

Shell Here Document（内嵌文档/嵌入文档）

Shell 还有一种特殊形式的重定向叫做“Here Document”，目前没有统一的翻译，你可以将它理解为“嵌入文档”“内嵌文档”“立即文档”。

所谓文档，就是命令需要处理的数据或者字符串；所谓嵌入，就是把数据和代码放在一起，而不是分开存放（比如将数据放在一个单独的文件中）。有时候命令需要处理的数据量很小，将它放在一个单独的文件中有点“大动干戈”，不如直接放在代码中来得方便。

Here Document 的基本用法为：

```
command <<END
  document
END
```

`command` 是 Shell 命令，`<<END` 是开始标志，`END` 是结束标志，`document` 是输入的文档（也就是一行一行的字符串）。

这种写法告诉 Shell 把 `document` 部分作为命令需要处理的数据，直到遇见终止符 `END` 为止（终止符 `END` 不会被读取）。

注意，终止符 `END` 必须独占一行，并且要定顶格写。

分界符（终止符）可以是任意的字符串，由用户自己定义，比如 `END`、`MARKER` 等。分界符可以出现在正常的数据流中，只要它不是顶格写的独立的一行，就不会被作为结束标志。

【实例1】 `cat` 命令一般是从文件中读取内容，并将内容输出到显示器上，借助 Here Document，`cat` 命令可以从键盘上读取内容。

```
[mozhiyan@localhost ~]$ cat <<END
> Shell教程
> http://c.biancheng.net/shell/
> 已经进行了三次改版
> END
Shell教程
http://c.biancheng.net/shell/
已经进行了三次改版
```

`<` 是第二层命令提示符。

正文中也可以出现结束标志 `END`，只要它不是独立的一行，并且不顶格写，就没问题。

```
[mozhiyan@localhost ~]$ cat <<END
> END可以出现在行首
> 出现在行尾的END
> 出现在中间的END也是允许的
> END
END可以出现在行首
出现在行尾的END
出现在中间的END也是允许的
```

【实例2】在脚本文件中使用 Here Document，并将 document 中的内容转换为大写。

```
01.  #!/bin/bash
02.  #在脚本文件中使用立即文档
03.
04.  tr a-z A-Z <<END
05.  one two three
06.  Here Document
07.  END
```

将代码保存到 test.sh 并运行，结果为：

```
ONE TWO THREE
HERE DOCUMENT
```

忽略命令替换

默认情况下，正文中出现的变量和命令也会被求值或运行，Shell 会先将它们替换以后再交给 command，请看下面的例子：

```
[mozhiyan@localhost ~]$ name=C语言中文网
[mozhiyan@localhost ~]$ url=http://c.biancheng.net
[mozhiyan@localhost ~]$ age=7
[mozhiyan@localhost ~]$ cat <<END
> ${name} 已经${age}岁了，它的网址是 ${url}
> END
C语言中文网已经7岁了，它的网址是 http://c.biancheng.net
```

你可以将分界符用单引号或者双引号包围起来使 Shell 替换失效：

```
[mozhiyan@localhost ~]$ name=C语言中文网
[mozhiyan@localhost ~]$ url=http://c.biancheng.net
[mozhiyan@localhost ~]$ age=7
[mozhiyan@localhost ~]$ cat <<'END' #使用单引号包围
```

```
> ${name} 已经${age} 岁了，它的网址是 ${url}
> END
${name} 已经${age} 岁了，它的网址是 ${url}
```

忽略制表符

默认情况下，行首的制表符也被当做正文的一部分。

```
01. #!/bin/bash
02.
03. cat <<END
04.     Shell教程
05.     http://c.biancheng.net/shell/
06.     已经进行了三次改版
07. END
```

将代码保存到 test.sh 并运行，结果如下：

```
Shell教程
http://c.biancheng.net/shell/
已经进行了三次改版
```

这里的制表符仅仅是为了格式对齐，我们并不希望它作为正文的一部分，为了达到这个目的，你可以在 `<<` 和 `END` 之间增加 `-`，请看下面的代码：

```
01. #!/bin/bash
02.
03. #增加了减号-
04. cat <<-END
05.     Shell教程
06.     http://c.biancheng.net/shell/
07.     已经进行了三次改版
08. END
```

这次的运行结果为：

```
Shell教程
http://c.biancheng.net/shell/
已经进行了三次改版
```

总结

如果你尝试在脚本嵌入一小块多行数据，使用 Here Document 是很有用的，而嵌入很大的数据块是一个不好的习惯。你应该保持你的逻辑（你的代码）和你的输入（你的数据）分离，最好是在不同的文件中，除非是输入一个很小的数据集。

Here Document 最常用的功能还是向用户显示命令或者脚本的用法信息，例如类似下面的函数：

```
01. usage() {  
02.     cat <<-END  
03.     usage: command [-x] [-v] [-z] [file ...]  
04.     A short explanation of the operation goes here.  
05.     It might be a few lines long, but shouldn't be excessive.  
06. END  
07. }
```