

1:新建 shell 脚本

随便使用任意文本编辑器，新建文本，扩展名为 .sh

输入一些代码：

```
1. #!/bin/bash
2. echo "Hello World !"
```

“#!” 是一个约定的标记，它告诉系统这个脚本需要什么解释器来执行，即使用哪一种 Shell。echo命令用于向窗口输出文本。

2: 执行shell脚本

运行Shell脚本有两种方法。

(1) 作为可执行程序

将上面的代码保存为test.sh，并 cd 到相应目录：

```
chmod +x ./test.sh #首先要使脚本具有执行权限
```

```
./test.sh #执行脚本
```

注意，一定要写成./test.sh，而不是test.sh。运行其它二进制的程序也一样，直接写test.sh，linux系统会去PATH里寻找有没有叫test.sh的，而只有/bin, /sbin, /usr/bin, /usr/sbin等在PATH里，你的当前目录通常不在PATH里，所以写成test.sh是会找不到命令的，要用./test.sh告诉系统说，就在当前目录找。

通过这种方式运行bash脚本，第一行一定要写对，好让系统查找到正确的解释器。

这里的“系统”，其实就是shell这个应用程序（想象一下Windows Explorer），但我故意写成系统，是方便理解，既然这个系统就是指shell，那么一个使用/bin/sh作为解释器的脚本是不是可以省去第一行呢？是的。

(2) 作为解释器参数

这种运行方式是，直接运行解释器，其参数就是shell脚本的文件名，如：

```
/bin/sh test.sh
```

```
/bin/php test.php
```

这种方式运行的脚本，不需要在第一行指定解释器信息，写了也没用。

再看一个例子。下面的脚本使用 `read` 命令从 `stdin` 获取输入并赋值给 `PERSON` 变量，最后在 `stdout` 上输出：

1. `#!/bin/bash`
2. `# Author : mozhiyan`
3. `# Copyright (c) http://see.xidian.edu.cn/cpp/linux/`
4. `# Script follows here:`
5. `echo "What is your name?"`
6. `read PERSON`
7. `echo "Hello, $PERSON"`

运行脚本：

```
chmod +x ./test.sh
```

```
$. /test.sh
```

```
What is your name?
```

```
mozhiyan
```

```
Hello, mozhiyan
```

```
$
```