# 安装win10 git客户端

GitHub 中公开的代码大部分都是以 Mac 或 Linux 中的 LF（LineFeed）换行。然而，由于 Windows 中是以 CRLF（Carriage Return ＋Line Feed）换行的，所以在非对应的编辑器中将不能正常显示。Git 可以通过设置自动转换这些换行符。使用 Windows 环境的各位，请选择推荐的“Checkout Windows-style, commit Unix-style line endings”选项。换行符在签出时会自动转换为 CRLF，在提交时则会自动转换为LF。其他均可按默认选择进行安装。

# Git Bash初始设置

设置用户名与邮箱，提高命令行输出命令可读性

git config --global user.name "github账号"

git config --global user.email "github邮箱"

git config --global color.ui auto

# 本地Git客户端通过SSH连接到github

$ ssh-keygen -t rsa -C "your\_email@example.com"

Generating public/private rsa key pair.

Enter file in which to save the key

(/Users/your\_user\_directory/.ssh/id\_rsa): 按回车键

Enter passphrase (empty for no passphrase): 输入密码

Enter same passphrase again: 再次输入密码

其中会生成SSH连接的私钥与公钥，id\_rsa 文件是私有密钥，id\_rsa.pub 是公开密钥。

在 GitHub 中添加公开密钥，今后就可以用私有密钥进行认证了。点击右上角的账户设定按钮（Account Settings），选择 SSH Keys 菜单。点击 Add SSH Key 之后，会出现如图 3.2 的输入框。在 Title 中输入适当的密钥名称。Key 部分请粘贴 id\_rsa.pub 文件里的内容（所有内容）。id\_rsa.pub的内容可以用如下方法查看。

$ cat ~/.ssh/id\_rsa.pub

ssh-rsa 公开密钥的内容 [your\_email@example.com](mailto:your_email@example.com)

私钥连接github

ssh -T [git@github.com](mailto:git@github.com)

Hi xuejing6! You've successfully authenticated, but GitHub does not provide shell access.

在github上创建一个仓库，通过git clone git仓地址 命令来将远程仓克隆到本地

git clone git@github.com:fsliurujie/test.git --SSH协议

git clone git://github.com/fsliurujie/test.git --GIT协议

git clone https://github.com/fsliurujie/test.git --HTTPS协议

在本地仓添加文件进行编辑，最后同步到远程仓

git status #查看本地仓状态

git add filename #将文件添加到暂存区

git commit -a(同时add) -m logstr #将修改提交到本地仓 -m 添加日志

git log #查看提交日志

git push <远程主机名> <本地分支名>:<远程分支名> #同步到远程仓

如：git push origin main git push origin main:master

# git基本操作

**● git init——初始化仓库**

**● git status——查看仓库的状态**

**● git add——向暂存区中添加文件**

**● git commit——保存仓库的历史记录 -m 参数后跟修改日志信息**

**● git log——查看提交日志**

git log filename 查看单个文件的提交日志

git log -p 查看提交所带来的改动

git log filename -p查看单个文件提交所带来的改动

**● git diff——查看更改前后的差别**

不妨养成这样一个好习惯：在执行 git commit命令之前先执行git diff HEAD命令，查看本次提交与上次提交之间有什么差别，等确认完毕后再进行提交。

**● git branch——显示分支一览表**

-a命令查看当前分支的相关信息。添加 -a参数可以同时显示本地仓库和远程仓库的分支信息。

**● git checkout  - b——创建、切换分支**

如：git checkout -b BranchA

git branch BranchA

git checkout BranchA

git branch 查看所在分支

**● git merge——合并分支**

首先切换到 master 分支 git checkout master

合并分支 git merge --no-ff BranchA

**● git log  -- graph——以图表形式查看分支**

**● git reset——回溯历史版本**

这里回溯到创建BranchA之前为例：

git checkout -b BranchB--创建分支B

要让仓库的 HEAD、暂存区、当前工作树回溯到指定状态，需要用到 git rest --hard命令。只要提供目标时间点的哈希值

$ git reset --hard 982211406061dfc862183cba0dc04ab5e9158afb

HEAD is now at 9822114 test

**● git reflog——查看当前仓库的操作日志**

即便开发者错误执行了 Git 操作，基本也都可以利用 git reflog命令恢复到原先的状态

**● git commit  -- amend——修改提交信息**

**● git rebase  - i——压缩历史**

git rebase -i HEAD~2 2为需将commit次数合并成一个

git rebase --abort 会放弃合并，回到rebase操作之前的状态，之前的提交的不会丢弃；

git rebase --skip 引起冲突的commits会被丢弃

git rebase --continue 合并冲突，结合"git add 文件"命令一起用与修复冲突

**● git remote add——添加远程仓库**

git remote add origin [git@github.com:github-book/git-tutorial.git](mailto:git@github.com:github-book/git-tutorial.git)

按照上述格式执行 git remote add命令之后，git 会自动将git@github.com:github-book/git-tutorial.git远程仓库的名称设置为 origin（标识符）。

**● git push——推送至远程仓库**

如：$ git push -u origin master

推送至 master 以外的分支 ：

先创建新分支并推送到远程仓库：git checkout -b BranchA git push -u origin BranchA

推送远程仓库的 GitHub 页面就可以查看到了

删除远程仓分支：git push origin --delete BranchA

git checkout -b BranchA origin/BranchA

-b 参数的后面是本地仓库中新建分支的名称。为了便于理解，我们仍将其命名为 feature-D，让它与远程仓库的对应分支保持同名。新建分支名称后面是获取来源的分支名称。例子中指定了 origin/feature-D，就是说以名为 origin 的仓库（这里指 GitHub 端的仓库）的 feature-D 分支为来源，在本地仓库中创建 feature-D 分支

**● git pull——获取最新的远程仓库分支**