



Hello World

🕒 10 minute read

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!

Hello World 项目是计算机编程中历史悠久的传统。这是一个简单的练习，让你开始学习新的东西。让我们从 GitHub 开始吧！(2)

You'll learn how to:

你将学习如何：

- Create and use a repository
- Start and manage a new branch
- Make changes to a file and push them to GitHub as commits 对一个文件进行修改，并将它们作为提交文件推送到 GitHub
- Open and merge a pull request

Intro 介绍

[What is GitHub?](#)[Create a Repository](#)[Create a Branch](#)[Make a Commit](#)[Open a Pull Request](#)[Merge Pull Request](#)

What is GitHub?



什么是 GitHub?

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

GitHub是一个版本控制和协作的代码托管平台。它可以让你和其他人一起在任何地方的项目上工作。

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.

本教程向您介绍 GitHub 基础知识，比如 repositories、branches、commit 和 Pull Requests。您将创建自己的 Hello World 存储库，并学习 GitHub 的 Pull Request workflow，这是一种流行的创建和检查代码的方法。(2)

No coding necessary

不需要编码

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).

为了完成这个教程，你需要一个 github 的帐户和互联网接入。您不需要知道如何编写代码、使用命令行或安装 Git (版本控制软件 GitHub 是基于 Git 构建的)。(2)

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



提示: 在单独的浏览器窗口(或标签页)中打开本指南, 这样您可以在完成教程中的步骤时看到它。

Step 1. Create a Repository

第一步: 创建一个知识库⁽³⁾

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.*

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, 图片, 视频, 电子表格和数据集-任何你的项目需要。我们建议包含一个 README, 或者一个包含有关您的项目信息的文件。使得在创建新仓库的同时添加一个仓库变得非常容易。它还提供了其他常用选项, 如许可证文件。⁽³⁾

Your `hello-world` repository can be a place where you store ideas, resources, or even share and discuss things with others.

您的 hello-world 存储库可以是存储想法、资源甚至与他人分享和讨论事情的地方。

To create a new repository

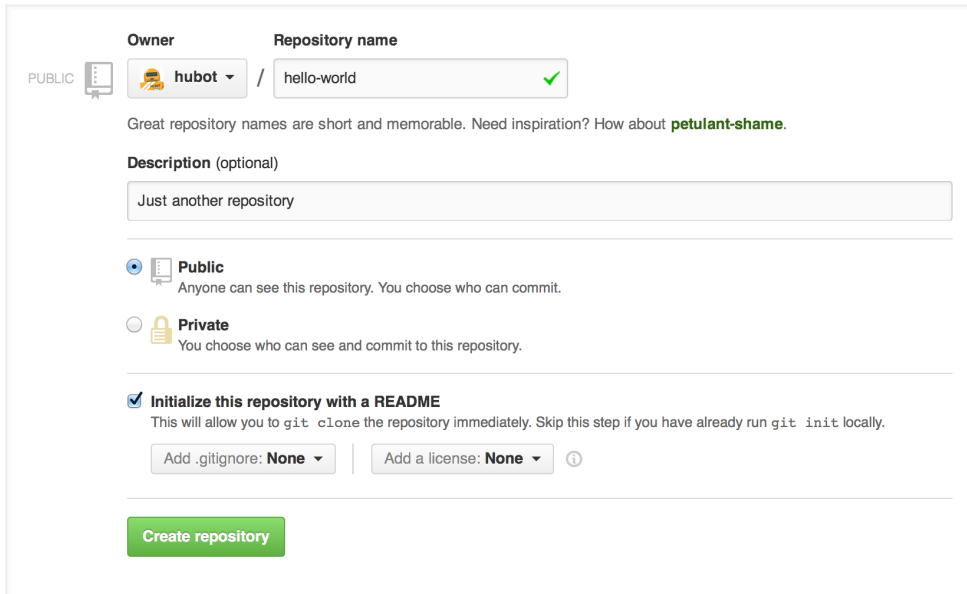
创建新的存储库

1. In the upper right corner, next to your avatar or identicon, click **+** and then select **然后选择New repository 新的储**



存库.

2. Name your repository 命名您的存储库 `hello-world` .
3. Write a short description. 写一个简短的描述
4. Select 选择 **Initialize this repository with a README** 使用 **README** 初始化此存储库.



Click **Create repository**.

单击 Create repository。

Step 2. Create a Branch

第二步：创建一个分支

Branching is the way to work on different versions of a repository at one time.

分支是同时处理存储库的不同版本的方法。

By default your repository has one branch named `main` which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to `main` .



默认情况下，存储库有一个名为 `main` 的分支，该分支被认为是最终的分支。我们使用分支进行实验，并在将它们提交到 `main` 之前进行编辑。

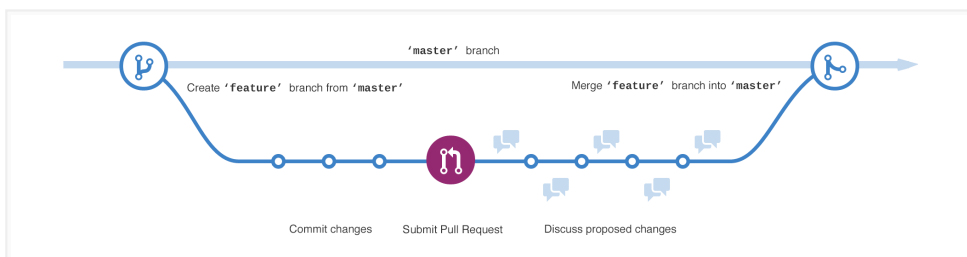
When you create a branch off the `main` branch, you're making a copy, or snapshot, of `main` as it was at that point in time. If someone else made changes to the `main` branch while you were working on your branch, you could pull in those updates.

当您在主分支之外创建一个分支时，您正在复制 `main` 的副本或快照，因为它在当时是这样的。如果其他人在您处理分支时对主分支进行了更改，您可以提取这些更新。

This diagram shows:

下图显示:

- The 这个 `main` branch 分支
- A new branch called 一个新的分支叫做 `feature` (because we're doing 'feature work' on this branch) (因为我们要在这个分支上做特写工作)
- The journey that 这段旅程 `feature` takes before it's merged into 在合并之前 `main`



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

你有没有保存过不同版本的文件:

- `story.txt`
- `story-joe-edit.txt`
- `story-joe-edit-reviewed.txt`



Branches accomplish similar goals in GitHub repositories.

在 GitHub 存储库中，分支实现了类似的目标。

Here at GitHub, our developers, writers, and designers use branches for keeping bug fixes and feature work separate from our `main` (production) branch. When a change is ready, they merge their branch into `main`.

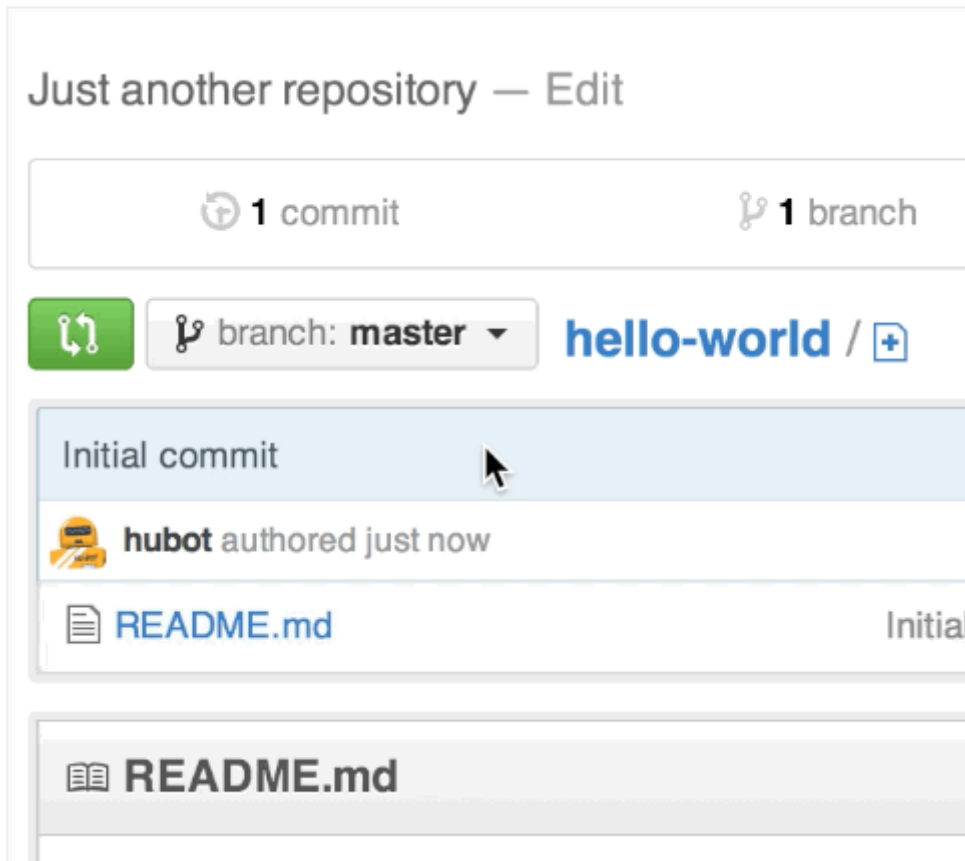
在 GitHub 上，我们的开发人员、编写人员 and 设计人员使用分支来将 bug 修复和特性工作与我们的主(生产)分支分离开来。当更改准备就绪时，它们将其分支合并到 main。

To create a new branch

创建一个新的分支

1. Go to your new repository 转到新的存储库 `hello-world`.
2. Click the drop down at the top of the file list that says 单击文件列表顶部的下拉菜单 **branch: main 分支: main**.
3. Type a branch name, 键入分支名称, `readme-edits`, into the new branch text box. , 放入新的分支文本框
4. Select the blue 选择蓝色 **Create branch 创建分支** box or hit “Enter” on your keyboard. 或者在键盘上按下“回车”键





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

现在您有两个分支，`main` 和 `readme-edits`。他们看起来完全一样，但不会长久！接下来，我们将向新分支添加更改。

Step 3. Make and commit changes

第三步：做出并提交改变

Bravo! Now, you're on the code view for your `readme-edits` branch, which is a copy of `main`. Let's make some edits.

太棒了！现在，您处于 `readme-edits` 分支的代码视图中，该分支是 `main` 的副本。我们来编辑一下。




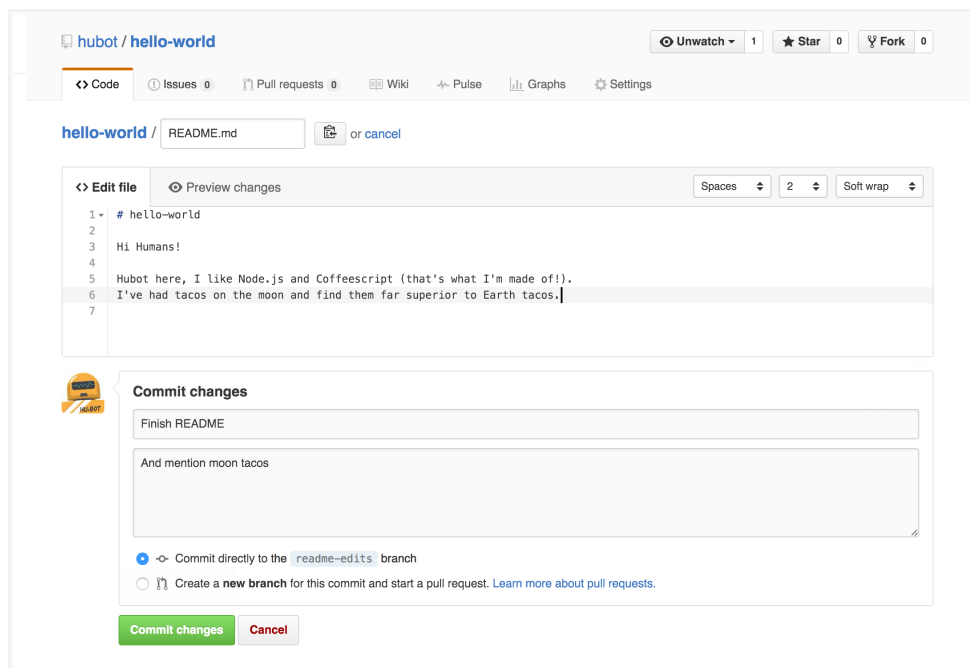
On GitHub, saved changes are called *commits*. Each commit has an associated *commit message*, which is a description explaining why a particular change was made. Commit messages capture the history of your changes, so other contributors can understand what you've done and why.

在 GitHub 上，保存的更改称为 commit。每个提交都有一个相关联的提交消息，这是一个解释为什么进行特定更改的描述。提交消息捕获更改的历史记录，这样其他贡献者就可以理解您做了什么以及为什么这样做。

Make and commit changes

做出并提交改变

1. Click the 点击 `README.md` file. 文件
2. Click the 点击  pencil icon in the upper right corner of the file view to edit. 文件视图右上角的铅笔图标进行编辑
3. In the editor, write a bit about yourself. 在编辑器中，写一些关于你自己的东西
4. Write a commit message that describes your changes. 编写一个提交消息，描述您的更改
5. Click 点击 **Commit changes** 提交改变 button. 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 `main`.

这些更改将只对 README 编辑分支上的 README 文件进行，因此现在这个分支包含与 main 不同的内容。

Step 4. Open a Pull Request

第四步：打开一个拉请求

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

编辑得不错! 现在您已经在 main 之外的一个分支中进行了更改，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.

请求是 GitHub 上协作的核心。当你打开一个拉请求，你提出你的变化，并要求有人审查和拉入你的贡献，并把他们合并到他们的分支。拉请求显示两个分支的内容的不同或差异。变化、增加和减少用绿色和红色表示。

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

一旦提交一个提交，你就可以打开一个拉请求并开始一个讨论，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.

通过在你的请求消息中使用 GitHub 的@mention 系统，你可以从特定的人或团队那里获得反馈，不管他们是在大厅里还是在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拉请求并自己合并它们。在从事大型项目之前，这是学习 GitHub 流程的一个很好的方法。

Open a Pull Request for changes to the README


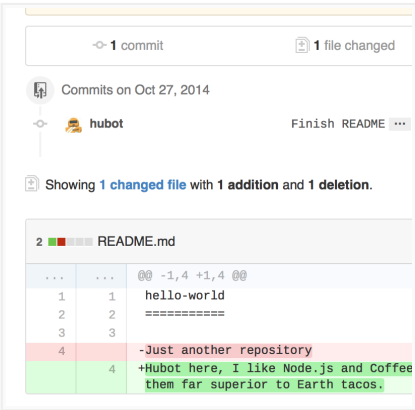
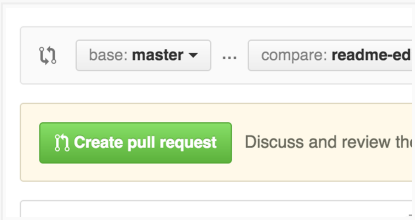
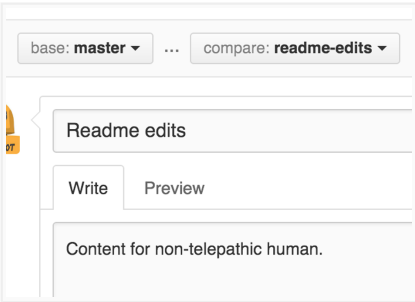
打开一个请求更改 README 的请求

Click on the image for a larger version

点击图片查看更大的版本

Step 第一步	Screenshot 截图
Click the Pull Request 拉请求 tab, then from the Pull Request page, click the green New pull request 新的拉请求 button. 按钮	



Step 第一步	Screenshot 截图
<p>In the Example Comparisons 例子比较 box, select the branch you made, 方框中，选择你做的树枝, <code>readme-edits</code>, to compare with, 以比较 <code>main</code> (the original). (原件)</p>	
<p>Look over your changes in the diffs on the Compare page, make sure they're what you want to submit. 在比较页面的差异中查看您的更改，确保它们是您想要提交的内容</p>	
<p>When you're satisfied that these are the changes you want to submit, click the big green Create Pull Request 创建拉请求 button. 按钮</p>	
<p>Give your pull request a title and write a brief description of your changes. 给你的拉请求一个标题，并写一个简短的描述你的变化</p>	

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



当您完成您的消息，单击创建拉请求！

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，拖放图片到评论和拉请求上。


Step 5. Merge your Pull Request

第五步：合并你的拉请求


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


在最后一步中，是时候将更改放在一起了——将自述文件编辑分支合并到主分支中。

1. Click the green **Merge pull request** 合并拉请求 button to merge the changes into `main` .
2. Click **Confirm merge** 确认合并.
3. Go ahead and delete the branch, since its changes have been incorporated, with the **Delete branch** 删除分支 button in the purple box. 紫色盒子里的纽扣




✓ 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!

庆祝吧

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，你已经学会了在 GitHub 上创建一个项目并发出一个拖动请求！

Here's what you accomplished in this tutorial:

下面是你在本教程中完成的：

- Created an open source repository 创建一个开放源代码仓库
- Started and managed a new branch 启动和管理一个新的分支
- Changed a file and committed those changes to GitHub 修改了一个文件并将这些修改提交给 GitHub
- Opened and merged a Pull Request 打开并合并拉请求

Take a look at your GitHub profile and you'll see your new [contribution squares](#)!

看看你的 GitHub 档案，你会看到你的新贡献方块！

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.

要了解拉请求的更多功能，我们推荐阅读 GitHub 流指南。你也可以访问 GitHub Explore，参与一个开源项目。

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 频道和点播培训, 了解更多如何开始使用 GitHub。

Last updated July 24, 2020

2020年7月24日



GitHub is the best way to build and ship software.
Powerful collaboration, code review, and code management for open source and private projects.

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。

