

Git 命令

2016年2月24日 13:48

- git init -----初始化git仓库
- git add file.xx ----- 把文件添加到仓库中
- git commit -m "说明" -----提交到仓库
- git status -----查看工作区状态
- git diff -----查看工作区修改的内容
- git log -----查看历史记录
 - pretty=oneline -----整洁多了
- git reset --hard commit_id -----回退之前commit_id所在的版本
 - HEAD~100 -----回退到前一百个版本
 - HEAD^ -----回退到前一个版本
- git reflog -----查看指令记录
- git diff HEAD -- filename.xxx -----比较工作区和版本库中的内容
- git checkout -- file -----撤销工作区的修改
- git reset HEAD file -----撤销暂存区的修改
- git rm filename.xx -----从版本库中删除该文件
 - o git commit -m "remove filename.xx"
- git push -----将本地库推送到远程
- git remote add origin git@server-name:path/repo-name.git -----关联远程库
- git push -u origin master -----推送master分支的所有内容
 - o 加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令。
- git clone git@github.com:liuwnzh/gitskills.git -----克隆远程库到本地
- git checkout -b dev -----创建并切换到dev分支
 - o -b 表示创建并切换
 - \$ git branch dev -----创建分支dev
 - \$ git checkout dev -----切换到分支dev
- git branch -----查看当前分支
- git checkout master -----切换到master分支
- git merge dev -----合并指定分支dev到当前分支
- git branch -d dev -----参数-d用于删除指定分支dev

在git提交环节，存在三大部分：working tree, index file, commit

这三大部分中：

- **working tree**：就是你所工作所在的目录，每当你在代码中进行了修改，working tree的状态就改变了。
 - index file：是索引文件，它是连接working tree和commit的桥梁，每当我们使用git-add命令来登记后，index file的内容就改变了，此时index file就和working tree同步了。
 - commit：是最后的阶段，只有commit了，我们的代码才真正进入了git仓库。我们使用git-commit就是将index file里的内容提交到commit中。
- 总结一下：
- git diff：是查看working tree与index file的差别的。
 - git diff --cached：是查看index file与commit的差别的。
 - git diff HEAD：是查看working tree和commit的差别的。（你一定没有忘记，HEAD代表的是最近的一次commit的信息）

来自 <http://blog.sina.com.cn/s/blog_40e9d4dd0100xi8p.html>

checkout 如果是撤销工作区修改操作时，必须加上 -- 符号