

# 封装一些简单的Shell脚本

作者: <https://github.com/linsir6>

原文: <http://www.jianshu.com/p/6f6330b0ab60>

自从上次发完[一篇有关Shell的脚本的文章](#)之后取得了很大程度的反响, 阅读量达到了6280, 喜欢达到了300+, 同时被收入了特别多的专题, 如下图所示, 所以打算再展示几个我封装的简单的脚本。



因为一般热爱编程的人, 大多选择GitHub作为代码管理工具, 我本人更喜欢用命令行来操作GitHub, 然后有一些常用的命令时经常被用到的, 所以可以对它们进行简单的封装, 这样即使每天提交个十几次代码, 也不会很麻烦。

一些常用的操作:

```
cd ~/WorkSpace

git pull

git pull origin master

git status

git branch

git push origin master

git checkout master

git init
git remote add origin url
git pull

git branch --set-upstream-to=origin/master master
```

当然常用的命令肯定不止这些, 不过我们只要掌握好, 简单的封装之后, 就可以很轻松的封装一个命令了。

如果你没有什么Shell方面的基础, 不妨先看看我的另一篇文章 [一篇文章学懂Shell脚本](#), 再返回来看这篇文章。

以一个简单的为例:

1. 我们先新建一个脚本: touch me

2. 给脚本权限: chmod +x me

3. 然后编写指令

```
SHell #!/bin/bash cd ~/WorkSpace
```

1. 如果我们想这个命令在哪里都可以应用, 需要将当前目录添加到系统目录下, 或直接将脚本放在系统文件夹内

然后我们便可以在命令窗口里通过 `. me` 来进入我们的文件夹下面里, 这里面需要加一个 `.` 是因为我们要让效果展示出来, 否则它会内部创建一个子脚本进入, 然后退出的。

自动push的脚本:

```
#!/bin/bash
git pull origin $2
git add .
git commit -m $1
git push origin $2
```

我们需要通过 `push "fix" master` 可以指定描述, 指定执行上传到的分支。

自动pull的脚本:

```
#!/bin/bash
if [ "$1" = "" ]
then
    git branch --set-upstream-to=origin/master master
    git pull
else
    git pull origin $1
fi
```

我们可以通过 `pull` 命令就可以执行 `git pull`，通过 `pull master` 就可以将远程仓库中的`master`分支pull回来。

创建git仓库的脚本:

```
#!/bin/bash
git init
git remote add origin $1
pull
```