Git

1. Git 三个工作区域的概念
2. Git 基本使用命令

2.1 配置用户和邮箱

git conifg --global user.name ‘name’

git config --global user.mail [example@163.com](mailto:example@163.com)

2.2 查看所有配置

git config --list

2.3 在现有目录初始化仓库

git init

2.4 添加到暂存区域

git add . // 添加所有文件

2.5 提交

git commit -m ‘注释’

2.6 检查当前文件状态

git status

git status –s (git status --short) // 状态简览

2.7 忽略文件

.gitignore

文件 .gitignore 的格式规范如下：

• 所有空行或者以 ＃ 开头的行都会被 Git 忽略。

• 可以使用标准的 glob 模式匹配。

• 匹配模式可以以（/）开头防止递归。

• 匹配模式可以以（/）结尾指定目录。

• 要忽略指定模式以外的文件或目录，可以在模式前加上惊叹号（!）取反。

详细文档: <https://github.com/github/gitignore>

2.8 查看已暂存和未暂存的修改

git diff

git diff --cached // 查看已暂存的将要添加到下次提交里的内容

2.9 跳过使用暂存区域

git commit -a -m ‘注释’

2.10 移除文件

git rm

git rm -f // 从暂存区域强制删除

git rm --cached // 从 Git 仓库删除, 但仍保留在本地工作目录

2.11 移动文件

git mv file\_from file\_to

2.12 查看提交历史

git log

git log -p -2 // -p，用来显示每次提交的内容差异。你也可以加上 -2 来仅显示最近两次提交

git log --stat // 每次提交的简略的统计信息

--pretty

git log --pretty=oneline // 在一行显示

git log --pretty=short

git log --pretty=full

git log --pretty=fuller







2.13 撤销操作

git commit --amend

2.14 取消暂存的文件

git reset HEAD filename // 只会修改暂存区域

2.15 查看远程仓库

git remote

git remote -v // 显示URL

2.16 添加远程仓库

git remote add <shortname> <url>

git remote add origin <https://github.com/xxx/demo.git>

git push origin master // 将本地master分支内容及版本信息推送到GitHub

git push origin dev // 将本地dev分支内容及版本信息推送到GitHub

2.17 创建分支

git branch dev // dev 是分支别名

git branch dev origin/dev // 创建dev分支且和远程dev分支同步

git checkout dev // 切换到 dev 分支

git push origin dev //提交dev分支内容到远程GitHub托管仓库的dev分支

git pull origin dev //从远程GitHub仓库获取dev分支最新内容，并合并到本地

git fetch origin dev //从GitHub仓库获取dev分支最新内容到版本库的分支

git merge origin/dev //将版本库的分支内容合并到工作区