产品

· 主页

· 工作

· 博客

Git 和 GitHub 入门指南（教程）

2015 年 10 月 1 日/作者 MEGHAN NELSON

· 喜欢

· 分享

· 喜欢

8 月，我们在 HubSpot 公司举办了女性工程师聚会，并为初学者举办了有关使用 git 和 GitHub 的研讨会。我首先浏览了有关 git 的基础知识和背景的幻灯片演示，然后我们分成小组来运行我创建的用来模拟大型协作项目的教程。

活动结束后我们得到了反馈，反馈者称这是一个实用性的操作介绍。因此，如果您还是 git 的新手，请按照以下步骤操作，可轻松更改代码库，打开 pull request（PR）并将代码合并到 master 分支中。任何重要的 git 和 GitHub 术语均以粗体显示，并带有指向官方 git 参考资料的链接。

步骤 0：安装 git 并创建一个 GitHub 帐户

首先您现在需要做的是安装 git 并创建一个免费的 GitHub 帐户。

请按照此处的说明安装 git（如果尚未安装）。请注意，对于本教程，我们将仅在命令行上使用 git。尽管有一些很棒的 git GUI（图形用户界面），但我认为先使用 git 特定的命令学习 git，然后在更熟悉该命令后尝试 git GUI 会更容易。

完成此操作后，在此处创建一个 GitHub 帐户。 （公共仓库是免费的，但私人仓库是收费的。）

步骤 1：建立本地 git 存放区

使用 git 在本地计算机上创建新项目时，首先要创建一个新的仓库（或简称为“ repo”）。

要使用 git，我们将使用终端。如果您对终端和基本命令没有太多经验，请查看本教程（特别是“导航文件系统”和“移动”部分）。

首先，打开一个终端，然后使用 cd（更改目录）命令移动到要在本地计算机上放置项目的位置。例如，如果您的桌面上有一个“项目”文件夹，则可以执行以下操作：

要在文件夹的根目录中初始化 git 仓库，请运行 git init 命令

步骤 2：将新文件添加到存储库 使用任何一种您喜欢的文本编辑器或运行 touch 指令，继续将新文件添加到项目中。

一旦您在包含 git 存储库的文件夹中添加或修改了文件，git 就会注意到在存储库中进行了 更改。但是，除非您明确告知 Git，否则 Git 不会正式跟踪该文件(即，将该文件提交-接下 来我们将详细讨论提交)。

创建新文件后，可以使用 **git status** 命令查看 git 中已经存在的文件。

这基本上是说：“嘿，我们注意到您创建了一个名为 mnelson.txt 的新文件，但是除非您使用 'git add'指令，否则我们将不会对其进行任何处理。

一段插曲：过渡环境、提交和你的关系

初学 git 时，最容易混淆的部分之一是暂存环境的概念及其与提交的关系。

commit 是自上次提交以来您更改了哪些文件的记录。本质上，您可以对存储库进行更改（例如， 添加文件或修改文件），然后告诉 git 将这些文件提交。

提交则表明项目创建成功，并随时可回到项目原始状态。

那么，如何告诉 git 提交哪些文件呢？这就是

staging environment 或 **index** 的引入。如步骤 2 所示，当您 对存储库进行更改时，git 会注意到文件已更改，但不会对其进行任何操作（例如在提交中 添加文件）。

要将文件添加到提交中，首先需要将其添加到登台环境中。为此，您可以使用 **git add** ****指令（请参见下面的步骤 3）。

使用 git add 指令将所需的所有文件添加到暂存环 境后，您可以使用 **git commit** 指令，告诉 git 将它们打包到提交中。

注意： 暂存环境，也称为“暂存”，这是一个新的首选术语，但您也可以将其称为“索引”

步骤 3:将文件添加到临时环境中

使用 git add 命令将文件添加到临时环境中。

如果您重新运行 git 状态指令，您将看到 git 已经将文件添加到 临时环境(请注意“要提交的更改”行)。

重申一下，该文件尚未添加到提交中，但即将添加到提交中。

步骤 4:创建提交

是时候进行您的第一次提交了！

运行命令 git commit m“关于提交消息”

提交末尾的消息应该与提交所包含的内容有关-可能是一项新功能，可能是一个

错误修复，也许只是在修复输入错误。不要输入“ asdfadsf”或“ foobar”之类的消息。那会让其他看到你的提交的人难过。非常非常悲伤。

步骤 5：创建一个新分支

现在，您已经进行了新的提交，让我们尝试一些更高级的东西。

假设您要制作一个新功能，但担心在开发功能时对主项目进行更改。这是 **git branches** 进来的地方。

分支允许您在项目的“状态”之间来回移动。例如，如果要向网站添加新页面，则可以为该页面创建一个新分支，而不会影响项目的主要部分。

完成页面后，您可以从合并您的更改 分支到主分支。当您创建一个新分支时，Git 将继续跟踪提交的分支，因此它知道所有文件的历史记录。

假设您在主分支上，希望创建一个新分支来。开发你的网页。下面是你要做的:

运行 git 结帐 分支名称。运行 gitcheckout -b <我的分支名称>。此命令将自动

创建一个新分支然后在其上“签出”，这意味着 git 会将您从主分支移到该分支。

运行上述命令后，可以使用 **git branch** 命令来确认您的分支已创建：

分支名称旁边带有星号表示在给定时间指向的分支。

现在，如果您切换回主分支并进行其他提交，则在您将这些更改合并到新分支之前，新分支将看不到任何更改。

步骤 6：在 GitHub 上创建一个新的存储库

如果只想在本地跟踪代码，则无需使用 GitHub。但是，如果您想与团队合作，则可以使用 GitHub 协同修改项目的代码。

要在 GitHub 上创建新的存储库，请登录并转到 GitHub 主页。

您应该看到一个绿色的“ +新存储库”按钮：

单击按钮后，GitHub 会要求您命名存储库并提供简短说明：

填写完信息后，请按“创建存储库”按钮以创建新的存储库

GitHub 会询问您是否要从头开始创建新的存储库，或者是否要添加在本地创建的存储库。在这种情况下，由于我们已经在本地创建了新的存储库，因此我们希望将其推送到 GitHub 上，因此请遵循“ ....或从命令行推送现有存储库”部分：

（由于您的 GitHub 用户名和存储库名称不同，因此您需要将第一个命令行中的 URL 更改为本节中 GitHub 列出的内容。）

步骤 7：将分支推送到 GitHub

现在，我们将分支中的提交推送到新的 GitHub 存储库。 这使其他人可以看到您所做的更改。 如果它们是由存储库所有者批准的，则可以将更改合并到主分支中。

要将更改推送到 GitHub 上的新分支，您需要运行

git push origin yourbranchname. GitHub 将自动创建

远程存储库的分支：

您可能想知道上面命令中“来源”一词的含义。发生的事情是，当您将远程存储库克隆到本地计算机时，git 为您创建了一个别名。在几乎所有情况下，此别名都称为“来源”。它实际上是远程存储库 URL 的简写。

因此，要将更改推送到远程存储库，可以使用以下命令之一：**git push git@github.com:git/git.git yourbranchname** or **git push origin yourbranchname**

（如果这是您第一次在本地使用 GitHub，则可能会提示您使用 GitHub 用户名和密码登录。）

如果刷新 GitHub 页面，您会看到注释，说您的名字的分支刚刚被推送到存储库中。您也可以单击“分支”链接以查看此处列出的分支。

现在，单击上方屏幕快照中的绿色按钮。我们将提出请求！

步骤 8：创建拉取请求（PR）

拉取请求（或 PR）是一种提醒存储库所有者想要的方式,对他们的代码进行一些更改.它允许他们查看代码并在将更改放入主分支之前，请确保它看起来不错。

这是提交 PR 页之前的样子：

这是您提交 PR 请求后的样子：

您可能会在底部看到一个绿色的大按钮，上面写着“合并拉” 请求'。单击这意味着您将您的更改合并到主数据库中 科。

请注意，此按钮并非总是绿色。在某些情况下，它会变成灰色，这意味着您**面临合并冲突**。 是当有一个 一个文件中的更改与另一个文件中的更改冲突，而 git 无法找出要使用的版本。您必须手动输入并告诉 git 使用哪个版本。

有时，您将是回购的共同所有者或唯一所有者，在这种情况下您可能不需要创建 PR 来合并您的更改。 但是，它仍然做一个好方法这样您就可以保留更完整的历史记录更新，并确保在创建时始终创建一个新分支变化。

步骤 9：合并 PR

继续并单击绿色的“合并拉取请求”按钮。单击这意味着您将您的更改合并到主数据库中。

将合并 您所做的更改进入主分支。

完成后，建议您删除分支（分支太多可能会变得凌乱），因此也请点击灰色的“删除分支”按钮。

您可以通过单击再次仔细检查您的提交是否已合并在新出现的首页上的“提交”链接。这将显示该分支中所有提交的列表。

你可以看到 混列码 您还可以在右侧看到提交的混列码。一种混列码是该特定提交的唯一标识符。这很有用

用于引用特定的提交以及撤消更改时 (用 **git 撤销 混**< 密码 > 命令还原).

步骤 10：将 GitHub 上的更改返回到您的计算机

现在，GitHub 上的仓库看起来与本地计算机上的仓库有所不同。例如，您在分支中进行的提交并合并到 master 分支中的提交在本地计算机上的 master 分支中不存在。为了获得您或其他人已在 GitHub 上合并的最新更改，请使用 git pull origin master 命令（在 master 分支上工作时）。这将向您显示所有已更改的文

件及其更改方式。 现在我们可以再次使用 git log 命令来查看所有新提交。（您可能需要将分支切换回主分支。您可以使用 git checkout master 命令执行此操作。）

步骤 11：享受你的 git glory

您已经成功进行了 PR，并将您的代码合并到 master 分支中。恭喜你！如果您想更深入一点，请查看 this Git101 folder 中的文件，以获取有关使用 git 和 GitHub 的更多提示和技巧。 我也还建议您花一些时间与您的团队一起模拟一个较小的小组项目，就像我们在这里所做的那样。让您的团队使用您的团队名称创建一个新文件夹，并向其中添加一些带有文本的文件。然后，尝试将这些更改推送到此远程仓库中。这样一来，您的团队就可以开始更改最初未创建的文件，并使用公关功能。并且，使用 GitHub 上的 git blame 和 git history 工具来获取熟悉跟踪文件中进行了哪些更改以及谁进行了这些更改。您使用 git 的次数越多，您将越感到舒适。（我无法抗拒。）Written by Meghan Nelson

话题

· PRODUCT

· ENGINEERING

· DESIGN

· PRODUCT MANAGEMENT

· UX

· CULTURE

· CAREER GROWTH

What is HubSpot|Our Story|Our Products|Culture Code

Facebook|Twitter|Instagramhttps://instagram.com/hubspotdev/)