

定义：
Git是一个分布式的VCS(Version Control System 版本控制系统).

场景模型：

想象这么一个场景,你是一位作家,你有一张工作台,上面乱七八糟地堆着一堆稿件.而工作台上有一个盒子,你觉得不错的稿件你会丢到盒子里面暂时保存起来,你还有一个书柜,你觉得写的不错的文章会把他们收集起来放到书柜里面做成样书.最后你觉得写的非常好的书会拿给远方的出版商出版.

1.几个关键的场景:1.工作台 2.盒子 3.书柜. 4.出版商 很显然,里面稿件的价值 工作台 < 盒子 < 书柜 < 出版商.

2.这个过程是严格一步步走下去的,不能跳任何一步,你希望把工作台上的稿件直接放到书柜里是不可能的,必须先放到盒子里面才能放到书柜里面

3.一旦进入了书柜,书柜里面每本书就有个长度为40的一个sha1字符串作为书的编号,保证唯一.每本书对应我们下面所说的一个commit(节点)

工作流程:		
现在我们开始工作.		
	下行命令	上行命令
工作台	git add	git checkout + 路径
盒子	git commit	git reset + 路径
书柜	git push	git reset + commit(节点)
出版商	—	git fetch

解释以下几个命令:

git add 把临时修改的文件添加到暂存区(工作台 -> 盒子)

git checkout + 路径 丢弃路径下所有的临时修改(丢弃稿件)

git commit 把暂存区里的修改做本地提交(盒子 -> 书柜)

git reset + 路径 把暂存区里面的修改变回临时修改(盒子 -> 工作台)

git reset + 节点 (书柜 -> 盒子)

git reset —hard 这个命令必须要说一下是因为这个命令比较暴力,等于你把工作台上所有的东西

git push 把本地的修改推到远程服务器(书柜 -> 出版商)

git fetch 从出版商获得最新的书籍列表(出版商 -> 书柜)

中间特别注意的是

理解了之后有兴趣再去了解一下

git clean

git stash即可

分支管理:

git merge

git rebase

git cherry-pick

仓库创建

git init 在本地目录下创建git仓库,会在当前目录下创建一个隐藏的.git文件夹

git clone + URL 从远端拉一个仓库到当前目录下

git submodule 子仓库管理,具体请查阅手册.

配置

.git/config 这个文件里面存放着基本信息,比如远端仓库地址等

.git/.gitignore 忽略文件列表,满足.gitignore里面格式的文件都会被忽略

git config —global 配置自己提交的用户名和邮箱