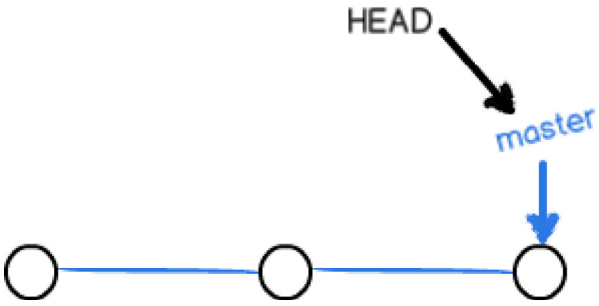
创建与合并分支

阅读: 17289098

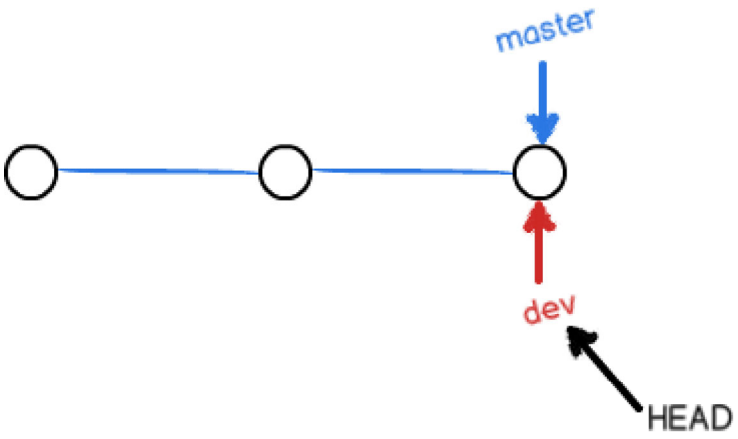
在版本回退里，你已经知道，每次提交，Git都把它们串成一条时间线，这条时间线就是一个分支。截止到目前，只有一条时间线，在Git里，这个分支叫主分支，即`master`分支。`HEAD`严格来说不是指向提交，而是指向`master`，`master`才是指向提交的，所以，`HEAD`指向的就是当前分支。

一开始的时候，`master`分支是一条线，Git用`master`指向最新的提交，再用`HEAD`指向`master`，就能确定当前分支，以及当前分支的提交点：



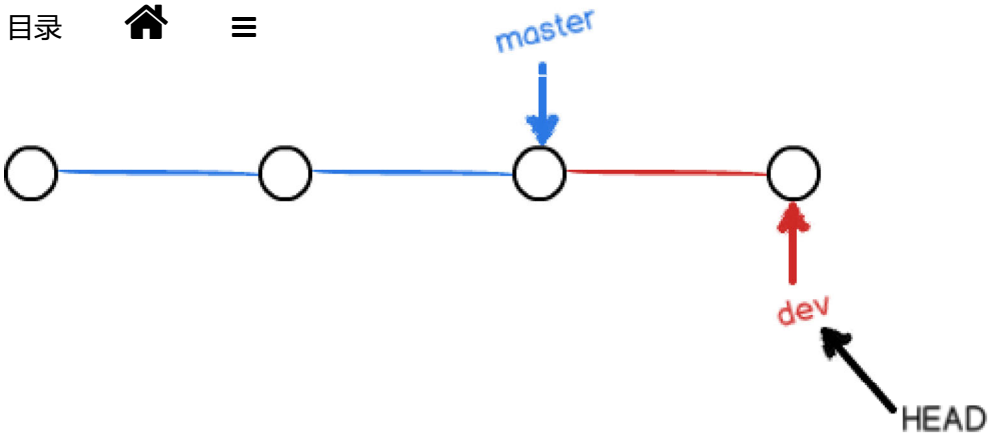
每次提交，`master`分支都会向前移动一步，这样，随着你不断提交，`master`分支的线也越来越长。

当我们创建新的分支，例如`dev`时，Git新建了一个指针叫`dev`，指向`master`相同的提交，再把`HEAD`指向`dev`，就表示当前分支在`dev`上：

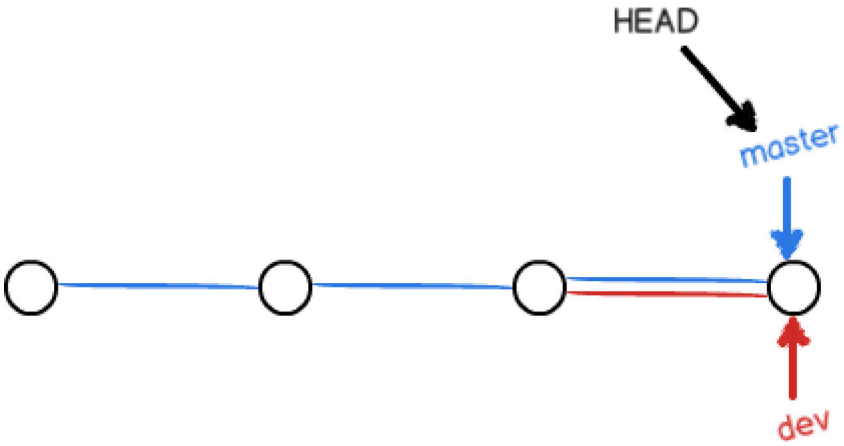


你看，Git创建一个分支很快，因为除了增加一个`dev`指针，改改`HEAD`的指向，工作区的文件都没有任何变化！

不过，从现在开始，对工作区的修改和提交就是针对`dev`分支了，比如新提交一次后，`dev`指针往前移动一步，而`master`指针不变：

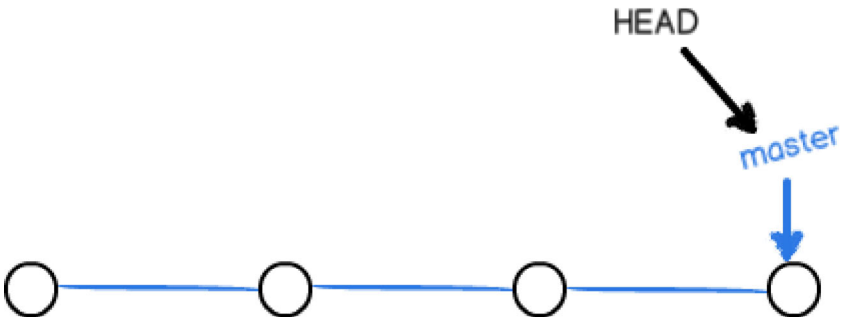


假如我们在`dev`上的工作完成了，就可以把`dev`合并到`master`上。Git怎么合并呢？最简单的方法，就是直接把`master`指向`dev`的当前提交，就完成了合并：



所以Git合并分支也很快！就改改指针，工作区内容也不变！

合并完分支后，甚至可以删除`dev`分支。删除`dev`分支就是把`dev`指针给删掉，删掉后，我们就剩下了一条`master`分支：



真是太神奇了，你看得出来有些提交是通过分支完成的吗？

下面开始实战。

首先，我们创建`dev`分支，然后切换到`dev`分支：

```
$ git checkout -b dev
Switched to a new branch 'dev'
```

`git checkout`命令加上`-b`参数表示创建并切换，相当于以下两条命令：

```
$ git branch dev
$ git checkout dev
Switched to branch 'dev'
```

然后，用`git branch`命令查看当前分支：

```
$ git branch
* dev
  master
```

`git branch`命令会列出所有分支，当前分支前面会标一个`*`号。

然后，我们就可以在`dev`分支上正常提交，比如对`readme.txt`做个修改，加上一行：

```
Creating a new branch is quick.
```

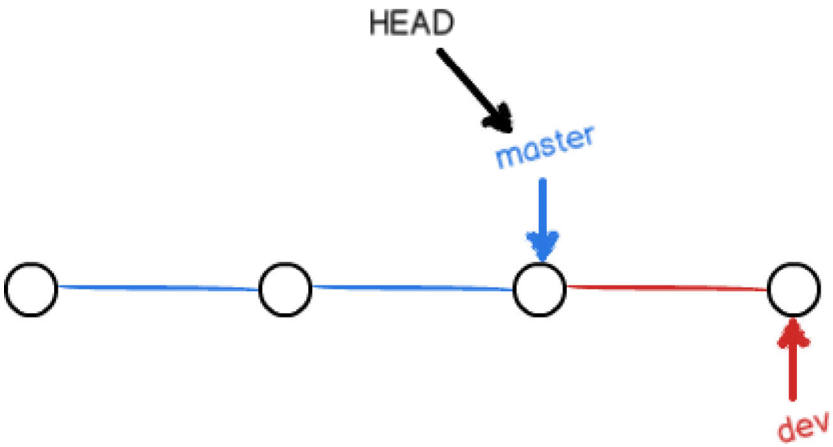
然后提交：

```
$ git add readme.txt
$ git commit -m "branch test"
[dev b17d20e] branch test
 1 file changed, 1 insertion(+)
```

现在，`dev`分支的工作完成，我们就可以切换回`master`分支：

```
$ git checkout master
Switched to branch 'master'
```

切换回`master`分支后，再查看一个`readme.txt`文件，刚才添加的内容不见了！因为那个提交是在`dev`分支上，而`master`分支此刻的提交点并没有变：



现在，我们把`dev`分支的工作成果合并到`master`分支上：

```
$ git merge dev
Updating d46f35e..b17d20e
Fast-forward
 readme.txt | 1 +
 1 file changed, 1 insertion(+)
```

`git merge`命令用于合并指定分支到当前分支。合并后，再查看`readme.txt`的内容，就可以看到，和`dev`分支的最新提交是完全一样的。

注意到上面的`Fast-forward`信息，Git告诉我们，这次合并是“快进模式”，也就是直接把`master`指向`dev`的当前提交，所以合并速度非常快。

当然，也不是每次合并都能`Fast-forward`，我们后面会讲其他方式的合并。

合并完成后，就可以放心地删除`dev`分支了：

```
$ git branch -d dev
Deleted branch dev (was b17d20e).
```

删除后，查看`branch`，就只剩下`master`分支了：

```
$ git branch
* master
```

因为创建、合并和删除分支非常快，所以Git鼓励你使用分支完成某个任务，合并后再删掉分支，这和直接在`master`分支上工作效果是一样的，但过程更安全。

目录1 人在看： 者荣耀大仙，裴擒虎到底该不该出… 立即围观 >

扫一扫 手机看

00:00 / 00:41 360P

进入bilibili,一起发弹幕吐槽!

去吐槽

小结

Git鼓励大量使用分支：

查看分支：`git branch`

创建分支：`git branch <name>`

切换分支：`git checkout <name>`

创建+切换分支：`git checkout -b <name>`

合并某分支到当前分支：`git merge <name>`

删除分支：`git branch -d <name>`

读后有收获可以支付宝请作者喝咖啡：

还可以分享给朋友：

分享到微博

上一页 下一页

广告 ×

专业留学辅导来北京新东方

北京新东方最新留学方案

查看

广告 ×

专业留学辅导来北京新东方

北京新东方最新留学方案

查看

评论

合并

为你而微99690 created at March 29, 2019 6:21 PM, Last updated at July 3, 2019 11:30 AM

老师，这个每次有改动都需要去add和commit一下嘛？？但是我发现，在重新创建一个分支之后，如果不做提交操作的话，直接切换到主分支之后也是可以看到其他分支做的修改操作，那这样的话，干嘛其他分支还要add和提交呢？？也不需要切换到主分支之后在进行合并操作了呀？？？直接在主分支push操作就可以了呀

runningman765
Created at April 4, 2019 2:27 PM

你确定，不把`dev`分支的代码`merge`到`master`分支，你能在`master`分支上看到`dev`分支的代码?

runningman765
Created at April 4, 2019 2:30 PM

除非你`master`分支`track`的是你的`dev`分支，这样是有问题的

SunnyTH_21
Created at April 9, 2019 2:04 PM

在dev分支上修改了文件，但是并没有执行git add. git commit命令，然后切换到master分支，仍然能看到dev分支的改动，请问这个现象怎么解释?

Hi_LP
Created at April 11, 2019 4:00 PM

的确，如果在在dev分区修改了文件，然后切换到主分区，查看status，是可以看到状态的，就可以在主分区来add 和commit dev 分区的改动

Hi_LP
Created at April 11, 2019 4:08 PM

但是如果在dev分区修改了文件，并且add 和commit 了 那么切换到主分区就看不到修改了

kache1995
Created at April 12, 2019 9:40 PM

你在dev分支修改了文件，但是你没有提交到仓库，实际上就是相当于你在本地手动修改了这个文件，仓库并不能保存你做的改动，所以在master分支能看到文件被改动了（相当于你没用dev分支直接修改了这个文件一样），所以你可以用master分支add、commit

目录

新e风艺术工作室
Created at April 21, 2019 11:07 AM

同意楼上

x573676252511614141
Created at April 30, 2019 3:53 PM

对于所有分支而言， 工作区和暂存区是公共的。

https://blog.csdn.net/stpeace/article/details/84351160 https://blog.csdn.net/w522301629/article/details/81331273

浅唱忧凉
Created at July 3, 2019 11:30 AM

分支修改后没有提交是无法切换到另一分支的

$ git checkout master

error: Your local changes to the following files would be overwritten by checkout:

README.md

Please commit your changes or stash them before you switch branches.

全部讨论 回复

删除远程分支

薄葬殘香 created at April 16, 2019 9:31 AM, Last updated at April 16, 2019 2:48 PM

先通过`git branch -d <branch name>`删除本地分支，再通过`git push origin :<branch name>`删除远程分支，可以参考如何删除远程的标签

薄葬殘香
Created at April 16, 2019 2:48 PM

`git push origin -d <branch name>`好像效果一样

全部讨论 回复

md后缀文件显示不正确相关答疑

长宁特产 created at February 24, 2019 7:31 PM, Last updated at February 24, 2019 7:31 PM

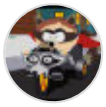
假如是中文乱码，确定修改了vimrc之后还需要再VIM里面输入命令，而且汉字不能夹带中文括号。 以及对于换行问题，md文件每一行末尾留两个空格才能正确换行

全部讨论 回复

建立分支一系列操作后，进行push到GitHub上，并不能同步操作

用户6462391622 created at December 26, 2018 9:24 AM, Last updated at February 24, 2019 7:17 PM

在本地进行git push 之后显示成功，然而在GitHub远程仓库中不能同步本地的数据，而且已经进行commit了，求解?



长宁特产
Created at February 24, 2019 7:17 PM

刷新一下?

全部讨论 回复

多个分支合并能自动解决冲突吗?

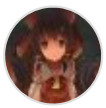
嗷大彬彬 created at August 27, 2017 11:28 PM, Last updated at February 12, 2019 10:46 AM

假设某项目进行到master的时候，需要开发A功能，于是创建了a分支，由甲程序员复制开发，其他程序员继续进行master主要功能的开发，进行到某个时候又需要进行B功能的开发，此时A功能分支还未合并，于是创建了B分支，过一段时间后A合并到master，再过一段时间B又合并到主分支。这个过程应该会有代码冲突吧？git会自动解决这些冲突吗？另外，svn能实现这部分功能吗？盼复。

--------master-------master-------A分支合并-----B分支合并 | | A分支 B分支

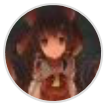
廖雪峰
Created at August 28, 2017 10:20 AM

任何合并冲突都必须人去解决，任何工具能解决那它就已经有智能了



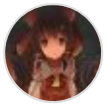
mizuku
Created at January 20, 2018 6:01 PM

这个如果a和b都改变了同一个文件的同一个地方，改的还不一样，那么第二



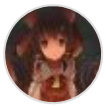
mizuku
Created at January 20, 2018 6:13 PM

试了试，直接提示conflict，不让你这么干



mizuku
Created at January 20, 2018 6:20 PM

可以手动修改conflict，然后提交或者--abort抛弃merge b的操作



mizuku
Created at January 20, 2018 6:22 PM

我蠢死了，没想到下一课就有怎么解决冲突QAQ

当知此处
Created at February 12, 2019 10:46 AM

冲突就手动解决冲突嘛，但是工程开发的时候不会有太多冲突的地方的，开发A功能肯定就在A模块里开发，开发B功能就在B模块里开发，最多就是同时更改了某个main文件

全部讨论 回复

文章太赞了

胡浪同學 created at December 26, 2018 6:15 PM, Last updated at December 26, 2018 6:15 PM

比其他教程真的好太多，容易懂太多

全部讨论 回复

实验：处在落后的分支上时，去merge领先的分支

ddd滚去碎觉bbb created at October 8, 2017 6:51 PM, Last updated at November 16, 2018 3:55 PM

我做了个实验。当处在落后的分支上时，去merge领先的分支。用文中例子就是：

```
$ git checkout dev
$ git merge master
```

得到的提示是：

```
Already up-to-date.
```

说明这样的操作（当处在落后的分支上时，去merge领先的分支。），是不能完成合并的。

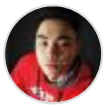
我说的对么？老师。

励志做大V
Created at October 12, 2017 1:54 PM

你之前的分支应该没有commit，所以合并的时候会提示Already up-to-date. 把另一个分支的修改commit，然后再merge就可以了

Lifangchao668
Created at July 20, 2018 4:53 PM

我去试了一下，确实不行，就算master分支commit了也不行



Jolyon_Chong
Created at August 2, 2018 4:02 PM

当你处在落后的分支上（master），去merge领先的分支， 使用 git checkout dev git merger master

> 那么问题来了，你当前的分支是 `master` 运行了指令 `git checkout dev` 就切换到分支dev（领先分支） 在用 `merge merge master` 合并指定分支到当前分支，就是将落后（master分支）的合并到领先的（dev分支），这样是用领先的分支去合并落后的分支，是不可行的，和你提出的问题--落后的分支去合并领先的分支是不可行的完全相反！

槲寄君
Created at November 16, 2018 3:55 PM

楼上说得对，是你没弄清楚啊，dev是领先分支啊，master才是落后分支

全部讨论　回复

此文只应天上有，人间能得几回闻。

用户6186236375 created at September 23, 2018 8:07 PM, Last updated at September 23, 2018 8:07 PM

从未见过如此思路清晰美妙的git文章。

全部讨论 回复

优雅

mttsui_37457 created at September 10, 2018 9:36 AM, Last updated at September 10, 2018 9:36 AM

命令行并没有**浪费**一个字

全部讨论 回复

这个问题和对暂存区的概念有关么?

Sansiro_Santon created at April 25, 2018 4:21 PM, Last updated at April 25, 2018 4:21 PM

创建了分之后，修改了readme.txt， commit之前还要 git add "readme.txt"， 这个如果补充说明一下，就更好了。

全部讨论 回复

可以详细讲下rebase命令么

逗逼程序猿的搬砖生活 created at February 8, 2018 11:11 AM, Last updated at February 8, 2018 11:11 AM

大神，可以详细讲下rebase命令么，看了些文章感觉还是没有理解很透彻

全部讨论 回复

这是怎么回事？有哪位大神遇到过吗？求解决方法

沐瑶的奶爸 created at December 25, 2017 3:48 PM, Last updated at December 26, 2017 1:27 PM

$ git commit -m "woyebuzhidao" On branch master Untracked files: .raedme.txt.swp

nothing added to commit but untracked files present

lautumn1990

Created at December 26, 2017 1:27 PM

先`git add .`,或者`git add .raedme.txt.swp`,然后`git commit -m "woyebuzhidao"`

全部讨论 回复

一个工作流程的问题

TONGSir丶 created at November 29, 2017 5:29 PM, Last updated at November 29, 2017 5:29 PM

有这样一个问题。。 创建devA，开发功能1， 创建devB，开发功能2， 合并devA到master， => v1.0 合并devB到master， => v2.0 此时devA对应的应该是v1.0吧， 这个时候在不想删除掉devA的情况下如何同步masterV2.0的代码，延用devA这个分支继续开发？ 还是需要删掉devA，再创建masterV2.0的分支devA呢？ 这个地方没有完全理解，求指教。

全部讨论 回复

无法checkout new index,这是什么情况

一部1994年上映的电影 created at September 18, 2017 4:15 PM, Last updated at September 18, 2017 4:15 PM

代码如下：

$ git checkout dev fatal: Unable to create 'C:/Users/acer1/learngit/.git/index.lock': File exists.

If no other git process is currently running, this probably means a git process crashed in this repository earlier. Make sure no other git process is running and remove the file manually to continue.

一部1994年上映的电影
Created at September 18, 2017 4:17 PM

问题解决了

一部1994年上映的电影
Created at September 18, 2017 4:20 PM

根据上面的提示我把那个已经存在的文件删除了

本地删除分支后，如何在github中同步？

我又回来了围脖君 created at September 6, 2017 10:06 AM, Last updated at September 6, 2017 10:06 AM

我在本地用git branch -d dev删除了dev分支 但是不知道如何将这个操作同步到github上？ 需要手动在github删除分支吗?

一个代号5069
Created at September 12, 2017 3:02 PM

本地操作完之后，用`git push origin master`推送到远程GitHub上就行。

空白的画框
Created at September 13, 2017 5:49 PM

可以用git push origin :dev 删除远程分支dev， origin和:dev之间有空格

全部讨论 回复

MP4 在谷歌浏览器上面播放不了

innerpeacenxn created at August 18, 2017 10:50 PM, Last updated at August 18, 2017 10:50 PM

报错信息如下: `Failed to load resource: net::ERR_CONNECTION_TIMED_OUT`

这个有办法解决吗?

全部讨论 回复

理解分支最好的方法

不吃草的咩咩 created at August 16, 2017 9:22 PM, Last updated at August 16, 2017 9:22 PM

把分支想象成RPG游戏的第二个存档，更多分支就是第三第四个存档 这些存档多数都是用来做master第一个存档不敢做的事情 在这些分支存档上做的事情不会影响第一个master存档

全部讨论 回复

git pull 和 git merge的区别?

雪子V博 created at August 16, 2017 6:49 PM, Last updated at August 16, 2017 6:49 PM

项目中比较习惯直接自己的分支,去pull 别人的分支,一样能把修改的代码合下来,想知道git pull origin devbranch 和 git merge 的区别。。 谢谢..

LaoKia
Created at August 29, 2017 1:24 PM

pull after merge

职业搬砖工Zin
Created at August 29, 2017 6:32 PM

git pull = git fetch + git merge

全部讨论 回复

赞?

大男子主义的小女生有个小理想 created at August 16, 2017 3:25 PM, Last updated at August 16, 2017 3:25 PM

老哥的教程真心给力!

全部讨论 回复

关于HEAD

重庆_87877 created at July 18, 2017 9:01 PM, Last updated at July 18, 2017 9:01 PM

前面的章节不是说HEAD指向当前版本吗？怎么指向的是当前分支了？

廖雪峰
Created at July 18, 2017 9:37 PM

HEAD指向的当前分支就是当前版本

全部讨论 回复

发表评论

目录

登录后发表评论

廖雪峰的官方网站©2019
Powered by iTranswarp
本网站运行在阿里云上并使用阿里云CDN加速。

友情链接: 中华诗词 - 阿里云 - SICP - 4clojure