

- 目录
- Git教程
- Git简介
- 安装Git
- 创建版本库
- 时光机穿梭
- 版本回退
- 工作区和暂存区
- 管理修改
- 撤销修改
- 删除文件
- 远程仓库
- 分支管理
- 标签管理
- 使用GitHub
- 使用码云
- 自定义Git
- 期末总结

关于作者

## 工作区和暂存区

阅读: 791818

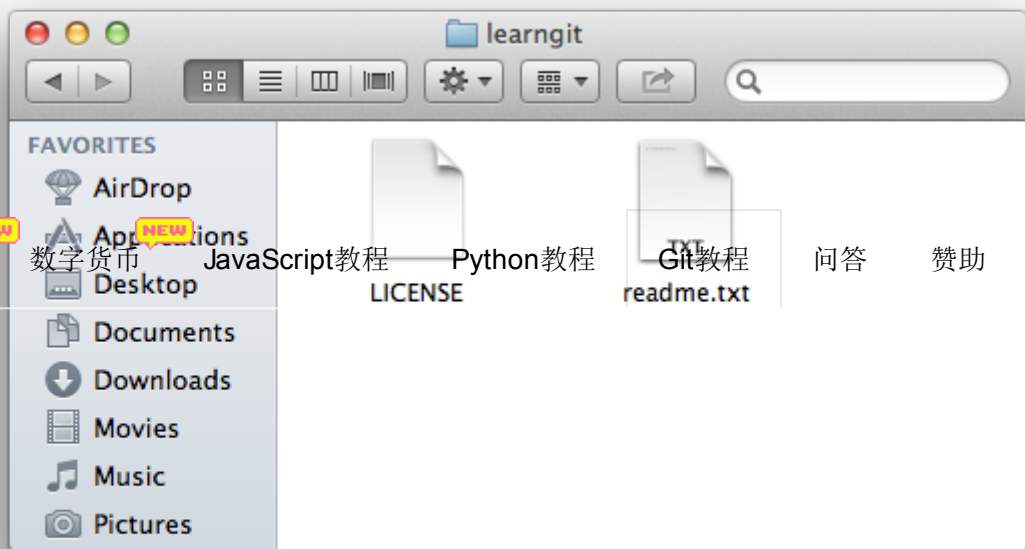


Git和其他版本控制系统如SVN的一个不同之处就是有暂存区的概念。

先来看名词解释。

### 工作区（Working Directory）

就是你在电脑里能看到的目录，比如我的 `learngit` 文件夹就是一个工作区：



### 版本库（Repository）

登录



廖雪峰 北京 朝阳区

加关注

廖雪峰老师

自己的Python课程

Python商业爬虫全解码

让天下没有爬不到的数据！

Python爬虫

+

数据分析



编程

读书

JavaSE教程

Python机器学习

+

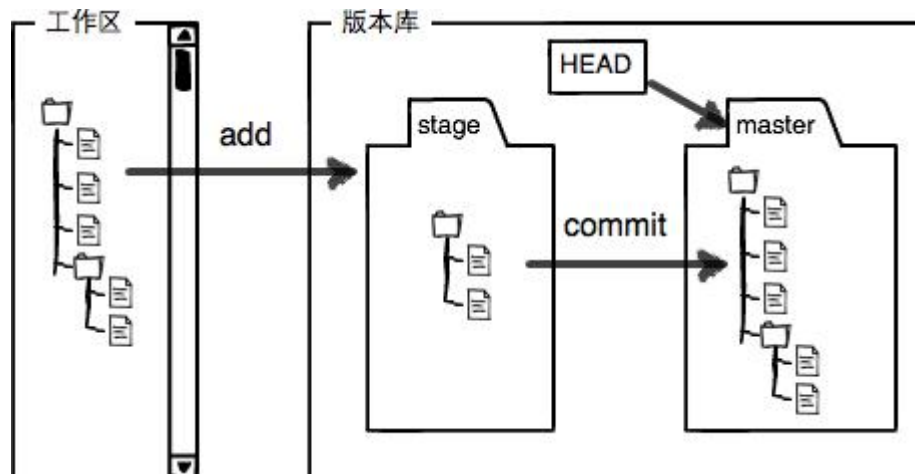
深度学习

.....

找廖雪峰老师

工作区有一个隐藏目录`.git`，这个不算工作区，而是Git的版本库。

Git的版本库里存了很多东西，其中最重要的就是称为stage（或者叫index）的暂存区，还有Git为我们自动创建的第一个分支`master`，以及指向`master`的一个指针叫`HEAD`。



分支和`HEAD`的概念我们以后再讲。

前面讲了我们把文件往Git版本库里添加的时候，是分两步执行的：

第一步是用`git add`把文件添加进去，实际上就是把文件修改添加到暂存区；

第二步是用`git commit`提交更改，实际上就是把暂存区的所有内容提交到当前分支。

因为我们创建Git版本库时，Git自动为我们创建了一个`master`分支，所以，现在，`git commit`就是往`master`分支上提交更改。

你可以简单理解为，需要提交的文件修改通通放到暂存区，然后，一次性提交暂存区的所有修改。

俗话说，实践出真知。现在，我们再练习一遍，先对`readme.txt`做个修改，比如加上一行内容：

```
Git is a distributed version control system.
Git is free software distributed under the GPL.
Git has a mutable index called stage.
```

然后，在工作区新增一个`LICENSE`文本文件（内容随便写）。

先用`git status`查看一下状态。

登录

廖雪峰

担任课程研发战略顾问

Web全栈架构师

2989元VIP视频  
限时免费领取

包括Angular JS. ES6 node.js. vue.js  
Web全栈架构师正式课部分

立即领取

编程

读书

JavaSE教程

JavaEE教程

数字货币

JavaScript教程

Python教程

Git教程

问答

赞助

登录

用 `git status` 查看工作区状态:

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    LICENSE

no changes added to commit (use "git add" and/or "git commit -a")
```

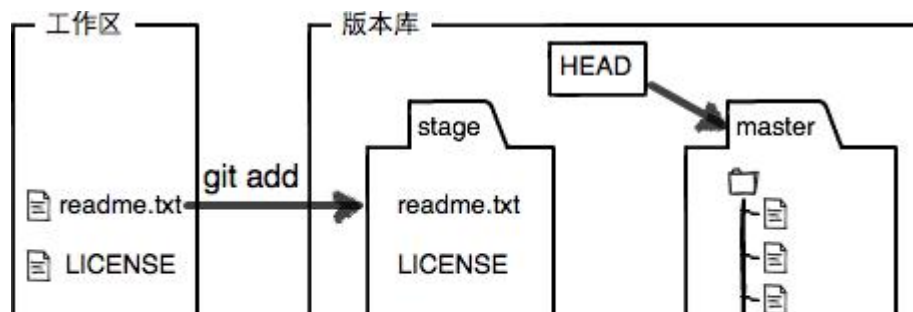
Git非常清楚地告诉我们, `readme.txt` 被修改了, 而 `LICENSE` 还从来没有被添加过, 所以它的状态是 `Untracked`。

现在, 使用两次命令 `git add`, 把 `readme.txt` 和 `LICENSE` 都添加后, 用 `git status` 再查看一下:

```
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   LICENSE
    modified:   readme.txt
```

现在, 暂存区的状态就变成这样了:



# 助力2018世界杯，免费抽奖 - 每天3次免费抽奖，豪礼等你拿

助力2018俄罗斯世界杯，最强赛事来袭，免费抽奖每天3次机会，美金等你拿！

xs9999.com



编程

读书

JavaSE教程

JavaEE教程

数字货币

JavaScript教程

Python教程

Git教程

问答

赞助

登录

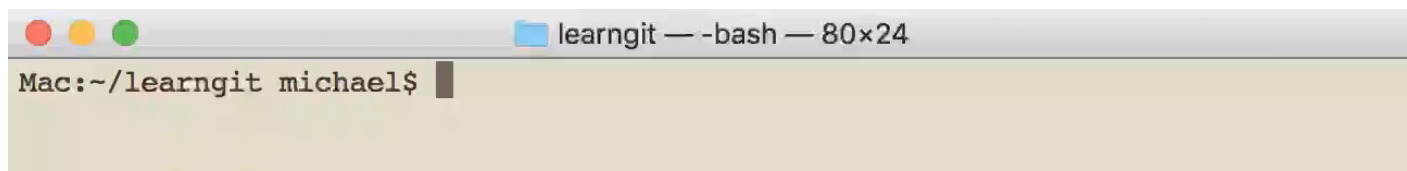
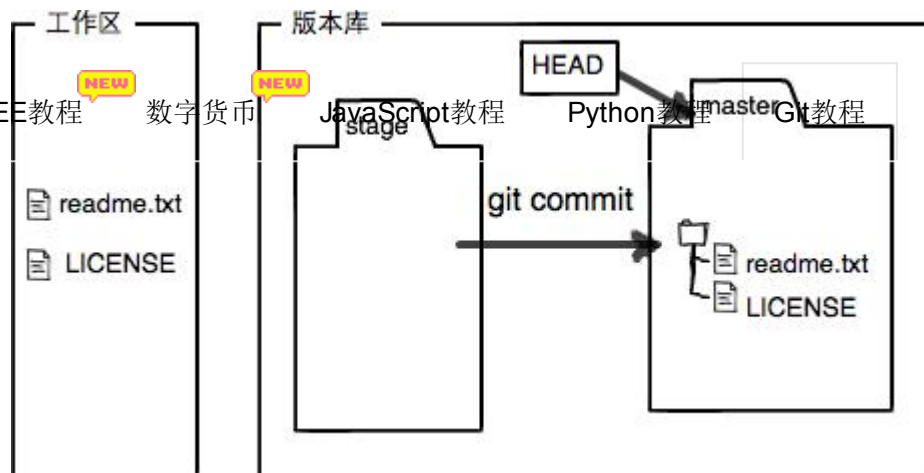
所以，`git add` 命令实际上就是把要提交的所有修改放到暂存区（Stage），然后，执行 `git commit` 就可以一次性把暂存区的所有修改提交到分支。

```
$ git commit -m "understand how stage works"
[master e43a48b] understand how stage works
 2 files changed, 2 insertions(+)
 create mode 100644 LICENSE
```

一旦提交后，如果你又没有对工作区做任何修改，那么工作区就是“干净”的：

```
$ git status
On branch master
nothing to commit, working tree clean
```

现在版本库变成了这样，暂存区就没有任何内容了：



广告X

GoJavaC++SwiftPythonNode.js

一个SDK  
一套极简API

▼

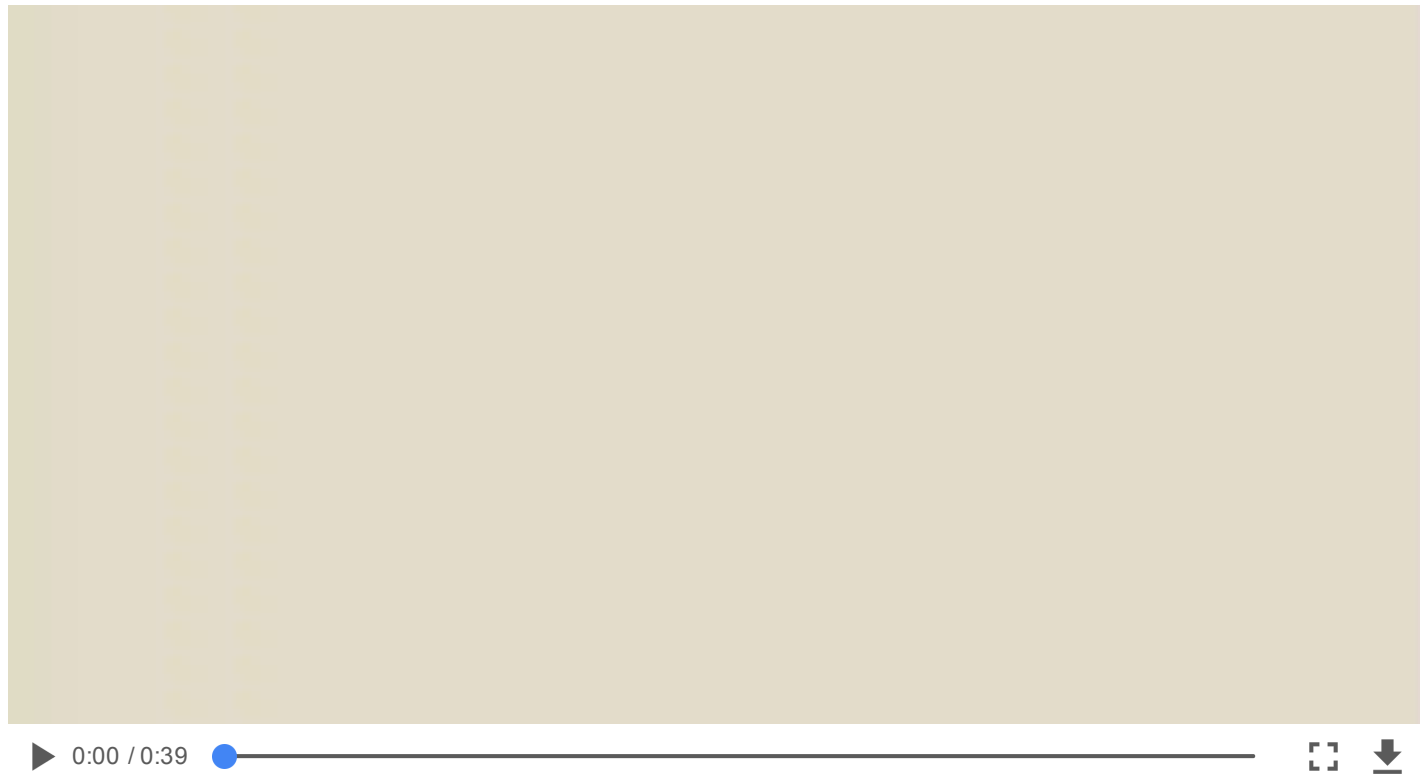
视频通话  
全互动直播

每月1万分钟免费 立即试用

编程 读书 JavaSE教程 JavaEE教程 JavaScript教程 Python教程 Git教程 问答 赞助

agora.io 声网  
Enable Real-time Internet

广告



▶ 0:00 / 0:39

## 小结

NEW 暂存区是Git非常重要的概念，弄明白了暂存区，就弄明白了Git的很多操作到底干了什么。  
NEW 没弄明白暂存区是怎么回事的童鞋，请向上滚动页面，再看一次。

感觉本站内容不错，读后有收获？

¥ 我要小额赞助，鼓励作者写出更好的教程

还可以分享给朋友

分享到微博



## Node.js全栈开发

React

Vue

Angular

webpack

微信开发

koa

专家讲师陪伴您成长

老男孩教育&路飞学城

## 免费内训课

学不到东西算我输!

▶ Python开发入门七天集训营

▶ 爬虫实战密训班

立即加入>>>

廖雪峰

官方 独家

Python

商业爬虫全解码

廖雪峰自己的Python课程  
让天下没有爬不到的数据

Web全栈架构师

国内最顶级

的Web全栈课程

授课深度对标百度T6-T7

猛 戳



编程

读书

JavaSE教程

JavaEE教程

数字货币

JavaScript教程

Python教程

Git教程

问答

赞助

登录

广告X

广告X



PayPal

PayPal

一个账户，收款全球。0费用开户，享  
安全保障 高达2亿用户



腾讯云

腾讯云企业级  
数据库12元/月

腾讯云

支持主从实时热备,自动容灾,备份,恢复,  
监控,迁移等数据全套功能。满足入门



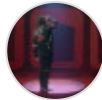
头条头条牌， 飘逸2亿用户。

学习,小规模应用场景

打开

打开

## 评论



作者的 概念图 是 错的把

性感而不令人反感 风流而不下流 created at 2017-10-11 0:38, Last updated at 4-27 20:31

作者的 概念图 是 错的把

stage 并不是 working directory 和 HEAD 之间的 临时枢纽，不能按照 "内存"、"购物车" 来理解 stage，否则你会很难理解 git diff、git diff --staged

stage 中拥有着 和 working directory 中一模一样的 目录结构、文件内容

看 官网上的文章 就知道了：<https://git-scm.com/blog>  
里面有 详细的 概念图



编程

读书

JavaSE教程

JavaEE教程

NEW

数字货币



NEW

性感而不令人反感 风流而不下流

JavaScript教程

Python教程

Git教程

问答

赞助

登录



耳倾若幽

Created at 2017-10-20 0:02, Last updated at 2017-10-20 0:02

It's not technically a tree structure, it's a flattened manifest, but for our purposes it's close enough.

看了INDEX 介绍里面的一句话，也就是stage里面应该只是平面结构而不是树形把..



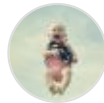
浅滩上的小螃蟹

Created at 4-27 11:10, Last updated at 4-27 11:10



作者概念图是错误的！请大家不要被误导了，**git diff** 确实是比较当前文件和暂存区文件差异，什么是暂存区？就是你还没有执行 **git add** 的文件！！！！  
而文章中说 **add** 之后文件放到暂存区是错误的！！

可以实验一下 你修改 **readme.txt** ,不做 **add** 操作，然后 **git diff** 会显示文件做了那些改动，但是你 **add** 之后，**git diff** 就没有东西了



二水是我的笔名

Created at 4-27 20:31, Last updated at 4-27 20:31

你**add**之后，隐藏的库文件就被更新成最新版了呀，然后**diff**当然没区别了啊！没区别是不是就什么都不显示了吧？我认为作者画的没错啊，**add**之后就是最终版本了，所谓再进行**commit**就只为了给版本变化添加注释和加指针吧！

☰ 全部讨论

↩ 回复



编程

读书

JavaSE教程

JavaEE教程



分析正确，我一直理解有误

浅滩上的小螃蟹 created at 4-27 11:33, Last updated at 4-27 11:33

数字货币  
分析的很好

JavaScript教程

Python教程

Git教程

问答

赞助

➡ 登录

☰ 全部讨论

↩ 回复



我说的才是对的

joychine created at 4-11 17:19, Last updated at 4-11 17:19

**git diff** 是你工作区跟仓库（仓库也叫版本库）的比较，这个时候可以看你开发过程中修改了哪些内容  
**git diff --cached** 是看你**stge**区和仓库分支上的比较，你**add**后但是没有**commit**，这个时候只是在**stae**中，可以确认下修改是否正确，如果正确无误可以**commit**合并到分支



≡ 全部讨论

↩ 回复



### git diff 和 git diff --cached 的理解

shulun\_ly created at 2016-7-9 8:59, Last updated at 2017-11-29 10:43

```
git diff    #是工作区(work dict)和暂存区(stage)的比较
```

```
git diff --cached    #是暂存区(stage)和分支(master)的比较
```

老师, 这样理解正确吗?



廖雪峰

Created at 2016-7-10 20:19, Last updated at 2016-7-10 20:19

正确



tuiide

Created at 2017-2-13 16:02, Last updated at 2017-2-13 16:02

如果git diff是工作区与暂存区的比较,那么将第一次修改的readme.txt文件add并commit后,根据廖老师上面说的图谱来看,暂存区应该是空的吧,我再次修改readme.txt文件后,执行git diff readme.txt,貌似没有跟空内容比较啊,是跟上一轮的对比吧?



蛛蛛200601

Created at 2017-2-20 22:41, Last updated at 2017-2-20 22:41

试验了一下才理解, **stage**或**cache**与虽说是暂存区,缓冲区,但**commit**并不是像想像那样把这个区清空,估计只是打个同步的标志,内容还在,就能理解了。

**add** 是 把工作区的更新到暂存区, **commit**是把缓冲区更新到仓库。所以经过**add**, **commit**,修改再**add**,再修改,就会出现工作区、缓冲区、和仓库三者都不同。

就可以下面的比较了:



编程

读书

JavaSE教程

JavaEE教程

数字货币

JavaScript教程

Python教程

Git教程

将第一次修改的readme.txt文件add并commit

后,根据廖老师上面说的图谱来看,暂存区应该是空的吧,我再次修改readme.txt文件后,执行git diff readme.txt,貌似没有跟空内容比较啊,是跟上一轮的对比吧?

试验了一下才理解, stage或cache与虽说是暂存区,缓冲区,但commit并不是像想像那样把这个区清空,估计只是打个同步的标志,内容还在,就能理解了。

git diff 是工作区和 中间区比较，git diff --cached是中间区和仓库比较。



[张悦寒a](#)

Created at 2017-4-27 14:55, Last updated at 2017-4-27 14:55

git diff #是确定是否要add

git diff --cached #是确定是否要commit

这样理解对吗？



[暴力膜Justify](#)

Created at 2017-5-11 14:48, Last updated at 2017-5-11 14:48

不对吧，git diff应该是工作区和master对比。只add不commit的情况：

```
OptiPlex-790:~/gitlearn$ echo fourth comm >>Readme.txt
OptiPlex-790:~/gitlearn$ git diff
diff --git a/Readme.txt b/Readme.txt
index a1fc811..e6ab6d3 100644
--- a/Readme.txt
+++ b/Readme.txt
@@ -1,3 +1,4 @@
 git init
second commit
third comm
+fourth comm
OptiPlex-790:~/gitlearn$ echo fourth plus comm >>Readme.txt
OptiPlex-790:~/gitlearn$ git diff
diff --git a/Readme.txt b/Readme.txt
index a1fc811..f28ca9d 100644
--- a/Readme.txt
+++ b/Readme.txt
@@ -1,3 +1,5 @@
 git init
second commit
```

▼ [Read More](#)



[编程](#)

[读书](#)

[JavaSE教程](#)

[JavaEE教程](#)

NEW

[数字货币](#)

NEW

[JavaScript教程](#)

[Python教程](#)

[Git教程](#)

[问答](#)

[赞助](#)

[登录](#)



[暴力膜Justify](#)



不好意思，忘记add了。 。 。 。 。 。 。 。 。 。 。



Created at 2017-7-30 17:09, Last updated at 2017-7-30 17:09

`git diff` 是你工作区跟**stage**的比较，这个时候可以看你开发过程中修改了哪些内容  
`git diff --cached` 是看你**stge**区和仓库分支上的比较，你**add**后但是没有**commit**，这个时候只是在**stage**中，可以确认下修改是否正确，如果正确无误可以**commit**合并到分支



Created at 2017-8-6 23:04, Last updated at 2017-8-6 23:04

git diff #是工作区(work dict)和暂存区(stage)的比较  
git diff --cached #是暂存区(stage)和分支(master)的比较

回复上面的哥们的疑问：第一次修改的readme.txt文件add并commit后，根据廖老师上面说的图谱来看，暂存区应该是空的，再次修改readme.txt文件后，执行git diff readme.txt。此时工作区有修改，而暂存区是还没被add，没有修改，可以看作是空的，所以执行结果是有被修改了。



读书

JavaSE教程

JavaEE教程

## 数字货币

## JavaScript教程

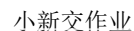
# Python教程

# Git教程

## 问答

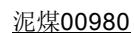
贊助

➡ 登录



Created at 2017-9-27 18:36, Last updated at 2017-9-27 18:36

git diff test.txt 和 git diff -- test.txt 都是stage的test.txt与工作区的test.txt作比较；  
git diff HEAD -- test.txt 是branch的test.txt与工作区的test.txt作比较；



Created at 2017-11-29 10:43, Last updated at 2017-11-29 10:43

个人理解:

楼主需再加一行：

如果git add之后，git status 虽然什么都不给我们显示（感觉暂存区没有东西了），但是此时暂存区是和分支一样子的，git diff 还是和暂存区作比较的，可以理解为暂存区有和

我们目前只是将分支与分支的git diff是查看自己本地分支与远程分支一模一样的东西，但是不给我们显示。

全部讨论

回复



### 暂存区和库的形象比喻

孤云飘飘zhao created at 2017-8-31 18:14, Last updated at 2017-8-31 18:14

你可以把暂存区里的stage理解为内存，master理解为硬盘，工作区可以理解为cpu以及CPU的各种缓存区和寄存器

工作区主要用来处理数据之类的

暂存区stage用来临时缓冲数据用的，可以进行增删改查，方便处理。

master专门用来长期保存数据用的

全部讨论

回复



### git diff

小l babla created at 2017-7-26 0:55, Last updated at 2017-7-26 0:55

为什么diff是工作区(work dict)和暂存区(stage)的比较呢？前面不是提到说commit之后，暂存区就没有任何内容了吗？git diff应该是工作区和版本库中的当前分支进行比较吧？

还是说『暂存区就没有任何内容了』的含义其实是指待提交的文件数量是空的，而文件内容和版本库中的文件是同步的？



我姓冷叫前

Created at 2017-9-24 14:03, Last updated at 2017-9-24 14:03

你自己试试吧。git commit之后，再git diff是没有反馈的。将readme.txt更改并保存之后，git diff就会有修改反馈了。这样看来，git diff是将工作区和缓存区进行比较了。

全部讨论

回复



编程

读书

JavaSE教程

JavaEE教程

NEW

数字货币

JavaScript教程

Python教程

Git教程

问答

赞助

登录



### 评论: 工作区和暂存区

z酱 created at 2015-8-12 15:46, Last updated at 2017-7-6 17:53

暂存区的意义何在呢?

感觉比较麻烦啊, 每次commit之前还需要add一次



z酱

Created at 2015-8-12 15:47, Last updated at 2015-8-12 15:47

如果我忘记add(这个文件之前已经add过)

直接commit会有什么情况出现呢



新的开始爱自己

Created at 2015-9-21 10:49, Last updated at 2015-9-21 10:49

暂存区就是来确定你修改了哪些文件吧, 修改过的文件放置到暂存区。等确定完哪些需要提交之后, 再在暂存区一次性commit到分支里面。没有这个区域的话怎么知道哪些文件是修改过的?

个人理解



编程

读书

JavaSE教程

JavaEE教程

NEW

数字货币

NEW

JavaScript教程  
oneMortale

Python教程

Git教程

问答

赞助

登录



Created at 2015-10-3 12:34, Last updated at 2015-10-3 12:34

这么理解呢, 所谓暂存区是不是跟CoreData里的context差不多, 是内存上适时分配的一部分空间。当你源码做出修改后, git会基于之前的记录实时更新你的修改, 但是如果你不进行commit, 这些更改是不是被写入到硬盘里的。



oneMortale

Created at 2015-10-3 12:44, Last updated at 2015-10-3 12:44

看完下一章的话, 应该就是这样了吧, add 会提示git我们进行了修改, 然后git会去找到修改的具体内容, 找到的结果会放在暂存区, 如果没有提示修改, git不会去查看这个文件, 修改就会遗漏。这个暂存区是在硬盘上的还是内存上的不知道。



[sphenisciforme](#)

Created at 2015-11-11 15:06, Last updated at 2015-11-11 15:06

如果没有暂存区的概念，我们在容器中任意的修改commit的时候都会被加入到branch里，但是有可能我们添加一个临时文件，但是我们不想把它commit上去。



[来撸一发吧](#)

Created at 2015-11-30 13:02, Last updated at 2015-11-30 13:02

\*\*或许可以把它类比为内存缓冲区，add就类似于先在这个缓冲区里填入一些数据，然后commit就类似于一次性缓冲区里的数据写入硬盘，能提高效率。。。个人理解，刚接触git，希望大神纠正错误，不要让我误导更小白的小童鞋，哈哈。



[海绵他奶](#)

Created at 2015-12-4 18:11, Last updated at 2015-12-4 18:11

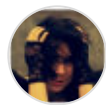
我觉得嘛，暂存区就像购物车，没到付款的时候你都不确定购物车里的东西全部都要的。。。每拿一件商品就付一次款。。。那才麻烦大了



[用户ks4g78j0my](#)

Created at 2015-12-20 10:56, Last updated at 2015-12-20 10:56

购物车那个比喻，真是形象~~  $O(n\_n)O\sim$



[fengengChen](#)

Created at 2016-1-4 17:08, Last updated at 2016-1-4 17:08

购物车的比喻很形象



[善良的随行](#)

Created at 2016-2-24 17:50, Last updated at 2016-2-24 17:50

...



[编程](#)

[读书](#)

[JavaSE教程](#)

[JavaEE教程](#)

NEW

[数字货币](#)

NEW

[JavaScript教程](#)

[Python教程](#)

[Git教程](#)

[问答](#)

[赞助](#)

[登录](#)



当多个文件修改完成放入暂存区的时候，发现其中一个文件的代码有问题，这时候你可以用checkout单独将这个文件还原重改；如果你将这些文件一次性全部放进库里，等你发现有问题的就不能单独拿出一个文件了，只能版本回档，那时你就要重新修改所有文件的代码。

☰ 全部讨论

↩ 回复



### git status后出现错误

兰德莛 created at 2017-6-4 16:40, Last updated at 2017-6-4 16:40

总会出现“HEAD detached from 09d5a14”错误，有没有遇到的呢？

☰ 全部讨论

↩ 回复



> <

Lao\_Er created at 2017-4-27 14:53, Last updated at 2017-5-1 19:54

git commit 命令为什么返回

'675 insertions(+)



廖雪峰

Created at 2017-5-1 19:54, Last updated at 2017-5-1 19:54

就是告诉你本次commit新增了675行

☰ 全部讨论

↩ 回复



### git diff是本地和版本库版本比较

武汉老徐2011 created at 2017-4-17 14:50, Last updated at 2017-4-17 14:50

git diff是比较本地和版本库



编程

读书

JavaSE教程

JavaEE教程

NEW

数字货币

NEW

JavaScript教程

Python教程

Git教程

问答

赞助

➡ 登录

git diff --cached是比较本地和版本库

☰ 全部讨论

↩ 回复

发表评论

登录后发表评论

[廖雪峰的官方网站](#)©2017 vbd90c68

Powered by [iTranswarp.js](#)

由[阿里云](#)托管

[广告合作](#)



[意见反馈](#)

[使用许可](#)

友情链接: [中华诗词](#) - [阿里云](#) - [金山云](#) - [SICP](#) - [4clojure](#) - [linuxprobe](#)



编程

读书

JavaSE教程

JavaEE教程

NEW

数字货币

NEW

JavaScript教程

Python教程

Git教程

问答

赞助

➡ 登录