

数组的基本遍历和指针操作

1. 数组的遍历

(1) foreach基本语法

```
array_foreach.php
<?php
echo "<pre>";

$arr = array("a" => "1", "b" => "2", "c" => "3");

foreach ($arr as $key => $value) { //提供遍历过程的key和value
    echo "(key:$key, value:$value)". "\t";
}
```

www.gudashi.com/day7/array_ X
www.gudashi.com/day7/array_foreach.php
(key:a, value:1) (key:b, value:2) (key:c, value:3)

(2) 使用for和next进行数组遍历

对于php数组，往往不能使用for循环进行遍历，for循环只能遍历下标为连续整数的数组；当php数组为键值对数组（ios中叫字典），此时遍历仅用for是完成不了的，还得借助next的数组指针操作，具体示例如下：

```
$arr1 = array(1 => "dfa", "b" => "r2r", "3" => "mvd");
for ($i=0; $i < count($arr1); $i++) {
    # code...
    $key = key($arr1);
    $value = current($arr1);
    echo "key:$key value:$value\t";
    next($arr1);
}
```

key:1 value:dfa key:b value:r2r key:3 value:mvd

(3) while_each_list遍历

1) 关于each(\$arr)函数

`<body>`
`<?php`
`//使用该数组来演示each()的含义和使用方法:`
`$arr4 = array(`
`1=>3,`
`'a1'=>11,`
`3=>5,`
`);`
`$result1 = each($arr4); //此时取得"1=>3"这项, 并结果为数组`
`echo "
result1为: <pre>"; print_r($result1);`
`$result2 = each($arr4); //此时取得"'a1'=>11"这项, 并结果为数组`
`echo "
result2为: <pre>"; print_r($result2);`
`?>`
`</body>`
`</html>`

可见，each(\$arr4)取值的时候，只取得数组\$arr4的某一项的数据（含键和值）但存储的时候，做了“双份存储”

这种方式获得的数据，既可以使用数字下标进行取用，也可以使用字符串下标进行使用，只是提供了更多方便而已。

result1为:

```
Array
(
    [1] => 3
    [value] => 3
    [0] => 1
    [key] => 1
)
```

result2为:

```
Array
(
    [1] => 11
    [value] => 11
    [0] => a1
    [key] => a1
)
```

2) 关于list()函数的使用

```
$arr3 = array(0 => 3, 5 => 5, 1 => 8);
list($v1, $v2) = $arr3;
echo "$v1, $v2";
```

3, 8

3) each、list结合while实现数组遍历

```
echo "<pre>";
$arr4 = array("a" => "aa", "1" => "bb", 0 => "cc");
while (list($key, $value) = each($arr4)) {
    echo "key:$key value:$value \t\t";
}
```

key:a value:aa

key:1 value:bb

2. 数组的指针操作及遍历原理

(1) 指针操作

```
var_dump(current($arr)); // 获取当前指向的数据单元
var_dump(key($arr));
echo "<br>";

next($arr); // 指向下一个数据单元
var_dump(current($arr));
var_dump(key($arr));
echo "<br>";

reset($arr); // 指向第一个数据单元
var_dump(current($arr));
var_dump(key($arr));
echo "<br>";

prev($arr); // 指向前一个数据单元
var_dump(current($arr));
var_dump(key($arr));
echo "<br>";

end($arr); // 指向最后一个数据单元
var_dump(current($arr));
var_dump(key($arr));
echo "<br>";
```

(key:a, value:1)	(key:b, value:2)	(key:c, value:3)
string(1) "1"		
string(1) "a"		
	string(1) "2"	
	string(1) "b"	
string(1) "1"		
string(1) "a"		
	bool(false)	
	NULL	
		string(1) "3"
		string(1) "c"

指针操作都有返回值，指针移动后会读取

(2) 遍历原理

