Git入门

一. Git基础

1. git 是什么

git是一种分布式版本控制系统

更直白说,团队开发时,管理代码用的软件.

面试时,容易被问到的一个东西.

1.2 git的对比

1、Git是分布式的SCM，SVN是集中式的

2、Git每个历史版本存储完整的文件，SVN存储文件差异

3、Git可离线完成大部分操作，SVN则相反

4、Git有着更优雅的分支和合并实现

5、Git有更强的撤销修改和修改版本历史的能力

6、Git速度更快，效率更高

2. git安装

官网安装 https://git-scm.com/

3. Git最基本的配置

在你用git之前,要先报家门,否则代码不能提交.

```
git config --global user.name #你是谁
$ git config --global user.email #怎么联系你
```

3.1 查看配置

```
git config --list
```

4. git的理论

4.1 git记录的是什么

git是讲每个版本都独立保存

4.2 工作区,暂存区,和git仓库

git的通过维护这三个区来达到版本管理控制点

工作区就是你工作的哪个文件件夹区域

暂存区,就是用来给你暂时存放代码的

git就是你代码最终存放的地方,里面有一个指针指向你最新的版本

4.3 工作流程

工作流程一般是这样的

1. 在工作目录中添加,修改文件
2. 将需要进行的版本管理的文件放入暂存区
3. 将暂存区域的文件提交到Git仓库

4.4 Git文件状态

Git管理的文件有三种状态

1. 已修改(modified)
2. 已暂存(staged)
3. 已提交(committed)

二. git 本地操作

1. 创建本地版本库(初始化)

```
git init
```

注意:

> 不要把仓库建在中文目录下,可能出问题.
>
> .git是个隐藏目录,不要乱碰.(你的每一次代码修改它都帮你记录着呢)

2. 添加文件查看状态

实时的查看工作状态

```
git status
```

可见,此时git发现有一个新文件,但并没有把此文件纳入管理.

如果文件修改了与版本库不一样,会提示,要不用commit提交更新版本

或者用版本库中的代码覆盖工作区

```
git checkout -- 文件名
```

3. 将文件提交暂存区

```
git add index.html
```

可以通过下面命令将提交暂存区的文件删除

```
git rm --cached README.md
```

4. 将暂存区的代码提交版本库

```
git commit -m "备注信息"
```

尝试修改文件,查询状态重新操作2,3,4步

5.快速讲修改后的代码提交版本库

这种简写的方式只针对修改文件

```
git commit -a -m "备注信息"
```

也可以如下写法

```
git commit -am "备注信息"
```

6. 查看版本

```
git log
```

简版日志

```
git log --oneline
```

7. 从暂存区回到工作修改状态

checkout 命令

```
git checkout -- 文件名
```

8. 版本回退

reset 命令

```
git reset HEAD~
```

HEAD~ ~ 的没一个波浪线就是一次快照,有几个波浪线就是前多少个快照

如果超过了版本库中版本,就会报错告诉你,无法在当前版本库中找到这个版本

如果~线太多 很容易写错,怎么办呢 可以简化 加入你有10个波浪线可以这么写HEAD~10

8.1 默认

将版本库中快照滚动到暂存区

```
git reset  --mixed HEAD~
```

这里 --mixed 是默认的 不写就是这个

写这行命令回发生两件事

1. 移动HEAD的指向,将其指向上一个快照
2. 将HEAD移动后指向的快照回滚到暂存区

8.2 版本库移动

只移动HEAD在版本库中的移动

```
git reset --soft HEAD~
```

这行命令只发生一件事

1. 移动HEAD的指向,将其指向上一个快照

8.3 替换工作区

将版本库中快照滚动暂存区同时覆盖工作区

```
git reset --hard HEAD~
```

这行命令只发生三件事

1. 移动HEAD的指向,将其指向上一个快照
2. 将HEAD移动后指向的快照回滚到暂存区
3. 将暂存区域的文件还原到工作目录

使用这个命令要小心,这个命令会将工作区内容替换掉

9. 版本前滚

reset 命令不仅可以回滚 还可以前滚

```
git reset 版本快照的ID号
```

这里也同样有 --mixed --soft --hard的区别

10 比较三个去文件的不同

10.1 比较工作区与暂存区的不同

```
git diff
```

如果显示不下 : j向下 k向上, f向下翻页 u向上翻页

10.2 比较两个历史快照

```
git diff 快照ID1  快照ID2
```

10.3 比较当前目录和Git仓库中的快照

```
git diff 快照ID
```

如果想比较最新的快照

```
git diff HEAD
```

10.4 比较暂存区与仓库的快照

```
git diff --cached 快照ID
```

11. 修改最后一次提交

```
git commit --amend
```

修改最后一次提交,不会生成新的快照

按i修改,修改后ESC 输入:wq 吗

可以看

```
git commit --amend -m '修改备注信息'
```

12 删除文件

不用担心文件被删除

如果一不小心把某个文件删除了,也不用太担心,git会实行检测你的每一步行为,

高度你可以通过checkout命令恢复

那么问题来了 我如何删除文件呢

12.1 删除工作目录和暂存区的文件

```
git rm '文件名'
```

该命令只会删除工作目录和暂存区的文件,也就是取消跟踪,在下次提交时不纳入版本管理

12.2 暴力删除

如果Git发现你暂存区和工作区的文件内容不一样,

你在执行rm删除命令时,会提醒你,免得你删除错误

当然你可以添加-f来达到暴力删除

```
git rm -f '文件名'
```

12.3 执行删除暂存区的文件

```
git rm --cached '文件名'
```

这个命令使用的前提是你工作区文件的内容和暂存区文件的内容相同

如果工作区文件内容有修改 使用这个命令就需要添加 -f 强制

```
git rm --cached -f '文件名'
```

13. 修改文件名

修改文件的名字会让git懵逼它会认为你是不是删除了前面的哪个文件,又添加了一个新的文件,其实我们只是单纯的觉得之前的哪个文件名不好了,修改了一下

所以如果想要修改文件名 让git帮你做是做好的

```
git mv '旧文件的文件名'  '新文件的文件名'
```

改完以后自动保存在暂存区,只需要提交到版本库就可以了

原理(git就是把下面散步结合在一起)

1. ren/mv 旧文件名 新文件名
2. git rm 旧文件名
3. git add 新文件名

总结:

1. add

   用于把工作目录的文件放入暂存区域

2. commit

   用于将暂存区的文件提交到Git仓库

3. reset

   用于把Git仓库文件还原到暂存区

4. checkout

   用于把暂存区域的文件还原到工作目录

三. 本地分支及分支管理操作

分支是什么概念呢,有那么重要吗? 就是我有一个大项目上线了,突然需要修改某些功能,咱们能在原来的项目上更改吗,不可以,所以咱们可以创建分支来在分支上修改

1. 查看当前仓库分支

```
git branch
```

2. 创建当前仓库分支

```
git branch '分支名'
```

一般不会出错,除非你的分支已经存在了,保存告诉你分支已存在

3. 切换分支

```
git checkout '分支名'
```

4. 创建并切换分支

```
git checkout -b '分支名'
```

5. 图形查看各分支上到版本信息

```
git log --graph --all --oneline
```

6. 分支的合并

天下大事合久必分,分久必合,仓库里的快照都是按照时间来存放的,我们把串联这些快照的时间轴称为分支,

默认情况下git 只有一条master分支,显示开发中,从来不存在一条分支定乾坤的事情

时间开发中分支分布应该是这样

```
git merge '分支名'
```

6.1 如果没有冲突

没有冲突会显示Past-forward快进

6.2 如果合并有冲突

会显示conflict 冲突

如果自动合并失败 就需要手动解决冲突,然后合并提交

7. 删除分支

7.1 普通删除

对于合并后不予要的分支要及时删除,面对乱糟糟的

全写

```
git branch --delete '分支名'
```

简写

```
git branch -d '分支名'
```

删除注意事项

1. 不能再当前分支删除当前分支
2. 分支的删除必须是合并了此分支的分支上删除

7.2 强制删除

```
git branch -D '分支名'
```

四. 远程仓库

世界上最大的代码存放网站和开源社区,因此我们常常戏称世界上最大的同性交友网站.

远程仓库可以方便大家协同开发

1. 目的

借助github 托管你的项目代码

2. 注册在线仓库的账号

国外: http://www.github.com

国内: http://git.oschina.net

github.com也是目前程序员的装逼利器,在github上挂个项目,逼格瞬间提升两三档.

不过由于是国外网站,速度不咋样.

3. 在远程仓库创建项目

远程仓库提供了两种连接仓库的方式 SSH HTTP

4. 远程操作

4.1 查看远程仓库的名字

```
git remote
```

4.2 查看远程仓库的关联

```
git remote -v
```

4.3 代码推送远程

```
git push origin master
```

5. 多人协作开发

5.1 拉去远程仓库代码

1. 方式一

   ```
   git fetch
   ```

   fetch 拉去并不合并

2. 方式二

   ```
   git pull
   ```

pull拉去并合并