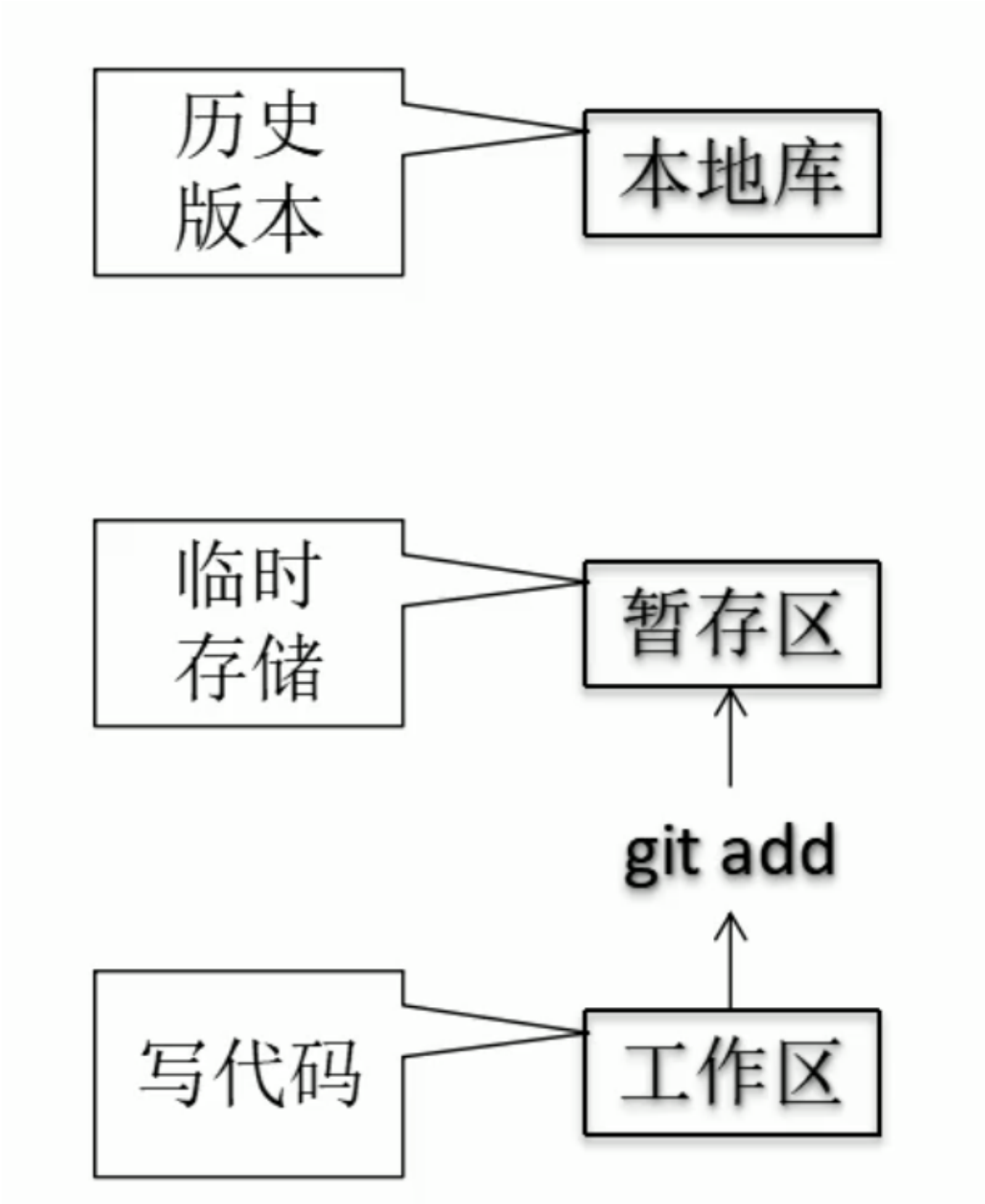
git结构



Git和代码托管中心

局域网环境下

- GitLab服务器

外网环境下

- GitHub
- 码云

Git命令行操作

本地库操作

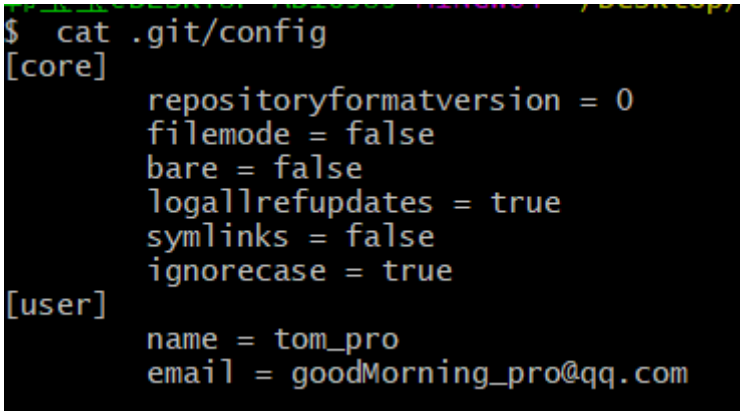
本地库初始化

- 命令
  - git int

设置签名

- 形式

  用户名：tom

  Email地址：309643066@qq.com
- 作用：区分不同开发人员的身份
- 辨析：这是设置的签名和登录远程库（代码托管中心）的账号、密码没有任何关系
- 命令
  - 项目级别/仓库级别：仅在当前本地库范围内有效
    - **git config user.name tom_pro**
    - **git config user.email goodMorning_por@qq.com**
    - 信息保存的位置：.git/config 文件
    - 

  - 系统用户级别：登录当前操作系统的用户范围
    - **git config --global user.name tom_pro**
    - **git config --global user.email goodMorning_por@qq.com**
    - 信息保存的位置：~

```
韩宝宝@DESKTOP-AD1690J MINGW64 ~
$ cat .gitconfig
[filter "lfs"]
        required = true
        clean = git-lfs clean -- %f
        smudge = git-lfs smudge -- %f
        process = git-lfs filter-process
[user]
        name = tom_global
        email = goodMorning_glb@qq.com
```

- 级别优先级
  - 就近原则：项目级别优先于系统用户级别，二者都有是采用项目级别的签名
  - **二者都没有：不允许**

基本操作

状态查看操作

- git status

  查看工作区、暂存区状态

添加操作

- git add [file name]

  将工作区的“新建/修改”添加到暂存区

提交操作

- git commit -m "commit message" [file name]

查看历史记录

- git log

```
韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/WeChat (master)
$ git log
commit f68bdcd8945737a0bdafc17e52bd934d4cb59cd8 (HEAD -> master)
Author: tom_pro <goodMorning_pro@qq.com>
Date:   Tue Jun 2 22:19:13 2020 +0800

    fot test history3
```

- git log --pretty=oneline

```
韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/WeChat (master)
$ git log --pretty=oneline
f68bdcd8945737a0bdafc17e52bd934d4cb59cd8 (HEAD -> master) fot test history3
58f2bbf895fc3fdeef437554cab4ed1c78090c88 for test history2
817a301dcebadaed13ed0d8dafd4c959b263e5a8 for test history
c7bfc1b97bb882a4f661a5131e2d52474090b437 My second commmit
bda7414ec54f130e8d501df5e2aad86014a0c266 My first commit.new file good.txt
```

- git log --oneline

```
韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/WeChat (master)
$ git log --oneline
f68bdcd (HEAD -> master) fot test history3
58f2bbf for test history2
817a301 for test history
c7bfc1b My second commmit
bda7414 My first commit.new file good.txt
```

- git reflog

```
韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/WeChat (master)
$ git reflog
f68bdcd (HEAD -> master) HEAD@{0}: commit: fot test history3
58f2bbf HEAD@{1}: commit: for test history2
817a301 HEAD@{2}: commit: for test history
c7bfc1b HEAD@{3}: commit: My second commmit
bda7414 HEAD@{4}: commit (initial): My first commit.new file good.txt
```

  - HEAD@{}：表示到这个版本需要几步

版本的前进后退

本质就是将HEAD指向不同的版本号

基于索引值操作

- git reset --hard [索引值]

使用^符号

只能后退

- git reset --hard HEAD^
  - 一个^代表退后一步
- git reset --hard HEAD~10
  - 后退10步

reset命令的三个参数对比

- soft
  - 仅仅在本地库移动HEAD指针
- mixed
  - 在本地库移动HEAD指针
  - 重置暂存区
- hard
  - 在本地库移动HEAD指针
  - 重置暂存区
  - 重置工作区

删除文件并找回

- 前提：删除前，文件存在时的状态提交到了本地库
- 操作：git reset --hard [指针]
  - 删除操作已经提交到本地库：指针位置指向历史记录
  - 删除操作尚未提交到本地库：指针位置使用HEAD

比较文件差异

- git diff [文件名]
  - 将工作区中的文件和暂存区进行比较
- git diff [本地库中历史版本] [文件名]
  - 将工作区中的文件和本地库的历史比较
- 如果不知道文件名，就会展示所有文件的差异

分支管理

在版本控制过程中，使用多条线同时推进多个任务

好处

- 同时并行推进多个功能开发，提高开发效率
- 各个分支在开发过程中，如果某一个分支以开发失败，不会对其他分支有任何影响。失败的分支删除重新开始即可

分支操作

创建分支

- git branch [分支名]

查看分支

- git branch -v

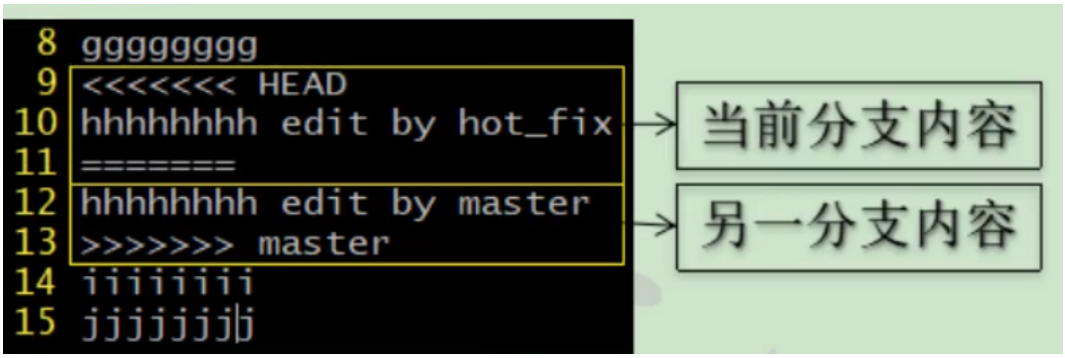
切换分支

- git checkout [分支名]

合并分支

- 第一步：切换到接受修改的分支上（增加新内容的分支）
- 第二步：git merge [被合并的分支名]

解决冲突

- 冲突的表现
  - 

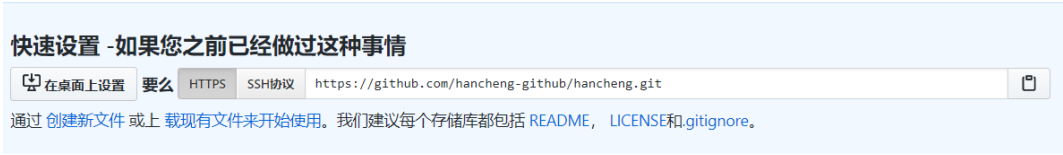
- 冲突的解决
  - 第一步：编辑文件，删除特殊符号
  - 第二步：把文件修改到满意

- 第三步：git add [文件名]
- 第四步：git commit -m "日志信息"
  - 注意：此时commit不能带文件名

远程库操作

推送

把本地库上传到远程库上

- 第一步：复制github连接
  - 

- 第二步：将连接保存在本地
  - git remote add [连接]
  - ```
    韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/git/ziliao (master)
    $ git remote add github https://github.com/hancheng-github/ziliao.git
    ```
- 第三步：上传
  - git push github master
  - ```
    韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/git/ziliao (master)
    $ git push github master
    ```

克隆

把远程库下载到本地库

- 第一步：新建一个文件夹并进入
- 第二步：复制github链接
- 第三步：git clone [链接]
  - ```
    韩宝宝@DESKTOP-AD1690J MINGW64 ~/Desktop/git
    $ git clone https://github.com/hancheng-github/ziliao.git
    Cloning into 'ziliao'...
    remote: Enumerating objects: 3, done.
    remote: Counting objects: 100% (3/3), done.
    remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
    Unpacking objects: 100% (3/3), 202 bytes | 9.00 KiB/s, done.
    ```

效果

- 完整的把远程库下载到本地
- 创建origin远程地址别名
- 初始化本地库

如何加入团队
    ```

- 然后就可以用协助者的身份上传到github了

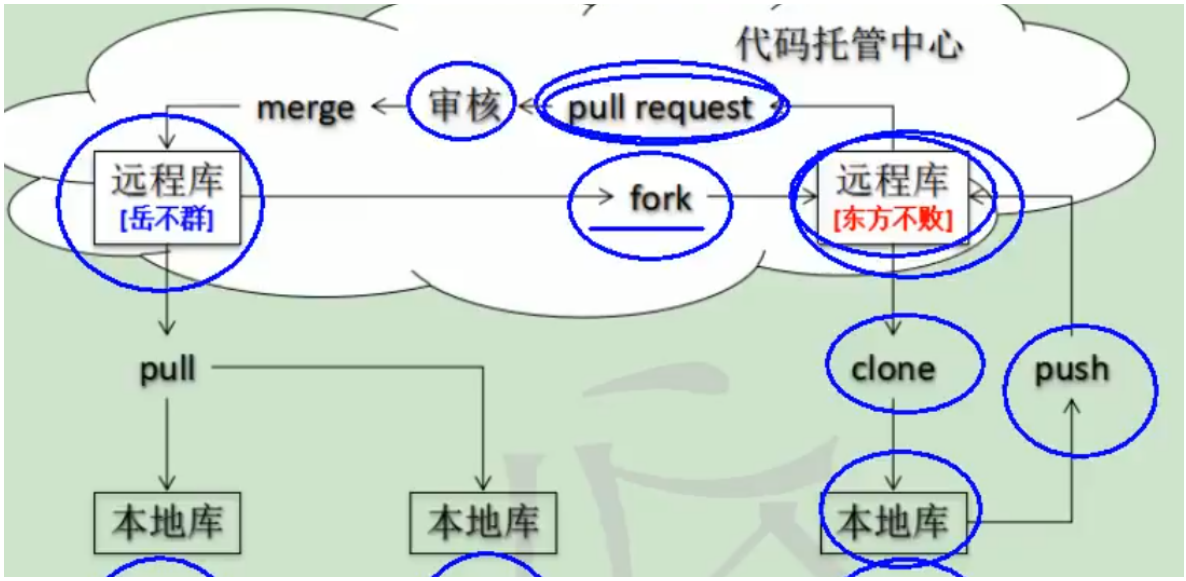
拉取

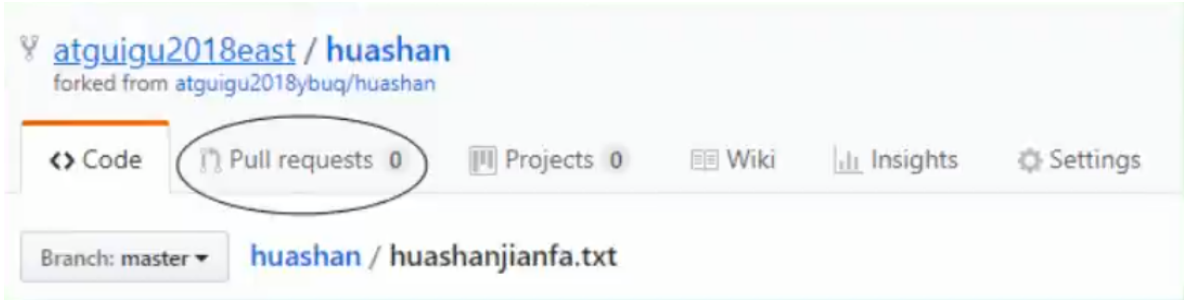
- pull=fetch+merge
- git fetch [远程库地址别名] [远程分支名]
- git merge [远程地址的别名/远程分支名]
- git pull [远程库地址别名] [远程分支名]

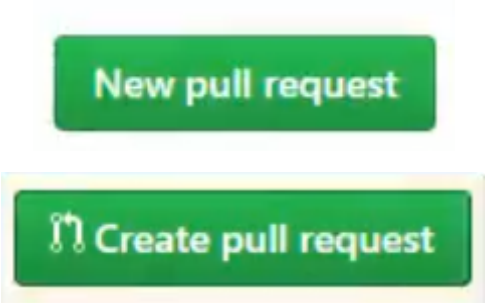
解决冲突

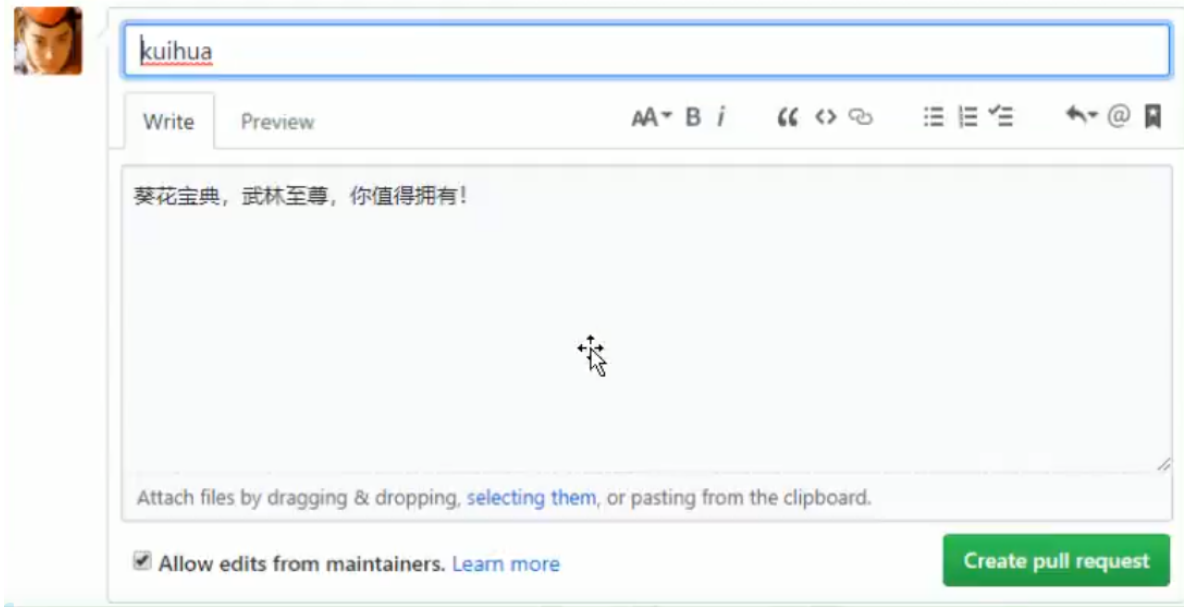
- 要点
  - 如果不是基于GitHub远程库的最新版所作的修改，不能推送，必须先拉去
  - 拉取下来后如果冲突状态，则按照"分支冲突解决"操作解决即可

跨团队协作


代码托管中心
merge
审核
pull request
远程库
[岳不群]
fork
远程库
[东方不败]
pull
clone
push
本地库
本地库
本地库


atguigu2018east / huashan
forked from atguigu2018ybuq/huashan
Code
Pull requests 0
Projects 0
Wiki
Insights
Settings
Branch: master
huashan / huashanjianfa.txt

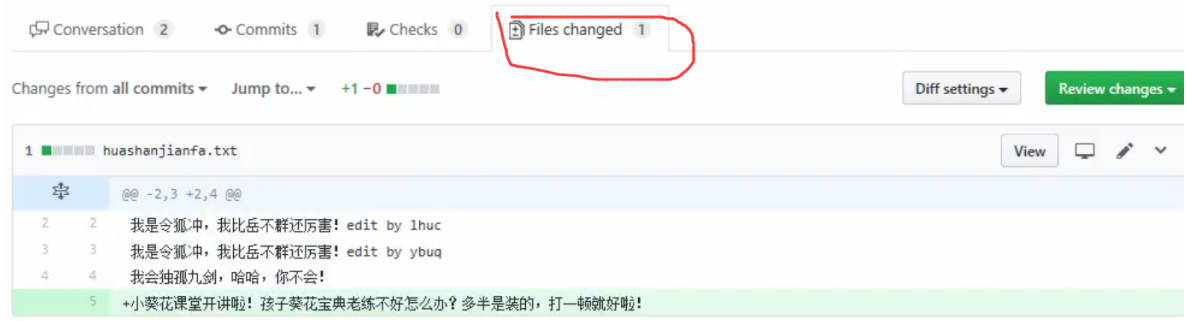

New pull request
Create pull request


kuihua
Write
Preview
葵花宝典，武林至尊，你值得拥有！
Attach files by dragging & dropping, selecting them, or pasting from the clipboard.
Allow edits from maintainers. Learn more
Create pull request

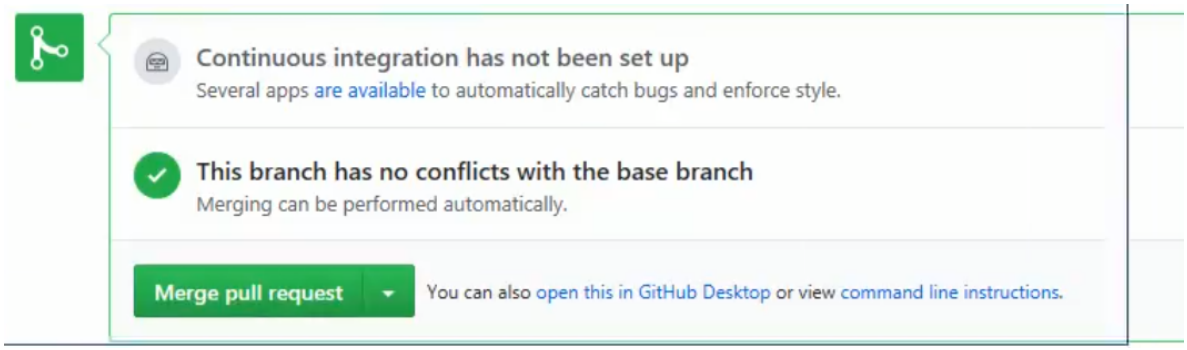

atguigu2018ybuq / huashan
Code
Issues 0
Pull requests 1
“岳不群”点这里


1 Open
0 Closed
老岳，你拜托我的事情办好啦！
#1 opened 2 minutes ago by atguigu2018east
“岳不群”点这里

审核代码


Conversation 2
Commits 1
Checks 0
Files changed 1
Changes from all commits
Jump to...
+1 −0
Diff settings
Review changes
1 huashanjianfa.txt
View
@@ -2,3 +2,4 @@
我是令狐冲，我比岳不群还厉害！edit by lhuc
我是令狐冲，我比岳不群还厉害！edit by ybuq
我会独孤九剑，哈哈，你不会！
+小葵花课堂开讲啦！孩子葵花宝典老练不好怎么办？多半是装的，打一顿就好啦！

合并


Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.

SSH登录

- 进入当前用户的加目录
  - cd ~
- 删除.ssh 目录
  - rm -rf .ssh
- 运行命令生成.ssh密钥目录
  - ssh -keygen -t tsa -C hanchengdyx@aliyun.com
- 进入.ssh目录查看文件列表
  - cd .ssh
  - ls -al
- 查看**id_rsa.pub**文件内容
  - cat id id_rsa.pub
- 复制**id_rsa.pub**文件内容，登录GitHub仓库---》Settings---》Deploy keys
- NEW Key
- 输入复制的密钥信息
- 回到git bash创建远程地址别名
  - git remote add origin_ssh git@github.com:hancheng-github/ziliao.git
- 推送文件进行测试