

오픈 소스 얘기

홍민희

3기 최종찬 군과의 대화

<disjukr> 그냥 적당히 한시간 하고 남은시간 놀라고 하면 다들 좋아할걸요

<hongminhee> disjukr: 그렇게들 많이 하나요?

<disjukr> 많이 그랬던걸로(...)

<hongminhee> 그럼 그렇지 2시간이나 어떻게 채워

발표자 소개

발표자 소개

- 홍민희

발표자 소개

- 홍민희
- Wand 메인테이너

발표자 소개

- 홍민희
- Wand 메인테이너
- sphinxcontrib 커미터

발표자 소개

- 홍민희
- Wand 메인테이너
- sphinxcontrib 커미터
- <http://github.com/dahlia>

발표자 소개

- 홍민희
- Wand 메인테이너
- sphinxcontrib 커미터
- <http://github.com/dahlia>
- <http://dahlia.kr/>

발표자 소개

발표자 소개

- 2010년 소프트웨어 마에스트로 1기
때부터 멘토로 활동

발표자 소개

- 2010년 소프트웨어 마에스트로 1기 때부터 멘토로 활동
- 1999년 초등학교 5학년 때부터 프로그래밍 시작

포켓몬스터 홈페이지

포켓몬스터 홈페이지

- 1999년 한국에서도 SBS에서 포켓몬스터가 방영되기 시작

포켓몬스터 홈페이지

- 1999년 한국에서도 SBS에서 포켓몬스터가 방영되기 시작
- 나모 웹 에디터를 사용해서 포켓몬스터 홈페이지를 만듦

포켓몬스터 홈페이지

- 1999년 한국에서도 SBS에서 포켓몬스터가 방영되기 시작
- 나모 웹 에디터를 사용해서 포켓몬스터 홈페이지를 만듦
- 디지몬을 숭배하는 이교도들의 홈페이지에 있는 게시판이 부러워 보였음

포켓몬스터 홈페이지

- 1999년 한국에서도 SBS에서 포켓몬스터가 방영되기 시작
- 나모 웹 에디터를 사용해서 포켓몬스터 홈페이지를 만듦
- 디지몬을 숭배하는 이교도들의 홈페이지에 있는 게시판이 부러워 보였음
- 하지만 정적인 HTML로는 게시판 같은 것을 만들 방법이 떠오르지 않음

포켓몬스터 홈페이지

- 1999년 한국에서도 SBS에서 포켓몬스터가 방영되기 시작
- 나모 웹 에디터를 사용해서 포켓몬스터 홈페이지를 만듦
- 디지몬을 숭배하는 이교도들의 홈페이지에 있는 게시판이 부러워 보였음
- 하지만 정적인 HTML로는 게시판 같은 것을 만들 방법이 떠오르지 않음



당시에 먹여주던 게시판

CrazyWWWBoard

CrazyWWWBoard

- C++로 만든 CGI 게시판

CrazyWWWBoard

- C++로 만든 CGI 게시판
- 빌드된 x86 리눅스/리눅스용 바이너리로만 배포됨

CrazyWWWBoard

- C++로 만든 CGI 게시판
- 빌드된 x86 리눅스/리눅스용 바이너리로만 배포됨
- 어쨌거나 내 무료 드림엑스 계정에는 CGI 스크립트를 설치할 수 없었음

Logged Members : 1
 Total 3 articles, 1 pages / Now page is 1
 SIGN IN LOGIN

NAME	srneti
HOME PAGE	http://srneti.com
SUBJECT	그냥.

테스트,

IP Address : 211.219.194.97

srneti [2001/09/30] :: 코멘트 테스트

Name :
 Memo :
 Password :
CONFIRM

LIST VIEW
WRITE
RECOMMEND
REPLY
MODIFY
DELETE

NO	C	SUBJECT	NAME	DATE	HIT	VOTE
>>>	<input type="checkbox"/>	그냥. [1]	srneti	2001/09/30	18	0
2	<input type="checkbox"/>	테스트 2번째.	srneti	2001/09/30	14	0
1	<input type="checkbox"/>	테스트.	srneti	2001/09/30	11	0

LIST VIEW
1
WRITE

☐ Name
 ☒ Subject
 ☐ Contents

Find it!
Return

Copyright 1999-2001 Zeroboard / skin by SR2i-finagle

얼마 후 나타난
더 트렌디한 게시판

제로보드 4

제로보드 4

- 무료로 PHP 소스 코드 형태로 배포했었음

제로보드 4

- 무료로 PHP 소스 코드 형태로 배포했었음
- 지금 와서는 안 그래보이지만 그 당시에는 게시판 중에서 가장 디자인이 좋았음

제로보드 4

- 무료로 PHP 소스 코드 형태로 배포했었음
- 지금 와서는 안 그래보이지만 그 당시에는 게시판 중에서 가장 디자인이 좋았음
- 이것 덕분에 PHP 스크립트를 지원하는 호스팅 서비스가 많이 생겨남

제로보드 4

- 무료로 PHP 소스 코드 형태로 배포했었음
- 지금 와서는 안 그래보이지만 그 당시에는 게시판 중에서 가장 디자인이 좋았음
- 이것 덕분에 PHP 스크립트를 지원하는 호스팅 서비스가 많이 생겨남
- 초딩이라 돈이 없어서 무료 호스팅을 열심히 찾아서 썼음

제로보드 4

- 무료로 PHP 소스 코드 형태로 배포했었음
- 지금 와서는 안 그래보이지만 그 당시에는 게시판 중에서 가장 디자인이 좋았음
- 이것 덕분에 PHP 스크립트를 지원하는 호스팅 서비스가 많이 생겨남
- 초딩이라 돈이 없어서 무료 호스팅을 열심히 찾아서 썼음

제로보드 4는 왜
오픈 소스가 아니었나?

제로보드 4는 왜 오픈 소스가 아니었나?

- 분명, 무료로 PHP 소스 코드 형태로
배포했었음

제로보드 4는 왜 오픈 소스가 아니었나?

- 분명, 무료로 PHP 소스 코드 형태로 배포했었음
- 하지만 제로보드 4는 오픈 소스가 아님

제로보드 4는 왜 오픈 소스가 아니었나?

- 분명, 무료로 PHP 소스 코드 형태로 배포했었음
- 하지만 제로보드 4는 오픈 소스가 아님
- 이유는?

제로보드 4는 왜 오픈 소스가 아니었나?

- 분명, 무료로 PHP 소스 코드 형태로 배포했었음
- 하지만 제로보드 4는 오픈 소스가 아님
- 이유는?
- (힌트: 제로보드 4에 왜 그렇게 보안 버그가 많았을까?)

오픈 소스에 관한
널리 퍼진 오해들

오픈 소스에 관한 널리 퍼진 오해들

- 오픈 소스는 사용 권리에 관한 것이다

오픈 소스에 관한 널리 퍼진 오해들

- 오픈 소스는 사용 권리에 관한 것이다
- 혹은, 소스 코드 공개에 관한 것이다

오픈 소스에 관한 널리 퍼진 오해들

- 오픈 소스는 사용 권리에 관한 것이다
- 혹은, 소스 코드 공개에 관한 것이다
- (즉, 소프트웨어의 내부를 볼 수 있게 해주는 것이다)

오픈 소스에 관한 널리 퍼진 오해들

- 오픈 소스는 사용 권리에 관한 것이다
- 혹은, 소스 코드 공개에 관한 것이다
- (즉, 소프트웨어의 내부를 볼 수 있게 해주는 것이다)
- 혹은, 사용 권리와 소스 코드를 함께 제공하는 것이다

오픈 소스는
사용 권리에 관한 것인

오픈 소스는 사용 권리에 관한 것인

- 만약 그렇다면 꿀뷰, 초기에 배포되던 알집, Visual Studio Express, Xcode, Internet Explorer, Safari 등 무료로 사용 가능한 많은 소프트웨어들은 오픈 소스라고 해야 할 것이다

오픈 소스는 사용 권리에 관한 것인

- 만약 그렇다면 꿀뷰, 초기에 배포되던 알집, Visual Studio Express, Xcode, Internet Explorer, Safari 등 무료로 사용 가능한 많은 소프트웨어들은 오픈 소스라고 해야 할 것이다
- 일반인 사이에서 널리 퍼진 오해

오픈 소스는 사용 권리에 관한 것인

- 만약 그렇다면 꿀뷰, 초기에 배포되던 알집, Visual Studio Express, Xcode, Internet Explorer, Safari 등 무료로 사용 가능한 많은 소프트웨어들은 오픈 소스라고 해야 할 것이다
- 일반인 사이에서 널리 퍼진 오해
- 적어도 이 자리에 있는 사람들 중 이 오해를 믿는 사람은 없을 것임

오픈 소스는 사용 권리에 관한 것인

- 만약 그렇다면 꿀뷰, 초기에 배포되던 알집, Visual Studio Express, Xcode, Internet Explorer, Safari 등 무료로 사용 가능한 많은 소프트웨어들은 오픈 소스라고 해야 할 것이다
- 일반인 사이에서 널리 퍼진 오해
- 적어도 이 자리에 있는 사람들 중 이 오해를 믿는 사람은 없을 것임

오픈 소스는 소스 코드
공개에 관한 것인가

오픈 소스는 소스 코드 공개에 관한 것인가

- 즉, 소프트웨어의 내부 구조를 드러내면
오픈 소스인가

오픈 소스는 소스 코드 공개에 관한 것인가

- 즉, 소프트웨어의 내부 구조를 드러내면
오픈 소스인가
- 제로보드 4는 소스 코드를 공개했지만
오픈 소스가 아님

오픈 소스는 소스 코드 공개에 관한 것인가

- 즉, 소프트웨어의 내부 구조를 드러내면 오픈 소스인가
- 제로보드 4는 소스 코드를 공개했지만 오픈 소스가 아님
- 결국 이것 역시 매우 널리 퍼진 오해

오픈 소스는 소스 코드 공개에 관한 것인가

- 즉, 소프트웨어의 내부 구조를 드러내면 오픈 소스인가
- 제로보드 4는 소스 코드를 공개했지만 오픈 소스가 아님
- 결국 이것 역시 매우 널리 퍼진 오해
- 심지어 개발자들 사이에서도 널리 퍼져있는 오해

오픈 소스는 소스 코드 공개에 관한 것인가

- 즉, 소프트웨어의 내부 구조를 드러내면 오픈 소스인가
- 제로보드 4는 소스 코드를 공개했지만 오픈 소스가 아님
- 결국 이것 역시 매우 널리 퍼진 오해
- 심지어 개발자들 사이에서도 널리 퍼져있는 오해

오픈 소스는 기본적으로
자유에 관한 것임

오픈 소스는 기본적으로 자유에 관한 것임

- 오픈 소스가 말하는 자유는
사용의 자유가 아님

오픈 소스는 기본적으로 자유에 관한 것임

- 오픈 소스가 말하는 자유는
사용의 자유가 아님
- 소프트웨어를 고칠 수 있는 자유를
주자는 것이 오픈 소스의 본질임

자유를 구현하기 위한
두 가지 측면

자유를 구현하기 위한 두 가지 측면

1. 법적 측면: 고쳐서 재배포할 수 있는
권리를 제공해야 함

자유를 구현하기 위한 두 가지 측면

1. 법적 측면: 고쳐서 재배포할 수 있는 권리를 제공해야 함
2. 기술 측면: 고칠 수 있는 (가공 가능한) 형태여야 함

두 가지 측면의 오픈 소스

두 가지 측면의 오픈 소스

1. 오픈 소스 라이선스

두 가지 측면의 오픈 소스

1. 오픈 소스 라이선스
2. 컴파일된 바이너리는 가공에 용이하지 않으므로 소스 코드로 배포한다

제로보드 4가
오픈 소스가 아닌 이유

제로보드 4가 오픈 소스가 아닌 이유

- 힌트: 제로보드 4에 왜 그렇게 보안 버그가 많았을까?

제로보드 4가 오픈 소스가 아닌 이유

- 힌트: 제로보드 4에 왜 그렇게 보안 버그가 많았을까?
- Repraise: 제로보드 4의 보안 버그는 왜 대부분 고쳐지지 못했을까?

제로보드 4가 오픈 소스가 아닌 이유

- 힌트: 제로보드 4에 왜 그렇게 보안 버그가 많았을까?
- Repraise: 제로보드 4의 보안 버그는 왜 대부분 고쳐지지 못했을까?
- 제로보드 4는 수정된 버전을 재배포하는 것을 금지했음

제로보드 4가
오픈 소스가 아닌 이유

제로보드 4가 오픈 소스가 아닌 이유

- 고쳐서 퍼뜨릴 수 없으면 오픈 소스가
아님

제로보드 4가 오픈 소스가 아닌 이유

- 고쳐서 퍼뜨릴 수 없으면 오픈 소스가 아님
- 보안 패치를 해주고 싶어도 고쳐서 퍼뜨릴 수가 없음

제로보드 4가 오픈 소스가 아닌 이유

- 고쳐서 퍼뜨릴 수 없으면 오픈 소스가 아님
- 보안 패치를 해주고 싶어도 고쳐서 퍼뜨릴 수가 없음
- 혼자서 만드는 제로보드는 수많은 보안 버그를 고치는 데 한계가 있음

오픈 소스의 인프라스트럭처

오픈 소스의 인프라스트럭처

오픈 소스 생태계의 다양한 형태들은 소프트웨어를
고쳐서 다시 퍼뜨릴 수 있게 하기 위한 것임

오픈 소스의 인프라스트럭처

오픈 소스 생태계의 다양한 형태들은 소프트웨어를
고쳐서 다시 퍼뜨릴 수 있게 하기 위한 것임

- 라이선스

오픈 소스의 인프라스트럭처

오픈 소스 생태계의 다양한 형태들은 소프트웨어를
고쳐서 다시 퍼뜨릴 수 있게 하기 위한 것임

- 라이선스
- 소스 코드

오픈 소스의 인프라스트럭처

오픈 소스 생태계의 다양한 형태들은 소프트웨어를
고쳐서 다시 퍼뜨릴 수 있게 하기 위한 것임

- 라이선스
- 소스 코드
- diff, patch

오픈 소스의 인프라스트럭처

오픈 소스 생태계의 다양한 형태들은 소프트웨어를
고쳐서 다시 퍼뜨릴 수 있게 하기 위한 것임

- 라이선스
- 소스 코드
- diff, patch
- 분산형 버전 관리 시스템

오픈 소스의 인프라스트럭처

오픈 소스 생태계의 다양한 형태들은 소프트웨어를
고쳐서 다시 퍼뜨릴 수 있게 하기 위한 것임

- 라이선스
- 소스 코드
- diff, patch
- 분산형 버전 관리 시스템
- 풀 리퀘스트

오픈 소스가 소프트웨어에
관한 것이 아니었다면

오픈 소스가 소프트웨어에 관한 것이 아니었다면

- 저작물을 고쳐서 퍼뜨릴 수 있게 한다는 개념은 사실 소프트웨어 말고도 적용 가능한 것임

오픈 소스가 소프트웨어에 관한 것이 아니었다면

- 저작물을 고쳐서 퍼뜨릴 수 있게 한다는 개념은 사실 소프트웨어 말고도 적용 가능한 것임
- 따라서 형태들을 고치면 다른 분야에서도 오픈 소스를 할 수 있음

오픈 소스가 디자인에 관한 것이었다면

다음과 같이 대응될 수 있음:

오픈 소스	소프트웨어	디자인
라이선스	GPL, MIT, BSD...	Creative Commons
가공 용이성	소스 코드	PSD, AI 파일
병합 도구	diff, patch	(아직 없음)
변화 관리	분산형 버전 관리 시스템	(부족하지만) Dropbox

오픈 소스는 왜 소프트웨어
웨어에 관한 것일까

오픈 소스는 왜 소프트웨어 웨어에 관한 것일까

- 왜 다른 분야에는 오픈 소스가 없을까?

오픈 소스는 왜 소프트웨어 웨어에 관한 것일까

- 왜 다른 분야에는 오픈 소스가 없을까?
- 법적 문제와 달리 기술적 문제를 집단 내부에서 해결할 수 없음

오픈 소스는 왜 소프트웨어 웨어에 관한 것일까

- 왜 다른 분야에는 오픈 소스가 없을까?
- 법적 문제와 달리 기술적 문제를 집단 내부에서 해결할 수 없음
- 소프트웨어 개발자들은 자신의 도구 대부분을 스스로 만들어내는 몇 안되는 집단

오픈 소스는 왜 소프트웨어에 관한 것일까

- 왜 다른 분야에는 오픈 소스가 없을까?
- 법적 문제와 달리 기술적 문제를 집단 내부에서 해결할 수 없음
- 소프트웨어 개발자들은 자신의 도구 대부분을 스스로 만들어내는 몇 안되는 집단
- 그래픽 디자이너는 Adobe Photoshop의 마음에 안드는 동작을 스스로 고칠 수 없음

오픈 소스는 왜 소프트웨어에 관한 것일까

- 왜 다른 분야에는 오픈 소스가 없을까?
- 법적 문제와 달리 기술적 문제를 집단 내부에서 해결할 수 없음
- 소프트웨어 개발자들은 자신의 도구 대부분을 스스로 만들어내는 몇 안되는 집단
- 그래픽 디자이너는 Adobe Photoshop의 마음에 안드는 동작을 스스로 고칠 수 없음

오픈 소스가 발전시킨
소프트웨어 개발 도구

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch
- 표준적인 코딩 스타일의 많은 부분은 병합 도구의 오동작을 피하는 방향으로 발전했음

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch
- 표준적인 코딩 스타일의 많은 부분은 병합 도구의 오동작을 피하는 방향으로 발전했음
- 버전 관리 시스템: CVS, Subversion, Darcs, Mercurial, Git

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch
- 표준적인 코딩 스타일의 많은 부분은 병합 도구의 오동작을 피하는 방향으로 발전했음
- 버전 관리 시스템: CVS, Subversion, Darcs, Mercurial, Git
- 테스트 자동화: xUnit, xSpec

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch
- 표준적인 코딩 스타일의 많은 부분은 병합 도구의 오동작을 피하는 방향으로 발전했음
- 버전 관리 시스템: CVS, Subversion, Darcs, Mercurial, Git
- 테스트 자동화: xUnit, xSpec
- 빌드 자동화 (CI): Buildbot, Hudson, Jenkins, Travis CI

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch
- 표준적인 코딩 스타일의 많은 부분은 병합 도구의 오동작을 피하는 방향으로 발전했음
- 버전 관리 시스템: CVS, Subversion, Darcs, Mercurial, Git
- 테스트 자동화: xUnit, xSpec
- 빌드 자동화 (CI): Buildbot, Hudson, Jenkins, Travis CI
- 위의 도구 모두가 소프트웨어를 고치기 쉽게 하기 위한 용도임

오픈 소스가 발전시킨 소프트웨어 개발 도구

- 병합 도구: diff, patch
- 표준적인 코딩 스타일의 많은 부분은 병합 도구의 오동작을 피하는 방향으로 발전했음
- 버전 관리 시스템: CVS, Subversion, Darcs, Mercurial, Git
- 테스트 자동화: xUnit, xSpec
- 빌드 자동화 (CI): Buildbot, Hudson, Jenkins, Travis CI
- 위의 도구 모두가 소프트웨어를 고치기 쉽게 하기 위한 용도임

오픈 소스를 왜 하나

오픈 소스를 왜 하나

- 대체 소프트웨어를 고칠 수 있으면
뭐가 좋나?

오픈 소스를 왜 하나

- 대체 소프트웨어를 고칠 수 있으면 뭐가 좋나?
- 남이 만든 소프트웨어를 내가 고칠 수 있으면 뭐가 좋은가?

오픈 소스를 왜 하나

- 대체 소프트웨어를 고칠 수 있으면 뭐가 좋나?
- 남이 만든 소프트웨어를 내가 고칠 수 있으면 뭐가 좋은가?
- 반대로, 내가 만든 소프트웨어를 남이 고칠 수 있으면 뭐가 좋은가?

노력의 중복을 줄이기

노력의 중복을 줄이기

- 제로보드 4의 보안 버그가 훨씬 적을 수도 있었음

노력의 중복을 줄이기

- 제로보드 4의 보안 버그가 훨씬 적을 수도 있음
- 오픈 소스는 수고(effort)의 중복을 줄이는데 탁월한 효과가 있음

노력의 중복을 줄이기

- 제로보드 4의 보안 버그가 훨씬 적을 수도 있었음
- 오픈 소스는 수고(effort)의 중복을 줄이는데 탁월한 효과가 있음
- 모든 개발자들은 자신들이 생각하는 것 이상으로 서로 비슷한 것을 구현하는데 같은 수고를 함

노력의 중복을 줄이기

- 제로보드 4의 보안 버그가 훨씬 적을 수도 있음
- 오픈 소스는 수고(effort)의 중복을 줄이는데 탁월한 효과가 있음
- 모든 개발자들은 자신들이 생각하는 것 이상으로 서로 비슷한 것을 구현하는데 같은 수고를 함
- 내가 하룻밤 불태워 만들고 잊어버릴 소프트웨어조차 꾸준히 관리될 수 있음
(물론, 그게 효용이 있다면)

노력의 중복을 줄이기

- 제로보드 4의 보안 버그가 훨씬 적을 수도 있음
- 오픈 소스는 수고(effort)의 중복을 줄이는데 탁월한 효과가 있음
- 모든 개발자들은 자신들이 생각하는 것 이상으로 서로 비슷한 것을 구현하는데 같은 수고를 함
- 내가 하룻밤 불태워 만들고 잊어버릴 소프트웨어조차 꾸준히 관리될 수 있음
(물론, 그게 효용이 있다면)

안 좋은 생각: NIH Syndrome

안 좋은 생각: NIH Syndrome

- “Not Invented Here” (바퀴의 재발명)

안 좋은 생각: NIH Syndrome

- “Not Invented Here” (바퀴의 재발명)
- 흔히 놓치는 것들

안 좋은 생각: NIH Syndrome

- “Not Invented Here” (바퀴의 재발명)
- 흔히 놓치는 것들
 - 당신은 당신이 만든 소프트웨어를 몇년
씩 붙잡고 유지 보수할 수 없음

안 좋은 생각: NIH Syndrome

- “Not Invented Here” (바퀴의 재발명)
- 흔히 놓치는 것들
 - 당신은 당신이 만든 소프트웨어를 몇년씩 붙잡고 유지 보수할 수 없음
 - 다른 사람의 오픈 소스 구현에서 보이는 부족한 점 1에 비해 당신의 조건을 만족하는 9가 있음

안 좋은 생각: NIH Syndrome

- “Not Invented Here” (바퀴의 재발명)
- 흔히 놓치는 것들
 - 당신은 당신이 만든 소프트웨어를 몇년씩 붙잡고 유지 보수할 수 없음
 - 다른 사람의 오픈 소스 구현에서 보이는 부족한 점 1에 비해 당신의 조건을 만족하는 9가 있음
 - 오픈 소스이므로 당신이 그 부족한 1을 채워줄 수 있음

안 좋은 생각: NIH Syndrome

- “Not Invented Here” (바퀴의 재발명)
- 흔히 놓치는 것들
 - 당신은 당신이 만든 소프트웨어를 몇년씩 붙잡고 유지 보수할 수 없음
 - 다른 사람의 오픈 소스 구현에서 보이는 부족한 점 1에 비해 당신의 조건을 만족하는 9가 있음
 - 오픈 소스이므로 당신이 그 부족한 1을 채워줄 수 있음

안 좋은 생각: NIH Syndrome

안 좋은 생각: NIH Syndrome

- 남의 소스 코드를 고치기 싫어하는 사람의 대부분은 소스 코드 읽는 훈련이 부족한 경우

안 좋은 생각: NIH Syndrome

- 남의 소스 코드를 고치기 싫어하는 사람의 대부분은 소스 코드 읽는 훈련이 부족한 경우
- 현대의 프로그래밍 교육은 쓰기 교육을 주로 강조하고 읽기에 대한 교육을 거의 하지 않기 때문

안 좋은 생각: NIH Syndrome

- 남의 소스 코드를 고치기 싫어하는 사람의 대부분은 소스 코드 읽는 훈련이 부족한 경우
- 현대의 프로그래밍 교육은 쓰기 교육을 주로 강조하고 읽기에 대한 교육을 거의 하지 않기 때문
- 다행히, 오픈 소스 소프트웨어는 대부분의 그렇지 않은 소프트웨어에 비해 평균적으로 소스 코드 품질이 높으므로 읽기가 쉬운 편

안 좋은 생각: NIH Syndrome

안 좋은 생각: NIH Syndrome

- 천하 고수들의 심득을 얻을 수 있음!

안 좋은 생각: NIH Syndrome

- 천하 고수들의 심득을 얻을 수 있음!
- 까놓고 말해, 당신이 혼자서 10년 동안 OS를 만들면서 배우는 것보다, 1년 동안 리눅스 소스 코드를 읽고 패치 몇개 보내는 것이 배우는 양과 깊이 양쪽에서 훨씬 뛰어날 것이다

안 좋은 생각: NIH Syndrome

- 천하 고수들의 심득을 얻을 수 있음!
- 까놓고 말해, 당신이 혼자서 10년 동안 OS를 만들면서 배우는 것보다, 1년 동안 리눅스 소스 코드를 읽고 패치 몇개 보내는 것이 배우는 양과 깊이 양쪽에서 훨씬 뛰어날 것이다
- 메일링 리스트를 눈팅하면 디자인이 아니라, 디자인 결정의 방식 자체를 배울 수 있음!

품질을 올리기

품질을 올리기

- 소프트웨어의 품질을 평가하는 덕목 중 하나는 얼마나 고치기 쉬운가임

품질을 올리기

- 소프트웨어의 품질을 평가하는 데목 중 하나는 얼마나 고치기 쉬운가임
- 오픈 소스 프로젝트는 상대적으로 수정이 쉬운 형태를 유지하는 데 의식적인 노력을 함

품질을 올리기

- 소프트웨어의 품질을 평가하는 데목 중 하나는 얼마나 고치기 쉬운가임
- 오픈 소스 프로젝트는 상대적으로 수정이 쉬운 형태를 유지하는 데 의식적인 노력을 함
- 전혀 연고가 없는 다른 대륙의 프로그래머도 고칠 수 있으려면 문서화도 되어 있어야 함

품질을 올리기

- 소프트웨어의 품질을 평가하는 데목 중 하나는 얼마나 고치기 쉬운가임
- 오픈 소스 프로젝트는 상대적으로 수정이 쉬운 형태를 유지하는 데 의식적인 노력을 함
- 전혀 연고가 없는 다른 대륙의 프로그래머도 고칠 수 있으려면 문서화도 되어 있어야 함
- 버전 관리가 대체로 잘 되므로 코드의 특정 부분이 언제 어떤 이유로 추가됐는지 추적하기 쉬움

품질을 올리기

- 소프트웨어의 품질을 평가하는 데목 중 하나는 얼마나 고치기 쉬운가임
- 오픈 소스 프로젝트는 상대적으로 수정이 쉬운 형태를 유지하는 데 의식적인 노력을 함
- 전혀 연고가 없는 다른 대륙의 프로그래머도 고칠 수 있으려면 문서화도 되어 있어야 함
- 버전 관리가 대체로 잘 되므로 코드의 특정 부분이 언제 어떤 이유로 추가됐는지 추적하기 쉬움

재미

재미

- 결국은 개인적인 재미가 가장 큰 동기

재미

- 결국은 개인적인 재미가 가장 큰 동기
- 관심 있는 프로젝트의 이슈 트래커에 있는 이슈를 하나씩 고쳐나가면 퀘스트를 깨는 기분

재미

- 결국은 개인적인 재미가 가장 큰 동기
- 관심 있는 프로젝트의 이슈 트래커에 있는 이슈를 하나씩 고쳐나가면 퀘스트를 깨는 기분
- 내가 만든 소프트웨어에 버그 리포팅이나 패치가 왔을 때의 기쁨

오픈 소스 시작하기

오픈 소스 시작하기

- 가장 쉬운 시작은 자신이 평소에 자주 쓰던 소프트웨어에 패치를 보내는 것

오픈 소스 시작하기

- 가장 쉬운 시작은 자신이 평소에 자주 쓰던 소프트웨어에 패치를 보내는 것
- 분명 매일 썼으면 버그나 부족한 점 하나 정도는 찾았을 것이다

오픈 소스 시작하기

- 가장 쉬운 시작은 자신이 평소에 자주 쓰던 소프트웨어에 패치를 보내는 것
- 분명 매일 썼으면 버그나 부족한 점 하나 정도는 찾았을 것이다
- 큰 프로젝트보다는 작은 프로젝트에서 시작하는 게 부담이 적음

오픈 소스 시작하기

- 가장 쉬운 시작은 자신이 평소에 자주 쓰던 소프트웨어에 패치를 보내는 것
- 분명 매일 썼으면 버그나 부족한 점 하나 정도는 찾았을 것이다
- 큰 프로젝트보다는 작은 프로젝트에서 시작하는 게 부담이 적음
- 매일 쓰는 “애플리케이션”보다는 매일 쓰는 “라이브러리”가 더 프로젝트 규모가 작을 가능성이 높음

오픈 소스 시작하기

- 가장 쉬운 시작은 자신이 평소에 자주 쓰던 소프트웨어에 패치를 보내는 것
- 분명 매일 썼으면 버그나 부족한 점 하나 정도는 찾았을 것이다
- 큰 프로젝트보다는 작은 프로젝트에서 시작하는 게 부담이 적음
- 매일 쓰는 “애플리케이션”보다는 매일 쓰는 “라이브러리”가 더 프로젝트 규모가 작을 가능성이 높음

오픈 소스에 참여하기
위해 알아야 할 것들

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법
- 병합 도구 (diff, patch)

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법
- 병합 도구 (diff, patch)
- 버전 관리 시스템 (Git, Mercurial, Subversion)

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법
- 병합 도구 (diff, patch)
- 버전 관리 시스템 (Git, Mercurial, Subversion)
- 풀 리퀘스트

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법
- 병합 도구 (diff, patch)
- 버전 관리 시스템 (Git, Mercurial, Subversion)
- 풀 리퀘스트
- 빌드 자동화 시스템 (Buildbot, Travis CI)

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법
- 병합 도구 (diff, patch)
- 버전 관리 시스템 (Git, Mercurial, Subversion)
- 풀 리퀘스트
- 빌드 자동화 시스템 (Buildbot, Travis CI)
- 처음부터 이 모든 것을 알아야 하는 것은 아님!

오픈 소스에 참여하기 위해 알아야 할 것들

- 이슈 트래커
- 테스트 코드 작성법
- 병합 도구 (diff, patch)
- 버전 관리 시스템 (Git, Mercurial, Subversion)
- 풀 리퀘스트
- 빌드 자동화 시스템 (Buildbot, Travis CI)
- 처음부터 이 모든 것을 알아야 하는 것은 아님!

버그 리포팅

버그 리포팅

- 소스 코드를 고치지 않더라도,
알려지지 않은 버그를 리포팅하는 것도
훌륭한 기여가 될 수 있음

버그 리포팅

- 소스 코드를 고치지 않더라도,
알려지지 않은 버그를 리포팅하는 것도
훌륭한 기여가 될 수 있음
- 단, 버그 리포팅에도 좋은 방법과
나쁜 방법이 있음!

버그 리포팅: 나쁜 방법

버그 리포팅: 나쁜 방법

- “파일 저장에 실패했습니다”

버그 리포팅: 나쁜 방법

- “파일 저장에 실패했습니다”
- “안된다”는 너무 불명확한 서술임

버그 리포팅: 나쁜 방법

- “파일 저장에 실패했습니다”
- “안된다”는 너무 불명확한 서술임
- 안되는 방식에는 여러 가지 종류가 있을 수 있음

버그 리포팅: 나쁜 방법

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음
- 저장 버튼은 보이지만 회색으로 표시될 수도 있음

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음
- 저장 버튼은 보이지만 회색으로 표시될 수도 있음
- 저장 버튼은 제대로 보이지만 클릭했을 때 아무 반응이 없을 수도 있음

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음
- 저장 버튼은 보이지만 회색으로 표시될 수도 있음
- 저장 버튼은 제대로 보이지만 클릭했을 때 아무 반응이 없을 수도 있음
- 저장 버튼을 클릭하면 프로그램이 뺏는다는 얘기일 수도 있음

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음
- 저장 버튼은 보이지만 회색으로 표시될 수도 있음
- 저장 버튼은 제대로 보이지만 클릭했을 때 아무 반응이 없을 수도 있음
- 저장 버튼을 클릭하면 프로그램이 뺏는다는 얘기일 수도 있음
- 저장 버튼을 클릭하면 파일 선택 대화상자가 뜨지만 그 안에서 저장 버튼이 안 보일 수도 있음

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음
- 저장 버튼은 보이지만 회색으로 표시될 수도 있음
- 저장 버튼은 제대로 보이지만 클릭했을 때 아무 반응이 없을 수도 있음
- 저장 버튼을 클릭하면 프로그램이 뺏는다는 얘기일 수도 있음
- 저장 버튼을 클릭하면 파일 선택 대화상자가 뜨지만 그 안에서 저장 버튼이 안 보일 수도 있음
- 대부분의 경우 “뭐가 안돼요”는 매우 안 좋은 버그 리포팅임

버그 리포팅: 나쁜 방법

- 파일 저장 버튼이 안 보일 수도 있음
- 저장 버튼은 보이지만 회색으로 표시될 수도 있음
- 저장 버튼은 제대로 보이지만 클릭했을 때 아무 반응이 없을 수도 있음
- 저장 버튼을 클릭하면 프로그램이 뺏는다는 얘기일 수도 있음
- 저장 버튼을 클릭하면 파일 선택 대화상자가 뜨지만 그 안에서 저장 버튼이 안 보일 수도 있음
- 대부분의 경우 “뭐가 안돼요”는 매우 안 좋은 버그 리포팅임

버그 리포팅:
좋은 방법

버그 리포팅: 좋은 방법

- “파일 저장 버튼을 누르면 파일 선택 창이 떠야 할텐데, 눌러도 아무 일이 일어나지 않습니다.”

버그 리포팅: 좋은 방법

- “파일 저장 버튼을 누르면 파일 선택 창이 떠야 할텐데, 눌러도 아무 일이 일어나지 않습니다.”
- “기대한 결과”(expected)와
“실제 결과”(actual)을 모두 말하면 됨

버그 리포팅: 좋은 방법

- “파일 저장 버튼을 누르면 파일 선택 창이 떠야 할텐데, 눌러도 아무 일이 일어나지 않습니다.”
- “기대한 결과”(expected)와
“실제 결과”(actual)을 모두 말하면 됨
- 가능하면 누구나 재현 가능한 절차를 나열하면
더 좋음

버그 리포팅: 좋은 방법

- “파일 저장 버튼을 누르면 파일 선택 창이 떠야 할텐데, 눌러도 아무 일이 일어나지 않습니다.”
- “기대한 결과”(expected)와
“실제 결과”(actual)을 모두 말하면 됨
- 가능하면 누구나 재현 가능한 절차를 나열하면
더 좋음
- 재현 절차를 적는 것보다 테스트 코드를 작성할
수 있다면 테스트 코드를 첨부하는 것이 큰 도움
이 됨

버그 리포팅:
좋은 방법 (테스트 코드)

버그 리포팅: 좋은 방법 (테스트 코드)

재현을 위한 테스트 코드는 애플리케이션보다 라이브러리 프로젝트에서 선호되는 방식

버그 리포팅: 좋은 방법 (테스트 코드)

재현을 위한 테스트 코드는 애플리케이션보다는 라이브러리 프로젝트에서 선호되는 방식

특정 이미지 파일의 경우 너비와 높이가 서로 뒤바뀌어서 나옵니다. 재현을 위한 테스트 코드와 테스트에 쓰인 이미지 파일을 첨부합니다.

버그 리포팅: 좋은 방법 (테스트 코드)

재현을 위한 테스트 코드는 애플리케이션보다 라이브러리 프로젝트에서 선호되는 방식

특정 이미지 파일의 경우 너비와 높이가 서로 뒤바뀌어서 나옵니다. 재현을 위한 테스트 코드와 테스트에 쓰인 이미지 파일을 첨부합니다.

```
img = Image("test-100x200.jpg")
```

버그 리포팅: 좋은 방법 (테스트 코드)

재현을 위한 테스트 코드는 애플리케이션보다 라이브러리 프로젝트에서 선호되는 방식

특정 이미지 파일의 경우 너비와 높이가 서로 뒤바뀌어서 나옵니다. 재현을 위한 테스트 코드와 테스트에 쓰인 이미지 파일을 첨부합니다.

```
img = Image("test-100x200.jpg")
```

```
assert img.size == (100, 200)
```

병합 도구

병합 도구

- **diff**: 전체 소스 코드에서 내가 고친 부분만 추출하는 도구

병합 도구

- **diff**: 전체 소스 코드에서 내가 고친 부분만 추출하는 도구
- **patch**: diff로 추출한 부분을 원래 소스 코드에 합치는 도구

병합 도구

- **diff**: 전체 소스 코드에서 내가 고친 부분만 추출하는 도구
- **patch**: diff로 추출한 부분을 원래 소스 코드에 합치는 도구
- 옛날에는 메일로 원 저자에게 내가 수정한 부분의 diff를 첨부해서 보냈음

병합 도구

- **diff**: 전체 소스 코드에서 내가 고친 부분만 추출하는 도구
- **patch**: diff로 추출한 부분을 원래 소스 코드에 합치는 도구
- 옛날에는 메일로 원 저자에게 내가 수정한 부분의 diff를 첨부해서 보냈음
- 요즘에는 이 둘을 직접 다룰 일은 드물게 되었음

병합 도구

- **diff**: 전체 소스 코드에서 내가 고친 부분만 추출하는 도구
- **patch**: diff로 추출한 부분을 원래 소스 코드에 합치는 도구
- 옛날에는 메일로 원 저자에게 내가 수정한 부분의 diff를 첨부해서 보냈음
- 요즘에는 이 둘을 직접 다룰 일은 드물게 되었음
- 그럼에도 불구하고 둘의 개념과 기본적인 사용법은 숙지하고 있어야 함

병합 도구

- **diff**: 전체 소스 코드에서 내가 고친 부분만 추출하는 도구
- **patch**: diff로 추출한 부분을 원래 소스 코드에 합치는 도구
- 옛날에는 메일로 원 저자에게 내가 수정한 부분의 diff를 첨부해서 보냈음
- 요즘에는 이 둘을 직접 다룰 일은 드물게 되었음
- 그럼에도 불구하고 둘의 개념과 기본적인 사용법은 숙지하고 있어야 함

diff, patch를 쓰면
편해지는데 것

diff, patch를 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?

diff, patch를 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?
1. 압축 파일을 5개를 받아서 모든 변경 사항들을 눈으로 찾아내 손으로 통합된 소스 트리를 한 땀한땀 만들어낸다.

diff, patch를 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?
 1. 압축 파일을 5개를 받아서 모든 변경 사항들을 눈으로 찾아내 손으로 통합된 소스 트리를 한 땀한땀 만들어낸다.
 2. 각자 diff 를 이용해 변경된 부분의 패치를 5개 만들어서 한 곳에서 patch로 합친다.

버전 관리 시스템

버전 관리 시스템

- CVS, Subversion 등 여러가지 버전 관리 시스템이 있었지만 이제는 많이 쓰이지 않음

버전 관리 시스템

- CVS, Subversion 등 여러가지 버전 관리 시스템이 있었지만 이제는 많이 쓰이지 않음
- 오픈 소스 특성상 동시적인 소스 코드의 수정과 병합이 수월한 도구 쪽으로 넘어가게 되어 있음

버전 관리 시스템

- CVS, Subversion 등 여러가지 버전 관리 시스템이 있었지만 이제는 많이 쓰이지 않음
- 오픈 소스 특성상 동시적인 소스 코드의 수정과 병합이 수월한 도구 쪽으로 넘어가게 되어 있음
- 분산형 버전 관리!

버전 관리 시스템을
쓰면 편해지는 것

버전 관리 시스템을 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?

버전 관리 시스템을 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?
1. 압축 파일을 5개를 받아서 모든 변경 사항들을 눈으로 찾아내 손으로 통합된 소스 트리를 한 땀한땀 만들어낸다.

버전 관리 시스템을 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?
 1. 압축 파일을 5개를 받아서 모든 변경 사항들을 눈으로 찾아내 손으로 통합된 소스 트리를 한 땀한땀 만들어낸다.
 2. 각자 diff 를 이용해 변경된 부분의 패치를 5개 만들어서 한 곳에서 patch로 합친다.

버전 관리 시스템을 쓰면 편해지는 것

- 같은 소스 트리에서 다섯 사람이 동시에 다른 부분을 수정하고 나서, 그 모든 것을 합치려면 어떻게 해야 할까?
 1. 압축 파일을 5개를 받아서 모든 변경 사항들을 눈으로 찾아내 손으로 통합된 소스 트리를 한 땀한땀 만들어낸다.
 2. 각자 diff 를 이용해 변경된 부분의 패치를 5개 만들어서 한 곳에서 patch로 합친다.
 3. 각자 수정한 것을 버전 관리 시스템에 커밋한다.

DVCS

DVCS

- Git, Mercurial, Darcs 등이 있음

DVCS

- Git, Mercurial, Darcs 등이 있음
- 과거의 중앙형 버전 관리 시스템(CVS, Subversion)은 버전을 관리하는 반면 분산형 버전 관리 시스템은 변경을 관리

DVCS

- Git, Mercurial, Darcs 등이 있음
- 과거의 중앙형 버전 관리 시스템(CVS, Subversion)은 버전을 관리하는 반면 분산형 버전 관리 시스템은 변경을 관리
- 버전이란 변경된 상태를 뜻하고 변경은 변경하는 행위를 뜻함

DVCS


















- Git, Mercurial, Darcs 등이 있음
- 과거의 중앙형 버전 관리 시스템(CVS, Subversion)은 버전을 관리하는 반면 분산형 버전 관리 시스템은 변경을 관리
- 버전이란 변경된 상태를 뜻하고 변경은 변경하는 행위를 뜻함
- DVCS는 매 버전마다 스냅샷을 찍는 방식

DVCS

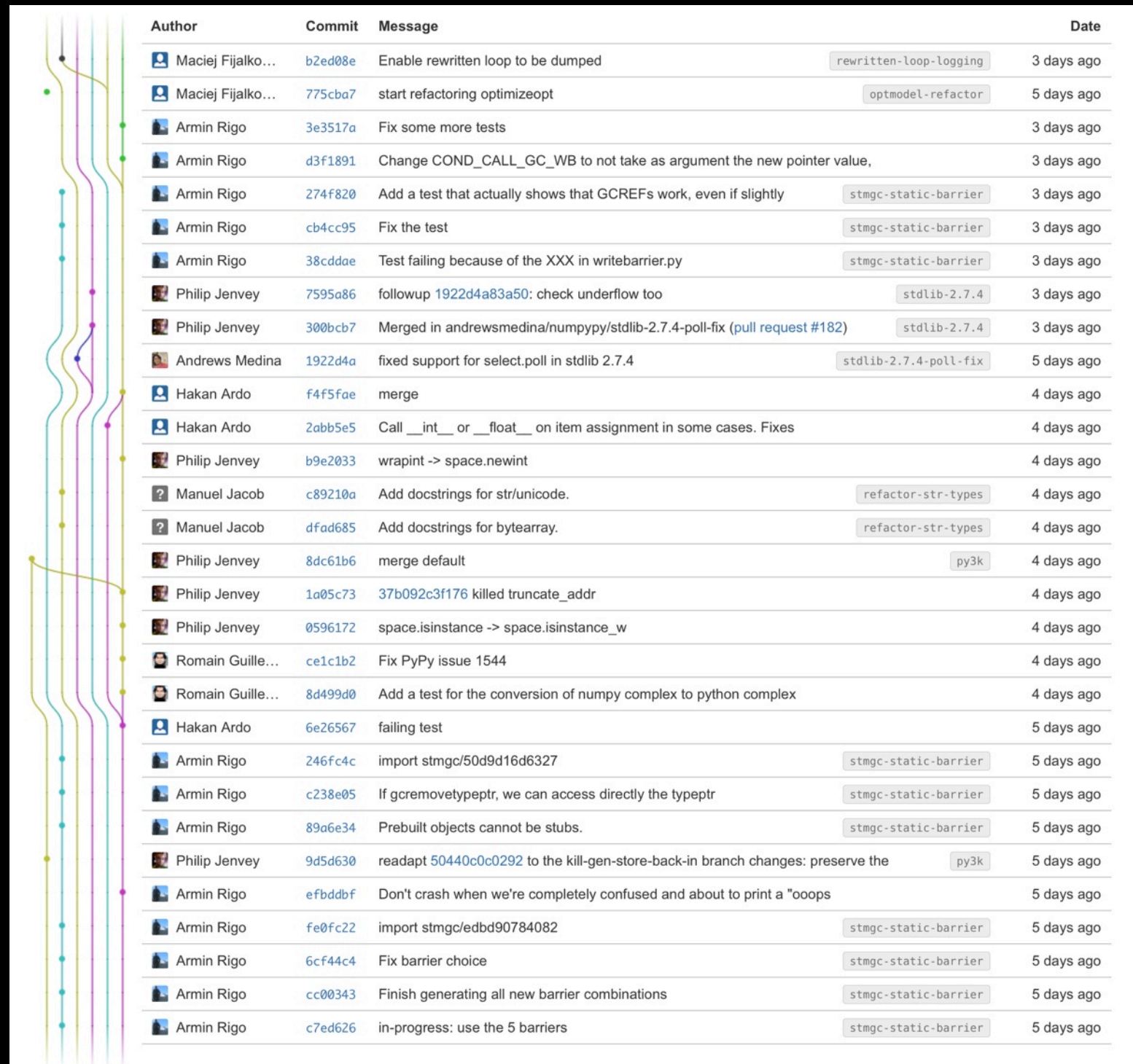
- Git, Mercurial, Darcs 등이 있음
- 과거의 중앙형 버전 관리 시스템(CVS, Subversion)은 버전을 관리하는 반면 분산형 버전 관리 시스템은 변경을 관리
- 버전이란 변경된 상태를 뜻하고 변경은 변경하는 행위를 뜻함
- DVCS는 매 버전마다 스냅샷을 찍는 방식
- DVCS는 diff를 쌓아두는 방식

DVCS

- Git, Mercurial, Darcs 등이 있음
- 과거의 중앙형 버전 관리 시스템(CVS, Subversion)은 버전을 관리하는 반면 분산형 버전 관리 시스템은 변경을 관리
- 버전이란 변경된 상태를 뜻하고 변경은 변경하는 행위를 뜻함
- DVCS는 매 버전마다 스냅샷을 찍는 방식
- DVCS는 diff를 쌓아두는 방식

Rev	Age	Author	Log Message
@25052 	31 hours	dd32	WP_HTTP: Curl: When using Stream-to-file on servers using ...
@25051 	31 hours	dd32	WP_HTTP: Curl: When using Stream-to-file on servers using ...
@25050 	43 hours	ocean90	Introduce save_post_{ \$post_type } hook. The hook is fired before the ...
@25049 	2 days	ocean90	Flush the cache in Recent Comments widget on edit_comment. props pento, ...
@25048 	2 days	dd32	WP_Updater: Don't activate maintenance mode in bulk_upgrade() when no ...
@25047 	2 days	dd32	Make use of the recursive option in mkdir() in wp_mkdir_p(). Avoids a ...
@25046 	3 days	dd32	WP_HTTP: Cookies: When following redirects, include the request cookies in ...
@25045 	3 days	azaozz	In wp-login.php check if cookies are enabled before attempting to log the ...
@25044 	3 days	dd32	WP_HTTP: Cookies: Fill the defaults for the Cookie object based on the ...
@25043 	3 days	duck_	Introduce a notoptions cache for site options. Props wonderboymusic. ...
@25042 	3 days	nacin	Twenty Eleven: Fix some mobile layout bugs on single post format view. ...
@25041 	3 days	nacin	Obey \$prefix when \$display = false in single_term_title(), ...
@25040 	3 days	duck_	Search for wp-util.min.js in the correct directory when auto-defining ...
@25039 	3 days	helen	Make sure quick edited non-alternate list table rows don't become ...
@25038 	3 days	nacin	In populate_network(), insert the first site as blog_id 1, overriding any ...
@25037 	3 days	nacin	Add a function to return an empty string, for filters. props wpsmith, ...
@25036 	3 days	nacin	send_origin_headers() in admin-post.php, props nickdaugherty, fixes

Subversion/CVS: 선형



DVCS: 그래프

DVCS

DVCS

- Git, Mercurial, Darcs 등이 있음

DVCS

- Git, Mercurial, Darcs 등이 있음
- 하지만 기여하려는 프로젝트가 쓰고 있는 것을 써야함

DVCS

- Git, Mercurial, Darcs 등이 있음
- 하지만 기여하려는 프로젝트가 쓰고 있는 것을 써야함
- 우리가 선택할 수 없음

DVCS

- Git, Mercurial, Darcs 등이 있음
- 하지만 기여하려는 프로젝트가 쓰고 있는 것을 써야함
- 우리가 선택할 수 없음
- Git이 가장 많이 쓰임

Git \neq GitHub

Git \neq GitHub

- GitHub은 Git이 아닙니다

Git \neq GitHub

- GitHub은 Git이 아닙니다
- Git은 GitHub이 아닙니다

Git \neq GitHub

- GitHub은 Git이 아닙니다
- Git은 GitHub이 아닙니다
- GitHub은 호스팅 서비스고 Git은 DVCS

Git \neq GitHub

- GitHub은 Git이 아닙니다
- Git은 GitHub이 아닙니다
- GitHub은 호스팅 서비스고 Git은 DVCS
- Web : Chrome/Firefox/IE/...
= Git : GitHub/Bitbucket/...

Git

Git

- Linus Torvalds가 Linux 프로젝트에서 Bitkeeper를 대체하기 위해 만듦

Git

- Linus Torvalds가 Linux 프로젝트에서 Bitkeeper를 대체하기 위해 만듦
- 다른 DVCS에 비해 개발이 활발한 편

Git

- Linus Torvalds가 Linux 프로젝트에서 Bitkeeper를 대체하기 위해 만듦
- 다른 DVCS에 비해 개발이 활발한 편
- 기능이 굉장히 방대하지만 반대로 중구난방인 면도 있음

GitHub

GitHub

- Git의 대중화에 큰 기여를 한 호스팅 서비스

GitHub

- Git의 대중화에 큰 기여를 한 호스팅 서비스
- Git 저장소 외에도 이슈 트래커와 위키 등을 제공

GitHub

- Git의 대중화에 큰 기여를 한 호스팅 서비스
- Git 저장소 외에도 이슈 트래커와 위키 등을 제공
- 오픈 소스 프로젝트에게는 프로모션 효과도 있음

GitHub

- Git의 대중화에 큰 기여를 한 호스팅 서비스
- Git 저장소 외에도 이슈 트래커와 위키 등을 제공
- 오픈 소스 프로젝트에게는 프로모션 효과도 있음
- 그 전까지 fork라는 단어에 있던 부정적인 뉘앙스를 없앴

GitHub

- Git의 대중화에 큰 기여를 한 호스팅 서비스
- Git 저장소 외에도 이슈 트래커와 위키 등을 제공
- 오픈 소스 프로젝트에게는 프로모션 효과도 있음
- 그 전까지 fork라는 단어에 있던 부정적인 뉘앙스를 없앴
- Fork & pull request!

GitHub

- Git의 대중화에 큰 기여를 한 호스팅 서비스
- Git 저장소 외에도 이슈 트래커와 위키 등을 제공
- 오픈 소스 프로젝트에게는 프로모션 효과도 있음
- 그 전까지 fork라는 단어에 있던 부정적인 뉘앙스를 없앴
- Fork & pull request!



Fork

68

Send pull request

풀 리퀘스트

풀 리퀘스트

- 이전까지 메일링 리스트나 이슈 트래커에 패치를 첨부하던 오래된 관습을 거의 대체

풀 리퀘스트

- 이전까지 메일링 리스트나 이슈 트래커에 패치를 첨부하던 오래된 관습을 거의 대체
- 기여할 프로젝트의 저장소를 포크

풀 리퀘스트

- 이전까지 메일링 리스트나 이슈 트래커에 패치를 첨부하던 오래된 관습을 거의 대체
- 기여할 프로젝트의 저장소를 포크
- 내 포크된 저장소에 커밋

풀 리퀘스트

- 이전까지 메일링 리스트나 이슈 트래커에 패치를 첨부하던 오래된 관습을 거의 대체
- 기여할 프로젝트의 저장소를 포크
- 내 포크된 저장소에 커밋
- 업스트림 저장소에 풀 리퀘스트를 보냄

풀 리퀘스트

- 이전까지 메일링 리스트나 이슈 트래커에 패치를 첨부하던 오래된 관습을 거의 대체
- 기여할 프로젝트의 저장소를 포크
- 내 포크된 저장소에 커밋
- 업스트림 저장소에 풀 리퀘스트를 보냄
- 업스트림 메인테이너가 패치를 리뷰하고
관찮으면 풀(머지)

빌드 자동화 (CI)

빌드 자동화 (CI)

- Continuous Integration

빌드 자동화 (CI)

- Continuous Integration
- 매 커밋마다 빌드, 테스트, 패키징, 문서 빌드 등을 수행

빌드 자동화 (CI)

- Continuous Integration
- 매 커밋마다 빌드, 테스트, 패키징, 문서 빌드 등을 수행
- 항상 릴리즈 가능한 상태에 가깝게 유지

빌드 자동화 (CI)

- Continuous Integration
- 매 커밋마다 빌드, 테스트, 패키징, 문서 빌드 등을 수행
- 항상 릴리즈 가능한 상태에 가깝게 유지
- 신규 기여자가 실수하는 것을 방지하는 효과도 있음

빌드 자동화 (CI)

- Continuous Integration
- 매 커밋마다 빌드, 테스트, 패키징, 문서 빌드 등을 수행
- 항상 릴리즈 가능한 상태에 가깝게 유지
- 신규 기여자가 실수하는 것을 방지하는 효과도 있음
- Buildbot, Jenkins 같은 것을 직접 서버에 설치해서 쓰기도 하지만, 요즘에는 Travis CI라는 오픈 소스 프로젝트를 위한 무료 CI 서비스도 많이 쓰임

빌드 자동화 (CI)

- Continuous Integration
- 매 커밋마다 빌드, 테스트, 패키징, 문서 빌드 등을 수행
- 항상 릴리즈 가능한 상태에 가깝게 유지
- 신규 기여자가 실수하는 것을 방지하는 효과도 있음
- Buildbot, Jenkins 같은 것을 직접 서버에 설치해서 쓰기도 하지만, 요즘에는 Travis CI라는 오픈 소스 프로젝트를 위한 무료 CI 서비스도 많이 쓰임

earthreader/libearth


Libearth is the core implementation of Earth Reader, without any frontends.















Current

Build History

Pull Requests

Branch Summary



Build	Message	Commit	Committer	Duration	Finished
 70	Common codecs	1e0e99e (master)	Hong Minhee	3 min 59 sec	3 days ago
 69	Codec	efbb695 (master)	Hong Minhee	51 min 30 sec	3 days ago
 66	More docstrings	3de62da (master)	Hong Minhee	3 min 19 sec	4 days ago
 65	Fixed-offset timezone	45a4f34 (master)	Hong Minhee	4 min 2 sec	4 days ago
 64	CodecError, EncodeError, DecodeError	98d281b (master)	Hong Minhee	4 min 6 sec	4 days ago
 62	Use IntegrityError instead of SyntaxError	36d1ea1 (master)	Hong Minhee	3 min 23 sec	5 days ago
 61	Self-referential children	9a1baea (master)	Hong Minhee	3 min 33 sec	5 days ago
 60	Element validation according to schema	697d77c (master)	Hong Minhee	3 min 36 sec	5 days ago
 56	Allow slicing on ElementList	9fbf6ce (master)	Hong Minhee	3 min 8 sec	6 days ago
 55	ElementList becomes mutable	15054ef (master)	Hong Minhee	3 min 39 sec	6 days ago
 51	Encoder	1e66280 (master)	Hong Minhee	3 min 9 sec	8 days ago
 50	Python 2.6 has no fromstringlist() function	6f9e491 (master)	Hong Minhee	3 min 5 sec	8 days ago
 49	Fix bug that crashes when there's non-ASCII chars	60552b1 (master)	Hong Minhee	3 min 7 sec	8 days ago
 48	Missing docstring for write() function	2ce1d22 (master)	Hong Minhee	3 min 33 sec	9 days ago

earthreader/libearth

Libearth is the core implementation of Earth Reader, without any frontends.

Current


Build History

Pull Requests






Branch Summary

Build #49

 ▼

Build	 49	Commit	60552b1 (master)
State	Failed	Author	Hong Minhee
Finished	8 days ago	Committer	Hong Minhee
Duration	3 min 7 sec		
Message	Fix bug that crashes when there's non-ASCII chars		
Config	-		

Build Matrix

Job	Duration	Finished	Python
 49.1	35 sec	8 days ago	2.6
 49.2	34 sec	8 days ago	2.7
 49.3	38 sec	8 days ago	3.2
 49.4	36 sec	8 days ago	3.3
 49.5	44 sec	8 days ago	pypy

오픈 소스 예절

오픈 소스 예절

- 오픈 소스도 결국 사람이 하는 일

오픈 소스 예절

- 오픈 소스도 결국 사람이 하는 일
- 예의 있게 행동해야 함

오픈 소스 예절

- 오픈 소스도 결국 사람이 하는 일
- 예의 있게 행동해야 함
- 버그 리포팅 시에는 언제나 좋은 소프트웨어를 만들어준 것에 대한 감사를 표시

오픈 소스 예절

- 오픈 소스도 결국 사람이 하는 일
- 예의 있게 행동해야 함
- 버그 리포팅 시에는 언제나 좋은 소프트웨어를 만들어준 것에 대한 감사를 표시
- 패치, 풀 리퀘스트 제출 시에는 항상 코드를 리뷰해줘서 감사하다고 표현

오픈 소스 예절

오픈 소스 예절

- 반대로, 남이 자신의 프로젝트에 패치를 보내줬을 때도 감사를 표시할 것!

오픈 소스 예절

- 반대로, 남이 자신의 프로젝트에 패치를 보내줬을 때도 감사를 표시할 것!
- 과거에는 함부로 프로젝트를 포크하는 것은 대놓고 상대방을 존중하지 않는 무례한 행동이었지만, 지금은 GitHub이 포크에 대한 인식을 좋은 것으로 바꿨음

눈치 보기

눈치 보기

- 메인테이너 눈밖에 나서 좋을 것은 하나도 없음

눈치 보기

- 메인테이너 눈밖에 나서 좋을 것은 하나도 없음
- 괜히 자신만의 코딩 스타일 고수하지 말 것

눈치 보기

- 메인테이너 눈밖에 나서 좋을 것은 하나도 없음
- 괜히 자신만의 코딩 스타일 고수하지 말 것
- 코딩 스타일은 합의적 기능이 가장 크므로, 서로 일치하지 않는다면 애초에 지킬당 위성이 많이 떨어짐

눈치 보기

눈치 보기

오픈 소스 세상에서 너무 자신만의 코딩 스타일을 고수하면 중2병 정도로 보임

눈치 보기

오픈 소스 세상에서 너무 자신만의 코딩 스타일을 고수하면 중2병 정도로 보임

“크큭... 난 나만의 스타일이 있다구!”

예뵤 받기

예뵐 받기

- 이슈 트래커의 해묵은 버그를 고친다던가

예뵐 받기

- 이슈 트래커의 해묵은 버그를 고친다던가
- 다른 사람의 패치를 나서서 리뷰한다던가

예뵐 받기

- 이슈 트래커의 해묵은 버그를 고친다던가
- 다른 사람의 패치를 나서서 리뷰한다던가
- 메일링 리스트 등에서 이슈로 떠오른 문제를 먼저 해결한다던가

예뵤 받기

- 이슈 트래커의 해묵은 버그를 고친다던가
- 다른 사람의 패치를 나서서 리뷰한다던가
- 메일링 리스트 등에서 이슈로 떠오른 문제를 먼저 해결한다던가
- 하면 메인테이너가 눈여겨 보게 됨

예뵘 받기

- 이슈 트래커의 해묵은 버그를 고친다던가
- 다른 사람의 패치를 나서서 리뷰한다던가
- 메일링 리스트 등에서 이슈로 떠오른 문제를 먼저 해결한다던가
- 하면 메인테이너가 눈여겨 보게 됨
- 커미터를 딸 수 있는 지름길!

영어

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음
- 오픈 소스 프로젝트에는 국경이 없기 때문

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음
- 오픈 소스 프로젝트에는 국경이 없기 때문
- Nginx는 러시아 사람이 만들었지만 러시아 프로젝트가 아님

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음
- 오픈 소스 프로젝트에는 국경이 없기 때문
- Nginx는 러시아 사람이 만들었지만 러시아 프로젝트가 아님
- Netty는 한국의 이희승 씨가 만들었지만 한국 프로젝트가 아님

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음
- 오픈 소스 프로젝트에는 국경이 없기 때문
- Nginx는 러시아 사람이 만들었지만 러시아 프로젝트가 아님
- Netty는 한국의 이희승 씨가 만들었지만 한국 프로젝트가 아님
- 둘 다 공식 채널에서 러시아어나 한국어 쓰는 일 없음

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음
- 오픈 소스 프로젝트에는 국경이 없기 때문
- Nginx는 러시아 사람이 만들었지만 러시아 프로젝트가 아님
- Netty는 한국의 이희승 씨가 만들었지만 한국 프로젝트가 아님
- 둘 다 공식 채널에서 러시아어나 한국어 쓰는 일 없음
- 오픈 소스 활동을 하려면 영어를 쓸 수밖에 없긴 함!

영어

- “국산” 오픈 소스 프로젝트 같은 것은 없음
- 오픈 소스 프로젝트에는 국경이 없기 때문
- Nginx는 러시아 사람이 만들었지만 러시아 프로젝트가 아님
- Netty는 한국의 이희승 씨가 만들었지만 한국 프로젝트가 아님
- 둘 다 공식 채널에서 러시아어나 한국어 쓰는 일 없음
- 오픈 소스 활동을 하려면 영어를 쓸 수밖에 없긴 함!

실용 영어

실용 영어

- 중세 시대와 근대 유럽의 학문용 언어였던 라틴어는 대부분의 유명한 학자에게도 모국어가 아니었음

실용 영어

- 중세 시대와 근대 유럽의 학문용 언어였던 라틴어는 대부분의 유명한 학자에게도 모국어가 아니었음
- 그럼에도 불구하고 그들은 라틴어의 부분 집합인 “실용 라틴어”로 훌륭한 논문을 썼음

실용 영어

- 중세 시대와 근대 유럽의 학문용 언어였던 라틴어는 대부분의 유명한 학자에게도 모국어가 아니었음
- 그럼에도 불구하고 그들은 라틴어의 부분 집합인 “실용 라틴어”로 훌륭한 논문을 썼음
- 오픈 소스도 마찬가지로, 매우 유창한 영어를 요구하는 것이 아님

실용 영어

- 중세 시대와 근대 유럽의 학문용 언어였던 라틴어는 대부분의 유명한 학자에게도 모국어가 아니었음
- 그럼에도 불구하고 그들은 라틴어의 부분 집합인 “실용 라틴어”로 훌륭한 논문을 썼음
- 오픈 소스도 마찬가지로, 매우 유창한 영어를 요구하는 것이 아님
- 영어를 잘 하는 것보다, 영어를 피하려는 태도를 없애는 게 훨씬 중요

실용 영어

- 중세 시대와 근대 유럽의 학문용 언어였던 라틴어는 대부분의 유명한 학자에게도 모국어가 아니었음
- 그럼에도 불구하고 그들은 라틴어의 부분 집합인 “실용 라틴어”로 훌륭한 논문을 썼음
- 오픈 소스도 마찬가지로, 매우 유창한 영어를 요구하는 것이 아님
- 영어를 잘 하는 것보다, 영어를 피하려는 태도를 없애는 게 훨씬 중요
- 차라리 테스트 코드를 잘 작성하는 쪽이 서로에게 편한 경우도

실용 영어

- 중세 시대와 근대 유럽의 학문용 언어였던 라틴어는 대부분의 유명한 학자에게도 모국어가 아니었음
- 그럼에도 불구하고 그들은 라틴어의 부분 집합인 “실용 라틴어”로 훌륭한 논문을 썼음
- 오픈 소스도 마찬가지로, 매우 유창한 영어를 요구하는 것이 아님
- 영어를 잘 하는 것보다, 영어를 피하려는 태도를 없애는 게 훨씬 중요
- 차라리 테스트 코드를 잘 작성하는 쪽이 서로에게 편한 경우도



Salvatore Sanfilippo

@antirez



Follow

Many pull requests from China: ten words, a small example, very useful fixes for hard to find bugs. More please.



Reply



Retweet



Favorite



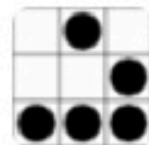
More

19

RETWEETS

7

FAVORITES



1:17 AM - 4 Feb 13

 [-] **antirez** 17 points 4 years ago

Every day tons of developers that are not native English speakers try to do their best to make one thing happen: a world wide network of programmers. They comment on hacker news and programming reddit, write open source code commented in english, blog posts, and so on. Most of the times their English is not perfect but understandable and probably much better than the average US or UK guy understanding of a second language.

So what about appreciating this effort instead of focusing on errors? (p.s. at least do the effort of expressing lengths in meters and weights in Kg).

[permalink](#) [parent](#) [give gold](#)

끝

질문?

bit.ly/soma-2013-08-20-oss