工作目录下的每一个文件都不外乎两种状态：已跟踪 或 未跟踪。已跟踪的文件是指那些被纳入了版本控制的文件，在上一次快照中有它们的记录，在工作一段时间后， 它们的状态可能是未修改，已修改或已放入暂存区。简而言之，已跟踪的文件就是 Git 已经知道的文件。

工作目录中的所有文件都属于**已跟踪文件**，并**处于未修改状态**，因为 Git 刚刚检出了它们， 而你尚未编辑过它们

Git add file可用于将未跟踪文件放入暂存区，也可以将修改状态的文件放入暂存区；

Git commit 操作将暂存区提交到工作区。提交时记录的是放在暂存区域的快照。 任何还未暂存文件的仍然保持已修改状态，可以在下次提交时纳入版本管理。 每一次运行提交操作，都是对你项目作一次快照，以后可以回到这个状态，或者进行比较。

git diff 本身只显示尚未暂存的改动，而不是自上次提交以来所做的所有改动。 所以有时候你一下子暂存了所有更新过的文件；即如果改动被暂存则其只显示暂存之后改动的文件修改内容。

要从 Git 中移除某个文件，就必须要从已跟踪文件清单中移除（确切地说，是从暂存区域移除），然后提交。 可以用 git rm 命令完成此项工作，并连带从工作目录中删除指定的文件，这样以后就不会出现在未跟踪文件清单中了。

如果要删除之前修改过或已经放到暂存区的文件，则必须使用强制删除选项 -f（译注：即 force 的首字母）。 这是一种安全特性，用于防止误删尚未添加到快照的数据，这样的数据不能被 Git 恢复

另外一种情况是，我们想把文件从 Git 仓库中删除（亦即从暂存区域移除），但仍然希望保留在当前工作目录中。 换句话说，你想让文件保留在磁盘，但是并不想让 Git 继续跟踪。 当你忘记添加 .gitignore 文件，不小心把一个很大的日志文件或一堆 .a 这样的编译生成文件添加到暂存区时，这一做法尤其有用。 为达到这一目的，使用 --cached 选项

移动文件：git mv file\_from file\_to

相当于：mv README.md README

git rm README.md

git add README

在 Git 中任何 已提交 的东西几乎总是可以恢复的。 甚至那些被删除的分支中的提交或使用 --amend选项覆盖的提交也可以恢复 （阅读 [数据恢复](https://git-scm.com/book/zh/v2/ch00/_data_recovery) 了解数据恢复）。 然而，任何你未提交的东西丢失后很可能再也找不到了

