

git 使用:

1. `git init`: 创建新的 git 仓库
2. `git clone path`: 从远端服务器或本地检出仓库
3. `git add <filename>` 或者 `git add *`: 添加改动到缓存区
4. `git commit -m '代码提交信息'`: 实际提交改动, 将改动提交到本地仓库的 HEAD 中
5. `git remote add origin <server>`: 添加远程仓库
6. `git push origin master`: 将改动提交到远端仓库

查看远程分支: `git branch -a`

查看本地分支: `git branch`

删除本地分支: `git branch -d xxxx`

删除远程分支: `git branch -r -d origin/xxxx`

分支是用来将特性开发绝缘开来的。在创建仓库的时候, `master` 是默认的。

1. `git checkout -b feature`: 创建一个叫做 “feature” 的分支, 并切换到该分支
2. `git checkout master`: 切换回主分支
3. `git branch -d feature`: 删除 feature 分支
4. `git push origin <branch>`: 将分支推送到远端仓库

更新与合并

1. `git pull`: 将更新本地仓库至最新改动
2. `git merge <branch>`: 在当前工作目录中获取并合并远端的改动; 要合并其他分支到当前分支
3. `git add <filename>`: 解决冲突之后, 执行该命令表示合并成功
4. `git diff <source_branch> <target_branch>`: 在合并改动之前, 使用该命令查看

替换本地改动:

1. `git checkout -- <filename>`: 使用 HEAD 中的最新内容替换掉工作目录中的文件, 已添加到缓存区的改动, 以及新文件不受影响。
2. 丢弃所有的本地改动与提交, 可以到服务器上获取最新的版本并将你本地主分支指向它:
`git fetch origin git reset --hard origin/master`