

## 初始化

### 创建版本库

git 基于文件夹(工作树)进行版本控制, 在一个文件夹中创建 git 版本库:

```
$ cd project/ $ git init ❶ Initialized empty Git repository in .git/
```

❶ 输出信息: 在当前文件夹的 .git/ 目录下创建版本库

将文件提交到 git 索引:

```
git add file1 file2 file3 .....
```

更方便的作法是将当前文件夹中的所有文件全部加入到索引中

```
git add .
```

- 可以在 .gitignore 文件中设置排除的文件(通常把临时文件排除)



注意

git 只负责管理被索引的文件

此时, 文件还没有被提交到版本库。向版本库提交第一个版本:

```
git commit ❶ git commit -m "备注"
```

❶ 调用系统默认编辑器编辑备注内容

### 版本库状态

使用 **git status** 命令查看版本库状态。先创建一个演示版本库:

```
mkdir sandbox #新建一个文件夹 cd sandbox/ #进入该文件夹 git init
```

这时使用 **git status** 查看版本库状态:

```
# On branch master
nothing to commit (working directory clean)
```

对文件进行一些操作:

```
vi a #编辑 arm b #删除 b touch c #新建 c
```

再用 **git status** 查看:

```
# On branch master #在 master 分支上 # Changes to be committed: #已提交到索引, 等待提交到版本库 (其
```



注意

如果只是想删除该文件夹中的版本库, 只要删除 .git/ 目录即可

```
rm -rf .git
```

### 配置

git 初始化后, 会在 .git/ 目录下创建一个版本库, 其中 .git/config 为配置文件。

### 用户信息

为当前版本库添加用户信息<sup>[62]</sup>:

```
[user]      name = kardinal      email = 2999am@gmail.com
```

也使用全局用户信息，在`~/.gitconfig`中写入上述内容，或者使用命令：

```
git config --global user.name "kardinal"git config --global user.email 2999am@gmail.com
```

## 语法高亮

在`~/.gitconfig`文件中添加如下语句，使用容易阅读的彩色来输出信息：

```
[color]      branch = auto      diff = auto      status = auto
```

或者自己定义：

```
branch.current      # color of the current branchbranch.local      # color of a local bra
◀ |                 ▶
```

---

<sup>[62]</sup> 这是必需的，请不要忽略