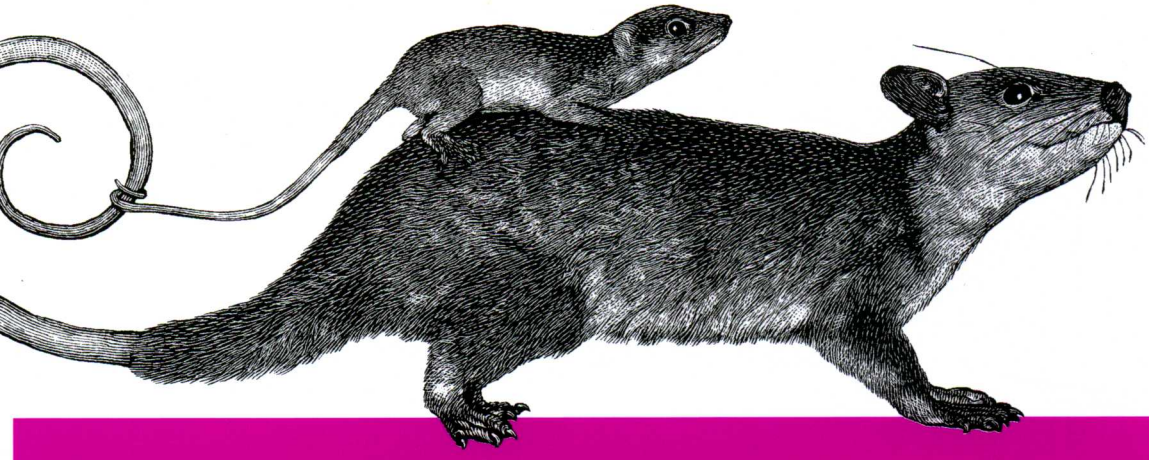
O'REILLY®



GitHub入门

Introducing GitHub

Peter Bell & Brent Beer 著
李新叶 译

中国电力出版社

GitHub入门

Peter Bell, Brent Beer 著

李新叶 译

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo O'REILLY®

O'Reilly Media, Inc.授权中国电力出版社出版

中国电力出版社

图书在版编目（CIP）数据

GitHub入门/（美）贝尔（Bell, P.），（美）比尔（Beer, B.）著；李新叶译. —北京：中国电力出版社，2015.8

书名原文：Introducing GitHub

ISBN 978-7-5123-7943-5

I. ①G… II. ①贝… ②比… ③李… III. ①软件工具－程序设计 IV. ①TP311.56

中国版本图书馆CIP数据核字（2015）第140903号

北京市版权局著作权合同登记

图字：01-2015-3235号

封面设计/ Karen Montgomery，张健

出版发行/ 中国电力出版社（http://www.cepp.sgcc.com.cn）

地　　址/ 北京市东城区北京站西街19号（邮政编码100005）

经　　销/ 全国新华书店

印　　刷/ 北京丰源印刷厂

开　　本/ 787毫米×980毫米　16开本　8.25印张　152千字

版　　次/ 2015年8月第一版　2015年8月第一次印刷

印　　数/ 0001－3000册

定　　价/ 28.00元（册）

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

前言

GitHub正在改变着软件编写的方式。它最初的构想是为开发者提供一种更容易开发源码项目的途径。目前，GitHub正迅速成为默认的软件开发平台。GitHub不仅用于存储源代码，还可以对软件提供详细说明、讨论和评估软件。

本书适用人群

如果你正在与开发人员协作开发软件项目，那么本书正适合于你，无论你属于下述哪种角色：

- 想要了解项目进行程度的企业利益相关者。
- 需要确保软件按时并且能在预算之内交付的产品或项目经理。
- 需要将项目原型转换成HTML/CSS的设计者。
- 将营销文案或其他内容添加到一个网站或应用程序的撰写人。
- 评估一个项目的法律含义或撰写术语和条款或隐私政策的律师。
- 需要对一个项目进行评审、评论或对项目作出其他贡献的团队成员。
- 不熟悉GitHub且想学习如何使用GitHub进行团队协作的开发人员。

如果你需要查看一个正在开发的软件的进展情况，或者希望能够评论其进展情况，或你想发表对项目的修改意见，本书将向你展示如何有效地使用GitHub与软件开发团队协作。

软件之外

尽管GitHub主要用于软件的协作开发，但它还是为团队提供了一种在更广泛项目上合作的极好方式。从书籍(如本书) 的编写和3D打印模型的分配到法规的制作，只要团队人员需要合作收集文档，你就应该考虑使用GitHub进行管理。本书中的示例是以软件协作为前提的，因为这是GitHub目前最常见的使用方式。不过，不管你正在处理哪种项目，本书是介绍通过GitHub进行协作的最佳指南。

本书不适宜人群

本书目的是展示使用GitHub进行有效协作所需的核心技能。如果你已经熟悉分叉、克隆，并且能够使用功能分支和拉请求来进行协作，你可能不再需要阅读本书。

同样，如果你正在寻找深入介绍Git版本控制系统的书，本书也不符合你的要求。本书仅涵盖了足够的Git以引入GitHub，但不是一本对Git作全面介绍的书籍。要想全面了解Git，你应该阅读Jon Loeliger和Matthew McCullough 编写的优秀的《Version Control with Git》（O'Reilly出版）一书。

怎样使用这本书

我们特意将这本书编写的尽可能简洁，你应该能快速地阅读。如果你想真正理解GitHub是什么以及了解如何使用它，试着将这本书从头读到尾吧。

第一章简要介绍了Git、GitHub以及一些需要了解的关键术语，这些术语有助于理解本书其余部分内容。然而，我们知道你很忙。如果你工作很繁忙，请略过第一章。然后随意进入你需要看的章节。我们在编写本书时试图做到每章都提供具体的工作流程，因此你仅需阅读要看的那章内容便可完成特定任务。

本书约定

本书使用以下排版的约定：

斜体字（italic）

表示新术语、网址、电子邮件地址、文件名和文件扩展名。

等宽字体（`constant width`）

用于表示程序清单，以及在段落中引用的程序元素如变量、函数名、数据库、数据类型、环境变量、语句和关键字。

等宽黑体(**constant width bold**)

表示由用户输入的命令或其他文本。

等宽斜体(*constant width italic*)

表示可由用户提供值或语境决定的值替换的文本。

Safari在线图书

Safari在线图书是一个按需服务的数字图书馆，它以书籍和视频的形式提供了来自全球范围内技术和企业领域资深作者撰写的专业内容。

专业技术人员、软件开发人员、网页设计师、商业和创意专业人士使用Safari在线图书作为研究解决问题、学习和认证培训的首要资源。

Safari在线图书为企业、政府、教育部门和个人提供了一系列计划和定价方案。

用户可在一个完全可搜索的数据库中获得成千上万的书籍、培训视频以及出版前的手稿，这些数据库来自出版商如O'Reilly Media，Prentice Hall Professional，Addison-Wesley Professional，Microsoft Press，Sams，Que，Peachpit Press，Focal Press，Cisco Press，John Wiley & Sons，Syngress，Morgan Kaufmann，IBM Redbooks，Packt，Adobe Press，FT Press，Apress，Manning，New Riders，McGraw-Hill，Jones & Bartlett，Course Technology以及其他上百个甚至更多的出版商。有关Safari在线图书的更多信息,请访问我们的网站。

联系我们

请将您关于这本书的意见和问题反映给以下出版商：

美国：

O'Reilly Media，Inc.
1005 Gravenstein Highway North
Sebastopol，CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）
奥莱利技术咨询（北京）有限公司

我们有一个关于这本书的网页，在网页上列出了勘误表、示例和其他信息。可以在*http://bit.ly/intro-github*访问此页。

发表评论或咨询有关这本书的技术问题，请发送电子邮件至*bookquestions@oreilly.com*。

关于我们的书籍、课程、会议和新闻的更多信息，请参阅我们的网站*http://www.oreilly.com*。

在Facebook上联系：*http://facebook.com/oreilly*

在Twitter上关注我们：*http://twitter.com/oreillymedia*

在YouTube上观看：*http://www.youtube.com/oreillymedia*

致谢

Peter：比起不希望有致谢部分，我更愿意在此感谢我的妻子，她不知疲倦地花费时间和精力支持我写这本书以及许多其他项目。这些工作曾使我无暇顾及她太多。我还想感谢我的妈妈，她总是超出自己所能支持我，让我能始终遵循我的梦想，甚至在经常困难的情况下。

Brent：我要感谢我的妈妈不断鼓励我阅读，否则，我可能永远不会找到对这项工作的喜爱。同时还要感谢我的父亲，假如没有他让我在电脑前观看他工作，陪我在安装于苹果电脑上的Oscar the Grouch垃圾桶软件中娱乐，鼓励我去学习编程，我将不可能取得现在的成就。

我们同时还要感谢不断鼓励我们的Matthew和Jordan McCullough以及其余的GitHub团队成员，感谢他们对本书所反馈的建议、想法和支持。这本书的很多精华之处都源于他们。我们还要感谢O'Reilly出版社的Meg Blanchette，没有他，这本书将不会构思、编写或传阅。再次对Meg致以真挚的感谢!

第1章

概述

在这一章里，我们将首先介绍什么是Git和GitHub，它们之间的区别是什么，以及为什么你想使用它们。然后，将介绍一些其他常用术语，当人们讨论GitHub时，你会经常听他们提到这些术语。这样，你就能够更容易理解和参与讨论你的项目。

Git是什么?

Git是一个版本控制系统。版本控制系统是设计用于跟踪文件随时间变化状态的一款软件。更具体地说，Git是一个分布式的版本控制系统。这就意味着，在Git中参与项目的每个程序员不仅能拥有文件的当前状态，还能拥有项目完整的历史记录。

Github是什么?

GitHub是一个网站，你可以向该网站上传一个Git数据库副本。使用GitHub使你与他人合作一个项目变得更容易，而这归功于GitHub提供的下述机制：一个用以共享库的集中位置，一个基于Web的界面以及分叉（forking）、拉请求（pull request）、提出问题（issue）、维基（WiKi）等功能，这些功能使你和你的团队能更有效地对所做的修改进行说明、讨论和评估。

为什么使用Git?

即使是你独自一人工作时，如果你正在编辑文本文件，使用Git也会有许多好处，如下所示：

撤销更改功能

如果编译过程中出错，你可以及时回到前一个时间点，以恢复较早的一个版本。

一个包含所有更改的完整历史记录

如果你想要了解一天前、一周前、一个月前甚至一年前的项目状态，你可以检查项目的先前版本以详细查看当时文件的状态。

对改动原因进行文档记录

通常我们很难记住当时为什么要做出修改。使用Git的提交信息（commit message）功能，就很容易对你所做改动的原因进行文档记录，以备将来参考。

修改一切的信心

因为使用Git很容易恢复前一个版本的项目，你可以有信心做出任何你想要的修改。如果项目修改后不能正常工作，你还可以随时回到较早前的一个版本。

历史记录的多种分支流

你可以通过创建历史记录的不同分支，来尝试对内容做出不同的更改，或独立构建不同的功能分支。然后你可以将这些不同分支合并到主项目历史记录(即主分支，master branch)，或者在它们最终不工作时将其删除。

当你在一个团队工作时，使用Git跟踪你的更改会得到更大范围的好处。与一个团队合作时使用Git的一些关键好处如下：

解决冲突的能力

使用Git，许多人可以在同一时间对相同的文件进行更改。通常，Git能够自动合并这些更改。如果不能自动合并，它将向你展示有何冲突并且将使你更易解决这些冲突。

历史记录的独立分支流

在一个项目中不同的人可以在不同的分支工作，这使他们能独立工作在不同的功能上，然后在各自完成后合并这些功能。

为什么使用GitHub?

GitHub不仅仅是一个用来存储Git库的地方，它提供了许多额外好处，包括执行以下操作的能力：

文档需求

使用提出问题（issues），你可以记录缺陷或指定新的功能，这些新功能需要你的团队进行开发。

独立分支流记录的协作

使用分支（branches）和拉请求（pull requests），你可以在不同的分支或功能下进行合作。

评估工作进展

通过查看拉请求（pull requests）列表，你可以看到目前处于工作状态下的所有不同功能，通过单击列出的任何一个拉请求命令，你可以看到最新的变化以及所有关于改动的讨论。

看到团队的进展

通过使用pulse或提交历史（commit history）命令，你可以看到团队的工作进展。

关键概念

为有效使用Git和GitHub工作，你需要理解一些关键概念。下面列出一些最常用的术语，每个术语都有一个简短描述以及在对话中怎样使用的例子：

提交（commit）

无论何时你将一个或多个文件修改保存到Git的历史记录，你都会创建一个新的提交。示例用法:“让我们提交这些更改并发送到GitHub。”

提交消息（commit message）

每次做出提交时，你需要提供一个消息，描述为什么要进行这种改动。当以后试图理解为什么实现特定的修改时，提交的这一消息是非常有用的。示例用法:“在提交消息时一定要包括苏珊关于SEC新准则的评论。”

分支（branch）

就是存放在一侧的独立的系列提交，你可以使用它来进行一个实验或者创建一个新的功能。示例用法:“让我们创建一个分支来实现新的搜索功能。”

主分支(master branch)

无论什么时候创建一个新的Git项目，都会创建一个默认的分支，称为主分支。这个分支一旦准备发布，你的工作则应完全停止。示例用法:“记住不要直接提交给主分支。”

功能分支(feature branch)

不论何时构建一个新的功能，都将创建一个分支，称为功能分支。示例用法:“我们已有太多的功能分支，让我们集中解决其中一两个并完成部署。”

发布分支（release branch）

如果你有一个手动QA（质量管理）流程，或者为满足客户需求而必须支持旧版本的软件时，你需要一个发布分支以存放必要的补丁或更新记录。功能分支跟发布分支没有任何技术差别，但是在和团队谈论项目时，区别两者是有用的。示例用法:“我们必须解决所有支持的发布分支下的安全漏洞。”

合并（merge）

合并是将一个分支完成的全部工作归并到另一个分支。通常情况下是将一个功能分支合并到主分支。示例用法:“ 有关‘我的帐户’功能的伟大工作。你能将其合并到主分支吗?这样我们可以将其发布。”

标签（tag）

引用一个特定历史的提交。最常用于记录发布版本，据此你可以知道发布的是哪个版本的代码以及何时生成的。示例用法:“让我们标记这个版本，并使其发布吧。”

查看（check out）

找到一个不同版本的项目历史记录，以及时查看该时间点的文件。通常你会找一个分支以查看在其上完成的所有工作。但其实你可以查看任何类型的提交。示例用法:“你可以查看最后发布的版本标签吗?发布时有一个错误，需要你复制和修复。”

拉请求（pull request）

最初，拉请求是用来请求别人复查已经完成的分支工作，并将它合并到主分支。现在，拉请求常用在一个流程的早期阶段，用以讨论可能的功能。示例用法:“为新的投票功能去创建一个拉请求，这样我们可以看到团队其他人是如何考虑的。”

提出问题（issue）

GitHub有一个称为提出问题的功能，可以用来讨论功能、跟踪缺陷，或两者兼备。示例用法:“你是对的，在手机上这个登录行不通。你可以在GitHub上创建一个issue，记录复制这个错误的步骤吗?”

维基（WiKi）

维基最初由Ward Cunningham开发，是一个轻量级的Web页面创建方式，创建的Web页面之间用简单的链接相联系。GitHub项目经常使用WiKi进行文档记录。示例用法:“你能在WiKi中添加一个页面，解释如何配置运行在多个服务器上的项目吗?”

克隆（clone）

通常你要从GitHub下载一个项目的副本，这样你就可以在本地工作。将项目库复制到你的电脑的过程称为克隆。示例用法:“你能克隆存储库，修复错误，然后今晚把修复发送回GitHub吗?”

分叉(fork)

有时候你不具备直接改变一个项目的必要的许可。也许这是一个你不知道的人写的开源代码，或者是你公司另一团队写的项目，而你不曾过多参与其中。如果你想对这样的一个项目提交修改，首先需要在GitHub上你的用户帐户下复制这个项目。这一过程被称为分叉存储库。然后你可以克隆、修改，并使用拉请求将其提交回最初的项目。示例用法:“我想看看你将如何重写主页的营销文案。请分叉项目并提交一个拉请求，提出你的修改。”

起初所有术语会令你感到无所适从，请不要担心。一旦你开始一些实际项目时，它们将展现出更多的意义！在下一章将看到GitHub项目的各种元素，以及如何使用它们来开展一个项目的流程。

第2章

查看

在本章我们将看到如何查看项目的状态，以了解其进展。我们将以流行的Bootstrap开源项目（*http://getbootstrap.com/*）为例加以介绍。

项目页面介绍

Bootstrap是一个项目，该项目使开发人员能快速开发引人瞩目的Web应用程序。单击GitHub中的项目页面（*https://github.com/twbs/bootstrap*），在主页上会有大量信息。首先来看一下页面上的一些重要元素（见图2-1）。

在页面左上角首先看到项目名称为“Bootstrap”，并且项目拥有者是一个名叫“twbs”的用户（这里是一个组织）。如果打开*https://github.com/twbs*，你将看到该组织在GitHub上拥有的所有项目清单。在该组织名称的左侧，你还会看到一个图标，该图标清楚地表明，这是一个公共库，任何人都可以查看。不过你所从事的许多项目将会有一个闭合的锁图标，表明这些项目是私有的，仅能被已明确添加为合作者的人查看。

项目名称的右边，可以看到在截图时的那一刻（见图2-1），有3857人一直在关注这个库，以期望每次有新的变化时能得到通知；有68 928人将其标上星号，表示这是他们最喜欢的项目之一；25 292人曾经分叉这一库文件，在GitHub上制作自己的副本，并通过GitHub上传对项目的修改以及与他人分享这些修改。

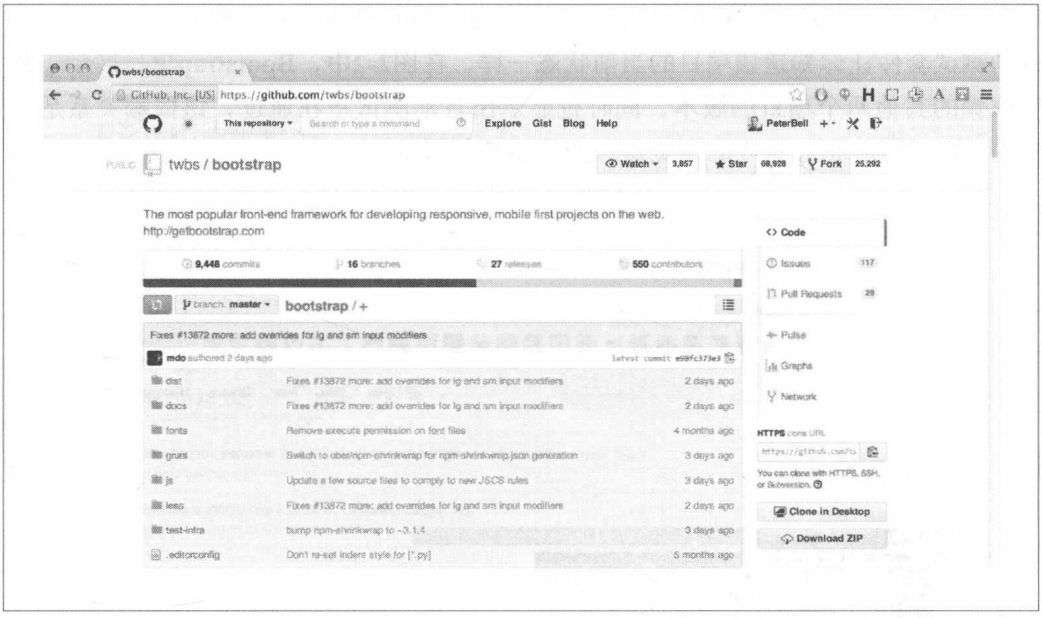


图2-1：GitHub上的Bootstrap项目主页

继续往下看，你可以看到项目的一个简短描述，在其下方你将看到该项目已经有总共9 448次变化（提交），目前正在开发（分支）的不同的历史数据流有16个，随着时间的推移，已有27个版本的软件推荐给人们使用（发布），并且有550人编写了代码的某些部分（贡献）。

你还可以看到目前我们正在查看的是主分支，位于根目录bootstrap文件夹下，最近提交到主分支的是“Fixes #13872 more: add overrides for lg and sm input modifiers”（不必关心其代表什么），并且这一提交的GitHub用户是“mdo”（*https://github.com/mdo*）。当你继续往图下方看时，你可以看到项目根目录（顶层）文件夹下包含的文件夹（有时称为目录）以及文件。

查看README.md文件

如果在项目根目录中有一个叫README.md的文件，该文件的内容将显示在项目主页上文件夹和文件列表的正下方。该文件提供了项目说明和其他额外信息，这些信息对合作者非常有用，例如，如何安装软件，如何运行任何自动化测试，如何使用代码，以及如何对项目做出贡献。