git使用

2020年3月1日

13:15

## 工作区

缓存区

本地库

## 

## 1.创建本地库

mkdir git\_work

mkdir scala

## 2.本地仓库初始化

git init

## 3.设置签名（表明开发人员身份）

git config --global user.name wangjun

git config --global user.email 1559924775@qq.com

## 4.将工作区的文件上传到本地库

上传到暂存区——》提交到本地库

git add for循环.pdf //存入暂存区



git commit -m "首次提交" for循环.pdf

### 当修改工作区文件内容是其和本地库不一样时：

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed) 修改本地库（提交到本地库）——以工作区为主

(use "git checkout -- <file>..." to discard changes in working directory) 工作区改回去——以本地库为主

checkout演示：

计算机生成了可选文字:
/F/Desktop/git—work/scala(master)
15599@LAPTOP2284047H、仃NG64
mvforfi环.pdfsalca—for.pdf
/F/Desktop/git—work/scala(master)
15599@LAPTOP2284047H、仃NG64
gitcheckoutforfi环.pdf
Updated1pathfromtheindex
/F/Desktop/git—work/scala(master)
15599@LAPTOP2284047H、仃NG64
total944
1155991976094825607月710：49for彳盾环.pdf
1155991976094825607月710：46salca_for.pdf
/F/Desktop/git—work/scala(master)
15599@LAPTOP2284047H、仃NG64

checkout注意：

当工作区和缓存区一样（add），而和本地库不一样时。

先要使用git reset HEAD 使缓存区和本地库一致。

所以checkout的作用其实是使工作区变的和缓存区一致。

### 缓存区和本地库不一致：

修改工作区文件后，使用add命令使得缓存区和本地库不一致

通过git reset HEAD 文件 将缓存区的修改取消

这时工作区和缓存区、本地库不一样：

计算机生成了可选文字:
/F/Desktop/git—work/scala(master)
15599@LAPTOP2284047H、仃NG64
$gitstatus
Onbranchmaster
)changesnotstagedforcommit:
(use'tadd<饷]e>．
toupdatewhatwil]becommitted)
(use'tcheckout__<fi]e>．
modified:
he]]0．txt
nochangesaddedtocommit(use
todiscardchangesnworkingdirectory)
gitadd"and/or"gitcommit—a")

红色是工作区的提示

这时使用git checkout可以将工作区修改的和本地库一致

修改后提交到缓存区，提示是绿色的：

计算机生成了可选文字:
15599@LAPTOP2284047H
$vimhello.txt
15599@LAPTOP2284047H
$gitaddhello.txt
15599@LAPTOP2284047H
$gtistatus
/F/Desktop/git—work/scala(master)
、仃NG64
/F/Desktop/git—work/scala(master)
、仃NG64
/F/Desktop/git—work/scala(master)
、仃NG64
bash:gti：commandnotfound
/F/Desktop/git—work/scala(master)
15599@LAPTOP2284047H、仃NG64
$gitstatus
Onbranchmaster
Changestobecommitted:
(use'gitresetHEAD<fi]e>．
tounstage)
modied。
hello.txt

计算机生成了可选文字:
工作区
eh.ekout-<fil.>
r000tE磊0

## git log:

计算机生成了可选文字:
镘USER一20150323E/glt—work/d•o(master)
5git10g
匚。“1t4586b〔7022f匚Odfb8bde501dbb0226匚66132d(HEAD”master)
Author·
·weixin—global孓315759265@qq·（om
Date:
TueNOV1914：38：26201g+0800
（001tfe（827bb557ae44a3be（d615e3263b3d2792
Author：e气n一global315艿g26弓@qq．co的、
Date：
TueNOV1914：24：252019十0800
第二次提交
gb28858g仁aeb38e8856a43adf177b亡313f7f3f2
0001t
weixin_global《31575g2b5@qq．匚00>
TueNOV1914：19：092019十0300
甘次生交
4貊nistratorUSER一2018082彐乬VI、GW€-4/d/git—work/demo〔master)
Sgit]og--pretty—onel，ne
《500：0220=“05匕匕0：=00
D->master)次生交
C82了bb55《ae44已3b已匚0515b匕3253ba3匕2《92苤二老扌思《
9b288E8g匚deb38e3356d43dd4f177b匚313f7f3f2曾扌生交

git log --oneline 可以更短 ——通过这个命令是文件恢复到之前的某个版本

计算机生成了可选文字:
5git10g——oneline
453b3db0还AD”。盈st七第．三辶《提交
fecS27b=次交
gb288E8甘次交
Ad'巧nistratorUSER一20180823EYIVG•"€4/d/git—work/demo〔master)
5cathello-txt
、nistrator@USER．2013082氵EYIVGJa•64/d/git—work/deno(master)
§gifreset--hardfec827b
HEAD0nowatfe匚827b二次提交
，nistrator&USER-201SOS23EWY是长64/d/git—work/deno(master)
git10g．．onel，ne
亡827b0焱0”master)苤=次交
gb288E8甘次交

第二次提交的时候没有ccccc

### 所以本地库中记录了所有的版本，通过reset可以将工作区的文件恢复到任意版本。

### git reset --hard 版本哈希号

如果a.txt版本一 add和commit提交后，

a.txt修改为版本二，add到缓存区

a.txt修改为版本三，工作区

这时本地库中只有a.txt的版本一，如果要恢复版本git reset --hard HEAD会使工作区恢复到版本一，这时缓存取也会同步（可使用diff验证）

计算机生成了可选文字:
四3．3．永久删除后找回“
gitrm命令用于从工作区和索引中删除文件．“
1、工作区创建又件，通过命令提交到本地库“
Vimdl.txtgitadddl•txtgitcommit—m'上传到本地库dl.txt'dl.txte
执行命令进行删除：
2、
gitrmdl.txt兮工作区删除后，提交到本地除d1慧t'd1．t难」
通过a,±gitreflog找回“
$gitreset
——prd15d52“
11查看即可。“
4、
注意：git只要在本地库发生操作，都会进行版本记录。

通过rm -rf 删除也可以通过reset恢复版本

计算机生成了可选文字:
四3．4．文件比较
gitdiff命令用于显示提交和工作树等之间的更改。此命令比较的是工作目录
中当前文件和暂存区域快照之间的差异，也就是修改之后还没有暂存起来的变化
内容。（默认星和暂存区文件讲行比较
1、示例！giddiff将工作区中的文件和暂存区讲行比较，」
vimgit.txt修改内容“
gitdiffgit．txte
00k'•eixin%ESKiUP-•CQ3943H、以\辶/d/〕ava/g1tworkSpace/WeChat(master)
5Vimgit.txt
uoweixin@DESKTOP-FQ3943HMINGW64/d/java/git/Gitworkspace/wechat(master)
S、tdiff巧t、txt
．．=嵊7币气．'b/t．txt
、ndex3230530．．b3（43（e100644
a/git.txt
++b/git.txt
一1，3+1，3@@
he110git
2githahaha

如果版本不一致，使用add或checkout恢复

git diff HEAD git.txt 比较本地库和缓存区

git diff git.txt 比较工作区和缓存区

主要是想知道缓存区的数据现在是什么样的。

## 本地库删除文件：

先在工作区删除文件

在使用git commit -m "info" 就可以删除本地库了

计算机生成了可选文字:
[5599@LAPTOP2284047H
gitaddhello.txt
[5599@LAPTOP2284047H
gitaddt．txt
[5599@LAPTOP2284047H
/F/Desktop/git—work/scala(master)
、仃NG64
/F/Desktop/git—work/scala(master)
、仃NG64
/F/Desktop/git—work/scala(master)
、仃NG64
gitcommit—mnfo》
[master3e6df9c]info
2fileschanged,4deletions(-)
deletemode100644he]]0．txt
deletemode100644t．txt
/F/Desktop/git—work/scala(master)
[5599@LAPTOP2284047H、仃NG到64
gitpush一uoriginmaster

## 

## 文件上传到远程库：

先给远程库起个别名：

