

PHP 超级全局变量

超级全局变量在PHP 4.1.0之后被启用, 是PHP系统中自带的变量, 在一个脚本的全部作用域中都可用。

PHP 超级全局变量

PHP中预定义了几个超级全局变量 (superglobals) , 这意味着它们在一个脚本的全部作用域中都可用。 你不需要特别说明, 就可以在函数及类中使用。

PHP 超级全局变量列表:

- \$GLOBALS
- \$_SERVER
- \$_REQUEST
- \$_POST
- \$_GET
- \$_FILES
- \$_ENV
- \$_COOKIE
- \$_SESSION

本章节我们将讲解几个常用的超级全局变量,其余变量我们在接下来几个章节会介绍到。

PHP \$GLOBALS

\$GLOBALS 是PHP的一个超级全局变量组, 在一个PHP脚本的全部作用域中都可以访问。

\$GLOBALS 是一个包含了全部变量的全局组合数组。变量的名字就是数组的键。

以下实例介绍了如何使用超级全局变量 \$GLOBALS:

实例

```
<?php
$x = 75;
$y = 25;
function addition()
{
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
?>
```

运行实例 »

以上实例中 z 是一个\$GLOBALS数组中的超级全局变量，该变量同样可以在函数外访问。

PHP \$_SERVER

\$_SERVER 是一个包含了诸如头信息(header)、路径(path)、以及脚本位置(script locations)等等信息的数组。这个数组中的项目由 Web 服务器创建。不能保证每个服务器都提供全部项目；服务器可能会忽略一些，或者提供一些没有在这里列举出来的项目。

以下实例中展示了如何使用\$_SERVER中的元素:

实例

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

运行实例 »

下表列出了所有 \$_SERVER 变量中的重要元素:

元素/代码	描述
\$_SERVER['PHP_SELF']	当前执行脚本的文件名，与 document root 有关。例如，在地址为 http://example.com/test.php/foo.bar 的脚本中使用 \$_SERVER['PHP_SELF'] 将得到 /test.php/foo.bar。__FILE__ 常量包含当前(例如包含)文件的完整路径和文件名。从 PHP 4.3.0 版本开始，如果 PHP 以命令行模式运行，这个变量将包含脚本名。之前的版本该变量不可用。
\$_SERVER['GATEWAY_INTERFACE']	服务器使用的 CGI 规范的版本；例如，"CGI/1.1"。
\$_SERVER['SERVER_ADDR']	当前运行脚本所在的服务器的 IP 地址。
\$_SERVER['SERVER_NAME']	当前运行脚本所在的服务器的主机名。如果脚本运行于虚拟主机中，该名称是由那个虚拟主机所设置的值决定。(如: www.runoob.com)
\$_SERVER['SERVER_SOFTWARE']	服务器标识字符串，在响应请求时的头信息中给出。(如：Apache/2.2.24)

<code>\$_SERVER['SERVER_PROTOCOL']</code>	请求页面时通信协议的名称和版本。例如, "HTTP/1.0"。
<code>\$_SERVER['REQUEST_METHOD']</code>	访问页面使用的请求方法; 例如, "GET", "HEAD", "POST", "PUT"。
<code>\$_SERVER['REQUEST_TIME']</code>	请求开始时的时间戳。从 PHP 5.1.0 起可用。(如: 1377687496)
<code>\$_SERVER['QUERY_STRING']</code>	query string (查询字符串), 如果有的话, 通过它进行页面访问。
<code>\$_SERVER['HTTP_ACCEPT']</code>	当前请求头中 Accept: 项的内容, 如果存在的话。
<code>\$_SERVER['HTTP_ACCEPT_CHARSET']</code>	当前请求头中 Accept-Charset: 项的内容, 如果存在的话。例如: "iso-8859-1,* ,utf-8"。
<code>\$_SERVER['HTTP_HOST']</code>	当前请求头中 Host: 项的内容, 如果存在的话。
<code>\$_SERVER['HTTP_REFERER']</code>	引导用户代理到当前页的前一页的地址 (如果存在)。由 user agent 设置决定。并不是所有的用户代理都会设置该项, 有的还提供了修改 HTTP_REFERER 的功能。简言之, 该值并不可信。)
<code>\$_SERVER['HTTPS']</code>	如果脚本是通过 HTTPS 协议被访问, 则被设为一个非空的值。
<code>\$_SERVER['REMOTE_ADDR']</code>	浏览当前页面的用户的 IP 地址。
<code>\$_SERVER['REMOTE_HOST']</code>	浏览当前页面的用户的主机名。DNS 反向解析不依赖于用户的 REMOTE_ADDR。
<code>\$_SERVER['REMOTE_PORT']</code>	用户机器上连接到 Web 服务器所使用的端口号。
<code>\$_SERVER['SCRIPT_FILENAME']</code>	当前执行脚本的绝对路径。
<code>\$_SERVER['SERVER_ADMIN']</code>	该值指明了 Apache 服务器配置文件中的 SERVER_ADMIN 参数。如果脚本运行在一个虚拟主机上, 则该值是那个虚拟主机的值。(如: someone@runoob.com)
<code>\$_SERVER['SERVER_PORT']</code>	Web 服务器使用的端口。默认值为 "80"。如果使用 SSL 安全连接, 则这个值为用户设置的 HTTP 端口。
<code>\$_SERVER['SERVER_SIGNATURE']</code>	包含了服务器版本和虚拟主机名的字符串。
<code>\$_SERVER['PATH_TRANSLATED']</code>	当前脚本所在文件系统 (非文档根目录) 的基本路径。这是在服务器进行虚拟到真实路径的映像后的结果。
<code>\$_SERVER['SCRIPT_NAME']</code>	包含当前脚本的路径。这在页面需要指向自己时非常有用。__FILE__ 常量包含当前脚本(例如包含文件)的完整路径和文件名。

`$_SERVER['SCRIPT_URI']`URI 用来指定要访问的页面。例如 `"/index.html"`。

PHP \$_REQUEST

PHP `$_REQUEST` 用于收集HTML表单提交的数据。

以下实例显示了一个输入字段 (`input`) 及提交按钮(submit)的表单(form)。当用户通过点击 "Submit" 按钮提交表单数据时, 表单数据将发送至`<form>`标签中 `action` 属性中指定的脚本文件。在这个实例中, 我们指定文件来处理表单数据。如果你希望其他的PHP文件来处理该数据, 你可以修改该指定的脚本文件名。然后, 我们可以使用超级全局变量 `$_REQUEST` 来收集表单中的 `input` 字段数据:

实例

```
<html>
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Name: <input type="text" name="fname">
<input type="submit">
</form>
<?php
$name = $_REQUEST['fname'];
echo $name;
?>
</body>
</html>
```

[运行实例 »](#)

PHP \$_POST

PHP `$_POST` 被广泛应用于收集表单数据, 在HTML form标签的指定该属性: `"method="post"`。

以下实例显示了一个输入字段 (`input`) 及提交按钮(submit)的表单(form)。当用户通过点击 "Submit" 按钮提交表单数据时, 表单数据将发送至`<form>`标签中 `action` 属性中指定的脚本文件。在这个实例中, 我们指定文件来处理表单数据。如果你希望其他的PHP文件来处理该数据, 你可以修改该指定的脚本文件名。然后, 我们可以使用超级全局变量 `$_POST` 来收集表单中的 `input` 字段数据:

实例

```
<html>
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
Name: <input type="text" name="fname">
<input type="submit">
</form>
<?php
$name = $_POST['fname'];
echo $name;
?>
```

```
</body>
</html>
```

[运行实例 »](#)

PHP \$_GET

PHP \$_GET 同样被广泛应用于收集表单数据，在HTML form标签的指定该属性："method="get"。

\$_GET 也可以收集URL中发送的数据。

假定我们有一个包含参数的超链接HTML页面：

```
<html>
<body>

<a href="test_get.php?subject=PHP&web=runoob.com">Test $GET</a>

</body>
</html>
```

当用户点击链接 "Test \$GET", 参数 "subject" 和 "web" 将发送至"test_get.php",你可以在 "test_get.php" 文件中使用 \$_GET 变量来获取这些数据。

以下实例显示了 "test_get.php" 文件的代码:

实例

```
<html>
<body>
<?php
echo "Study " . $_GET['subject'] . " @ " . $_GET['web'];
?>
</body>
</html>
```

[运行实例 »](#)

提示： 你如果想学习更多关于 \$_POST 和 \$_GET 的知识，请访问我们的 [PHP 表单](#) 章节。

[← PHP完整表单实例](#)[PHP 表单验证 →](#)[✍ 点我分享笔记](#)

