

软件开发中，总有无穷无尽的新的功能要不断添加进来。

添加一个新功能时，你肯定不希望因为一些实验性质的代码，把主分支搞乱了，所以，每添加一个新功能，最好新建一个 **feature** 分支，在上面开发，完成后，合并，最后，删除该 **feature** 分支。

现在，你终于接到了一个任务：开发代号为 **Vulcan** 的新功能，该功能计划用于下一代星际飞船。

于是准备开发：

```
$ git checkout -b feature-vulcan

Switched to a new branch 'feature-vulcan'
```

5 分钟后，开发完毕：

```
$ git add vulcan.py

$ git status

On branch feature-vulcan

Changes to be committed:

  (use "git reset HEAD <file>..." to unstage)

    new file:   vulcan.py

$ git commit -m "add feature vulcan"

[feature-vulcan 287773e] add feature vulcan

1 file changed, 2 insertions(+)

create mode 100644 vulcan.py
```

切回 **dev**，准备合并：

```
$ git checkout dev
```

一切顺利的话，**feature** 分支和 **bug** 分支是类似的，合并，然后删除。

但是！

就在此时，接到上级命令，因经费不足，新功能必须取消！

虽然白干了，但是这个包含机密资料的分支还是必须就地销毁：

```
$ git branch -d feature-vulcan  
  
error: The branch 'feature-vulcan' is not fully merged.  
  
If you are sure you want to delete it, run 'git branch -D feature-vulcan'.
```

销毁失败。Git 友情提醒，`feature-vulcan` 分支还没有被合并，如果删除，将丢失掉修改，如果要强行删除，需要使用大写的 `-D` 参数。。

现在我们强行删除：

```
$ git branch -D feature-vulcan  
  
Deleted branch feature-vulcan (was 287773e).
```

终于删除成功！

小结

开发一个新 feature，最好新建一个分支；

如果要丢弃一个没有被合并过的分支，可以通过 `git branch -D <name>` 强行删除。