## 第一节 shell运行

**第一个shell脚本**

打开文本编辑器(可以使用 vi/vim 命令来创建文件)，新建一个文件 test.sh，扩展名为 sh（sh代表shell），扩展名并不影响脚本执行，见名知意就好，如果你用 php 写 shell 脚本，扩展名就用 php 好了。

**输入一些代码，第一行一般是这样：**

***#!/bin/bash***

***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。

**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 #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。

**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。

## 第二节 shell变量

**1 定义变量时**

不加美元符号 （$ 在PHP语言中的变量需要），如：

your\_name="runoob.com"

除了显式地直接赋值，还可以用语句给变量赋值，如：

for file in `ls /etc`

或

for file in $(ls /etc)

*# 以上语句将 /etc 下目录的文件名循环出来。*

**2 使用变量时**

只要在变量名前加上 美元符号$即可，如：

your\_name="qinjx"

echo $your\_name

echo ${your\_name}

又或者：

for skill in Ada Coffe Action Java; do

echo "I am good at ${skill}Script"

done

*#变量名外面的花括号是可选的，加不加都行，加花括号是为了帮助解释器识别变量的边界，如果不给skill变量加花括号，写成echo "I am good at $skillScript"，解释器就会把$skillScript当成一个变量（其值为空），代码执行结果就不是我们期望的样子了。*

**3只读变量**

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。

#!/bin/bash

myUrl="http://www.w3cschool.cc"

readonly myUrl

**4 删除变量**

使用 unset 命令可以删除变量。语法：

unset variable\_name

## 第三节 Shell流程控制

### 1． if else

#### （1）if

if 语句语法格式：

if condition

then

command1

command2

...

commandN

fi

*# 末尾的fi就是if倒过来拼写，后面还会遇到类似的。*

*#在sh/bash里可不能这么写，如果else分支没有语句执行，就不要写这个else。*

#### （2）if else

if else 语法格式：

if condition

then

command1

command2

...

commandN

else

command

fi

### 2 for循环

与其他编程语言类似，Shell支持for循环。 for循环一般格式为：

for var in item1 item2 ... itemN

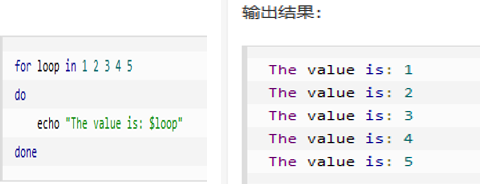
do

command1

command2

...

commandNdone

如：