

Git

## Git是什么?

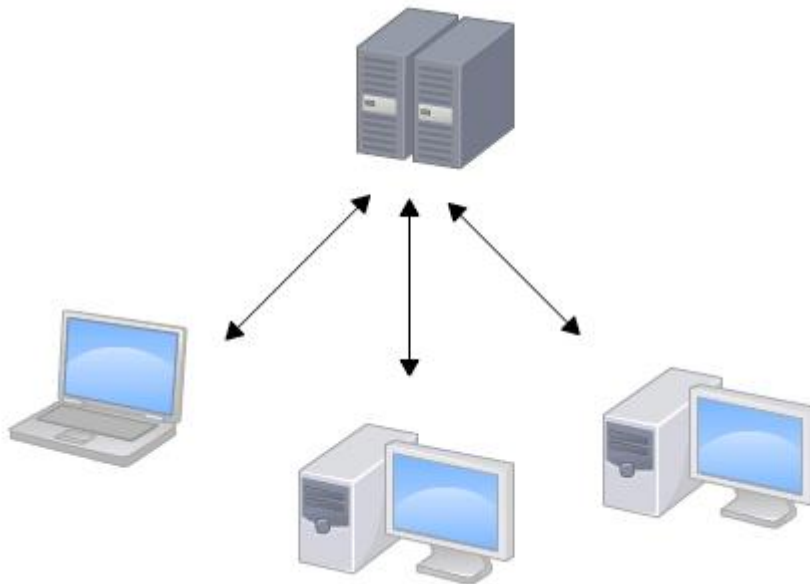
Git是分布式版本控制系统

## 版本控制系统的作用

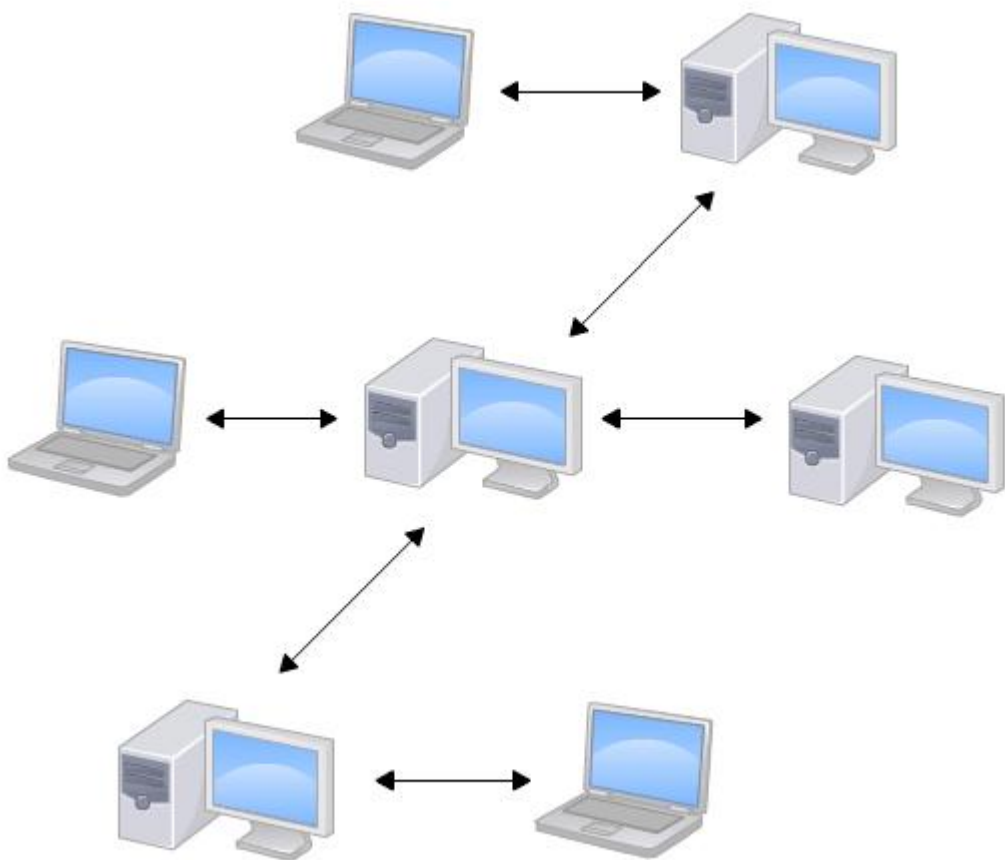
1. 保存修改记录,更方便的恢复文件状态
2. 方便协同开发

## 分布式 Vs 集中式

集中式



分布式



### 集中式版本控制系统的问题

1. 网络原因导致工作时代码的上传下载速度限制
2. 服务器损坏或宕机时,无法正常进行工作
3. 本地条件下无法完成代码回退的需求

## Git的安装使用

1. 从官方网站下载Git安装包
2. 傻瓜式安装
3. 声明git的用户的名字和邮箱  

```
$ git config --global user.name "Your Name"
```

```
$ git config --global user.email "email@example.com"
```

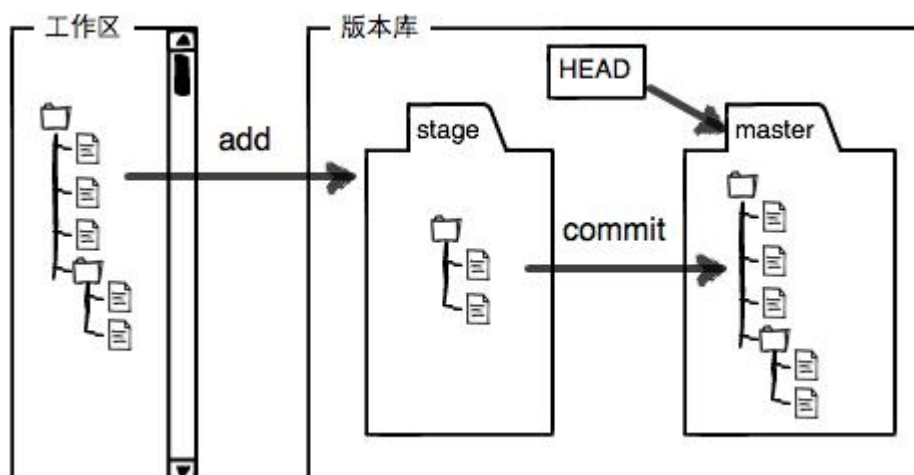
## Git的常用指令

1. `$ git init` 初始化版本库
2. `$ git add xxx` 将文件修改添加至暂存区
3. `$ git commit` | `$ git commit -m "xxx"` 将暂存区的所有内容提交至版本库
4. `$ git status` 查看版本库状态  
    红色: 未添加  
    绿色: 未提交  
    cleaning: 已提交

## Git的版本回退

1. `$ git log` 查看历史版本日志 展示详细的信息  
`$ git log --pretty=oneline` 展示精简版历史日志信息
2. `$ git reset --hard 版本号` 回溯某个版本 既可以向前 也可以向后
3. `$ git reflog` 查看所有版本日志信息 包括版本回溯的信息

## Git的工作区 | 版本库



## Git的撤销修改

1. `$ git checkout xxx` 撤销工作区的修改
2. `$ git reset head xxx` 撤销暂存区的修改

## Git的分支管理

1. `$ git checkout -b dev` 创建分支并切换至dev分支  
= `$ git branch dev` 创建分支 + `$ git checkout dev` 切换分支
  2. `$ git checkout master` 切换回主干分支 准备合并
  3. `$ git merge dev` 合并分支
  4. `$ git branch -d dev` 删除分支
- 当分支合并时有文件冲突的时候
1. `git merge dev` 合并分支
  2. `git status` 冲突的文件 是红色显示
  3. 多人协商 解决冲突
  4. `git add` | `git commit`
- 最终将冲突文件保存在版本库中

## Git的远程仓库

1. `$ git remote add origin URL` 与远程仓库建立连接
2. `$ git push (-u) origin master` 第一次推送需要使用-u参数 推送本地文件
3. `$ git pull origin master` 把远程库代码拉下来
4. `$ git remote remove origin` 接触与远程仓库的连接

远程仓库代码冲突问题

1. `$ git push origin master` 报错  
`// $ git push -f origin master` 强制推送
2. `$ git pull origin master` 把远程库代码拉下来
3. 手动在本地解决冲突问题 | `git add` | `git commit`
4. `$ git push origin master`

#### 5. SSH 免密推送

1. `$ ssh-keygen -t rsa -C "邮箱"` 生成公钥(敲回车 输出y)
2. 在C:\Users\Administrator\.ssh文件夹中找到id\_rsa.pub
3. 将id\_rsa.pub中的公钥配置在远程仓库的SSH公钥位置

#### 6. `$ git clone URL` 下载远程仓库至本地

## Git与Idea的集成

1. 在Setting中选择git的位置
2. 打开版本控制 VCS -- > Enable Version Control -- > git
3. VCS -- > Add | Commit
4. 在Setting中搜索Terminal 将cmd.exe 更换为 sh.exe
5. 使用命令的形式进行版本控制
6. 如需关闭版本控制 在.idea文件夹中删除vcs.xml