第 5 章

PHP 数组应用

数组(Array)是一组数据的存储空间,这一组存储空间在内存中是相邻接的,每个存储空间存储了一个数组元素,元素之间使用“键”(key)来识别,通过数组名和键的组合,实现对数组中每一个元素的访问。本章详细讲解数组的基本概念,以及数组常用的处理函数,并对数组遍历的几种方法进行比较。

5.1 数组的基本概念

数组由多个元素组成,元素之间相互独立,并使用键来识别,每个元素相当于一个变量,用来保存数据。因此,可以将数组视为一串内存空间连续的变量组合。

5.1.1 为什么引入数组

使用标量数据类型定义的变量只能存储单个“数据”,仅依靠标量数据类型远不能解决现实生活中的一些常见问题,例如一个设置个人兴趣爱好的页面,如图 5-1 所示。

图 5-1 设置个人兴趣爱好的页面

从图 5-1 可以得出以下两点。

(1) 用户可选的“兴趣爱好”选项的个数有 35 项之多,编程过程中不可能为 35 个“兴趣爱好”选项,设置 35 个变量与之对应。

(2) “兴趣爱好”选项的个数以后有可能会继续增加。

为此,需引入数组数据类型以更好地解决上述问题。

5.1.2　数组的概念

数组是一组数据的集合，将数据按照一定规则组织起来，形成一个可操作的整体。数组是对大量数据进行有效组织和管理的手段之一，通过数组函数，可以对大量性质相同的数据进行存储、排序、插入、删除等操作，从而可以有效地提高程序开发效率，改善程序的编写方式。

数组的本质是存储、管理和操作一组变量。数组与变量的示意如图 5-2 所示。

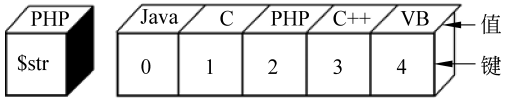

图 5-2　变量与数组

变量中保存的是单个数据，而数组中保存的则是多个变量的集合。使用数组的目的就是将多个相互关联的数据组织在一起形成一个整体，作为一个单元来使用。

数组中的每个实体都包含两项：键(key)和值(value)。其中，键(也称为数组的下标)可以是数字、字符串或者数字和字符串的组合，用于标识数组中相应的值；而值被称为数组中的元素，可以定义为任意数据类型，甚至是混合类型。最终通过键来获取相应的值。例如，一个足球队通常会有几十人，要认识他们，首先会把他们看作某队的成员，然后通过他们的号码来区分每一名队员。这时候，球队就是一个数组，而号码就是数组的下标(键)。当指明几号队员的时候，也就找到了这名队员(值)。

5.1.3　数组的类型

PHP 中根据数据的维数将数组分为一维数组、二维数组和多维数组。无论是一维还是多维，都可以按照键的数据类型，统一将数组分为两种：数字索引数组(indexed array)和关联数组(associative array)。数字索引数组使用数字作为键名(图 5-2 中展示的就是一个数字索引数组)，关联数组则使用字符串作为键名(如图 5-3 所示)。

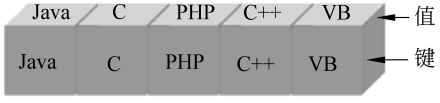

图 5-3　关联数组

1. 数字索引数组

数字索引数组的下标(键名)由数字(整数)组成，默认从 0 开始，每个数字对应数组元素在数组中的位置，不需要特别指定，PHP 会自动为数字索引数组的键名赋一个整数值，然后从这个值开始自动增量。当然，也可以指定从某个具体位置开始保存数据。

数组中的每个实体都包含两项：键名和值。可以通过键名来获取相应数组元素(值)，如果键名是数值，那么就是数字索引数组；如果键名是数值与字符串的混合，那么就是关联数组。

声明一个数字索引数组的代码如下：

```
//声明数字索引数组
$arr_int =array ("虚拟现实技术及应用","网站开发与设计","数据库原理及应用基础");
```

2. 关联数组

关联数组的下标(键名)由数值和字符串混合的形式组成。如果在一个数组中，有一个键名不是数字，那么这个数组就称为关联数组。

关联数组使用字符串键名来访问存储在数组中的值，如图 5-2 所示。

声明一个关联索引数组的代码如下：

```
//声明关联数组
$arr_string =array ("PHP"=>"PHP 程序设计","Java"=>"Java 程序设计","C#"=>"C#
程序设计");
```

说明：关联数组的键名可以是任何一个整数或字符串。如果键名是一个字符串，则要给这个键名或索引加上个定界符——单引号(')或双引号(")。对于数字索引数组，为了避免不必要的麻烦，最好也加上定界符。

5.2 声 明 数 组

PHP 中声明数组的规则如下。

(1) 数组的名称由一个符号($)开始，第一个字符是字母或下画线，其后是任意数量的字母、数字或下画线。

(2) 在同一个程序中，标量变量和数组变量都不能重名。例如，如果已经存在一个名为 $ string 的变量，而又创建一个名为 $ string 的数组，那么前一个变量就会被覆盖。

(3) 数组的名称区分大小写，如 $ String 与 $ string 是不同的。

声明数组的方法有两种，分别为用户声明和函数声明。下面介绍用户自己创建数组，以及使用函数直接创建数组。

5.2.1 用户创建数组

在定义数组时，有时不知道需要创建多大的数组，或数组的大小可能会发生变化，这时就可以使用直接为数组元素赋值的方式来创建数组，其语法格式如下：

```
$arr['key'] =value;
$arr['0'] =value;
```

其中，key 可以是整型或者字符串型数据，value 可以是任意数据类型的值。

【例 5-1】 应用标识符[]创建数组 array，然后应用 print_r()函数输出数组元素。

```
<?php
    $array['0']="数据库原理及应用基础";        //通过标识符[]定义数组元素值
```

```
    $array['1']="虚拟现实技术及应用";          //通过标识符[]定义数组元素值
    $array['2']="Office高级应用";              //通过标识符[]定义数组元素值
    $array['3']="网站开发与设计";              //通过标识符[]定义数组元素值
    print_r($array);                           //输出所创建数组的结构
?>
```

运行结果为：

```
Array ( [0] =>数据库原理及应用基础 [1] =>虚拟现实技术及应用 [2] =>Office高级应用
[3] =>网站开发与设计 )
```

说明：本例中使用print_r()函数输出数组元素，因为使用print_r输出数组时，将会按照一定格式输出数组中所有的键名和元素。使用echo语句则可以输出数组中指定的某个元素。

注意：

(1) 用户创建数组，比较适合创建未知大小的数组，或者创建大小可能会发生改变的数组。

(2) 通过标识符[]直接为数组元素赋值时，同一数组元素中的数组名必须相同。

(3) 如果数组元素中的键是一个浮点数，则该键将被强制转换为整数(例如浮点数8.0将被强制转换为整数8)；如果键是true或false，则该键将被强制转换为整数1或0。

(4) 如果数组元素中的键是一个字符串，且该字符串完全符合整数格式时，则该键将被强制转换为整数(例如"9"将被强制转换为整数9)。

(5) 由于数组元素中的键唯一标识一个元素，因此数组中元素的键不能相等(使用==比较)。如果两个数组元素的键相等，则该键对应的值将被覆盖。

(6) 不要在array()语言结构中使用诸如"red=>"red""键值对的方式来创建数组元素，也不要使用诸如$colors[red]="red"的赋值语句来创建数组元素，否则程序的可读性及运行效率将大打折扣。

(7) 在直接为数组元素赋值时，键名是可以省略的，下标默认从0开始。

5.2.2 函数创建数组

PHP中创建数组的最常用函数是array()，其语法如下：

```
array array ( [mixed…])
```

参数mixed的格式为"key =>value"，多个参数mixed用逗号分开，分别定义键和值。

应用array()函数声明数组时，数组下标(键名)既可以是数字索引也可以是关联索引。下标与数组元素值之间用"=>"进行连接，不同数组元素之间用逗号进行分隔。

应用array()函数定义数组时，可以在函数体中只给出数组元素值，而不必给出键名。

说明：

(1) 数组中的索引可以是字符串或数字。如果省略了索引，会自动产生从0开始的整数索引。如果索引是整数，则下一个产生的索引将是目前最大的整数索引加1。如果

定义了两个完全相同的索引，则后一个会覆盖前一个。

（2）数组中各数据元素的数据类型可以不同，也可以是数组类型。当 mixed 是数组类型时，就是二维数组。

【例 5-2】 应用 array()函数声明数组，并输出数组中的元素。

```
<?php
    $arr_string=array('one'=>'php','two'=>'java');
                                                    //以字符串作为数组索引，指定关键字
    print_r($arr_string);                           //通过 print_r()函数输出数组
    echo "<br>";
    echo $arr_string['one']."<br>";                 //输出数组中的索引为 Java 的元素
    $arr_int=array('php','java'); //以数字作为数组索引，从 0 开始，没有指定关键字
    var_dump($arr_int);                             //输出整个数组
    echo "<br>";
    echo $arr_int['0']."<br>";                      //输出数组中的第 1 个元素
    $arr_key=array (0 =>'数据库原理及应用基础', 1 =>'虚拟现实技术及应用', 1 =>'
                Office 高级应用');
    //指定相同的索引
    print_r($arr_key);                              //输出整个数组，发现只有两个元素
?>
```

运行结果为：

```
Array ( [one] =>php [two] =>java )
php
array(2) { [0]=>string(3) "php" [1]=>string(4) "java" }
php
Array ( [0] =>数据库原理及应用基础 [1] =>Office 高级应用 )
```

上述示例的 var_dump()函数可以输出数组中每个元素与值的数据类型。从此示例可以看出，var_dump()函数与 print_r()函数用法类似，但 var_dump()函数的功能更强大些。

5.2.3 创建二维数组

上述创建的数组都是只有一列数据内容的，因此称为一维数组。如果将两个一维数组组合成一个数组，那么就称为二维数组。

【例 5-3】 用 array()函数创建一个二维数组，并输出数组的结构。

```
<?php
    $str =array (
        "计算机类图书"=>array ("数据库原理及应用基础","大数据技术及应用",
       "Office 高级应用"),"历史图书"=>array ("1"=>"明朝那些事儿","2"=>"鱼羊野史",
```

```
        "3"=>"从晚清到民国"),"文学图书"=>array ("地平线",3=>"编年史","摇滚记")
    );                    //声明二维数组
    print_r ($str) ;      //输出数组元素
?>
```

运行结果为：

```
array ( [计算机类图书] =>array ( [0] =>数据库原理及应用基础 [1] =>大数据技术及应用
[2] =>Office 高级应用 )
[历史图书] =>array ( [1] =>明朝那些事儿 [2] =>鱼羊野史 [3] =>从晚清到民国 )
[文学图书] =>array ( [0] =>地平线 [3] =>编年史 [4] =>摇滚记 ) )
```

5.3　数组遍历与输出

5.3.1　访问数组元素

访问数组元素值的方法和访问变量值的方法相同：通过指定数组名并在方括号内指定键的方式访问数组元素的值。使用这样的方法访问数组，不仅可以读取某个数组元素的值，还可以为数组添加数组元素以及修改数组元素的值，并可以像访问变量那样访问数组元素的值。

【例 5-4】 数组元素访问应用。

```
<?php
    $colors =array("red"=>"red","green"=>"green","white"=>"white","blue"
             =>"blue");
    $colors["black"] ="black";    //为数组添加数组元素：  "black"=>"black"
    $colors["red"] ="#FF0000";    //修改键为 red 的元素值：  "red"=>"#FF0000"
    print_r($colors);
    echo "<br/>";
    if(isset($colors["green"])){
                          //使用 isset()函数判断键为 green 的数组元素是否已定义
        echo "我喜欢绿色。";
    }
    echo "<br/>";
    unset($colors["green"]);      //使用 unset()函数取消键为 green 的数组元素定义
    if(!isset($colors["green"])){
        echo "我又不喜欢绿色了。";
    }
    echo "<br/>";
    echo gettype($colors["blue"]);
                          //使用 gettype()函数查看键为 blue 的数组元素的数据类型
```

```
    echo "<br/>";
    var_dump($colors["blue"]);
                        //使用 var_dump()函数得到键为 blue 的数组元素的数据类型
?>
```

运行结果为：

```
array(["red"]=>#FF0000["green"]=>green["white"]=>white["blue"]=>blue["black"]=>
black)
我喜欢绿色。
我又不喜欢绿色了。
string
string(4)"blue"
```

上述示例中的 unset()函数可以删除整个数组，也可以删除数组的某个元素。

PHP 提供两种变量赋值方式：传值赋值和传地址赋值，对于数组同样适用。

【例 5-5】 数组传值赋值示例。

```
<?php
    $colors1 =array("red"=>"red","green"=>"green","white"=>"white");
    $colors2 =$colors1;
    $colors2["blue"] ="blue";    //为数组$colors2 添加元素："blue"=>"blue"
    $colors2["red"] ="#FF0000";
                        //修改数组$colors2 键为 red 的元素值："red"=>"#FF0000"
    print_r($colors1);
    echo "<br/>";
    print_r($colors2);          //输出
?>
```

运行结果为：

```
array ( [red] =>red [green] =>green [white] =>white )
array ( [red] =>#FF0000 [green] =>green [white] =>white [blue] =>blue )
```

【例 5-6】 数组传地址赋值示例。

```
<?php
    $colors1 =array("red"=>"red","green"=>"green","white"=>"white");
    $colors2 =&$colors1;
    $colors2["blue"] ="blue";
                    //为数组$colors1 和$colors2,添加数组元素："blue"=>"blue"
    $colors2["red"] ="#FF0000";
                    //修改数组$colors1 和$colors2 的元素值："red"=>"#FF0000"
    print_r($colors1);
```

```
    echo "<br/>";
    print_r($colors2);
?>
```

运行结果为：

```
array ( [red] =>#FF0000 [green] =>green [white] =>white [blue] =>blue )
array ( [red] =>#FF0000 [green] =>green [white] =>white [blue] =>blue )
```

5.3.2 数组遍历方式

遍历数组就是按照一定的顺序依次访问数组中的每个元素，直到访问完为止。PHP可以通过流程控制语句(foreach和for循环语句)和函数(list()和each())来遍历数组。下面分别介绍这几种遍历数组的方法。

1. 通过foreach循环语句遍历数组

在PHP中，foreach循环语句可以很方便地遍历数组，其语法格式如下：

```
foreach ($variable as [$key =>] $value){
    循环体
}
```

其中，$variable表示需要遍历的数组，as是一个关键字，键变量$key是可选的，值变量$value是必需的。每次循环时，foreach循环语句会把键赋给$key，值赋给$value。接下来演示通过foreach语句遍历数组，如例5-7所示。

【例5-7】 使用foreach循环语句遍历一维数组$str。

```
<?php
    //创建数组
    $str=array('中国农业大学'=>'www.cau.edu.cn',
               '教育部'=>'www.moe.gov.cn',
    );
    echo "原数组: ";
    print_r($str);
    echo "<br>";
    echo "遍历后的值: ";
    foreach($str as $link){          //遍历数组
        echo $link."  ";
    }
?>
```

运行结果为：

```
原数组: array ( [中国农业大学] =>www.cau.edu.cn [教育部] =>www.moe.gov.cn )
遍历后的值: www.cau.edu.cn www.moe.gov.cn
```

上例是将数组的值遍历输出。要将数组的键名和元素值都遍历输出，只需将上例中的 foreach 循环语句修改为：

```
foreach($str as $key=>$link){
    echo "$key----$link"."<br>";          //输出数组中的键名和元素值
}
```

运行结果为：

```
原数组: array ( [中国农业大学] =>www.cau.edu.cn [教育部] =>www.moe.gov.cn )
遍历后的值: 中国农业大学----www.cau.edu.cn
教育部----www.moe.gov.cn
```

此处需注意，$key 和 $value 保存的键与值是通过值传递的方式赋值的。如果使用引用传递，只需在值变量前加上 & 符号即可，但键变量不能写成引用形式。

【例 5-8】 遍历二维数组示例。

```
<?php
    //定义数组,存储订货单中商品信息
    $goods =array(
        array('name'=>'主板','price'=>'379','producing'=>'深圳','num'=>3),
        array('name'=>'显卡','price'=>'799','producing'=>'上海','num'=>2),
        array('name'=>'硬盘','price'=>'589','producing'=>'北京','num'=>5)
    );
    //商品价格总计
    $total =0;
    //拼接订货单中信息
    $str ='<h2>商品订货单</h2>';
    $str .='<table class="bordered">';
    $str .='<tr><td>商品名称</td><td>单价(元)</td><td>产地</td><td>数量
            (个)</td><td>总价(元) </td></tr>';
    //循环数组
    foreach($goods as $values){
        $str .='<tr>';
        foreach($values as $v){
            $str .='<td>'.$v.'</td>';
        }
        //计算并拼接每件商品的总价格
        $sum =$values['price']*$values['num'];
        $str .='<td>'.$sum.'</td>';
          $str .='</tr>';
        //计算订货单中所有商品总价格
        $total +=$sum;
```

```
    }
    $str .='<tr><td colspan="5">小计: <span>'.$total.'元</span></td></tr
    ></table>';
    echo $str;
?>
```

运行结果为：

```
商品订货单
商品名称        单价(元)        产地        数量(个)        总价(元)
主板            379             深圳          3             1137
显卡            799             上海          2             1598
硬盘            589             北京          5             2945
小计 5680 元
```

2. 通过 for 循环语句遍历数组

如果要遍历的数组是数字索引数组，并且数组的索引值为连续整数时，可以使用 for 循环语句来遍历，但前提条件是需要应用 count()函数获取数组中元素的数量，然后将获取的元素数量作为 for 循环语句执行的条件，才能完成数组的遍历。

【例 5-9】 使用 for 循环语句来遍历数组。

```
<?php
    $array=array(                          //定义数组
        "0"=>"数据库原理及应用基础",
        "1"=>"虚拟现实技术及应用",
        "2"=>"Office 高级应用",
        "3"=>"网站开发与设计"
    );
    for($i=0;$i<count($array);$i++){       //使用 for 循环遍历数组
        echo $array[$i]."<br>";            //输出数组元素
    }
?>
```

运行结果为：

```
数据库原理及应用基础
虚拟现实技术及应用
Office 高级应用
网站开发与设计
```

3. 通过数组函数 list()和 each()遍历数组

list()与 each()函数结合起来也可以遍历数组。接下来分别介绍一下这两个函数。

list()函数将数组中的值赋给一些变量，该函数仅能用于数字索引的数组，且数字索引从 0 开始。其语法格式如下：