*新文件：Untracked files*

*修改的文件：Changes not staged for commit*

**git add**

**git reset --hard HAED**

**仓库**

**（本地）**

HEAD

HEAD^

HEAD^^

…

HEAD~N

…

（批量）

**git commit**

**暂存库**

*Changes to be Comitted*

|  |  |
| --- | --- |
|  |  |
| $ git init | 把当前目录变成Git可以管理的仓库 |
| $ git add <file> |  |
| $ git commit –m <commit log> |  |
| $ git status |  |
| $ git diff <file> |  |
| $ git log [--pretty=oneline] | 版本历史记录 |
| $ git reflog | 命令历史记录 |
| $ git reset --hard HEAD^ | 工作区回退到上一版本 |
| $ git reset --hard HEAD^^ | 工作区回退到上上版本 |
| $ git reset --hard HEAD~<n> | 工作区回退到前n版本 |
| $ git reset --hard <commit id> | 工作区回退到指定提交id的版本 |
| $ git reset HEAD <file> | 工作区回退到暂存区的版本，同时也撤销暂存区的版本 |
| $ git diff HEAD -- <file> |  |
| $ git checkout -- <file> | 工作区回退到暂存区的版本（若有），若没有暂存，则回到版本库最近一次提交的版本  若想同时撤消工作区和暂存区的修改，必须先git reset HEAD <file>，相当于清理暂存区，然后git checkout -- <file>，从版本库取回最近一次提交的版本 |
| $ git rm <file> | 实际使用提示要加-r参数？删除后要commit |
|  |  |

远程

|  |
| --- |
| $ git remote add <remote repo name> git@github.com:<user>/<repo name>.git |
| $ git push –u <remot repo name> master  首次推送到远程，加-u参数关联两个master分支 |
| $ git push <remote repo name> master |
| $ git clone git@github.com:<user>/<repo name>.git |
|  |

分支

|  |  |
| --- | --- |
|  |  |
| $ git checkout –b <branch> | git checkout加上-b参数表示创建并切换，相当于命令组合：  $ git branch <branch>  $ git checkout <branch> |
| $ git checkout <branch> | 切换到分支 |
| $ git branch | 查看分支，列出所有分支，当前分支前面有\*号 |
| $ git branch <branch> | 创建新分支 |
| $ git checkout <branch> | 切换到分支 |
| $ git merge <branch> | 把指定分支合并到当前分支 |
| $ git branch –d <branch> | 删除分支 |
| $ git log --graph --pretty=oneline –abbrev-commit | 查看分支合并图 |

