

# Git 版本控制

---

## Git 介绍

---

什么是版本控制？

代码上一步可能健步如飞，下一秒就灰飞烟灭。

Git是版本控制得一种工具

Linus Torvals

时间：2005-4月3号开始开发 4月6日完成，4月7号上线使用

## Git 安装

---

Linux

```
sudo apt-get install git
```

Mac

```
gem install git
```

Windows

<https://git-scm.com/downloads>

## 一些基础Linux命令

---

- ls List
- cd change dictionary

## Git 本地使用

---

### 快速入门

---

创建版本库

```
git init
```

查看工作区状态

```
git status
```

新建的文件如果 untracked，表明该文件没有被我们的版本控制所监视。

**添加文件** add 要么监视起来，新的变更监视起来

```
git add <Filename>
```

添加文件文件后，我们的文件状态就变成了 staged

## 提交版本

```
git commit -m '版本的描述'
```

当提交版本之后，文件的状态 unmodified

怎么查看版本库

```
git log
```

```
cailei@LAPTOP-3VSUT4NC ~/mywork
$ git log
commit 18cbd6d9644003e3f4fa5b3d4e3
49ce52e706d46 (HEAD -> master)
Author: cailei <caileid@gmail.com>
Date: Tue Mar 9 15:19:13 2021 +0800
```

完成第一首诗

```
commit 956151a4b0f8e7631fea6ac78a1
908f5c82e9e50
Author: cailei <caileid@gmail.com>
Date: Tue Mar 9 15:12:06 2021 +0800
```

完成第一首诗前两行

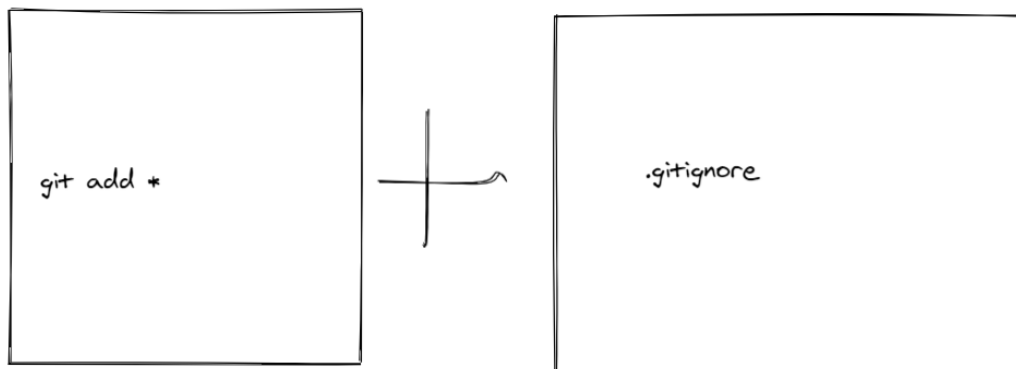
```
cailei@LAPTOP-3VSUT4NC ~/mywork
```

## 一些补充

多个变更进行暂存

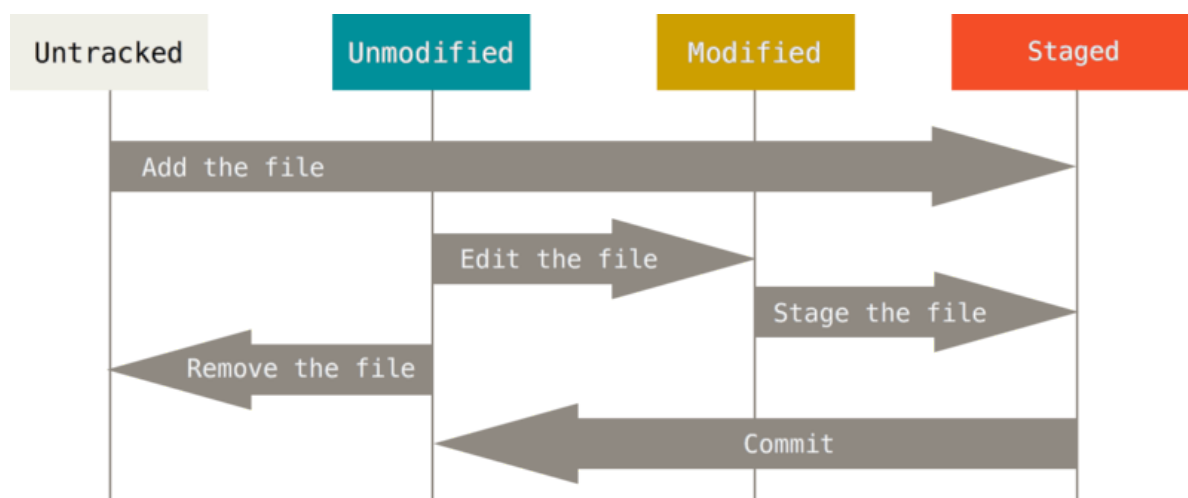
```
git add *
```

.gitignore 定义忽略哪些文件的不去监视变更



## 文件的状态

站在版本控制角度上，文件有4种状态。



- untracked: 未跟踪，文件在工作区当中，还未加入到版本库当中。
- unmodified : 未修改，文件已经进入暂存区，暂存区和版本库一致
- modified 文件被修改，文件进入暂存区。还未提交
- Staged暂存态： 被标记了修改，形成版本时会被提交的文件

## 文件的回退操作

获取到每个版本散列码，以散列码为表示，进行版本回退

```
git reset --[选项] [SHA code]
```

- hard 不保留所有变更
- soft 保留变更且变更内容处于staged状态
- mixed 保留变更切变更内容处于 modified 状态

## 分支

什么是分支？就是软件项目中新启动的一条独立开发线

为什么要有分支？

- 一个分支可以代表单独的客户开发
- 一个分支也封装一个开发阶段
- 隔离特性的开发，研究特别BUG

如何建立分支

```
git checkout -b <branchName> <TemplateName>
```

```
git branch <branchName> <TemplateName>
```

如何切换分支

```
git checkout <branchName>
```

如何查看分支

```
git branch -v
```

如何进行分支合并

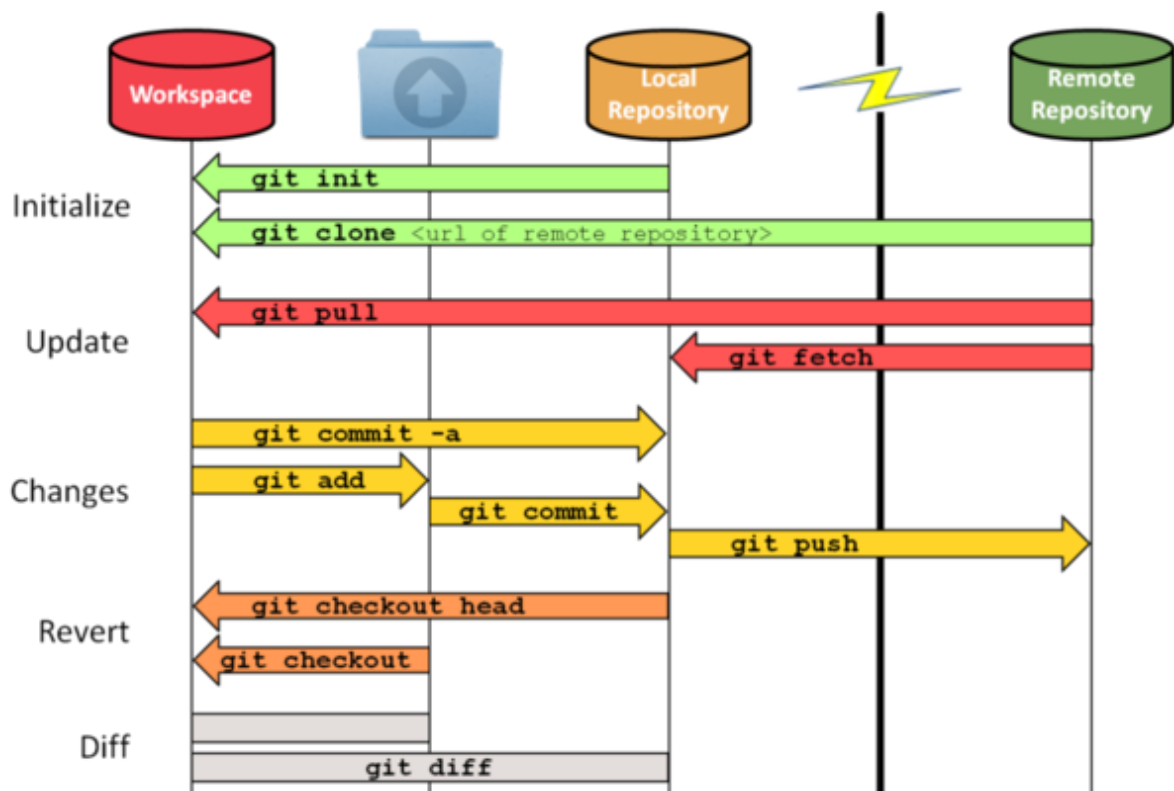
```
git merge <branchName>
```

解决冲突

- 手动解决冲突
- 将冲突文件添加后提交

## Git 网络使用

为什么要使用远程仓库：分布式开发的需要，我们需要一个远程仓库



与远程仓库建立联系

```
git remote add <remoteName> <remoteURL>
```

与远程仓库进行交互 向远程仓库push分支

```
git push <仓库名称> <分支名称>
```

克隆远程仓库

```
git clone <remoteURL>
```