第

# PHP 与浏览器交互编程

在互联网应用中,许多 Web 应用程序往往需要获取上网用户的信息,例如登录的用户 名和密码、留言内容、用户是否在线、在线时间等,这些信息的采集可以通过本章的相关知识 实现。本章主要介绍 PHP 与浏览器之间数据交互的编程技术,包括提交 Web 表单数据、接 收 Web 表单数据、文件上传、保存用户会话状态的 Session 和 Cookie 应用。

#### 5.1 Web 表单数据的提交



浏览器与 Web 服务器的 Web 应用程序之间采用 HTTP 进行数据传输,它基于"请求/ 响应"模式。浏览器向 Web 服务器提交数据的方式有两种:GET 方式和 POST 方式。当 浏览器向 Web 服务器某个 PHP 程序发送一个"GET 请求"时,浏览器以 GET 方式向 Web 服务器的某个 PHP 程序提交数据;当浏览器向 Web 服务器某个 PHP 程序发送一个 "POST 请求"时,浏览器以 POST 方式向 Web 服务器的某个 PHP 程序提交数据。

提交 Web 表单数据的方法有两种:GET 方式和 POST 方式。具体采用哪种方法提交 表单数据,由<form>表单元素的 method 属性指定。

#### 5.1.1 使用 GET 方式提交表单数据

GET 方式是提交表单数据的默认方法。GET 方式将表单数据以查询字符串形式附加 到 URL 后面,并作为 URL 的一部分内容发送到 Web 服务器。GET 方式传递参数的格式 如下。

http://URL?name1=value1&name2=value2...

其中, URL为 Web 服务器某个数据接收程序的网址, 问号"?"后面的内容为查询字符串, 列出了需要传递给 PHP 程序的各个参数, 参数以"参数名=参数值"格式定义, name1, name2

<u>HP+jQ</u>uery+Vue.js全栈开发

D)

从入门到实战・微课视频版

等是参数名,也就是表单的元素名,参数名要根据 PHP 程序中的变量名而定;value1,value2 等是参数值,即表单中对应元素的值。各参数之间用符号"&"相连,URL 与查询字符串之间用符号"?"相连。

【例 5.1】 定义一个用户注册表单网页文件 register.html。

```
<html>
<head>
<meta charset="utf-8">
<title>注册用户</title>
</head>
<body>
<form action="register.php" method="get">
用户名:<input type="text" name="username"><br>
密码:<input type="password" name="password" ><br>
密码:<input type="password" name="confirmpassword" ><br>
<input type="submit" value="注册">
</form>
</body>
</html>
```

在该网页文件中, < form > 表单元素的 action 属性指定了表单数据由 PHP 程序 register.php(其代码参见 5.2.1 节)接收并处理, method 属性指定了表单数据以 GET 方式 向 register.php 程序传递数据。因此,访问 register.html 文件,输入表单的注册信息,如 图 5.1 所示,然后单击"注册"按钮,显示如图 5.2 所示的结果。

<b>同</b> 注册用户	×	x
← → C	③ localhost/ch05/register.html ☆	;
用户名:	user1	*
密 码:		
确认密码:	•	100
1200		•

#### 图 5.1 注册用户界面



图 5.2 PHP 程序接收表单数据后的运行结果

从如图 5.2 所示的浏览器的地址栏可以看出,以 GET 方式传递表单数据时,将表单数 据附加到接收表单数据的 PHP 程序网址之后。而这对于传递一些重要信息而言,建议不 要使用 GET 方式提交表单数据。

#### 5.1.2 使用 POST 方式提交表单数据

POST 方式提交表单数据是通过在<form>表单元素中加入属性"method=post",将 表单的数据以 POST 方式提交到 Web 服务器的 PHP 程序,再由 PHP 程序接收表单数据 并进行处理。

【例 5.2】 一个简单的调查问卷网页文件 eg5\_2.html。

```
<html>
<head>
<meta charset="utf-8">
<title>问卷调查</title>
</head>
<body>
<form name="form1" method="post" action="eg5_4.php">
姓名 <input name="name" type="text" id="name"><br>
喜爱的运动 <input name="love" type="text" id="love"><br>
<input type="submit" name="Submit" value="确定">
</form>
</body>
</html>
```

在该网页文件中, < form > 元素的 method 属性指定以 POST 方式提交表单数据, action 属性指定了接收并处理表单数据的 Web 服务器程序是 eg5\_4.php 程序(其代码参见 5.2.2 节)。访问 eg5\_2.html 文件, 如图 5.3 所示, 输入图中内容, 单击"确定"按钮后, 由 eg5\_4.php 程序接收表单数据, 进行处理, 输出结果如图 5.4 所示。

🛛 问卷调查	×	×
← → C [	Iocalhost/ch05/eg5_2.html	\$ :
性名 小明		
喜爱的运动 日	已球	
确定		

图 5.3 输入表单数据

R localhos	t/ch05/eg5_4. ×	¢
$\boldsymbol{\varepsilon}  \Rightarrow  \boldsymbol{G}$	🛈 localhost/ch05/eg5_4.php 🛛 🛧	:
小明,您喜爱	的运动是:足球	

图 5.4 PHP 程序接收表单数据的运行结果

从图 5.4 可以看出,采用 POST 方式时,表单上输入的数据不会附加到 action 属性所设

| 从入门到实战・微课视频版

定的 URL 之后发送,这对于传递重要信息而言,POST 方式是一种安全的提交数据方法。

#### 5.1.3 两种提交方式的差别

GET 方式和 POST 方式都可以提交浏览器中表单的数据,但两者存在如下差别。

(1) POST 方式比 GET 方式安全。GET 提交方式把数据附加到 URL 后面的查询字 符串中,并存放在历史记录中,用户可以修改 URL 的参数值,达到不同目的。不建议采用 GET 方式。

(2) POST 方式可以提交更多的数据。POST 方式提交的数据没有大小限制,而GET 方式提交的数据大小受到 URL 长度的限制,只用于提交少量的数据。

(3) 不同的提交方式, Web 服务器接收数据的方式也有所不同。5.2 节介绍其接收数据的具体方法。

# 5.2 PHP 接收 Web 表单数据

视频讲解

PHP 提供了预定义全局变量 \$\_GET、\$\_POST、\$\_REQUEST,用来获取浏览器表单数据,这些全局变量都是数组类型。

#### 5.2.1 利用 \$ \_GET 全局变量接收表单数据

当浏览器以 GET 方式提交表单数据时,Web 服务器的 PHP 程序可以用 \$ \_GET 全局数组变量读取表单的各个元素值。通过数组元素" \$ \_GET [下标]"获取表单的某一元素值,其中,下标是表单的元素名,必须与网页中表单的元素名称相同,而且大小写也须一致。

【例 5.3】 接收例 5.1 注册用户表单数据的 PHP 程序 register.php。

```
<?php
echo "接收到的注册信息如下<br>";
echo "用户名:".$_GET["username"];
echo "<br>密码:".$_GET["password"];
echo "<br>确认密码:".$_GET["confirmpassword"];
?>
```

在该程序中, \$\_GET["username"]数组元素读取表单中元素名为 username 的元素值, 即 输入的用户名, \$\_GET["password"]获取表单的密码, \$\_GET["confirmpassword"]获取表单的确认密码。

#### 5.2.2 利用 \$\_POST 全局变量接收表单数据

当浏览器以 POST 方式提交表单数据时,Web 服务器的 PHP 程序可以用 \$\_POST 全局数组变量读取表单的各个元素值。通过数组元素" \$\_POST[下标]"获取表单的某一元

素值,其中,下标是表单的元素名,必须与网页中表单的元素名称相同,而且大小写也须 一致。

【例 5.4】 接收例 5.2 调查问卷表单数据的 PHP 程序 eg5\_4.php。

```
<?php
echo $_POST["name"];
echo ",您喜爱的运动是:".$_POST["love"];
?>
```

在该程序中,\$\_POST["name"]数组元素获取表单中输入的姓名,\$\_POST["love"] 数组元素获取表单中输入的运动名称。

#### 5.2.3 利用 \$\_REQUEST 全局变量接收表单数据

使用 \$ \_REQUEST 全局变量获取 POST 方式、GET 方式,以及 HTTP Cookie 传递到 PHP 程序的数据。例 5.3 和例 5.4 的程序中,全局变量 \$ \_GET 和 \$ \_POST 都可以用 \$ \_REQUEST 替换, \$ \_Cookie 全局变量也可以用 \$ \_REQUEST 替换。

在 HTML 表单中,除了使用文本框元素外,还可以使用单选按钮、复选框、下拉列表等 表单元素输入数据。对于同一组单选按钮元素,它们的名字必须相同。对于同一组的复选 框元素,它们的名字必须相同,并且在其名字后加上一对方括号"[]"。这样,PHP 程序把这 些选中的复选框的值作为一个数组处理。下面给出一个表单综合实例。

【例 5.5】 创建一个填写用户个人信息的表单网页文件 eg5\_5.php,它包含文本框、复选框、单选按钮和下拉列表元素。

eg5\_5.php 文件的主要代码如下。

```
...
<body>
<form name="form1" method="post" action="eq5 6.php">
 生名 <input name="myname" type="text"><br>
   性别<input name="gender" type="radio" value="男" checked>男
   <input type="radio" name=" gender" value="女">女<br>
   出生年月<select name="year">
     <?php
     for ($i=2019;$i>=1900;$i--) {
       echo "<option value='".$i."'>".$i."</option>";
     }
     ^{2}
     </select>年
     <select name="month">
       <?php
       for ($i=1;$i<=12;$i++) {</pre>
         echo "<option value='".$i."'>".$i."</option>";
       }
       ? >
```

从入门到实战・微课视频版

在该网页文件中,两个单选按钮元素的 name 属性值相同,均为 gender;4 个复选框元素的 name 属性值都是 interest[],这样 PHP 程序把提交的复选框元素视为一个数组。

【例 5.6】 接收并处理例 5.5 表单数据的 PHP 程序(eg5\_6.php)。

```
<?php
echo $_REQUEST["myname"];
echo "<br>你的性别是:".$_REQUEST["gender"];
echo "<br>出生年月:".$_REQUEST["year"]."年".$_REQUEST["month"]."月";
echo "<br>感兴趣的编程语言是:";
for ($i=0;$i<count($_REQUEST["interest"]);$i++) {
    echo $_REQUEST["interest"][$i]." ";
}
2>
```

首先访问 eg5\_5.php,显示如图 5.5 所示的页面,输入用户信息后,单击"确定"按钮,将 表单数据传送给 Web 服务器端的 eg5\_6.php 程序,由它进行处理,生成新的网页,返回给用 户浏览器,输出结果如图 5.6 所示。



图 5.5 访问 eg5\_5.html 显示的页面



图 5.6 返回的个人信息

# 5.3 文件上传

在许多 Web 应用程序中经常使用文件上传功能,例如把图片、文档文件、压缩文件等上 传到 Web 服务器。在 PHP 中实现文件上传,首先要在配置文件 php.ini 中对上传文件的配 置做一些设置,然后编写 PHP 程序,利用全局变量 \$\_FILES 获取上传文件的有关信息,再 利用 move\_uploaded\_file()函数实现文件的上传。

### 5.3.1 上传文件的设置

PHP 配置文件 php.ini 的[File Uploads]节定义了一些与上传文件相关的配置选项,包括是否支持上传文件、上传文件的临时目录、上传文件的大小、指令执行的时间等,修改这些选项的值,可以满足特定的文件上传需要。上传相关的选项如下。

(1) file\_uploads: 是否允许 HTTP 文件上传。其默认值为 On,表示支持文件上传。

(2) upload\_tmp\_dir:存储上传文件的临时目录。一般使用系统默认目录,也可以自行 指定其目录。XAMPP 套件中其默认目录为"C:\xampp\tmp"。

(3) upload\_max\_filesize: 允许上传文件大小的最大值,以 MB 为单位,默认值为 2MB。 如果要上传超过 2MB 的文件,需要修改此选项的值。

(4) max\_file\_uploads: 允许一次上传的文件数量的最大值,默认值为 20。

除了以上 File Uploads 节的上传相关选项外,往往还需要修改以下几个影响文件上传的选项。

(5) max\_size: 以 POST 方式提交表单数据时,PHP 能接收的 POST 数据的上限值,默 认为 8MB,表示表单的所有数据(单行文本框、多行文本框等)的大小之和必须小于 8MB,否则 PHP 不能获取表单的所有数据,即 \$\_GET、\$\_POST、\$\_FILES 将是空数组。如果该选项值设置为 0,则不限制表单数据大小。

(6) max\_execution\_time: 配置单个 PHP 程序在 Web 服务器端上执行的最长时间,单位是 s,默认值为 30s。如果要上传大文件,必须将该选项值修改为更长的时间,否则超过了 PHP 程序执行的最大时间,仍然不能实现上传文件。如果该选项值设为 0,表示不限制执行时间。

(7) memory\_limit: 单个 PHP 程序在 Web 服务器端执行时占用的最大内存空间,默认值为 128MB。当设置为一1 时,表示不限制。

更改了 php.ini 配置文件的选项后,必须重新启动 Apache Web 服务器,使配置生效。

## 5.3.2 \$\_FILES 全局变量

当表单以 POST 方式上传文件时,PHP 程序可以用 \$\_FILES 全局数组变量获取上传 文件相关的信息,如上传的文件名、文件大小、文件 MIME 类型、临时文件名等。 \$\_FILES 是一个二维数组,假设表单中上传文件元素的名称为 filename,那么 \$\_FILES 数组的各个

从入门到实战・微课视频版

数组元素含义如表 5.1 所示。

名称	含 义	
\$_FILES['filename']['name']	上传文件的文件名	
\$_FILES['filename']['type']	上传文件的 MIME 类型。MIME 类型规定各种文件格式的类型,如"image/gif"表示上传的是 gif 类型的文件	
\$_FILES['filename']['size']	上传文件的大小,单位为 B	
\$_FILES['filename']['tmp_name']	文件被上传后在 Web 服务器端存储的临时文件名	
\$_FILES['filename']['error']	文件上传结果的状态码。其值和意义如下: 0:文件上传成功 1:文件大小超过 upload_max_filesize 选项的上限值 2:文件大小超过 FORM 表单中 MAX_FILE_SIZE 参数的上限值 3:文件只有部分被上传 4:没有选择上传文件	

表 5.1 \$\_FILES 全局数组元素的含义



## 5.3.3 文件上传的实现

在 PHP 中,要实现文件上传,首先用 is\_uploaded\_file()函数判断指定的文件是否通过 HTTP POST 方式上传,如果是,则返回 true,再继续执行 move\_uploaded\_file()函数完成 文件的上传。

1. is\_uploaded\_file()函数

is\_uploaded\_file()函数判断指定的文件是否通过 HTTP POST 方式上传。如果是,返回 true。其语法格式为

```
bool is_uploaded_file(string $filename)
```

其中,\$filename参数是\$\_FILES['filename']['tmp\_name']的值,不是\$\_FILES['filename'] ['name']的值。

#### 2. move\_uploaded\_file()函数

move\_uploaded\_file()函数将上传的文件移到指定的目录。如果成功,返回 true,否则 返回 false。其语法格式为

bool move\_uploaded\_file(string \$filename, string \$destination)

其中, \$ filename 参数是上传文件的临时文件名, 即 \$ \_FILES['filename']['tmp\_name']的值; \$ destination 是上传的文件在服务器端存储的目标路径和文件名。

【例 5.7】 创建一个表单,在表单中添加一个文件域,用来上传小于 2MB 的文件,将上传的文件存储到 Web 服务器站点根目录下的 upload 目录(eg5\_7.php)。

eg5\_7.php 文件的主要代码如下。

```
<form method="post" action="eg5_7.php" enctype="multipart/form-data">
```

```
选择要上传的文件:
   <input type="file" name="upfile" >
   <input type="submit" name="submit" value="上传">
 </form>
<?php
$dir="../upload";
if(isset($ FILES["upfile"])) {
   $fileinfo=$ FILES["upfile"];
   if($fileinfo['error']>0) {
      echo "上传错误:";
      switch($fileinfo['error']) {
          case 1:
             echo "上传文件大小超过配置文件规定的最大值";
             break;
          case 2:
              echo "上传文件大小超过表单数据的最大值";
             break;
          case 3:
             echo "上传文件不全";
             break;
          case 4:
             echo "没有上传文件";
             break;
       }
   }else{
     if (!is dir($dir)) {
         mkdir($dir);
      }
     $destfile=$path."/".$fileinfo["name"];
      if(is uploaded file($fileinfo["tmp name"])) {
         if(move uploaded file($fileinfo["tmp name"],$destfile))
             echo "文件上传成功";
          else
             echo "上传失败";
      }else{
          echo "文件上传不合法";
      }
   }
}
? >
```

在上述 PHP 程序中, \$\_FILES["upfile"]的下标 upfile 必须与表单的文件域的 name 属性值相同。另外, 在执行本程序之前, 需要预先在 C:\xampp\htdocs 文件夹中创建 upload 子文件夹, 才能执行本程序。

一个表单不仅可以上传一个文件,也可以同时上传多个文件,只需要在表单中添加多个 文件域,这些文件域的 name 属性值为同一个数组即可,PHP 程序用循环语句将上传的多

从入门到实战・微课视频版

个文件逐个移到指定的目录。

```
【例 5.8】 同时上传多个文件的 PHP 程序(eg5_8.php)。 该程序的主要代码如下。
```

```
<form method="post" action="eg5 8.php" enctype="multipart/form-data">
  文件 1:<input type="file" name="upfiles[]" ><br>
  文件 2:<input type="file" name="upfiles[]" ><br>
  文件 3:<input type="file" name="upfiles[]" ><br>
  文件 4:<input type="file" name="upfiles[]" ><br>
  文件 5:<input type="file" name="upfiles[]" ><br>
  文件 6:<input type="file" name="upfiles[]" ><br>
  <input type="submit" name="submit" value="上传">
</form>
<?php
$uploads dir = '../upload';
if(isset($ FILES["upfiles"])) {
   foreach ($ FILES["upfiles"]["error"] as $key => $error) {
       $name = $ FILES["upfiles"]["name"][$key];
       if(!empty($name)){
           if ($error == UPLOAD ERR OK) {
              $tmp name = $ FILES["upfiles"]["tmp name"][$key];
              move uploaded file($tmp name, "$uploads dir/$name");
              echo "文件:$name, 上传成功<br>";
          }else{
              echo "文件:$name, 上传不合法<br>";
           }
       }
   }
}
? >
```

在浏览器中访问网址"http://localhost/ch05/eg5\_8.php",显示如图 5.7 所示的界面, 选择一些要上传的文件,单击"上传"按钮,输出效果如图 5.8 所示。同时,在 C:\xampp\ htdocs\upload 文件夹中存放了上传的文件。

←⇒	C 🛈 la	calhost/ch05/eg5_8.php 🔂	:
选择要上	传的文件	ŧ	
文件1:	选择文件	eg5 2.html	- 1
文件2:	选择文件	eq5 4.php	- 1
文件3:	选择文件	eg5 5.php	- 1
文件4:	选择文件	未选择任何文件	- 1
文件5:	选择文件	未选择任何文件	- 1
文件6:	选择文件	未选择任何文件	- 1

图 5.7 选择上传文件

			8 0	×
/ 🗵 上储	多个文件	×		
$\leftarrow \rightarrow$	C 🛈 lo	calhost/ch05/eg5_8.	php 🕁	÷
选择要」	上传的文件	Þ1		^
文件1:	选择文件	未选择任何文件		. 1
文件2:	选择文件	未选择任何文件		- 11
文件3:	选择文件	未选择任何文件		- 11
文件4:	选择文件	未选择任何文件		- 11
文件5:	选择文件	未选择任何文件		- 11
文件6:	选择文件	未选择任何文件		- 11
上传				
2.01				- 11
又件:eg	5_2.html,	上传成功		- 11
×1+:eg	5_4.pnp,	上传成初		
x1+.eg	s_s.pnp,			•

图 5.8 上传文件后的结果