

5分钟git

[TOC]

前言

很多人说git的使用5分钟就会了。但是不少人使用的时候却遇到一些难题就卡很久。

其实，5分钟是有效学习的五分钟。还有很多时间要用在查找上，排错上，等等。

现在让我们真正开启五分钟（git）。

以及15min(github)

ps：以下资料均为网络查找。

PS:前置知识↓
会打开控制台，会查询、使用控制台命令
懂得自己去官网下载需要的软件。

git的使用

简介：git是一个版本管理工具，它能够用来存储过去对文件的修改为版本，可以回退到任意保存好的版本中。
可以在控制台或者git bush中输入相应命令来使用这个工具。

创建仓库：

```
git init
```

会把当前文件初始化为仓库

生成版本

```
git add .  
git commit -a -m "版本名"
```

该两行命令会在当前分支生成当前仓库内所有信息的一个版本。如果当前没有分支，会创建新的分支

"版本名"为自定义的对当前版本的注释

查看版本

```
git log
```

该命令可以查看当前版本以及之前的版本的版本编号和版本注释

```
git reflog
```

该命令可以查看所有版本的版本命名和版本注释

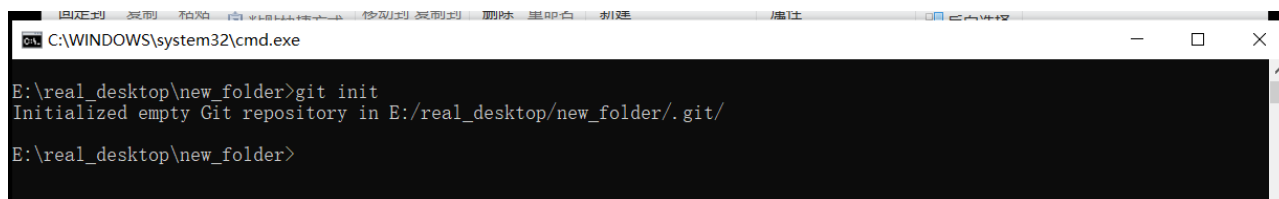
还原版本

```
git reset --hard 版本号
```

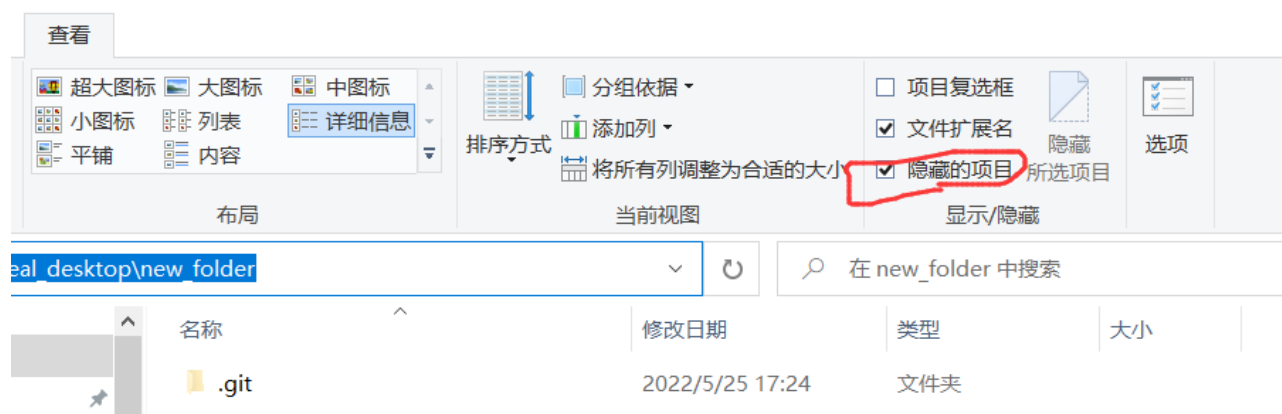
通过版本编号还原到对应版本

使用示例

1. 创建仓库



在控制台进入到要创建为仓库的文件夹路径下，输入命令git init成功后下面会有说明创建了一个空仓库的提示，也就是创建了一个还没有保存任何版本的仓库



在修改查看属性后能够看到在初始化为仓库的文件夹下面出现了一个.git文件夹，该文件夹会用来保存git进行版本管理需要的所有信息，

该文件夹默认为隐藏状态，不需要去管！！

2. 生成版本

做出一个修改后，生成版本.重复两次

在 new_folder 中搜索				
名称	修改日期	类型	大小	
1.txt	2022/5/25 17:34	文本文档	0 KB	

```
E:\real_desktop\new_folder>git add .
E:\real_desktop\new_folder>git commit -a -m "添加1.txt文件"
[master (root-commit) ca2e2e7] 添加1.txt文件
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
```

在 new_folder 中搜索				
名称	修改日期	类型	大小	
1.txt	2022/5/25 17:34	文本文档	0 KB	
2.txt	2022/5/25 17:37	文本文档	0 KB	

```
E:\real_desktop\new_folder>git add .
E:\real_desktop\new_folder>git commit -a -m "添加2.txt文件"
[master d3aa68d] 添加2.txt文件
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.txt
```

3. git log查看版本

```
E:\real_desktop\new_folder>git log
commit d3aa68d0a4bf8f24002b4d2c0b3558aa615b58c8 (HEAD -> master)
Author: cncsmonster <cncsmonster@163.com>
Date: Wed May 25 17:41:41 2022 +0800

    添加2.txt文件

commit ca2e2e74be4524e34b583abbee4d3da82d5b9e0e
Author: cncsmonster <cncsmonster@163.com>
Date: Wed May 25 17:35:00 2022 +0800

    添加1.txt文件
```

4. 还原到指定版本

```
E:\real_desktop\new_folder>git reset --hard ca2e2e74be4524e34b583abbee4d3da82d5b9e0e
HEAD is now at ca2e2e7 添加1.txt文件
```

在 new_folder 中搜索				
名称	修改日期	类型	大小	
1.txt	2022/5/25 17:34	文本文档	0 KB	

5. git reflog查看所有版本

在4还原到第一个版本后在第一个版本的状态下再创建新版本，再回到原本的第二个版本--“添加2.txt”

创建新版本↓

名称	修改日期	类型	大小
1.txt	2022/5/25 17:34	文本文档	0 KB
3.txt	2022/5/25 17:48	文本文档	0 KB

```
E:\real_desktop\new_folder>git add .
E:\real_desktop\new_folder>git commit -a -m "添加文件3.txt"
[master eabeeeea] 添加文件3.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 3.txt
```

再使用git log查看版本。则只能查看到当前版本及之前的版本，查看不到原本的第二个版本↓

```
E:\real_desktop\new_folder>git log
commit eabeeeea068f731773533c001151f4fdac464dd82 (HEAD -> master)
Author: cncsmonster <cncsmonster@163.com>
Date: Wed May 25 17:49:45 2022 +0800

    添加文件3.txt

commit ca2e2e74be4524e34b583abbee4d3da82d5b9e0e
Author: cncsmonster <cncsmonster@163.com>
Date: Wed May 25 17:35:00 2022 +0800

    添加1.txt文件
```

而使用git reflog查看版本可以查看到所有创建过的版本以及版本之间的转移信息↓

```
E:\real_desktop\new_folder>git reflog
eabeeeea (HEAD -> master) HEAD@{0}: commit: 添加文件3.txt
ca2e2e7 HEAD@{1}: reset: moving to ca2e2e74be4524e34b583abbee4d3da82d5b9e0e
d3aa68d HEAD@{2}: commit: 添加2.txt文件
ca2e2e7 HEAD@{3}: commit (initial): 添加1.txt文件
```

里面最前面提供的版本编号缩写也可以用来还原到对应版本↓

```
E:\real_desktop\new_folder>git reset --hard d3aa68d
HEAD is now at d3aa68d 添加2.txt文件
```

在 new_folder 中搜索				
名称	修改日期	类型	大小	
1.txt	2022/5/25 17:34	文本文档	0 KB	
2.txt	2022/5/25 17:52	文本文档	0 KB	

git原理

git不是每个版本都保存了仓库内所有文件的信息。每个版本保存的只是该版本相对之前版本的修改。这种保存方式可以大大减少要保存的信息量。

另外，git有高效的数据压缩方法，能够把要保存的信息压缩成很小的数据。

于是，能够使用git进行大规模的大量的版本管理而不用担心给存储空间造成太大的负担。