

# git2

□

- **workspace**: 本地的工作目录。（记作A）
- **index**: 缓存区域，临时保存本地改动。（记作B）
- **local repository**: 本地仓库，只想最后一次提交HEAD。（记作C）
- **remote repository**: 远程仓库。（记作D）

## 初始化

- `git init` //创建
- `git clone /path/to/repository` //检出
- `git config --global user.email "you@example.com"` //配置email
- `git config --global user.name "Name"` //配置用户名

## 撤销操作

- `git reset <file>` //某个文件索引会回滚到最后一次提交， $C \rightarrow B$
- `git reset` //索引会回滚到最后一次提交， $C \rightarrow B$
- `git reset --hard` // 索引会回滚到最后一次提交， $C \rightarrow B \rightarrow A$
- `git checkout` // 从index复制到workspace， $B \rightarrow A$
- `git checkout -- files` // 文件从index复制到workspace， $B \rightarrow A$
- `git checkout HEAD -- files` // 文件从local repository复制到workspace， $C \rightarrow A$

## 分支相关

- `git checkout -b branch_name` //创建名叫“branch\_name”的分支，并切换过去
- `git checkout master` //切换回主分支
- `git branch -d branch_name` // 删除名叫“branch\_name”的分支
- `git push origin branch_name` //推送分支到远端仓库
- `git merge branch_name` // 合并分支branch\_name到当前分支(如master)
- `git rebase` //衍合，线性化的自动， $D \rightarrow A$

## 冲突处理

- `git diff` //对比workspace与index
- `git diff HEAD` //对于workspace与最后一次commit
- `git diff <source_branch> <target_branch>` //对比差异
- `git add <filename>` //修改完冲突，需要add以标记合并成功

## 其他

- `gitk` //开灯图形化git
- `git config color.ui true` //彩色的 `git` 输出
- `git config format.pretty oneline` //显示历史记录时，每个提交的信息只显示一行
- `git add -i` //交互式添加文件到暂存区