1、Git 查看未add的修改：

$ git diff <file>

* 1. add commit 合并

git commit –m “\*\*\*” <file1> <file2> <filen>

2、Git 查看未add的修改：

$ git log -p <file>

3.1、git 撤销在add 之前的修改：

$ git checkout -- <file>

3.2、git 撤销在add 之后的修改：

$ git checkout <file>

4、验证一下会出现你添加成功的远程仓库：

$ git remote –v

5、远程仓库连接本地仓库：

$ git remote add origin(代称) [git@github.com:Lolita-Sian/Lolita.git（SSH](mailto:git@github.com:Lolita-Sian/Lolita.git（SSH)）

6、不想连接这个远程仓库了，只需要输入git remote remove +代称。

比如我刚才添加的远程仓库代称是origin，那我就要写：$ git remote remove origin

7、**进行内容同步：**

$ git push -u 代称 master

8、这时候你就可以把你本地仓库的东西推到远程仓库了，并且你commit的信息也会同步过去。 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

9、在以后的推送或者拉取时就可以简化命令，只输入（如下）就OK：

$ git push origin master

10、查看当前版本信息（也就是最近一个commit）

$ git log

11、显示某个文件的版本历史，包括文件改名

$ git log --follow 文件名.文件类型

$ git whatchanged 文件名.文件类型

12、显示所有提交过的用户，按提交次数排序

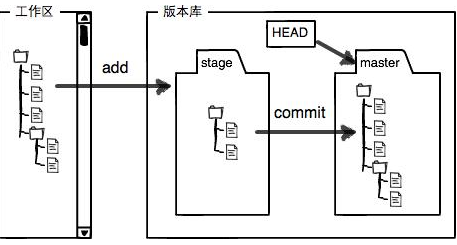
$ git shortlog -sn

13、显示指定文件是什么人在什么时间修改过

$ git blame –L 起始行，终止行 文件名.文件类型

14、Git add之后的内容已放到缓存区，如果想add的文件回到工作区：这里commit之后就不行了

$ git reset HEAD <file>



针对add之后的文件的撤销：

Git reset HEAD file

针对add之后的文件的撤销，并将修改去掉：

Git reset HEAD file

Git checkout -- file

15、创建分支：

$ git branch dev(分支名)

16、切换分支：

$ git checkout dev

17、创建并切换分支：

$ git checkout –b dev

18、创建多个分支后，对文件的修改没有提交到缓存区（git add），任何分支都能看到修改，只有某一个分支提交并commit，其他分支才看不见

19、合并A分支到B分支

切换到B分支，$ git merge A

20、推送本地分支A到远程分支（远程分支并没有建立）

$ git push –-set-upstream ssh:（或者远程仓库代称） A(名字可以换成别的，最好和本地一样)

这样远程就建立了一个A分支，并且本地分支A和远程分支A已经建立关联，下次修改可直接：git pull & git push提交修改。

或者使用：$ git push -u ssh:（或者远程仓库代称origin） A

21、获取远程分支的文件：

$ git pull ssh:（或者远程仓库代称origin）BranchName

22、删除分支：

$ git push ssh:（或者远程仓库代称origin）-d BranchName

删除权限不够：

hook declined FATAL: W refs/heads DENIED by fallthru error

这时远程仓库的分支删除了，但是本地查看还有是因为本地有remote-tracking branch

23、删除本地的远程分支（remote-tracking）：

$ git branch -r -d origin/BranchName(这个名字可以在本地查看获得git branch -r)

-d 有修改会提示

-D 强制删除

24、列出本地分支：

$ git branch

25、查看全部分支：

$ git branch -a

26、查看本地分支：

$ git branch -l

27、查看远程分支：

$ git branch -r

28、删除本地分支：

$ git branch -D BranchName

其中-D也可以是--delete，如：

29、查看本地分支及追踪的分支

$ git branch -vv

30、本地master分支提交到远程其他不是master分支（branch2）：

$ git push origin master:branch2

这里没有建立追踪关系：

这是只打git push命令，不会push到branch2而是本地master tracking的远程master.

另一种方法：切到branch2 merge master,然后再从本地branch2直接提交。

31、撤销本地分支与远程分支的映射关系

$ git branch --unset-upstream

32、将当前分支跟踪origin/AAA：

$ git branch -u origin/AAA

branch2没有追踪到远程分支：git branch --unset-upstream



branch2追踪到远程dev分支：git branch -u origin/dev



branch2追踪到远程branch2分支：git push --set-upstream origin branch2



33、本地分支A推送到远程没有的分支B：

$ git push origin A:B

34、查看本地跟踪的远程分支：

$ git config -l | grep 'branch\.分支'

35、git图形化查看分支：

$ gitk &

36、git 版本回退

查看：

$ git log 查看各种版本

$ git log --pretty=oneline 一行查看

切换：

$ git reset --hard HEAD^ 回到上一个版本

$ git reset --hard 1094a 回退到指定版本号上

回到最新：

一旦回退到某个以前版本上，便不能通过git log查看回退版本之后的新版本

通过$ git reflog查看

**GitHub操作：**

1、在本地建好一个用于作为克隆的本地仓库的文件夹，在文件夹内右键-Git Bash Here。或者直接在Git Bash中进入。然后在Git Bash中输入

$ git clone 加上你刚才复制的那一块

例如，我要输入$ git clone git@github.com:Lolita-Sian/000.git

2、在你刚才init或者Clone的过程中，Git会自动在本地分支与远程分支之间，建立一种追踪关系(tracking)。比如，在git clone的时候，所有本地分支默认与远程主机的同名分支，建立追踪关系，也就是说，本地的master分支自动连接到远程的origin/master分支。

那么建立远程关系之后，如果你要取回远程文件，你只需要输入“ $ git pull origin”。

$ git pull origin

先执行命令 git log,这个老重要咯。。几乎决定生死。。。。

按键盘q键，Enter退出

一、建立追踪关系的三种方式

1.1 手动建立追踪关系

$ git branch --set-upstream-to=<远程主机名>/<远程分支名> <本地分支名>

1 U网试了没起作用！

1.2 push时建立追踪关系

$ git push -u <远程主机名> <本地分支名>

1

加上-u参数，这样push时，本地指定分支就和远程主机的同名分支建立追踪关系。

1.3 新建分支时建立跟踪关系

$ git checkout -b <本地分支名> <远程主机名>/<远程分支名>

1

新分支指针指向 <远程主机名>/<远程分支名> 所指的位置。具体位置可用 git log --oneline --graph 查看。

上传文件到SVN:

桌面右击选Tortoise SVN –> Repo-browser –> 选定位置 –>右键：Add File

更新SVN文件：

SVN Checkout —> 更改后 –> SVN updata –> SVN commit

Git 避免建立多个分支操作：修改代码在master（git checkout master）上，可git pull拉最新，以修改和检查。代码修改完成，在分支上将master分支内容merge进来，然后再push到远程分支上。再交给其他人merge远程分支到远程master上。

**1、在master改好东西后，再新建分支会同步过去。**

**2、Pull新版本的时候，不要将之前版本改动的文件直接复制到新版本中，涉及到公共文件的改动会有冲突！**

参考：

1. git bash使用操作：<https://blog.csdn.net/qq_36667170/article/details/79085301>
2. 查看/删除分支：<https://www.cnblogs.com/linx/p/11328741.html>
3. 本地与远程分支关系：<https://www.jianshu.com/p/526eb3eec83e>
4. 本地与远程分支关系：<https://www.cnblogs.com/mafeng/p/10137244.html>
5. 本地与远程分支关系：<https://blog.csdn.net/qq_42780289/article/details/97762596>