Git指令

Gl9N7UxuM0t+YuIi/EdjlBsSjD4L/HKmaD3pTvipeMs

$ git config --global user.name "Your Name"

$ git config --global user.email "email@example.com"

注意git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。

$ pwd指令用于显示当前目录；

**创建版本库**

$ git init 把目录变成git可以管理的仓库；

$ ls -ah 用于显示隐藏目录；

$ git clone 项目地址；将项目从git hub 上克隆下来；

**添加及提交**

$ git add <file>用于把文件提交到暂存区；后面加“.”表示把该目录下所有文件都提交；

$ git comit -m "" 把暂存区的内容提交到当前分支；引号内输入备注内容（修改了什么东西或其他）；

可以多次 add 然后一次 comit；

**查询状态日志以及不同**

$ git status 命令可以让我们时刻掌握仓库当前的状态；

$ git diff <file>查看文件的不同之处；

$ git log查看版本库日志，可以试试加上--pretty=oneline参数；

**版本的回退及其他**

$ git reset --hard HEAD^版本回退，几个^就表示回退到当前版本上几个版本；也可以写成HEAD~数字（1,2,3,4,5等等）；

$ git reset --hard +版本id 表示直接转到某个版本，可用用git reflog来查看版本号；

$ git reflog 用来记录每一次命令；

$ git diff HEAD -- <file>查看工作区和版本库里面最新版本的区别；

**撤销修改：**

$ git checkout -- file意思就是，把文件在工作区的修改全部撤销，这里有两种情况：

一种是file自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

一种是file已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

总之，就是让这个文件回到最近一次git commit或git add时的状态。

git checkout -- file命令中的--很重要，没有--，就变成了“切换到另一个分支”的命令。$ git reset HEAD file可以把暂存区的修改撤销掉（unstage），重新放回工作区。

**场景**

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD file，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，版本回退，不过前提是没有推送到远程库。

**删除文件**

$ rm file 删除文件

$ git rm file 删除文件

$ git checkout -- file误删可用这个回退；

**远程仓库**

$ git remote add origin git@github.com:wugaopan19(用户名)/testgit(项目).git 本地库和远程库关联

$ git push (第一次加该参数-u，以后可省略) origin master ，实际上是把当前分支master推送到远程；

$ ssh-keygen -t rsa -C "826232750@qq.com" ,一路回车

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有id\_rsa和id\_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，第2步：登陆GitHub，打开“Account settings”，“SSH Keys”页面：

然后，点“Add SSH Key”，填上任意Title，在Key文本框里粘贴id\_rsa.pub文件的内容

**分支相关**

**创建分支**

$ git checkout -b dev

git checkout命令加上-b参数表示创建并切换，相当于以下两条命令：

$ git branch dev创建分支

$ git checkout dev切换分支

$ git branch 查看分支

git branch命令会列出所有分支，当前分支前面会标一个\*号

**合并分支**

$ git merge dev

$ git branch -d dev删除分支

解决冲突