**第3章 数组**

**1. 什么是数组？**

**Bob的产品，可以用一个数组保存起来，数组变量名为$products , 三个变量值分别为：Tires 、 Oil 、Spark Plugs**

**2. 数字索引数组**

**数字索引默认值从0开始，也可以改变它。**

**A. 数字索引数组的初始化**

**$products = array('Tires' , 'Oil ' , 'Spark Plugs');**

**print\_r($products);**



**按升序排列保存一个数组，用range()函数自动创建。**

**$numbers = range(1,10);**

**$odds = range(1,10,2);**

**$letters = range('a','z');**



**B. 访问数组的内容**

**$products[0] , $products[1], $products[2]**

**$products[0] = 'Fuses'; 赋值修改元素值**

**$products[3] = 'Fuses'; 赋值添加元素**

**显示**

**echo "$products[0] , $products[1], $products[2], $products[3] ";**

**$products[0] = 'Tires'; 创建一个元素的数组**

**$products[1] = 'Oil'; 为这个数组添加新值**

**$products[2] = 'Spark Plugs'; 为这个数组添加新值**

**数组大小，根据所增加的元素多少动态变化**

**等价于：**

**$products = array('Tires' , 'Oil' , 'Spark Plugs');**

**C. 使用循环访问数组**

**数字索引，用for循环**

**for($i = 0; $i<3; $i++){**

**echo $products[$i] . " ";**

**}**

**也可以使用foreach循环**

**foreach($products as $current){**

**echo $current;**

**}**

**3. 使用不同索引的数组**

**A. 初始化关联数组**

**$prices = array('Tires' =>100, 'Oil' =>10 , 'Spark Plugs'=>4);**

**B. 访问数组元素**

**$prices['Tires'] $prices['Oil'] $prices['Spark Plugs']**

**也可以这样创建关联数组**

**$prices = array('Tires' =>100);**

**$prices['Oil'] = 10;**

**$prices['Spark Plugs'] = 4;**

**还可以这样创建关联数组**

**$prices['Tires'] =100;**

**$prices['Oil'] = 10;**

**$prices['Spark Plugs'] = 4;**



**C. 使用循环**

**关联数组的索引不是数字，无法用for循环，但是可以用foreach循环或list()和each()结构。**

**foreach($prices as $key => $value){**

**echo $key . " - " . $value . "<br/>" ;**

**}**



**while($element = each($prices) ){**

**echo $element['key'];**

**echo " - ";**

**echo $element['value'];**

**echo '<br/>';**

**}**



**each()函数，按顺序返回数组中的每一个元素，到达数组末尾，循环终止。**

**$element是一个数组。**

**reset($prices);**

**while( list($product,$price) = each($prices) ){**

**echo "$product - $price<br/>" ;**

**}**

 **list()允许为新变量命名。**

**4. 数组操作符**

**+ == === != <> !==**

**$a+$b 将$b中的元素添加到$a的末尾，如果二者中有一些元素索引系统，将不会被添加。**

**5. 多维数组**

**$products = array(array('TIR' , 'Tires' , 100),**

**array('OIL' , 'Oil' , 10),**

**array('SPK' , 'Spark Plugs' , 4));**

**显示这个数组**

**echo '|' . $products[0][0] . '|' . $products[0][1] . '|' . $products[0][2] . '<br/>';**

**echo '|' . $products[1][0] . '|' . $products[1][1] . '|' . $products[1][2] . '<br/>';**

**echo '|' . $products[2][0] . '|' . $products[2][1] . '|' . $products[2][2] . '<br/>';**



**或**

**for($row = 0 ; $row < 3; $row++){**

**for($col =0; $col <3; $col++){**

**echo '|' . $products[$row][$col] ;**

**}**

**echo ' | <br/>';**

**}**



**创建列名称来代替数字**

**$products = array(array('Code' =>'TIR' ,**

**'Description' => 'Tires' ,**

**'Price' => 100),**

**array('Code' =>'OIL' ,**

**'Description' => 'Oil' ,**

**'Price' => 10),**

**array('Code' =>''SPK' ,**

**'Description' =>'Spark Plugs' ,**

**'Price' => 4)**

**);**



**显示这个二维数组：**

**for($row = 0 ; $row < 3; $row++){**

**echo '|' . $products[$row]['Code'] . '|' . $products[$row]['Description' ] . '|' . $products[$row]['Price' ] . '<br/>' ;**

**}**



**for($row = 0 ; $row < 3; $row++){**

**while(list($key,$value) = each($products[$row])) {**

**echo " | $value";**

**}**

**echo " | <br/>";**

**}**



**6. 排序**

**A. 使用sort()函数**

**$products = array('Tires' , 'Oil' , 'Spark Plugs');**

**sort($products);**

**数组元素的顺序将变为：Oil, Spark Plugs, Tires.**



**$priecs = array(100,10,4);**

**sort($prices);**

**数组元素的顺序将变为：4,10,100.**

**B. 使用 asort() ksort() 函数对关联数组排序**

**$prices = array('Tires' =>100, 'Oil' =>10 , 'Spark Plugs'=>4);**

**asort($prices);**

**asort()根据数组的每个元素值升序排序。**



**$prices = array('Tires' =>100, 'Oil' =>10 , 'Spark Plugs'=>4);**

**ksort($prices);**

**ksort()根据数组的每个元素的关键字升序排序。**



**C. 对应的反向排序函数是**

**rsort() arsort() krsort()**

**7. 多维数组的排序**

**A. 用户定义排序**

**$products = array(array('TIR' , 'Tires' , 100),**

**array('OIL' , 'Oil' , 10),**

**array('SPK' , 'Spark Plugs' , 4));**

**需要定义排序规则。需要自己编写比较函数，用usort() 函数告诉PHP如何比较各个元素。**

**如按第二列字母排序**

**function compare($x,$y){**

**if($x[1] == $y[1]) {**

**return 0;**

**}else if($x[1] < $y[1]) {**

**return -1;**

**}else{**

**return 1;**

**}**

**}**

**usort($products, 'compare');**



**如按第三列排序**

**function compare($x,$y){**

**if($x[2] == $y[2]) {**

**return 0;**

**}else if($x[2] < $y[2]) {**

**return -1;**

**}else{**

**return 1;**

**}**

**}**

**usort($products, 'compare');**



**u，代表 user， usort() uasort() uksort() 都要求传入用户定义的比较函数。**

**B. 对应的反向排序函数是**

**rsort() arsort() krsort()**

**function reverse\_compare($x,$y){**

**if($x[2] == $y[2]) {**

**return 0;**

**}else if($x[2] < $y[2]) {**

**return 1;**

**}else{**

**return -1;**

**}**

**}**

**usort($products, 'reverse\_compare');**



**8. 重新排序**

**A. shuffle()将数组各元素进行随机排序。**

**bobs\_front\_page.php——用PHP为Bob的汽车配件商店制作一个动态的首页**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**<?php**

**$pictures = array('tire.jpg','oil.jpg','spark\_plug.jpg','door.jpg','steer\_wheel.jpg');**

**shuffle($pictures);**

**?>**

**<html>**

**<head>**

**<title>Bob's Auto Parts - Order Results</title>**

**</head>**

**<body>**

**<h1>Bob's Auto Parts</h1>**

**<div align="center">**

**<table width=100% border=1>**

**<tr>**

**<?php**

**for ($i=0; $i <3 ; $i++) {**

**echo "<td align = \"center\"><img src=\" ";**

**echo $pictures[$i];**

**echo "\" /><td>";**

**}**

**?>**

**</tr>**

**</table>**

**</div>**

**</body>**

**</html>**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**B. array\_reverse()给出一个原来数组的反向排序。**

**用range()创建升序序列，使用rsort() , array\_reverse()将数组变为降序。**

**$numbers = range(1,10);**

**$numbers = array\_reverse($numbers);**

**也可以**

**$numbers = range(10,1,-1);**

**或者用for循环，一次一个元素的方式创建数组：**

**$numbers = array();**

**for($i=10; $i>0; $i--){**

**array\_push($numbers, $i);**

**}**

**9. 从文件载入数组**

**vieworders.php——使用PHP显示Bob的订单orders.txt内容**



**—————————————————————————————————<?php**

**//创建一个简短变量**

**$DOCUMENT\_ROOT = $\_SERVER['DOCUMENT\_ROOT'];**

**$orders = file("$DOCUMENT\_ROOT/../orders/orders.txt");**

**$number\_of\_orders = count($orders);**

**if($number\_of\_orders == 0) {**

**echo "<p><strong>no orders pending,Please try again later. </strong></p>" ;**

**}**

**for ($i=0; $i <$number\_of\_orders ; $i++) {**

**echo $orders[$i] . "<br/>" ;**

**}**

**?>**

**—————————————————————————————————**

**该脚本使用了file()函数将整个文件载入一个数组$orders 中。文件中的每行则成为数组中的一个元素。count()计算数组中的元素个数，从而决定循环的次数。**

**还可以将订单行中的每个区段载入到单独的数组元素中，从而可以分开处理每个区段，或将它们更好地格式化。**



**—————————————————————————————————**

**<html>**

**<head>**

**<title>Bob's Auto Parts - Order Results</title>**

**</head>**

**<body>**

**<h1>Bob's Auto Parts</h1>**

**<h2>Order Results</h2>**

**<?php**

**//创建一个简短变量**

**$DOCUMENT\_ROOT = $\_SERVER['DOCUMENT\_ROOT'];**

**$orders = file("$DOCUMENT\_ROOT/../orders/orders.txt");**

**$number\_of\_orders = count($orders);**

**if($number\_of\_orders == 0) {**

**echo "<p><strong>no orders pending,Please try again later. </strong></p>" ;**

**}**

**?>**

<table border=1>

<tr>

<th>Order Date</th>

<th>Tires</th>

<th>Oil</th>

<th>Spark Plugs</th>

<th>Total</th>

</tr>

**<?php**

**for ($i=0; $i <$number\_of\_orders ; $i++) {**

**$line = explode("\t", $orders[$i] ) ;**

**$line[1] = intval($line[1]) ;**

**$line[2] = intval($line[2]) ;**

**$line[3] = intval($line[3]) ;**

**echo '<tr>**

**<td>' . $line[0]. '</td>**

**<td>'. $line[1]. '</td>**

**<td>'. $line[2]. '</td>**

**<td>'. $line[3]. '</td>**

**<td>'. $line[4]. '</td>**

**</tr>';**

**}**

**echo "</table>" ;**

**?>**

**</body>**

**</html>**

**—————————————————————————————————**

**10. 其他数组操作函数**

**A. 在数组中浏览：each() current() reset() end() next() pos() prev()**

**$arr = array(1,2,3);**

**$value = end($arr);**

**while($value){**

**echo "$value<br/>" ;**

**$value = prev($arr);**

**}**



**B. 对数组中的每一个元素应用任何函数 array\_walk()**

**C. 统计数组元素个数：count() sizeof() array\_count\_values()**

**$arr = array(4,5,1,2,3,1,2,1);**

**$ac = array\_count\_values($arr);**

**print\_r($ac);**



**D. 将数组转换成标量变量 extract()**