소셜 코딩으로 이끄는 GitHub 실천 기술

GitHub JISSEN-NYUMON
by Hiroki Otsuka

소셜 코딩으로 이끄는 GitHub 실천 기술

초판 1쇄 발행 2015년 1월 31일

지은이 오오츠카 히로키
옮긴이 윤인성
펴낸이 장성두
펴낸곳 제이펍

출판신고 2009년 11월 10일 제406-2009-000087호
주소 경기도 파주시 문발로 141 뮤즈빌딩 403호
전화 070 – 8201 – 9010 / **팩스** 02 – 6280 – 0405
홈페이지 www.jpub.kr / **이메일** jeipub@gmail.com

편집부 이민숙, 이 슬, 이주원 / **소통·기획팀** 현지환
본문디자인 북아이 / **표지디자인** 미디어픽스
용지 신승지류유통 / **인쇄** 해외정판사 / **제본** 광우제책사

ISBN 979-11-85890-10-4 (93000)
값 26,000원

소셜 코딩으로 이끄는 GitHub 실천 기술

Git과 GitHub를 직접 따라하며 배운다

Jpub 제이펍

※ 드리는 말씀

- 이 책에 기재된 내용을 기반으로 한 운용 결과에 대해 저자, 역자, 소프트웨어 개발자 및 제공자, 제이펍 출판사는 일체의 책임을 지지 않으므로 양해 바랍니다.
- 이 책에 등장하는 각 회사명, 제품명은 일반적으로 각 회사의 등록 상표 또는 상표입니다. 본문 중에는 ™, ©, ® 마크 등이 표시되어 있지 않습니다.
- 이 책에서 사용하고 있는 제품 버전은 독자의 학습 시점이나 환경에 따라 책의 내용과 다를 수 있습니다.
- 본문 중 일본 내의 실정에만 국한되어 있는 내용이나 그림은 일부를 삭제하거나 국내 실정에 맞도록 변경하였으니 참고 바랍니다.
- 이 책의 소스 코드는 제이펍 홈페이지(www.jpub.kr)의 상세 소개 페이지에서 다운로드하실 수 있습니다.
- 책 내용과 관련된 문의사항은 역자나 출판사로 연락해 주시기 바랍니다.
  - 역　자: rintiantta@naver.com
  - 출판사: jeipub@gmail.com

차 례

CHAPTER 6

Pull Request를 해보자 _ 147

CHAPTER 7

Pull Request가 도착한다면 _ 165

CHAPTER 8

GitHub와 연계되는 툴과 서비스 _ 179

CHAPTER 10

회사에서 GitHub 사용하기 _ 289

APPENDIX A

GitHub GUI 클라이언트 _ 301

APPENDIX B

코드를 Gist로 쉽게 공유하기 _ 307

옮긴이 머리말

웹 개발에 관심이 있는 사람이라면 jQuery, angualr.js, bootstrap, backbone.js, node.js, rails, django, cakePHP, sinatra, flask, play, symphony, bower, CoffeeScript를 들어 보았을 것이며, 데이터베이스에 관심이 있는 사람이라면 MariaDB, MongoDB, Redis, Neo4j, CouchDB 등을 들어본 적이 있을 것입니다. 또한 루비, 파이썬, 노드를 사용한 독자라면 루비젬(rubygem), pip, npm을 사용해본 적이 있을 것이며, 맥으로 무언가를 개발하고 있다면 홈브류(homebrew)를 사용해 보았을 것입니다. 이 소프트웨어들의 공통점을 꼽는다면 어떤 것이 있을까요? 그것은 바로 GitHub를 중심으로 사람들이 모여 개발했다는 점입니다.

GitHub는 Git 리포지토리를 호스팅해 주는 서비스입니다. 일부 개발자들이 Git과 GitHub를 같은 것이라 생각하는 경우가 종종 있습니다만, 사실 이들은 완전히 다른 용어입니다.

이렇게 말하면 독자분들께서는 "그럼 이 책은 GitHub 책이니까 Git을 따로 공부하고 봐야겠네요?"라고 생각하실 수 있습니다. 그러나 이 책에도 일반 실무에서 쓰이는 Git 관련 내용은 모두 포함되어 있으므로 읽으면서 공부하면 됩니다. 참고로 현재 국내에도 약간의 Git 책이 출간되어 있긴 하지만, GitHub 책은 나와 있지 않습니다(여러 출판사에서 GitHub 책을 집필하려고 저자를 섭외하고 있지만 아직 소식이 들려오지 않네요). 때문에 이 책이 출간된다면 국내에서 첫 번째 GitHub 관련 도서가 될 것입니다.

이 책의 원서를 처음 접했던 것은 지난 3월 일본 여행을 갔을 때로, 아키하바라에서 일주일간 체류하던 중 요도바시 카메라에 있는 서점에서 베스트셀러 목록을 발견하게 되었습니다. 헌데 일반적으로 경제경영, 인문학 책이 대부분인 해당 목록

에 GitHub를 다루는 책이 올라가 있다는 것이 신기해서 잠시 앉아서 책을 살펴보았었는데, 그 후 국내에는 GitHub 책이 존재하지 않는다는 것을 깨닫고 '어떻게 GitHub를 다룬 서적이 하나도 없을 수가 있나' 하고 개탄하며 외국의 여러 GitHub 책들을 뒤져보다가 결국엔 역시 이 책을 번역하는 것이 좋겠다는 마음을 먹게 되었네요.

책에 대해 간략하게 소개를 해보자면, 1장은 기본적인 소개 내용이고 2장부터 5장까지의 내용은 일반적인 기본 Git 내용입니다. 타 Git 도서와 GitHub 책에서도 볼 수 있는 부분입니다만, 다소 과하다 싶을 정도로 꼼꼼하고 자세히 쓰여 있습니다. 너무 자세해서 '편리하긴 하지만 일반적으로는 파악하기 힘든 기능'까지 확인할 수 있을 지경입니다. 그리고 6장에서는 GitHub의 핵심 기능인 Pull Request를 독자가 직접 보내는 경험을 하도록 돕기도 합니다. 진짜로 Pull Request를 보내면 저자가 반영해 주는 방식입니다(한국에서는 역자가 해드립니다. 혼자 해내느라 대응이 늦을 수도 있겠습니다만, 최선을 다하겠습니다). 저자가 직접 반영해 준다는 점을 제외하면 다른 GitHub 책에서도 쉽게 볼 수 있는 내용입니다만, 그와 달리 8장부터 10장은 읽다가 서점에서 코피를 터뜨리며 "그래! 이 책을 번역해야겠어!" 하고 결심하게 된 결정적 계기를 만들어 주었는데요. 저자가 실무에서 오랫동안 GitHub를 활용해 본 경험이 풍부한지라 실무에서 어떤 식으로 활용할 수 있는지에 대해 타의 추종을 불허할 만큼 정말 자세히 설명되어 있습니다. 이 책의 핵심 부분이라고도 할 수 있겠네요.

책을 번역하고 있는 중이었던 8월에 다시 일본 서점에 갔을 때도 베스트셀러 도서 책장에 여전히 이 책이 꽂혀 있었습니다. 그리고 역자 서문을 쓰고 있는 현재 11월에도 아마존 베스트셀러에 머물러 있네요. 어째서 이 책이 일본에서 오랜 기간 베스트셀러로 사랑을 받는지, 어떤 연유로 역자인 제가 서점에서 코피까지 터뜨렸는지는 직접 책을 읽으면서 확인하시기 바랍니다.

참고로 번역서를 접하실 독자의 이해도를 높이기 위해 번역하면서 한 가지 규칙을 정했습니다. GitHub라는 사이트 자체가 대부분 영어로 구성되어 있는지라 Pull Request, Fork 외에 Git을 사용하면서 자주 접하게 될 pull, push, checkout, merge 등도 번역하면서 모두 일본어를 모두 영어 또는 영문을 한글로 음차하여 표

기하였습니다(예: マージ를 merge로 번역).

이 책을 번역할 기회를 주신 제이펍 장성두 실장님과 현지환 대리님, 담당자 이주원 님께 감사를 표하며, 또한 책 교정에 도움을 주신 윤아현, 안광섭, 조선미, 송종근, 장창이, 김주아 님께도 감사드립니다.

2015년 1월

윤인성

이 책은 전 세계의 수많은 개발자가 사용하고 있는 GitHub를 실무에서 어떻게 사용하는지 설명하는 책입니다. 따라서 GitHub의 기본적인 사용 방법뿐만 아니라, GitHub를 활용한 개발 진행 과정과 개발을 지원해 주는 추가적인 도구들도 함께 설명합니다.

실무에서 코드를 개발하면서 다음과 같은 생각 또는 행동을 한 적이 있나요?

- 코드 리뷰가 충분하지 않고, 리뷰가 느리다고 생각한 경우
- 작성한 본인밖에 모르는 코드, 불안한 느낌의 코드가 실제 환경에서 Deploy한 경우
- 코드 입력 오류, 스스로 착각에 빠져서 잘못된 코드를 작성한 경우
- 코드를 서로서로 리뷰하며 지식 공유, 상호 학습, 지적, 개선하는 기회가 없는 경우
- 하루에도 여러 개의 기능을 추가할 수 있는 빠른 개발 진행 과정이 도입되지 않은 경우

GitHub를 활용하면 이러한 문제를 개선할 수 있을 것입니다.

GitHub는 실제 개발 현장에서 일어나는 다양한 문제를 해결할 수 있는 기능을 제공하는데, 이 책에는 그런 문제 해결을 위한 기능을 실무에서 어떻게 활용하느냐에 대한 노하우가 가득 담겨 있습니다.

여러 기업에서 GitHub로 다양한 개발 진행 과정을 개선해 왔던 저자의 풍부한 경험을 토대로 정리한 책이므로, 실무에서 GitHub를 활용하는 데 많은 도움이 될 것입니다.

감사의 말

이 책을 집필하면서 많은 사람들의 도움을 받았습니다. @yamanetoshi 님, 마쓰다 타카시(@masutaka) 님, bakorer 님, @ainame 님, 야마시나 유우키 님, 테라다 와타루 님, Tatsuma Murase 님, 스기노 야스히로 님, 사와 요시카즈 님 모두 감사드립니다.

또한, 책과 관련하여 기술평론사의 이케다 히로키 님에게 오랜 시간 동안 도움을 받았습니다. 진심으로 감사드립니다.

오오츠카 히로키

이 책은 열 개의 장과 두 개의 부록으로 구성되어 있습니다.

● 1장 GitHub 세계에 어서오세요

GitHub가 무엇이고, 무엇이 혁신적인지, 어떤 기능을 제공하는지 등을 설명합니다. GitHub는 오픈 소스 소프트웨어 세계에 혁명이라 부를 수 있는 소셜 코딩의 개념을 제공했는데요. 소셜 코딩이 무엇이며, 이것을 하면 어떤 혜택이 있는지도 설명합니다.

● 2장 Git 기본

GitHub를 사용하려면 버전 관리 시스템 Git을 알아야 합니다. Git의 기본적인 개념을 배우고, 설치와 설정을 수행합니다.

● 3장 GitHub 사용 준비

GitHub 계정(무료)을 작성하고 기본적인 설정을 수행합니다. 또한, 실제로 작동되는지 동작을 확인합니다. GitHub에서 리포지토리를 작성하고 코드를 공개하는 방법도 설명합니다.

● 4장 Git을 직접 사용하면서 배우기

GitHub를 사용하는 데 반드시 필요한 Git을 직접 코드를 입력해 가면서 배웁니다. 기본적인 조작 방법부터 여러 사람이 함께 개발할 때에 필요한 조작까지 직접 입력해 보기 바랍니다.

● 5장 GitHub의 기능을 확실하게 알아보자

화면을 보면서 GitHub의 기능을 하나하나 설명하고, 소스 코드 확인 기능도 자세히 설명합니다. 이미 GitHub를 사용하는 사람들도 한번 훑어보기 바랍니다. 곧바로 사용할 수 있는 팁 등을 찾을 수 있을 것입니다.

● 6장 Pull Request를 해보자

GitHub를 대표하는 기능인 Pull Request를 설명합니다. 직접 따라 하면서 확인할 수 있게 구성했으므로 Pull Request를 보내보기 바랍니다.

● 7장 Pull Request가 도착한다면

Pull Request가 들어온 경우에 어떻게 해야 하는지 리포지토리를 관리하는 사람의 입장이 되어 설명합니다.

● 8장 GitHub와 연계되는 툴과 서비스

앞부분에서는 CLI 환경에서 GitHub를 쉽게 조작할 수 있게 해주는 hub 명령어를 설명합니다. 그리고 뒷부분에서는 지속적 통합(CI)을 GitHub와 연동할 수 있는 Travis CI, Jenkins의 구축과 설정 방법을 설명합니다. 또한, GitHub와 연동할 수 있는 다른 서비스들도 소개합니다.

● 9장 GitHub를 사용하는 경우의 개발 진행 과정

GitHub를 중심으로 개발하는 GitHub Flow, Git Flow라는 두 개의 개발 진행 과정을 설명합니다. 각각의 개발 진행 과정을 팀에서 활용할 때의 기본적인 방식, 각 개발 진행 과정의 특징을 설명하고 직접 코드를 입력하면서 수행해 봅니다.

● 10장 회사에서 GitHub 사용하기

회사에서 GitHub를 도입할 때 생각할 것과 도움이 되는 정보를 정리했습니다. 보안, 장애 정보, GitHub Enterprise 등 실제로 GitHub를 회사에서 도입할 때 알아야 하는 것들과 노하우를 설명합니다.

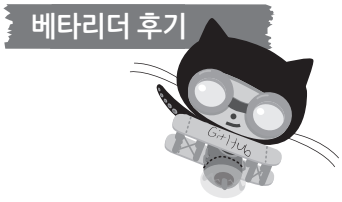
● 부록 A GitHub GUI 클라이언트

팀원 모두가 CLI 조작에 익숙한 것은 아니므로 GitHub를 지원하는 GUI 클라이언트 도구도 소개합니다.

● 부록 B Gist로 코드를 쉽게 공유하기

샘플 코드 또는 로그를 다른 사람과 공유할 때 편리하게 사용할 수 있는 Gist를 설명합니다. 작은 코드는 일반적으로 Gist를 활용하면 편리합니다.

베타리더 후기



김민수(프리랜서)

VCS와 Git, 그리고 최근의 대세라는 Github에 관심과 궁금함은 있는데 아직 입문도, 활용도 하지 못하고 있었습니다. 오직 빨리 적용해 보고 싶은 마음만 앞서는 상태에서 이 책을 만났고, 이도 저도 못하고 있던 제게 아주 적절한 길라잡이가 되어 주었어요. 매우 친절히 설명해 주고, 과정을 찬찬히 밟아 나가는 도중에도 부분마다 깨알 같은 조언도 놓치지 않고 담아낸 점이 인상적이었습니다.

김태경(다음카카오)

GitHub는 옵션이 아닌 BugTracker처럼 프로젝트의 진행을 위한 필수 제품입니다. 그러나 git 명령어 책은 많아도 GitHub에서 Pull Request를 어떻게 해야 하는지 알려주는 책은 찾기 힘들었습니다. 이 책을 마지막까지 읽는다면 이와 같은 궁금증을 해소할 수 있는 것은 물론이고, 많은 GitHub(+엔터프라이즈)를 사용하는 IT 기업들의 개발 방법론을 습득한 자신을 발견하게 될 것입니다.

백경윤(다음카카오)

이 책은 굉장히 쉽고 친절한 설명으로 소셜 코딩이 그리 어려운 일이 아니라는 자신감을 심어 줍니다. 평소 GitHub에 관심은 있었지만 사용해 보기가 어려웠다거나 거리감이 느껴졌다면, 이 책이 큰 도움이 되어 줄 것입니다. 베타리딩을 통해 저도 GitHub에 대하여 좀 더 자세히 깨우치는 좋은 기회가 되었으며, 주위에 더욱더 GitHub를 알리고 널리 전파하는 중입니다. 꼭 한 번 읽어 보세요.

송영준(ZUM internet)

적은 페이지 수에도 불구하고 Git을 중심으로 GitHub, branch 등의 활용 전략과 Git과 연계되는 소프트웨어에 대해 제법 포괄적으로 다루고 있습니다. 그래서 부족

한 부분도 다소 있으나 자주 사용하는 부분을 중심으로 하여 전체를 훑으므로 평소 GitHub에 접근하기 어려워했다거나 Git을 처음 시작할 때 이 책이 좋은 지침이 될 수 있을 것입니다.

이아름

많이 들어는 봤지만 직접 쓸 일이 없던 것 중 하나가 바로 GitHub였습니다. 처음 입문하기에는 어렵다 싶어 주저했는데, 이 책으로 차근차근 따라 해보니 쉽게 이해할 수 있었습니다.

이원제(레진엔터테인먼트)

이 책을 베타리딩하면서 저 역시 GitHub를 사용한 지 오래되었지만, 익숙하지 않았던 것들을 다시 한 번 복습하고 몰랐던 기능들을 찾아보는 좋은 기회가 되었습니다. 특히, GitHub와 연동되는 다른 툴들의 소개는 무척 유익했습니다. 이 책을 통해 혼자서만 코드를 관리하던 개발자들이 다른 개발자들과 협업하는 방법에 익숙해지기를 기대해 봅니다. 또한, 처음 GitHub를 접하는 사람들은 무턱대고 GitHub가 뭐냐고 막연히 묻지만 말고, 이 책을 꼭 읽어 볼 것을 추천합니다.

최해성(티켓몬스터)

입문서로는 괜찮은 도서로, Git의 기본 사용법과 브랜치에 대한 내용을 그림을 통해 쉽게 알려 주는 부분은 정말 좋았습니다. 또한, Pull Request를 하는 방법은 오픈소스 진영에 참여하는 데 있어 그 장벽을 상당히 낮추었다는 생각이 듭니다. 다만, 적은 분량 안에서 여러 가지를 하려다 보니 다소 아쉬운 부분들이 있었습니다. 어려운 개념의 툴은 가볍게 소개하는 정도만 하고, 기본 개념을 심화하는 편이 더 좋지 않았을까 합니다.

제이펍은 책에 대한 애정과 기술에 대한 열정이 뜨거운 베타리더들로 하여금
출간되는 모든 서적에 사전 검증을 시행하고 있습니다.

CHAPTER 1

GitHub 세계에 어서오세요

GitHub란 무엇일까요? 어째서 세계의 많은 개발자들이 GitHub를 사용하고 있는 것일까요? 이번 장에서는 GitHub가 전 세계적으로 미친 영향을 살펴보도록 합시다.

1.1 GitHub란?

GitHub는 친구, 동료는 물론 낯선 사람과 함께 코드를 공유하고자 만든 Git 리포지토리의 호스팅 서비스입니다.

GitHub 회사와 octocat

GitHub는 미국 샌프란시스코에 있는 회사입니다. GitHub는 octocat이라고 불리는 문어와 고양이를 합쳐놓은 것 같은 마스코트 캐릭터(그림 1.1)도 있습니다. octocat은 그림 1.2처럼 다양하게 변형된 모습들이 존재합니다[주1].

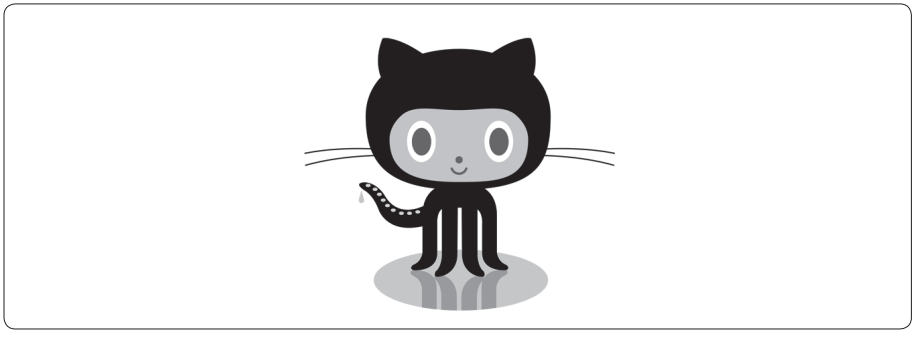


그림 1.1 octocat

주1 https://octodex.github.com/

그림 1.2 octocats

일반적인 Git 리포지토리 호스팅 서비스가 아니다

GitHub는 Git 리포지토리(repository) 호스팅 기능 이외에도 개발자와 팀이 빠른 속도로 좋은 품질의 코드를 만들어 낼 수 있도록 하는 기능도 제공합니다. 이러한 기능들은 다음 장부터 자세히 설명하겠습니다.

GitHub는 초기에 창업자 중 한 명인 크리스 완스트레스(Chris Wanstrath)가 친구들과 함께 코드를 쉽게 공유할 수 있는 Git 리포지토리 호스팅 서비스를 필요로 한 덕에 만들어지게 되었습니다. 이것이 GitHub란 프로덕트의 첫 번째 목표였다는 것은 그가 발표했던 프레젠테이션에 잘 나타나 있습니다[주2].

주2 http://www.slideshare.net/rubymeetup/inside-github-with-chris-wanstrath

GitHub 이용 형태

GitHub는 2014년 12월 기준으로 1,780개 이상의 리포지토리를 호스팅했습니다[주3]. 전 세계의 많은 개발자들이 밤낮으로 연일 이용하고 있는 서비스입니다.

Column GitHub와 Git의 차이

GitHub와 Git의 차이에 대해서 살펴보겠습니다. GitHub와 Git은 완전히 다른 것으로, 이 책에서는 GitHub와 Git을 계속 구별해서 표기합니다.

Git은 Git 리포지토리라고 불리는 데이터 저장소에 소스 코드 등을 넣어서 이용하는 것으로, 이러한 Git 리포지토리를 인터넷상에서 제공하는 서비스가 바로 GitHub입니다.

한마디로 GitHub에서 공개되는 소프트웨어 소스 코드는 모두 Git으로 관리됩니다. Git에 대해서 이해해야 GitHub를 능숙하게 사용할 수 있기 때문에 Git에 대해서는 2장에서 자세히 설명하겠습니다.

1.2 GitHub를 사용하면 무엇이 달라질까?

전 세계의 프로그램 개발 현장은 GitHub의 등장과 함께 많은 변화가 생겼습니다. 가히 혁명이 일어났다 해도 과언이 아닐 정도입니다. 이번 장에서는 아직 GitHub를 본격적으로 사용해 보지 않은 독자를 위해 일반적인 프로그램 개발에서 어떻게 GitHub를 도입하고 있는지 등을 간단히 살펴보겠습니다.

주3 https://github.com/features/hosting

협업 형태 변화

여러 사람이 함께 일할 때 사용되는 소프트웨어는 굉장히 많습니다. 대표적인 소프트웨어로는 그룹웨어와 CRM(Customer Relationship Management, 고객 관계 관리) 등이 있으며, 이러한 소프트웨어는 전 세계의 일반 사무 직종에서 두루 사용되고 있습니다. 물론 여러분이 일하고 있는 회사에서도 이러한 소프트웨어들이 사용되고 있을 것입니다.

하지만 개발자들이 협업할 수 있는 개발자 중심의 소프트웨어는 좀처럼 등장하지 않았습니다. 그래서 소프트웨어 개발자들은 버전 관리 시스템, 버그 트래킹 시스템, 코드 리뷰 시스템, 메일링 리스트, IRC 등 다양한 툴을 만들어 협업에 사용했습니다.

이런 형태로 지금까지 당연하게 여겨지고 있던 소프트웨어 공동 개발의 형태가 GitHub에 의해 상당수가 변경되었습니다. 다음부터 몇 가지 기능을 소개하겠습니다.

● 개발자들이 함께 이야기할 수 있는 Pull Request

세계의 많은 개발자들이 참가하고 있는 GitHub에는 상상도 하지 못할 것들이 무서운 속도로 생겨나고 있습니다. 이런 것을 '소프트웨어 개발자들이 한데 모여 서로 화학 반응을 일으킨다'고 표현합니다. 각자 지구 반대편에서 생활하는 개발자들이 함께 소프트웨어를 작성하고 있습니다. 이런 일이 가능한 가장 큰 이유는 바로 'Pull Request'라고 불리는 기능 때문입니다(그림 1.3).

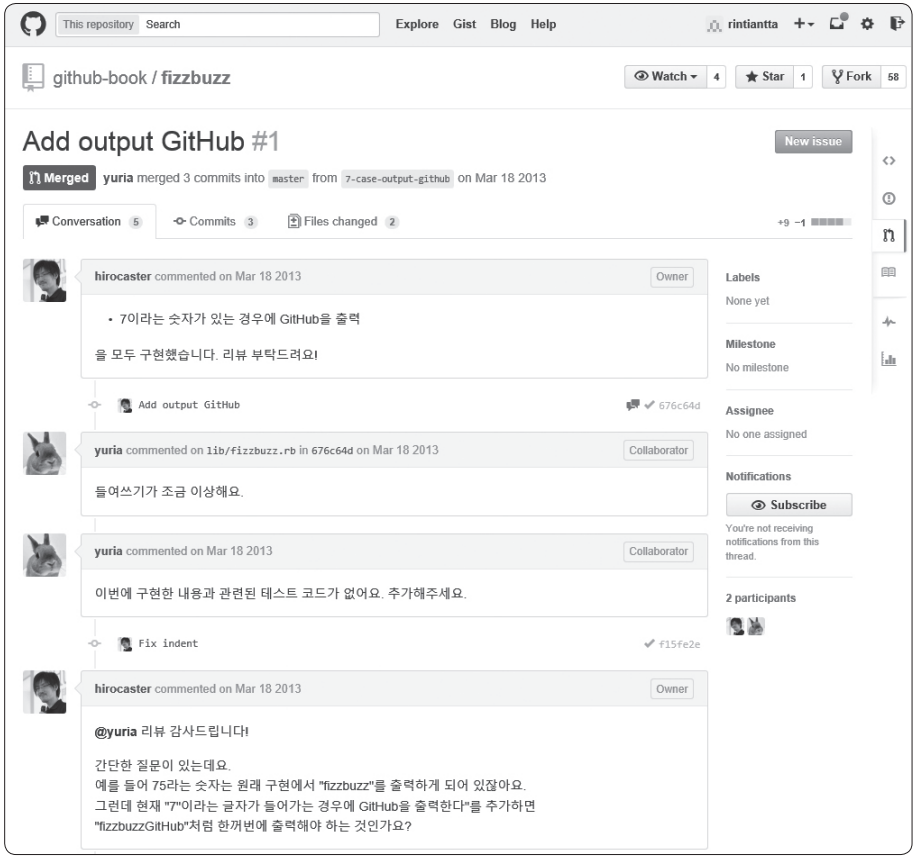


그림 1.3 Pull Request 화면

Pull Request는 GitHub에 있는 Git 리포지토리에서 변경하고 싶은 소스 코드를 주고 수정해 달라고 요청하는 기능입니다. Pull Request를 기반으로 댓글을 주고받을 수도 있습니다. '버그를 고쳤는데, 이렇게 수정해 주시면 안 될까요?' 같은 댓글부터 '새로운 기능을 작성했는데, 이 코드를 넣어 주시면 안 될까요?' 같은 댓글도 있습니다. 간단하게 소스 코드를 변경하고, 변경하고 싶은 기능을 넣도록 요청할 수 있습니다. 물론, 해당 소프트웨어 프로젝트의 정책에 어긋나는 변경 사항은 반영하지 않을 자유도 있습니다.

GitHub의 Pull Request는 소스 코드 변경 이력을 쉽게 확인할 수 있습니다. 또

한, 그림 1.4처럼 특정한 위치에 댓글을 달 수도 있습니다. 이렇게 하면 해당 코드와 관련된 구체적인 토론과 리뷰가 가능해집니다.

그림 1.4 실제 코드 위에 댓글을 다는 모습

● 특정 사용자에게 이야기

지금까지 살펴본 기능 외에도 편리한 기능이 많습니다. 일정 관리 또는 버그 보고는 Issue(이슈) 기능을 사용합니다. 특정한 사용자에게 Issue를 보여 주고 싶을 때는 '@사용자 이름'이라고 적어 줍니다. 이렇게 하면 해당 사용자에게 Notification[주4]이 보내집니다. 또한, Wiki 기능도 제공하므로 쉽게 문서를 작성하고 공개, 공유할 수 있습니다. Wiki 기능은 문서에서 변경된 부분을 차근차근 저장하기에 관리가 무척 쉽습니다.

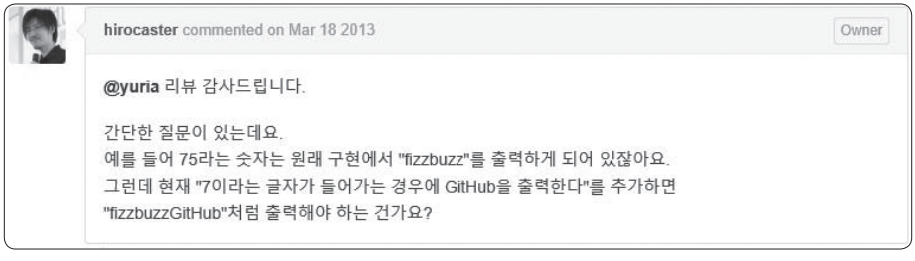


그림 1.5 '@사용자 이름'을 적어서 댓글을 다는 모습

주4 Notification에 대해서는 5장(142쪽)에서 자세히 설명하겠습니다.

● GitHub Flavored Markdown

GitHub에서는 사용자가 작성하는 모든 글이 GitHub Flavored Markdown(이하 GFM)이라 불리는 기법으로 작성됩니다. 따라서 읽기 쉽게 댓글을 작성하거나 문서를 만드는 것이 가능합니다[주5]. 한 개의 기법을 기억하는 것만으로 다양한 곳에 활용할 수 있는 것은 굉장히 효율적인 일입니다. 그중에서 특징적인 기능으로 댓글에 그림을 넣을 수도 있는데, 이렇게 그림과 글자를 함께 사용하면 다른 사람과 더 쉽게 소통할 수 있습니다.

GitHub의 보급으로 인해 Markdown 문법을 사용하는 서비스들이 늘고 있습니다.

Column **이렇게 작성하는 것도 가능하다!!**

'@사용자 이름'의 형태뿐만 아니라 GitHub에서 사용할 수 있는 기법이 더 있습니다.

'@Organization 이름'을 입력하면 Organization에 소속된 사용자 전부에게 Notification을 보낼 수 있습니다. 또한, '@Organization 이름/팀 이름'을 입력하면 팀에 소속된 사용자 전부에게 Notification을 보낼 수 있습니다[주a]. 이렇게 한 번에 여러 사람에게 Notification을 보내는 것도 가능합니다.

'#번호'를 입력하면, 해당 리포지토리의 Issue 번호의 링크가 만들어집니다. '@사용자 이름/리포지토리 이름#번호'를 입력하면 지정한 리포지토리의 Issue 번호 링크가 만들어집니다. 이렇게 특정한 형식으로 작성하기만 하면 자동으로 링크가 생성됩니다. 이러한 기능을 활용하면 다른 사람과 효율적으로 커뮤니케이션할 수 있습니다.

주a Organization과 관련된 상세 내용은 10장의 291쪽에서 설명하겠습니다.

다른 팀이 작성하던 소프트웨어를 더 자세히 볼 수 있다

GitHub는 자신의 팀뿐만 아니라 다른 팀과도 협업할 수 있는 기능을 제공합니다. 다른 팀이 만든 흥미 있는 리포지토리를 Watch에 등록하면, 해당 리포지토리와 관

주5 GFM과 관련된 내용은 3장의 42쪽, 5장의 112쪽에서도 설명하고 있습니다.

련된 정보가 News Feed(뉴스 피드)에 나옵니다(그림 1.6).

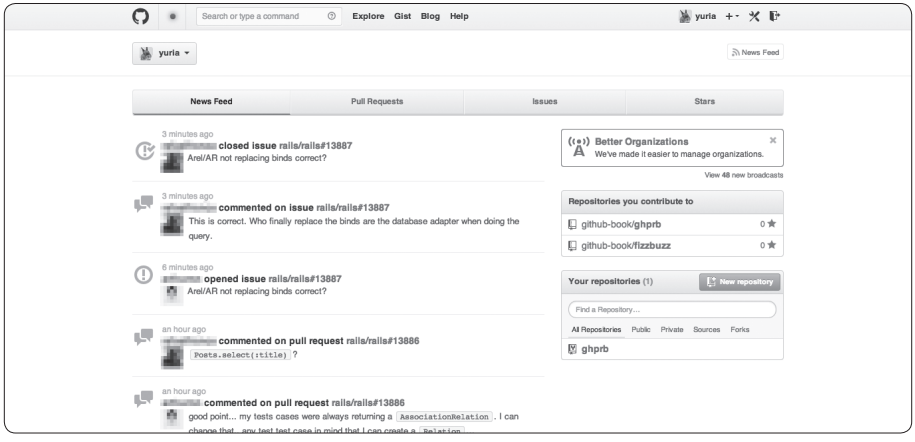


그림 1.6 다양한 리포지토리의 정보가 News Feed에 나오는 모습

예를 들어, 회사에서 사용하고 있는 소프트웨어의 리포지토리를 Watch에 등록하면, 새로운 버전의 새로운 기능과 버그 수정 정보 등을 실시간으로 파악할 수 있습니다. 물론, 독자가 직접 해당 소프트웨어 개발에 참가하는 것도 가능하므로, 적극적으로 의견 제시를 해봅시다. 필요하다면 Pull Request로 코드를 전송하는 것도 가능합니다.

옆 팀이 개발하고 있는 리포지토리를 Watch에 등록하면, 기능이 어느 정도 구현되고 있는지를 매일매일 확인할 수도 있습니다. 유용한 기능 또는 라이브러리를 개발하고 있다면, 이런 기능을 자신의 팀에도 적용할 수 있습니다. 또한, 이미 제공되고 있는 라이브러리 등을 자신의 리포지토리로 옮겨와 새로운 리포지토리를 작성하고 발전시키는 것도 가능합니다. 이러한 것들이 제대로 이루어지면 다른 개발자들과 곧바로 협업할 수 있게 되는 것입니다.

공개 소프트웨어 세계와 같은 개발 스타일

GitHub를 회사에서 사용하면 공개 소프트웨어 개발과 같은 과정으로 개발할 수 있게 됩니다. 따라서 이전에 공개 소프트웨어를 개발하고 있던 개발자는 회사마다 있는 독자적인 툴을 따로 배울 필요가 없어집니다. 때문에 결과적으로는 회사에 들어가자마자 곧바로 개발을 시작할 수 있습니다. 마찬가지로 회사에서 GitHub를 이용하면 대학에서 졸업하고 곧바로 회사에 들어온 1년차 프로그래머도 곧바로 공개 소프트웨어 개발에 참여할 수 있게 됩니다.

결국, 공개 소프트웨어 세계에서의 소프트웨어 개발과 회사에서의 소프트웨어 개발의 차이가 없어지는 셈입니다. 물론, 기업에서 Git 리포지토리가 공개되어 있는지 아닌지의 차이는 있을 수 있습니다.

1.3 소셜 코딩이란?

GitHub는 공개 소프트웨어 개발에 소셜 코딩이라는 개념을 만들어 낸 서비스입니다. 이러한 개념은 세계의 많은 개발자에게 영향을 끼쳤으며, 소프트웨어 개발 방법에 혁명을 일으켰다고 해도 과언이 아닙니다. 여기서는 이러한 소셜 코딩이라는 개념이 무엇인지 자세하게 설명해 보겠습니다.

'소셜 코딩(Social Coding)'이라는 말을 들어 본 적이 있으신가요? 들어 본 적이 없다면, 그림 1.7의 로고는 본 경험이 있으신가요? 다음 그림은 과거의 GitHub의 로고 입니다. 로고를 보면 'SOCIAL CODING'이라는 영문이 작게 들어가 있습니다. 참고로 2013년 4월부터는 그림 1.8의 로고로 바뀌었습니다.

그림 1.7 GitHub의 옛날 로고

그림 1.8 GitHub의 새로운 로고

GitHub[주6]는 소셜 코딩이라는 개념을 만들어 낸 서비스입니다. GitHub가 등장하면서 '소스 코드를 소유하는 권리'의 의미가 새로운 의미로 재탄생되었습니다. 현재는 모든 사람이 쉽게 소스 코드를 소유하고, 변경하고, 공유할 수 있으며, GitHub를 사용하면 모든 개발자가 소스 코드를 공개할 수도 있습니다. 이로 인해 날마다 수많은 개발이 GitHub에서 이루어지고 있습니다.

GitHub가 등장하기 이전에는 Committer라고 불리는 소스 코드를 수정할 수 있는 특권 계층에 의해서 프로그램 개발이 이루어졌습니다. 따라서 일반 개발자가 소스 코드를 수정하거나, 공개를 요청하기 위해서는 이러한 특권 계층을 설득해야만 했습니다. 이러한 이유로 초기에는 빠르게 개발되던 유명 프로그램도 시간이 지나면서 쓰이지 않게 되는 일이 허다해졌습니다.

그러나 GitHub가 등장하면서 개발자 세계에는 '민주화'의 바람이 불었고, 모든 사람이 평등하게 소스 코드를 수정할 수 있는 권리를 얻었습니다. 이것은 프로그램 개발계의 혁명입니다. 그리고 GitHub가 슬로건으로 내걸고 있는 것 또한 이러한 '소

주6 http://github.com/

셜 코딩'이라는 특성입니다.

지금까지 소셜 코딩의 개념에 대해서 설명하고, 소셜 코딩이 이루어질 수 있게 만들어 준 GitHub에 대해서 설명했습니다. GitHub의 개별적인 기능과 관련된 이야기는 3장부터 자세히 설명하겠습니다.

1.4 소셜 코딩을 해야 하는 이유

IT 산업은 잦은 이직이 점점 일반화되고 있는 업계입니다. 유명 개발자들의 블로그를 보면 월초에 '입사했습니다!'라고 쓰여 있는데 반해, 월말에는 '퇴사하게 되었습니다.'라고 쓰인 경우가 허다합니다.

갑자기 당신이 개발자 채용 담당자가 되었다고 가정해 봅시다.

- 지금까지 작성한 코드를 확인할 수 있는 개발자 or 없는 개발자
- 최신 소프트웨어를 잘 알고 있는 개발자 or 잘 모르는 개발자
- 프로그래밍 언어 또는 소프트웨어 세계의 다양한 문화를 이해하고 있는 개발자 or 이해하지 못하는 개발자

이런 개발자들이 있다면 어느 쪽을 채용하고 싶으신가요? 이 경우 후자에 속하지 않기 위해서라도 소셜 코딩 또는 GitHub가 중요합니다.

드넓은 개발 세계

회사라는 갇힌 세계에서만 프로그래밍을 한다면 새로운 기술을 접할 수 없습니다. 일에서 사용하는 기술뿐만 아니라 다양한 기술 지식을 얻으려면 다양한 개발 문화를 접해야 합니다. 또한, 전 세계적으로 많이 사용되는 주목받는 소스 코드 또는 기술, 설계, 문화 등에 눈을 돌리면 자신의 소스 코드와 성과에도 큰 영향을 줍니다. 저

자도 유명한 프레임워크의 구현 방식 등을 참고해 가며 개발중인 소프트웨어를 좋은 방향으로 이끌었던 경험이 있습니다.

코드를 작성할 수 있는 개발자

특히, 빠른 속도로 변화하는 웹 업계에서는 프로그래머가 실제로 코드를 작성할 수 있는지를 중요히 여깁니다. 과거에는 어느 정도의 경력을 가졌고, 인간성이 좋으며, 쉽게 협조를 잘하고 관리 능력이 있는 사람을 중시했습니다. 하지만 최근에는 빠른 기술 발달로 인해 한 개의 서비스를 만드는 데에도 다양한 프로그래밍 언어와 기술이 필요하게 되었습니다. 그로 인해 다양한 프로그래밍 언어를 사용해 코드를 잘 작성하는 개발자가 주목받게 되었습니다.

그런데 코드를 잘 작성한다는 것을 대체 어떻게 판별할까요? 이는 작성한 코드를 직접 보는 것이 가장 확실한 방법일 것입니다. 지금은 GitHub의 등장으로 누구나 쉽게 소스 코드를 공개할 수 있게 되었습니다. 페이스북과 트위터를 보면 해당 사람이 어떤 사람인지 알 수 있듯, GitHub를 봐도 해당 사람이 어떤 성향을 가진 개발자인지 알 수 있습니다.

앞으로도 소셜 코딩을 하는 사람은 점점 늘어날 것입니다. 그리고 회사들도 당신이 작성한 코드를 보고 채용을 결정하는 시대가 코앞으로 다가왔습니다. 자신이 만든 것을 세상에 보여 주는 것이 점점 중요해지는 시대입니다. 따라서 개발을 생업으로 삼고 있는 직업 프로그래머도 반드시 소셜 코딩을 해야 합니다.

GitHub의 가장 큰 특징은 '사람을 바라본다'는 것

이번에는 GitHub가 다른 단순한 리포지토리 호스팅 시스템과 무엇이 다른지, 저자가 중요하게 생각하는 부분을 소개하겠습니다.

GitHub는 지금까지의 리포지토리 호스팅 시스템과 다르게 중심에 사람이 있습니다. 지금까지의 리포지토리 호스팅 서비스는 하나의 프로젝트를 중심으로 구현되어 있었습니다. 따라서 해당 리포지토리의 관리자가 누구인지까지는 알 수 있지만, 그 사람이 다른 무슨 프로젝트에서 무엇을 하는지는 쉽게 알 수 없었습니다.

하지만 GitHub는 프로젝트뿐만 아니라 사람이라는 특성에 주목하였고, 그 사람이 공개하고 있는 프로젝트는 모두 확인할 수 있습니다. 또한, 대시보드[주7]에 표시되는 News Feed로 그 사람이 GitHub에서 하고 있는 모든 개발 활동을 한눈에 볼 수 있습니다.

한마디로 당신이 관심을 두고 있는 개발자가 하는 것을 자세히 확인할 수 있게 되는 것입니다. 관심 있게 지켜보던 개발자가 전 세계적으로 유명해질 수도 있으며, 학교 동급생 또는 회사 동료가 될 수도 있습니다. 사람과 코드 모두를 확인할 수 있는 것이야말로 GitHub가 제공하는 새로운 서비스인 것입니다.

1.5 GitHub가 제공하는 주요한 기능

GitHub는 개발자가 좋은 코드를 효율적으로 만들 수 있게 다양한 기능을 제공합니다. 이번 절에서는 GitHub에서 제공하는 기능들을 간단히 살펴보겠습니다.

Git 리포지토리

GitHub에서 제공되는 Git 리포지토리(repository)는 기본적으로 무료이며, 몇 개를 작성해도 전혀 문제가 없습니다. 하지만 이는 모든 사람에게 공개되는 공개 리포지토리의 경우에 한해서입니다. 자신만 볼 수 있게 만들거나, 한정된 사람만이 열람할 수 있게 만들고 싶다면 비공개 리포지토리를 만들어야 합니다. 비공개 리포지토리를 만들기 위해서는 매달 7달러 이상을 GitHub에 지불해야 합니다. 금액과 관련된 자세한 사항은 https://github.com/plans를 참고하세요.

주7 대시보드와 관련된 내용은 5장 96쪽에서 자세히 설명하겠습니다.

Organization

일반적으로 개인 용도로만 GitHub를 사용한다면 개인 계정만으로도 충분합니다. 하지만 회사 등에서 GitHub를 사용한다면 Organization 계정을 사용하는 것이 좋습니다. Organization 계정을 사용하면 계정의 권한 관리 등을 일괄적으로 할 수 있으며, 지불 방법 등을 통일할 수 있다는 장점이 있습니다.

공개 리포지토리만 사용한다면 무료로 제공되는 Organization 계정을 사용하면 됩니다. 이런 계정은 스터디 또는 IT 계열의 커뮤니티에서 소프트웨어를 개발할 경우 활용하기 좋습니다. 그러나 회사에서 GitHub를 사용하려면 유료 계정을 사용하는 것이 좋습니다. 이에 관련된 내용은 10장에서 자세히 설명하겠습니다.

Issue

한 개의 작업 또는 문제를 해결할 때는 하나의 Issue를 생성합니다. Issue를 생성하면 해당 작업 또는 문제를 해결하는 과정을 관리할 수 있습니다. 일종의 버그 관리 시스템과 같은 방법으로 사용할 수 있는 것입니다. 참고로 이후에 설명하는 Pull Request가 만들어지는 경우에도 하나의 Issue가 생깁니다.

일반적으로 한 개의 기능 추가 또는 수정을 할 때는 하나의 Issue가 만들어지며, 토론 등도 해당 Issue를 중심으로 이루어집니다. 따라서 Issue를 보면 해당 변경 사항과 관련된 내용을 한눈에 볼 수 있습니다.

Git commit 메시지에 '#7'처럼 Issue ID를 추가하면 GitHub에서 자동으로 commit 링크가 붙습니다. 특정한 형식으로 commit 메시지를 작성하는 것만으로도 Issue를 close할 수 있는 매우 편리한 기능입니다. 자세한 내용은 5장의 111쪽에서 설명하겠습니다.

Wiki

Wiki 기능은 문서를 공동으로 작성 또는 변경할 수 있는 기능입니다. 개발 문서 또는 매뉴얼 등을 기록할 때 사용합니다. 문서를 작성할 때는 GFM이라는 형식을 사용하는데, 이는 5장 125쪽에서 자세히 설명하겠습니다.

Wiki 페이지도 Git 리포지토리에서 관리되므로 변경 내역 등이 모두 기록됩니다. 따라서 안심하고 페이지를 작성해도 됩니다. clone해서 편집 또는 확인하는 것도 가능하므로 개발자들이 웹 브라우저로 접속하지 않아도 볼 수 있습니다.

Pull Request

Pull Request는 다른 사람의 리포지토리에 자신이 push한 변경 사항 또는 기능 추가 사항을 넣어 달라고 요구하는 기능입니다. Pull Request를 보내면 다른 사람의 리포지토리 관리자들이 자신들에게 보내진 Pull Request의 내용 또는 포함된 코드의 변경 사항들을 확인합니다.

또한, 그렇게 제출된 Pull Request의 내용이나 소스 코드를 함께 토론하기 위한 기능도 제공되고 있습니다. 예를 들어, 소스 코드의 한 줄마다 댓글을 작성하며 프로그래머들이 문제를 두고 효율적으로 소통하는 것도 가능합니다.

Pull Request와 관련된 내용은 6장에서 더욱 자세히 설명하겠습니다.

Column GitHub에서 주목받고 있는 소프트웨어

GitHub에서 개발되고 있는 소프트웨어를 몇 가지 소개하겠습니다(표 a). 이 책을 읽고 있는 독자분들도 한 번쯤 사용해 보았거나, 들어본 적이 있을 것입니다. 현재 GitHub에서 주목받고 있는 소프트웨어들은 https://github.com/trending에서 추가로 확인할 수 있습니다.

표 a GitHub에서 주목받고 있는 소프트웨어

이름	설명	GitHub 경로
Ruby on Rails	루비(Ruby)에서 사용하는 대표적인 오픈 소스 Web 프레임워크	https://github.com/rails/rails
node	자바스크립트(JavaScript)와 관련된 가장 인기 있는 플랫폼으로 Node.js라고 부릅니다.	https://github.com/joyent/node
jQuery	현재 전세계 모든 곳에서 사용되고 있는 자바스크립트 라이브러리	https://github.com/jquery/jquery
Symfony2	PHP로 작성된 풀 스택 웹 프레임워크	https://github.com/symfony/symfony
Bootstrap	트위터 같은 인터페이스를 작성할 수 있는 컴포넌트 집합	https://github.com/twbs/bootstrap

1.6 정리

이번 장에서는 소셜 코딩이 실제로 이루어지고 있는 GitHub에 대해서 간단히 살펴보았습니다. 자세한 내용은 앞으로 차근차근 알아가도록 하겠습니다.

기호 · 숫자

A

B

C

H

I

J

L

M

N

O

P

ㄱ

ㄷ

ㄹ

ㅁ

ㅂ

ㅅ

ㅇ

ㅈ

ㅊ

ㅋ

ㅌ

ㅍ

ㅎ