



创建存储库

[创建一个分支](#)[做一个提交](#)[打开拉取请求](#)[合并拉取请求](#)

你好，世界

🕒 10分钟阅读

该**的Hello World**项目是计算机编程历史悠久的传统。这是一个简单的练习，可以让你在学习新东西时开始学习。让我们开始使用GitHub！

您将学习如何：

- 创建和使用存储库
- 启动并管理新分支
- 对文件进行更改并将其作为提交推送到GitHub
- 打开并合并拉取请求

什么是GitHub？

GitHub是用于版本控制和协作的代码托管平台。它可以让您和其他人在任何地方协同工作。

本教程教您GitHub基本知识，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 Request工作流，这是一种创建和检查代码的流行方法。

无需编码

要完成本教程，您需要[GitHub.com帐户](#)和Internet访问权限。您不需要知道如何编码，使用命令行或安装Git（基于版本控制软件GitHub）。

提示：在单独的浏览器窗口（或选项卡）中打开本指南，以便在完成本教程中的步骤时看到它。

[介绍](#)[什么是GitHub？](#)[创建存储库](#)[创建一个分支](#)[做一个提交](#)[打开拉取请求](#)[合并拉取请求](#)

步骤1.创建存储库


一个**库**通常用于举办单个项目。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 - 您的项目需要的任何内容。我们建议包括**README**或包含项目信息的文件。GitHub可以在创建新存储库的同时轻松添加一个。它还提供其他常见选项，例如许可证文件。

您的 `hello-world` 存储库可以是存储想法，资源甚至与他人共享和讨论事物的地方。


创建新存储库

1. 在右上角，在您的头像或identicon旁边，单击 **+**然后选择**New repository**。
2. 命名您的存储库 `hello-world`。
3. 写一个简短的描述。
4. 选择**使用自述文件初始化此存储库**。


PUBLIC



Owner

 hubot


Repository name

hello-world 


Great repository names are short and memorable. Need inspiration? How about **petulant-shame**.

Description (optional)

Just another repository

☒  **Public**

Anyone can see this repository. You choose who can commit.


☐  **Private**

You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None**

Add a license: **None** 

Create repository

单击**创建存储库**。🎉

第2步。创建一个分支

[介绍](#)[什么是GitHub？](#)

分支是一次处理不同版本的存储库的方法。

[创建存储库](#)

默认情况下，您的存储库有一个名为branch的分支 `master`，该分支被认为是权威分支。我们使用分支进行实验并在提交之前进行编辑 `master`。

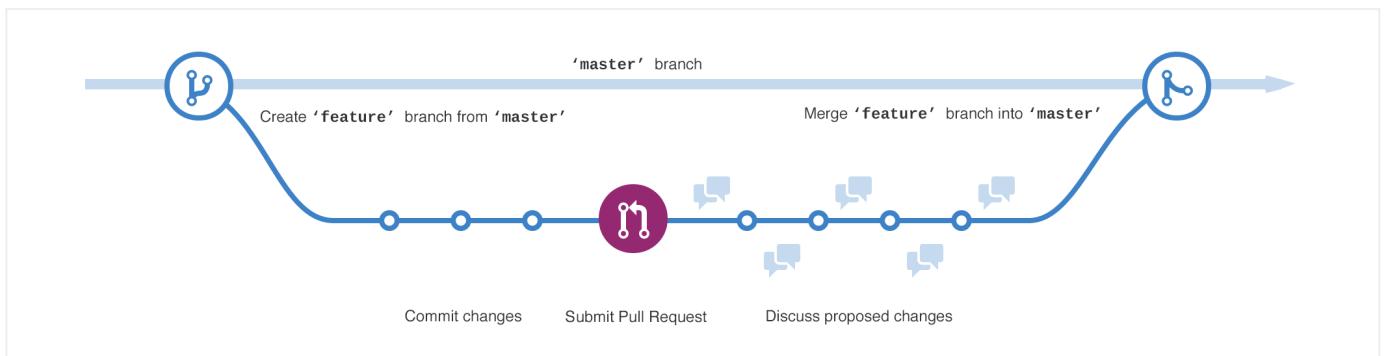
[创建一个分支](#)[做一个提交](#)

当您在分支机构上创建分支时 `master`，您正在制作该 `master` 时间点的副本或快照。如果其他人 `master` 在您的分支机构上工作时对分支进行了更改，则可以提取这些更新。

[打开拉取请求](#)[合并拉取请求](#)

该图显示：

- 该 `master` 分支
- 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`



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

- `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`.

To create a new branch

介绍

什么是GitHub？

创建存储库

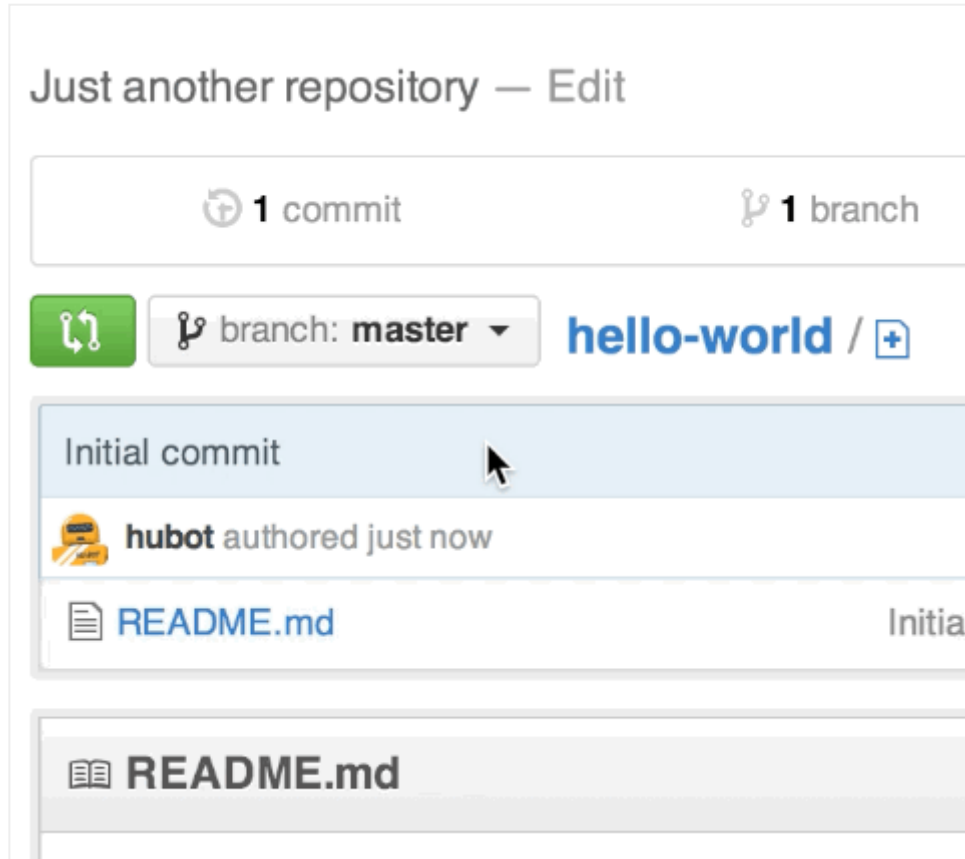
创建一个分支

做一个提交

打开拉取请求

合并拉取请求

1. Go to your new repository `hello-world`.
2. Click the drop down at the top of the file list that says **branch: master**.
3. Type a branch name, `readme-edits`, into the new branch text box.
4. Select the blue **Create branch** box or hit “Enter” on your keyboard.




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.

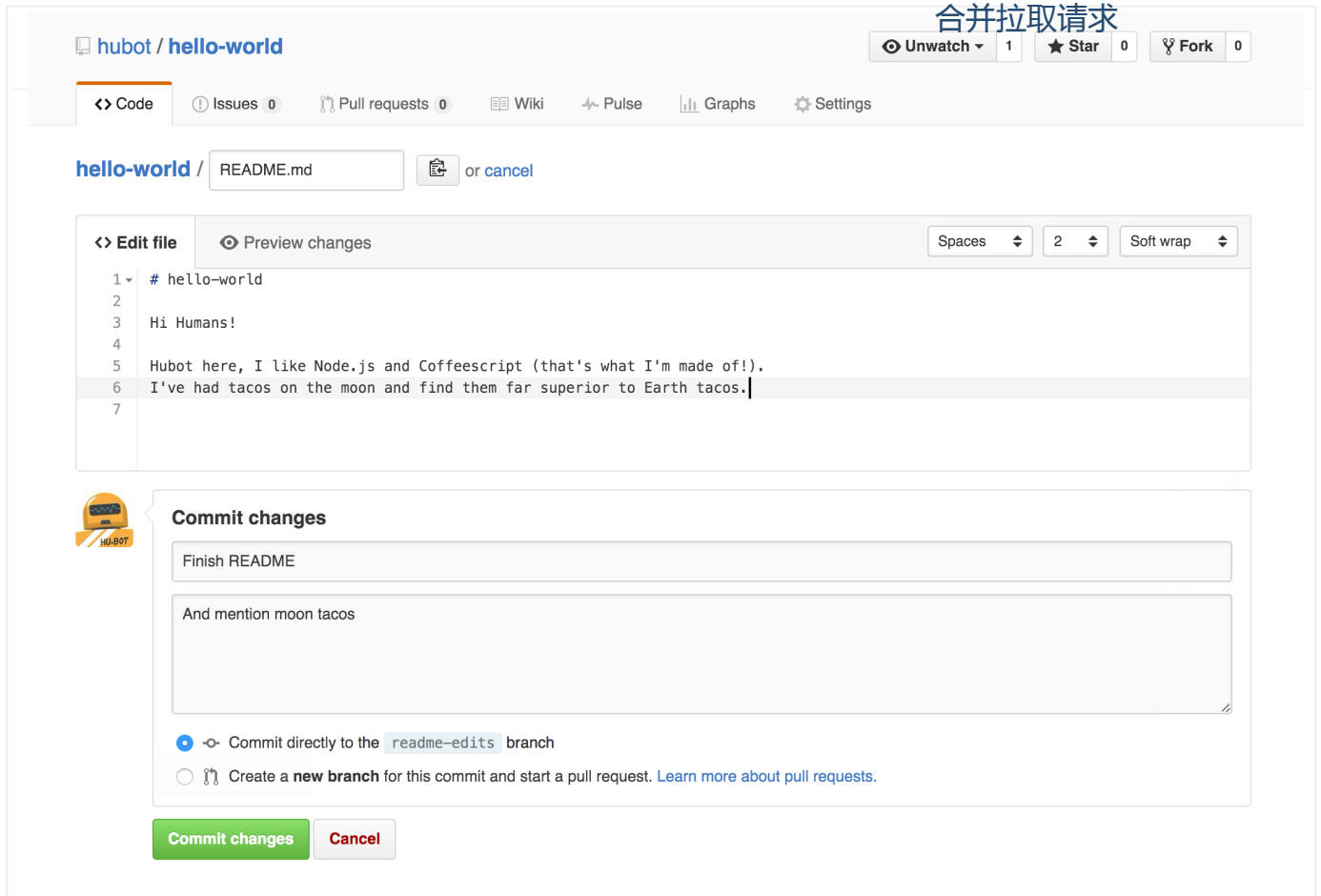
Step 3. Make and commit changes

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.

Make and commit changes

1. Click the `README.md` file.
2. Click the  pencil icon in the upper right corner of the file **创建存储库**.
3. In the editor, write a bit about yourself. **创建一个分支**
4. Write a commit message that describes your changes. **做一个提交**
5. Click **Commit changes** button. **打开拉取请求**

[介绍](#)[什么是GitHub？](#)[创建存储库](#)[创建一个分支](#)[做一个提交](#)[打开拉取请求](#)[合并拉取请求](#)

The screenshot shows the GitHub web interface for the repository 'hubot / hello-world'. At the top, there are buttons for 'Unwatch', 'Star', and 'Fork'. Below this is a navigation bar with 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area shows the 'hello-world / README.md' file being edited. The editor has a line number margin on the left and a toolbar on the right with 'Spaces', '2', and 'Soft wrap' options. The file content is as follows:

```
1 # hello-world
2
3 Hi Humans!
4
5 Hubot here, I like Node.js and Coffeescript (that's what I'm made of!).
6 I've had tacos on the moon and find them far superior to Earth tacos.
7
```

Below the editor is the 'Commit changes' dialog. It has a text input field with 'Finish README' and a larger text area with 'And mention moon tacos'. There are two radio buttons: the first is selected and labeled 'Commit directly to the `readme-edits` branch', and the second is labeled 'Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)'. At the bottom of the dialog are two buttons: 'Commit changes' (green) and 'Cancel' (red).

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`.

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*.

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.


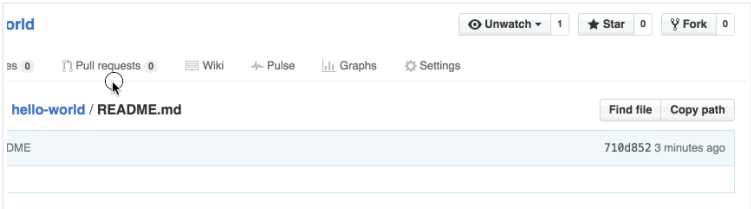

As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

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.

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.

Open a Pull Request for changes to the README

Click on the image for a larger version

Step	Screenshot
Click the  Pull Request tab, then from the Pull Request page, click the green New pull request button.	
In the Example Comparisons box, select the branch you made, <code>readme-edits</code> , to compare with <code>master</code> (the original).	

Step	Screenshot																					
<p>Look over your changes in the diffs on the Compare page, make sure they're what you want to submit.</p>	<div><div>什么是GitHub ?</div><div>1 commit 创建存储库 1 file changed</div><div>Commits on Oct 27, 2014 创建一个分支</div><div>hubot 做一个提交 finish README ...</div><div>打开拉取请求</div><div>合并拉取请求</div><div>Showing 1 changed file with 1 addition and 1 deletion.</div><div>2 README.md</div><div><table><tr><td>...</td><td>...</td><td>@@ -1,4 +1,4 @@</td></tr><tr><td>1</td><td>1</td><td>hello-world</td></tr><tr><td>2</td><td>2</td><td>=====</td></tr><tr><td>3</td><td>3</td><td></td></tr><tr><td>4</td><td></td><td>-Just another repository</td></tr><tr><td></td><td>4</td><td>+Hubot here, I like Node.js and Coffee</td></tr><tr><td></td><td></td><td>them far superior to Earth tacos.</td></tr></table></div></div>	@@ -1,4 +1,4 @@	1	1	hello-world	2	2	=====	3	3		4		-Just another repository		4	+Hubot here, I like Node.js and Coffee			them far superior to Earth tacos.
...	...	@@ -1,4 +1,4 @@																				
1	1	hello-world																				
2	2	=====																				
3	3																					
4		-Just another repository																				
	4	+Hubot here, I like Node.js and Coffee																				
		them far superior to Earth tacos.																				
<p>When you're satisfied that these are the changes you want to submit, click the big green Create Pull Request button.</p>	<div><div>base: master ... compare: readme-ed</div><div>Create pull request Discuss and review the</div></div>																					
<p>Give your pull request a title and write a brief description of your changes.</p>	<div><div>base: master ... compare: readme-edits</div><div>Readme edits</div><div>Write Preview</div><div>Content for non-telepathic human.</div></div>																					

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

Tip: You can use [emoji](#) and [drag and drop images and gifs](#) onto comments and Pull Requests.


[介绍](#)[什么是GitHub？](#)[创建存储库](#)[创建一个分支](#)[做一个提交](#)[打开拉取请求](#)[合并拉取请求](#)

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.

1. Click the green **Merge pull request** button to merge the changes into `master`.
2. Click **Confirm merge**.
3. Go ahead and delete the branch, since its changes have been incorporated, with the **Delete branch** button in the purple box.




This branch has no conflicts with the base branch
Merging can be performed automatically.

 **Merge pull request** You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Pull request successfully merged and closed
You're all set—the `readme-edits` branch can be safely deleted.

 **Delete branch**

Celebrate!

By completing this tutorial, you've learned to create a project and make a pull request on GitHub! 🎉 🐙 ⚡

Here's what you accomplished in this tutorial:

- Created an open source repository
- Started and managed a new branch
- Changed a file and committed those changes to GitHub
- Opened and merged a Pull Request

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 🐙

[创建存储库](#)

[创建一个分支](#)

[做一个提交](#)

[打开拉取请求](#)

[合并拉取请求](#)

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

最后更新于2016年4月7日



GitHub是构建和发布软件的最佳方式。
开源和私有项目的强大协作，代码审查和代码管理。