# Git常用命令

Git官方网站http://git-scm.com。

## 1基础命令

### ■配置Git

当安装完 Git 应该做的第一件事就是设置你的用户名称和邮件地址（只需执行一次）：

$ git config --global user.name "Chuanhui Long"

$ git config --global user.email linuxdoc@163.com

如果想要检查你的配置，可以使用git config --list 命令。

配置信息保存在 C:\用户\$USER\.gitconfig 文件。

### ■初始化本地仓库

在项目目录下运行Git Bash（Windows下点右键），执行以下命令初始化本地仓库：

**$ git init**

运行此命令将创建.git隐藏文件。

克隆托管平台上远程仓库（需先初始化本地仓库）：

**$ git clone** https://github.com/libgit2/libgit2 /\*最后一个分量为仓库地址\*/

### ■跟踪新文件

开始跟踪一个文件（或修改过的文件，添加到暂存）：

**$ git add filename**

**$ git add \*.c**

**$git add .** /\*添加所有文件\*/

**$ git add \***  /\*添加所有文件\*/

### ■提交更新

**$ git commit [file1] [file2] ... -m 'initial project version'**  /\*注释\*/

**$git commit --amend** **[file1] [file2] ...**  /\*重做（覆盖）上一次提交\*/

**$git commit -v**  /\*显示所有diff信息\*/

### ■删除文件

**$git rm filename**  /\*从当前工作目录和暂存区删除文件，再提交\*/

**$git rm --cache filename** /\*只删除暂存区中文件，工作目录中保留\*/

### ■移动文件

**$ git mv** file\_from file\_to /\*两个文件在同一目录下，就是对文件重命名\*/

### ■显示提交历史

**$git log**  /\*可添加-p、-2等选项\*/

**$git reflog**

### ■穿回到某个版本

**$ git reset --hard** [commit] /\*commit为git reflog命令显示的版本号，如0a49166\*/

/\*重置当前分支的HEAD为指定commit，同时重置暂存区和工作区，与指定commit一致\*/

### **■**查看状态

**$git status**

**$git status -s**

**$git status --short**

**$git diff** /\*显示暂存区和工作区的差异\*/

**$git diff --cached** /\*显示暂存区和上一个commit的差异\*/

**$git diff --staged** /\*比git diff --cached更好的命令\*/

**$ git log -p [file]** /\*显示指定文件相关的每一次diff\*/

**$ git diff HEAD** /\*显示工作区与当前分支最新commit之间的差异\*/

**$ git diff** [first-branch]...[second-branch] /\*显示两次提交之间的差异\*/

$ git show [commit] /\*显示某次提交的元数据和内容变化\*/

$ git show --name-only [commit] /\*显示某次提交发生变化的文件\*/

$ git show [commit]:[filename] /\*显示某次提交时，某个文件的内容\*/

### ■生成压缩包

**$ git archive** /\*生成一个可供发布的压缩包\*/

## 2分支命令

### ■分支创建

**$ git branch** testing /\*testing为新分支名称\*/

**$ git checkout -b** hotfix /\*创建并同时跳转到hotfix分支\*/

**$ git branch** [branch] [commit] /\*创建新分支并指向指定commit（版本 ）\*/

### ■查看分支

**$git branch -r**  /\*列出所有远程分支\*/

**$git branch -a** /\*列出所有远程和本地分支\*/

**$ git log --oneline** **--decorate**

**$ git log --oneline --decorate --graph --all**

/\*输出提交历史、各个分支的指向以及项目的分支分叉情况\*/

**$ git branch --track** [branch] [remote-branch] /\*新建一个分支，与指定的远程分支建立追踪关系\*/

### ■分支切换

**$ git checkout** testing /\*testing为分支名称\*/

**$ git checkout -**  /\*切换到上一个分支\*/

### ■合并分支

**$ git merge** hotfix /\*将hotfix分支合并到当前分支\*/

**$ git cherry-pick** [commit] /\*选择一个commit，合并进当前分支\*/

### ■删除分支

**$ git branch -d** hotfix /\*删除分支hotfix\*/

$ git push origin --delete [branch-name] /\*删除远程分支\*/

$ git branch -dr [remote/branch]

## 3工作区与暂存区命令

**$ git checkout** [file] /\*恢复暂存区的指定文件到工作区\*/

$ git checkout [commit] [file] /\*恢复某个commit的指定文件到暂存区和工作区\*/

$ git checkout . /\*恢复暂存区的所有文件到工作区\*/

$ git reset [file] /\*重置暂存区的指定文件，与上一次commit保持一致，但工作区不变\*/

$ git reset --hard /\*重置暂存区与工作区，与上一次commit保持一致\*/

$ git reset [commit] /\*重置当前分支的指针为指定commit，同时重置暂存区，但工作区不变\*/

$ git reset --keep [commit] /\*重置当前HEAD为指定commit，但保持暂存区和工作区不变\*/

$ git revert [commit] /\*新建一个commit，用来撤销指定commit\*/

/\*后者的所有变化都将被前者抵消，并且应用到当前分支\*/

$ git stash /\*暂时将未提交的变化移除，稍后再移入\*/

$ git stash pop

## 4标签命令

待续...

## 5远程仓库命令

下面以Gitee为例，简要说明本地仓库与远程仓库的操作命令，需先注册并登陆Gitee帐号。

### ■**生成SSH公钥（Windows）**：

生成SSH公钥用于推送时免登陆。

（1）**$ ssh-keygen -t rsa**  /\*rsa为加密算法\*/

（2）找到用户目录下C:\用户\$USER\.ssh\id\_rsa.pub文件，打开复制公钥。

（3）打开Gitee并登陆，点击“个人设置”“SSH公钥” ，粘贴公钥即可添加。

### ■添加远程仓库

**$git remote add** linuxdoc <https://gitee.com/linuxla/linuxdoc.git> /\*linuxdoc为别名\*/

**$ git remote -v** /\*显示所有远程仓库\*/

$ git remote show [remote] /\*显示某个远程仓库的信息\*/

**$ git remote rm** remote /\*移除远程仓库\*/

### ■远程同步

**$ git fetch** [remote] /\*下载远程仓库的所有变动，remote为远程仓库别名\*/

$ git pull [remote] [branch] /\*取回远程仓库的变化，并与本地分支合并\*/

**$ git push** [remote] [branch] /\*上传本地指定分支到远程仓库\*/

$ git push [remote] --force /\*强行推送当前分支到远程仓库，即使有冲突\*/

**$ git push** [remote] --all /\*推送所有分支到远程仓库\*/

## 5团队协作

待续...