**Git命令学习**

**基本操作**

工作区🡺缓冲区🡺主分支==🡺==推送远程github

Git add readme.txt 添加工作区文件到缓冲区

Git commit -m ‘提交readme.txt’ 把缓存区文件提交到分支

Git status 查看文件状态

Git log 查看历史记录

git log --pretty=oneline 查看少部分日志信息

Git diff readme.txt 比较当前修改了哪些内容

**版本回退**

Git reset --hard HEAD^ 回退到上一个版本

Git reset --hard HEAD^^ 回退到上两个版本

Git reset --hard HEAD~100 回退到前100个版本

回退错了怎么恢复呢？

先使用git reflog查看日志找到想恢复的版本号” cf06077”

再使用git reset --hard cf06077恢复到这个版本

**Git撤销修改和删除文件操作**

readme.txt自动修改后，还没有放到暂存区，使用 撤销修改就回到和版本库一模一样 的状态。

Git checkout -- readme.txt 表示丢弃工作区修改(有点类似SVN的撤销)

另外一种是readme.txt已经放入暂存区了，接着又作了修改，撤销修改就回到添加暂存 区后的状态

Git checkout -- readme.txt 撤销修改就回到添加暂存区后的状态

**远程连接和推送**

使本地项目与远程项目相连接

Git remote add origin <https://github.com/zhouwuquan/GitTest.git>

推送本地代码到远程github服务器(第一次加 -u 之后不需要加)

git push -u origin master

**如何从远程库克隆**

Git clone <https://github.com/zhouwuquan/GitTest.git>

**创建与合并分支**

Git branch dev 创建dev分支

Git checkout -b dev 创建并切换到dev分支

Git branch 查看当前分支

Git checkout master 切换分支

Git merge dev 合并dev分支到当前分支

Git branch -d dev 删除dev分支

**如何解决冲突？**

合并时候有冲突，解决冲突后直接提交

**分支管理策略**

创建一个dev分支。

Git checkout -b dev

修改readme.txt内容。

添加到暂存区。

Git add readme.txt

添加到dev分支

Git commit -m ‘dev添加’

切换回主分支(master)。

Git checkout master

合并dev分支，使用命令 git merge –no-ff  -m “注释” dev

Git merge --no-ff -m ‘merge dev branch’ dev

删除分支

Git branch -d dev

查看历史记录

Git log --graph --pretty=oneline --abbrev-commit

**bug分支(保留现场)**

当我们正在dev开发是，突然来了个bug很紧急，怎么办？手里的活不干了？

保留现场可以帮助您

Git stash 把当前的工作现场隐藏起来

Git status 查看当前状态

git checkout master 切换到主分支，下面去解决bug

Git checkout -b issue-404 切换并创建一个解决bug的分支issue-404分支

Git add readme.txt 解决bug添加文件至缓存区

Git commit -m ‘fix bug 404’ 解决bug并提交

git checkout master 切换到主分支

git merge --no-ff -m ‘merge fix bug 404’ 合并bug分支上的提交到主分支

git branch -d issue-404 删除issue-404分支

git checkout dev 切换到dev分支

git status 查看状态

git stash list 查看保留的工作现场

恢复之前的dev工作现场

方法一：git stash apply 恢复工作现场

Git stash drop 删除stash内容

方法二:git stash pop 恢复工作现场并删除stash内容

**多人协作。**

当你从远程拉代码的时候，实际上是把本地master和远程的master分支对应起来，并且远程库的默认名称是origin

Git remote 查看远程库信息

Git remote -v 查看远程库详细信息

推送分支

推送分支是把分支所有的本地提交到远程，推送时，要指定本地分支，这样，git就会把改分支推送到远程对应的分支上

Git push origin master 推送master到远程master分支

Git push origin dev 推送dev分支到远程dev分支

一般情况下，哪些分支需要推送？

1. master分支是主分支，因此要时刻与远程同步
2. 一些修复的bug分支不需要推送到远程去，可以先合并到主分支，然后把主分支推送到远程去。

**抓取分支**

多人协作时，大家都会往master分支上推送各自的修改，现在我们可以模拟另外一个同事，可以在另一台电脑上(注意把SSH key添加到github上)或者同一台电脑上另外目录克隆，新建一个名称testgit2

先pull下来，看下冲突，手动解决冲突再提交。

CSDN博客

https://blog.csdn.net/youzhouliu/article/details/78952453