**定义变量**

your\_name="runoob.com"

### **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="qinjx"

echo $your\_name

echo ${your\_name}

变量名外面的花括号是可选的，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 skill in Ada Coffe Action Java; do

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

done

### **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.google.com"

readonly myUrl

myUrl="http://www.runoob.com"

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：

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。

### **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
2. **环境变量**

**3) shell变量** shell变量是由shell程序设置的特殊变量。

## 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**

str='this is a string'

单引号字符串的限制：

单引号字符串中的变量是无效的；

### **双引号**

your\_name='runoob'

str="Hello, I know you are \"$your\_name\"! \n"

echo -e $str

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

### **拼接字符串**

your\_name="runoob"# 使用双引号拼接

greeting="hello, "$your\_name" !"

greeting\_1="hello, ${your\_name} !"

echo $greeting $greeting\_1# 使用单引号拼接

greeting\_2='hello, '$your\_name' !'

### **获取字符串长度**

string="abcd"

echo ${#string} #输出 4

### **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：

string="runoob is a great site"

echo ${string:1:4} # 输出 unoo

### **查找子字符串**

查找字符 **i** 或 **o** 的位置(哪个字母先出现就计算哪个)：

string="runoob is a great site"

echo `expr index "$string" io` # 输出 4

**注意：** 以上脚本中 **`** 是反引号，而不是单引号 **'**，不要看错了哦。

## Shell 数组

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。

### **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

### **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[@]}

### **获取数组的长度**

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

## Shell 注释

以 **#** 开头的行就是注释，会被解释器忽略。

通过每一行加一个 **#** 号设置多行注释，像这样：

#--------------------------------------------

# 这是一个注释

### **多行注释**

多行注释还可以使用以下格式：

:<<EOF注释内容...注释内容...注释内容...

EOF

EOF 也可以使用其他符号:

:<<'

注释内容...

注释内容...

注释内容...

'

:<<!注释内容...注释内容...注释内容...!