1.1 建立一个目录（最好是空目录），然后进行初始化。Git init;

1.2 添加文件。Git add 文件+后缀名；

1.3 添加文件说明。Git commit -m”文件说明”（可以add许多文件之后使用commit进行说明）；

2.1 git status查看仓库的当前状态，看看是否被修改；

2.2 git diff当仓库发生修改之后，查看修改了什么内容，修改了之后需要git add，然后进行git commit-m之后结束，这个过程随时可以用git status查看状态；

2.3 git log查看修改的版本，内容比较详细；git log –pretty=oneline可以查看简单内容（使用twoline不可以实现）；

2.4 git reset –-hard HEAD^ 回退到上一个版本，hard的参数意义在后续；当回到上一个版本之后想要再返回最新那个版本（前提是没有关闭命令行窗口），把HEAD^更换成之前git log所看到的最新版的版本号就可以实现（随便打版本号前几位就可以）；如果已经关掉了找不到之前最新版本的id，可以使用git reflog来查看（这个命令记录了自己的每一次命令），在这里可以找到之前的id；

2.5 git add是先将文件提交到缓存区，如果修改完不使用git add指令，之后无法提交到分支上去。然后再使用git commit-m将文件提交到master分支

2.6 撤销修改的相关操作

2.7 删除文件的相关操作

2.8 远程库提交的问题。首先初始化一个远程库(init)，add，commit；然后使用指令git remote add origin （需要关联的github位置，第一次关联即可），最后git push -u origin master(第一次提交的需要-u，后续的提交则不需要，origin是远程库的名字)，没有网络的时候可以先把修改后的文件commit一下，然后等到有网的时候push

2.9 从远程库克隆文件，先找好一个位置（clone的库存储在这个位置），然后 git clone +地址；克隆的时候可以选择ssh模式或者http模式，前者速度快一些，但是某些公司只用http（http模式push的时候需要输入用户名和密码，比较麻烦）

3.0（忘记顺序了，但是这个对标分支管理的第一个）分支的问题大概就是增加一个分支（假设分支命名为dev）的指针，然后head开始指向dev分支，master分支不做任何变动，直到最后分支完成工作之后将master和dev进行合并

3.1 创建并切换新的分支git checkout -b dev等效于指令 git branch dev（创建新的分支） + git checkout dev（切换到新的分支），随后可以用 git checkout dev/master进行随便切换。切换到master分支之后，使用git merge dev实现分支的融合，最后git branch -d dev删除dev分支。其中切换分支使用git switch dev/master更合理，这一节Liao有整理常用的几个指令

3.2 cat 文件名+后缀 查看该文件的内容。当master和分支都对一个文件进行了修改的时候，不能实现简单粗暴的融合分支，此时可以查看区别，而这个区别也会显示在该文件中，可以对该文件进行修改，修改之后master和分支都会指向该新的文件。总而言之就是当master和分支都修改了内容要进行融合的时候，需要先解决冲突，然后进行融合。git log –graph可以查看分支合并图

3.3 上面中的那种ff融合（简单粗暴融合）合并之后看不出来曾经做过合并，不是很好，建议添加参数 git merge–-no-ff -m”提交注释” 使用普通模式合并

3.4 git stash的使用（一般我们是在分支上干活，假设分支dev），当分支上的活没干完，还不可以提交，但这时需要去master分支修复一个bug，此时可以先使用该命令将目前的修改工作暂存不提交，然后切换到master分支，再创建修建bug分支进行bug处理。处理完bug之后删除bug分支后切回dev（此时由于bug是在master上进行处理的，所以bug也会存在于dev，测试结果也是如此，此时还需要对dev进行bug处理），这个时候在上面master融合处理bug分支的时候就不可以使用ff模式，否则dev分支想要复制该操作会出错，这个操作有点问题，有待改善

3.5 Feature分支：在分支dev干活，再创建分支，然后创建后的分支活干完了可能要融合进dev，可能不融合进

3.6 多人协作的时候可以把自己的分支推送到master或者远端origin上去 git push origin dev/master 将master或者dev上的东西推给github上去。推送之前git remote -v查看远程（github）的地址，查看是否可以抓取（fetch）或者推送（push）。

3.7 多人合作里，clone下origin库，然后创建分支dev进行提交，之后会自动在github的repository里创建一个dev分支，还有一个当别人在github上更新东西之后，此时自己需要提交自己的任务，先将别人的修改pull下来（此过程可能会出现本地分支与远程分支没有链接的问题），然后自己修改合并后进行提交 。在本地创建和远程分支对应的分支git checkout -b branch-name origin/branch-name

建立本地分支和远程分支的关联 git branch –-set-upstream branch-name origin/branch-name

3.8 Windows下的文本编辑。已收藏