分支的管理

到目前为止，你已经学会了如何创建、合并和删除分支。除此之外，我们还需要学习如何管理分支，在日后的常规工作中会经常用到下面介绍的管理命令。

git branch 命令不仅仅能创建和删除分支，如果不加任何参数，它会给出当前所有分支的清单：

$ git branch

iss53

\* master

testing

注意看 master 分支前的 \* 字符：它表示当前所在的分支。也就是说，如果现在提交更新，master 分支将随着开发进度前移。若要查看各个分支最后一个提交对象的信息，运行 git branch -v：

$ git branch -v

iss53 93b412c fix javascript issue

\* master 7a98805 Merge branch 'iss53'

testing 782fd34 add scott to the author list in the readmes

要从该清单中筛选出你已经（或尚未）与当前分支合并的分支，可以用 --merged 和 --no-merged 选项（Git 1.5.6 以上版本）。比如用 git branch --merged 查看哪些分支已被并入当前分支（译注：也就是说哪些分支是当前分支的直接上游。）：

$ git branch --merged

iss53

\* master

之前我们已经合并了 iss53，所以在这里会看到它。一般来说，列表中没有 \* 的分支通常都可以用 git branch -d 来删掉。原因很简单，既然已经把它们所包含的工作整合到了其他分支，删掉也不会损失什么。

另外可以用 git branch --no-merged 查看尚未合并的工作：

$ git branch --no-merged

testing

它会显示还未合并进来的分支。由于这些分支中还包含着尚未合并进来的工作成果，所以简单地用 git branch -d删除该分支会提示错误，因为那样做会丢失数据：

$ git branch -d testing

error: The branch 'testing' is not fully merged.

If you are sure you want to delete it, run 'git branch -D testing'.

不过，如果你确实想要删除该分支上的改动，可以用大写的删除选项 -D 强制执行，就像上面提示信息中给出的那样。