Hello world 是一个历史悠久的计算机编程项目。它是一个在你学习新东西的时候让你起步的简单练习。现在让我们一起来学习下GitHub。

你将学习如何：

* 创建并使用一个repository
* 开始并管理一个branch
* 编辑文档并提交到GitHub
* 打开和合并一个pull request

**什么是GitHub？**

GitHub是一个用于版本控制和协作的代码托管平台。有了它，你将可以和任何地方的人一起开发同一个项目。

这个教程将指导你学习GitHub的一些主要功能比如仓库、分支、文件提交还有请求。接下来你将创建一个你自己的Hello World 仓库，同时认识到一种创建和审查代码的十分流行的方法，那就是GitHub的pull request 工作流。

**不一定要会coding**

完成这个教程，你需要一个GitHub账号并且能够连接网络。你可以不懂如何编程或者不懂如何使用命令行，或者是安装Git（版本控制软件GitHub所运行的平台）。

提示：在一个单独的浏览器窗口（或者网页）打开这个导航，，这样你就能够在完成导航步骤的同时也能看到该教程。

**第一步：创建一个repository**

一个仓库一般用来管理一个项目。仓库包含文件夹、文件、图片、视频、电子表格还有数据集，只要是你项目所需要的都包括在内。我们建议每个仓库都附带一个readme或者一个介绍你项目基本信息的文档。GitHub上能够很方便地在你创建新的repository时添加一个此类文档。同样，GitHub也提供了其他的一些通用选项比如授权文档。

你的hello world repository 可以是一个存放你的想法、资源的地方，甚至也可以是一个你和他人共享资源和讨论问题的地方。

**如何创建一个repository**

1. 在右上角，靠近你的头像或者身份标识的地方，点击“+”然后选择新建一个repository
2. 给你的repository命名为hello-world
3. 简单描述下你的repository
4. 选择初始化这个repository并创建一个readme

如图

点击 创建 repository

**第二步: 创建一个branch**

Branching 是一种实现在一个repository中同时在其不同版本下工作的方法。默认设置中每一个repository都会有一个叫做master的分支，这个分支是用于存放一些已经被验证无问题的文档的。我们使用其他branches 来测试并修改，最后在提交到master中。

当你从master branch中创建一个branch时，最好做个拷贝，或者做个快照。当其他人在你更改你的branch时更改了master branch，你就能够将这些更新回退。

下面这个图标可以看到：

* Master branch
* 一个命名为feature的新的branch（因为我们正在对这个branch进行featurework）
* Feature这个branch在合并到master 这个branch之前所经历的过程

你有没有保存过一个文档的多个不同版本？比如：

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

Branch在GitHub repositories 中完成的任务就跟上述类似。

在GitHubzhong，我们的开发者、码农和架构师使用branches来保存bug修复、完成feature work，从而和我们的master（就是产品）branch区分开来。

当一次更新完成后，就可以把他们合并到master中。

**如何创建一个新的Branch**

1. 进到你新创建的repository hello-world
2. 点击branch:master 旁边的下拉框
3. 填入一个新的branch名字，readme-edits
4. 点击蓝色的 创建branch 按钮，或者直接按回车键

现在你有了两个branch，master和readme-edits。它们现在看起来一样，但是很快就不会了。接下来我们将更新刚才创建的branch。

**第三步:编辑并提交更新**

太厉害了！现在，你在readme-edits这个branch的编辑窗口，此时readme-edits是master的一个复制品。让我们一起来编辑一下。

在GitHub上，保存更新被称作是commits。每一个commit都有一个关联的commit信息，这个信息是用来描述更新目的的。通过这个commit信息，其他人可以简单明了地看出你更新的内容以及更新的原因。

**更新并提交**

1. 点击 readme.md文档
2. 点击右上角笔图标，开始编辑
3. 在编辑器中，简单介绍下你自己
4. 写一个commit信息来描述你的更新
5. 点击提交更新按钮。

**第四步 打开一个 pull request**

干得漂亮！现在你已经更新了你从master 拷贝出来的branch，你可以打开一个pull request了。

Pull requests 是GitHub上协同工作的核心。当你打开一个pull request时，你就提交了你的更新，然后申请某个人来审查，审查完毕后提交你的成果，将它合并到master中。Pull request会标记出两个branch不同的地方。更改、添加以绿色标识，删除以红色标识。

但那个你确认了更新之后，你就可以打开一个pull request然后开启一个讨论区，即便是编辑已经完毕。

通过使用GitHub中 的@ 某人 功能到你的pull request信息中，你可以请求某个特定的人或者组织给以反馈，不管他们是否在线或者是跨十个时区。

你也可以在你自己的repository中打开一个pull request，然后自己合并它们。在做大项目之前，这是一个很好的学习GitHub 工作流程的方法。

**给readme 中的更改 打开一个 pull request**

* 点击pull request 按钮，然后在pull request 页面中点击绿色的新建pull request 按钮
* 选择你创建的branch，和master进行对比
* 在对比页面中浏览你的更改内容，确保就是你想要提交的内容
* 当你确定了这些就是你要提交的更新，点击绿色的创建pull request按钮
* 给你的pull request 命名，然后写一个简短的介绍

当你编辑完了你的信息后，点击创建pull request！

提示：你可以在评论和pull requests 使用表情、图片和动图。

**第五步:合并你的pull request**

这是最后一步，此时，将会把你所有的更新汇聚起来，将你的readme-edits branch 合并到master branch 中。

1. 点击绿色的合并pull request按钮，即可将更新合并到master中
2. 点击确认合并
3. 现在已经将更新合并了，继续通过灰色的删除branch 按钮即可删除该branch

**恭喜！**

完成了本教程后，你已经学会了如何创建一个项目和如何在GitHub中使用pull request!

下面是在本教程中你所完成的内容：

* 创建了一个开源的repository
* 开启并管理了一个新的branch
* 更新了文件然后提交这些更新到GitHub
* 打开并合并了一个pull request

查看你的GitHub文档，你将发现你有了新的贡献文档。

如果你需要学习更多有关pull requests的功能，我们推荐你阅读GitHub Flow Guide。你也可以访问GitHub Explore 然后了解一下其他的开源项目。

提示：更多信息请访问我们其他的guide和youtube 频道。