|  |  |  |
| --- | --- | --- |
| Git命令学习 | | |
|  | **git diff [文件名]** | 比较文件内容修改前后的差异 |
|  | **git log** | 查看提交历史 |
|  | **git log *--graph*** | 查看详细分支图 |
|  | **git log *--graph --pretty=oneline --abbrev-commit*** | 查看简要分支图 |
|  | **git log --graph** | 看到分支合并图 |
|  | **git reflog** | 查看命令历史 |
|  | **git log --pretty=oneline** | 查看简略的日志信息 |
|  | **HEAD** | 指向当前分支的头指针，HEAD^上个版本，HEAD^^上上个，HEAD~100上100版本 |
|  | **git checkout -- file** | 丢弃修改，回到add或者commit之前  比如文件修改了还没有add，那么就回到最开始未修改的状态，如果修改了又add了，再修改，那么就回到add添加的时候，撤销再修改。如果要取消add，就要用git reset HEAD <file>,回到add之前。 |
|  | **git reset HEAD <file>** | 撤销暂存区的文件，回到add之前 |
|  | **git reset --hard HEAD^**  **或git reset --hard [id]** | 回退到历史的某个版本 |
|  | **git add/rm <file>** | 提交已经在本地删除的文件 |
|  | **git branch** | 查看本地分支 |
|  | **git branch <分支名>** | 创建分支 |
|  | **git branch -d(-D) <分支名>** | 删除分支，-d,分支已经合并情况下用；  -D，没合并强制删除分支 |
|  | **git checkout <分支名>** | 切换分支 |
|  | **git checkout -b <分支名>** | 创建并切换分支 |
|  | **git merge <分支名>** | 合并指定分支到当前分支（默认Fast-forward模式）  无冲突合并图：    有冲突合并图： |
|  | **git merge --no-ff -m "描述"<分支名>** | --no-ff 表示禁用Fast-forward模式，-m:表示commit，所以后面会跟描述信息 |
|  | **git stash** | 暂存当前工作区，如当前分支正在修改，还没有add或add后没commit，马上要切换到其他分支，用这个命令就可以保存现场，工作区会还原到没修改之前最后一次commit的状态  保存工作现场是以堆栈的形式，最先保存的在最下面，最后保存的在最上面  特殊例子:master分支test.txt写入1后add，再stash。工作区现在空了，再在test.txt写入2后，直接stash，恢复stash1（因为是堆栈形式，所以stash1是最先的），再恢复stash0，这时会提示错误，需要我们把test.txt提交add后才能恢复stash0（暂存工作区，之前add或者修改没add，恢复工作区后，都会变成没add状态），我们add后，继续恢复stash0，这是他会自动merge，然后我们手动解决冲突 |
|  | **git stash list** | 查看暂存现场 |
|  | **git stash apply <stash@{0}>** | 恢复工作区，stash暂存不删除 |
|  | **git stash drop <stash@{0}>** | 删除stash暂存 |
|  | **git stash clear** | 删除所有暂存 |
|  | **git stash pop <stash@{0}>** | 恢复的同时把stash内容也删了 |
|  | **git remote -v** | 查看远程库，-v查看库，包括库地址 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |