[header() 函数 1](#_Toc278910028)

[headers\_list() 函数 4](#_Toc278910029)

[headers\_sent() 函数 5](#_Toc278910030)

[setcookie() 函数 7](#_Toc278910031)

[setrawcookie() 函数 11](#_Toc278910032)

**PHP HTTP 函数**

PHP：指示支持该函数的最早的 PHP 版本。

|  |  |  |
| --- | --- | --- |
| **函数** | **描述** | **PHP** |
| [header()](http://www.w3school.com.cn/php/func_http_header.asp) | 向客户端发送原始的 HTTP 报头。 | 3 |
| [headers\_list()](http://www.w3school.com.cn/php/func_http_headers_list.asp) | 返回已发送的（或待发送的）响应头部的一个列表。 | 5 |
| [headers\_sent()](http://www.w3school.com.cn/php/func_http_headers_sent.asp) | 检查 HTTP 报头是否发送/已发送到何处。 | 3 |
| [setcookie()](http://www.w3school.com.cn/php/func_http_setcookie.asp) | 向客户端发送一个 HTTP cookie。 | 3 |
| [setrawcookie()](http://www.w3school.com.cn/php/func_http_setrawcookie.asp) | 不对 cookie 值进行 URL 编码，发送一个 HTTP cookie。 | 5 |

# header() 函数

[PHP HTTP 函数](http://www.w3school.com.cn/php/php_ref_http.asp)

## 定义和用法

header() 函数向客户端发送原始的 HTTP 报头。

认识到一点很重要，即必须在任何实际的输出被发送之前调用 header() 函数（在 PHP 4 以及更高的版本中，您可以使用输出缓存来解决此问题）：

<html>

<?php

// 结果出错

// 在调用 header() 之前已存在输出

header('Location: http://www.example.com/');

?>

### 语法

header(string,replace,http\_response\_code)

|  |  |
| --- | --- |
| **参数** | **描述** |
| string | 必需。规定要发送的报头字符串。 |
| replace | 可选。指示该报头是否替换之前的报头，或添加第二个报头。  默认是 true（替换）。false（允许相同类型的多个报头）。 |
| http\_response\_code | 可选。把 HTTP 响应代码强制为指定的值。（PHP 4 以及更高版本可用） |

## 提示和注释

注释：从 PHP 4.4 之后，该函数防止一次发送多个报头。这是对头部注入攻击的保护措施。

## 例子

### 例子 1

<?php

// Date in the past

header("Expires: Mon, 26 Jul 1997 05:00:00 GMT");

header("Cache-Control: no-cache");

header("Pragma: no-cache");

?>

<html>

<body>

...

...

注释：用户可能会设置一些选项来更改浏览器的默认缓存设置。通过发送上面的报头，您可以覆盖任何这些设置，强制浏览器不进行缓存！

### 例子 2

提示用户保存一个生成的 PDF 文件（Content-Disposition 报头用于提供一个推荐的文件名，并强制浏览器显示保存对话框）：

<?php

header("Content-type:application/pdf");

// 文件将被称为 downloaded.pdf

header("Content-Disposition:attachment;filename='downloaded.pdf'");

// PDF 源在 original.pdf 中

readfile("original.pdf");

?>

<html>

<body>

...

...

注释：微软 IE 5.5 存在一个阻止以上机制的 bug。通过升级为 Service Pack 2 或更高的版本，可以解决该 bug。

# headers\_list() 函数

[PHP HTTP 函数](http://www.w3school.com.cn/php/php_ref_http.asp)

## 定义和用法

headers\_list() 函数返回已发送的（或待发送的）响应头部的一个列表。

该函数返回包含报头的数组。

### 语法

headers\_list()

## 提示和注释

提示：如需确定是否已发送报头，请使用 headers\_sent() 函数。

## 例子

<?php

setcookie("TestCookie","SomeValue"');

header("X-Sample-Test: foo");

header('Content-type: text/plain');

?>

<html>

<body>

<?php

// 发送哪些报头？

var\_dump(headers\_list());

?>

</body>

</html>

# headers\_sent() 函数

[PHP HTTP 函数](http://www.w3school.com.cn/php/php_ref_http.asp)

## 定义和用法

headers\_sent() 函数检查 HTTP 报头是否发送/已发送到何处。

如果报头已发送，则返回 true，否则返回 false。

### 语法

headers\_sent(file,line)

|  |  |
| --- | --- |
| **参数** | **描述** |
| file,line | 可选。如果设置 file 和 line 参数，headers\_sent() 会把输出开始的 PHP 源文件名和行号存入 file 和 line 变量中。 |

## 提示和注释

注释：一旦报头块已经发送，就不能使用 header() 添加多个报头行。

注释：可选的 file 和 line 参数是 PHP 4.3 中新加的。

## 例子

### 例子 1

<?php

// 如果报头未发送，则发送一个

if (!headers\_sent())

{

header("Location: http://www.w3school.com.cn/");

exit;

}

?>

<html>

<body>

...

...

### 例子 2

使用可选的 file 和 line 参数：

<?php

// 传递 $file 和 $line，供日后使用

// 不要预先为它们赋值

if (!headers\_sent($file, $line))

{

header("Location: http://www.w3school.com.cn/");

exit;

// Trigger an error here

}

else

{

echo "Headers sent in $file on line $line";

exit;

}

?>

<html>

<body>

...

...

# setcookie() 函数

[PHP HTTP 函数](http://www.w3school.com.cn/php/php_ref_http.asp)

## 定义和用法

setcookie() 函数向客户端发送一个 HTTP cookie。

cookie 是由服务器发送到浏览器的变量。cookie 通常是服务器嵌入到用户计算机中的小文本文件。每当计算机通过浏览器请求一个页面，就会发送这个 cookie。

cookie 的名称指定为相同名称的变量。例如，如果被发送的 cookie 名为 "name"，会自动创建名为 $user 的变量，包含 cookie 的值。

必须在任何其他输出发送前对 cookie 进行赋值。

如果成功，则该函数返回 true，否则返回 false。

### 语法

setcookie(name,value,expire,path,domain,secure)

|  |  |
| --- | --- |
| **参数** | **描述** |
| name | 必需。规定 cookie 的名称。 |
| value | 必需。规定 cookie 的值。 |
| expire | 可选。规定 cookie 的有效期。 |
| path | 可选。规定 cookie 的服务器路径。 |
| domain | 可选。规定 cookie 的域名。 |
| secure | 可选。规定是否通过安全的 HTTPS 连接来传输 cookie。 |

## 提示和注释

注释：可以通过 $HTTP\_COOKIE\_VARS["user"] 或 $\_COOKIE["user"] 来访问名为 "user" 的 cookie 的值。

注释：在发送 cookie 时，cookie 的值会自动进行 URL 编码。接收时会进行 URL 解码。如果你不需要这样，可以使用 [setrawcookie()](http://www.w3school.com.cn/php/func_http_setrawcookie.asp) 代替。

## 例子

### 例子 1

设置并发送 cookie：

<?php

$value = "my cookie value";

// 发送一个简单的 cookie

setcookie("TestCookie",$value);

?>

<html>

<body>

...

...

<?php

$value = "my cookie value";

// 发送一个 24 小时候过期的 cookie

setcookie("TestCookie",$value, time()+3600\*24);

?>

<html>

<body>

...

...

### 例子 2

检索 cookie 值的不同方法：

<html>

<body>

<?php

// 输出个别的 cookie

echo $\_COOKIE["TestCookie"];

echo "<br />";

echo $HTTP\_COOKIE\_VARS["TestCookie"];

echo "<br />";

// 输出所有 cookie

print\_r($\_COOKIE);

?>

</body>

</html>

输出：

my cookie value

my cookie value

Array ([TestCookie] => my cookie value)

### 例子 3

通过把失效日期设置为过去的日期/时间，删除一个 cookie：

<?php

// 把失效日期设置为一小时前

setcookie ("TestCookie", "", time() - 3600);

?>

<html>

<body>

...

...

### 例子 4

创建一个数组 cookie：

<?php

setcookie("cookie[three]","cookiethree");

setcookie("cookie[two]","cookietwo");

setcookie("cookie[one]","cookieone");

// 输出 cookie （在重载页面后）

if (isset($\_COOKIE["cookie"]))

{

foreach ($\_COOKIE["cookie"] as $name => $value)

{

echo "$name : $value <br />";

}

}

?>

<html>

<body>

...

...

输出：

three : cookiethree

two : cookietwo

one : cookieone

# setrawcookie() 函数

[PHP HTTP 函数](http://www.w3school.com.cn/php/php_ref_http.asp)

## 定义和用法

setrawcookie() 函数不对 cookie 值进行 URL 编码，发送一个 HTTP cookie。

cookie 是由服务器发送到浏览器的变量。cookie 通常是服务器嵌入到用户计算机中的小文本文件。每当计算机通过浏览器请求一个页面，就会发送这个 cookie。

cookie 的名称指定为相同名称的变量。例如，如果被发送的 cookie 名为 "name"，会自动创建名为 $user 的变量，包含 cookie 的值。

必须在任何其他输出发送前对 cookie 进行赋值。

如果成功，则该函数返回 true，否则返回 false。

### 语法

setcookie(name,value,expire,path,domain,secure)

|  |  |
| --- | --- |
| **参数** | **描述** |
| name | 必需。规定 cookie 的名称。 |
| value | 必需。规定 cookie 的值。 |
| expire | 可选。规定 cookie 的有效期。 |
| path | 可选。规定 cookie 的服务器路径。 |
| domain | 可选。规定 cookie 的域名。 |
| secure | 可选。规定是否通过安全的 HTTPS 连接来传输 cookie。 |

## 提示和注释

注释：可以通过 $HTTP\_COOKIE\_VARS["user"] 或 $\_COOKIE["user"] 来访问名为 "user" 的 cookie 的值。

注释：setrawcookie() 与 [setcookie()](http://www.w3school.com.cn/php/func_http_setcookie.asp) 几乎完全相同，不同的是不会在发往客户机时，对 cookie 值进行自动 URL 编码。

## 例子

### 例子 1

设置并发送 cookie：

<?php

$value = "my cookie value";

// 发送一个简单的 cookie

setrawcookie("TestCookie",$value);

?>

<html>

<body>

...

...

<?php

$value = "my cookie value";

// 发送一个 24 小时候过期的 cookie

setrawcookie("TestCookie",$value, time()+3600\*24);

?>

<html>

<body>

...

...

### 例子 2

检索 cookie 值的不同方法：

<html>

<body>

<?php

// 输出个别的 cookie

echo $\_COOKIE["TestCookie"];

echo "<br />";

echo $HTTP\_COOKIE\_VARS["TestCookie"];

echo "<br />";

// 输出所有 cookie

print\_r($\_COOKIE);

?>

</body>

</html>

输出：

my cookie value

my cookie value

Array ([TestCookie] => my cookie value)

### 例子 3

通过把失效日期设置为过去的日期/时间，删除一个 cookie：

<?php

// 把失效日期设置为一小时前

setrawcookie ("TestCookie", "", time() - 3600);

?>

<html>

<body>

...

...

### 例子 4

创建一个数组 cookie：

<?php

setrawcookie("cookie[three]","cookiethree");

setrawcookie("cookie[two]","cookietwo");

setrawcookie("cookie[one]","cookieone");

// 输出 cookie （在重载页面后）

if (isset($\_COOKIE["cookie"]))

{

foreach ($\_COOKIE["cookie"] as $name => $value)

{

echo "$name : $value <br />";

}

}

?>

<html>

<body>

...

...

输出：

three : cookiethree

two : cookietwo

one : cookieone