

## 1.13

### 预定义变量

所谓预定义变量，其实指，php 这个语言工具中，预先就定义好的变量；我们只是“拿来使用”。

### 综述

主要有：\$\_GET, \$\_POST, \$\_REQUEST, \$\_SERVER, \$GLOBALS, 均是数组

系统定义与维护——即我们不应该去给其赋值或销毁其值，只应该去“用其值”。

具有超全局作用域——哪里都可以使用。

不同情形下可能具有不同的值

### \$\_POST 变量

含义：

它代表用户通过表单以 post 方式（method="post"）提交的时候所提交的所有数据——这个称为 post 数据。

基本演示：

7form\_post.html 页面：

```
13 <!--  
14 以下表单的数据，点击提交后，会将所有表单数据提交给  
15 8post_data.php这个页面（文件），并在该文件中去处理  
16 其实就是指：程序进入该文件中运行。  
17 -->  
18 <form action="8post_data.php" method="post">  
19 » 数据1: <input type="text" name="data1" />  
20 » <br />  
21 » 数据2: <input type="text" name="data2" />  
22 » <br />  
23 » <input type="submit" value="提交" />  
24 </form>
```

8post\_data.php 页面及输出结果为::

```

13 <?php
14 //先一个一个获取数据:
15 $d1 = $_POST['data1'];
16 $d2 = $_POST['data2'];
17 //然后输出, 看看:
18 echo "d1 = $d1, d2 = $d2";
19
20 echo "<hr />";
21 //再来看看整个$_POST这个数组数据!
22 print_r($_POST);
23 ?>

```

网页标题 PHP核心知识档案

← → ↻ www.php39.com/day2/8p

d1 = 12, d2 = 34

---

Array ( [data1] => 12 [data2] => 34 )

**isset(变量):** 判断变量是否存在, 或变量是否为空 (null); 如果存在, 就是 true, 否则就是 false

**empty(变量):** 判断变量的“内容”是否为空的 (不是 null 的空, 而是没有内容), 基本上, 是一些硬性规定, 如下数据都是“空的”: 0, “”, “0”, false, null, array() 空数组也是空

如果一个变量内容是空的, empty() 返回的结果是 true, 否则是 false

小案例: php 页面计算器:

## \$\_GET 变量

含义:

它代表用户通过 get 方式 (有 5 种 get 形式) 提交的时候所提交的所有数据——这个称为 get 数据。

小提示: get, post 没有翻译!

提交 get 数据有 5 种形式:

形式 1:

```

<form action="目标文件.php" method="get">
    <input type="text" name="data1" />
    <input type="text" name="data2" />
    <input type="submit" value="提交" />
</form>

```

这种形式的 get 数据, 跟 post 数据类似, 数据内容由用户填写或选择而得到!

形式 2:

```

<a href="目标文件.php?data1=5&data2=cctv&age=18">链接文字</a>

```

说明:

- 1, 它只是一个链接而已, 只是在链接文件名的后面加上 “?”, 然后一个一个 “串接数据”;
- 2, 数据形式为: 数据项名称=数据值, 相互之间用 “&” 符号隔开
- 3, 这种形式的数据也同样是 “点击链接” 就提交的 get 数据, 但用户只能选择点还是不点, 而不能修改数据。

形式 3:

```
<script>
    location.href = “目标文件.php? data1=5&data2=cctv&age=18” ;
</script>
```

说明:

- 1, 该语句可以看做是通过 js 技术实现的页面跳转功能, 跟 a 标签的连接功能完全一样!
- 2, 其中该语句, 通常都是放在一个函数中, 然后因为某个事件发生而去调用该函数!

形式 4:

```
<script>
    location.assign( “目标文件.php? data1=5&data2=cctv&age=18” );
</script>
```

说明:

该语句其实跟形式 3 完全一样功能, 只是 location 这个对象实现页面跳转的另一个语法形式而已!

形式 5: php 的跳转语法

```
<?php
    //语法形式: header( “location: 目标网页地址” );

    header( “location: 目标文件.php? data1=5&data2=cctv&age=18” );
?>
```

小结:

上述多种 get 形式提交数据, 都要理解为:

跳转到某个页面, 并 “同时” 携带 (提交) 一定的 get 数据过去!

不管哪种形式的 get 数据提交, 接收 get 数据, 都只有一种形式, 跟 post 类似:

```
$v1 = $_GET[ ‘数据项名称’ ];    //取得一个 get 数据项的值; 比如
$_GET[ ‘data1’ ],  $_GET[ “age” ];
```

也可以 “输出” 所有 get 数据:

```
print_r($_GET); 或  var_dump($_GET);
```

## **\$\_REQUEST 变量**

含义：

一句话，它是\$\_GET 变量和\$\_POST 变量数据的“合集”：即，它里面同时存储了这两种数据

怎么能同时具有 get 和 post 数据呢？

只有一个方式：

```
<!--以下 action 中的？号后面的数据都会以 get 方式提交-->
<form    action= " 目标文件.php ? data1=5&data2=cctv&age=18 "
method=" post" >
    <!--以下表单项的数据都会以 post 方式提交-->
    <input type=" text"    name=" n1"    />
    <input type=" text"    name=" n1"    />
    <input type=" submit"  value=" 提交"  />
</form>
```

举例：

提交后结果为：

request 数据取值时，跟 get 数据和 post 也完全一样！

```
$_REQUEST[ '数据项名称' ];
```

当 post 数据和 get 数据的数据项名称相同时（其实我们反对这么做），默认是 post 数据覆盖了 get 数据。

不过这个状况同样可以在 php.ini 中设置：

## **\$\_SERVER 变量**

含义：

它代表在一次浏览网页的过程中的浏览器端的一些信息或服务器端的一些信息。

我们只是在程序中可以拿到这些信息，并用于编程所需！比如：取得用户的访问 IP 地址。

注意：

这种信息，随着不同的页面，和不同的服务器，以及不同的时刻，都可能不同！

要求：

大约有 30 个左右的信息，我们只要知道其中 5 个左右！主要有：

\$\_SERVER[ 'REMOTE\_ADDR' ]：获取访问者的 ip 地址

`$_SERVER[ 'SERVER_ADDR' ]`: 获取服务器所在的 ip 地址  
`$_SERVER[ 'SERVER_NAME' ]`: 获取服务器的名字, 其实就是站点设置中的 `servername`  
`$_SERVER[ 'DOCUMENT_ROOT' ]`: 获取站点的真实物理地址, 其实就是站点设置中的 `documentroot`  
`$_SERVER[ 'PHP_SELF' ]`: 获取当前网页地址 (不含域名部分)  
`$_SERVER[ 'SCRIPT_FILENAME' ]`: 获取当前网页地址物理路径  
`$_SERVER[ 'QUERY_STRING' ]` 获取当前网页地址中的所有 `get` 数据 (就是?号后面部分), 但只是一个整体的字符串而已。

```
14 echo "<pre>";
15 print_r($_SERVER);
16 echo "</pre>";
17
18 echo "<table border='1'>";
19 foreach($_SERVER as $key => $value){
20     echo "<tr>";
21     echo "<td>$key</td>";
22     echo "<td>$value</td>";
23     echo "</tr>";
24 }
25 echo "</table>";
```

## **\$GLOBALS 变量**

含义:

它也是一个“重复性数据”, 它里面存储了我们自己定义的所有“全局变量”。

举例:

`$v1 = 1;` //定义了一个全局变量,

此时, 就有了这样一个数据: `$GLOBALS[ 'v1' ]`, 其值就是 1

`echo $v1 ;` //输出 1

`echo $GLOBALS[ 'v1' ];` //输出 1

这个变量, 主要是用于在局部范围不可以使用全局变量的时候, 又需要该全局变量的值, 此时就可以用它来取得该全局变量的值。