

安装Git客户端

配置Git

配置ssh key

代码上传到GitHub

创建项目

本地初始化一个项目

完成第一次提交

后续上传代码步骤

注意

错误2

错误3

删除远程仓库里的文件

关于Git upstream和Git origin的说明

GitHub 仓库设置

创建和删除仓库中的分支

创建分支

删除分支

添加README.md

Clion 的 Git 配置

<https://www.jianshu.com/p/9ab51e00f4b3>

https://blog.csdn.net/loner_fang/article/details/80488385

安装Git客户端

[windows用户请戳](#)

配置Git

执行如下代码 生成 .git 文件夹,表示本地 git 创建成功

(注意 .git 文件夹是隐藏文件)

```
git config --global user.name "你的名字"
```

```
git config --global user.email "你的邮箱"
```

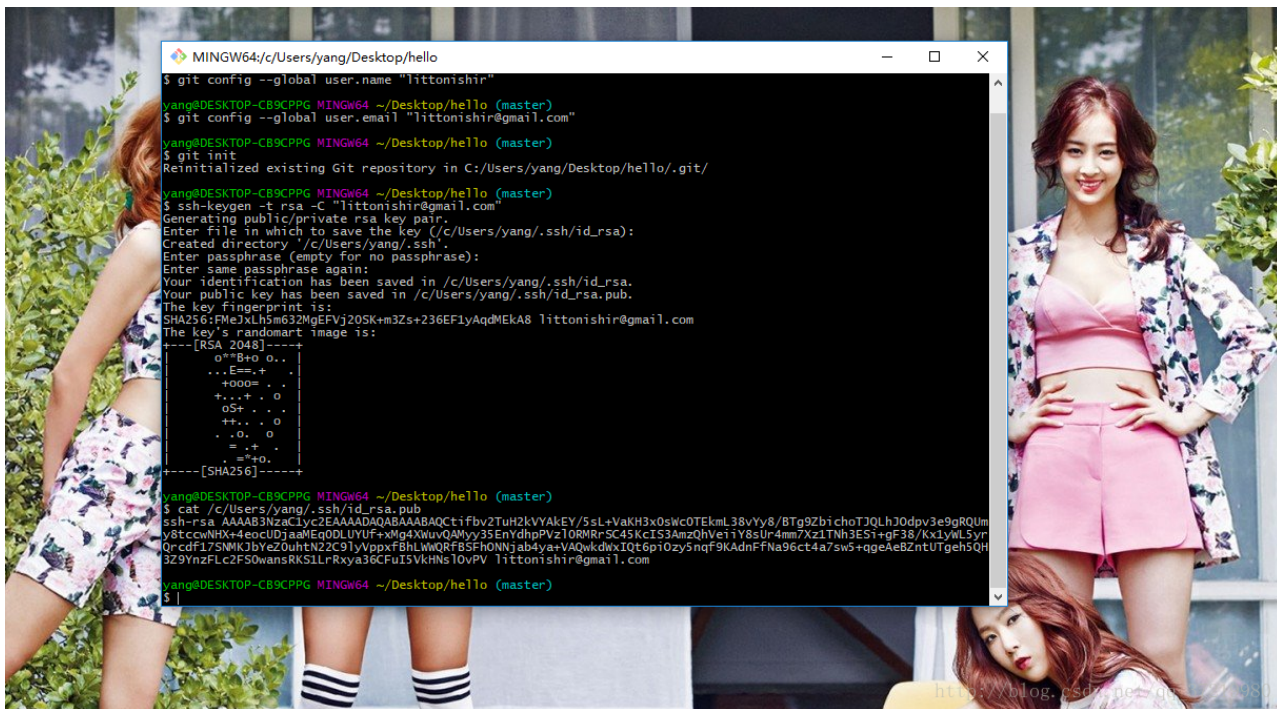
```
git init # 初始化Git仓库
```

配置ssh key

(我在用hexo搭建个人博客的时候以及配置了)

"你的邮箱" 然后一路回车, 直到出现randomart image 既可

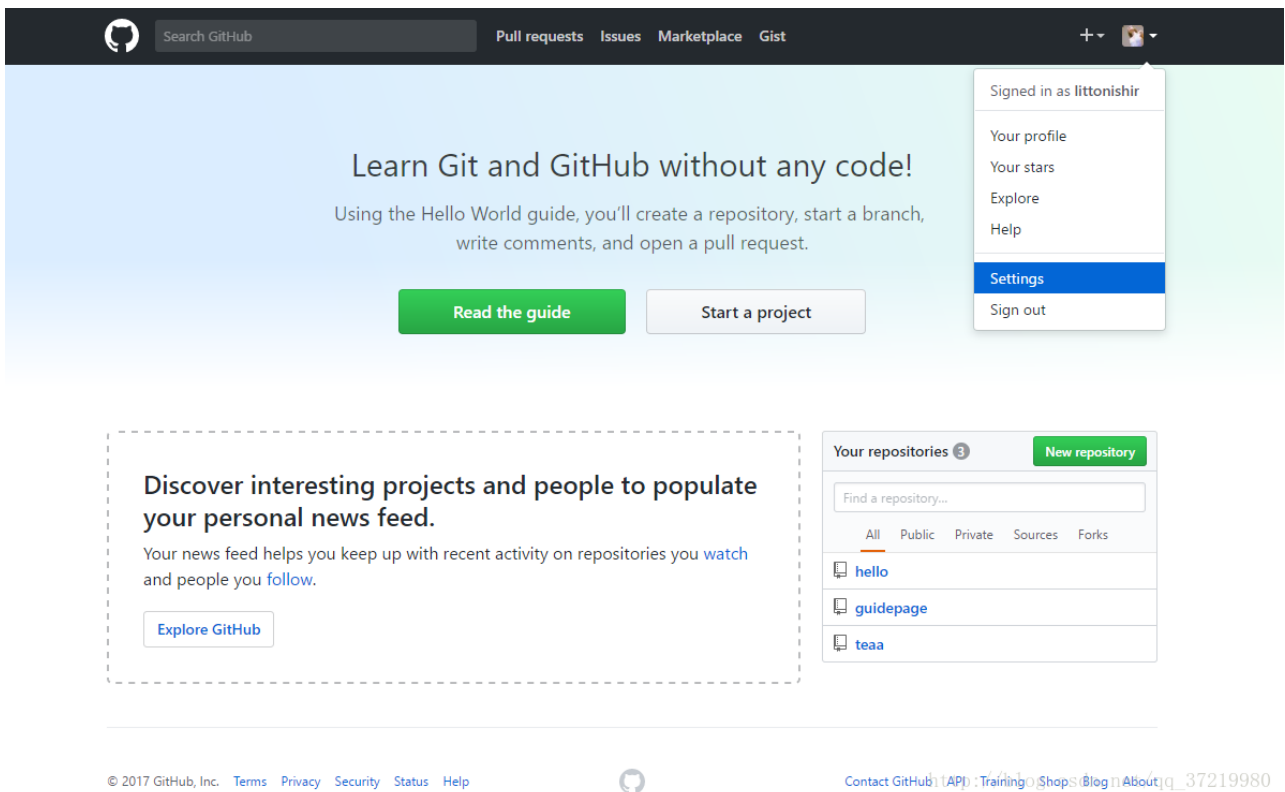
```
ssh-keygen -t rsa -C "1589353529@qq.com"
```



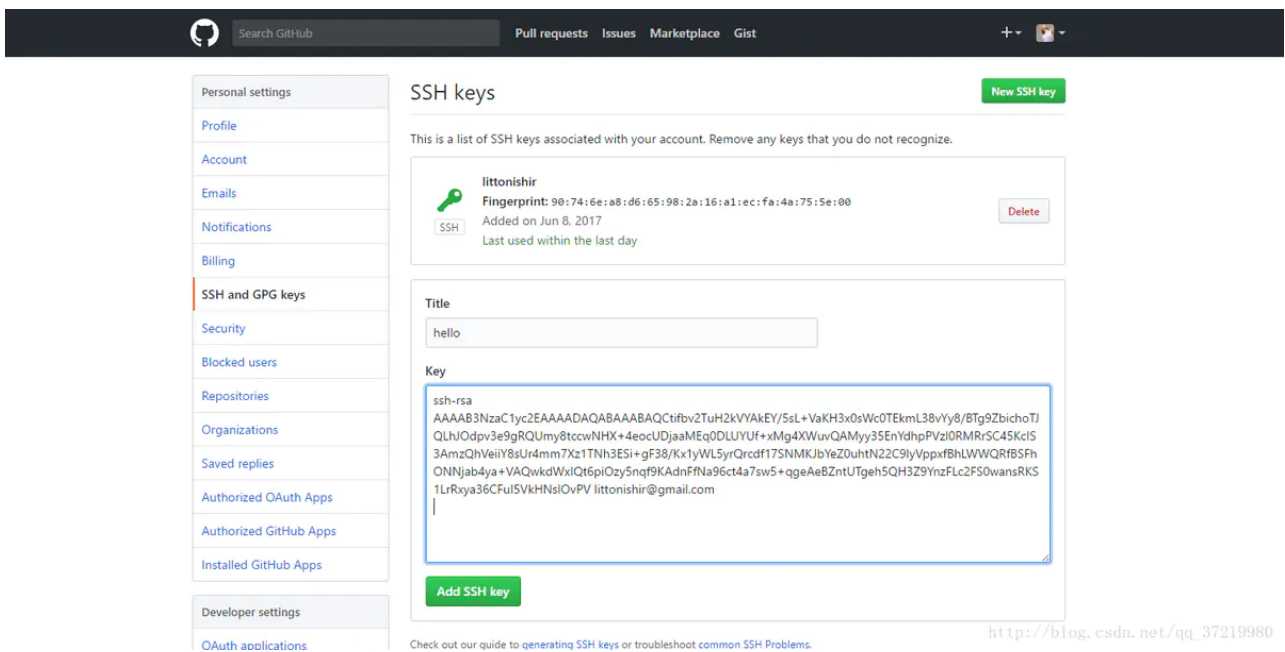
cat 一下 把key 复制下来

```
cat /c/Users/CAIYUAN/.ssh/id_rsa.pub
```

打开github,选择Account Settings



左边选择SSH and GPG Keys,点击New SSH key title不重要 将你复制的key粘贴进来 Add SSH Key既可



验证是否成功

```
ssh -T git@github.com
```

回车就会看到:

```
Hi <xxx>! You've successfully authenticated, but GitHub does not provide shell access.
```

这就表示已成功连上github.

```
CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ git log --oneline
475f34a (HEAD -> master) 21赛季最终版本

CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ ssh -T git@github.com
git@github.com: Permission denied (publickey).

CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ cat /c/Users/caihanyuan/.ssh/id_rsa.pub
cat: /c/Users/caihanyuan/.ssh/id_rsa.pub: No such file or directory

CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ ssh -T git@github.com
Hi Nepqiu! You've successfully authenticated, but GitHub does not provide shell
access.

CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ |
```

代码上传到GitHub

创建项目

1. 点击+号->new repository 填写Repository的名字 填写Repository的描述 (Initialize this repository with a README)可选
2. 创建完成后点击clone or download 复制Repository的地址

本地初始化一个项目

1. git 的基础配置，作用是告诉 git 你是谁

```
git config --global user.name "nepqiu"
```

```
git config --global user.email "1589353529@qq.com"
```

2. 在你的需要初始化版本库的文件夹中执行

```
git init
```

```
git remote add origin git@github.com:Nepqiu/ACE_Infantry_2021.git
```

完成第一次提交

1. 进入你已经初始化好的或者克隆项目的目录,然后执行

```
git pull origin master
```

```
git add .
```

2. 若出现该warning: LF will be replaced by CRLF in xxx文件(解决办法如下代码), 其实不管也可以

```
git config --global core.autocrlf false //禁用自动转换
```

3. 提交信息

```
git commit -m "提交的注释"
```

4. 正式上传

```
git push origin master
```

后续上传代码步骤

1. `git init` 将文件夹设置为本地仓库, 只有这样才能把本地的文件传入github仓库
2. `git remote add origin git@github.com:Nepqiu/ACE_Infantry_2021.git` 将本地仓库与github仓库进行关联 (SSH的地址)
3. `git pull origin master` 将GitHub上仓库的内容pull到本地仓库, 两者保持一致
4. `git add /` / `git add .` 需要上传的文件添加文件到本地库 (后面加 `.` 是指全部文件)
5. `git commit -m "xxx"` 提交文件到本地库, `xxx` 是提交注释的内容
6. `git push origin master` 上传文件, **origin 是别名**
7. `git push -u origin master` 上传代码到 **github** 远程仓库

如果要上传的文件是在一个新的文件夹里, 那么就需要执行前3步, 将新文件夹作为本地仓库与github关联;

如果要上传的文件是在之前的文件夹里, 那么之前已经关联过了, 只需要直接执行后3步就可以了。

```
git push -u origin master -f
```

 将刚刚添加的文件push到github。

刷新GitHub界面 看到刚刚添加的文件

注意

`origin` 只是别名而已

刚上传代码到 github 的时候,当我输入 `git push -u origin master` 的时,提示:

```
fatal: 'origin' does not appear to be a git repositoryfatal: Could not read from remote repository.
```

解决办法:

控制台中输入

```
//git remote add origin git@github.com:xxse/xx.gitgit remote add origin git@github.com:Nepqiu/ACE_Infantry_2021.git
```

再次输入

```
git push -u origin master
```

就可以正常提交了.

错误2

git pull 代码时报错:

```
CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ git push -u origin master
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
```

```
fatal: Could not read from remote repository.
Please make sure you have the correct access rights
and the repository exists.
```

在使用git提交代码时出现 **fatal: Could not read from remote repository** 这个错误, 很简单

首先, 将本地生成的 `id_rsa` 以及 `id_rsa.pub` 这两个文件删除掉

```
rm -rf ~/.ssh/id_rsa
rm -rf ~/.ssh/id_rsa.pub
```

然后, 使用命令 `ssh-keygen -t rsa` 重新生成密钥

```
CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ ssh-keygen -t rsa -C 1589353529@qq.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/CAIYUAN/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/CAIYUAN/.ssh/id_rsa
Your public key has been saved in /c/Users/CAIYUAN/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:pDmzKHxG55RsyRQM+5CL6tsMWxRCwtMaNI2WqVYGBnE 1589353529@qq.com
The key's randomart image is:
+---[RSA 3072]-----+
| BBE .o. |
| +X.= o.. |
| o.*.+ . . |
| .o...B * |
| . . .o # S |
| o.. * + |
| ..o.+ o |
| . *+ |
| +.o |
+----[SHA256]-----+
```

错误3

<https://blog.csdn.net/dietime1943/article/details/85682688>

https://blog.csdn.net/qg_39416311/article/details/102219428

问题描述：在git bash中键入 \$ git push origin master 进行提交的时候出现 如下错误：

error: failed to push some refs to '<https://github.com/bluetata/>'

问题原因：远程库与本地库不一致造成的，在hint中也有提示把远程库同步到本地库就可以了。

解决办法：使用命令行：

```
git pull --rebase origin master
```

该命令的意思是把远程库中的**更新合并到**（pull=fetch+merge）本地库中，--rebase的作用是取消掉本地库中刚刚的commit，并把他们接到更新后的版本库之中。出现如下图执行pull执行成功后，可以成功执行git push origin master操作。

```

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

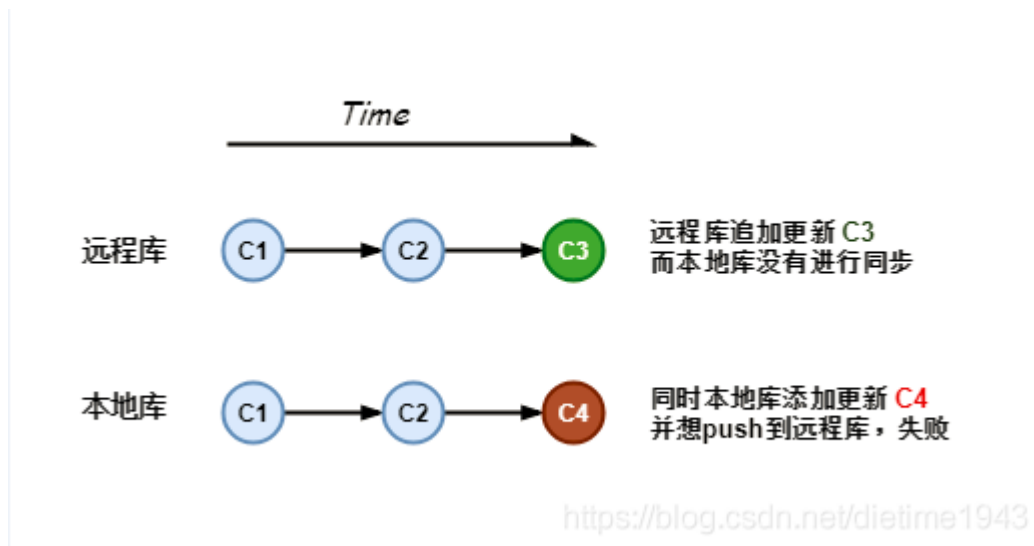
CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ git pull --rebase origin master
remote: Enumerating objects: 918, done.
remote: Counting objects: 100% (918/918), done.
remote: Compressing objects: 100% (453/453), done.
remote: Total 918 (delta 449), reused 915 (delta 449), pack-reused 0
Receiving objects: 100% (918/918), 4.93 MiB | 31.00 KiB/s, done.
Resolving deltas: 100% (449/449), done.
From github.com:Nepqiu/ACE_Infantry_2021
* branch      master      -> FETCH_HEAD
* [new branch] master      -> origin/master
Successfully rebased and updated refs/heads/master.

CAIYUAN@DESKTOP-MIV53ED MINGW64 /d/Experiment/ACE/CodeData/Infantry/git_infantry
(master)
$ git push -u origin master
Enumerating objects: 922, done.
Counting objects: 100% (922/922), done.
Delta compression using up to 12 threads
Compressing objects: 100% (660/660), done.
Writing objects: 100% (921/921), 5.61 MiB | 587.00 KiB/s, done.

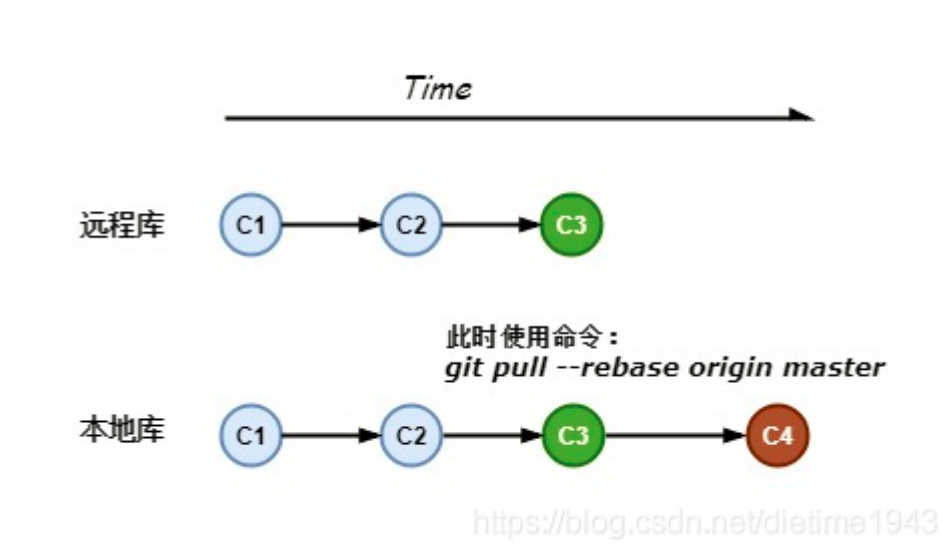
```

图形描述问题发生的原因及解决办法:

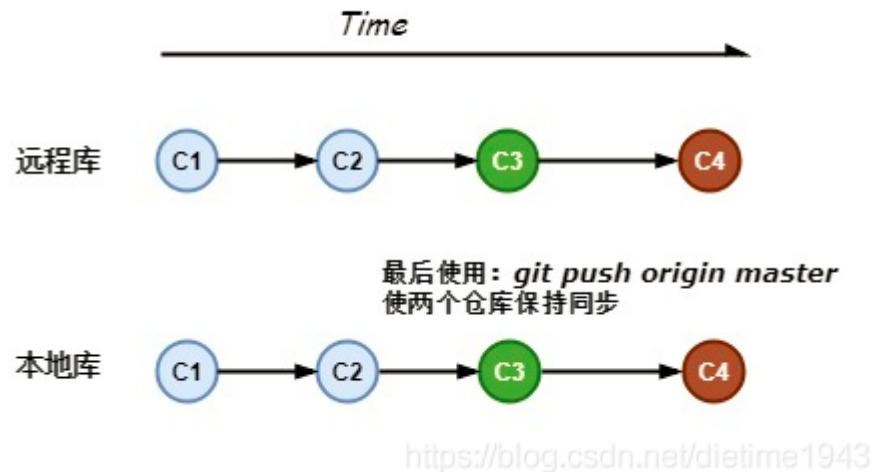
1. 发生问题时候的状态:



2. 执行 `git pull --rebase origin master` 操作，意为先取消commit记录，并且把它们临时保存为补丁(patch)(这些补丁放到“.git/rebase”目录中)，之后同步远程库到本地，最后合并补丁到本地库之中。



3. 最后把本地库push到远程库当中，使本地与远程仓库保持一致。



删除远程仓库里的文件

进入本地仓库：

`git pull origin master` 本地同步远程仓库，将远程仓库里的内容拉下来

`git rm -r --cached 文件名` 删除文件

`git commit -m "delete dir"` 提交并添加说明

`git push origin master` 将本次更改更新到github项目上去

关于Git upstream和Git origin的说明

1. `git remote -v`
2. `git remote add upstream https://github.com/xxx.git`
3. `git remote -v`

<https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/configuring-a-remote-for-a-fork>

1. `git fetch upstream`
2. `git checkout master`
3. `git merge upstream/master`

<https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/syncing-a-fork>

GitHub 仓库设置

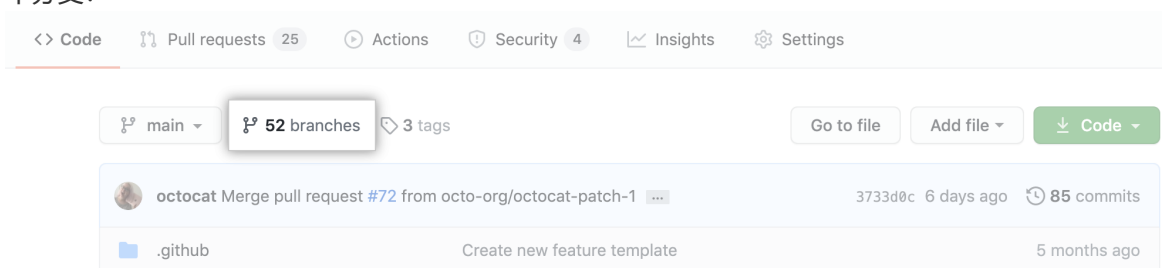
创建和删除仓库中的分支

[官方文档](#)

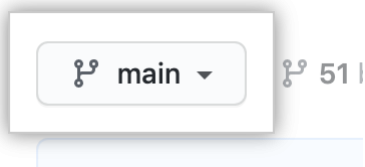
您可以直接在 GitHub 上创建或删除分支。

创建分支

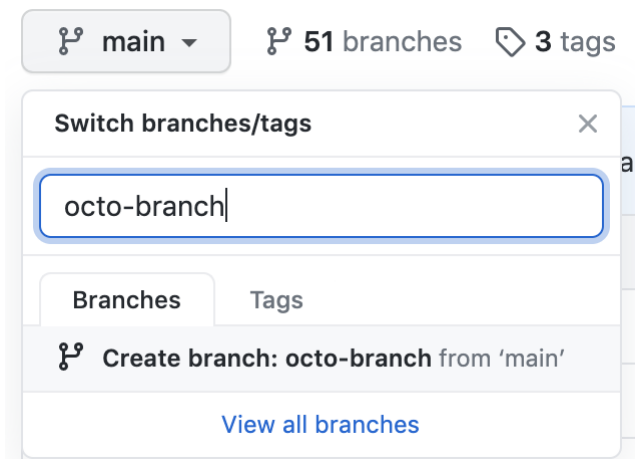
1. 在 GitHub 上，导航到仓库的主页面。
2. (可选) 如果要从仓库的默认分支以外的分支创建新分支，请单击 ***NUMBER* 分支**，然后选择另一个分支：



3. 单击分支选择器菜单。



4. 为新分支键入唯一名称，然后选择 **Create branch (创建分支)**。



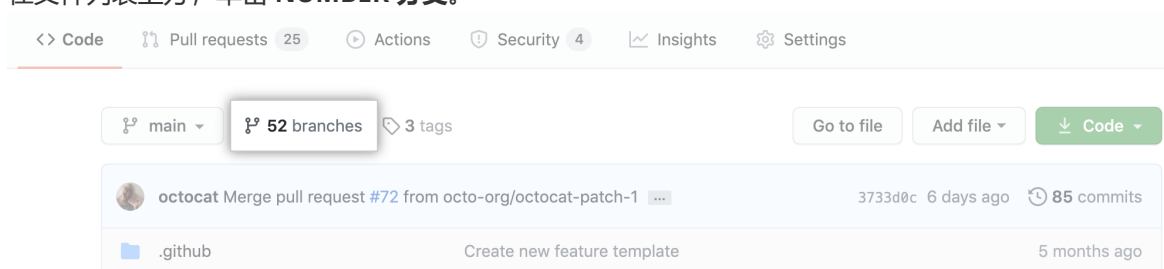
删除分支

您可以让头部分支在仓库中的拉取请求合并后自动删除。更多信息请参阅[“管理分支的自动删除”](#)。

注意：如果要删除的分支是仓库的默认分支，则在删除该分支之前必须选择新的默认分支。更多信息请参阅[“更改默认分支”](#)。

如果要删除的分支与打开的拉取请求关联，则在删除该分支之前必须合并或关闭拉取请求。更多信息请参阅“[合并拉取请求](#)”和“[关闭拉取请求](#)”。

1. 在 GitHub 上，导航到仓库的主页面。
2. 在文件列表上方，单击 **NUMBER 分支**。



3. 滚动到要删除的分支，然后单击。



如果您在拉取请求合并后删除头部分支，GitHub 会检查将已删除分支指定为基础分支的同一仓库中的任何开放拉取请求。GitHub 将自动更新任何此类拉取请求，将其基础分支更改为合并的拉取请求的基础分支。

更多信息请参阅“[关于分支](#)”。

添加README.md

<https://www.cnblogs.com/sxdcgaq8080/p/8058699.html>

Clion 的 Git 配置

<https://blog.csdn.net/Wistral/article/details/84400924>