

3.6 Shell 数组_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 3.6 Shell 数组涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

根据我们学习 C 语言的经验，数组就是一系列数据的集合，这个数据就是我们之前学习的存储单个元素的最小单元变量，也就是说将一些列的元素整合到一个集合内，这个集合的名称就叫数组。当然与其他语言一样，数组具备几个条件，在 Shell 中数组仅支持一维数组，数组元素的下标从 0 开始，数组元素没有最大限制等。

在我们之前学习 C 语言的过程中，大家应该有所体会，当我们操作批量数据的时候，一个一个变量操作非常不便，此时我们可以使用一个数组集合，对整个数组集合进行遍历或其他操作，最终实现批量的效果，数组使得我们的脚本更具扩展性。

数组的定义有两种方式，可分为直接定义和单元素定义。

数组类似于变量定义，只不过将里面的值用小括号括起来，其中每个元素使用空格分隔。shell 数组中元素的类型可以不一样，例如其中可以包含数字与字符串。

示例:

```
SHELL_ARRAY=(1 2 3 "hello Shell")
```

说明: SHELL_ARRAY 为数组名

1, 2, 3 为保存的数字, "hello Shell" 为保存的字符串

Shell 中数组下标从 0 开始，利用单个元素来定义数组。操作和 C 语言类似

示例:

```
[linux@linux]# SHELL_ARRAY[0]=1
[linux@linux]# SHELL_ARRAY[1]=2
[linux@linux]# SHELL_ARRAY[2]=3
[linux@linux]# SHELL_ARRAY[3]="hello Shell"
```

与变量的引用一样，数组可以获取单个位置的元素，利用 `${SHELL_ARRAY[num]}` 。

示例用法:

```
[linux@linux]# echo ${SHELL_ARRAY[0]}           //获取AEG1数组中第一个元素
1
[linux@linux]# echo ${SHELL_ARRAY[3]}           //获取AEG1数组中第四个元素
hello Shell
```

获取数组全部元素使用 `${SHELL_ARRAY[*]}` 或 `${SHELL_ARRAY[@]}` 。

示例用法:

```
[linux@linux]# echo ${SHELL_ARRAY[@]}
1 2 3 hello Shell

[linux@linux]# echo ${SHELL_ARRAY[*]}
1 2 3 hello Shell
```

- 获取数组长度

- 获取整个数组长度

- 数组长度及数组中元素的个数，可以利用 `${#SHELL_ARRAY[*]}` 或 `${#SHELL_ARRAY[@]}`，我们发现其实就是在获取数组全部元素前添加 `#` 来获取数组个数。

- ```
[linux@linux]# echo ${SHELL_ARRAY[*]}
4

[linux@linux]# echo ${SHELL_ARRAY[@]}
4
```
- 获取单个元素的长度
- 对于数组中的单个元素我们也可以进行长度的获取，可以利用 `${#SHELL_ARRAY[num]}`。
- ```
[linux@linux]# echo ${SHELL_ARRAY[@]}
100 2 3 hello Shell

[linux@linux]# echo ${SHELL_ARRAY[3]} //获取第四个元素内容为: hello Shell
hello Shell

[linux@linux]# echo ${#SHELL_ARRAY[3]} //获取四个元素长度为11
11
```

- 操作数组

- 增加

- 对数组元素的增加，直接对单个位置元素增加即可，例如：

- ```
[linux@linux]# SHELL_ARRAY[4]=10
[linux@linux]# echo ${SHELL_ARRAY[@]}
100 2 3 hello Shell 10
[linux@linux]# echo ${#SHELL_ARRAY[@]}
5
```

- 删除

- 删除数组可以使用 `unset`，`unset SHELL_ARRAY[num]` 可以删除对应下标的数组元素，如果不带下标则删除数组的全部元素，例如：

- ```
[linux@linux]# echo ${SHELL_ARRAY[@]}
100 3 hello Shell 10
[linux@linux]# unset SHELL_ARRAY[0] //删除下标为0的元素
[linux@linux]# echo ${SHELL_ARRAY[@]}
3 hello Shell 10
[linux@linux]# unset SHELL_ARRAY //删除整个数组元素
[linux@linux]# echo ${SHELL_ARRAY[@]}
```

- 关联数组

上述教大家使用的都是我们的普通数组，shell 中还提供了一种关联性数组，在使用关联数组前，必须先使用 `declare -A` 声明它。

示例代码:

```
[linux@linux]# declare -A ARGFILE //定义管理!
[linux@linux]# ARGFILE=[name1]=Shell [name2]=linux [name3]=arg //关联数组元素赋值
[linux@linux]# echo ${ARGFILE[@]} //查看所有:
arg linux Shell //输出顺序和定义的顺序无关

[linux@linux]# echo ${#ARGFILE[@]}
3

[linux@linux]# echo ${ARGFILE[name1]} //查看索引为name1的元素值
Shell

[linux@linux]# echo ${!ARGFILE[@]} //查看所有的key值
name3 name2 name1
```

```
#!/bin/bash

array=(1 2 3 4 5)
i=0
for var in ${array[@]}
do
    echo "array[$i] = $var"
    i=`expr $i + 1`
done

declare -A ARGLIST

ARGLIST[key1]=1
ARGLIST[key2]=2
ARGLIST[key3]="shell"

for item in ${!ARGLIST[@]}
do
    echo "${item} ---> ${ARGLIST[$item]}"
done
```

要求用循环输出下列数组中元素的下标。

```
array=(7 5 6 3 2)
```

定义下列数组，要求用 for 循环打印下面这句话中字母个数不大于 6 的单词。

```
array=(I am westos teacher welcome to westos training class)
```

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

