Git和GitHub

官网:github.com

详细介绍:https://github.com/guancgsuccess/Git_Learn

介绍Git

定义:是一个分布式的版本控制系统. - 源头是linux[开源]发展过程中的一个产品.

维度

四维 - 五维

分布式

本来一个应用存在于一台计算机上,现在将应用打散成多个逻辑,然后分布在不同的计算机中,通过网络协议

多个计算机上面的应用共同协作.

- 分布式的缓存框架 - redis
- 分布式的文件管理系统 - hdfs
- 分布式的框架 - dubbo
- 微服务架构 -springboot springcloud

安装路径

- 不要放在C盘
- 不要出现中文路径
- 目录名中不能含有特殊符号[Program files(x86)]

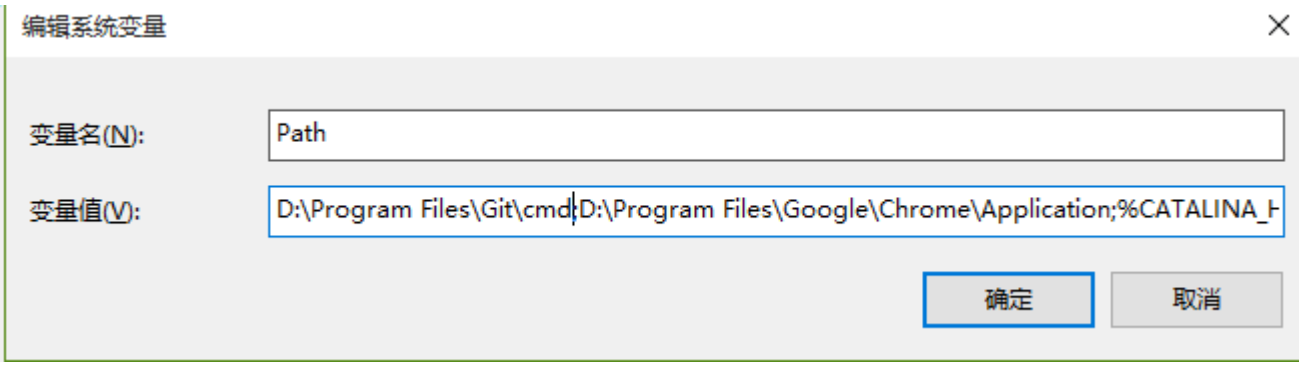
环境变量的配置

What - Why - How -> 3W法则

最终希望的结果:是在任意的目录下输入命令,都能够起到作用.

右击计算机 - 属性 - 高级系统设置 - 高级 - 环境变量 - 系统变量 - Path - 编辑

D:\Program Files\Git\cmd;其余的路径....



注意:环境变量配置成功之后,一定要重启控制台才有效.

人机交互的方式

- **GUI - Graphical User Interface** - 图形化用户界面方式 - 所谓的"傻瓜式"的操作 - 鼠标操作 - windows操作系统 - 鼠标操作的底层都是采用DOS命令.
- **CLI - Command Line Interface** - 命令行方式 - DOS操作系统 - 缺点:需要记忆大量的DOS命令.

windows操作系统执行命令的流程

1. 首先是到当前目录下边去寻找是否存在该命令,比如notepad.exe

2. 如果当前目录下边不存在该命令,则windows操作系统会继续寻找环境变量中的Path中的路径.

   从左到右的顺序,只要有一个目录中存在该命令,则执行

3. 如果当前目录中存在该命令,肯定就是直接执行的.

Git常用命令

1. git init

   初始化一个目录为git的仓库.[**目录中含有一个.git为后缀的目录(隐藏)]**

2. git status

   查看当前项目的状态[git-repo中的任何的操作.针对的是update操作]

3. git add .

   将当前目录中的所有的内容提交到缓冲区.

   git add 目录名称

   git add 文件名称

4. git commit -m "新建了一个文本文件test01.txt"

   真正意义上的提交.

5. git branch

   查看当前项目下的所有的分支

   默认会提供一个master主分支.

6. git branch 分支名

   比如:git branch issue01[分支是和任务挂钩的]

   注意:默认是会继承master分支下的数据的.

7. git checkout 分支名

   比如:git checkout issue01[切换到issue01的分支下]

8. git merge 分支名N

   将分支名N的分支合并到当前分支.

配置用户名和邮箱

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

HelloWorld - Git操作

1. 新建一个空的目录,在该目录下执行git init
2. 新建一个文本文件xxx.txt
3. 执行了git status
4. 执行git add .
5. 执行git status
6. git commit -m "新建了一个文本文件test01.txt"
7. git status
8. 创建新的分支 git branch issue01
9. 切换分支git checkout issue01
10. 在issue01分支下进行开发...创建新的文件
11. 重复到第四个步骤.[add commit]
12. 切回到master,观察git-repo中的目录的内容.
13. 又新建了一个分支issue03 [前提是先切换到master分支]

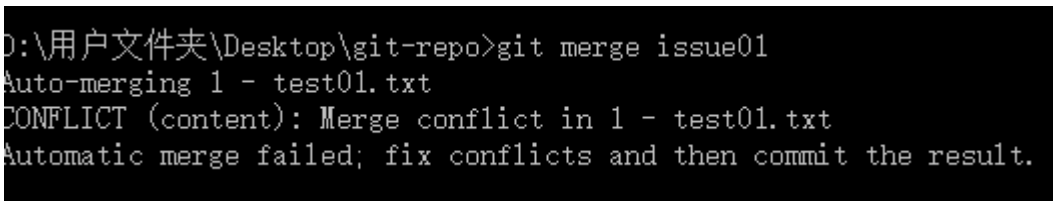
    进行新建文件[add commit]
14. 分别切换三个分支,查看一下内容数据....
15. 切换到master分支,分别执行各个分支的合并操作.

分支冲突

场景:当不同的分支操作同一个文件[同一行]的时候,会导致这个问题.

- 在master分支中,操作test01.txt,然后add commit
- 切换到issue01分支,执行git merge master

- 切换到issue03分支,执行git merge master
- 在issue03分支中修改test01.txt文件中的第三行代码.[add commit]
- 在issue01分支中修改test01.txt文件中的第三行代码.[add commit]
- 切换到master分支,先执行git merge issue03
- 接着执行git merge issue01



解决冲突问题

由master分支去查看test01.txt文件,保留该保留的,去除该去除的,执行add commit.