Git的操作:

在本地代码的文件夹内,上传,或者在github上克隆到本地文件夹(.git)文件夹不可以上传,这样会把原来文件覆盖掉,上传克隆过来的文件需要删除.git文件

1,本地上传,cmd进入文件夹的盘符, git init-----git add .-----git commit -m “ ”

Git push

2.git 配置github 使用 git config查看,修改,

git config --global user.name 你的姓名

git config --global user.email 你的邮箱

1. 保存git账号密码:

git config --global credential.helper store

4, 查看配置 git config --global -l

**二:**

* **创建版本库,**
* **git clone <url> 克隆远程版本库**
* **git init 初始化本地版本库**

**1.查看状态与变更**

* **git status 查看状态**
* **git diff 查看变更内容**

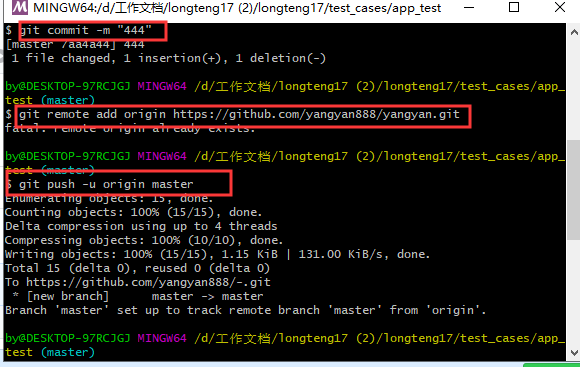
**2.追踪文件并生成快照到暂存区**

* **git add . 跟踪所有改动过的文件**
* **git add <file> 跟踪指定文件**

**4.提交版本到本地仓库**

* **git commit -m "commit msg" 提交所有更新过的文件**
* **git commit --amend 修改最后一次提交**
* **5,第一次提交到远程仓库**

**git remote add origin https://github.com/yangyan888/yangyan.git**

* ****git push -u origin master****
* 

### 三: 远程操作

* git remote -v 查看远程版本库信息
* git remote show <remote> 查看指定远程版本库信息
* git remote add <remote> <url> 添加远程版本库
* git fetch <remote> 从远程库获取代码
* git pull <remote> <branch> 下载代码及快速合并
* git push <remote> <branch> 上传代码及快速合并
* git push <remote> :<branch/tag name> 删除远程分支或标签
* git push --tags 上传所有标签
* 四:不常用的命令:

1. touch A.txt cmd中创建一个文件
2. echo 1234 >>A 把1234内容放到1234里
3. Cat a 打开a文件读取a文件的内容
4. git init // 初始化 在工作路径上创建主分支
5. git clone 地址 // 克隆远程仓库
6. git clone -b 分支名 地址 // 克隆分支的代码到本地
7. git status // 查看状态
8. git add 文件名 // 将某个文件存入暂存区
9. git add b c //把b和c存入暂存区
10. git add . // 将所有文件提交到暂存区
11. git add -p 文件名 // 一个文件分多次提交
12. git stash -u -k // 提交部分文件内容 到仓库 例如本地有3个文件 a b c 只想提交a b到远程仓库 git add a b 然后 git stash -u -k 再然后git commit -m "备注信息" 然后再push push之后 git stash pop 把之前放入堆栈的c拿出来 继续下一波操作
13. git commit -m "提交的备注信息" // 提交到仓库
14. 若已经有若干文件放入仓库，再次提交可以不用git add和git commit -m "备注信息" 这2步， 直接用
15. git commit -am "备注信息" // 将内容放至仓库 也可用git commit -a -m "备注信息"
16. \* git commit中的备注信息尽量完善 养成良好提交习惯 例如 git commit -m "变更(范围)：变更的内容
17. git hi -5 // 查看前5条内容
18. git hi --grep hello // 过滤提交信息里有hello字眼的内容
19. 回退上传内容:
20. git commit --amend -m "提交信息" // 回撤上一次提交并与本次工作区一起提交
21. git reset HEAD~2 --hard // 回撤2步
22. git reset --files // 从仓库回撤到暂存区
23. git reset HEAD // 回撤暂存区内容到工作目录
24. git reset HEAD --soft 回撤提交到暂存区
25. git reset HEAD --hard // 回撤提交 放弃变更 (慎用)
26. git reset HEAD^ // 回撤仓库最后一次提交
27. git reset --hard commitid // 回撤到该次提交id的位置

28.

git branch 分支名 // 新建分支

git branch // 查看当前所有分支

git checkout 分支名 // 检出分支

git checkout -b 分支名 // 创建并切换分支

git branch -v // 查看分支以及提交hash值和commit信息

git merge 分支名 // 把该分支的内容合并到现有分支上

git branch -d 分支名 // 删除分支

git branch -D 分支名 // 强制删除 若没有其他分支合并就删除 d会提示 D不会