

Git：什么是悬空的提交/ blob，它们来自何处？

□ 2020-08-30

□ [git](#) [git-commit](#) [git-dangling](#) [git-fsck](#)

Git: what is a dangling commit/blob and where do they come from?

我正在寻找有关悬挂提交和斑点的基本信息。

我的仓库似乎很好。但是我第一次运行 `git fsck` 看看它做了什么，并且有一长串的"悬挂斑点"和一个"悬挂提交"。

这些是什么？哪儿来的？它们表明我的回购状态是否有异常(好坏)吗？



在使用git存储库的过程中，您可能最终会退出操作，并做出其他导致中介blob的动作，甚至git为您做的某些事情也有助于避免信息丢失。

最终(有条件地，根据git gc手册页)，它将执行垃圾回收并清理这些东西。您也可以通过调用垃圾回收进程 `git gc` 来强制执行此操作。

有关此信息，请参阅git-scm站点上的维护和数据恢复。

默认情况下，手动运行GC将在安全网此命令运行之前2周离开。实际上，建议您偶尔运行GC，以确保高效地使用git存储库。但是，像其他任何事情一样，在销毁那些可能对您很重要的东西之前，您应该先了解它的作用。

□ [相关讨论](#)



悬挂斑点=更改到暂存区/索引但从未提交的更改。git令人惊奇的一件事是，一旦将其添加到暂存区中，您就可以随时将其取回，因为这些blob的行为就像是提交，因为它们也具有哈希！

悬挂提交=未与任何子提交，分支，标签或其他引用直接链接的提交。您也可以将其取回！

□ [相关讨论](#)



如何从git存储库中删除所有悬空的提交

<http://www.tekkie.ro/news/howto-remove-all-dangling-commits-from-your-git-repository/>

```
1 git reflog expire --expire=now --all
2 git gc --prune=now
```

确保您确实要删除它们，因为您可能最终决定需要它们。

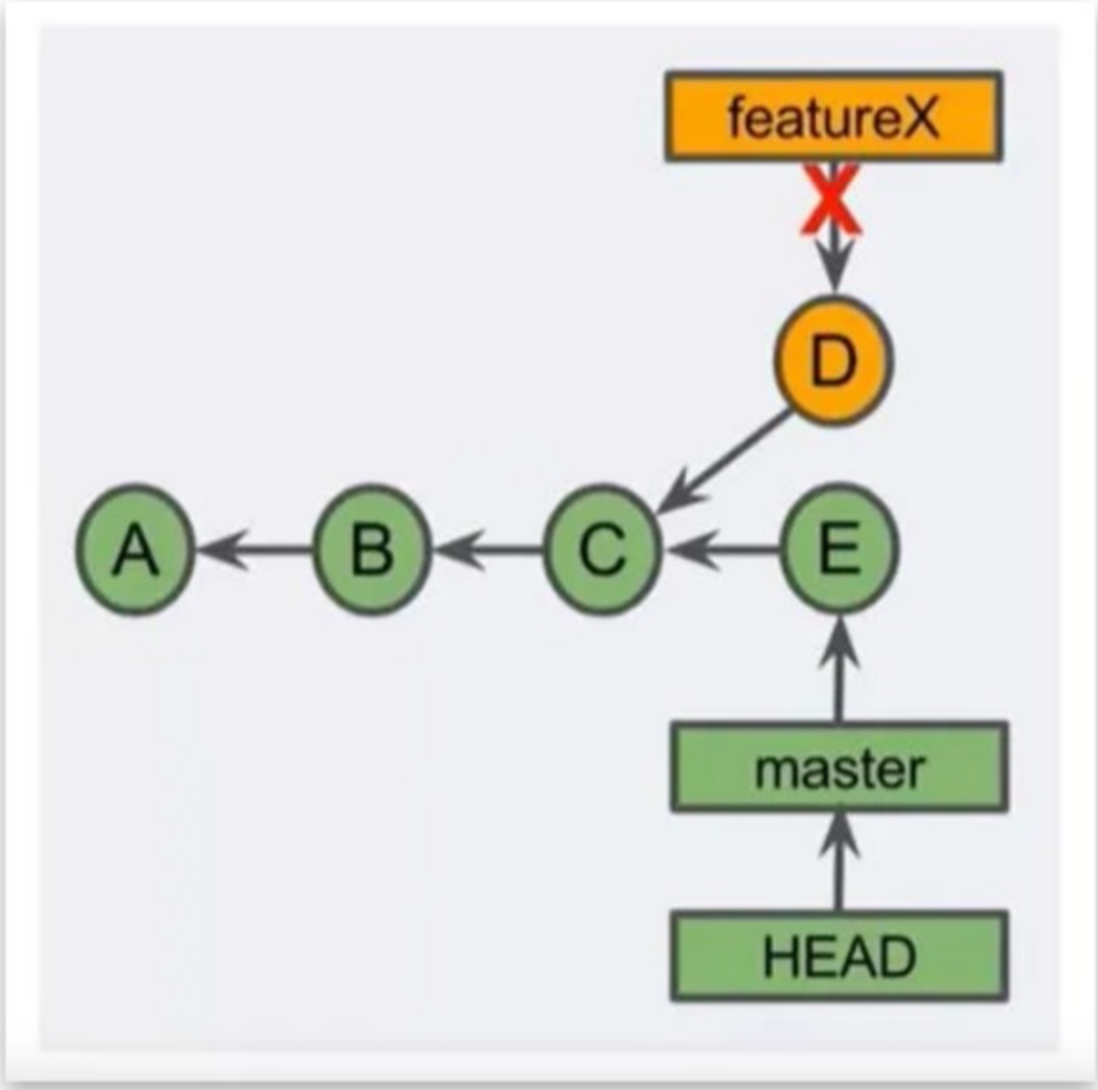
□ [相关讨论](#)



悬空提交是与引用无关的提交，即无法到达引用。

例如，考虑下图。假设我们删除分支FeatureX而不合并其更改，则提交D将变为悬空提交，因为没有与之关联的引用。如果将它合并到master中，那么HEAD和master引用将指向提交D，即使我们删除了featureX，它也不再悬而未决。阅读图表后的注释可更好地理解这一点。

Git自动垃圾收集(即处理)悬空的提交。我们可以使用 `git reflog` 来恢复分支(悬挂提交)，该分支在不合并的情况下被删除。只有本地对象存储中存在已删除的提交，我们才能恢复已删除的提交。如果是垃圾回收，那么我们将无法恢复。



请注意，分支名称(即分支标签)实际上是对分支上最新提交的引用，即分支的尖端。在上图中，featureX，master和HEAD只是对特定提交的引用。featureX和主标签引用各自分支上的最新提交。HEAD通常是指当前已签出分支的尖端(在这种情况下为master)。如果您在当前分支上签出较旧的提交，则HEAD将处于分离状态，即它将指向较早的提交而不是最新的提交。还要注意，HEAD之所以称为符号引用，是因为它实际上指向当前分支标签，并且任何分支标签始终指向该分支的尖端。因此，在正常情况下，HEAD间接指向最新提交。

顺便说一句，请注意，Git将其提交图/历史表示为有向无环图。每个提交都有对其父对象的引用。因此，提交图中的箭头从子提交指向父提交。我们需要引用最新的子提交，才能到达分支上的较早提交。

PS-以上图表和理解是从此免费课程中获得的。尽管课程很老，但知识仍然很重要。

