

## 第8章 PHP中对URL、HTTP的处理

- 由于Web开发涉及到很多和URL、HTTP协议相关的内容，PHP程序中会经常遇到对URL和HTTP相关内容的处理，其中包括对URL的编码和解码、设置或获取一些HTTP头信息，以及通过cookie验证用户身份等。PHP提供了不少函数，为开发人员提供这些方面的支持，本章就分别介绍和URL、HTTP有关的处理函数。



## 8.1 URL地址分析

- 在PHP的实际应用中，对URL地址的处理主要涉及URL的编码、解码及分析三个方面，PHP提供了3个函数，对URL进行处理。他们是：urlencode、urldecode和parse\_url。本节将分别介绍这三个函数的用法。



## 8.1.1 为什么要对URL进行处理

- 有时需要将一个URL作为一个整体，通过另一个URL传递给一个程序，即将一个URL作为参数传递给另外一个URL。因为被传递的URL里也可能带有查询字符串，为避免和原URL的参数冲突，需要将被传递的URL进行处理，也就是要对URL做编码。此外，对于中文参数值的传递，也应该进行URL编码。因此，对URL编码是PHP开发中经常碰到的问题，初学者有必要掌握基本的处理函数。



## 8.1.2 对URL进行编码

- PHP提供函数urlencode对URL进行编码，该函数语法如下所示。
- `string urlencode ( string $str);`
- 函数urlencode()接受一个字符串参数作为输入，返回值也是一个字符串，返回值字符串中所有的非字母和数字字符变成一个百分号(%)和一个两位的十六进制数，如字符串“&”会被转换成“%26”。需要特被说明的是，空格则会被转换成一个加号(+)。另外，这个函数不会对-、\_和.（英文句点）符号做转换。



### 8.1.3 对URL进行解码

- PHP提供函数urldecode()对编码过的字符串解码，这个函数可以看作是上小节所介绍的函数urlencode()的逆运算。该函数语法如下所示。
- `string urldecode (string $str )`
- 函数urldecode()的参数是一个字符串，此函数将编码后诸如###形式的字符串解码，如将“id%3D007”转换成“id=007”。代码8-2所示程序会将上小节编码的URL地址解码，然后将结果输出到页面。



## 8.1.4 对URL地址进行分析

- 一个有效的URL地址由多个部分组成，如协议部分、主机（或IP地址）部分、端口号部分等。有时在程序中需要使用URL地址的某些部分，单纯通过字符串处理函数也可以获取这些部分，但那样做会降低效率，也较容易出现一些错误。PHP提供函数 `parse_url()`，专门用来分析一个有效的URL地址，通过这个函数获取URL地址的各个部分。该函数语法如下所示。
- `parse_url ( $url );`
- 函数 `parse_url()` 的参数是一个URL地址字符串，返回值是一个关联数组，此数组中存储了URL地址的各个部分，关联数组的每个索引对应URL地址的每个部分。这个关联数组包括的内容如下所示。
- `scheme`: 协议。                      `host`: 主机IP地址或域名。
- `port`: 端口号。                      `user`: 用户名。
- `pass`: 用户密码。                      `path`: 访问路径。
- `query`: 查询参数字符串。                      `fragment`: 参数片段。



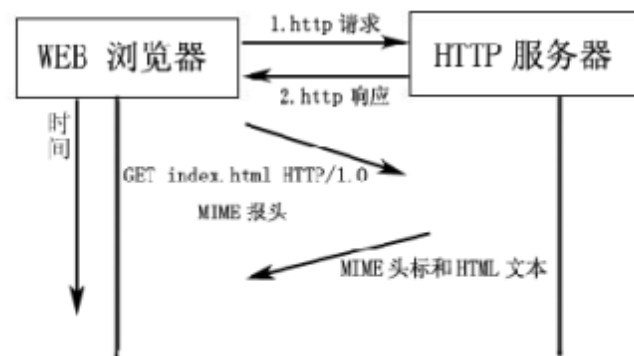
## 8.2 HTTP应用

- 在第1章的第4节简要地阐述了HTTP协议，对那节内容的理解将有助于读者学习本节的知识。这一节主要介绍一些和HTTP处理有关的函数。这些函数主要如下所示。
- 函数header()。
- 函数setcookie()。



## 8.2.1 生成一个HTTP头

- 服务器在将HTML文档传送至客户端之前，会先发送一些数据的说明信息到浏览器，最后发送HTML文档数据，这些说明信息被称作头标。



HTTP 通信过程





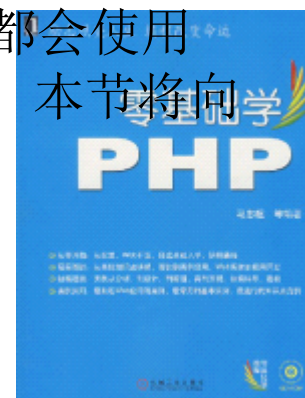
## 8.2.2 获取HTTP头信息

- 通过PHP的HTTP预定义变量\$\_SERVER可以获取页面的HTTP头信息。这个变量是一个关联数组，其每个索引都对应一个HTTP头信息，下面是一些表示主要的头信息的索引值。
- HTTP\_HOST：表示HTTP主机。
- SERVER\_SIGNATURE：表示服务器签名。
- SERVER\_NAME：表示服务器名称。
- SERVER\_ADDR：表示服务器地址。
- SERVER\_PORT：表示服务器端口。
- SERVER\_PROTOCOL：表示服务器协议。



## 8.3 在PHP中生成cookie

- cookie是存放在客户端的一组数据，它由服务器端的脚本程序生成、读写，并且以文件的形式存放在远程客户端，它的作用主要是用来识别用户身份。PHP中的session也通过cookie来区分不同的用户，它可以很容易地为每一个用户在服务器端建立一个临时的数据通道，通过这个通道，服务器实现了对不同用户的区分或者验证了不同用户的身份。
- cookie在Web开发中向来扮演着比较重要的角色，它是Web开发中历史悠久并且经常使用的技术之一，在PHP中体现的更为明显，在设计用户身份验证的系统中，往往都会使用cookie。PHP中可以通过函数方便地使用cookie，本节将向读者介绍如何在PHP程序中使用cookie。



## 8.3.1 生成cookie

- 函数setcookie()可以在PHP程序中生成cookie。由于cookie是HTTP头标部分的内容，因此必须在输出任何数据之前调用setcookie()，这个限制和函数header()类似。函数setcookie()的语法如下所示。
- `bool setcookie( string name [, string value [, int expire [, string path [, string domain [, bool secure]]]] )`
- setcookie共有6个参数，分别是：
- name: 表示cookie的名称。
- value: 表示该cookie的值，保存在客户端，因此不要保存敏感或机密的数据。这个参数为空字符串时，表示撤销客户端中该cookie的资料。
- expire: 表示该cookie有效的截至时间，即过期时间，该参数必须是整型。 path: 表示该cookie有效路径。
- domain: 表示该cookie有效的域名。
- secure: 表示在https的安全传输时才有效。



## 8.3.2 获取cookie

- 当cookie设置后，可以通过PHP预定义变量\$\_COOKIE来获取cookie。不过，只能在其他页面使用这个变量来获取设置过的cookie，因为PHP中，被设置的cookie并不会在本页生效，除非该页面被刷新。



显示多个 cookie 的值



### 8.3.3 cookie的有效期

- cookie有生命周期，即cookie只在一段时间内是有效的。通常，当用户退出IE或者Mozilla浏览器时，cookie就会被删除。如果希望延长或者缩短cookie的有效期，可以向函数setcookie()传递第3个参数，来设置cookie的有效期。下面的示例代码演示了为cookie设置不同的失效时间。
- ```
setcookie('cookie_one','i_am_cookie1',time() + 60*60);
```

 // 设置cookie 1小时后失效
- ```
setcookie('cookie_two','i_am_cookie2',time() + 60*60*24);
```

 // 设置cookie 1天后失效
- ```
//设置cookie于2008年1月1号中午12点失效
```
- ```
setcookie('cookie_three','i_am_cookie3',mktime(12,0,0,1,1,2008));
```



## 8.3.4 cookie的有效路径

- 通常，客户端的cookie只会回送给那些和设置这个cookie的程序在同一目录（或下级目录）的页面。例如，一个由 `http://www.somesite.com/index.php` 设置的cookie，会被所有到 `www.somesite.com` 请求回送至服务器，因为 `index.php` 在服务器的根目录下。而由 `http://www.somesite.com/users/list.php` 设置的cookie，随着请求，客户端的cookie会被回送到 `users` 目录下的其他页面，比如可以将cookie回送到 `http://www.somesite.com/users/login.php` 但不能回送至 `http://www.somesite.com/orders/info.php`
- 如果需要客户端的请求把cookie传回到不同的路径下，可以通过向函数 `setcookie()` 传入第4个参数，通过该参数设置cookie在服务器端的有效路径。最灵活的方式是，设置cookie的有效路径为 `/`，它表示用 `setcookie()` 设置的cookie在整个服务器域名内有效。设置为 `/mypath/`，那么，该cookie只在域名的 `/mypath` 目录及其子目录下有效。下面的代码设置cookie的有效路径为一个指定的目录。
- `setcookie( 'mycookie' , ' delicious' , 0, ' /ck_path/' );`



## 8.3.5 删除cookie

- 在PHP程序中删除cookie比较简单，也是通过函数 `setcookie()` 完成。通过下面的代码就可以实现删除一个cookie。
- `setcookie( 'mycookie' , ' ' );`
- 这段代码通过将cookie的值设为空，来达到删除cookie的目的。如果设置cookie时，为函数 `setcookie()` 每个参数都提供了特定的值，那么在删除cookie时，仍然需要提供这些参数，以便PHP可以正确地删除cookie。



## 8.4 小结

- 本章讲述了在PHP程序中处理URL、HTTP信息及使用cookie的函数，涉及到的函数和内容如下所示。
- 对URL进行编码的函数urlencode()。
- 对URL进行解码的函数urldecode()。
- 分析URL内容的函数parse\_url()。
- 处理HTTP头的函数header()。
- 通过全局预定义变量\$\_SERVER获取HTTP头信息。
- 通过函数setcookie()完成对cookie的操作和处理。

