

Shell 变量

主讲人：杨老师

定义变量时，变量名不加美元符号（\$，PHP语言中变量需要），如：

```
your_name="dajiangtai.com"
```

注意，变量名和等号之间不能有空格，这可能和你熟悉的所有编程语言都不一样。同时，变量名的命名须遵循如下规则：

- 首个字符必须为字母（a-z，A-Z）。
- 中间不能有空格，可以使用下划线（_）。
- 不能使用标点符号。
- 不能使用bash里的关键字（可用help命令查看保留关键字）。

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 /home/hadoop/data/`;do echo $file;done
```

以上语句将 /home/hadoop/data/下目录的文件名循环出来。

Shell变量---使用变量

使用一个定义过的变量，只要在变量名前面加美元符号即可，如：

```
your_name="dajiangtai"
```

```
echo $your_name
```

```
echo ${your_name}
```

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，比如下面这种情况：

```
for course in hadoop storm spark; do
```

```
    echo "I am good at ${course}technology"
```

```
done
```

如果不给course变量加花括号，写成echo "I am good at \${course}technology"，解释器就会把

\$coursetechnology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。

推荐给所有变量加上花括号，这是个好的编程习惯。已定义的变量，可以被重新定义，如：

```
your_name="unix"
```

```
echo $your_name
```

```
your_name="linux"
```

```
echo $your_name
```

这样写是合法的，但注意，第二次赋值的时候不能写\$your_name="linux"，使用变量的时候才加美元符（\$）。

Shell 变量---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。

下面的例子尝试更改只读变量，结果报错：

```
#!/bin/bash
```

```
myUrl="http://www.dajiangtai.com/"
```

```
readonly myUrl
```

```
myUrl="http://hadoop2.dajiangtai.com/index.do"
```

运行脚本，结果如下：

```
./testURL.sh: line 4: myUrl: readonly variable
```

Shell 变量---删除变量

使用 unset 命令可以删除变量。语法：

```
unset variable_name
```

变量被删除后不能再次使用。unset 命令不能删除只读变量。

实例

```
#!/bin/sh
```

```
myUrl="http://www.dajiangtai.com/"
```

```
unset myUrl
```

```
echo $myUrl
```

以上实例执行将没有任何输出。

运行shell时，会同时存在三种变量：

- 1) 局部变量 局部变量在脚本或命令中定义，仅在当前shell实例中有效，其他shell启动的程序不能访问局部变量。
- 2) 环境变量 所有的程序，包括shell启动的程序，都能访问环境变量，有些程序需要环境变量来保证其正常运行。必要的时候shell脚本也可以定义环境变量。
- 3) shell变量 shell变量是由shell程序设置的特殊变量。shell变量中有一部分是环境变量，有一部分是局部变量，这些变量保证了shell的正常运行

字符串是shell编程中最常用最有用的数据类型（除了数字和字符串，也没啥其它类型好用了），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

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；

单引号字符串中不能出现单引号（对单引号使用转义符后也不行）。

示例：

```
str='I love dajiangtai'
```

```
echo $str
```

```
str2='I love dajiangtai $str'
```

```
echo $str2
```

```
str3='I 'love' dajiangtai'
```

```
echo $str3
```

● 双引号

```
your_name='dajiangtai'
```

```
str="Hello Shell, Hello \"${your_name}\"! \n "
```

```
echo $str
```

双引号的优点：

- 1) 双引号里可以有变量
- 2) 双引号里可以出现转义字符

- 拼接字符串

```
your_name="dajiangtai"
```

```
welcome="hello, "$your_name" ! "
```

```
welcome_1="hello, ${your_name} ! "
```

```
echo $welcome $welcome_1
```

Shell 字符串---获取字符串长度

- 获取字符串长度

```
course="hadoop"
```

```
echo ${#course}
```

- 提取子字符串

以下实例从字符串第 2 个字符开始截取 4 个字符：

```
welcome="I love dajiangtai !"
```

```
echo ${welcome:1:4}
```

- 查找子字符串

查找字符 "o" 的位置：

```
welcome="I love dajiangtai !"
```

```
echo `expr index "$welcome" o`
```

注意：以上脚本中 "`" 是反引号，而不是单引号 "'"，不要看错了哦。

bash支持一维数组（不支持多维数组），并且没有限定数组的大小。类似与C语言，数组元素的下标由0开始编号。获取数组中的元素要利用下标，下标可以是整数或算术表达式，其值应大于或等于0。

● 定义数组

在Shell中，用括号来表示数组，数组元素用"空格"符号分割开。定义数组的一般形式为：数组名=(值1 值2 ... 值n)

例如：

```
array_name=(value0 value1 value2 value3)
```

或者

```
array_name=(
```

```
value0
```

```
value1
```

```
value2
```

```
value3
```

```
)
```

还可以单独定义数组的各个分量：

```
array_name[0]=value0
```

```
array_name[1]=value1
```

```
array_name[n]=valuen
```

可以不使用连续的下标，而且下标的范围没有限制。

● 读取数组

读取数组元素值的一般格式是：

`${数组名[下标]}`

例如：

`valuen=${array_name[n]}`

使用@符号可以获取数组中的所有元素，例如：

`echo ${array_name[@]}`

示例：

`course=(hadoop storm spark)`

`echo ${course[@]}`

● 获取数组长度

获取数组长度的方法与获取字符串长度的方法相同，例如：

取得数组元素的个数

```
length=${#array_name[@]}
```

或者

```
length=${#array_name[*]}
```

取得数组单个元素的长度

```
lengthn=${#array_name[n]}
```

示例：

```
course=(hadoop storm spark)
```

```
length=${#course[@]}或者length=${#course[*]}
```

```
length=${#course[2]}
```

```
echo ${length}
```


● 注释

- 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。
- 脚本中需要有基本的注释信息方便以后维护，如参数、执行示例、脚本存放位置、日志文件地址、创建人、创建日期等信息。
- 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。

● 脚本注释示例：

```
#####  
#说明：将用户数据从HDFS导入MySQL  
#参数CONNECTURL          说明：MySQL 连接地址  示例：jdbc:mysql://dajiangtai/djtdb_test  
#参数USERNAME            说明：MySQL 账号  示例：root  
#参数PASSWORD            说明：MySQL 密码  示例：root  
#执行示例： sh /home/hadoop/app/djt/sh/sqoop/sq_ods_d_djt_user_copy.sh  
#脚本存放地址：/home/hadoop/app/djt/sh/sqoop  
#日志文件地址：/home/hadoop/app/djt/sh/log  
#创建人：dajiangtai  
#创建日期：20161201  
#最新修改日期：20170101  
#修改人、修改时间、修改内容：XXXXXXXXXXXXXXXXXX  
#修改人、修改时间、修改内容：XXXXXXXXXXXXXXXXXX  
#####
```



THANKS

黄丽老师 : 3354223855

小夏老师 : 972628726