初始化 git 库：　git init
提交新版本：　git add .
　　git commit
　　(git commit -m “版本和开发信息” ；该参数直接输入信息，不在跳出 vi 窗口)

介绍提交用户：（未介绍的话为系统登录用户）

git config --global user.name “Your Name”

git config --global user.email “you@example.com”

查看日志信息：　git log
　　git log -p　　//详细信息
　　git show “commit ID”　　//显示相关 commit 编号的详细内容，可输入前 4-6 个编号
　　git show exp　　//显示分支信息
　　git show HEAD　　//显示最近一次 commit 信息
　　git show HEAD^　　//查看 HEAD 的父母的信息
　　git show HEAD^^　　//查看 HEAD 的父母的父母的信息
　　git show HEAD~4　　//查看 HEAD 上溯 4 代的信息
　　git tag V3 5b888　　//以后可以用 V3 来代替复杂的名称(5b888…)

创建分支 exp：　git branch exp
显示当前分支：　git branch　　//标注*为当前所在分支
转移到 exp 分支：git checkout exp　　//切换分支前需将当前分支内容提交
将某一版本创建分支：git branch exp1 V3 //V3 为 commit ID 编号，exp1 为分支名

合并分支：　git merge exp　　//将 exp 分支合并到当前分支，合并后需再次提交

删除分支：　git branch -d exp　　//因为 exp 分支已提交，所以可安全删除此分支
　　git branch -D exp　　//由于分支被证明失败，因此使用-D 来放弃并删除该分支

检查源码改动：　git diff　　//在 git add 之前使用有效
　　git diff --cached　　//在 git add 之后在 git commit 之前有效

检查状态：　git status　　//这个命令在 git commit 之前有效，查看整体改动信息
可以看到提示信息“changed but not updated”，就是说 git 发现你有已经修改了但还未 git add 的内容。
如果 git 提示说“Changes to be committed”，那就是表明 git 发现了你已经 git add 但还未 git commit 的内容。
如果 git 提示说“Untracked files”，那么就是你增加了新文件或者在某个子目录下增加了新文件。

撤销：

git reset --soft V2 //commit 提交了 V1、V2、V3 三次版本，该命令撤销了 V3 的提交日志信息，但是具体开发内容不变。可修改后再次提交 V3

git reset --hard V2 //恢复到 V2 版本，彻底删除 V3 的所有信息，如想保留 V3 的信息，则不使用该命令，而是使用 git branch exp1 V2 命令创建分支

user 合作开发 PM 的项目：

克隆 pm 用户的 hello 目录下的项目到自己 user 的 hello_temp 目录，开发成功后提交，并通知 pm

```
git clone /home/pm/hello  hello_temp
cd hello_temp
…...
git add .
git commit
```

PM 合并 user 的开发内容：

确信 user 的开发内容正确，直接合并：

```
cd /home/pm/hello
git pull /home/user/hello_temp
```

不确信 user 开发内容，检查后合并：

```
git fetch /home/user/hello_temp master:exp1 //提取 user 的开发内容，放到 PM 工作目录下的 exp1 分支中
git whatchanged -p master exp1          //查看 user 改动了哪些内容
git checkout master                     //切换到主分区
git pull . exp1                         //检查正确后，可以用 pull 将 exp1 分支合并
git branch -D exp1                      //如果我检查后很不满意，就可以用-D 来放弃这个分支就可以了
```

user 再次开发 PM 的项目：

```
git pull                        //在 hello_temp 目录下执行 pull 即可
```