1. 安装Git bash
2. 配置：git config –global user.name “Gossip”

git config –global user.emai [decesion@126.com](mailto:decesion@126.com)

mkdir Projects

cd projects

git init 创建版本库

1. Git add test.txt 把文件添加到仓库

Git commit –m ”it is a test” 把文件提交到仓库

Git status 查看仓库状态

Git diff 查看文件内容有什么不同

Git log 查看提交日志

Git log –prety=online 可以得到详细参数，得到详细版本数

Git reset --HEAD^ 回退到上一个版本 ^^上上一个版本

Git reset –hard 3628164 回退到指定版本号，git log可以得到

Git reflog 记录你每一次的Git命令

Git diff HEAD – test.txt 查看工作区与版本库里最新版本的区别

Git checkout -- file 让文件回到最近一次git commit或git add 时的状态

Git rm file 并且git commit 从版本库中删除文件，直接rm也可以删除，但是版本库和工作区就不一致了 ，删除恢复使用git checkout – file

1. 创建SSh Key：

ssh-keyen –t rsa –C “[decesion@126.com](mailto:decesion@126.com)”,然后在用户主目录里找到.ssh目录，里面有id\_rsa（私钥）和id\_rsa.pub（公钥），把id\_rea写入git官网账号配置信息，创建仓库。

1. 建立关联：

git remote add origin [git@github.com:Goossip/Projects.git](mailto:git@github.com:Goossip/Projects.git) 如果报错，先git remote rm origin，再建立关联

推送：

git push -u origin master，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,之后使用：git push origin master

1. 从远程库克隆：

git clone [git@github.com:Goossip/bootstrap.git](mailto:git@github.com:Goossip/bootstrap.git)

1. 创建分支与合并分支：

git checkout -b dev相当于a与b

1. git branch dev
2. git checkout dev 创建dev分支并且切换到dev分支
3. git branch 查看当前分支
4. 在dev工作之后使用add与commit提交
5. git checkout master 返回master分支
6. git merge dev 在当前分支（master）上合并dev分支
7. git branch –d dev 合并分支之后删除dev分支
8. 解决合并冲突：
9. git checkout –b feature1 创建feature1分支并切换到
10. edit file 编辑master分支到文件
11. add and commit 提交
12. git checkout master 返回master分支
13. edit both file 编辑同一个文件
14. add and commit 提交
15. git merge feature1 执行到这一部会报错
16. git status 查看报错内容
17. cat file 查看提交文件并修改成一致文件
18. add and commit 提交
19. git log --graph --pretty=oneline --abbrev-commit 查看分支合并情况
20. git branch -d feature1 删除分支

Git用<<<<<<<，=======，>>>>>>>标记出不同分支的内容

1. 分支管理策略
2. git checkout -b dev 创建dev分支并且切换到dev分支
3. edit file 修改或编辑
4. add and commit 提交
5. git checkout master 切回master分支
6. git merge --no-ff -m "merge with no-ff" dev 合并dev分支，注意--no-ff参数，表示禁用Fast forward, 由于合并要创建一个新的commit，所以加上-m参数，并把commit描述写入.
7. git log 查看分支历史
8. Bug分支
9. git status 查看当前分支
10. git stash 保存当前工作
11. git status
12. git checkout master 切回master分支（要在master分支修复）
13. git checkout -b issue-101 创建issue-101分支并且切换到该分支
14. edit 修复
15. add and commit 提交
16. git checkout master 切回master分支
17. git merge --no-ff -m "merged bug fix 101" issue-101 合并
18. git branch -d issue-101 删除issue-101分支
19. git checkout dev 切回dev分支
20. git status 查看当前分支
21. git stash list 查看备份文件表
22. 恢复备份

1). git stash apply 恢复备份

git stash drop 删除备份

2). git stash pop 恢复并删除

o. git stash list 查看备份文件表

p. git stash apply stash@{0} 恢复指定stash

featuref分支，创建分支添加功能

a. git checkout -b feature-vulcan

b. edit add commit

c. git checkout dev

d. git branch -d feature-vulcan 错误

git branch -D feature-vulcan 正确 删除该分支必须在没有合并前删除

1. 多人合作

git remote 查看远程库

git remote –v 查看远程库详细信息

git push origin master 推送master分支

git push origin dev 推送dev分支

1. 标签管理
2. git branch 查看分支，当前分支带\*号
3. git checkout master 切换分支
4. git tag v1.0 创建标签v1.0,输入命令git tag <name>可以打一个新标签
5. git tag 查看所以标签
6. git log --pretty=oneline --abbrev-commit 查找历史提交commit
7. git tag v0.9 6224937 从上步查到6224937并为它打标签
8. git show v0.9 查看标签v0.9信息，git show <tagname>，查看标签到详细信息
9. 创建带有说明的标签，用-a指定标签名，-m指定说明文字：git tag -a v0.1 -m "version 0.1 released" 3628164

git show v0.1 创建之后查看

1. 通过-s用私钥签名一个标签：

git tag -s v0.2 -m "signed version 0.2 released" fec145a

1. git tag -d v0.1 删除v0.1标签
2. 推送某个标签到远程，使用命令git push origin <tagname>
3. git push origin v1.0 推送v.1.0标签到远程
4. git push origin –tags 一次性推送全部尚未推送到远程的本地标签
5. 标签已经推送到远程：

git tag -d v0.9 删除本地v.9标签

git push origin :refs/tags/v0.9 删除远程v.9标签

1. 克隆

1）. 在别人到项目上Fork

2）. git clone [git@github.com:Goossip/bootstrap.git](mailto:git@github.com:Goossip/bootstrap.git)

码云的使用：

注册，登录，修改，添加ssh，创建库

git remote add origin [git@gitee.com:Goossip/learngit.git](mailto:git@gitee.com:Goossip/learngit.git) 关联远程库 会报远程库已经存在

git remote –v 查看远程库

git remote rm origin 删除github远程库

git remote add origin [git@gitee.com:Goossip/learngit.git](mailto:git@gitee.com:Goossip/learngit.git) 关联

git remote –v 查看远程库

同时关联两个库：

git remote rm origin 删除当前关联到origin

git remote add origin [git@github.com:Goossip/Projects.git](mailto:git@github.com:Goossip/Projects.git) 关联github

git remote add origin1 git@gitee.com:Goossip/learngit.git 关联gitee

git remote –v 查看远程库信息

推送到GitHub，使用命令：

git push github master github是关联名（应该填写origin）

推送到码云，使用命令：

git push gitee master gitee是关联名 （应该填写origin1）

推送命令：

git push 关联名（默认origin） master

git config --global color.ui true 让git文件显示颜色

1. 忽略特殊文件

在Git工作区的根目录下创建一个特殊的.gitignore文件

git rm -r --cached . 清除本地缓存

git add . 提交

git commit -m 'update .gitignore' 提交