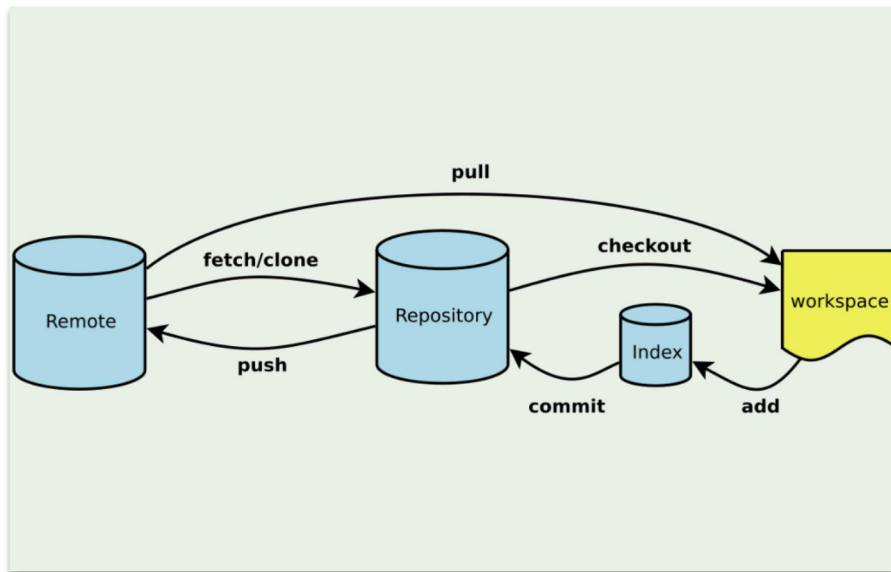


- 区分工作区、暂存区、本地仓库、远程仓库



首先我们平时修改代码都是在工作区即 workspace，通过执行 add 指令，即将代码提交到了暂存区即 index，通过 commit 指令又将暂存区的代码提交到了本地仓库 repository 中，再通过 pr 即提交到了 remote 中。

- Git 常见指令

#### 初始配置自己的用户名和邮箱

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

#### 创建删除目录

```
mkdir gitlearn //创建一个新目录
cd gitlearn//进入到该目录下
cd ../ // 切回到上一个工作路径
cd /// 进入根目录
rm 文件名//删除文件
cat 文件名//打开文件并读取出文件中的内容
```

#### 查看文件信息

```
ls // 查看当前路径下面的所有文件名
ls 文件夹名 // 查看对应文件夹中的内容
ls -l // 拉出最近 git 提交记录以及对应修改的文件名
```

ls -l -a // 拉出最近 git 提交记录以及对应修改的文件名，隐藏的文件也会显示

## 查看状态

git status

## 提交

git add 文件名 // 将某个文件存入暂存区

git add b c //把 b 和 c 存入暂存区

git add . // 将所有文件提交到暂存区

git commit -m "提交的备注信息" // 提交到仓库

## 初始化并在工作路径上创建主分支

git init

## 对比差异

git diff //工作区与暂存区的差异比对

git diff --cached // 暂存区与提交版本的差异

git diff HEAD // 工作区与仓库中最后一次提交版本的差别

## 分支操作

git branch 分支名 // 新建分支

git branch // 查看当前所有分支

git checkout 分支名 // 检出分支

git checkout -b 分支名 // 创建并切换分支

git branch -v // 查看分支以及提交 hash 值和 commit 信息

git merge 分支名 // 把该分支的内容合并到现有分支上

git branch -d 分支名 // 删除分支

git branch -D 分支名 // 强制删除 若没有其他分支合并就删除 d 会提示 D 不会

git branch -m 旧分支名 新分支名 // 修改分支名

git branch -M 旧分支名 新分支名 // 修改分支名 M 强制修改 若与其他分支有冲突也会创建

git branch -r // 列出远程分支(远程所有分支名)

git branch -a // 查看远程分支(列出远程分支以及本地分支名)

git fetch // 更新 remote 索引

### 回撤操作

git reset --soft HEAD^ //^为最近一次 ^2 为上上次 HEAD 可以变为指定版本号

#### 参数

--mixed 不删除工作空间改动代码，撤销 commit，并且撤销 git add .

--soft 不删除工作空间改动代码，撤销 commit，不撤销 git add .

--hard 删除工作空间改动代码 工作区回退到最近一次 commit 状态

### 日志

git log//查看提交历史记录

git log --oneline//或者 git log --pretty=oneline 以精简模式显示

### 远程仓库

git remote -v //查看远程仓库

git fetch origin master //从远程拉取分支

git pull origin master //拉取远程分支并且合并