

git初始命令

diff + hash / log

log  
git log  
git log --oneline  
git relog (包括reset)

diff

显示的差异是按照文件目录将其差异内容表示出来

总结				
git diff	工作区	VS	暂存区	
git diff HEAD	工作区	+	暂存区	VS 本地仓库
git diff --cached / git diff --staged			暂存区	VS 本地仓库
git diff <commit_hash> <commit_hash> / git diff HEAD~ HEAD	比较提交之间的差异			
git diff <branch_name> <branch_name> /	比较分支之间的差异			

git diff HEAD hash (最新版本和某一版本)  
git diff HEAD HEAD~ (最新版本和最新前1个版本)  
git diff HEAD~3 HEAD (最新前3个版本和最新版本)  
git diff HEAD~3 HEAD "file\_name"(file\_name的最新前3个版本和最新版本)

soft  
hard  
mixed

都可以通过 reflog , git reset hard hash撤销

git reset的三种模式		
git reset --soft	工作区	暂存区
git reset --hard	工作区	暂存区
git reset --mixed	工作区	暂存区

reset回退的是commit命令,也就是Head指针

reset

本地仓库



本地仓库文件状态



delete

- rm file\_name ; (只删除了working)  
git commit -am "..."; (覆盖)
- git rm file\_name ; (同时删除working+staging)  
git commit -m "..."; (覆盖)

公钥+密钥

- 公钥 (本地) : 可以创建多个
- 私钥 (github) : 可以创建多个

conflict

merge  
修改文件内容;  
git diff 查看区别;  
vi 修改, 再add, 再commit。  
合并时想中止merge: git merge --abort

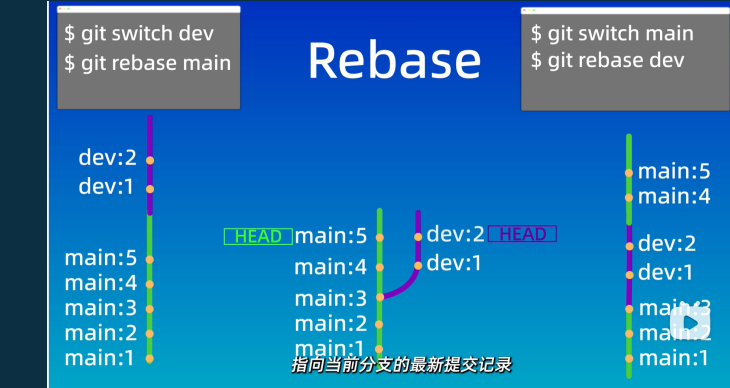
2

在本地仓库添加remote add  
本地仓库连接多个github仓库 (远程仓库) : 更改远程名称origin  
git remote add origin ssh  
git branch -M main  
git push -u(upstream) origin main:main(本地main: 远程main)  
git pull origin main

1

github仓库clone到本地

本地仓库+远程仓库连接



2

git rebase "branch\_name"  
无论在main还是branch操作均可, 最后形成在main上的线性提交列表

1

git merge "branch\_name"  
此时处于main分支操作

git branch (查看分支)  
git switch -c "branch\_name" (创建分支)  
git switch "branch\_name" (切换分支)  
git branch -d "branch\_name" (删除branch, 被merge后)  
git branch -D "branch\_name" (强制删除, 未被merge也行)

branch