

Git



Git 是一个开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。

Git 是 Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

安装Git

```
1  # Redhat-Centos
2  $ yum install git
3  # Debian-Ubuntu
4  $ apt-get install git
```

配置Git

config: (计算机系统)配置

用户设置

```
1  $ git config --global user.name " "
2  $ git config --global user.email " "
```

编辑器设置

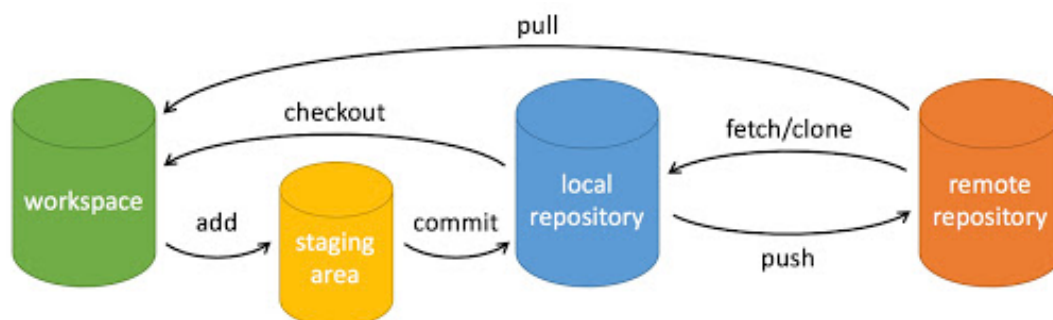
```
1  $ git config --global core.editor vim
```

查看配置信息

```
1 $ git config --list
```

概念

- 工作区 (**working dir**)：就是你在电脑里能看到的目录。
- 暂存区 (**stage area**)：英文叫 stage 或 index。一般存放在 **.git** 目录下的 index 文件 (**.git/index**) 中，所以我们把暂存区有时也叫作索引 (index)。
- 版本库 (**repository**)：工作区有一个隐藏目录 **.git**，这个不算工作区，而是 Git 的版本库。(repo)



```
1 $ git init
2 $ git add filename # --al与.
3 $ git commit -m "xxxx" # -m
4 $ git clone # 拷贝一份远程仓库，也就是下载一个项目。
```

- `git add .`：把当前目录及其子目录、子子目录.....中的异动全add到staging area，目录以外不执行。
- `git add --al`：不论在哪一层目录执行，效果一样，该项目中的所有异动均会被add到stagingarea。

```
1 $ git log
2 $ git rm
3 $ git mv
4 $ git blame filename
```

远程

```
1 $ git remote # 远程仓库操作
2 $ git fetch # 从远程获得代码库
3 $ git pull # 下载远程代码并合并
4 $ git push # 上传远程代码并合并
```

分支管理：branch

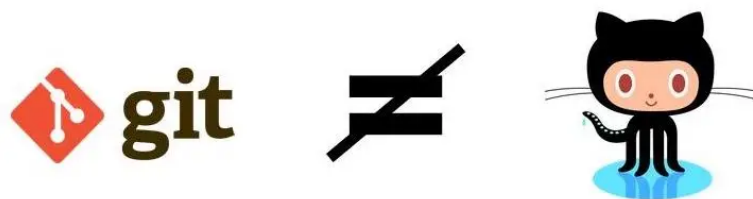
每一种版本控制系统都以某种形式支持分支。使用分支意味着你可以从开发主线上分离开来，然后在不影响主线工作的同时继续工作。

GitHub远程

Git是一款免费、开源的[分布式版本控制系统](#)

Github是用Git做版本控制的代码托管平台：本质是一个大的Git仓库

百度百科上的定义是：面向开源以及私有软件项目的代码托管平台，只支持git 作为唯一的版本库格式进行托管，故名GitHub。



```
1 | git remote add [shortname] [url] # 添加远程仓库
```

本地 Git 仓库和 GitHub 仓库之间的传输是通过SSH加密的，所以我们需要配置验证信息：

SSH： [Secure Shell](#)（外壳协议）

专为[远程登录](#)会话和其他网络服务提供安全性的协议。

SSH提供两种级别的安全验证：

- 第一种级别（基于口令的安全验证）
- 第二种级别（基于密匙的安全验证）*

```
1 | $ ssh-keygen -t rsa -C "youremail@example.com" # 配置验证信息，邮箱是你注册GitHub的邮箱
2 | $ ssh -T git@github.com # 验证
```

Gitee