

git常用操作整理

创建版本库

- `git init` 在当前目录下创建一个git仓库
- `git add .` 将当前目录下内容添加至仓库
- `git commit` 将当前已添加文件提交（可以多次添加一次提交）

版本控制

- `git status` 检查修改的文件/工作区状态, `git diff` 查看修改内容
- `git log` 查看历史提交版本
- `git reset --hard HEAD^` 可以回退至上一个版本, `git reset --hard [版本号]` 回退至相应的版本
- `git checkout --file` 丢弃工作区修改
- `git reset HEAD <file>` 撤销暂存区的修改
- `git rm <file>` 删除版本库中的某个文件

远程库管理

- `git remote add <name> <SSH>` 关联远程库
- `git push <name> <branch>` 提交至远程
- `git remote rm <name>` 解除与远程库的关联
- `git clone <SSH>` 从远程仓库克隆

分支管理

- `git branch <branchname>` 创建新的分支(-d 删除分支)
- `git checkout <branchname>` 切换到新分支
- `git switch <branchname>` 切换至新的分支（-c 同时创建分支）
- `git merge <branchname>` 合并指定分支到当前分支
 - 两个分支分别commit不同的内容后发生冲突
 - 发生冲突后需要解决冲突内容再进行merge
- 分支管理思路：每个人在自己的分支修改，合并至dev分支，dev分支得到稳定版本后发布至master。
- `git stash` 隐藏工作区修改，主要是解决尚未commit情况下的merge问题。`git stash pop` 恢复并删除stash内容。
- 常用feature分支表示新功能/目标

- `git remote` 查看更详细的信息
- `git push` 提交当前的更改至远程。一般的做法是先 `git pull` 在本地合并再push