一个非常好的入门网站，http://learngitbranching.js.org/

请坚持练习完上面所有的命令。

http://learngitbranching.js.org/?demo 向你演示所有的命令过程（像幻灯片一样）

**常用的命令**

|  |
| --- |
| **查看当前目录状态**：  $git status |
| **显示当前分支**  $ git branch  \* master  \*表示当前所在的分支。  加上参数-a 可以显示远程分支  $ git branch -a |
| **创建分支（注意，所有的变更都必须在分支中进行，这样，主分支master才不会被占用）**  $git branch 分支名 |
| **切换到分支**  $git checkout分支名 |
| **添加要提交的文件**  $ git add 文件名 |
| **提交到当前分支**  $ git commit -m “提交说明” |
| **合并分支**  $ git merge --no-ff 分支名 |
| **推送至远程**  $ git push -u origin 当前分支:远程分支  参数-u表示自动将origin仓库的master设置为本地仓库的上游（upstream） |
| **查看更新日志**  $ git log --graph  日志中有每次提交的UUID值，这个值可以用于回溯。 |
| **查看操作日志（显示所有操作步骤的命令，以及命令对应的哈希值）**  $ git reflog    画红圈的就是哈希值  如果使用reset命令可以使状态回退到指定的操作之后  $ git reset --hard 244cfa4  本地仓库回退到创建并切换至HR\_lsn01分支。 |
| **删除本地分支**  $ git branch -d分支名 |
| **删除远程分支（冒号后面没有空格）**  $ git push origin :分支名 |
| **重命名本地分支**  $ git branch -m 原分支名 新分支名 |
| **回溯**（回溯的目的是为了创建新的分支）  查看需要回溯到的分支点的UUID  $ git log --graph  本地回溯  $ git reset --hard 分支UUID  *此时再查看文件内容，已经回溯到了以前的版本*  创建并切换至新分支（加参数b表示创建分支）  $ git checkout -b 新分支名  对文件进行编辑（略）  添加到提交列表  $ git add 文件名  提交  $ git commit -m “提交说明” |
| **合并分支**（这是最重要的步骤，执行了它，更新才会生效）  切换到主分支  $ git checkout master  合并  $ git merge --no-ff 要合并的分支名  正常情况下会弹出文本编辑器，编写合并操作的日志，请务必填入合并说明  合并目的：  合并内容：  可能的影响：  保存关闭之后，合并结束。 |
| **修改上一次操作日志**  $ git commit --amend  将弹出上一次操作的日志文本编辑器，修改保存即可。 |