**分支管理**

master指向提交，HEAD指向的是当前分支。创建新的分支dev后，HEAD就会指向dev。

合并dev分支就是把master指向dev的当前提交。

git checkout –b dev 创建并切换到dev分支，相当于以下两条指令

git branch dev

git checkout dev

git branch 查看当前分支

git merge dev把dev分支的工作合并到master分支上

git branch –d dev删除分支

git switch –c feature1 使用switch代替checkout

**Bug分支**

每个bug都可以通过一个新的临时分支来修复

git stash 把当前工作现场储藏起来

git stash list查看工作现场

git stash pop 恢复的同时把stash内容也删除

git stash apply stash@{0} 恢复但不删除

git cherry-pick 4c805e2 复制一个特定的提交到当前分支

git branch –D feature 强行删除一个还没有合并的分支

**多人协作**

当你从远程仓库克隆时，实际上Git自动把本地的master分支和远程的master分支对应起来了，并且，远程仓库的默认名称是origin。

git remote 查看远程库的信息

git remote –v 更详细的信息

git push origin dev 推送分支

git checkout –b dev origin/dev 创建远程的origin的dev分支到本地

git branch –set-upstream-to=origin/dev dev指定本地分支与远程origin/dev分支的链接

git rebase 把分叉的提交历史整理成一条直线

**解决冲突**

git merge feature1

手动修改冲突内容，然后使用git add。

git log --graph --pretty=oneline --abbrev-commit 查看分支合并情况

git merge --no-ff -m "merge with no-ff" dev 禁用fast-forward，留下合并日志