

# 오픈소스와 GitLab 기반 LMS 시스템

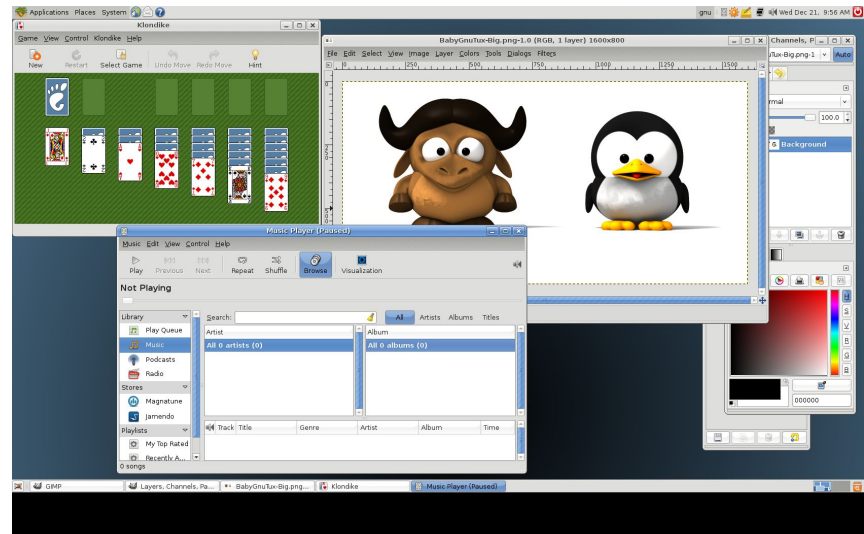
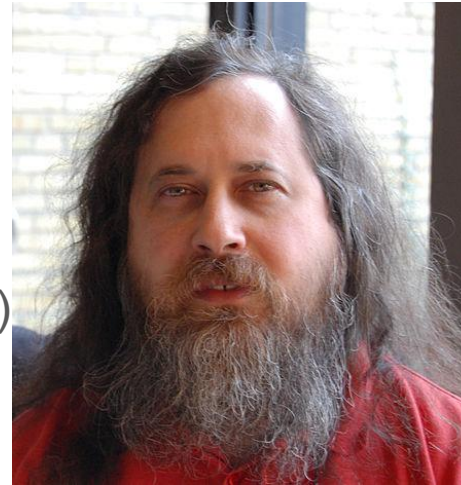
하 석 재  
sjha72@hanyang.ac.kr

# 목차

- 오픈소스의 간략한 역사
- 오픈소스의 현황
- 소프트웨어 협업도구의 소개
  - 소스버전관리/이슈관리/위키/지속적통합/프로젝트 품질관리기술
- Git의 소개
- GitLab의 소개
- GitLab기반의 LMS 사용법 소개

# 오픈소스의 시작

- GNU(GNU is Not Unix)
  - <http://gnu.org>
- FSF(Free Software Foundation)
- 리처드 스톨만(Richard Stallman)
- 성당과 시장모델(Cathedral vs. Market)
- Copyright vs Copyleft
- **GPL(General Public License)**
  - 상속률
  - GPL S/W를 사용한 S/W는
  - 자동으로 GPL라이센스가 됨
  - cf. LGPL(Lesser GPL)
- GNU 프로젝트들
  - Bash, Emacs
  - GCC/GDB, Gzip
  - CVS(?)



# 오픈소스의 발전

- 리눅스(GNU/Linux)
  - <https://www.kernel.org/>
  - Minix 기반
  - 라이너스 토발즈(Linus Torvalds)
  - 리눅스커널과 **Git** 개발
- 대표적인 오픈소스 OS
  - 안드로이드OS 커널도 리눅스
- 데스크탑 OS보다는 서버/모바일 OS시장 장악
- 커널만 동일하고 나머지는 선택
- 패키징(Packaging)
  - RedHat계열 vs. Debian계열
  - Fedora/RHEL/CentOS vs. Debian/Ubuntu
  - apt vs. yum



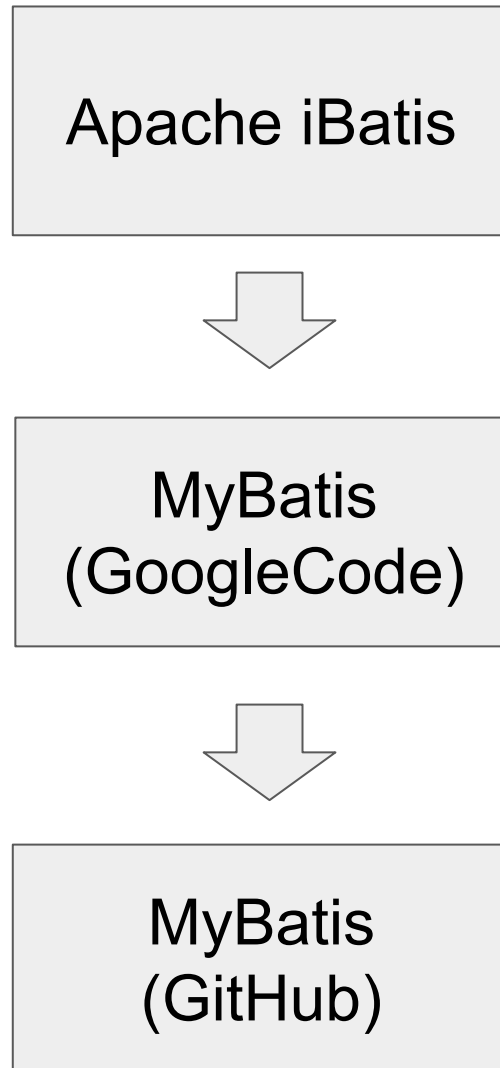
# 안드로이드 구조(참조)



# 오픈소스 프로젝트 현황

- 아파치 재단(Apache Foundation)
  - <http://www.apache.org>
  - 아파치(웹서버)
  - 자카르타 프로젝트(Jarkarta Project)
  - 톰캣(Tomcat), 메이븐(Maven), HBase, 하둡(Hadoop), 스파크(Spark), 루씬(Lucene), 주키퍼(Zookeeper), ...
- 구글코드(GoogleCode)
  - 현재는 서비스 접음
- 깃허브(GitHub)
  - <http://github.com>
  - 마이바티스(MyBatis), Angular.js/구글, 텐서플로우(Tensorflow)/구글
  - 도커(Docker), D3.js, Node.js, JQuery.js, React.js
  - 스프링(Spring Framework)/Pivotal, 엘라스틱서치(Elasticsearch)/Elastic, 웹킷(Webkit)/Apple, ...

# 세 진영을 모두 거친 오픈소스 프로젝트



# 오픈소스로 돈을 버는 방법?

- 개인
  - 일종의 재능기부
  - 명성(Reputation)을 쌓아 보상받음
  - 인맥
- 재단
  - 기부금을 받아 다시 재투자
  - 아파치 재단(Apache Foundation)이 대표적
- 회사
  - 소스나 프로그램 사용대가는 안 받지만
  - 유지보수, 지원대가는 있음
  - 요즘에는 오픈소스버전과 유료버전을 분리
  - Fedora, CentOS / RHEL(Redhat Linux Enterprise Edition)
  - MySQL Enterprise, CGE
  - GitLab Comunity Edition / Enterprise Edtion



소프트웨어 협업 도구

# 배경

- 프로그램 개발방법의 분화/전문화
- 대형화
  - 혼자 개발하는 경우가 잘 없음 -> 여러 명이 작업을 동시에 진행
- 의사소통 및 공동개발이 일반화됨
- 필요한 협업툴들이 진화/발달

# 소프트웨어 협업도구

- 소스 버전 컨트롤(Source Version Control)
- 이슈 트래킹 시스템(Issue Tracking System)
- 위키(Wiki)
- 지속적 통합(CI:Continuous Integration)
- 프로젝트 품질관리
- 그 외
  - 슬랙(Slack)
    - 개발자 전용 메신저
  - 트렐로(Trello)
    - 협업도구

# 소스 버전 관리(Source Version Control)

- 소프트웨어 형상관리(Software Configuration Management)
- 충돌(collision) 관리
  - **Optimistic lock** vs. Pessimistic lock
  - 충돌이 나면 직접 해결 vs. 원천 배제(락)
- CVS/**SVN**(subversion)
  - diff에 기반
  - 커밋을 할 경우는 계속 **revision**이 증가(통합 번호 관리)
  - **SVN**은 텍스트 외에 바이너리 포맷도 지원
- 저장소(Repository)에 저장(Commit)/체크아웃(Checkout)
  - cf. 거북이 **SVN**(Tortoise SVN)은 로컬 저장소/원격 저장소 동시지원
- 메인/마스터(Main/Master), 브랜치(Branch), 리비전(Revision)

# 충돌(Collision)이란

- 여러 명이 한 팀을 이뤄 프로젝트를 진행하는 경우
- 동일한 파일을 여러 명이 편집하면서 한 명이 변경사항을 수시로 업데이트 하지 않고 계속 개발 진행할 경우 업데이트를 시도할 때 충돌이 발생할 가능성이 있음
- 원격 저장소에 저장된 버전과 **push**하는 버전 간의 최신 버전 지정을 직접 해결 (Resolve)
- 요즘은 서버에서 파일 업데이트가 일어나면 해당 이벤트를 알려주는 프로그램/기술이 등장해 빈도가 낮아짐
- **개인별 프로젝트를 제출할 경우는 충돌문제 걱정할 필요없음**
- **팀별 과제를 진행할 경우는 해당 팀에서 해결할 문제(관리문제는 없음)**

# (예) 버전컨트롤(Blame)

The screenshot shows a code editor window titled "Annotate - intelhex/\_init\_.py". The left sidebar displays a list of revisions with their authors and commit messages. The main area shows the code with line numbers and the author of each line. The bottom right panel shows details for the selected revision (168.1.1 by Bernhard Leiner).

Rev	Message	Author
194	bump version to 1.4.1	Alexander Belchenko
190	update year in copyright string	Alexander Belchenko
188	bug #988148 IntelHex16bit should ...	Alexander Belchenko
182	merge python 3 compatibility fixes.	Alexander Belchenko
168.1.1	python 3 compatibility changes	Bernhard Leiner
181	IntelHex.frombytes: helper meth...	Alexander Belchenko
177	added docstring for diff_dumps f...	Alexander Belchenko
176	update versions in intelhex/*.py	Alexander Belchenko
172	diff_dumps: added function to co...	Alexander Belchenko
171	using print() for python 3.x com...	Alexander Belchenko
168	using shift instead of classic int divi...	Alexander Belchenko
167	avoid using dict.has_key() which is ...	Alexander Belchenko
156	size parameter is also adding to me...	Alexander Belchenko
155	bug #408748 tobinary: added siz...	Alexander Belchenko
153	hex2dump: better compatilbty with...	Alexander Belchenko

```
40 14 Alexander Belchenko
41 2 bialix
42
43 14 Alexander Belchenko
44 46 Alexander Belchenko
45
46 119 Alexander Belchenko
47
48 172 Alexander Belchenko
49 72 Alexander Belchenko
50 168.1.1 Bernhard Leiner
51 72 Alexander Belchenko
52
53
54 1 bialix
55
56 35 Alexander Belchenko
57 112.1.1 Zachary Clifford
58
59 31 Alexander Belchenko
60
61 112.1.1 Zachary Clifford
62
```

```
__docformat__ = "javadoc"

from array import array
from binascii import hexlify, unhexlify
from bisect import bisect_right
import os
import sys

from compat import asbytes, asstr

class IntelHex(object):
    ''' Intel HEX file reader. '''

    def __init__(self, source=None):
        ''' Constructor. If source specified, object will be initialized
        with the contents of source. Otherwise the object will be empty.

        @param source source for initialization
        (file name of HEX file, file object, addr dict or
        other IntelHex object)'''
```

**Revision: 168.1.1**  
revid:bleiner@gmail.com-20110409204414-0bz56m5j9f19foo6  
**Parents:** [168: using shift instead of classic int division for compatibil...](#)  
**Children:** [171.1.2: merge python 3 compatibility fixes from Bernhard Leiner](#)  
**Date:** 4/9/11 11:44 PM  
**Committer:** Bernhard Leiner <[bleiner@gmail.com](mailto:bleiner@gmail.com)>  
**Branch:** intelhex

python 3 compatibility changes

- add python3 build target in setup.py
- add and use asstr and asbytes compatibility function for python 3
- other compatibility changes


all tests pass now after running 2to3

# 이슈 트래킹 시스템(ITS:Issue Tracking System)

- 버그, 요구사항, 작업내용을 지시하고 진행사항을 관리하는 시스템
- 게시판과 유사한 형태
- 이슈의 종류/작업자/버전/우선순위를 지정
- 트랙(Trac)/레드마인(Redmine)/지라(JIRA)
- 보통은 소스버전관리 시스템과 연동해서 사용
- 사용형태(패턴)
  - 이슈를 등록(open)하고 작업자, 우선순위 지정(assign)
  - 작업자가 이를 수락(accept)하거나 다른 사람에게 위임(reassign)
  - 작업자가 작업을 끝내고 소스코드를 커밋하면서 이슈를 닫음(close)


# (예) 이슈관리

하석재 / sample



This project Search


1



Project Activity Pipelines **Issues 1** Merge Requests 0 Wiki

Issues Board Labels Milestones

Open 1 Closed 0 All 1

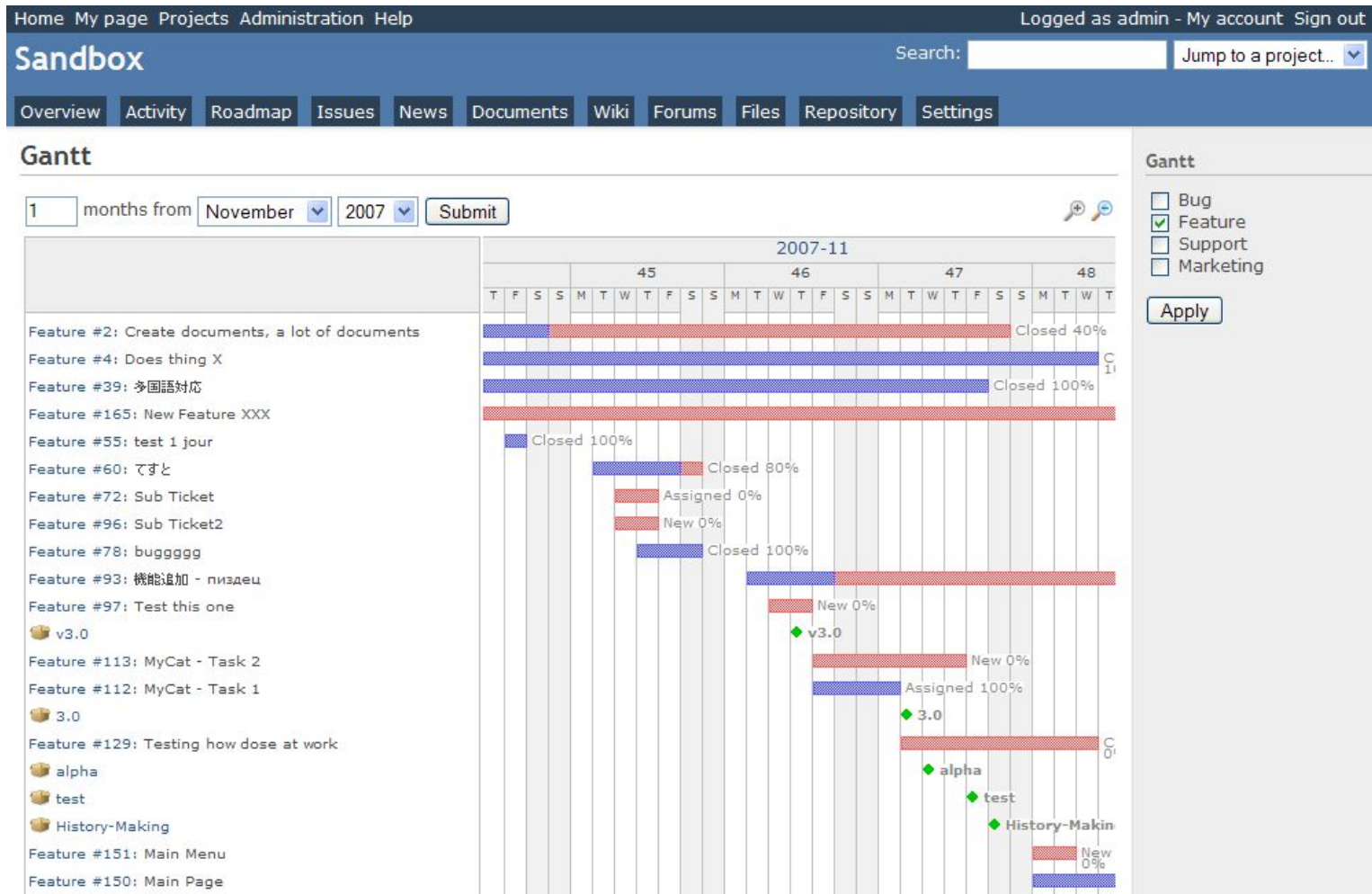
 [New Issue](#)

☐ Search or filter results... Last created

☐ 테스트 사이트 리뷰  
#1 · opened a week ago by 하석재 · 한양대 교수님 대상 세미나 준비 Feb 8, 2017 updated a week ago

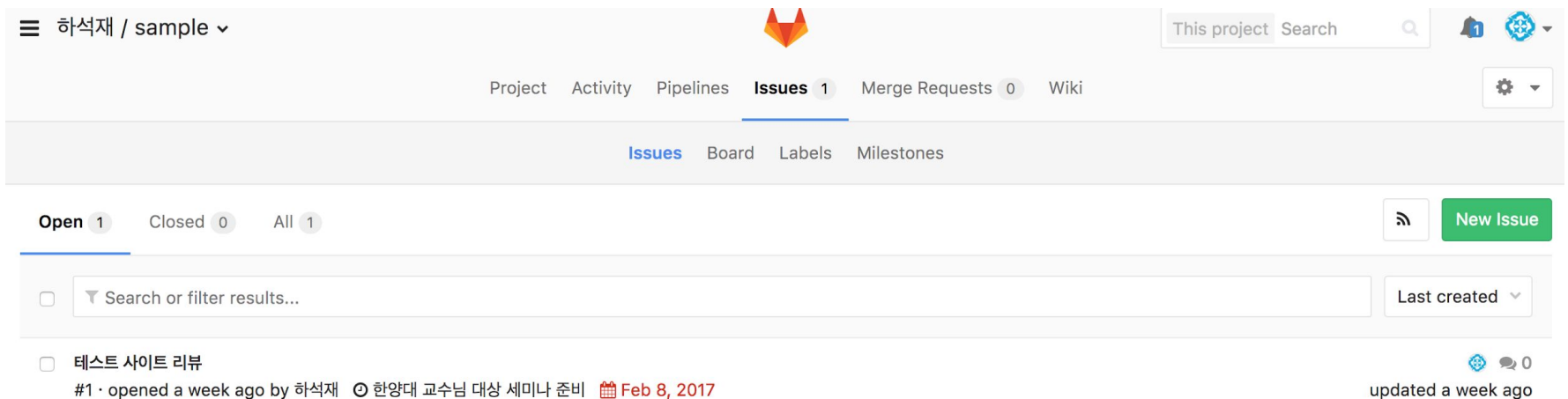


# (예) 레드마인



# Wiki

- 게시판과 같은 서비스를 제공
- **전용태그**를 사용해 페이지관리와 간단한 파일공유서비스 가능



The screenshot shows the GitHub interface for a project named 'sample' by user '하석재'. The top navigation bar includes 'Project', 'Activity', 'Pipelines', 'Issues' (with a count of 1), 'Merge Requests' (with a count of 0), and 'Wiki'. Below this, there are tabs for 'Issues', 'Board', 'Labels', and 'Milestones'. The 'Issues' tab is active, showing a list of issues. The first issue is titled '테스트 사이트 리뷰' (Review test site) and was opened a week ago by '하석재'. It has a due date of 'Feb 8, 2017'. The issue is currently open and has 0 comments. The interface also includes a search bar, a 'New Issue' button, and a 'Last created' dropdown menu.

하석재 / sample

This project Search

Project Activity Pipelines **Issues 1** Merge Requests 0 Wiki

Issues Board Labels Milestones

Open 1 Closed 0 All 1

New Issue

Search or filter results...

Last created

테스트 사이트 리뷰

#1 · opened a week ago by 하석재 · 한양대 교수님 대상 세미나 준비 Feb 8, 2017

updated a week ago

# CI(Continuous Integration)

- 지속적 통합
- 코드가 제출되면 일정시간마다 자동으로 빌드(Build)를 진행해 코드상의 문제를 자동으로 피드백할 수 있는 기술
- **젠킨스(Jenkins)**, GitLab CI

# (예) GitLab CI

하석재 / my-php-project

This project Search

Project Activity Repository Pipelines Graphs Issues 0 Merge Requests 0 Wiki

Pipelines Builds Environments Cycle Analytics

All 5 Running 0 Branches Tags

Run pipeline CI Lint

Status	Pipeline	Commit	Stages	
failed	#13 by  latest	master -> 8374142a test Hello.txt		2 days ago
passed	#12 by	master -> 9d72a896 first C++		00:04:42 1 week ago
passed	#11 by	master -> 075463f0 first python		00:03:32 1 week ago
passed	#10 by	master -> 3f95544d first C		00:05:12 1 week ago
failed	#9 by	master -> 959a1206 first java		00:03:49 1 week ago

# 소프트웨어/프로젝트 품질관리

- 코딩 수준 높이기
  - 코딩품질개선을 위한 정보제공
  - 진단
    - 리팩토링(Bad Smells)/취약점(Vulnerability)/버그(Bug)
    - 소스 중복, 복잡도, 테스트의 커버리지 및 잠재적인 버그 정도
  - 검출된 이슈는 판단 기준이 된 규칙의 세부 정보
  - 검출된 결과의 이슈 여부, 심각도 등을 판단
- 소나 큐브(Sonar Qube)

# (예) SonarQube

The screenshot shows the SonarQube web interface. The top navigation bar includes 'sonarqube', 'Dashboards', 'Issues' (selected), 'Measures', 'Rules', 'Quality Profiles', 'Quality Gates', 'Settings', and 'More'. The left sidebar contains filters for 'Issues', 'Severity', 'Resolution', 'Status', 'New Issues', and 'Rule'. The main content area displays three issues, each with a title, severity, and status.

Severity	Count
Blocker	0
Critical	9
Major	39
Minor	14
Info	0

Resolution	Count
Unresolved	62
False Positive	0
Removed	0
Fixed	0
Won't fix	0

Issue Title	Severity	Status
My Wrong usage of CannotDecreaseVisibilityAttribute	Major	Open
My Methods that could have a lower visibility	Major	Open
My Types tagged with FullCoveredAttribute should be 100% covered	Major	Open

# 코딩 컨벤션(Coding Convention)

- 명명규칙(Naming Convention)
  - 헝가리안 표기법(Hungarian Notation)
    - `ia = fb; /* i:integer, f:float, s:String, ... */`
    - `CObject(C:Class)`
  - 파스칼 표기법(Pascal Casing)/낙타 표기법(Camel Casing)
    - `IAmABoy/iAmABoy`
  - CoC(Convention over Configuration) 네이밍
    - Class Name - Table Name - File Name
    - Model - View - Controller Name
    - 기본키(id) / int / auto\_increment
    - 외래키(테이블명\_id)
- 들여쓰기(Indentation) 강제(Python)
  - Space over Tab(required)
- 코드리뷰

# Trello

**MDN Content Team Status** ☆ Public

**On Hold (committed but no work on it for the moment)**

- MercrediDocs/WednesdayDocs /MittwochDoks Q2 2015  
1 0/3
- Learning Area: 1st PathWay  
2 22/28 Nov 30, 2014
- Refactor HTMLElement  
6/19
- Refactor HTML "global attributes" documentation  
29/31
- Auto-generating API reference docs  
2
- CSS Tutorial  
3 Jun 30
- JS Tutorial  
1 Sep 30
- Review Web Animations API docs  
1/21

**Doing (Task committed and actively worked on this week)**

- Q1: Overhaul JavaScript guide  
11/18
- Follow-up triage meeting 2015-02-17/24 +03-03/10  
2 7/15
- Content Kits prototype content  
5 8/12 Feb 6
- Follow-up triage meeting 2015-02-10  
5/6 Feb 17
- Follow-up triage meeting 2015-02-03  
1 5/12 Feb 10
- Follow-up triage meeting 2015-01-27  
1 7/13 Feb 3
- Web Components - Custom Elements  
1/6

**Review needed**

- Document Fetch API  
5 5/5/7
- Channel messaging API  
6/6
- Service Workers (Review and complete docs)  
3 23/23
- Apps quickstart (formerly Recroom)  
2 7/7
- Update Web workers  
5 25/25
- CSS help in DevTools  
4 1/5

**No update in the last 14 days. PLEASE UPDATE**

- Contribution Pathways  
1 0/4 Dec 1, 2014

**Completed in March 2015**

- Write community blog article about MDN curriculum fellowship  
2
- MercrediDocs/WednesdayDocs /MittwochDoks Q1 2015  
5 6/7 Apr 1
- MDN 10th Anniversary Plan & brief  
5/6 Jan 30
- Content Kits meeting with tech evangelism  
1 Mar 27
- Social Media Plan for MDN  
6/8
- Inheritance Diagrams

**Completed in February 2015**

- FOSDEM talk  
Feb 1
- JFokus Conf Florian Feb 2 - 4  
6/6
- Web Compat Summit Mt. View.  
1
- Mt. View sprint APIRef / sidebar  
1 2/2
- Firefox 36 for developers  
5/6 Feb 24
- Follow-up triage meeting 2014-11-25  
4 14/14 Dec 9, 2014
- Glossary: Guidelines for using the glossary template  
3/3
- Update Learning Area Meta documentation  
7/7
- Follow-up triage meeting 2014-12-09

**Menu**

**Members**

- Jeremie Patonnier on MercrediDocs/WednesdayDocs/MittwochDoks Q2 2015
  - added Jean-Yves Perrier
  - added Florian Scholz
  - added Ali S
  - joinedMar 27 at 15:55
- Jeremie Patonnier on MercrediDocs/WednesdayDocs/MittwochDoks Q2 2015
  - I need to find a sponsor for May 6th as I will be PTO and not available to host the event.Mar 27 at 15:54
- Jeremie Patonnier added MercrediDocs/WednesdayDocs/MittwochDoks Q2 2015 to On Hold (committed but no work on it for the moment) and added Q2 Date. Mar 27 at 15:52
- Mark Giffin moved Web Components - Custom Elements from On Hold (committed but no work on it



# Slack

The screenshot displays the Slack web interface. On the left is a dark sidebar with the workspace name 'Acme Sites' and a list of channels including #api, #bugs, #cats, #engineering (with a red notification badge), #general, #marketing, #ops, #products (highlighted in green), #project-x, #project-y, #project-z, #random, #sales, #support, and #web. Below the channels are direct messages with slackbot, Kevin, and '+9 More...'. At the bottom of the sidebar are private groups: design-team and 'New private group...'. The main area shows the '#products' channel with a list of messages from Kevin, Jessica, Mike, David, and Don. A context menu is open over Don's message, offering options to 'Upload a file', 'Create a text snippet', and 'Create a post'. On the right, a 'Files' panel shows a document titled 'One-Way NDA Template.doc' (31KB DOC File) with a 'Download File' button. Below the file panel is a 'Comments' section with three comments from Sara, David, and Don. The bottom of the interface shows a message input area with a plus icon, a text field, and a send button.

**Acme Sites**

**CHANNELS**

- #api
- #bugs
- #cats
- #engineering 2
- #general
- #marketing
- #ops
- #products
- #project-x
- #project-y
- #project-z
- #random
- #sales
- #support
- #web
- Create a channel...

**DIRECT MESSAGES**

- slackbot
- Kevin
- +9 More...

**PRIVATE GROUPS**

- design-team
- New private group...

**#products**

**kevin** 11:03 AM  
they rock!

**jessica** 11:16 AM  
Shared a post: [Products Page Copy](#)  
Whether small or large, pick the product that's best for your team. Have a fear of commitment? Our 30 day trial is just the thing to help you make up your mind. The best part: no more worries, cancel anytime! We won't be hurt, promise. Ask about our ... more  
Edit · New window · Add comment

**mike** 11:24 AM  
liking the copy so far, Jessica!  
hey, guys. I'm just getting some things ready for the offsite, are you able to post the agenda so far? I have a few things I want to cover.

**david** 11:29 AM  
[Offsite Agenda - draft](#)  
Shared Google Drive Document · in #random, #general · Add comment

**mike** 11:38 AM  
looks good. I'd like to add an afternoon session for product & engineering on Tuesday. You could maybe schedule a parallel session for sales, marketing and support.

**tj** 11:37 AM  
sounds good to me! I'll run the Tuesday afternoon session for my group

**don** 12:01 PM  
[Mobile In-Store Research from Google](#)  
15MB PDF in #products · 2 comments

Upload a file · how mobile devices are affecting re habits.  
Create a text snippet  
Create a post · this'll come in handy!

**Files**

**saras** [One-Way NDA Template.doc](#)  
31KB DOC File  
Download File

Nov 22nd at 9:31 AM · 31KB DOC  
Shared in #products · Share  
Get public link

**Comments**

**saras** Nov 22nd at 9:31 AM  
Standard template for anyone working with external vendors or partners on pre-production, pre-release product.

**david** Nov 22nd at 9:34 AM  
Can I use this one for 2-way mutual NDA?

**david** Nov 22nd at 9:35 AM  
Oh, never mind, I got an answer from Mike. It's fine to use it for 2-way NDA agreements.

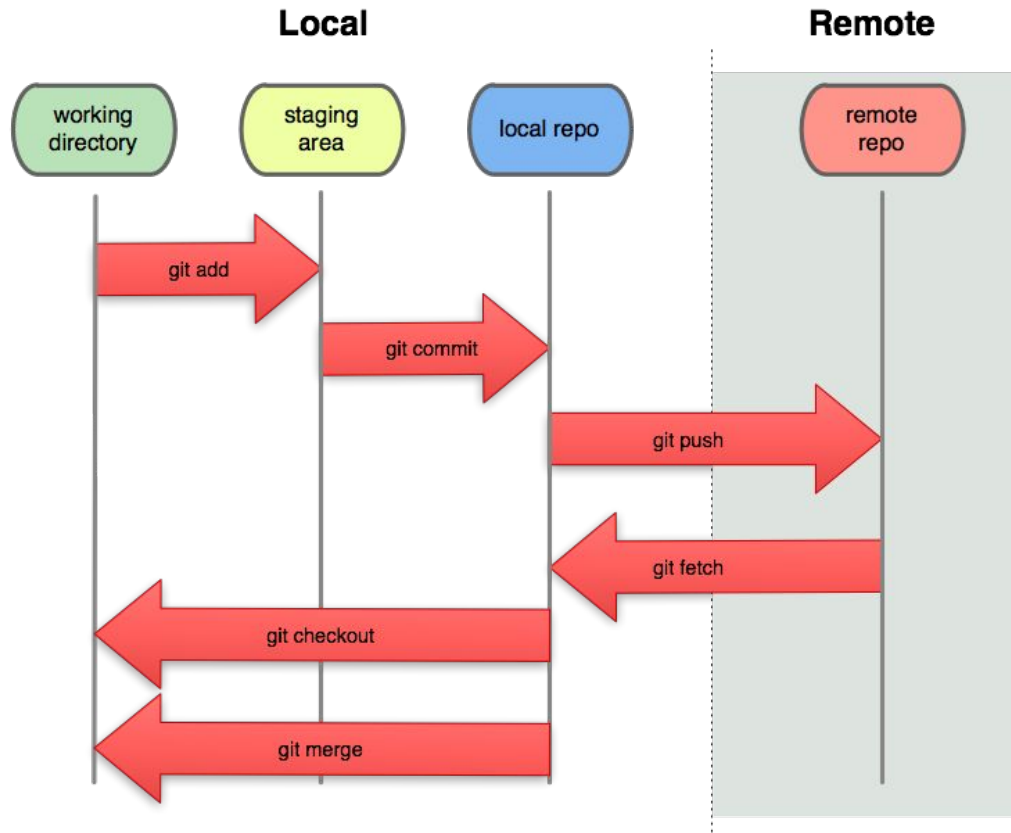
**don** Nov 22nd at 9:40 AM  
Thanks for sharing, Sara. I need to use this at least once a month and always spend time looking for it.

**david** online

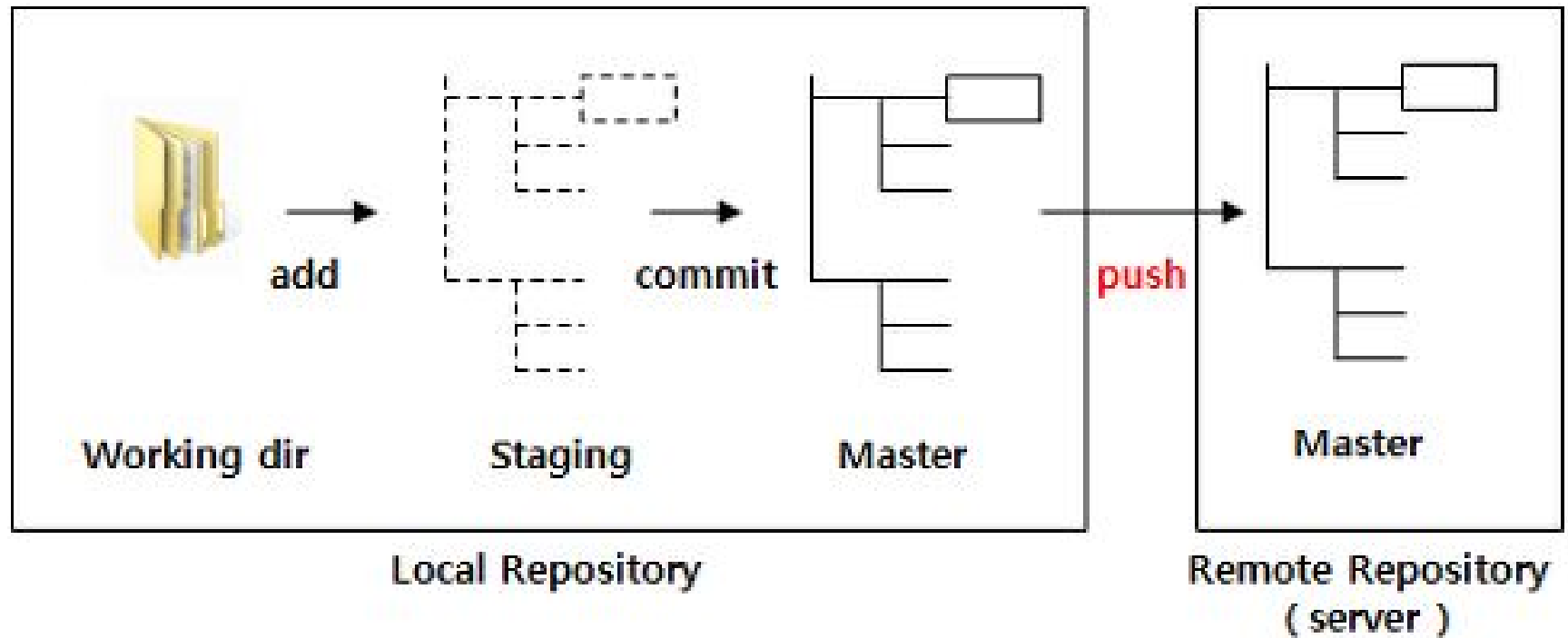
# Git 소개

# Git

- 분산(**Distributed**) 저장소를 지원하는 소스 버전 관리기술
  - 중앙 저장소의 데이터가 손실이 생겨도 로컬 저장소의 내용으로 복구가능
- 기본적으로 Git은 CUI(Command-line User Interface)임



# Git

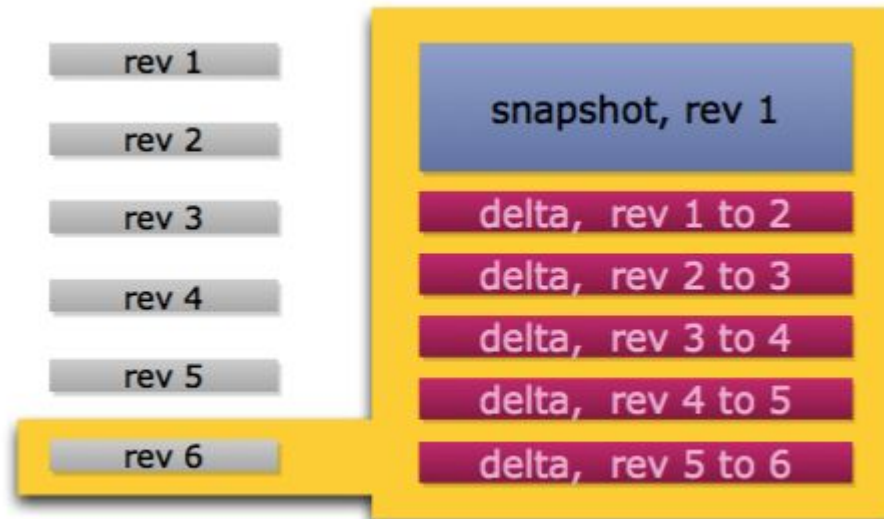


# Git

- 기존의 중앙집중식 소스버전관리툴은 네트워크 속도에 영향을 많이 받음
- **Git은 네트워크 속도의 영향을 작게 받음**
  - 심지어 네트워크에 연결되어 있지 않아도 동작
- 명령어 기반이나 **IDE** 개발환경에서 **GUI**를 제공하는 경우가 많음
  - 플러그인
    - Eclipse의 eGIT
    - Visual Studio의 MS Git Provider(Plugin)

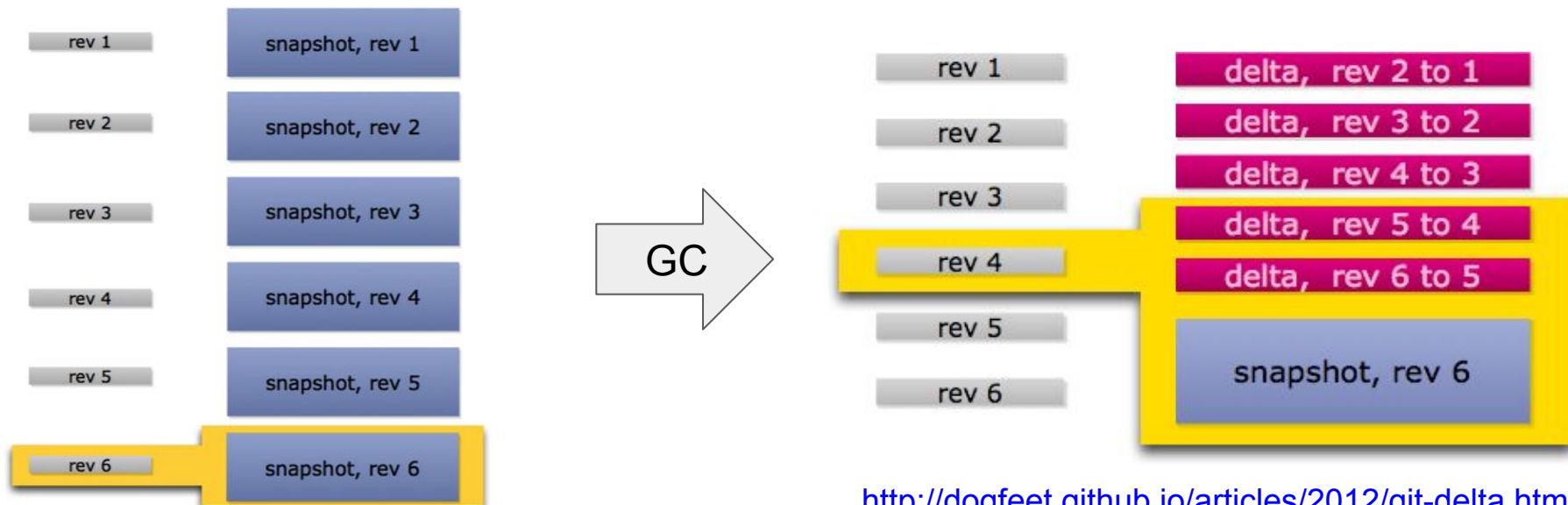
# Git의 특징

- 기존 소스 버전 관리방식
  - 델타(Delta)
  - 파일을 커밋(commit)하면 저장된 파일과 새로운 파일과의 비교를 통해 파일 간의 변경사항(Delta)만 저장함
  - 처음 커밋한 파일내용(HEAD) +Delta 1+Delta 2+Delta 3+...을 통해 현재 파일 내용을 구성함



# Git의 특징

- Git의 저장 방식
  - 처음에는 **스냅샷(Snapshot)**방식으로 저장
  - 파일을 커밋하면 .git 저장소에 원본파일과 새로운 파일이 계속 누적되어서 저장됨
  - 일정시점에서 GC(Garbage Collector)가 수행되면서 **마지막 리비전의 파일만 전체 저장하고 나머지는 델타만 저장함**
  - 대부분의 경우 마지막 파일에 요청이 집중됨



# Git 용어

- 체크아웃(Checkout)
  - 로컬 저장소의 특정 리비전의 코드를 로컬 작업공간으로 다운로드
- 커밋(Commit)
  - 로컬 저장소로 저장
- 스테이지(Stage)
  - 작업한 내용이 저장되는 임시 저장공간
- 추가(Add)/삭제(Remove)
  - 저장소이 관리대상에 추가
- 태그(Tag)
  - 특정버전의 커밋에 번호 외에 이름을 붙여 접근할 수 있도록 만듦



# Git 용어

- 태그(Tag)
  - 특정버전의 커밋에 번호 외에 이름을 붙여 접근할 수 있도록 만듦
- 푸시(Push)
  - 로컬저장소에서 원격저장소로 업로드하는 과정
- 풀(Pull)/복제(Clone)
  - 원격저장소에서 로컬저장소로 가져오는 과정
- 브랜치(Branch)
  - 메인 브랜치에서 별도의 가지를 생성해서 별도의 커밋을 할 수 있도록 만듦
  - (예) development/test/production
  - 마스터(Master) 브랜치
- 병합(Merge)
  - 두 개 이상의 브랜치를 합치는 것
  - 3-way merge
- 포크(Fork)
  - 기존 저장소를 복사해 새로운 독립 저장소 생성

# Git 리모트 저장소(Remote Repository)

- CUI(Command-line UI) vs. GUI(Web)
- 직접 생성해서 관리가능(DIY:Do It Yourself)
- 클라우드 형태의 서비스(GitHub)
- 설치 형태의 서비스
  - BitBucket vs. GitLab

# Git 기반 솔루션

- GitHub vs. GitLab
- 분산 저장소 외에 **이슈관리**, 위키 등 통합 제공(+소셜기능 Like/Star)
- **GitHub**는 클라우드 기반, **GitLab**은 호스팅 + 설치형
- 가격 정책도 상이
  - **GitHub**
    - Public Repository는 무료/Private Repository는 2개 이상은 유료
    - 사용자 수에 따라서 과금
  - **BitBucket**
    - Private 5명 이상 과금 사용
  - **GitLab**
    - <http://gitlab.com>
    - 로컬 설치형과 호스팅형이 있음
    - 커뮤니티 버전(**Community Edition**)은
      - 사용자 수와 프로젝트 수(public/private)에 영향받지 않음
    - 이 외 버전은 유료

# GitLab이란

- 오픈소스기반
- 버전관리(**Source Version Control**) + 이슈관리(**ITS:Issue Tracking System**)  
+ 지속적 통합(**CI:Continuous Integration**) + 위키 (**Wiki**)
- 웹기반 인터페이스
- 원격저장소(**Remote Repository**)
- 설치 및 관리가 복잡함

# 소프트웨어 협업도구(GitLab)

- 소스 버전 컨트롤(Source Version Control)
- 이슈 트래킹 시스템(Issue Tracking System)
- 위키(Wiki)
- 지속적 통합(CI:Continuous Integration)
- 프로젝트 품질관리
- 그 외
  - 슬랙(Slack)
    - 개발자 전용 메신저
  - 트렐로(Trello)
    - 협업도구

# GitLab기반 통합관리 시스템(LMS) 구축

- 네이버와 협력해 통합 **GitLab**기반 LMS 시스템(포털) 구축
  - 각 과목별 관리가능
  - 각 과목별 조교 사용법 교육

Git 및 GitLab LMS 실습

# GitLab 준비사항

- 1. GitLab 접속
  - <http://gitlab.pess.hanyang.ac.kr>
- 2. 한양포털 아이디로 로그인
- 3. 프로젝트 생성
  - Fetch로 샘플 프로젝트 생성



# Git 사용법

- 로컬 저장소 생성

**\$ git init**

Initialized empty Git repository in c:/dev/git/.git/

- 파일 추가

**\$ touch sample.txt or vi sample.txt**

**\$ git add sample.txt or git .**

- 현재 상태보기

**\$ git status**

- 로컬 저장소에 커밋하기

**\$ git commit -m "sample contents" --author="Seokjae <[sjha72@hanyang.ac.kr](mailto:sjha72@hanyang.ac.kr)>"**

# Git 사용법

- 커밋 히스토리 확인

```
$ git log master
```

- 태깅 및 태그 버전 확인

```
$ git tag -a 0.1-m "sampleTag v0.1"
```

```
$ git tag
```

# Git 사용법

- 브랜치 생성 및 브랜치 목록 확인

**\$ git branch "sampleBranch"**

**\$ git branch**

현재 사용중인 branch는 \*표로 표시됨

- 브랜치 전환하기

**\$ git checkout "sampleBranch"**

- 메인 master로 전환하기

**\$ git checkout "master"**

- merge 하기 (현재 브랜치에, 브랜치명의 파일들을 합치기)

**\$ git merge "sampleBranch"**

# Git 사용법

- 원격 저장소에 업로드

**\$ git push**

- 원격저장소내역 업데이트

**\$ git pull**

감사합니다