

# 后盾网 人人做后盾

[www.houdunwang.com](http://www.houdunwang.com)

## 数组

后盾网 2011-2016 v5.0

---

- 数组是在程序设计中，为了处理方便，把若干变量按有序的形式组织起来的一种形式。这些按序排列的同类数据元素的集合称为数组。
- 一个数组可以分解为多个数组元素，这些数组元素可以是基本数据类型或是构造类型。因此按数组元素的类型不同，数组又可分为数值数组、字符数组等各种类别。

# 基础知识

---

- 所谓的数组下标可以视为资料内容在此数组中的识别名称，通常被称为数组下标。
- 当索引值为数值时，也代表此资料内容在数组中的储存位置。
- 数组中有几个索引值就被称为几维数组。

数组分类：

- 在PHP中有两种数组：索引数组和关联数组。
- 索引（indexed）数组的索引值是整数，以0开始。当通过位置来标识东西时用索引数组。
- 关联（associative）数组以字符串做为索引值，关联数组更像操作表。索引值为列名，用于访问列的数据。

# 基础知识

---



- 由于 PHP 是属于弱类型数据，因此源代码中的数组并不需要经过特别的声明操作，直接将一组数值指定给某一数组元素即可。

直接赋值方式 如：

- `$a[0]='web.houdunwang.com';`
- `$a[1]='bbs.houdunwang.com';`

使用array函数 如：

- `$a=array("web.houdunwang.com ","bbs.houdunwang.com");`

## 数组常用的赋值方式

---

```
<?php
```

```
$a=array(1,2,3,4,5,6);
```

```
$b=array("one", "two", "three");
```

```
$c=array(0=>"aaa",1=>"bbb",2=>"ccc");
```

```
$d=array("aaa",6=>"bbb","ccc");
```

```
$e=array("name"=>"zhang", "age"=>20);
```

```
?>
```

## 数组常用的赋值方式

---

声明多维数组：

```
$arr = array(  
    array(  
        "title"=>"后盾网HDPHP框架发布",  
        "click"=>200  
    )  
);
```

## 多维数组

---



- foreach 仅能用于数组，当试图将其用于其它数据类型或者一个未初始化的变量时会产生错误。有两种语法，第二种比较次要但却是第一种有用的扩展。

方法1:

- `foreach (array_expression as $value)`

方法2:

- `foreach (array_expression as $key => $value)`

## 结构控制

---

## **\$\_GET**

\$\_GET 变量是一个数组，内容是由 HTTP GET 方法发送的变量名称和值

## **\$\_POST**

\$\_POST 变量用于收集来自method="post" 的表单中的值

## **\$\_REQUEST**

通过post和get方法提交的所有数据都可以通过\$\_REQUEST数组获得

## **\$\_SERVER**

保存一些服务器信息的资料，如客户端IP，主机IP地址

## **\$GLOBALS**

引用全局作用域中可用的全部变量

# 系统超全局数组

---



## key

- 关联数组中取得键名
- mixed key ( array &\$array )
- key() 返回数组中当前单元的键名

## current

- 返回数组中的当前单元
- mixed current ( array &\$array )
- 每个数组中都有一个内部的指针指向它“当前的”单元，初始指向插入到数组中的第一个单元

# 数组操作函数

---

next

- 将数组中的内部指针向前移动一位
- mixed next ( array &\$array )

prev

- 将数组的内部指针倒回一位 mixed prev
- ( array &\$array )

## 数组操作函数

---

## count

- 计算数组中的单元数目或对象中的属性个数
- `int count ( mixed $var [, int $mode ] )`

## in\_array

- 检查数组中是否存在某个值
- 语法: `in_array ( mixed $needle , array $haystack [, bool $strict ] )`
- 如果第三个参数 `strict` 的值为 `TRUE` 则 `in_array()` 函数还会检查 `needle` 的类型是否和 `haystack` 中的相同

# 数组操作函数

---



array\_key\_exists

- 检查给定的键名或索引是否存在于数组中
- array\_key\_exists ( mixed \$key , array \$search )

## 数组操作函数

---

## array\_filter

- 用回调函数过滤数组中的单元
- `array array_filter ( array $input [, callback $callback ] )`
- `array_filter()` 依次将 `input` 数组中的每个值传递到 `callback` 函数。如果 `callback` 函数返回 `TRUE`，则 `input` 数组的当前值会被包含在返回的结果数组中。数组的键名保留不变

# 数组操作函数

---

## array\_map

- 将回调函数作用到给定数组的单元上
- `array array_map ( callback $callback , array $arr1 [, array $... ] )`
- `array_map()` 返回一个数组，该数组包含了 `arr1` 中的所有单元经过 `callback` 作用过之后的单元。`callback` 接受的参数数目应该和传递给 `array_map()` 函数的数组数目一致。

# 数组操作函数

---



## array\_push

- 将一个或多个单元压入数组的末尾
- `int array_push ( array &$array , mixed $var [, mixed $... ] )`
- `array_push()` 将传入的变量压入 `array` 的末尾。`array` 的长度将根据入栈变量的数目增加。

# 数组操作函数

---

## array\_pop

- 将数组最后一个单元弹出
- mixed array\_pop ( array &\$array )
- array\_pop() 弹出并返回 array 数组的最后一个单元，并将数组 array 的长度减一。

# 数组操作函数

---

## array\_shift

- 将数组开头的单元移出数组
- mixed array\_shift ( array &\$array )
- array\_shift() 将 array 的第一个单元移出并作为结果返回。

# 数组操作函数

---



## array\_unshift

- 在数组开头插入一个或多个单元
- `int array_unshift ( array &$array , mixed $var [, mixed $... ] )`
- `array_unshift()` 将传入的单元插入到 `array` 数组的开头,
- 返回 `array` 数组新的单元数目。

# 数组操作函数

---

array\_keys

- 返回数组中所有的键名
- array\_keys() 返回 input 数组中的数字或者字符串的键名。

## 数组操作函数

---

## array\_values

- 返回数组中所有的值
- array array\_values ( array \$input )
- array\_values() 返回 input 数组中所有的值并给其建立数字索引。

# 数组操作函数

---



## array\_merge

- 合并一个或多个数组
- `array array_merge ( array $array1 [, array $array2] )`
- `array_merge()` 将一个或多个数组的单元合并起来。
- 如果输入的数组中有相同的字符串键名，则该键名后面的值将覆盖前一个值。然而，如果数组包含数字键名，后面的值将不会覆盖原来的值，而是附加到后面。

# 数组操作函数

---

## array\_change\_key\_case

- 返回字符串键名全为小写或大写的数组
- `array array_change_key_case ( array $input [, int $case ] )`
- `array_change_key_case()` 将 input 数组中的所有键名改为全小写或大写。改变是根据后一个选项 case 参数来进行的。可以在这里用两个常量，`CASE_UPPER` 和 `CASE_LOWER`。默认值是 `CASE_LOWER`。

# 数组操作函数

---

## var\_export

- 输出或返回一个变量的字符串表示
- `mixed var_export ( mixed $expression [, bool $return ] )`
- 此函数返回关于传递给该函数的变量的结构信息，它和 `var_dump()` 类似，不同的是其返回的表示是合法的 PHP 代码。
- 您可以通过将函数的第二个参数设置为 `TRUE`，从而返回变量的表示。

# 数组操作函数

---



1. 创建递归函数完成多维数组value的大小写转换
2. 创建出操作网站配置文件的C()函数

# 作业

---