Git

2020年11月3日 22:53

Git是个什么玩意儿?

分布式版本控制系统

通过git管理GitHub托管项目代码

★ *命令行模式*

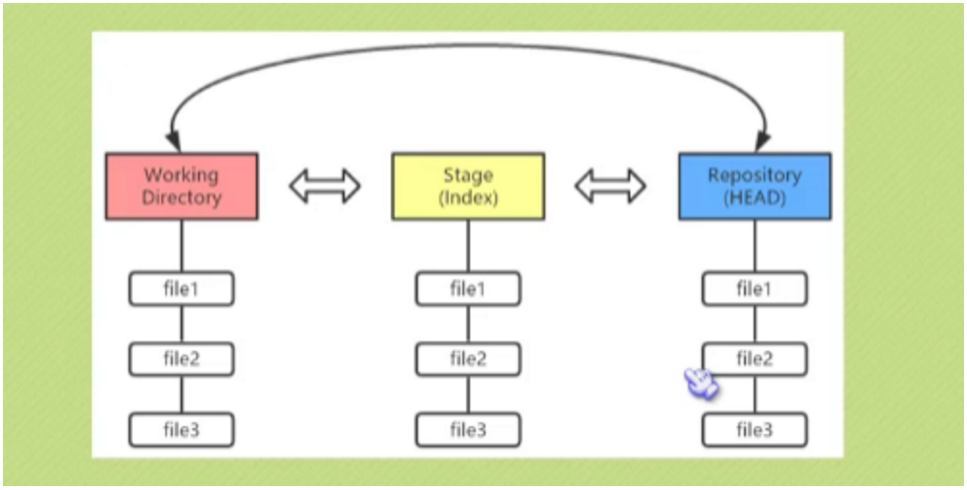
理论基础

Git工作区域:

（1）Git Repository（Git仓库）：最终确定的文件保存到仓库，成为一个新的版本，并且对他人可见

（2）暂存区：暂存已经修改的文件，最后统一提交到git仓库中

（3）工作区（working directory）：添加、编辑、修改文件等动作



Git工作流程

1.在工作目录中添加、修改文件

2.将需要进行版本管理的文件放入暂存区域

3.将暂存区域的文件提交到git仓库

Git管理的文件有三种状态

--已修改（modified）

--已暂存（staged）

--已提交（committed）

实际操作

git status查看文件状态

通过命令或图形界面将工作区中的文件提交到暂存区 git add +filename

暂存区到仓库 git commit -m "提交描述"

git初始化及仓库创建和操作

基本信息设置

1.设置用户名 git config --global user.name 'itcast'

2.设置用户名邮箱 git config --global user.name 'itcast@itcast.com'

初始化一个新的git仓库

1.创建文件夹

2.在文件内初始化git（创建git仓库）：进入文件夹内，git init初始化仓库

向仓库中添加文件

1.创建文件：（1）命令行中vim/touch命令创建（2）在仓库中右击创建

2.git add 命令将文件提交到暂存区

3.暂存区文件添加到仓库 git commit -m "提交描述"

修改仓库文件

使用vi或其他方式修改了已经提交到仓库的文件，使用git status命令可以查看修改提示

再使用git add和git commit -m命令提交到仓库

删除仓库文件

rm -rf filename

1.删除文件 rm test.txt

2.从git中删除文件 git rm test.txt

3.提交操作 git commit -m "提交描述"

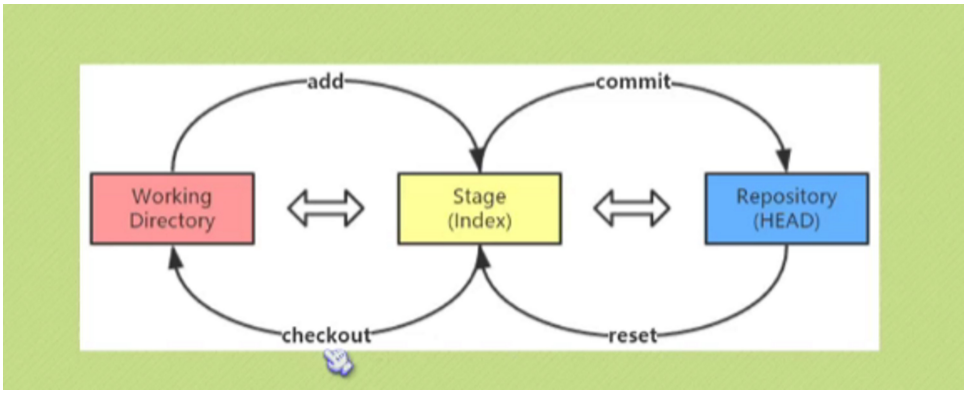
查看工作状态和历史提交

git status

将文件提交到本地仓库后，若重新修改了文件，使用 git checkout -- <file>可以将暂存区中的文件覆盖工作目录中修改的文件

git log 查看历史提交

回到过去



git reset --mixed HEAD~

-移动HEAD的指向，将其指向上一个快照

-将HEAD移动后指向的快照回滚到暂存区域

git reset --soft HEAD~

-移动HEAD的指向，将其指向上一个快照 soft相当于撤销commit

git reset --hard HEAD~

-移动HEAD的指向，将其指向上一个快照

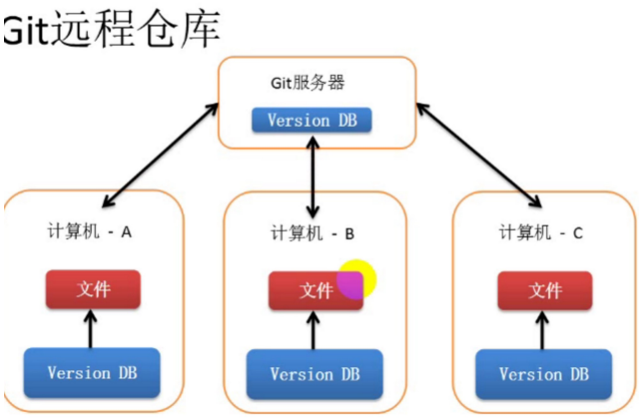
-将HEAD移动后指向的快照回滚到暂存区域
-将暂存区域的文件还原到工作目录
reset相当于抵消了add+commit
回滚个别文件 git reset 版本快照 文件名/路径
git reset 版本快照的ID号
git reflog 查看历史所有的commit id
版本对比
git diff 比较两个版本文件的不同 （哔哩哔哩小甲鱼git P5）
修改最后一次提交、删除文件和重命名文件

创建和切换分支

合并和删除分支

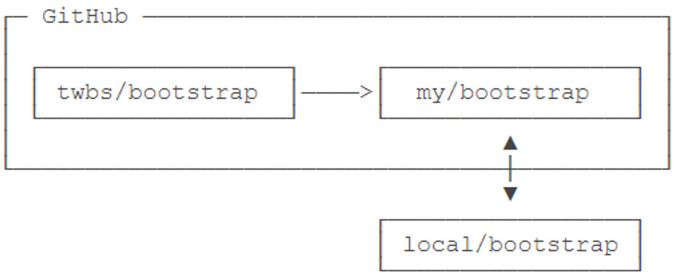
匿名分支和checkout命令

Git管理远程仓库
使用远程仓库的目的
作用：备份、实现代码共享集中化管理



将本地仓库同步到git远程仓库中
git push

Git 克隆操作
目的：将远程仓库（GitHub对应的项目）先用fork克隆到自己的账号中，
使用 "git clone git@github.com:username/仓库名" 将仓库复制到本地，只有从自己的账号下clone仓库，这样才能推送修改，如果直接使用 "git clone 仓库地址"，因为没有权限，将不能推送修改。



关联远程库 git remote add origin git@github.com:username/仓库名.git
添加后，远程库的名字就是 origin
本地库内容推送到远程库上 git push -u origin master
第一次推送，加 -u 参数，之后使用 git push origin master 把本地master分支的最新修改推送至Github。