# GIT学习

## 一．GIT 简介与时光穿梭机

1. Git bash打开git工具
2. git add：准备把文件添加到仓库
3. git commit -m “”：把文件提交仓库 -m后为本次提交的说明
4. 总结：修改/添加文件总共两步，add /commi
5. git status：命令可以让我们时刻掌握仓库当前的状态
6. git diff：让我们知道文件具体修改的内容
7. git log：查看历史提交记录 加上—pretty=oneline ：查看简单的历史提交记录
8. git reset—hard HEAD^：回退到上一个版本 HEAD^^：回退到上上个版本，HEAD~100，回退到上100个版本，hard后跟commit id参数，回退到指定版本
9. git reflog：记录你的每一次命令

现在总结一下：

* HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit\_id。
* 穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。
* 要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

1. Git checkout –“文件名”：把工作区的这个文件的修改全部撤销：：add到暂存区的文件的修改不能被撤销
2. 工作区：自己电脑能看到的目录，
3. 版本库：工作区有个隐藏的目录.git，就会说git的版本库
4. 暂存区(stage)：git add 就是把文件放到暂存区，然后git commi把文件提交到分支

又到了小结时间。

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，参考[版本回退](https://www.liaoxuefeng.com/wiki/896043488029600/897013573512192)一节，不过前提是没有推送到远程库。

## 二，远程仓库及github的使用

1. github建立自己的分支：

（1） 与远程仓库建立连接：$ git remote add origin <https://github.com/dadalingexulin/Git-study.git>.

（2）把本地分支推送远程分支：$ git push -u origin master

1. git branch:查看分支 git branch 分支名：创建分支 git branch -d 分支名 : 删除分支
2. git checkout 分支名：切换分支 git checkout -b 分支名：创建并切换分支
3. git merge 分支名：合并指定分支到当前分支
4. BUG分支：

1.修复bug时，我们会通过创建新的bug分支进行修复，然后合并，最后删除；

2.当手头工作没有完成时，先把工作现场git stash一下，然后去修复bug，修复后，再git stash pop(恢复并删除stash)，回到工作现场；git stash apply:恢复现场，git stash drop删除现场，git stash list：查看stash 记录

3.在master分支上修复的bug，想要合并到当前dev分支，可以用git cherry-pick <commit>命令，**把bug提交的修改**“复制”到当前分支，避免重复劳动，注意：是把提交的修改合并到当前分支

### 小结

* 查看远程库信息，使用git remote -v；
* 本地新建的分支如果不推送到远程，对其他人就是不可见的；
* 从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；
* 在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；
* 建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；
* 从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。
* 创建标签
* 命令git tag <tagname>用于新建一个标签，默认为HEAD，也可以指定一个commit id；
* 命令git tag -a <tagname> -m "blablabla..."可以指定标签信息；
* 命令git tag可以查看所有标签。