Github操作手册

The Hello World project is a time-honored tradition in computer programming. It is a simple exercise that gets you started when learning something new. Let s get started with GitHub! You ll learn how to:

﹒Create and use a repository

﹒Start and manage a new branch

﹒Make changes to a file and push them to GitHub as commits

﹒Open and merge a pull request

Hello World项目是计算机编程的一个历史悠久的传统。这是一个简单的练习，让你在学习新东西的时候开始。让我们从GitHub开始吧！您将学习如何：

﹒创建和使用存储库

﹒启动和管理一个新的分支

﹒对文件进行更改并将其推送到GitHub中

﹒作为提交打开并合并拉请求

**What is GitHub?**

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This tutorial teaches you GitHub essentials like repositories, branches, commits, and Pull Requests. You ll create your own Hello World repository and learn GitHub s Pull Request workflow, a popular way to create and review code.

GitHub是一个用于版本控制和协作的代码托管平台。它让你和其他人一起在任何地方的项目上合作。本教程将向您介绍GitHub的基本内容，如存储库、分支、提交和Pull请求。您将创建自己的Hello World存储库，并学习GitHub的Pull请求工作流，这是一种创建和评审代码的流行方式。

#### No coding necessary

To complete this tutorial, you need a GitHub.com account and Internet access. You don t need to know how to code, use the command line, or install Git (the version control software GitHub is built on).

Tip: Open this guide in a separate browser window (or tab) so you can see it while you complete the steps in the tutorial.

要完成本教程，您需要一个GitHub.com帐户和Internet access。您不需要知道如何编码、使用命令行或安装Git（版本控制软件GitHub是建立在上面的）。

提示：在一个单独的浏览器窗口（或选项卡）中打开这个指南，这样您就可以在完成本教程中的步骤时看到它。

## Step 1. Create a Repository

A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets anything your project needs. We recommend including a README, or a file with information about your project. GitHub makes it easy to add one at the same time you create your new repository. It also offers other common options such as a license file. Your hello-world repository can be a place where you store ideas, resources, or even share and discuss things with others.

存储库通常用于组织单个项目。存储库可以包含文件夹和文件、图像、视频、电子表格和数据集，任何您的项目需要的东西。我们建议包括一个自述文件，或者一个包含您的项目信息的文件。GitHub让您可以轻松地在创建新存储库的同时添加一个。它还提供了其他常见选项，比如许可证文件。您的hello-world存储库可以是您存储想法、资源、甚至与他人共享和讨论事物的地方。

### To create a new repository

In the upper right corner, next to your avatar or identicon, click and then select New repository. Name your repository hello-world. Write a short description. Select Initialize this repository with a README.

在右上角，在您的化身或id诱惑的旁边，点击并选择新的存储库。命名您的存储库hello-world。写一个简短的描述。选择用自述文件初始化这个存储库。



Click Create repository.

点击创建存储库。

## Step 2. Create a Branch

Branching is the way to work on different versions of a repository at one time. By default your repository has one branch named master which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to master. When you create a branch off the master branch, you re making a copy, or snapshot, of master as it was at that point in time. If someone else made changes to the master branch while you were working on your branch, you could pull in those updates.

分支是在同一时间处理存储库不同版本的方法。默认情况下，您的存储库有一个名为master的分支，它被认为是确定的分支。我们使用分支来进行实验和编辑，然后再提交给主人。当你在主分支上创建一个分支时，你正在复制或快照，就像在那个时间点上的那样。如果你在你的分支上工作时，其他人对主分支做了修改，你可以把这些更新拉进来。

This diagram shows:

﹒The master branch

﹒A new branch called feature (because we re doing feature work on this branch)

﹒The journey that feature takes before it s merged into master

这张图显示了：

﹒主分支

﹒一个叫做特性的新分支（因为我们在这个分支上做特性工作）

﹒这一特性在它被合并为大师之前



Have you ever saved different versions of a file? Something like:

你曾经保存过不同版本的文件吗？如：

* story.txt
* story-joe-edit.txt
* story-joe-edit-reviewed.txt

Branches accomplish similar goals in GitHub repositories. Here at GitHub, our developers, writers, and designers use branches for keeping bug fixes and feature work separate from our master (production) branch. When a change is ready, they merge their branch into master.

分支机构在GitHub库中实现了类似的目标。在GitHub上，我们的开发人员、作者和设计人员使用分支来保持bug修复和特性工作与我们的主（生产）分支分离。当变更准备好时，他们将他们的分支合并为master。

### To create a new branch

１.Go to your new repository hello-world.

去你的新存储库hello-world。

２. Click the drop down at the top of the file list that says branch: master.

点击文件列表顶部的下拉菜单，上面写着：master。

３. Type a branch name, readme-edits, into the new branch text box.

在新的分支文本框中输入一个分支名称，readme编辑器。

４.Select the blue Create branch box or hit Enter on your keyboard.

选择蓝色的创建分支框，或者在键盘上点击回车。



Now you have two branches, master and readme-edits. They look exactly the same, but not for long! Next we ll add our changes to the new branch.

现在你有了两个分支，主和读编辑。它们看起来完全一样，但不会持续太久！接下来，我们将把变更添加到新的分支。

## Step 3. Make and commit changes

Bravo! Now, you re on the code view for your readme-edits branch, which is a copy of master. Let s make some edits. On GitHub, saved changes are called commits. Each commit has an associated commit message, which is a description explaining why a particular change was made. Commit messages capture the history of your changes, so other contributors can understand what you ve done and why.

万岁!现在，您在readme编辑分支的代码视图中，这是一个主副本。让我们做些编辑。在GitHub上，保存的更改称为提交。每个提交都有一个关联的提交消息，这是一个描述解释为什么会发生特定的更改。提交消息捕获变更的历史，因此其他贡献者可以理解您所做的和为什么。

#### Make and commit changes

１. Click the README.md file.

单击README.md文件。

2. Click the pencil icon in the upper right corner of the file view to edit.

单击文件视图右上角的铅笔图标来编辑。

3. In the editor, write a bit about yourself.

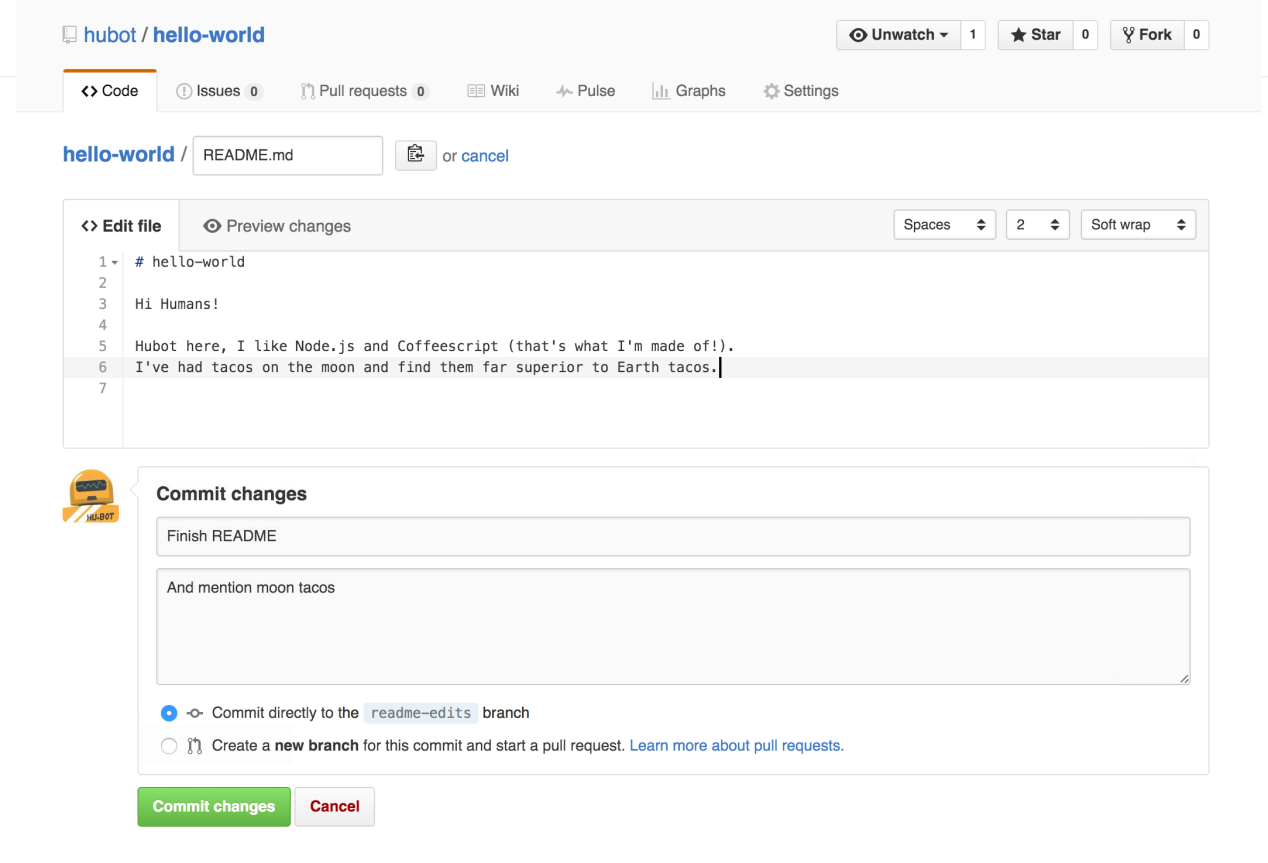
在编辑中，写一些关于你自己的文章。

4. Write a commit message that describes your changes.

写一个描述您的变更的提交消息。

5. Click Commit changes button

点击Commit更改按钮



These changes will be made to just the README file on your readme-edits branch, so now this branch contains content that s different from master.

这些更改只会在您的README编辑分支上的README文件中进行，所以现在这个分支包含了与主人不同的内容。

**Step 4. Open a Pull Request**

Nice edits! Now that you have changes in a branch off of master, you can open a pull request.

漂亮的编辑!既然您已经在主的分支上进行了更改，您可以打开一个pull请求。

Pull Requests are the heart of collaboration on GitHub. When you open a pull request, you re proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show diffs, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.

拉请求是GitHub上协作的核心。当你打开一个拉拽请求时，你会提出你的改变，并请求某人检查并拉出你的贡献，并将它们合并到他们的分支中。拉请求显示了来自两个分支的内容的扩散或差异。更改、添加和减色显示为绿色和红色。

As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

一旦您提交了一个提交，您就可以打开一个pull请求并开始讨论，甚至在代码完成之前。

By using GitHub s @mention system in your pull request message, you can ask for feedback from specific people or teams, whether they re down the hall or 10 time zones away.

通过在你的拉请求信息中使用GitHub的@提及系统，你可以向特定的人或团队寻求反馈，无论他们是在大厅还是10个时区之外。

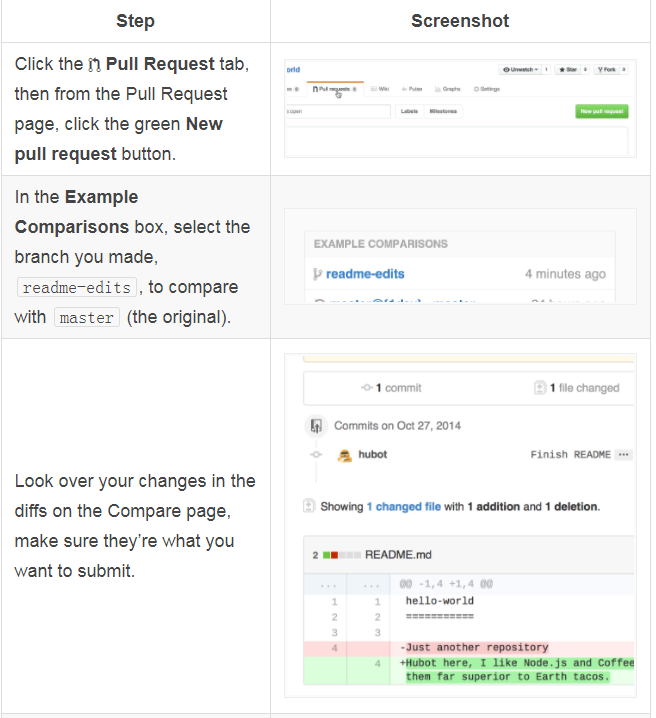
You can even open pull requests in your own repository and merge them yourself. It s a great way to learn the GitHub flow before working on larger projects.

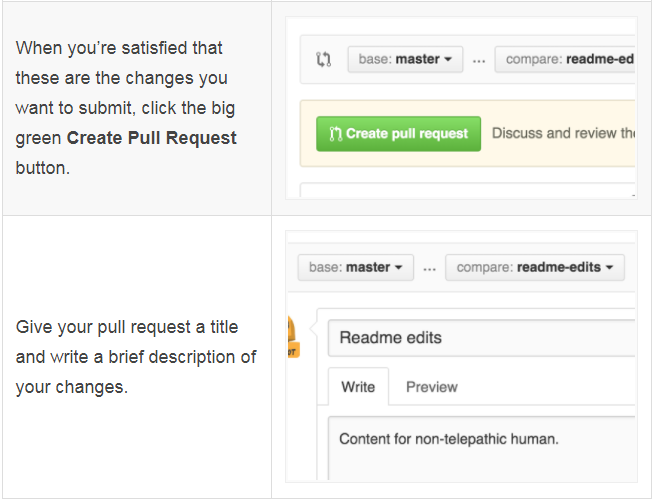
您甚至可以在自己的存储库中打开pull请求，并自己合并它们。在开发更大的项目之前，这是一个学习GitHub流程的好方法。

#### Open a Pull Request for changes to the README

Click on the image for a larger version

点击图片以获得更大的版本





When you re done with your message, click Create pull request

当您处理完您的消息时，单击Create pull request

Tip: You can use emoji and drag and drop images and gifs onto comments and Pull Requests.

提示：你可以使用表情符号，拖放图片和gif到评论和拉请求。

## Step 5. Merge your Pull Request

In this final step, it s time to bring your changes together merging your readme-edits branch into the master branch.

在这最后一步，是时候将您的变更合并到您的readme编辑分支中，并将其合并到主分支中。

1. Click the green Merge pull request button to merge the changes into master.

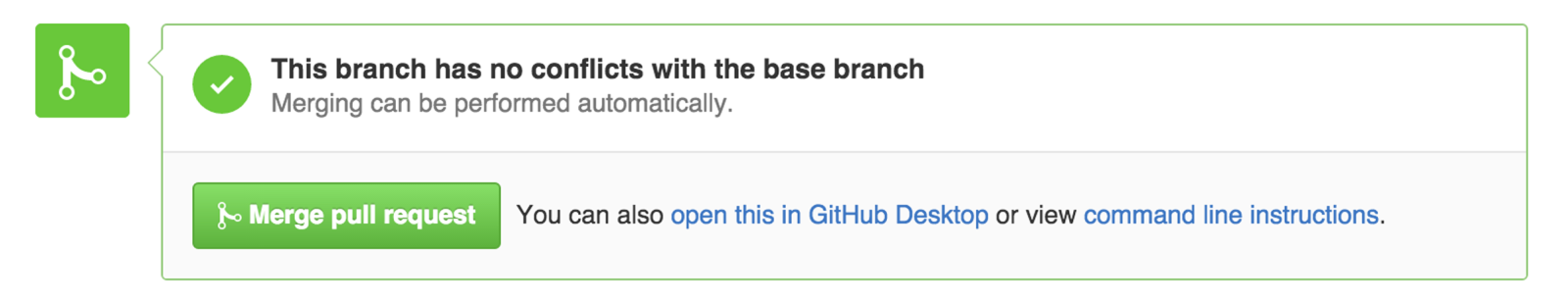
点击绿色合并拉请求按钮将更改合并为master。

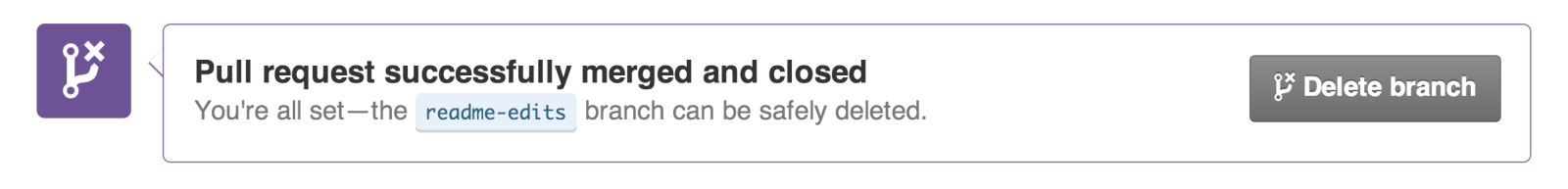
2. Click Confirm merge.

点击确认合并。

3. Go ahead and delete the branch, since its changes have been incorporated, with the Delete branch button in the purple box.

继续删除分支，因为它的更改已经被合并，在紫色的框中有删除分支按钮。





### Celebrate!

By completing this tutorial, you ve learned to create a project and make a pull request on GitHub

通过完成本教程，您已经学会了创建一个项目并在GitHub上发出拉请求

Here s what you accomplished in this tutorial

以下是您在本教程中所完成的工作:

* Created an open source repository
* **创建一个开源的仓库**
* Started and managed a new branch
* 开始并管理一个新的分支
* Changed a file and committed those changes to GitHub
* 修改了一个文件并将这些更改提交给GitHub
* Opened and merged a Pull Request

打开并合并一个拉请求

Take a look at your GitHub profile and you ll see your new contribution squares! To learn more about the power of Pull Requests, we recommend reading the GitHub flow Guide. You might also visit GitHub Explore and get involved in an Open Source project

看一看你的GitHub页面，你会看到你的新贡献方块！要了解更多关于拉请求的能力，我们建议阅读GitHub流指南。您还可以访问GitHub探索并参与到一个开放源码项目中

Tip: Check out our other Guides, YouTube Channel and On-Demand Training for more on how to get started with GitHub.

提示：查看我们的其他指南、YouTube频道和按需培训，了解如何开始使用GitHub。