

GitHub 快速上手教程

GitHub 快速上手教程

什么是GitHub

如何使用GitHub

Git 指南

本地创建

```
git clone /path/to/repository
```

```
git init
```

加入文件

```
git add ./Handout3.tex
```

提交

```
git commit -m "Adding files"
```

与远程同步

```
git remote add origin git@github.com:edmundwsy/new.git
```

```
git remote remove origin
```

```
git clone https://github.com/qyxf/Tutorials.git
```

```
git pull
```

```
git push origin master
```

设置公钥

分支

```
git checkout -b feature_x
```

```
git checkout master
```

```
git branch -d feature_x
```

```
git push origin <branch>
```

合并分支

要合并其他分支到你的当前分支（例如 master），执行：

```
git merge <branch>
```

```
git add <filename>
```

```
git diff <source_branch> <target_branch>
```

历史记录

```
git log
```

```
git log --author=bob
```

```
git log --pretty=oneline
```

```
git log --graph --oneline --decorate --all
```

```
git log --name-status
```

```
git log --help
```

一些问题

其它

什么是GitHub

github是一个基于git的代码托管平台，付费用户可以建私人仓库，我们一般的免费用户只能使用公共仓库，也就是代码要公开。目前已托管431万git项目

让人不再担心备份与远程协作，一切提交，均有记录。

这个星球上最流行的开源托管服务。

让人不再担心备份与远程协作，一切提交，均有记录。

如何使用GitHub

- 注册GitHub
 - 下载Git（GitHub依托的版本控制软件）或者是下载GitHub客户端（国内被墙）
-

Git 指南

本地创建

```
git clone /path/to/repository
```

创建一个本地仓库的克隆版本（从现有中拷贝）

```
git init
```

从本地创建新目录

初始化后，会在 newrepo 目录下会出现一个名为 .git 的目录，所有 Git 需要的数据和资源都存放在这个目录中。

加入文件

```
git add ./Handout3.tex
```

当然还有其它的选项，比如add文件夹里面所有文件 `git add --all`

提交

```
git commit -m "Adding files"
```

将修改提交版本库

- 工作区：就是你在电脑里能看到的目录。
- 暂存区：英文叫stage, 或index。一般存放在 ".git目录下" 下的index文件 (.git/index) 中，所以我们把暂存区有时也叫作索引 (index)。
- 版本库：工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。

与远程同步

```
git remote add origin git@github.com:edmundwsy/new.git
```

将本地库与远程关联

```
git remote remove origin
```

取消关联

```
git clone https://github.com/qyxf/Tutorials.git
```

将我们远程的repo克隆至本地

```
git pull
```

拉取远程更新，以在你的工作目录中 *获取 (fetch)* 并 *合并 (merge)* 远端的改动。

```
git push origin master
```

```
git push <远程主机名> <本地分支名> <远程分支名>
```

即是将本地的master分支推送到远程主机origin上的对应master分支，origin 是远程主机名，

```
git push -h
```

设置公钥

```
git config --global user.name "My Name" git config --global  
user.email myEmail@example.com
```

如果报错

```
$ git push origin master
git@github.com: Permission denied (publickey).
```

在GitHub设置远程公钥就可以解决这个问题

在GitHub网页端：Setting -> SSH Keys->New SSH key

- title处填写：id_rsa.pub
- `cat ~/.ssh/id_rsa.pub`
 - 若没有，则 `ssh-keygen -t rsa`
 - 一路回车.....
 - 最后，输入 `ssh -T git@github.com` 确认OK即可。
- 然后把这个复制到GitHub中，保存
- 再尝试就ok了

再解决不了了估计就是没有权限了。。。。。请联系仓库管理员

分支

```
git checkout -b feature_x
```

创建一个叫做“feature_x”的分支，并切换过去：

```
git checkout master
```

切换回主分支：

```
git branch -d feature_x
```

再把新建的分支删掉：

```
git push origin <branch>
```

除非你将分支推送到远端仓库，不然该分支就是 *不为他人所见的*

合并分支

要合并其他分支到你的当前分支（例如 master），执行：

```
git merge <branch>
```

在这两种情况下，git 都会尝试去自动合并改动。遗憾的是，这可能并非每次都成功，并可能出现*冲突 (conflicts)*。这时候就需要你修改这些文件来手动合并这些*冲突 (conflicts)*。改完之后，你需要执行如下命令以将它们标记为合并成功：

```
git add <filename>
```

在合并改动之前，你可以使用如下命令预览差异：

```
git diff <source_branch> <target_branch>
```

历史记录

如果你想了解本地仓库的历史记录，最简单的命令就是使用：

```
git log
```

你可以添加一些参数来修改他的输出，从而得到自己想要的结果。只看某一个人的提交记录：

```
git log --author=bob
```

一个压缩后的每一条提交记录只占一行的输出：

```
git log --pretty=oneline
```

或者你想通过 ASCII 艺术的树形结构来展示所有的分支，每个分支都标示了他的名字和标签：

```
git log --graph --oneline --decorate --all
```

看看哪些文件改变了：

```
git log --name-status
```

这些只是你可以使用的参数中很小的一部分。更多的信息，参考：

```
git log --help
```

一些问题

冲突怎么办？合并分支，手动修改 建议先把自己的提交，然后pull，最后commit，最大程度减少冲突的几率

其它

钱院官方站点: <https://github.com/qyxf>

git 简明指南 <http://rogerdudler.github.io/git-guide/index.zh.html>

git教程 <http://www.runoob.com/git/git-tutorial.html>

廖雪峰Git教程 <https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>

[廖雪峰Git教程](#)