

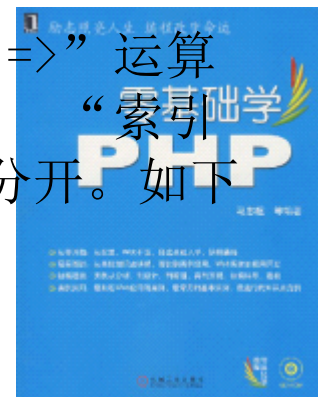
第4章 PHP对数组的处理

- PHP的一个优势就是提供了丰富的函数，用来处理各种类型的数据、完成一些相对复杂、经常性、重复性多或者和底层有关的操作。这些函数都可以在程序中直接使用。从本章开始，本书就为读者介绍这些PHP内置函数的用法。这一章主要讲述一些和数组处理有关的常用PHP函数。



4.1 建立数组的方法

- 在第2章介绍数据类型时，已经介绍过建立数组的方法。本节对建立数组的方法做进一步的说明。使用 PHP 的 `array()` 语言结构可以创建一个数组，如下代码所示。
- ```
$ms_office =
array("word", "excel", "outlook", "access");
```
- 上述代码创建了一个名叫 `$ms_office` 的数组，它包含 4 个单元，每个单元存储 1 个值，这些值是：word、excel、outlook 和 access，这些值也被称为数组元素。注意，这里的 `array()` 不是函数，它只是 PHP 的语言结构。
- 数组的索引又叫“键值”或者“下标”。使用“`=>`”运算符，可以为数组指定索引和值。它的语法格式是：“索引`=>`值”，每对“索引`=>`值”之间使用逗号“`,`”分开。如下代码使用“`=>`”为数组指定索引和值。



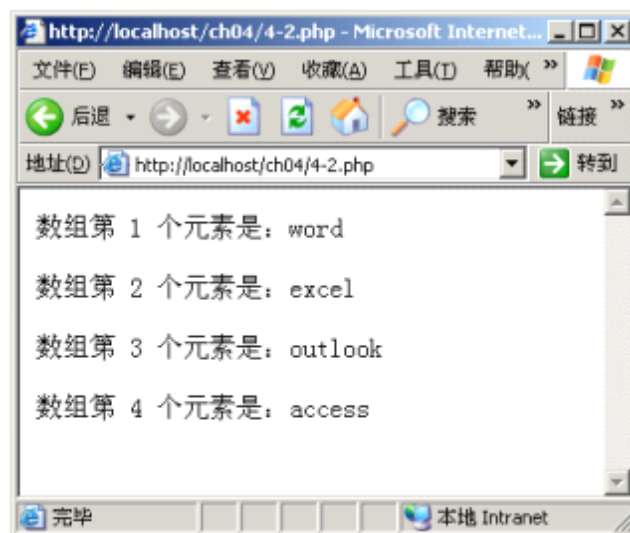
## 4.2 输出数组元素的方法

- 在实际开发中，经常要访问数组的值。如果每次都按数组索引来取数值元素，会非常麻烦。在PHP中可以通过各种循环来快速访问数组元素。本节就介绍常用的for和foreach。



## 4.2.1 使用for循环语句输出数组元素

- 对于一个按整数顺序索引的数组，可以通过for循环语句，来依次访问数组元素，



用 for 循环访问数组元素





### 4.2.3 使用print\_r函数显示数组元素

- 有时，为了在程序中查看一个数组的结构（即索引和值的对应关系），需要将数组结构显示到页面上。这可以通过PHP的print\_r函数实现。
- HTML中的“<pre>”和“</pre>”标签，可以把它们之间文本中的空格、回车、换行、tab键表现出来，即按照文本原先的布局显示。在代码4-6中，调用函数print\_r之后，也可以加上输出结束标签“</pre>”的代码。



## 4.3 计算数组元素个数

- 在PHP开发中，经常遇到对数组元素个数的计算，通常也称之为计算数组的大小。使用函数count计算数组元素的个数，该函数语法如下所示。该函数的返回值为一整数，即数组元素的个数。
- `int count(mixed $var);`
- 上述语法格式中的“mixed”表示参数var可以是多种不同的数据类型，这意味着，该函数不仅仅可以用来计算数组元素个数。“mixed”这种表示方法会在本书后续内容中继续使用。



## 4.4 对数组进行分割、合并

- 数组的分割是指将一个数组拆分成一个或多个数组，而数组的合并，可以看成是分割的逆运算，即将多个数组组合成一个新的数组。下面分别介绍和分割、合并数组有关的PHP内置函数。





## 4.4.1 实现数组分割

- 使用函数 `array_chunk` 可以对一个数组进行分割，该函数语法如下所示。
- `array_chunk(array $input_array, int $size, bool $preserve_key);`
- 该函数接受3个参数，`$input_array` 是要分割的数组（即原数组），`$size` 表示原数组被分割后，每个数组中元素的个数，`$preserve_key` 是可选参数，它为 `TRUE` 时，分割后的每个数组的索引使用原数组的索引名；它为 `FALSE` 时，分割后的每个数组的索引都将使用从 0 开始的整数索引。该函数的返回值是一个数组，这个数组的索引从 0 开始，该数组的每个元素仍然是一个数组，即原数组被分割后的一个数组。



## 4.4.2 实现数组合并

- 使用函数array\_merge可以合并一个或多个数组，该函数的语法如下所示。
- array array\_merge(\$array...);
- 该函数接受一个或多个数组作为参数，PHP5中，它只接受array类型的参数。该函数返回一个合并后的数组。



A screenshot of a Microsoft Internet Explorer browser window. The address bar shows 'http://localhost/ch04/4-9.php'. The main content area displays the output of the array\_merge function, which is an array containing five elements: Earth, Venus, Mars, Jupiter, and Saturn, indexed from 0 to 4. The browser's status bar at the bottom shows '本地 Intranet'.

```
Array
(
 [0] => Earth
 [1] => Venus
 [2] => Mars
 [3] => Jupiter
 [4] => Saturn
)
```

使用 array\_merge 合并数组



## 4.5 处理数组元素和键值位置

- PHP的每个数组内部均会有一个指针，指向当前元素。这里所谓的指针，可以把它理解成程序即将要处理的元素所在的位置。处理数组元素和键值（索引）的位置，就是获取数组某个位置上的元素，或获取该元素后，将该指针移动到其他位置。PHP中，处理数组元素和键值位置的函数主要有以下几个。
- `current`：该函数返回数组的当前元素，不移动当前指针位置。
- `next`：该函数返回当前元素的下一个元素的值，并将指针向后移动一位，下一个元素不存在时，返回 `FALSE`。
- `prev`：该函数返回当前元素的上一个元素的值，并将指针向前移动一位，上一个元素不存在时，返回 `FALSE`。
- `end`：该函数返回数组的最后一个元素的值。



## 4.6 对数组进行排序

- 对数组排序实际指的是对数组元素的排序。使用PHP的有关内置函数，可以对数组元素按多种方式进行排序，下面分别介绍这些函数的用法。



## 4.6.1 使用sort对数组进行排序

- 使用函数sort对数组元素按顺向排序，即按字母由前向后或按数字由小到大排序。该函数的语法如下所示。
- `bool sort(array &$arr [,int $sort_flag]);`
- 该函数接受一个数组作为输入参数，如果处理成功，该函数返回TRUE，否则，该函数返回FALSE。该函数将会删除所要排序的数组原有的索引值，从而为该数组使用新的索引值。代码4-11演示了如何使用该函数为数组元素排序。



## 4.6.2 使用rsort对数组进行逆向排序

- 使用函数rsort对数组元素做逆向排序，即按字母从后到前或按数字由大到小排序。该函数的语法如下所示。
- `bool rsort(array &$arr [,int $sort_flag]);`
- 该函数接受一个数组作为输入参数，如果处理成功，函数返回TRUE，否则返回FALSE。该函数将会删除所要排序的数组原有的索引值，从而为该数组使用新的索引值。



### 4.6.3 排序时保持原数组的索引

- 前面介绍的两个函数sort和rsort，都会为要排序的数组重新建立索引。有时，对数组排序后，仍需要保持原数组的索引，对于这种情况，在PHP中可以使用如下两个函数。
- asort：该函数对数组元素进行排序，并且保持数组的原索引关系不变。该函数仍然对数组元素按顺向排序。如果处理成功，该函数返回TRUE，否则返回FALSE。
- ksort：该函数对数组元素按索引名顺序排序，并且保持数组的原索引关系不变，如果处理成功该函数返回TRUE，否则返回FALSE。该函数主要用于关联数组。



## 4.6.4 数组的随机排序

- PHP提供一个随机排序数组元素的函数 `shuffle()`，该函数的语法如下所示。
- `bool shuffle (array $array)`
- 该函数为参数数组 `$array` 中的元素赋予新的键名，这将删除原有的键名而不仅仅是重新排序。如果执行成功该函数返回 `TRUE`。





## 4.6.5 数组的反向排序

- 本节最后，介绍一个可以将数组元素按原顺序反向排序的函数，该函数是`array_reverse()`。其语法如下所示。
- `array array_reverse (array $array [, bool $preserve_keys])`
- 该函数返回一个与原数组单元顺序相反的数组，原数组以参数`$array`传入该函数，该函数的第2个参数是可选的，如果第2个参数`$preserve_keys`设置为`TRUE`，则原数组的键名将会被保留。



## 4.7 重置一个数组

- PHP中，重置一个数组指的是，将数组中的内部指针移动到该数组的第一个元素所在的单元。使用函数reset可以完成对数组的重置。该函数的语法如下所示。
- `mixed reset(array &$arr);`
- 该函数接受一个数组类型的变量，将函数内部指针指向数组第一个单元，并将第一个单元的元素返回。如果数组为空，该函数返回FALSE。



## 4.8 用数组对变量赋值

- PHP中，可以将数组的元素作为值赋给变量，使用语言结构list()可以完成这个功能。严格意义上讲，list()和array()一样，不是函数。
- 在上述程序中，通过list，将数组\$planet的第一个元素'Earth'赋给了变量\$no1，将第二个元素赋给了变量\$no2，将第3个元素赋给了变量\$no3。



## 4.9 快速创建数组

- 函数`range()`可以快速创建制定元素范围的数组。其语法如下所示。
- `array range(mixed $low, mixed $high [, number $step])`
- 函数`range()`返回一个数组，其中元素为参数从`$low`到`$high`的序列，包括它们本身。例如`range(1, 9)`相当于`array(1, 2, 3, 4, 5, 6, 7, 8, 9)`，如果`$low > $high`，则序列将从`$high`到`$low`。该函数的第3个参数`$step`是可选的，如果给出了`step`的值，它将被作为元素之间的跨度值。`step`应该为正值。如果未指定，`step`则默认为1。



## 4.10 压入和弹出数组元素

- 这里所说的压入是指，向数组的末尾增加一个元素。而弹出是指，将数组中的最后一个元素取出。PHP提供两个函数分别完成这两个功能，这两个函数是`array_push()`和`array_pop()`。



## 4.10.1 压入数组元素

- 函数array\_push()将一个或多个元素压入数组的末尾，其语法如下所示。
- `int array_push(array $array, mixed $var)`
- 该函数将参数\$array当成一个数据栈，并将传入的参数\$var压入\$array的末尾。array的长度将根据入栈变量的数目增加，执行正常时，该函数返回数组\$array新的单元总数。



## 4.10.2 弹出数组元素

- 函数array\_pop()将数组最后一个元素弹出，其语法如下所示。
- mixed array\_pop(array \$array)
- 该函数弹出并返回参数\$array数组的最后一个单元，并将数组\$array的长度减一。如果参数\$array为空（或者不是数组），该函数将返回NULL。需要注意的是，使用本函数后会重置数组指针，相当于使用了函数reset()。



## 4.11 改变数组字符索引名的大小写

- PHP提供了一个函数，可以将以字符串作为数组索引（键名）的大小写全部更换，这个函数是 `array_change_key_case()`，该函数的语法如下所示。
- `array array_change_key_case(array $input [, int $case])`
- 函数 `array_change_key_case()` 将参数 `$input` 数组中的所有索引改为全小写或大写。可选参数 `$case` 用来指定大小写转换方式，它为两个常量值：`CASE_UPPER` 和 `CASE_LOWER`。没有指定该参数时，该函数使用的默认值是 `CASE_LOWER`。





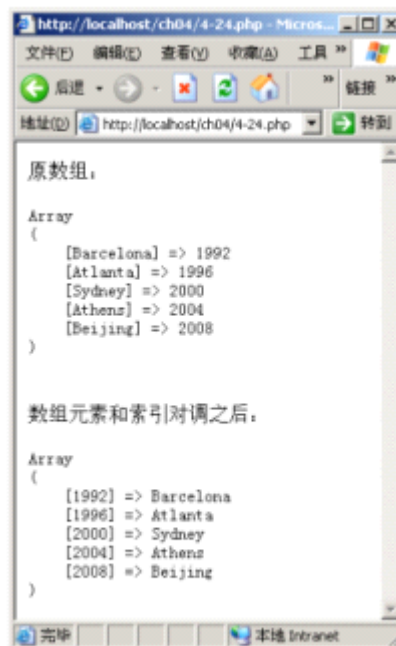
## 4.12 对数组的集合处理

- 函数array\_intersect()可以用来计算数组的交集，其语法如下所示。
- `array array_intersect(array $arr1, array $arr2 [, $arr ...])`
- 函数array\_intersect()返回一个数组，该数组包含了在参数数组\$arr1中出现，同时也出现在所有其它参数数组中的值，注意键名保留不变。



## 4.13 交换数组索引和元素

- PHP中的数组可以通过函数实现索引和元素的交换，这对程序实现某些操作带来极大的便利，这个函数就是array\_flip()，该函数的语法如下所示。



```
原数组:
Array
(
 [Barcelona] => 1992
 [Atlanta] => 1996
 [Sydney] => 2000
 [Athens] => 2004
 [Beijing] => 2008
)

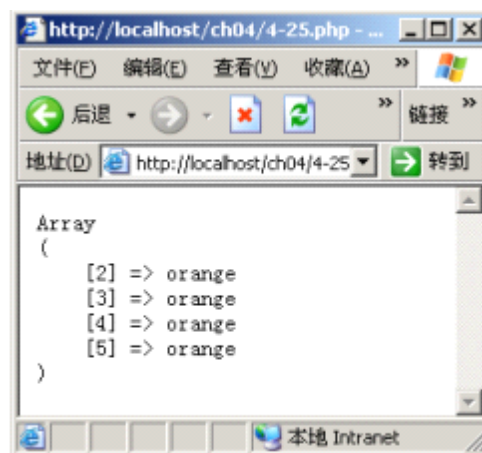
数组元素和索引对调之后:
Array
(
 [1992] => Barcelona
 [1996] => Atlanta
 [2000] => Sydney
 [2004] => Athens
 [2008] => Beijing
)
```

数组元素和索引的对换



## 4.14 快速填充数组元素

- PHP提供函数快速填充数组，将数组中的某些或所有元素都置成通一个值，这个函数是 `array_fill()`，其语法如下所示。
- `array array_fill(int $start, int $num, mixed $value)`
- 函数 `array_fill()` 有3个参数：其中参数 `$start` 指定要填充元素的开始索引，参数 `$num` 表示将 `$num` 个数组元素以参数 `$value` 填充。



快速填充数组元素



## 4.15 统计数组元素出现次数

- 函数`count()`用来计算整个数组元素的个数，而函数`array_count_values()`用来统计数组中各元素出现的次数，即数组中各元素的个数。函数`array_count_values()`的语法如下所示。
- `array array_count_values(array $arr)`
- 该函数返回一个数组，这个数组把参数数组`$arr`中的元素作为索引，元素出现的次数作为对应索引的值。
- 函数`array_count_values()`统计出数组`$arr`中，各个元素出现的次数，而且从这个执行结果可以看出，函数`array_count_values()`统计元素时，是元素大小写的，比如Say和say会分别被统计。



## 4.16 检查数组索引是否存在

- PHP提供函数array\_key\_exists()检查给定的索引是否存在于数组中，该函数语法如下所示。
- `bool array_key_exists(mixed $key, array $search)`
- 该函数查找由参数\$key指定的索引，是否存在于由参数\$search指定的数组中，如果存在，该函数返回TRUE，否则或者出错时，该函数返回FALSE。参数\$key可以是任何能作为数组索引的值。



## 4.17 取得数组中的所有索引

- PHP提供函数array\_keys()用来获取数组中所有的索引名，该函数语法如下所示。
- `array array_keys(array $arr [, mixed $search_value [, bool $strict]])`
- 该函数会将参数数组\$arr中，所有的索引名返回到一个数组中。参数search\_value是可选的，如果指定该参数，则函数array\_keys只返回该元素值对应的索引名，可选参数\$strict用来进行全等比较。



## 4.18 用回调函数处理数组

- PHP支持回调函数操作给定数组的元素上，这也是PHP数组函数的强大功能之一。函数array\_map()就可以完成这样的工作，它将回调函数作用到给定数组的元素上。其语法如下所示。
- array array\_map(callback\_func, array \$arr1 [, array ...])
- 函数array\_map()返回一个数组，该数组包含了参数\$arr1中的所有元素经过函数callback\_func处理过之后的元素。callback接受参数的数目应该和传递给array\_map()函数的数组数目一致。



## 4.19 其他数组处理函数

- 这节介绍两个在实际应用中经常会用到的两个PHP内置函数：in\_array()和key()。下面分别加以介绍。
- in\_array: 该函数可以判断某个值是否存在于数组中。如果在数组中找到该值，函数返回TRUE，否则返回FALSE。
- key: 该函数可以获取数组中当前单元的索引值。





## 4.19.1 判断某个值是否存在于数组

- 代码4-30演示了函数in\_array的用法。
- 注意：从该执行结果可以看出，in\_array是区分大小写的。“Mars”和“mars”被认为是两个不同的值。



使用 in\_array 检查数组中是否存在某值



## 4.19.2 获取数组中当前单元的索引值

- 代码(详细内容请参照本书)演示了函数key的用法。这段程序首先将当前指针指向数组的最后一个单元,然后通过函数key获取当前元素(即最后一个元素)的索引值。



使用 key 获取数组当前元素索引值



## 4.20 小结

- 本章介绍了对数组进行处理的一些主要的PHP内置函数和语言结构。他们包括：
- 使用array创建数组。
- 使用for、foreach等循环访问数组元素。
- 使用函数count()计算数组元素个数。
- 使用函数array\_chunk()/array\_merge()分割/合并数组。
- 处理数组元素和键值位置的函数current()、prev()、next()、end()等。
- 对数组元素排序的函数sort()、rsort()、asort()、ksort()等。
- 使用reset()重置数组。
- 使用list()将数组元素赋值给变量。
- 使用函数shuffle()对数组元素进行随机排序。
- 快速创建数组的函数range()。
- 使用函数array\_push()、array\_pop()压入和弹出数组元素。



## 4.20 小结

- 使用函数 `array_change_key_case()` 改变字符索引名的大小写。
- 对数组做集合处理的函数 `array_intersect()`、`array_intersect_assoc()`、`array_diff()` 和 `array_diff_assoc()`。
- 交换数组索引和元素的函数 `array_flip()`。
- 快速填充数组元素的函数 `array_fill()`。
- 使用函数 `array_count_values()` 统计数组元素出现的次数。
- 使用函数 `array_key_exists()` 检查索引是否存在。
- 使用函数 `array_keys()` 取得数组中的所有索引。
- 使用回调函数处理数组的函数 `array_map()`。
- 使用 `in_array()` 检查数组中是否存在某值。
- 使用 `key()` 获取数组中当前单元的索引值。
- 本章介绍了较多关于处理数组的 PHP 函数，读者需要再实践中慢慢掌握这些函数及其用法。

