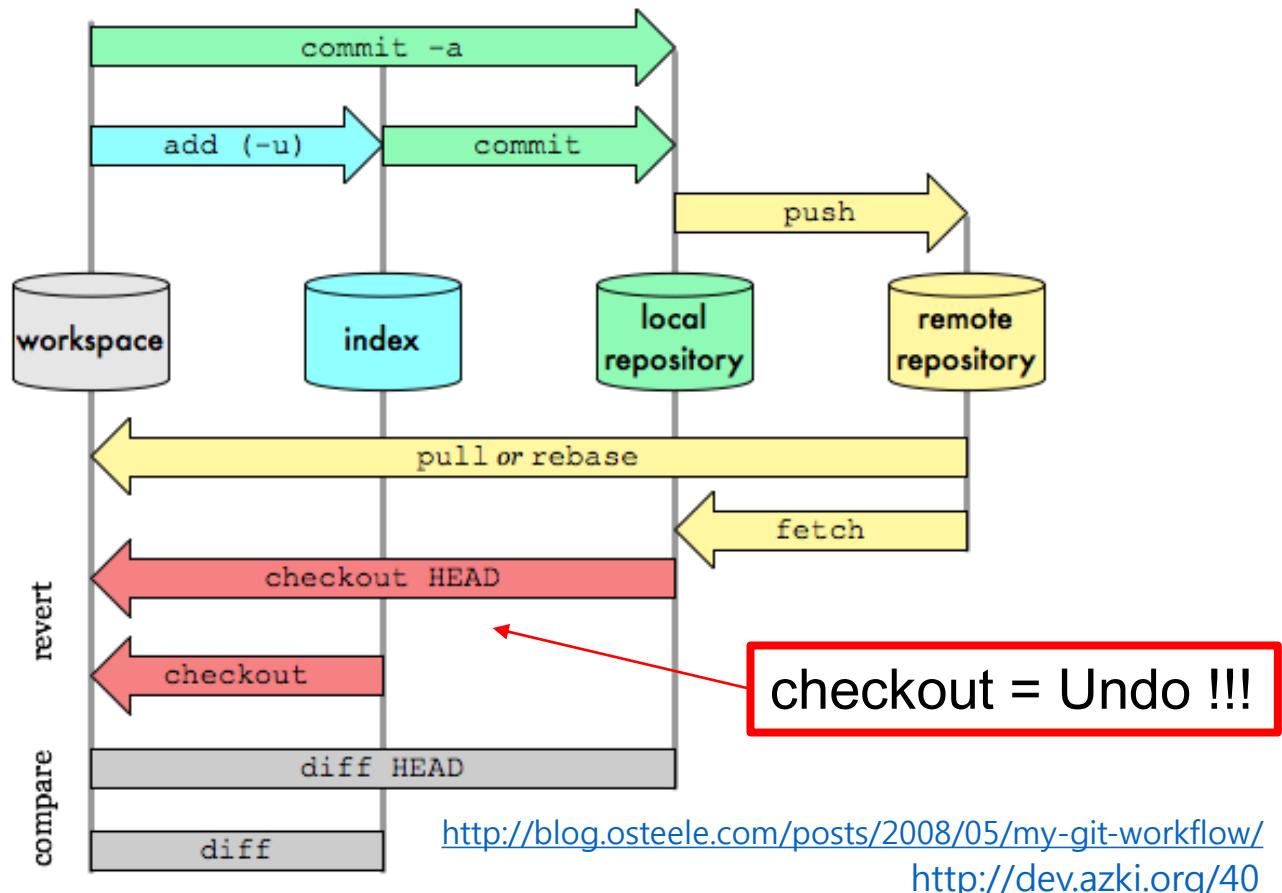
→ 'git add'한 모든 결과를 한번에 local repository로 commit

\$ git push origin master → github id, password 물어봄

→ local repository를 github의 master로 저장

Git Data Transport Commands

<http://osteale.com>



Clone & push (1/3)

```
/Users/minsuk/temp $ ls -la  
total 0  
drwxr-xr-x  2 minsuk  staff   68  6 15 21:46 .  
drwxr-xr-x+ 24 minsuk  staff  816  6 15 21:10 ..  
/Users/minsuk/temp $ git clone https://github.com/hl1itj/test-repo.git
```

Cloning into 'test-repo'...

remote: Counting objects: 18, done.

remote: Compressing objects: 100% (12/12), done.

remote: Total 18 (delta 6), reused 17 (delta 5), pack-reused 0

Unpacking objects: 100% (18/18), done.

Checking connectivity... done.

```
/Users/minsuk/temp $
```

```
/Users/minsuk/temp $ ls
```

test-repo

```
/Users/minsuk/temp $
```

```
/Users/minsuk/temp $ cd test-repo/
```

```
/Users/minsuk/temp/test-repo $
```

```
/Users/minsuk/temp/test-repo $ ls -la
```

total 24

```
drwxr-xr-x  6 minsuk  staff   204  6 15 21:47 .
```

```
drwxr-xr-x  3 minsuk  staff   102  6 15 21:47 ..
```

```
drwxr-xr-x 13 minsuk  staff   442  6 15 21:47 .git
```

```
-rw-r--r--  1 minsuk  staff   246  6 15 21:47 .gitignore
```

```
-rw-r--r--  1 minsuk  staff  1099  6 15 21:47 LICENSE
```

```
-rw-r--r--  1 minsuk  staff  3536  6 15 21:47 README.md
```

clone:
ID 질문 없이
가져옴

.git directory가
local index & repository

github에서 온 것

Clone & push (2/3)

```
/Users/minsuk/temp/test-repo $ cat > hello.c
main()
{
    printf("Hello World!\n");
}
```

hello.c 만들기

index에 등록

```
/Users/minsuk/temp/test-repo $ git add hello.c
```

```
/Users/minsuk/temp/test-repo $ git commit -m "add hello.c"
```

```
[master 641ae0b] add hello.c
```

1 file changed, 4 insertions(+)

create mode 100644 hello.c

local repository에
commit

```
/Users/minsuk/temp/test-repo $ git push origin master
```

```
Username for 'https://github.com': hl1itj
```

```
Password for 'https://hl1itj@github.com':
```

```
Counting objects: 3, done.
```

```
Delta compression using up to 4 threads.
```

```
Compressing objects: 100% (2/2), done.
```

```
Writing objects: 100% (3/3), 307 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 1), reused 0 (delta 0)
```

```
To https://github.com/hl1itj/test-repo.git
```

```
 310fc7c..641ae0b master -> master
```

```
/Users/minsuk/temp/test-repo $
```

github repository에
push

Clone & push (3/3)

hl1itj / test-repo GITHUB의 test-repo

Watch 1 Star 0 Fork 1

Test Repository

7 commits 1 branch 0 releases 2 contributors

branch: master / +

add hello.c

hl1itj authored 11 minutes ago latest commit 641ae0b988

.gitignore Initial commit 23 hours ago

LICENSE 22 hours ago

README.md Update README.md 12 hours ago

hello.c add hello.c 11 minutes ago

git push로 넘어온 hello.c

HTTPS clone URL
<https://github.com/>

You can clone with [HTTPS](#) or [Subversion](#).

Clone in Desktop Download ZIP

QUICK MARKDOWN EXAMPLE - touched by minsuklee

copied from <http://www.unexpected-vortices.com/sw/rippledoc/quick-markdown-example.html>

Git의 undo (1/4) : add 전에 undo 하기

```
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello World!\n");
}

/Users/minsuk/temp/test-repo $ vi hello.c
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello MERS!\n");
}

/Users/minsuk/temp/test-repo $ git checkout -- hello.c UNDO!!
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello World!\n");
}

/Users/minsuk/temp/test-repo $
```

hello.c 원래꺼

hello.c 수정본

hello.c 원래꺼로
복원 !!!

Git의 undo (2/4) : add 후에 undo 하기

```
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello MERS!\n");
}
/Users/minsuk/temp/test-repo $ git add hello.c ADD
/Users/minsuk/temp/test-repo $ git reset HEAD hello.c
Unstaged changes after reset:
M      hello.c
/Users/minsuk/temp/test-repo $ git checkout -- hello.c
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello World!\n");
}
/Users/minsuk/temp/test-repo $
```

hello.c 수정본

ADD

UNDO!!

hello.c 원래 꺼로 복원 !!!

Git의 undo (3/4) : commit 후에 undo 하기

```
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello MERS!\n");
}
/Users/minsuk/temp/test-repo $ git commit -a -m "test MERS"
[master 1e71c64] test MERS
 1 file changed, 1 insertion(+), 1 deletion(-)
/Users/minsuk/temp/test-repo $ git reset --hard HEAD~1 UNDO!!
HEAD is now at 1c6ed00 gitignore update
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello World!\n");
}
/Users/minsuk/temp/test-repo $
```

hello.c 수정본

add와 commit을 동시에

git reset --hard HEAD~1 UNDO!!

hello.c 원래 꺼로
복원 !!!

Git의 undo (4/4) : push 후에 undo 하기|-1/3

add와 commit을 동시에

```
/Users/minsuk/temp/test-repo $ vi hello.c
/Users/minsuk/temp/test-repo $ git commit -a -m "test MERS"
[master 167fc6a] test MERS
 1 file changed, 1 insertion(+), 1 deletion(-)
/Users/minsuk/temp/test-repo $ git push origin master PUSH
Counting objects: 6, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 629 bytes | 0 bytes/s, done.
Total 6 (delta 2), reused 0 (delta 0)
To https://github.com/hl1itj/test-repo.git
 641ae0b..167fc6a  master -> master
/Users/minsuk/temp/test-repo $
```

Git의 undo (4/4) : push 후에 undo 하기 - 2/4

Test Repository

9 Commits

 9 commits	 1 branch	 0 releases	 2 contributors		
  branch: master 					
test MERS					
 hl1itj authored 7 minutes ago	latest commit 167fc6a3ed 				
 .gitignore	gitignore update	an hour ago			
 LICENSE	add author, etc	11 days ago			
 README.md	Update README.md	11 days ago			
 hello.c	test MERS	7 minutes ago			
Push 결과, 바뀜					
 README.md					
QUICK MARKDOWN EXAMPLE - touched by minsuklee					

Git의 undo (4/4) : push 후에 undo 하기|-3/4

```
/Users/minsuk/temp/test-repo $ git reset --hard HEAD~1 UNDO (local)
HEAD is now at 1c6ed00 gitignore update
/PUSH
To https://github.com/hl1itj/test-repo.git
! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/hl1itj/test-repo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
/PUSH 억지로 PUSH
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/hl1itj/test-repo.git
 + 167fc6a...1c6ed00 master -> master (forced update)
/PUSH
```

Git의 undo (4/4) : push 후에 undo 하기|-4/4

Test Repository

8 Commits

8 commits			1 branch	0 releases	2 contributors
gitignore update	hl1itj authored an hour ago	latest commit 1c6ed0079b	branch: master	test-repo / +	≡
.gitignore	gitignore update	an hour ago			
LICENSE	add author, etc	11 days ago			
README.md	Update README.md	11 days ago			
hello.c	add hello.c	10 days ago			
hello.c 원복					
README.md					
QUICK MARKDOWN EXAMPLE - touched by minsuklee					

GIT = UNDO button

```
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
    printf("Hello World!\n");
}
```

생활 코딩의 git 강좌

<https://opentutorials.org/course/1492>

YouTube KR ≡ 🔍 업로드 로그인

버전 관리 시스템 Version Control System

0:58 / 14:17

Git : 버전관리란 무엇일까요? 생활코딩

2 Git : Git과 SourceTree설치 - 윈도우 생활코딩

3 Git : Git과 SourceTree 설치 - OSX 생활코딩

4 예제 설명 Git 생활코딩

5 저장소 만들기 Git 생활코딩

6,959

저장소와 버전 만들기 Git 12:42

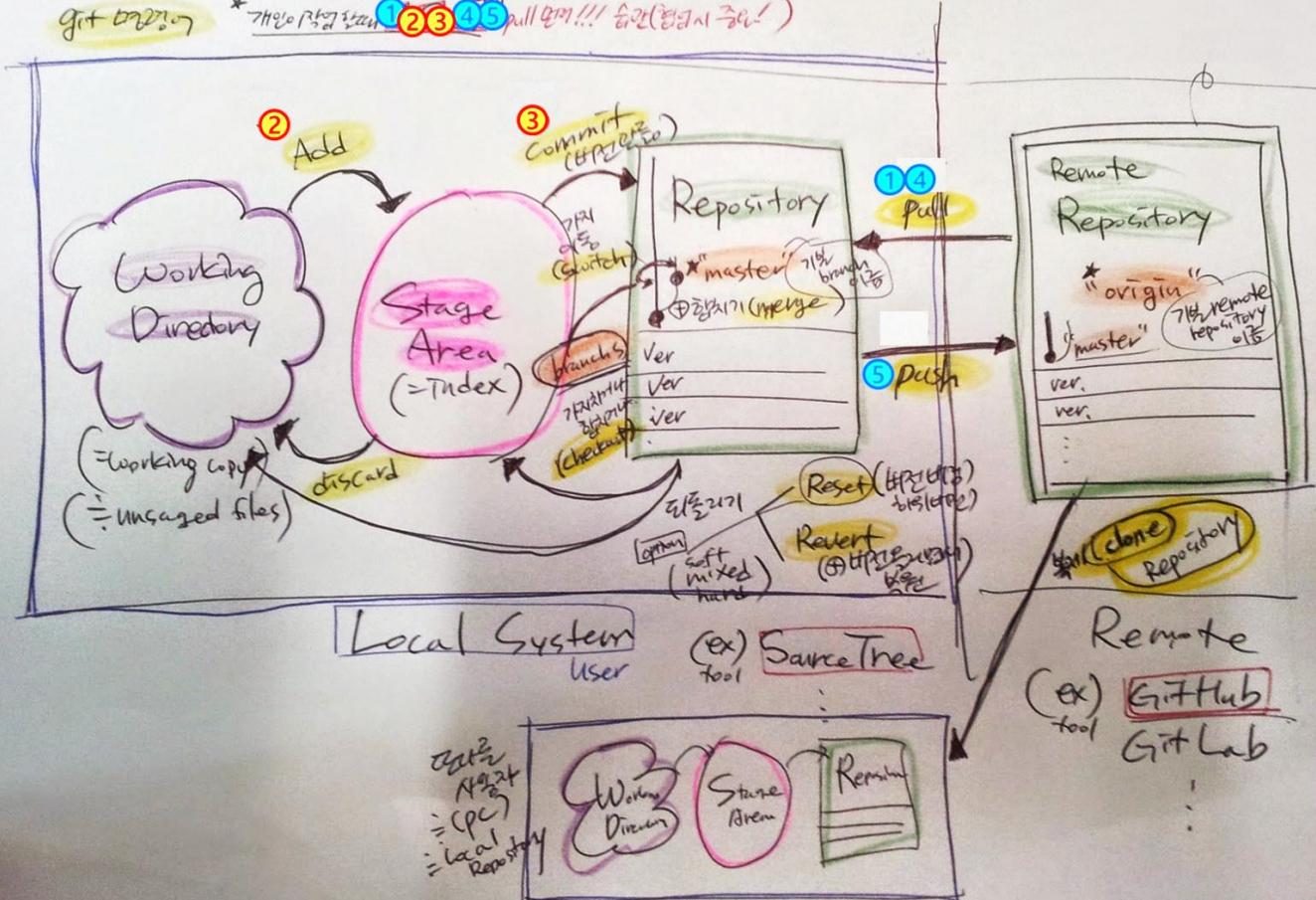
PHP란무엇인가 서버와클라이언트 서버와 클라이언트 생활코딩

<https://www.youtube.com/watch?v=XUEuYq64HKI&list=PLuHgQVnccGMCB06JE7zFIAOJtdcZBVrap>

Git (By FunnyRella)

<http://funnyrella.blogspot.kr/2014/04/97-git.html>

git 데모본
* 1 2 3 4 5 pull 했어!!! 끌었지 끌어!



Git의 branch

- Branch는 Sub directory나 그냥 복사본이 아니다...
결국은 master에 포함될 가능성이 있는 내용을 임시로
가지고 있는 놀이터..
(원본 소스와 격리된 상태에서 작업)
- 혼자 개발할 때는 별 쏘를 해도 상관 없다.
(undo를 할 수 있으므로)
- 여럿이 작업할 때는 main line(master branch)를 깨끗하게
유지해야 한다.
- 그래서 작업을 할 때는..
Branch를 만들어, 지지고 볶고 작업을 하고
할 수 있는 모든 Test를 다해서 확신이 생긴 뒤에
master branch에 merge

여러 버전으로 가지치기, branching (1/2)

1. 새로운 저장소를 만들면 기본으로 **master** 가지가 만들어짐
뭔가 정말 최소한의 의미를 가진 버전이 나올 때 까지는
여기서 작업...
2. Master는 두고 뭔가 개정(기능 추가 등)을 하려면..
3. "<branch_name>" 이라는 이름의 가지를 만들고 거기서 작업

```
$ git checkout -b <branch_name> (local에 생김)
```
4. 개발을 진행해서 commit를 하고...

```
$ git push origin <branch_name> (필요하면)
```

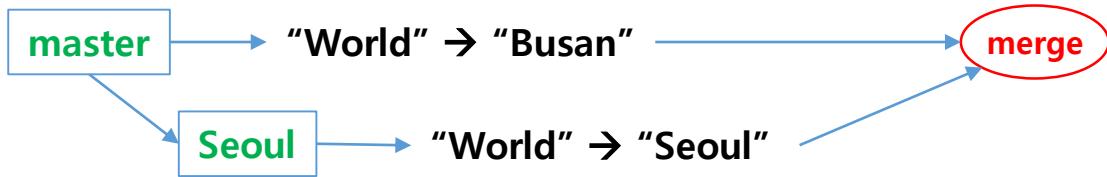
하면 새로운 <branch_name> 브랜치가 서버(github)에 생김
(즉, 이제서야 다른 사람에게도 보임)

여러 버전으로 가지치기, branching (2/2)

5. 브랜치에서의 작업이 완료되면 local의 master 가지로 돌아와
 \$ git checkout master
 6. <branch_name> 브랜치 (local에 있는) 내용을 Merge
 \$ git merge <branch_name>
- 여러 명이 같이 작업을 할 때는
충돌이 날 수도 있음

여러가 나고 해당 파일을 열면
어떤 문제가 있었는지 알게 됨..
7. Merge 된 결과를 repository master에 저장
 \$ git push origin <branch_name>
 8. 브랜치 지우기
 \$ git push origin --delete <branch_name> (github에서 삭제)
 \$ git branch -d <branch_name> (local의 branch 삭제)

Merge 에서의 충돌 상황



```
/Users/minsuk/temp/test-repo $ git merge Seoul
Auto-merging hello.c
CONFLICT (content): Merge conflict in hello.c
Automatic merge failed; fix conflicts and then commit the result.
```

```
/Users/minsuk/temp/test-repo $ cat hello.c
main()
{
<<<<< HEAD
    printf("Hello Busan!\n");
=====
    printf("Hello Seoul!\n");
>>>>> Seoul
}
```

hello.c 에 문제를 적어줌
(HEAD는 master branch의
최신 버전을 의미)
→ 수동으로 해결 !!!

Code Review

모든 창작 활동은
리뷰로 완성된다.

코드 리뷰는

- ✓ 개발 프로세스의 일부
- ✓ 잘못 찾기가 아닌, 잘 만들기 위한 협력
- ✓ 편안하고 자발적인 환경이 중요

Code Review

- 나 말고 다른 사람이 한번 더 보면 좋지 않겠는가?
 - 오류를 일찍 발견하여 디버깅 시간을 단축
- 내 코드가 아닌 우리의 코드로
 - 코딩 표준을 잘 지킨, 읽기 쉽고 품질 좋은 코드
- 신입 개발자에 대한 배려
 - 릴리즈 되어 망가지기 전에, 실수로부터 배움
- 개발자들 사이에 신뢰 관계 형성
 - 더 중요한 협력의 준비

코드 리뷰는

- ✓ 시간이 들지만, 아깝지 않다.
- ✓ 규칙적으로, 병렬적으로..

Code Review

- 모든 수정된 내용은 바로 리뷰를 받는 걸로

- 왕창 수정된 내용이라면 리뷰하기도 / 나중에 수정하기도 힘들겠죠

- 대개는 수정한 사람이 리뷰어를(들을) 지정

- 충분히 복잡하고 다양한 관점이 필요하면 여러명이 리뷰

- 리뷰어는 수정된 내용을 검토

- 어떻게 코드를 개선할 수 있을까에 대한 코멘트

- (모든 사항, 코딩 표준, Naming, 알고리즘)

- 필요하면 일부는 직접 수정

- 개발자는 리뷰 결과를 바탕으로

- 수정, 개선 / 테스트

- 그리고 다시 OK 될 때까지 리뷰 요청

코드 리뷰는

✓ 모두가 만족할 때까지

Code Review 해본 적이 없으시다면

- 처음엔 종이로 뽑아서, 한 줄씩 읽는 것으로 리뷰
 - 절대 화면을 보면서, 키보드에 손을 올리고 하면 안됨
 - 소스 라인 위에 잘못된 점, 더 좋은 제안을 적으면서 진행
 - 리뷰가 끝난 뒤, 한 번에 수정 및 개선
 - Every Detail을 리뷰
 - 코딩 표준, Naming, 알고리즘, 코멘트, 스펠링, ...
 - 내공이 쌓이면
 - 화면을 봐도 수정할 것이 보임
 - gerrit과 같은 도구 사용
 - 또는 Issue Tracking System을 이용
 - 리뷰 결과는 결국, 수정을 요하는 것이므로
- 그냥 github (또는 Bugzilla) 이용.

코드 리뷰는

✓ 우선, 하는 것이 중요

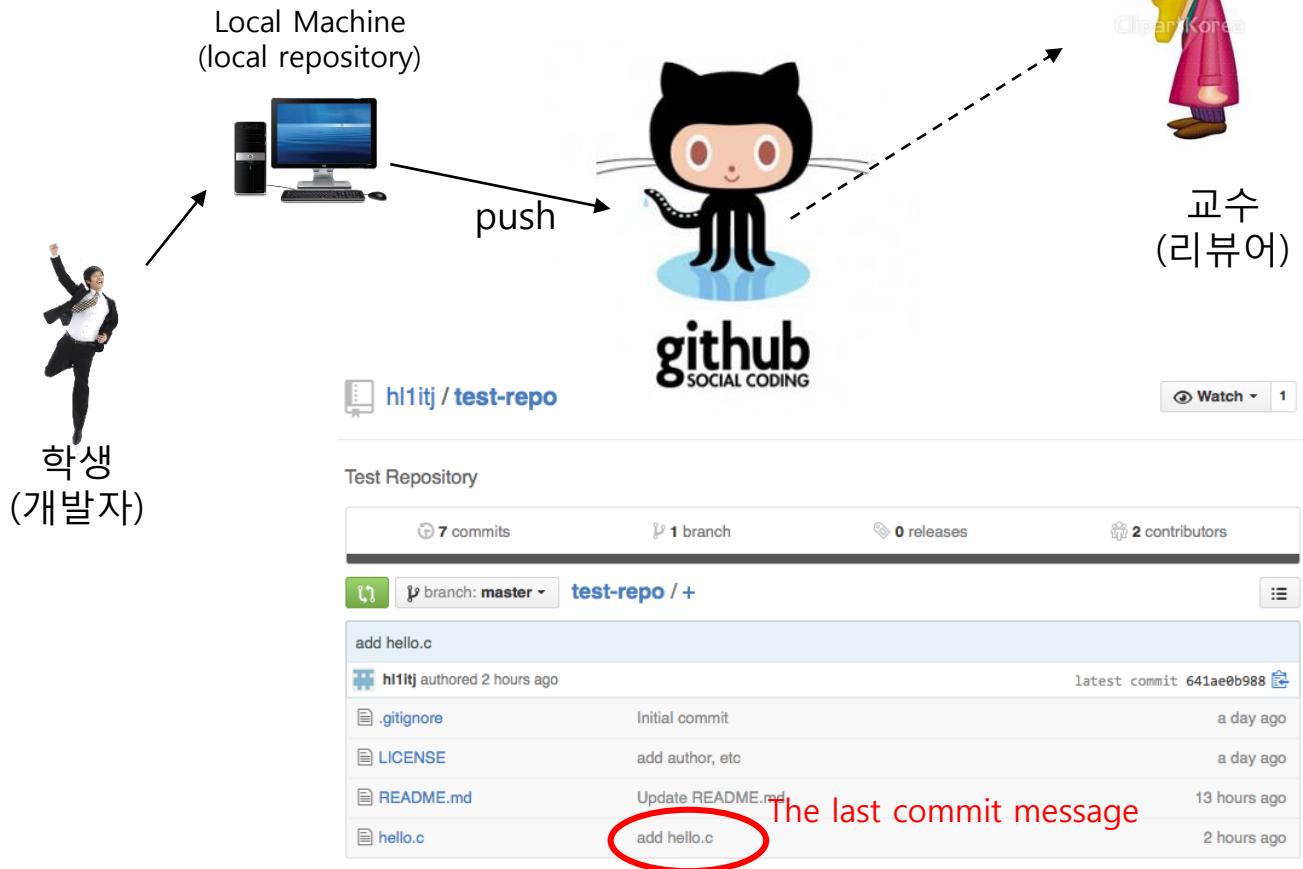
우리의 hello.c 를 Review 해보자.

```
/Users/minsuk/temp/test-repo $ cat hello.c
```

```
main()
{
    printf("Hello World!\n");
}
```

```
/Users/minsuk/temp/test-repo $ gcc hello.c
hello.c:1:1: warning: type specifier missing, defaults to 'int' [-Wimplicit-int]
main()
^
hello.c:3:4: warning: implicitly declaring library function 'printf' with type
    'int (const char *, ...)'
    printf("Hello World!\n");
^
hello.c:3:4: note: include the header <stdio.h> or explicitly provide a
      declaration for 'printf'
2 warnings generated.
/Users/minsuk/temp/test-repo $
```

Review상황 ...



hello.c commit log로 이동

- Last commit 메시지 클릭하거나
파일의 History에서 모든 commit에 접근 가능

The screenshot shows a GitHub commit page for the file 'hello.c'. At the top, it says 'add hello.c' and 'master'. Below that, it shows a commit by 'hl1itj' from 2 hours ago. The commit message is 'add hello.c'. It has 1 parent commit (310fc7c) and a commit hash (641ae0b988553cf53884218d3bdb23ceb1228f10). A red box highlights the commit hash. Below the commit details, it says 'Showing 1 changed file with 4 additions and 0 deletions.' On the right, there are 'Unified' and 'Split' buttons. The main area shows the diff for 'hello.c':
4 4 hello.c
... ... @@ -0,0 +1,4 @@
1 +main()
2 +{
3 + printf("Hello World!\n");
4 +}
A red arrow points from the text '최초로 hello.c 가 등록된 상황 (' '+' 만 있음)' to the '+' sign in the diff.
At the bottom, there's a comment section with 'Write' and 'Preview' buttons, a Markdown supported indicator, and an 'Edit in fullscreen' button. A red arrow points from the text '이 commit 자체에 대한 Comment를 기록할 수 있음' to the 'Leave a comment' input field.

Commit 번호

Commit 641ae0b988553cf53884218d3bdb23ceb1228f10

Showing 1 changed file with 4 additions and 0 deletions.

Unified Split

View

4 4 hello.c

... ... @@ -0,0 +1,4 @@

1 +main()
2 +{
3 + printf("Hello World!\n");
4 +}

최초로 hello.c 가 등록된 상황 ('+' 만 있음)

Leave a comment

Write Preview Markdown supported Edit in fullscreen

Comment on this commit

Attach images by dragging & dropping or selecting them.

이 commit 자체에 대한 Comment를 기록할 수 있음

Comment 달기

The screenshot shows a code editor interface with a red callout box highlighting a feature for adding comments. A red arrow points from the text in the callout box to a blue plus sign icon located on the line number 1 of the code editor.

라인 번호 쪽으로 마우스가 가면
“+” 표시가 나오고 클릭하면
Comment 창이 나옴

4 hello.c

... ... @@ -0,0 +1,4 @@

1 +main()

Write Preview Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping or selecting them.

Cancel Comment

2	+{
3	+ printf("Hello World!\n");
4	+}

꼼꼼하게 comment ...

4 hello.c Show notes View

```
... ... @@ -0,0 +1,4 @@
1 +main()
```

minsuklee added a note 4 minutes ago

main 앞에는 기본으로 int 타입 선언이 필요
int main()

Add a line note

```
2 +
3 + printf("Hello World!\n");
```

minsuklee added a note 3 minutes ago

printf()를 사용하여면 #include <stdio.h> 가 필요

←
Markdown 언어 !
₩<stdio.h₩>

Add a line note

```
4 +}
```

minsuklee added a note 2 minutes ago

int main()를 썼다면 integer를 return
return 0; 추가 !!

Unsubscribe를 하지 않으면
Comment를 달 때마다
메일이 하나씩 감 !!

Add a line note

Review 결과를 Issue로 등록 (1/3)

hl1itj / test-repo

Watch 1 Star 0 Fork 1

add hello.c
master

hl1itj authored 2 hours ago 1 parent 310fc7c commit 641ae0b988553cf53884218d3bdb23ceb1228f10

Showing 1 changed file with 4 additions and 0 deletions.

Unified Split

4 hello.c

... ... @@ -0,0 +1,4 @@
1 +main()

minsuklee added a note 15 minutes ago
main 앞에는 기본으로 int 타입 선언이 필요
int main()

Add a line note

2 +{
3 + printf("Hello World!\n");

minsuklee added a note 15 minutes ago

Browse files

Unified Split

Unified Split

1. Commit 번호를 복사한 뒤에
2. Issue 버튼을 눌러
3. 이 commit 로그의 코멘트
(즉, 리뷰 결과)를 author가
볼 수 있도록
4. Issue로 등록

Review 결과를 Issue로 등록 (2/3)

hl1itj / test-repo Watch

Issues Pull requests Labels Milestones

Review 결과

Write Preview Markdown supported Edit in fullscreen

다음 내용 검토 후 수정 바람
641ae0b988553cf53884218d3bdb23ceb1228f10

copy 했던
commit 번호
Paste

Attach images by dragging & dropping or selecting them.

Submit new issue

Review 결과를 Issue로 등록 (3/3)

hl1itj / test-repo

Watch 1 Star 0 Fork 1

Review 결과 #4

Open minsuklee opened this issue 12 seconds ago · 0 comments

minsuklee commented 12 seconds ago

다음 내용 검토 후 수정 바람
641ae0b

Labels
None yet

Milestone
No milestone

Assignee
No one assigned

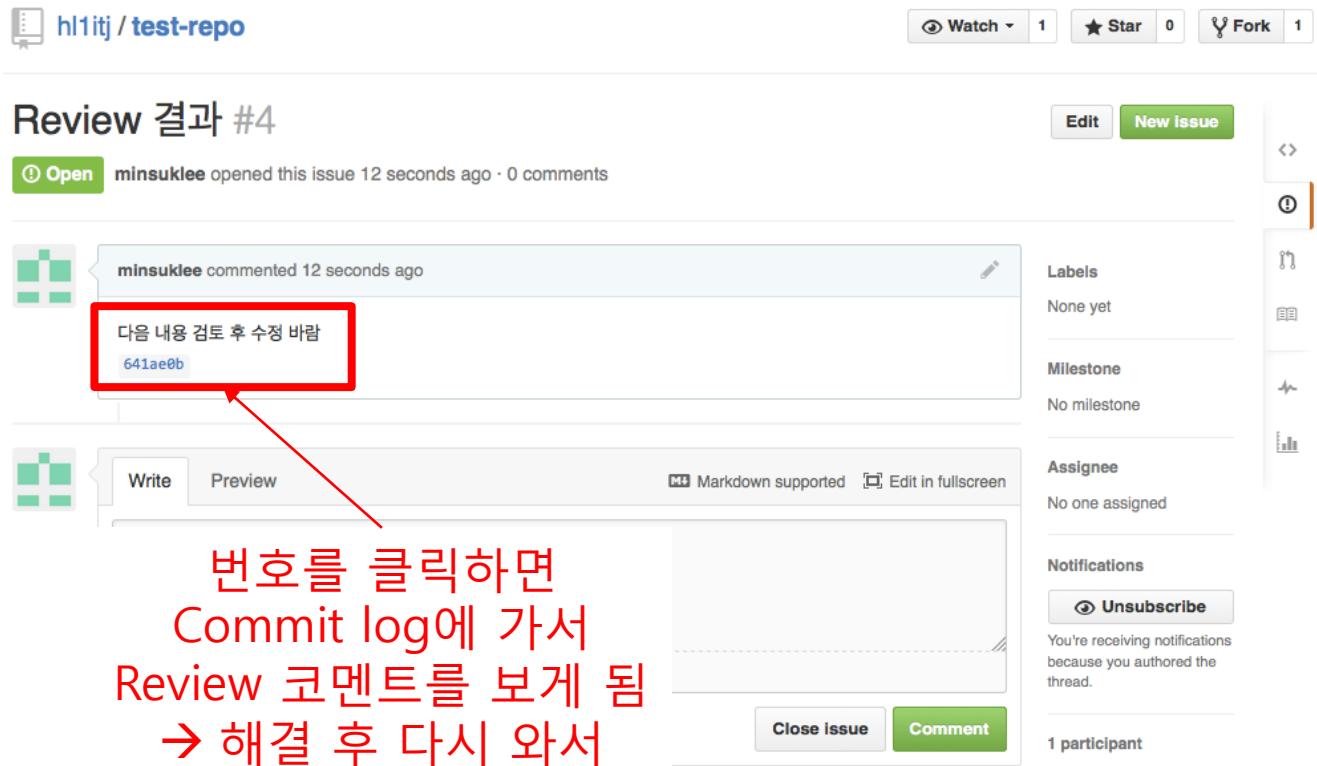
Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

Close issue Comment

번호를 클릭하면 Commit log에 가서 Review 코멘트를 보게 됨
→ 해결 후 다시 와서 issue close



Github만 가지고 코드 리뷰: 요약

- Commit log를 확인하여 line by line 으로 리뷰 (앞의 예):
 - Review 결과는 commit log 코멘트에 기록
 - Commit log 번호를 이슈에 기록
 - Author는 comment를 달 때, issue 등록할 때 메일을 받음
(안 받는다 해도 github의 issue로 등록되어 있음)
 - Author는 이슈를 확인하고, 이슈에 매달린 commit log의 review comment를 보고 소스 수정
- 그대로 소스에 리뷰 코멘트를 다는 방법:
 - 리뷰 대상 소스를 소스를 열어 편집하려 하면 fork하라고 나옴.
 - /* minsuk : comment */ 등으로 표시를 하여
 - review 코멘트를 적거나 필요한 수정을 한 뒤,
 - Pull request를 날림
 - Author는 자신의 Repository에서 Pull Request를 보게 되고
 - 소스가 어떻게 변경되었는지 확인 (즉 review 코멘트를 확인)
 - Merge 후 /* minsuk: .. */를 찾아 수정

유용한 Link

- [Github 실습](#) ←
- [Github의 git 15분 만에 배우기 Tutorial](#)
- [git-치트시트.pdf](#)
- [git을 시작하기 위한 간편 안내서](#)
- [Progit \(책\)](#) ←
- [Git 브랜치 배우기 \(online 따라하기\)](#)
- [Git 사용자 설명서 \(무려 김남형씨가 쓴\)](#) ←
- [누구나 쉽게 이해할 수 있는 git 입문](#)
- [SublimeText와 git 연동](#)
- [생활 코딩의 git 강좌 \(source tree 기반\)](#)
 - <https://opentutorials.org/course/1492> ←

slideshare.net 에 보면 엄청난 자료 !!

소프트웨어 교육을 위한 Good OSS Practice

- Linux/Eclipse/vi/gcc 사용 !!!
 - Virtual machine 활용 (virtual box + Ubuntu/CentOS/... !)
 - Oracle → Cubrid, MySQL, ...
 - Framework 사용 경험 (Spring, RoR, node.js, ...), 마치 게임 엔진 써 보듯이.
- github (또는 다른 OSS repository)의 적극적 활용
 - 샘플 코드를 github에서 fork/clone하도록
 - 과제 소스를 github의 자기 repository에 push 하도록
 - Detecting Software Plagiarism
 - <http://theory.stanford.edu/~aiken/moss/>
 - 모든 Project는 반드시 repository, issue tracking 사용하도록
- Code Review
 - 교수에 의한 review – learning 관점의 review
 - 수업 시간에 대표적인 코드 몇 개를 review
 - 동료/선배/조교에 의한 review (cosmetic review 포함)

최종본 슬라이드 PPT file과 모든 링크가 정리되어 있는 곳

<http://hl1itj.tistory.com/118>

Minsuk Lee

컴퓨터공학부.국민대학교

<https://www.facebook.com/minsuk.lee0>



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건변경허락 2.0
대한민국 라이선스]에 따라 이용할 수 있습니다'