

shell自动化编程之数组

-Author: 八度

-Email: hlions@163.com

-GitHub: <https://github.com/hlions>

1、数组定义

declare定义

使用declare -a 命令定义数组（数组的索引是从0开始计数的），接下来就可以通过[]操作符为不同索引位置的元素赋值。

```
declare -a names
names[0]=tom
names[1]=jack
```

在定义数组的同时，可以同时赋值，用()表示，各元素之间用空格隔开

```
#定义的同时直接赋值
declare -a names=(tom jack)
#增加元素
names[2]=sue
```

直接定义

创建数组最简单的方法是使用()直接定义数组，括号中元素用空格隔开；在括号中也可以声明下标。实例：

#使用()直接数组

```
days1=(one two three four five)
```

#或者在()中声明下标，默认从0开始

```
days2=([0]='one' [1]='two' [2]='three' [3]='four')
```

#下标可以不连续

```
days3=([0]='one' [2]='three')
```

带下标定义

使用[]操作符，为每一个指定下标赋值，下标也可以不连续

```
header[0]='user'
```

```
header[1]='pid'
```

```
header[2]='%CPU'
```

```
header[3]='%MEM'
```

从文件读取数组

```
days=(`cat days.txt`)
```

2、数组操作

赋值

指定下标，使用[]操作符为元素赋值，当下标不存在时，相当于为数组增加元素

```
declare -a names
```

```
names[0]=tom
```

取值

```
days=(one two three four five)
#取出第1个元素
echo ${days[0]}
#取出第3个元素
echo ${days[2]}
```

```
#!/bin/bash
#数组取值

days=(one two three four)
echo ${days}
echo ${days[0]}
echo ${days[1]}
echo ${days[@]}
echo ${days[*]}

echo "*****"
names=( )
names[1]=tom
names[2]=jack
echo ${names}
echo ${names[1]}
echo ${names[@]}
```

长度

利用 "@" 或 "*" 字符，将数组扩展成列表，然后使用 "#" 来获取数组元素的个数

```
#!/bin/bash
#数组个数

days=(one two three four)
echo "数组days的元素个数为: ${#days[@]}"

echo "*****"
names=( )
names[1]=tom
names[2]=jack
echo "数组names的元素个数为: ${#names[*]}"
```

遍历

```
#!/bin/bash
#数组遍历

days=(one two three four)
#for循环遍历
for day in ${days[*]} #或${days[@]}
do
    echo $day
done
echo "*****"

#for循环带下标遍历
for i in ${!days[@]}
do
    echo ${days[$i]}
done

echo "*****"
```

删除

删除一个数组或数组中元素用unset命令

`unset 数组名[索引] #删除索引下的元素`

`unset 数组名 #删除整个数组`

连接

用`()`将多个数组连接在一起，`()`中各个数组用空格隔开。

```
days=(one two three four)
names=(tom jack)
days=(${days[@]} ${names[@]})
```