1. git的基本命令
2. git init 初始化项目(会生成.git隐藏目录)
3. 配置命令

(1)git config --global user.name “zhupugang” 配置名字

(2)git config --global user.email [“327774162@qq.com”](mailto:\“327774162@qq.com\”) 配置邮箱

(3)git config --list 查看配置

注意：(2), (3)配置信息为了区分提交代码的人是谁(用户是谁)

1. 存储命令

(1)git add 文件的路径 (放到大门, 还没大到仓库, 暂存区)

注意：文件路径是 “./”时表示提交当前目录下的所有文件, 文件较多时使用

(2)git commit -m “提交说明” (放到了仓库里面, 版本库)

注意：git commit --all -m “提交说明” --all表示把所有修改的代码提交到仓库(版本库)

注意：(1), (2)重复使用(更新, 修改代码), 没有给定说明会跳到另一个界面可使用(esc + : + ! 这三个键强制退出)

1. git status 查看状态(修改代码的状态)

(1)红色表示修改的代码没有保存到大门口(没有执行git add命令)

(2)绿色表示修改的代码没有保存到git仓库里面(没有执行git commit命令)

(3)没有红色页面有绿色, 出现working directory clean表示修改的代码提交到了git仓库里面了

7>git log 提交记录(提交日志) git log --oneline简洁版日志(一行显示)

8>回退版本

(1)git reset --hard Head~0 回退到最近一次 1则是最近一次的上一次....

(2)git reset --hard 版本号 回退到指定版本号的版本(git log查看版本号, 生成的随机码)

(3)git reflog 查看版本切换记录(为了防止退回版本又要切换至原来版本)

9>分支

(1)git branch “分支名” 创建分支

(2)git branch 查看所在分支

(3)git branch -d “分支名” 删除分支

注意：删除分支需切换到主线上才能删除分支

(4)git branch --set-upstream-to=origin/master master

本地的分支和远程的分支之间建立联系(git pull 出现There is no tracking....使用)

(5)git checkout “分支名” 切换分支(分支名为master则是切换到主线上)

(6)git merge “分支名” 合并分支(一般用分支合并到主线)

注意：冲突则需要手动修改后, 在存入帮版本库(====== >>>>>>>)

1. GitHub(充当git服务器的作用， 不是git)
2. 通过地址提交代码值远程仓库

先打开github创建账号, 在里面初始化一个仓库

1. git push “远程仓库的地址” 分支名称(推送至远程仓库的哪个分支)

把本地仓库推送至远程仓库

1. git pull “远程仓库地址” 远程仓库的分支名称 从远程仓库下载到本地

注意：本地的仓库必须先初始化一个仓储 git init

1. git clone “远程地址” 克隆远程仓库, 如果多次执行会覆盖本地内容

注意：2>, 3>区别是pull会合并不同的地方, clone是克隆整个仓库, pull常用, clone一般第一次采用

四．通过ssh上传

1>ssh-keygen -t rsa -C [“327774162@qq.com”](mailto:\“327774162@qq.com\”) 生成.ssh文件(公钥私钥)

注意：找到对应的.ssh文件(一般在本机电脑用户里面)复制里面的字符串粘贴至github设置中的ssh设置(本地存储与远程存储关联)

五．简写命令

1>git remote add origin(后面提交的名字, 可以随便取什么名字) “远程地址”

配置后git pull origin(不需要填远程地址了) master

2>git remote rm origin(删除提交简写名字)

3>git push origin(配置的名字, 结合1>) -u master

配置后git pull(不需要填origin, 也不需要填远程地址了, -u(指定默认主机) )

注意：只能在git push上保存

1. 其他命令

1>git touch .gitignore 创建这个文档 文档里面写需要忽略上传的文件