

## 定义变量

定义变量时，变量名不加美元符号（\$），如：

```
1. variableName="value"
```

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）。

tip:shell变量存储的都是字符串

例如：

```
a=1+4
```

a存储的是：1+4 而不是数字 5

变量定义举例：

```
1. myUrl="http://see.xidian.edu.cn/cpp/linux/"  
2. myNum=100
```

## 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：

```
1. your_name="mozhiyan"  
2. echo $your_name  
3. 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：

```
1. for skill in Ada Coffe Action Java  
2. do  
3.   echo "I am good at ${skill}Script"  
4. done
```

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。

推荐给所有变量加上花括号，这是个好的编程习惯。

## 重新定义变量

已定义的变量，可以被重新定义，如：

1. myUrl="http://see.xidian.edu.cn/cpp/linux/"
2. echo \${myUrl}
3. myUrl="http://see.xidian.edu.cn/cpp/shell/"
4. echo \${myUrl}

这样写是合法的，但注意，第二次赋值的时候不能写

\$myUrl="http://see.xidian.edu.cn/cpp/shell/"，使用变量的时候才加美元符（\$）。

## 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：

1. #!/bin/bash
2. myUrl="http://see.xidian.edu.cn/cpp/shell/"
3. **readonly myUrl**
4. myUrl="http://see.xidian.edu.cn/cpp/danpianji/"

运行脚本，结果如下：

/bin/sh: NAME: This variable is read only.

## 删除变量

使用 unset 命令可以删除变量。语法：

1. **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。

举个例子：

1. #!/bin/sh
2. myUrl="http://see.xidian.edu.cn/cpp/u/xitong/"
3. **unset myUrl**

#### 4. echo \$myUrl

上面的脚本没有任何输出。

### 变量类型

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