

Git 笔记

1：初始化仓库

在相应的仓库目录下使用 `git init`

2：添加文件到仓库

首先 `git add <file>` //将文件添加到仓库中

然后 `git commit -m <message>` // message 一定要写有实质性内容的信息

#每次修改文件后，想要提交都要 add 和 commit 一次

使用 `git status` //查看提交结果 会给出文件修改和提交的信息

使用 `git diff` //查看文件被修改的内容

3:版本回退

使用 `git log` //查看每次提交对应的版本号 会显示 commit 时的信息

在 Git 中 HEAD 即为当前版本 上一个版本为 HEAD^上上个版本为 HEAD^^

使用 `git reset - -hard HEAD^` // 回溯到上个版本 或指定版本

使用 `git reset - -hard 版本号` //版本号可以不用写全

使用 `git reflog` //查看命令历史 附带版本号

4：工作区和暂存区

#工作区就是我们在电脑上看见的文件目录 而“工作区干净”的意思是 所有的

修改都已经被提交 暂存区就是我们使用 git 后把文件提交的地方

5 : 撤销修改

`git checkout --<file>` //如果,还没有进行 add 则回到版本库的状态,如果已经 add 了,就回到 add 时的状态

6 : 删除文件

首先手动删除要删除的文件,然后使用 `git rm <file>` 或 `git add <file>`

如果误删了,要从版本库恢复的话,则是 `git checkout --<file>`

7:远程库

先在 github 上创建一个新的库,然后在本地运行 github 给出的提示。

`git remote add origin git@—(网址)` //github 会给出提示 origin 为默认的名字

`git push -u origin master` //第一次提交的操作,以后用 `git push`

`git clone +github 上的网址` //可以克隆远程库到本地

`git remote -v` //查看远程库的信息

8 : 分支管理

`git branch` //查看分支

`git branch <name>` //创建分支

`git switch <name>` & `git checkout <name>` //切换分支

`git checkout -b <name>` & `git switch -c <name>` //创建与切换分支

`git merge <name>` //合并到当前分支

`git branch -d <name>` //删除分支

`git push origin +被推送的分支` //向远程库推送特定的分支

`git branch - -set-upstream-to=origin/远程库分支的名字 本地分支的名字`

// 用于链接本地的分支与远程库的分支

`git pull` //用于抓取远程库的分支的内容

`git checkout -b branch-name origin/branch-name` //创建和远程分支对

应的分支。

9:创建标签

`git tag` //查看所有的标签

`git tag v1.0` //打上一个标签

`git tag v1.0 版本号` //给指定的版本打上标签

`git show v1.0` //展示相应的标签的信息

`git tag -d v1.0` //删除指定的标签（本地）

`git push origin : refs/tags/v1.0` //删除远程库里指定的标签

`git push origin v1.0` //推送指定的标签

`git push origin -- tags` //推送所有的标签