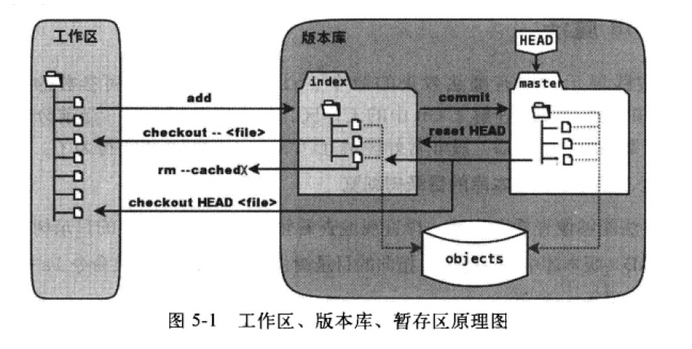
**Git操作记录**

恩首先说明下这只是为了避免自己忘记Git操作的时候重新跑去看一遍教程写的，只是作为记录，要教程的跑去下面的三个链接处。

首先要熟悉仓库结构（先忽略分支问题来说明吧） —— 本地的：工作区 暂存区 版本库 与Github交互的 —— 远程库 本地Git仓库

项目人Github仓库 →(Fork) 自己的Github仓库/项目仓库 →(git clone) → 本地git仓库

自己在github的项目克隆仓库修改后想要项目人接收你的代码，可发起pull request。但注意它只接收能被直接Merge的Pull Request // 不然项目负责人改冲突什么的都得累坏了啊

**配置**

apt-get install git 安装git

git init 把当前目录变为Git管理的本地仓库

git config --global user.email “[171767313@qq.com](mailto:171767313@qq.com)”

git config –global user.name “171767313” // --global 表示对本电脑所有git生效

**基本操作**

git add test/ 把工作区中的test添加到暂存区

git commit –m “explaination” 告诉git已有新修改的文件提交到仓库，并附加说明(恩，准确来说是扔到Objects那里去了

git status 当前仓库状态

git diff 查看working tree与index file的差别

git log 查看提交日志，包含commit ID, Author, Date.

git reset --hard HEAD^^^ 把暂存区的文件退回到前x个版本，其中x为^的数量，也可

git reset --hard commit\_ID 此命令也可用于回复到新版本的文件

git reflog 查看所有分支的所有操作记录 ，包括commit和reset的操作

git checkout -- test.c 取消test.c的修改，退回到暂存区时候

git reset HEAD test.c 把暂存区里test.c的修改撤销，重新扔回工作区

git rm test.c 从版本库位置删除test.c

**身份验证**

接着说到一个验证的问题。恩，你和Github想相互验证对方的真实身份该怎么做的呢？这个和单纯的tell me who you are不同，就好像说我部长想在我电脑上推代码上他的Github那里，Github得验证他身份，不然要是那个是我手贱操作的，他就得哭了。Github则是用SSH Key来识别是不是你在用自己的电脑工作——恩所以我部长说干完活整理完自己想弄的资料，一定要删，不然会祸害到他= =

本地Git仓库和Github仓库是通过SSH加密的。以下则是廖雪峰博客上的内容：

1. 创建SSH Key。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有id\_rsa和id\_rsa.pub这两个文件(前者为私钥，后者公钥)，如果已经有了，可以直接跳到下一步。如果没有，打开Shell，创建SSH Key

$ ssh-keygen –t rsa –C [youremail@example.com](mailto:youremail@example.com)

2、接着登录Github，打开”Account settings”, “SSH Keys”页面，点击”Add SSH Key”，填上任意Title，在Key文本框里粘贴”id\_rsa.pub”文件的内容

**分支问题**

Github算是一个用服务器专门的为Git提供保存、管理代码的地方(恩…可以参考Pypi，个人觉得部分相似)。如果需要用Github来托管一个新项目的代码，则需要在上面创建一个新的仓库。据说还能自己局域网下搭一个玩，这个打算学服务器的时候再自己拿个硬件平台来玩玩吧，不扯这个了，况且一般情况下也用不到。

下面是在Github上建立远程库，也是廖雪峰博客上抄过来的内容 ——

Github右上角Create a new repo，填入Repository name ，一路默认OK~

目前，在GitHub上的这个learngit (他的example\_name) 仓库还是空的，GitHub告诉我们，可以从这个仓库克隆出新的仓库，也可以把一个已有的本地仓库与之关联，然后，把本地仓库的内容推送到GitHub仓库。

现在，我们根据GitHub的提示，在本地的learngit仓库下运行命令：

$ git remote add origin git@github.com:michaelliao/learngit.git

请千万注意，把上面的michaelliao替换成你自己的GitHub账户名，否则，你在本地关联的就是我的远程库，关联没有问题，但是你以后推送是推不上去的，因为你的SSH Key公钥不在我的账户列表中。

下一步，就可以把本地库的所有内容推送到远程库上：

$ git push -u origin master

由于远程库是空的，我们第一次推送master分支时，加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。

$ git branch dev // 创立分支dev

$ git checkout dev // 切换至dev分支

$ git branch // 查看所有分支。其中当前分支前会有 \* 号

\*dev

master

$ git branch -d dev // 删除dev分支

$ git merge dev 将dev分支的内容整合到当前分支上 // 如有更新内容上的冲突，会提醒且显示失败，需要手动解决冲动后再合并

$ git log --graph 可以看到合并分支图

附： git remote add <host\_name> <website> 添加远程主机。相应的也有

git remote rm <host\_name> 删除远程主机

**分支管理的问题** —— 恩我上学期参与一个微信公众号开发的时候就因为这个被说了一次，当时我没建立自己的分支直接推上master去了，影响了工作。罪有应得啊= =。下面是阮一峰博客上的：

git fetch <remote\_host\_name> <remote\_branch\_name> 将某主机某分支(此项不写则视为取全部分支)更新整理到本地

git pull <remote\_host\_name> <remote\_branch\_name>:<local\_branch\_name> 取回远程主机某个分支的更新，再与本地的指定分支合并。其中最后一项可不写，此时则视为远程分支与当前分支合并

如果远程主机删除了某个分支，默认情况下，git pull 不会在拉取远程分支的时候，删除对应的本地分支。这是为了防止，由于其他人操作了远程主机，导致git pull不知不觉删除了本地分支。

但是，你可以改变这个行为，加上参数 -p 就会在本地删除远程已经删除的分支。如$git pull –p

git push <remote\_host\_name> <local\_branch\_name>:<remote\_branch\_name>将本地分支的更新推送到远程主机

如果省略远程分支名，则表示将本地分支推送与之存在"追踪关系"的远程分支（通常两者同名），如果该远程分支不存在，则会被新建。

没错，我当初就是直接写了master……明明自己的只是写出了框架，还有一些页面细节等着别人修改来着的= =啊我错了！

**资料参考** （教程的真的去看他们的吧，我只是简单记录）

<http://blog.chinaunix.net/uid-10415985-id-4112861.html> [**undeador**](http://blog.chinaunix.net/uid/10415985.html)博客

<http://www.ruanyifeng.com/blog/2014/06/git_remote.html> 阮一峰的博客

<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000> 廖雪峰的git教程