

10. Git版本管理工具使用

Git简介

Git是目前世界上最先进的分布式版本控制系统。

需求场景

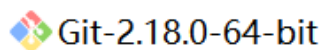
1. 产品工作人员常需在公司、家里各有一台电脑，那两台电脑间的文件同步是一个很大的问题。
2. 产品、设计人员的工作文件常产生多个版本进行保存和管理。在本地进行多版本管理，占据内存空间的同时易造成文件混淆和丢失。

工具优势

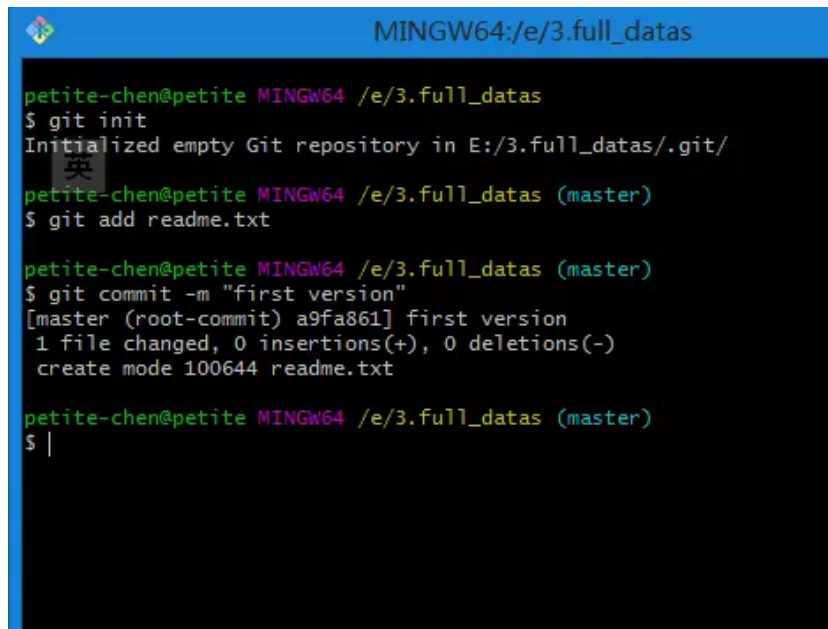
1. 工具免费；
2. 轻松实现文件远程同步；
3. 强大的版本管理功能。

使用教程

使用前准备：下载Git安装包，大小约39MB。完成安装。



一、本地创建版本库



```
MINGW64:/e/3.full_datas

petite-chen@petite MINGW64 /e/3.full_datas
$ git init
Initialized empty Git repository in E:/3.full_datas/.git/

petite-chen@petite MINGW64 /e/3.full_datas (master)
$ git add readme.txt

petite-chen@petite MINGW64 /e/3.full_datas (master)
$ git commit -m "first version"
[master (root-commit) a9fa861] first version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 readme.txt

petite-chen@petite MINGW64 /e/3.full_datas (master)
$ |
```

1. 新建一个文件夹

通过以下命令将其变成Git可以管理的工作区。

```
1 git init
```

如图：在E盘新建文件夹3.full_datas，其中，.git用于跟踪管理版本，在本地默认隐藏，不用理会。

2. 添加文件到Git仓库暂存区

使用以下任一命令添加文件。

```
1 git add <file>
2 git add -A
```

注意，可反复多次使用。

3. 提交文件到仓库分支

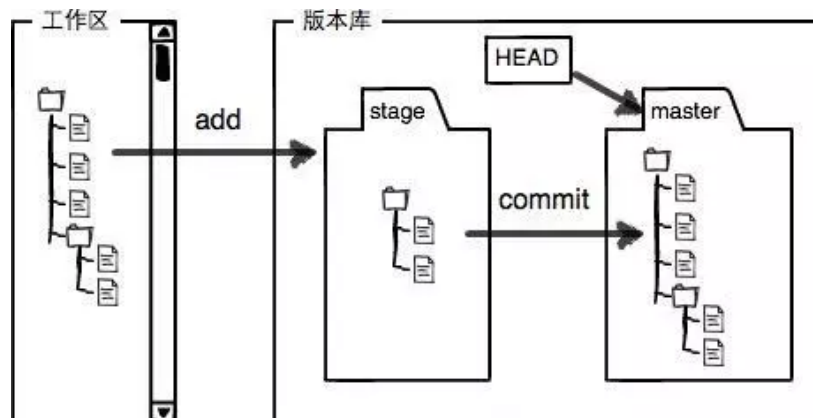
使用以下命令提交文件。

```
1 git commit
```

如图：在3.full_datas中新建文件readme.txt，通过`git add readme.txt`添加文件，使用`git commit -m "first version"`提交修改。其中-m是message（消息）的意思，“first version”是提交本版本的说明。

附加解释.git

.git 可以算是Git的版本库。其中存有很多信息，最重要的是称为**stage**（或者叫index）的**暂存区**和Git为我们自动创建的第一个分支**master**，以及指向master的一个指针叫**HEAD**。



`git add`把文件添加进去，实际上就是把文件修改添加到暂存区；

`git commit`提交更改，实际上就是把暂存区的所有内容提交到当前分支，若未新建分支，则实际上就是提交到**master**。

简单理解为，需要提交的文件修改通通放到暂存区，然后，一次性提交暂存区的所有修改。所以前面提到，可反复多次使用`git add`命令添加文件，然后进行一次性提交。

二、本地分支推送至远程仓库

1. 注册github账号，创建SSH Key

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Sessions

Blocked users

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

zhenyan@ouba
Fingerprint: 76:8c:d3:08:33:4b:7b:8c:eb:26:ae:43:cd:ce:65:30
Added on 3 Jan 2018
Last used within the last 4 months — Read/write

Delete

gold_book_20180724
Fingerprint: 11:6b:0a:9b:c6:9e:ea:1c:a5:b6:0b:0f:c4:28:a5:b7
Added on 24 Jul 2018
Last used within the last 2 months — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

2. 本地关联一个远程库

使用以下命令

```
1 git remote add origin git@server-name:path/repo-name.git
```

3. 推送分支内容到远程库

使用以下命令第一次推送master分支的所有内容：

```
1 git push -u origin master
```

此后，每次本地提交后，只要有必要，就可以使用命令`git push origin master`推送最新修改。

三、掌握工作区状态

```
petite-chen@petite MINGW64 /e/3.full_datas (master)
$ git status
On branch master
nothing to commit, working tree clean

petite-chen@petite MINGW64 /e/3.full_datas (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   readme.txt

no changes added to commit (use "git add" and/or "git commit -a")

petite-chen@petite MINGW64 /e/3.full_datas (master)
$ git diff
diff --git a/readme.txt b/readme.txt
index e69de29..8e49f37 100644
--- a/readme.txt
+++ b/readme.txt
@@ -0,0 +1 @@
+第一次修改
\ No newline at end of file
```

使用以下命令查看状态，如上图

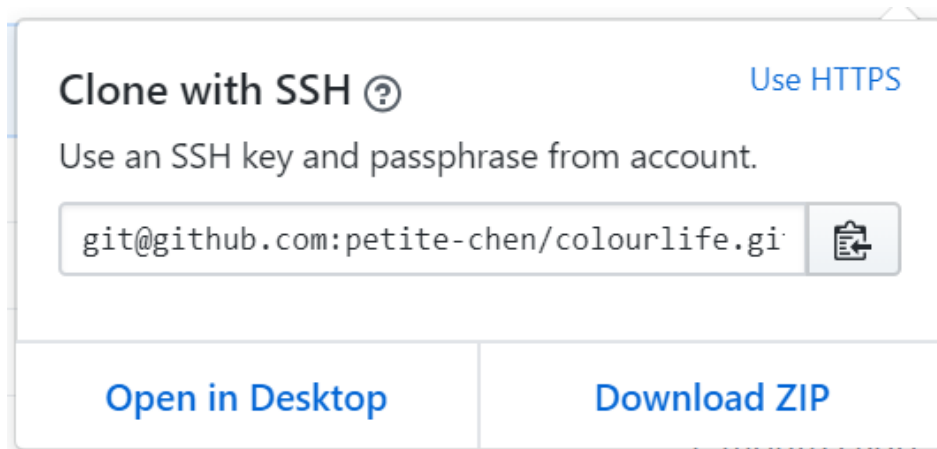
```
1 git status
```

如果`git status`告诉你有文件被修改过，用以下命令查看修改内容。

```
1 git diff
```

四、克隆远程仓库

要克隆一个仓库，首先必须知道仓库的地址，比如github里某分支地址



然后用Git执行以下命令克隆

```
1 git clone
```

Git支持多种协议，包括https，但通过ssh支持的原生git协议速度最快。

五、版本回退

HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令
以下实现。

```
1 git reset --hard commit_id。
```

穿梭前，用`git log`可以查看提交历史，以便确定要回退到哪个版本。

要重返未来，用`git reflog`查看命令历史，以便确定要回到未来的哪个版本。

重点问题场景

1. 如何撤销修改

场景1

当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，执行以下命令

```
1 git checkout -- file
```

用版本库里的版本替换工作区的版本。

场景2

当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令`git reset HEAD file`，就回到了场景1，第二步按场景1操作。

场景3

已经提交了不合适的修改到版本库时，想要撤销本次提交，参考[版本回退]，不过前提是没有推送到远程库。

2. 使用Git有一定门槛，有没有其他更好的工具来满足文件多端同步、版本管理的痛点？

关于多端同步、版本管理这两个痛点，常规使用的可能是以下两种类型的工具：

- 石墨：[可多人实时协作的云端文档与表格](#)。缺点：常规office文件类型没问题，但是比如原型文件等特殊文件类型没法在线编辑，则无法进行管理。此外大型文件容易崩溃。
- 百度云同步盘：只能同步，没法进行版本管理。且若不成为会员的话，上传和下载速度较慢，且有各种功能限制，成本较高。

与以上两种类似的其他工具不展开扩展。

git除了有一定的使用门槛，在确实有所需要的情况下，是一个非常强大的辅助工具了。

更多使用技能参考网站：<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>