1. 选择一个合适的文件目录：

       mkdir learngit：创建一个空的文件夹

1. git init : 把这个目录改变为Git可以管理的仓库
2. 创建一个txt文件到已经可以管理的仓库中，  
   git add (指定的文件名)，添加指定的文件  
   git add . ：添加所有改变
3. git **commit** -m “(改变的说明)“
4. 修改完查看当前仓库的状态  
   git status
5. git diff (指定文件名)：查看具体文件修改的内容
6. git log：查看最近提交版本信息  
   git log --pretty=oneline ：版本信息输出太多，简化信息
7. git re**set** --hard HEAD^：返回上一个版本
8. git re**set** --hard （需要返回的版本号）：版本号不用写全，前几位就好，系统会帮你补全
9. git reflog：关闭了当前的git管理，找回提交对应的版本号
10. Untracked：提示当前的文件未被添加
11. cat readme.txt : 查看当前文件的内容
12. git reset HEAD (文件名) : 把已经提交到暂存区（add），文件回退到未提交的状态（没有add的状态）
13. git checkout -- （文件名）：丢掉工作区的修改（未add的文件状态）
14. git checkout -- （文件名）：也可以用来恢复被删除的文件
15. git remote add origin （远程仓库地址） : 链接本地仓库到远程仓库
16. git push -u origin master ：推送本地仓库到所有内容到远程仓库
17. git clone (远程仓库) ：下载远程仓库
18. git branch dev : 创建dev分支 （dev分支，可以修改为其他名字）
19. git checkout -b dev: 创建并切换到dev分支
20. git checkout dev ： 切换到dev分支（dev分支，可以修改为其他名字）
21. git branch : 查看当前所处的分支
22. git merge dev : 合并dev分支，到当前的分支，注意目前所处的分支（dev分支，可以修改为其他名字）
23. git branch -d dev : 删除分支（dev分支，可以修改为其他名字，假如工程为空的情况下，无论创建多少分支，默认的都是在maser分支上的，删除分支时，需要切换到其他的分支，才能删除本分支）
24. git branch -v : 查看当前的版本情况
25. git log --graph --pretty=oneline --abbrev-commit : 查看分支合并状态
26. git merge --no-ff -m "merge with no-ff" dev : 使用--no-ff参数，使分支合并时，不会丢掉分支信息，因为本次合并要创建一个新的commit，所以将commit描述添加进去，使用21查看合并分支的情况