**git 配置**

查看配置 git config -l

查看当前用户(global)配置 git config --global --list

(用户名和密码必须要配)

git config --global user.name ""

git config --global user.email ""

user.name=zy

user.email=13925205491@163.com

user.signingkey=love878449

**创建本地仓库的方法有两种**

1.创建全新的仓库 git init

2.克隆远程仓库 git clone

**基本命令**

git status [filename] 查看指定文件状态

git status 查看所有文件状态

git add . 添加所有文件到暂存区

git status

git commit -m "消息内容" 提交暂存区中的内容到本地仓库

git push 远程

git reset --hard 版本号 版本切换

git reflog 查看版本信息

git log 查看版本详细信息

git branch -v 查看分支

git branch 分支名 创建分支

git checkout 分支名 切换分支

git merge 分支名 把指定分支合并到当前分支上

git branch -d 分支名 删除本地分支(注意，如果分支包含未合并的更改和未推送的提交，则该 *-d*标志将不允许删除本地分支。此时，如果你确定了不想要分支的内容，可以使用 *-D*替换 *-d*来强制删除此分支)

**合并冲突**

要合并的两个分支改动了同一个文件，git不知道要采用哪个，还是两个都采用，需要有你来决定。CONFLICT (content): Merge conflict in .. 此时处于合并的中间状态。

1. git status 查看冲突的文件

You have unmerged paths. 我们现在正处于合并的中间状态，有一些没有合并的文件。

Unmerged paths: 列出所有未合并的文件，都显示为红色

1. 查看冲突内容。冲突内容分隔线怎么看？

**未冲突的内容（两个分支都未改动）在分隔线外面**

**<<<<<<< HEAD**

**Git当前所在分支修改的内容（准确来说是HEAD指针指向的分支修改的内容）**

**=======**

**要合并过来的分支修改的内容**

**>>>>>>> 合并过来的分支名**

1. 编辑冲突文件，解决冲突(记得删除三行分割线)

git add 冲突文件

git commit -m “提交信息”

如果不想继续合并了，退出合并的中间状态 git merge –abort

前文中提到，如果执行git merge合并时发生冲突，则会进入合并的中间状态。合并的中间状态下将无法执行其他一些操作（如切换分支）。  
如果不想继续合并，要先用git merge --abort命令退出。该命令会使你回到执行git merge 分支之前的状态。

**远程仓库**

github 有墙，比较慢，国内一般可使用gitee，公司中也可搭建自己的服务器gitlab

发生冲突协商即可，重新提交

**git remote 列出所有远程主机名 （便于管理，git要求每个远程主机都带一个主机名）**

git remote -v 查看远程主机网址，查看当前所有远程地址别名

git clone -o jQuery https://github.com/jquery/jquery.git  （克隆版本库时所有主机自动被命名为origin。若用-o可以改变命名）

git remote show <主机名>  （查看主机详细信息）

git remote add <主机名,别名> <远程地址> （添加远程主机）

git remote rm <主机名> （删除远程主机）

git remote rename <原主机名> <新主机名> （修改远程主机名）

git push 别名(没别名用链接) 分支 推送本地分支到远程仓库

git pull 别名(没别名用链接) 分支 拉取远程分支到本地库

**git工作原理**

不是复制了很多个富文本，而是在本地库内存里面记录了很多个日志（版本信息），通过调用指针来指向不同的版本。

master分支，defelop分支其实都是指向具体版本记录的指针。当前所在分支其实是由HEAD决定。所以创建分支本质就是多创建一个指针。

HEAD如果指向master，那么我们现在就在master分支上。

HEAD如果指向develop，那么我们现在就在develop分支上。

所有切换分支的本质就是移动HEAD指针。

git底层其实就是两个指针，HEAD指针指向分支。分支在指向具体版本。

GitHub

账号：[13925205491@163.com](mailto:13925205491@163.com)

密码：gaodian878449

username：zhengyi0409