# git record

1. aaa

2. change user.name use git config --local

3. git config --local core.editor "" 改变默认的编辑器

4. 初次克隆某个仓库的时候，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。

5. git status -s 查看简短状态命令

6. git diff 比较的是工作目录中当前文件和暂存区快照之间的差异。也就是修改后还没有暂存起来的变化的内容。

7. 若要查看已暂存的将要添加到下次提交里的内容，可以用 git diff --staged 命令。 这条命令将比对已暂存文件与最后一次提交的文件差异。

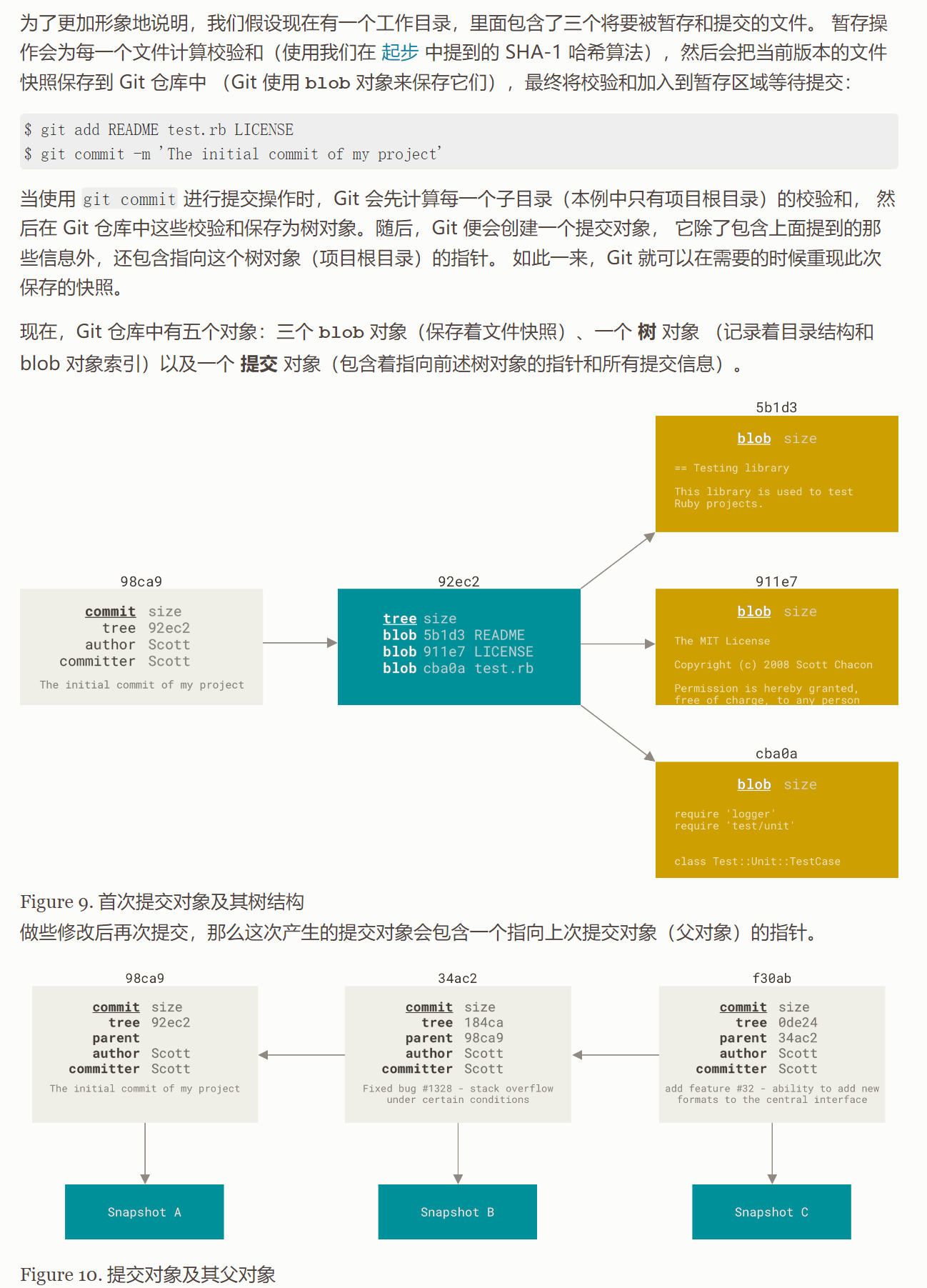
8. git rm 从已跟踪文件清单中删除，即从暂存区移除。这样以后就不会出现在未跟踪文件清单中了。。

9. git diff --staged

10.git fetch <remote> 命令从远程仓库获得数据，但是并不会合并到当前分支。例如远程分支fetch下来后有新的文件，那么在执行fetch后本地磁盘目录中并没有这个文件，。你需要把这个远程分支合并到本地，或者切换到这个远程分支查看。

11.Git 的分支，其实本质上仅仅是指向提交对象的可变指针。 Git 的默认分支名字是 master。 在多次提交操作之后，你其实已经有一个指向最后那个提交对象的 master 分支。 master 分支会在每次提交时自动向前移动。

那么，Git 又是怎么知道当前在哪一个分支上呢？ 也很简单，它有一个名为 HEAD 的特殊指针。 请注意它和许多其它版本控制系统（如 Subversion 或 CVS）里的 HEAD 概念完全不同。 在 Git 中，它是一个指针，指向当前所在的本地分支（译注：将 HEAD 想象为当前分支的别名）。





1. git stash：（stash）会处理工作目录的脏的状态——即跟踪文件的修改与暂存的改动——然后将未完成的修改保存到一个栈上， 而你可以在任何时候重新应用这些改动（甚至在不同的分支上）。