

高级 9：紧急情况：「立即给我打个包，现在马上！」

前面在讲 branch 的时候讲到，利用 branch 可以实现一人多任务的需求，从而可以轻松应对「嘿，这个先别做了，给你个新活」的情况。但有时，尤其是在互联网公司，你可能会遇到比这更紧急的情况：你正对着电脑发呆，忽然见到一个同事屁股着着火就跑来找你了：「快快快，立即给我打个包，现在马上，拜托拜托！」

这种情况和「这个 branch 先放放吧」不同，你没时间、也没必要当场慌慌张张把文件的所有改动做个临时的 commit 然后去救同事的火，救完火再重新把 commit 撤销回来。这时候你只要先把所有文件一股脑扔在一边就可以去给同事打包了，打完包再把刚才扔到一边的文件重新取过来就好。

这一「扔」一「取」，用的是 Git 的 stash 指令。

stash：临时存放工作目录的改动

"stash" 这个词，和它意思比较接近的中文翻译是「藏匿」，是「把东西放在一个秘密的地方以备未来使用」的意思。在 Git 中，stash 指令可以帮你把工作目录的内容全部放在你本地的一个独立的地方，它不会被提交，也不会被删除，你把东西放起来之后就可以去做你的临时工作了，做完以后再来取走，就可以继续之前手头的事了。

具体说来，stash 的用法很简单。当你手头有一件临时工作要做，需要把工作目录暂时清理干净，那么你可以：

```
git stash
```

就这么简单，你的工作目录的改动就被清空了，所有改动都被存了起来。

然后你就可以从你当前的工作分支切到 master 去给你的同事打包了.....

打完包，切回你的分支，然后：

```
git stash pop
```

你之前存储的东西就都回来了。很方便吧？

注意：没有被 track 的文件（即从来没有被 add 过的文件不会被 stash 起来，因为 Git 会忽略它们。如果想把这些文件也一起 stash，可以加上 `-u` 参数，它是 `--include-untracked` 的简写。就像这样：

```
git stash -u
```