

Git 使用规范说明

第一步：新建分支

每次开发新功能，新建一个单独的分支（这方面可以参考《Git 分支管理策略》）。

```
# 获取主干最新代码
$ git checkout master
$ git pull
# 新建一个开发分支myfeature
$ git checkout -b myfeature
```

第二步：提交分支commit

`git add` 命令的`all`参数，表示保存所有变化（包括新建、修改和删除）。从Git 2.0开始，`all`是 `git add` 的默认参数，所以也可以用 `git add .` 代替。

`git status` 命令，用来查看发生变动的文件。

`git commit` 命令的`verbose`参数，会列出 `diff` 的结果。

第三步：撰写提交信息

提交commit时，必须给出完整扼要的提交信息，下面是一个范本。

Present-tense summary under 50 characters

*** More information about commit (under 72 characters).**

*** More information about commit (under 72 characters).**

第一行是不超过50个字的提要，然后空一行，罗列出改动原因、主要变动、以及需要注意的问题。

第四步：与主干同步

分支的开发过程中，与主干保持同步。

```
$ git fetch origin
```

```
$ git rebase origin/master
```

第五步：合并commit

使用git rebase命令合并多个commit

```
$ git rebase -i origin/master
```

或者使用撤销commit后重新建立新的commit

```
$ git reset HEAD~5
```

```
$ git add .
```

```
$ git commit -am "Here's the bug fix that closes  
#28"
```

```
$ git push --force
```

第六步：推送到远程仓库

合并commit后，就可以推送当前分支到远程仓库了。

```
$ git push --force origin myfeature
```

git push命令要加上force参数，因为rebase以后，分支历史改变了，跟远程分支不一定兼容，有可能要强行推送。

第七步：发出Pull Request

发出Pull Request到master分支，然后请求别人进行代码review，确认可以合并到master。