**取得Git仓库**

1.初始化一个版本仓库

git init

执行上面的命令会在当前的目录的创建一个.git文件夹，这样就创建了一个仓库

2.clone远程版本库

git clone git@IP-server:xxx.git

执行上面的命令会从IP地址为IP-server的服务器上把名字为xxx.git的仓库克隆到本地，

本地的内容跟服务器上的内容完全一样，相当于从服务器上复制了一份。

3.添加远程版本库origin，语法为 git remote add [shortname] [url]

git remote add origin git@IP-server:xxx.git

[执行上面的命令相当于给git@IP-server：xxx.git](mailto:执行上面的命令相当于给git@IP-server：xxx.git)起了一个别名，这样以后就可以直接使用origin别名来代替服务器的仓库地址了。

4.查看已添加的远程仓库

git remote -v

**提交你的修改**

1.添加当前工作区中所有修改的文件到暂存区

git add .

2.提交你的修改

git commit –m ‘你的注释’

3.推送你的更新到远程服务器,语法为 git push [远程名] [本地分支]:[远程分支]

git push origin master

上面的命令是git push origin master:master的缩写，其中origin是我们利用git remote add origin ‘仓库地址’ 命令添加的仓库地址的别名。

4.查看文件状态

git status

执行上面的命令可以查看文件的状态，如果你现在不知道修改了哪些文件，或者你想确认下是否已经利用git add . 把所有修改的文件放到暂存区，这时候就可以利用git status来进行查看。

5.跟踪新文件

git add readme.txt

如果你新建了一个新的文件readme.txt，可以利用git add单独执行。

6.从当前跟踪列表移除文件，并完全删除

git rm readme.txt

7.仅在暂存区删除，保留文件在当前目录，不再跟踪

git rm –cached readme.txt

8.重命名文件

git mv reademe.txt readme

9.查看提交的历史记录

git log

10.修改最后一次提交注释，利用--amend参数

git commit --amend

11.假设你已经使用git add .，将修改过的文件a、b加到暂存区

现在你只想提交a文件，不想提交b文件，应该这样

git reset HEAD b

12.取消对文件的修改

git checkout -- readme.txt

**基本的分支管理**

1. 查看当前的分支

git branch

2.创建一个新分支

git branch new

3.切换工作目录到指定分支

git chekcout new

4.将上面的命令合在一起，创建new分支并切换到new

git chekcout –b new

5.合并new分支，当前工作目录为master

git merge new

如果要合并某个分支，当前的分支不能是要合并的分支，必须利用git checkout命令切换到其他分支。当merge命令自身无法解决冲突的时候，它会将工作树置于一种特殊的状态，并且给用户提供冲突信息，以期用户可以自己解决这些问题。当然在这个时候，未发生冲突的代码已经被git merge登记在了index file里了。如果你这个时候使用git diff，显示出来的只是发生冲突的代码信息。

在你解决了冲突之前，发生冲突的文件会一直在index file中被标记出来。这个时候，如果你使用git commit提交的话，git会提示：filename needs merge，其中filename是冲突的文件名。

在发生冲突的时候，如果你使用git status命令，那么会显示出发生冲突的具体信息。

在你解决了冲突之后，你可以使用如下步骤来提交：

第一步(如果你需要增加文件)：git add newfile

第二步：git commit -m ‘你的描述’

6.合并完成后，如果没有出现冲突，删除new分支

git branch –d new

如果要删除分支，要保证要被删除的分支已经被其他分支合并，否则会失败。如果不管分支是否被合并，你就是要删除它，可以利用git branch –D new 强制删除new分支。

7.拉取远程仓库的数据，语法为 git fetch [remote-name]

git fetch

8.拉取远程仓库的数据并自动合并

git pull

9.查看远程仓库的信息

git remote show origin

上面命令中的origin是远程仓库的别名

10.建立本地的dev分支追踪远程仓库的develop分支

git checkout –b dev origin/develop

执行上述命令会在本地创建一个新的dev分支，该dev分支跟远程仓库中的develop分支的内容相同，在本地切换到dev分支，以后直接执行git push跟git pull，上传跟拉取的数据就默认为develop分支的内容。