

# Lua 数组

数组，就是相同数据类型的元素按一定顺序排列的集合，可以是一维数组和多维数组。

Lua 数组的索引键值可以使用整数表示，数组的大小不是固定的。

## 一维数组

一维数组是最简单的数组，其逻辑结构是线性表。一维数组可以用for循环出数组中的元素，如下实例：

```
array = {"Lua", "Tutorial"}

for i= 0, 2 do
    print(array[i])
end
```

以上代码执行输出结果为：

```
nil
Lua
Tutorial
```

正如你所看到的，我们可以使用整数索引来访问数组元素，如果知道的索引没有值则返回nil。

在 Lua 索引值是以 1 为起始，但你也可以指定 0 开始。

除此外我们还可以以负数为数组索引值：

```
array = {}

for i= -2, 2 do
    array[i] = i *2
end

for i = -2,2 do
    print(array[i])
end
```

以上代码执行输出结果为：

```
-4
-2
0
```

```
2
4
```

## 多维数组

多维数组即数组中包含数组或一维数组的索引键对应一个数组。

以下是一个三行三列的阵列多维数组：

```
-- 初始化数组
array = {}
for i=1,3 do
    array[i] = {}
    for j=1,3 do
        array[i][j] = i*j
    end
end

-- 访问数组
for i=1,3 do
    for j=1,3 do
        print(array[i][j])
    end
end
```

以上代码执行输出结果为：

```
1
2
3
2
4
6
3
6
9
```

不同索引键的三行三列阵列多维数组：

```
-- 初始化数组
array = {}
maxRows = 3
maxColumns = 3
for row=1,maxRows do
    for col=1,maxColumns do
        array[row*maxColumns + col] = row*col
    end
end
```

```
    end
end

-- 访问数组
for row=1,maxRows do
    for col=1,maxColumns do
        print(array[row*maxColumns +col])
    end
end
end
```

以上代码执行输出结果为：

```
1
2
3
2
4
6
3
6
9
```

正如你所看到的，以上的实例中，数组设定了指定的索引值，这样可以避免出现 nil 值，有利于节省内存空间。

[← Lua 字符串](#)[Lua 迭代器 →](#)[✎ 点我分享笔记](#)