

Git

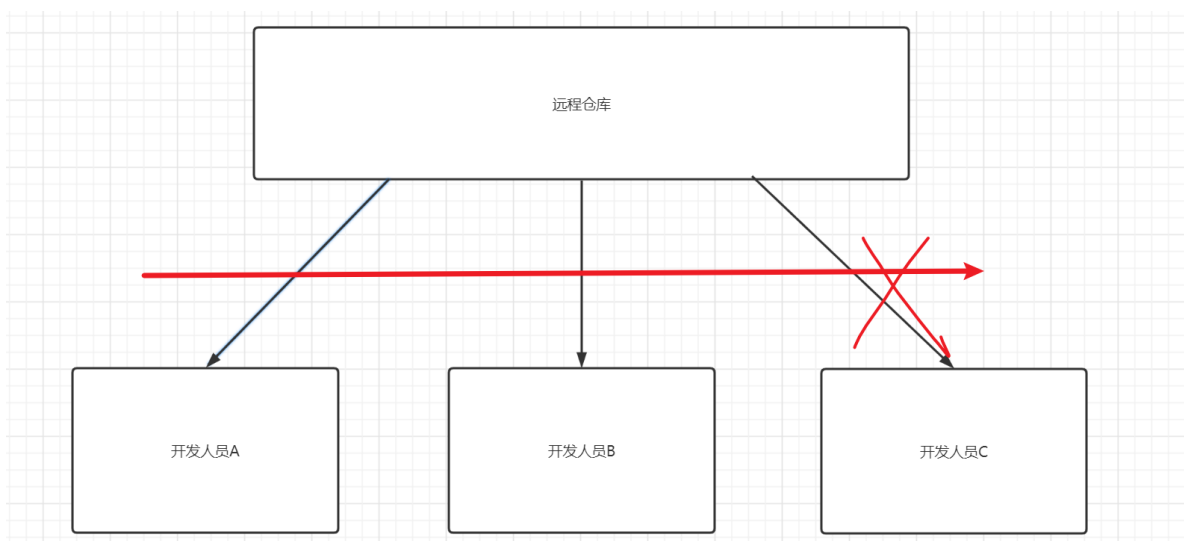
1. 介绍

Git是一个版本控制工具。可以追踪某个文件夹下的所有的文件的产生的变化和历史状态。

2. Git工作流程

集中式的版本控制工具

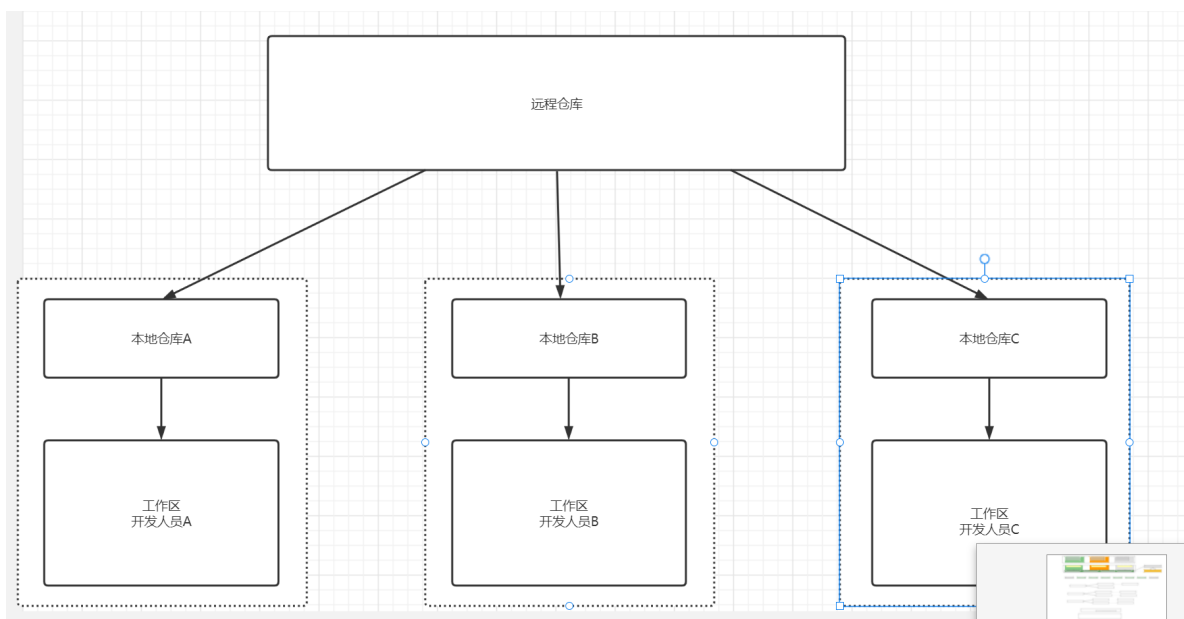
SVN。



1. 远程仓库管理所有代理，一旦远程仓库出现问题，那么导致大家都不能正常的使用了
2. 开发人员和 远程仓库之间直接通过网络连接，一旦网络出现问题，那么不可用

分布式的版本控制工具

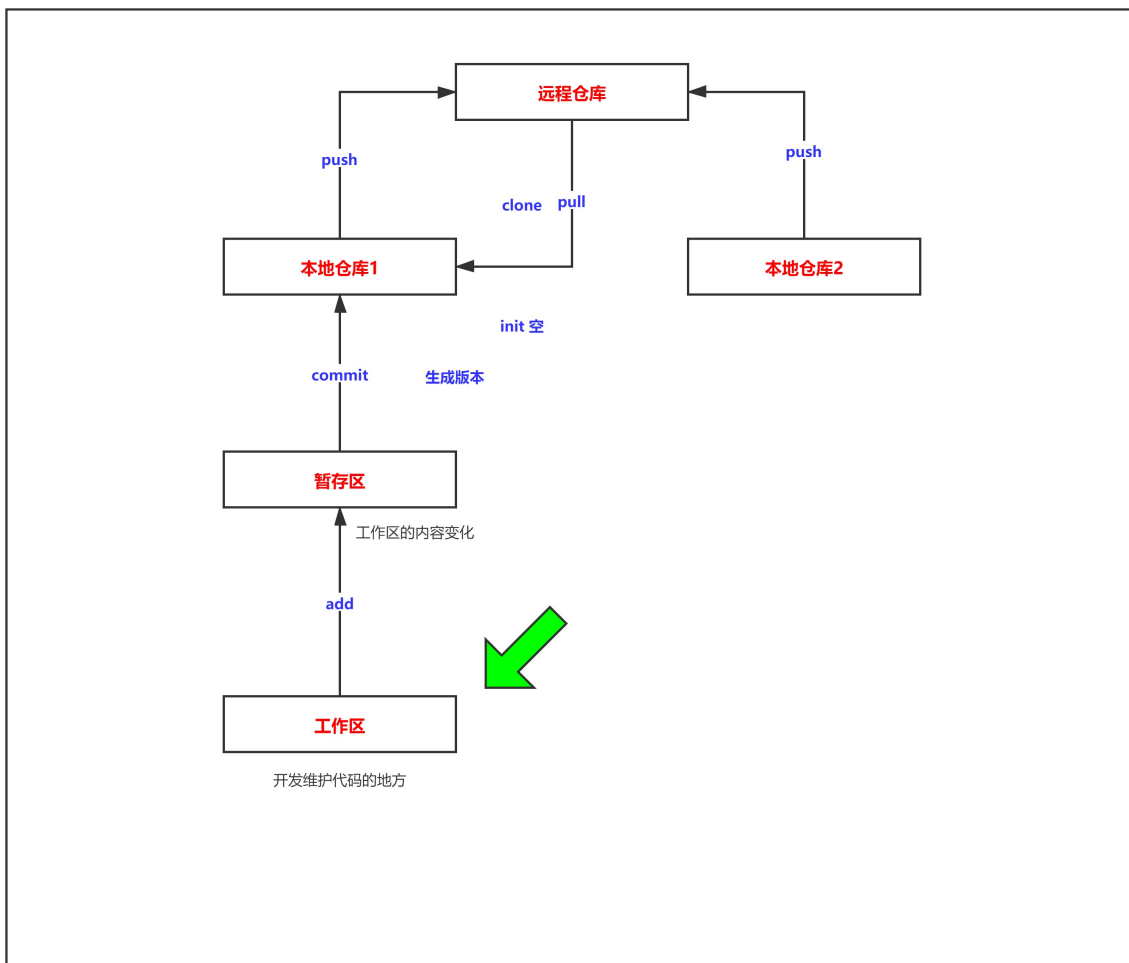
Git。



- 假如远程仓库不可用，没有关系，大家都可以正常使用

- 假如远程仓库和本地仓库之间网络出现故障，也没有关系，大家都可以在本地正常使用

3. Git核心流程



- 远程仓库
是指有一些网站提供的服务，可以帮助我们在远程建立一个仓库（也可以由公司自己来搭建）
www.github.com | www.gitee.com | www.gitlab.com（这些网站其实都是Gitlab这个开源项目的实例应用）
- 本地仓库
这个是本地的一个抽象的工作空间，其实对于我们用户来说，是看不见也摸不着的，由本地的Git来管理状态
- 暂存区
是在本地仓库和工作区之间提供的一个缓存区，给用户开发一个后悔的机会
- 工作区
其实是我们Git用户直接工作的区域，我们做的所有对文件直接的操作都是工作在工作区的

4. 安装Git

1. 下载

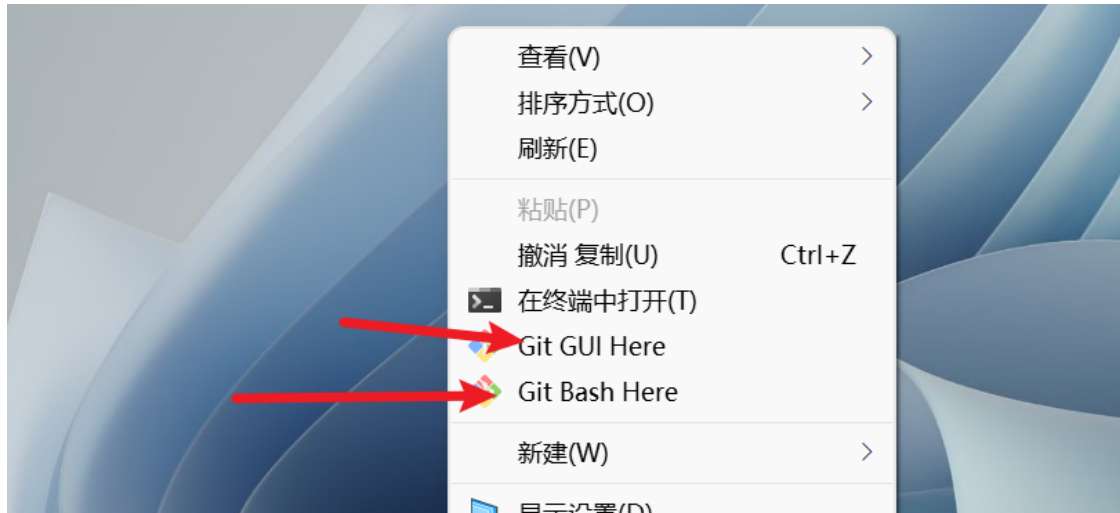
[官方下载地址](#)

软件 > git				在 git 中搜索	
名称	修改日期	类型	大小		
Git-2.23.0-64-bit.exe	2022/1/5 11:58	应用程序	46,584 KB		

2. 安装

傻瓜式的安装即可

如何验证自己是否已经安装成功呢？



5. 使用

5.1 建立远程仓库

- 去对应的网站注册账号

注册Gitee

记住：**用户名、密码、邮箱**

- 新建远程仓库



5.2 命令操作

clone

第一次把远程仓库中的内容拉取到本地，并且建立本地仓库。

第一次clone的时候，其实要输入对应网站的用户名和密码。

```
git clone 仓库地址
```

表示把远程仓库的内容拉取到指定的空的文件夹中，并且建立本地仓库

```
git clone 仓库地址 文件夹名字
```

add

把工作区的变化提交到暂存区中。

表示把指定的文件的变化保存到暂存区

```
git add fileName
```

把指定文件类型说产生的所有的变化保存到暂存区

```
git add *.txt
```

把某个文件夹中所有的变化保存到暂存区

```
git add dirName
```

把所有工作区中的变化保存到暂存区

```
git add .
```

status

可以查看工作区和暂存区产生的变化。

红色：工作区

绿色：暂存区

```
ciggar@ciggarBook MINGW64 ~/Desktop/test-44 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   runlist.txt  暂存区的变化是绿色的

Untracked files:
  (use "git add <file>..." to include in what will be committ
ed)
    swiminglist.txt  工作区的变化是红色的
```

commit

把暂存区中所有的变化提交到本地仓库。

注意：这一步会产生一个版本号，来记录本次的变化

第一次commit的时候，需要设置git的用户名和邮箱

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

这两步本质上其实就是去用户的家目录下面生成一个.gitconfig文件，在这个文件中保存用户的邮箱和用户名

```
git commit -m "提交的信息"
```

```
ciggar@ciggarBook MINGW64 ~/Desktop/test-44 (master)
$ git commit -m "第一次修改..."
[master 015d906] 第一次修改... 这一次提交的版本号
6 files changed, 1 insertion(+)
create mode 100644 User.java
create mode 100644 c3p0-config.xml
create mode 100644 cskaoan/1.txt
create mode 100644 cskaoan/2.txt
create mode 100644 runlist.txt
create mode 100644 swiminglist.txt
```

push

当本地仓库的版本领先于远程仓库的时候，可以把本地仓库中的变化，推送到远程仓库

```
git push
```

需要说明的是：commit和push其实都不能选择文件，是把所有的变化推送到远程仓库(或者提交到本地仓库)

```
ciggar@ciggarBook MINGW64 ~/Desktop/test-44 (master)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 539 bytes | 269.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-6.4]
To https://gitee.com/ciggar/test-44.git
91c33aa..015d906 master -> master
远程仓库的版本变化
旧版本... 新版本
```

pull

当本地仓库的版本落后于远程仓库的版本的时候，可以通过git pull拉取远程仓库中最新的变化

```
git pull
```

```
ciggar@ciggarBook MINGW64 ~/Desktop/cskaoyan (master)
$ git pull
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://gitee.com/ciggar/test-44
  91c33aa..015d906 master    -> origin/master
Updating 91c33aa..015d906
Fast-forward
 User.java      | 0
 c3p0-config.xml | 0
 cskaoyan/1.txt | 0
 cskaoyan/2.txt | 0
 runlist.txt    | 1 +
 swimminglist.txt | 0
6 files changed, 1 insertion(+)
create mode 100644 User.java
create mode 100644 c3p0-config.xml
create mode 100644 cskaoyan/1.txt
create mode 100644 cskaoyan/2.txt
create mode 100644 runlist.txt
create mode 100644 swimminglist.txt
```

log

查看本地仓库的版本信息

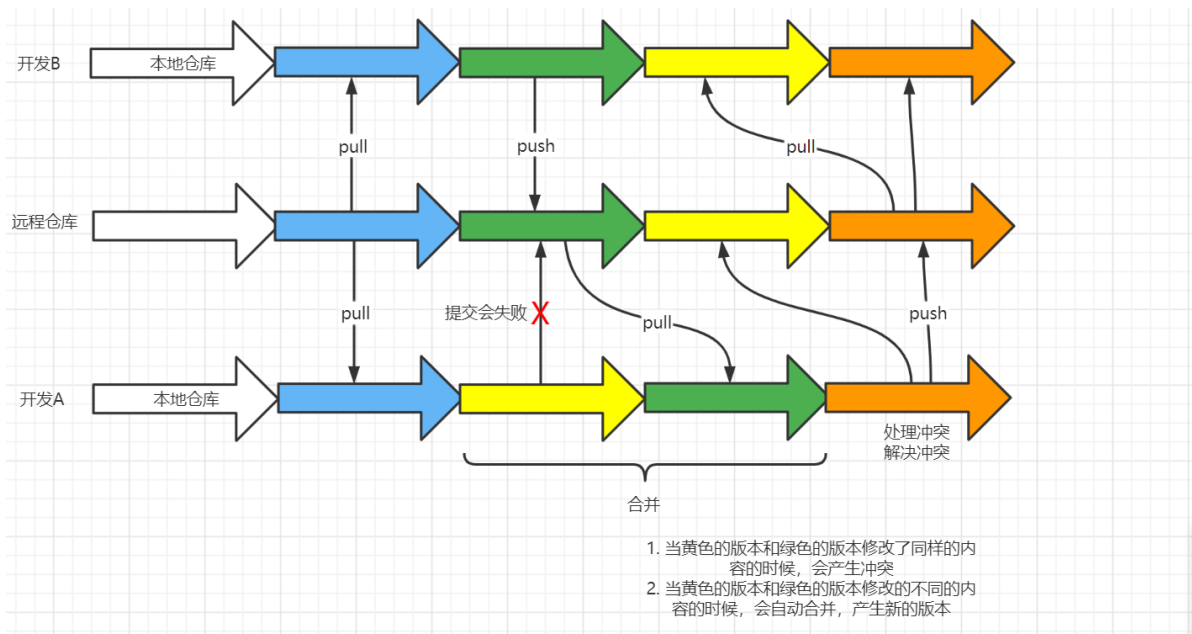
git log

6. 团队协作

6.1 添加仓库成员

The screenshot displays the Gitee repository management interface for the repository 'ciggar / test-44'. The left sidebar shows the '仓库成员管理' (Repository Member Management) section, which includes a list of roles: '所有' (All), '管理员' (Admins), and '开发者' (Developers). The '管理员' role is currently selected. The main content area shows the '管理员管理' (Admin Management) section, which lists the current administrators. There is one administrator listed: 'ciggar' (Repository Owner). A red arrow points from the '添加仓库成员' (Add Repository Member) button in the top right corner to the '管理员' section in the sidebar. Another red arrow points from the '管理员' section in the sidebar to the '我自己' (Myself) user in the list.

6.2 冲突问题

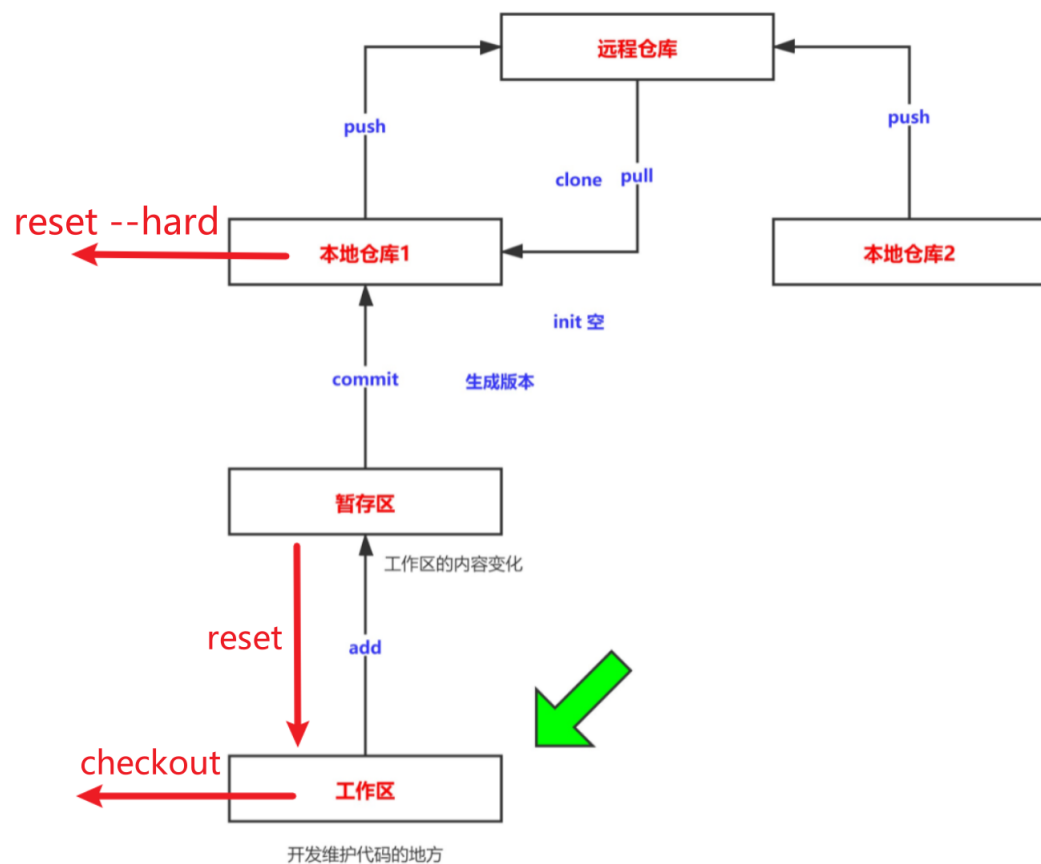


从这个冲突处理的过程中，我们能学到什么经验呢？

1. 慢提交的人解决冲突，我们可以在开发完代码之后，尽量早点提交
2. 及时的去同步代码（减少产生冲突的概率）

7. 后悔药

Git给我们提供了一些版本回退的机会。



- checkout

```
# 回退工作区的文件内容,相当于撤销
git checkout fileName
```

这个功能比较危险，因为撤销了之后无法恢复。

- reset

```
# 把暂存区的变化回退到工作区
# 和add命令一样，不会对文件中的内容产生影响，但是会改变文件变化的保存位置
git reset
```

- 本地仓库版本回退

```
# 把本地仓库中的代码回退到历史版本 | 比较危险，不要轻易操作
git reset --hard 版本号
```

8. 忽略文件

忽略文件其实就是让Git不去记录和追踪特定的文件产生的变化。

如何做呢？

1. 在仓库的根目录下生成一个 `.gitignore` 的文件
2. 在这个文件中，声明哪些文件或者是哪些格式的文件不被追踪

```
*.class
target
*.iml
*.tar
*.war
.idea
*.jar

# 配置文件的类型
# 配置文件的名称
# 配置文件夹的名称
```

假如一个文件曾经被本地仓库追踪了，那么再把这个文件加入到忽略文件中，Git是不会放弃它的追踪的，也就是忽略会失效，这个时候怎么做呢？

1. 在建立远程仓库的时候就声明好 `.gitignore` 这个文件
2. 把这个文件删掉，提交，然后再加入 `.gitignore` 这个文件