1、[Shell 教程](http://www.runoob.com/linux/linux-shell.html)

shell 与 shell脚本？

shell环境

Shell 编程只要有一个能编写代码的文本编辑器和一个能解释执行的**脚本解释器**就可以了。

Linux 的 Shell 种类众多，**/bin/bash** 是大多数Linux 系统默认的 Shell，由于易用且免费，在日常工作中被广泛使用。

在一般情况下，人们并不区分 Bourne Shell 和 Bourne Again Shell，所以，像 **#!/bin/sh**，它同样也可以改为 **#!/bin/bash**。（**#!** 告诉系统其后路径所指定的程序即是解释当前脚本文件的 Shell 程序）

shell脚本文件及执行

一般用vi/vim命令创建脚本文件，虽然文件扩展名并不影响脚本执行，但是一般都约定以sh作为文件扩展名。

要想执行test.sh文件，需要先cd到相应目录，再通过“chmod +x ./test.sh” 来使脚本具有执行权限，然后使用"**./**test.sh"来执行脚本。

另外，还可以通过直接运行解释器，并将shell脚本的文件名作为参数来执行脚本，如“/bin/sh test.sh”。（这种方式test.sh不需要在第一行指定解释器信息，写了也没用。另外这种方式test.sh也不需要具有执行权限。）

注意，执行脚本要写成 **./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。

理解什么是脚本语言？ 可参考该链接的笔记部分。

2、[Shell变量](http://www.runoob.com/linux/linux-shell-variable.html)

变量命名**只能**使用英文字母，数字和下划线，且首个字符不能以数字开头。

另外不能使用bash里的关键字（可用help命令查看保留关键字）。

|  |
| --- |
| # 定义变量注意等号两边**不能有空格**  username="zhangsan"  # 通过${变量名}使用一个定义过的变量，大括号用于帮助解释器识别变量的边界，有时候可以省略。  echo **${**username**}** |

|  |
| --- |
| your\_name="tom"  # 已定义的变量，可以被二次赋值，注意这里不是使用变量，不要用$  your\_name="alibaba"  echo $your\_name |

使用 readonly 命令可以将变量定义为只读变量，只读变量的值不能被改变，否则运行会报错。

|  |
| --- |
| username=”zhangsan”  **readonly** myUrl  myUrl=”lisi”     # 不能修改 |

“**unset** variable\_name”可以删除已定义的变量，变量删除后不可再次使用，不能删除只读变量。

运行shell时，会存在三种类型的变量？

2.2、shell字符串

字符串可以用单引号，也可以用双引号，也可以不用引号。

|  |  |
| --- | --- |
| 单引号字符串的限制 | 单引号里的任何字符都会原样输出，单引号字符串中的变量是无效的；  单引号字串中不能出现单引号（对单引号使用转义符后也不行）。 |
| 双引号的优点 | 双引号里可以有变量  双引号里可以出现转义字符 |

|  |
| --- |
| your\_name='jax'  greeting\_1="hello, \"${your\_name}\" !"  greeting\_2="hello, "${your\_name}" !"  greeting\_3="hello, ${your\_name} !"  echo $greeting\_1            # hello, "jax" !  echo $greeting\_2             # hello, jax !  echo $greeting\_3             # hello, jax ! |

|  |
| --- |
| str="abcdcba"  echo ${**#**str}                                         # 用#获取字符串长度            7  echo ${str**:1:4**}                                    # 截取子字符串              bcdc  # 注意下面脚本中外层是反引号`，与~在同一个键位。  echo **`expr index** "$str" cba**`**         # 查找’c或b或a’的位置               1 |

2.4、注释技巧

以"#"开头的行就是注释，会被解释器忽略。sh里没有多行注释，只能每一行加一个#号。

如果在开发过程中，遇到大段的代码需要临时注释起来，过一会儿又取消注释，怎么办呢？每一行加个#符号太费力了，可以把这一段要注释的代码用一对花括号括起来，定义成一个函数，没有地方调用这个函数，这块代码就不会执行，达到了和注释一样的效果。