

中心主题

基础篇

- git commit -m "<massage>"
- git branch <branch> 创建分支
- git checkout <branch> 切换分支
- git checkout -b <branch> 创建分支并切换
- git merge <branch> 合并分支到形成新提交
- git rebase <branch> 移动该分支到某分支的新一次提交
- 若目标分支在该分支的下方，则移动该分支到目标分支

高级篇

- git checkout <Hash value> 移动HEAD到某次提交
- cat .git/HEAD 查看HEAD指向
- git symbolic-ref HEAD 查看HEAD指向的引用
- git log 查看提交记录的哈希值
- 相对引用
 - ^向上移动一个提交记录
 - ~<num>向上移动多个提交记录
- git branch -f <branch> HEAD<相对引用> 移动分支到某次提交
- git reset HEAD<相对引用> 撤销本地提交
- git revert HEAD 撤销远程提交
- 恢复上一次提交的内容，并形成新的提交

移动提交记录

- git cherry-pick <Hash value> 移动某次提交到HEAD下，并形成新的提交
- 可以加多个哈希值，会按顺序形成新提交
- git rebase -i HEAD<相对引用> 交互界面移动提交

杂项

- 调整某个之前的提交
 - 先git rebase -i
 - 然后commit --amend
 - 接着git rebase -i
 - 最后把master移到最前端

push & pull

- git clone 克隆远程仓库
- 使用 Git 时，你的远程仓库默认为 origin
- git fetch 从远程仓库下载数据
- 更新远程分支指针(如o/master)
- git fetch之后合并本地与远程仓库
 - git cherry-pick o/master
 - git rebase o/master
 - git merge o/master
- git pull git pull是git fetch和git merge <just-fetched-branch>的缩写
- git push 推到远程
- git pull --rebase 拉取远程新代码，并将当前分支rebase到新拉取的commit上

origin和它的周边

- 远程追踪
 - clone时，默认设定 master 跟踪 o/master
 - git checkout -b <branch> o/master 创建新分支，使其跟踪o/master
 - git checkout -u o/master <branch> 使某分支跟踪o/master
 - 若HEAD在该分支上，可省略<branch>
- git push <remote> <place>
 - git push origin <branch> 将分支推到远程仓库，无论HEAD在哪里
 - git push origin <source>:<destination> 将<source>推到远程仓库的指定位置
- git fetch <remote> <place>
 - git fetch origin <branch> 将远程仓库的某分支拉下来
 - git fetch origin <source>:<destination> 将远程的<source>拉至指定位置
- 古怪的<source>
 - git push origin :<source> 删除远程仓库的<source>
 - git fetch origin :<source> 在本地新建分支<source>
- git pull
 - git pull origin <branch> 拉下远程<branch>后，合并HEAD所在位置和拉下的o/<branch>
 - git pull origin <source>:<destination> 将远程的<source>拉来，若<destination>不存在，则新建分支foo并合并分支