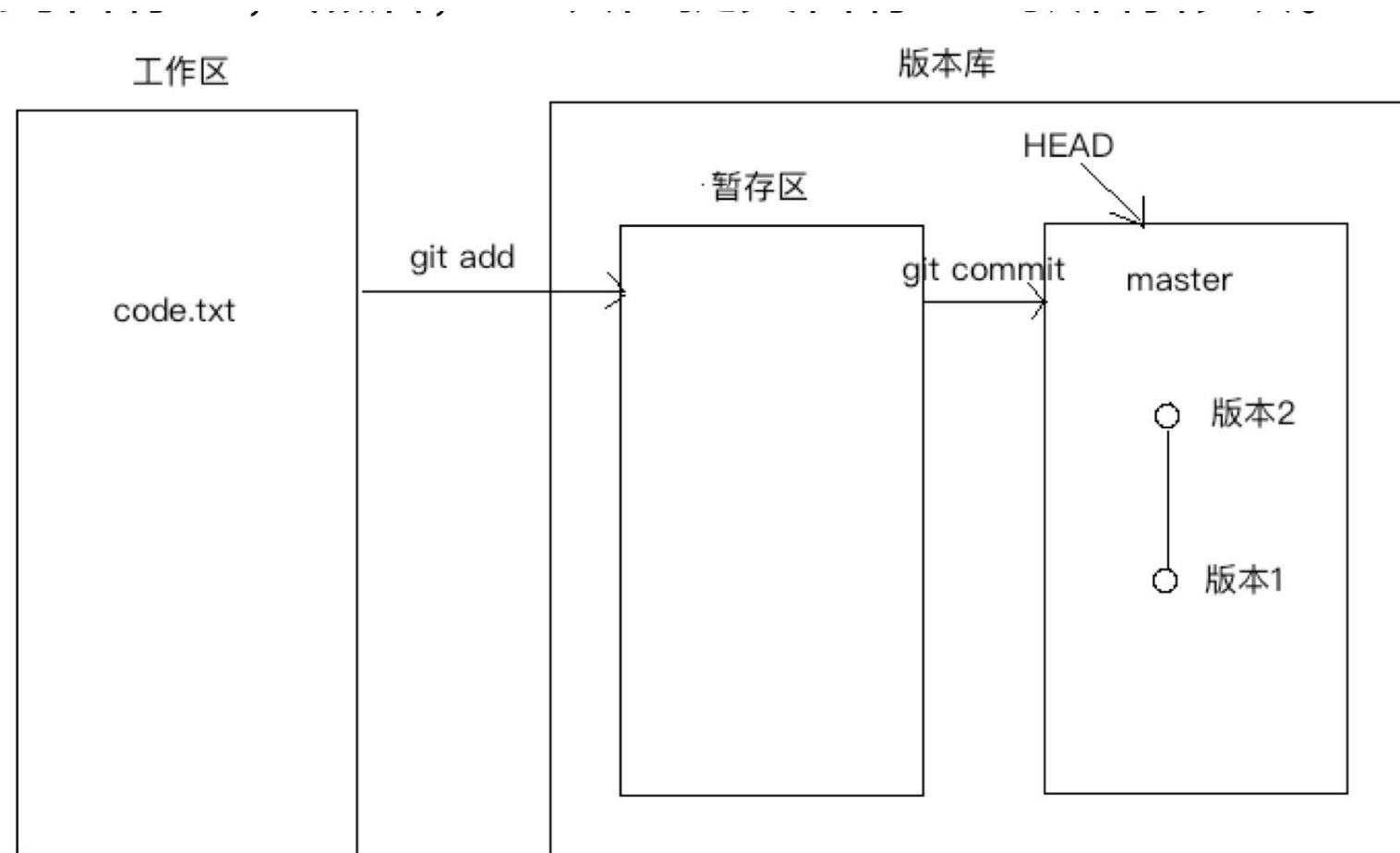


Git用法简介

特点

- 版本控制：可以解决多人同时开发的代码问题，也可以解决找回历史代码的问题。
- 分布式：Git是分布式版本控制系统，同一个Git仓库，可以分布到不同的机器上。首先找一台电脑充当服务器的角色，每天24小时开机，其他每个人都从这个“服务器”仓库克隆一份到自己的电脑上，并且各自把各自的提交推送到服务器仓库里，也从服务器仓库中拉取别人的提交。可以自己搭建这台服务器，也可以使用GitHub网站。



git init : 初始化

git add : 添加暂存区

git commit -m '版本' : 当前内容提交到当前分支 (默认的分主master)

git log : 查看版本记录

git log --pretty=oneline : 查看版本记录简短版

git reflog : 查看以前操作记录

git reset -- hard HEAD^ : 当前版本的前一个版本

HEAD~1: 当前版本的前一个版本

HEAD^^: 当前版本的前两个版本

git reset -- hard : 版本序号的前几位

git status : 查看当前状态

撤销修改：

- 在工作区中： `git checkout -- 文件`
- 在缓存区中： `git reset HEAD 文件`
`git checkout -- 文件`

比较不同：

- `git diff HEAD -- 文件`：-代表HEAD版本中文件内容，+代表工作区中文件内容
- `git diff HEAD HEAD^ -- 文件`

删除：

`rm`

`git rm`

恢复：

`git checkout -- 文件`

- 分支

git branch : 查看当前有几个分支

git branch -r : 查看远程分支

git branch -a : 列出本地分支和远程分支

git checkout -b 分支名 : 创建新分支并在上面工作

git checkout master : 切换到master分支

git merge 分支名 : 合并指定分支到当前分支

git branch -d 分支名 : 合并后删除

git merge --no-ff-m '说明信息' 分支名 : 禁用快速合并

git stash : 保存工作现场

git stash list : 列出

git stash pop : 回到现场

- Github

绑定github用户: emacs .gitconfig

检查是否有SSH key: cd ~/.ssh

生成SSH key : ssh-keygen -t rsa -C “注册邮箱”

将SSH key 添加到GitHub

检测是否连接成功 ssh -T git@github.com

- 克隆

git clone +SSH

- 上传代码

- ① 进入要上传的目录，变成git管理的仓库 `git init`
- ② 将文件添加到仓库，并查看状态 `git add ; git status`
- ③ 提交到本地仓库 `git commit -m '描述'`
- ④ 关联远程仓库

`git remote add origin 关联仓库的SSH`

- ⑤ 远程仓库有内容

Linux 中主分支master，githubz中为main，为了使其相互对应，将master改名成main

`git branch -m main`

`git pull origin main`

报错： `git pull origin main --allow-unrelated-history`

`git push origin main`