

Github小白入门

第一步：GitHub 小白入门（一）初识界面

第二步：GitHub 小白入门（二）仓库操作

第三步：Git的安装与使用

第四步：GitHub下载提交代码

Git安装：一切按照默认安装即可

Git基础指令：

查看所有配置：git config --list
查看所有配置及所在文件：git config --list --show-origin

配置用户信息：

用户名：git config --global user.name "用户名"
用户名邮箱：git config --global user.email 邮箱地址
查看用户配置：git config user.name
git config user.email

Git常用指令

→ 找到文件，右击设定库

git status：查看仓库状态
git init：创建一个空仓库，或者重新初始化一个已有仓库
git add：把文件添加到可提交列表（临时缓冲区）
git commit：提交改动（增删改）至仓库
git log：打印提交日志
git branch：查看、添加、删除分支
git checkout：切换分支、标签
git merge：合并分支
git tag：新建、查看标签

Git基础配置：

git --version：返回版本号
git --help：帮助文档
git config --list：查看所有配置
git config --list --show-origin：查看所有配置以及它们所在的文件
git config --global user.name "用户名"：全局配置用户名
git config --global user.email 邮箱地址：全局配置邮箱
git config user.name：查看用户名
git config user.email：查看邮箱

再次强调，如果使用了--global选项，那么该命令只需要运行一次，因为之后无论你在该系统上做任何事情，Git都会使用那些信息。当你想针对特定项目使用不同的用户名称与邮件地址时，可以在那个项目目录下运行没有--global选项的命令来配置。

基本语法：
git config --global alias.别名 git命令

设置别名：

比如给 commit 和 status 起别名：
git config --global alias.ci commit
git config --global alias.st status
下次再用这两个命令就可以写：
git ci
git st
除了简单命令还可以设置组合命令，给组合命令加引号即可：
git config --global alias.pullom 'pull origin master'
git config --global alias.pushom 'push origin master'

显示改动

还记得 git add 会把文件放入暂存区吧，在git add 之前使用 git diff 可以显示你对某一文件的改动。
除此之外，git diff 还有其他用法：
git diff <\$id1> <\$id2> # 比较两次提交之间的差异
git diff <branch1> ..<branch2> # 在两个分支之间比较
git diff --staged # 比较暂存区和版本库差异

版本回滚

用 git log 查看版本号（commit 后面那一长串就是版本号）
找到想要回退的版本号之后，在本地 Git 仓库执行如下命令，选一个即可：
git reset --hard 版本号（抛弃当前工作区的修改）
git reset --soft 版本号（回退到之前的版本，但保留当前工作区的修改，可以重新提交）

如果你的文件只在本地，执行上面的步骤就可以啦。
如果你的文件提交到 GitHub 上，还需要执行如下命令同步到远端：
git push origin 分支名
如果本地的版本落后于远端的版本，因此我们还需要在上述命令中加上--force参数：
git push origin 分支名 --force

SSH配置

Gitbash上验证ssh的安装：git ssh
生成SSH key：ssh-keygen -t rsa 连续三次回车，生成指定目录的秘钥（id_rsa）和公钥(id_rsa.pub)
网页端GitHub上setting添加SSH key
验证绑定：ssh -T git@github.com

通过Git下载代码

Download Zip直接下载
git clone with HTTPS：选好目录，右击，点击Git bash here，在Git bash中输入：git clone 复制好的URL，例如：git clone https://github.com/project/repo.git，等待下载，可以看到文件夹中已有仓库中的文件。
git clone with SSH: 同理，例如：git clone git@github.com:project/repo.git

通过Git提交代码：

两个代码

git push：翻译为推，当你的代码更新，需要把本地的推到远程仓库
git pull：翻译为拉，当远程仓库有更新，你需要把远程的拉到本地进行合并

提交代码

例：git push origin master # 把本地代码推到远程 master 分支
例：git pull origin master # 把远程最新的代码更新到本地

http与SSH下载异同：
同：
HTTPs 与 SSH 下的 git 都可以直接进行 git clone 操作
异：
HTTPs git clone 到本地，进行了一些文件的修改，当再次提交到 GitHub远程服务器的时候，都会进行账号与密码的输入
SSH git clone 到本地之后，由于已有 SSH Keys 授权，就不需要用户名和密码进行授权了。

一般我们在 push 之前都会先 pull，这样不容易冲突。