

PHP 入门

5.1 PHP 简介与开发环境搭建

5.1.1 PHP 简介

PHP(Page Hypertext Preprocessor)即页面超文本预处理器,是一种被广泛使用的 Web 后端开发技术。它通常被嵌入 HTML 之中,构成一个以 php 为扩展名的文件。下面给出一个 PHP 的例子。PHP 代码通过<?php ?>标签嵌入 HTML 中。

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Hello PHP!";
    ?>
  </body>
</html>
```

这个页面在服务器端会经过预先处理,PHP 代码会被执行,而其输出结果则被替换在 PHP 代码的位置上。浏览器收到的页面中并不会包含任何 PHP 代码,如下所示。PHP 代码中的 echo 函数是最常用的输出函数,用于向页面输出一个文本。

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    Hello PHP!
  </body>
</html>
```

那么,服务器端如何执行带有 PHP 的页面呢?使用什么来执行呢?答案是通过嵌

入 Web 服务器软件中的 PHP 插件来执行。使用最为广泛的 Web 服务器有 Apache、Nginx 和 IIS 等。PHP 插件包含 PHP 所使用的可执行程序 php.exe 和符合 Apache 和 Nginx 插件规范的 DLL 文件,图 5-1 展示了 PHP 目录中文件的一小部分。其中 php5apache2_filter 是用于将 PHP 和 Apache 结合起来的 Apache 插件,而 php5isapi.dll 则是 PHP 的 IIS 插件。

libeay32.dll	2010/7/27 17:45	应用程序扩展	1,048 KB
libmcrypt.dll	2010/7/27 17:45	应用程序扩展	163 KB
libmhash.dll	2010/7/27 17:45	应用程序扩展	162 KB
libmysql.dll	2010/8/13 0:08	应用程序扩展	2,304 KB
libpq.dll	2010/7/27 17:45	应用程序扩展	454 KB
license.txt	2010/7/27 17:45	文本文档	4 KB
msql.dll	2010/7/27 17:45	应用程序扩展	56 KB
news.txt	2010/7/27 17:45	文本文档	237 KB
ntwdblib.dll	2010/7/27 17:45	应用程序扩展	284 KB
php.exe	2010/7/27 17:45	应用程序	33 KB
php.gif	2010/7/27 17:45	GIF 图片文件	3 KB
php.ini-dist	2010/7/27 17:45	INI-DIST 文件	46 KB
php.ini-recommended	2010/7/27 17:45	INI-RECOMMEN...	49 KB
php5apache.dll	2010/7/27 17:45	应用程序扩展	37 KB
php5apache_hooks.dll	2010/7/27 17:45	应用程序扩展	57 KB
php5apache2.dll	2010/7/27 17:45	应用程序扩展	37 KB
php5apache2_2.dll	2010/7/27 17:45	应用程序扩展	37 KB
php5apache2_2_filter.dll	2010/7/27 17:45	应用程序扩展	37 KB
php5apache2_filter.dll	2010/7/27 17:45	应用程序扩展	37 KB
php5embed.lib	2010/7/27 17:45	LIB 文件	660 KB
php5isapi.dll	2010/7/27 17:45	应用程序扩展	29 KB
php5nsapi.dll	2010/7/27 17:45	应用程序扩展	29 KB
php5ts.dll	2010/7/27 17:45	应用程序扩展	4,845 KB
php-apache2handler.ini	2020/2/24 21:34	配置设置	50 KB
php-cgi.exe	2010/7/27 17:45	应用程序	49 KB
php-win.exe	2010/7/27 17:45	应用程序	33 KB

图 5-1 PHP 目录中的文件

5.1.2 PHP 安装过程

PHP 通常作为 Web 服务器的插件而存在,这意味着不能单独安装 PHP,而是需要在安装了 Web 服务器的基础上再安装 PHP。为简化安装过程,这里使用 PHP 的套件 PHPnow。套件是指包含了 PHP 运行所需的几个软件的结合。PHPnow 是一个稍显古老的套件,目前已经停止更新,但因为使用的 PHP 版本较低,所以方便用来复现本书中的例子。如果希望使用较新的版本,则可以使用最新的 PHPStudy 套件,因为所包含的 PHP 版本较高,所以一些旧的 PHP 漏洞的例子无法复现。

在 Linux 下有一套叫作 LAMP 的套件是中小型网站所常用的。LAMP 是由 Linux、Apache、MySQL 和 PHP 这 4 个单词的首字母构成的,整个单词正好有了“灯”的含义。

在 Windows 下,只需安装 AMP(Apache、MySQL 和 PHP 的首字母组合)即可。其中,Apache 是 Web 服务器、MySQL 是数据库,PHP 是服务器端语言支持。PHPnow 中

包含了这三者,且安装过程非常简单。这里以 PHPnow-1.5.6 版本为例进行说明。

首先下载 PHPnow-1.5.6.zip,解压到一个不包含中文路径和空格的文件夹下。不包含中文路径和空格这一点尤为重要,编者建议软件应安装在不包含空格的英文路径中。以管理员身份运行 cmd.exe,如图 5-2 所示。



图 5-2 以管理员身份运行 cmd.exe

接着,使用 cd 命令切换到解压的 PHPnow 目录下,如图 5-3 所示。

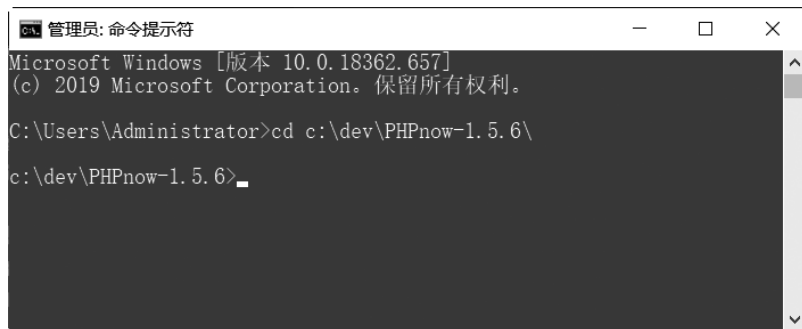


图 5-3 切换目录

执行 Setup.cmd,按照提示选择 Apache 和 MySQL 的版本,这里选择 Apache 2.0.63 版本和 MySQL 5.0.90 版本。在界面中输入 20,按 Enter 键,再输入 50,按 Enter 键,如图 5-4 所示。

随后可以看到文件在快速解压,并提示“是否执行 Init.cmd 初始化?”,这里输入“y”选择继续安装,如图 5-5 所示。



图 5-4 执行 Setup.cmd



图 5-5 执行 Init.cmd

这时可以看到,Apache 和 MySQL 在安装(见图 5-6),并提示为 MySQL 的 root 用户设置密码,这时输入 123456 这个弱口令,以方便演示。很多粗心的学生常常忘记自己设置的初始密码,所以在实验中不得不采用这种统一的密码。请切记不要在真正执行上线的项目(常被称作生产环境)中使用这种密码,这会让服务器不堪一击。



图 5-6 输入 MySQL 管理员口令

安装程序自动弹出一个浏览器界面并对软件进行测试。在“MySQL 用户密码”一栏输入 123456, 随后单击“连接”按钮, 就可以看到测试成功的提示, 如图 5-7 所示。

127.0.0.1 为何只能本地访问?
此服务器互联网 IP
112.231.52.231

Let's PHP now !

Server Information			
SERVER_NAME	127.0.0.1		
SERVER_ADDR:PORT	127.0.0.1:80		
SERVER_SOFTWARE	Apache/2.0.63 (Win32) PHP/5.2.14		
PHP_SAPI	apache2handler		
php.ini	C:/dev/PHPnow-1.5.6/php-5.2.14-Win32/php-apache2handler.ini		
网站主目录	C:/dev/PHPnow-1.5.6/htdocs		
Server Date / Time	2020-02-23 18:27:01 (+08:00)		
Other Links	phpinfo() phpMyAdmin		

PHP 组件支持	
Zend Optimizer	Yes / 3.3.3
MySQL 支持	Yes / client lib version 5.0.90
GD library	Yes / bundled (2.0.34 compatible)
eAccelerator	No

MySQL 连接测试			
MySQL 服务器	<input type="text" value="localhost"/>	MySQL 数据库名	<input type="text" value="test"/>
MySQL 用户名	<input type="text" value="root"/>	MySQL 用户密码	<input type="text"/>
			<input type="button" value="连接"/>

MySQL 测试结果	
服务器 localhost	OK (5.0.90-community-nt)
数据库 test	OK

Valid XHTML 1.0 Strict / Copyleft ! 2007-? by PHPnow.org

图 5-7 PHPnow 测试

至此,套件就安装成功了。这是一个正常的安装过程,当然也会出现意外。第一个可能出现的意外:没有以管理员身份运行导致无法安装 Apache 和 MySQL 服务。第二个可能出现的意外:端口 80 和 3306 被占用,这个问题在安装过程中会提示,只需输入一个新的端口号,如 81、3307 就可以解决了。另外,杀毒软件也有可能导致安装失败。

5.2 PHP 语法

5.2.1 PHP 执行过程

PHP 是一种服务器端描述语言,其主要功能是访问数据库和处理文本。其语法是由 Perl 语言演变而来的,而 Perl 语言本身就非常擅长处理文本,下面的代码在 Perl 和 PHP 中都可以运行。

```
$a = "world";  
$b = "hello $a";  
echo $b;
```

上述代码输出为 hello world。PHP 语法规定所有变量必须以 \$ 开头,变量不需要提前声明就可以直接使用,并且双引号字符串中出现变量名则会直接将内容替换,字符串中的 \$a 被替换为 world,所以 \$b 的值为 hello world。

当用户在浏览器中输入一个网址或单击一个链接时,浏览器会使用 HTTP 向服务器端发送一个 GET 请求,服务器端收到这个请求后会根据请求内容的不同做不同的处理。

如果请求得到一个 HTML 页面,或者 CSS 文件、JavaScript 文件,那么服务器端程序会直接读取文件内容并使用 HTTP 发送给浏览器作为这次请求的响应。如果浏览器请求的内容是以 php 为扩展名的文件,那么,支持 PHP 的服务器会调用 PHP 解释执行程序对页面中所包含的代码进行处理,将代码的输出结果发送给用户。

常见的 Web 服务器程序,如 Apache、Nginx、IIS,都可以通过配置添加对 PHP 的支持。初学者可以直接使用 PHPnow、PHPStudy 这种包含了 Apache、MySQL 和 PHP 的学习套件来进行一键安装,如图 5-8 所示。在学习 PHP 时也可以直接使用线上的 PHP 执行环境(http://www.runoob.com/try/runcode.php?filename=demo_intro&type=php)。

下面给出一个简单的 PHP 页面。

```
<!DOCTYPE html>  
<html>  
<body>  
<?php  
echo "hello world!";  
?>  
</body>  
</html>
```

名称	修改日期	类型	大小
Apache-22	2010-9-22 2:30	文件夹	
htdocs	2019-3-12 9:33	文件夹	
MySQL-5.0.90	2017-2-9 16:55	文件夹	
MySQL-5.0.90_bak	2017-12-17 21:22	文件夹	
php-5.2.14-Win32	2018-8-19 17:25	文件夹	
Pn	2017-12-17 21:22	文件夹	
PnCmnds	2010-9-22 2:51	文件夹	
Init.cm_	2010-9-22 17:06	CM_文件	10 KB
PnCp.cmd	2010-9-22 3:14	Windows 命令脚本	15 KB
Readme.txt	2010-9-22 17:49	文本文档	2 KB
更新日志.txt	2010-9-22 17:50	文本文档	3 KB
关于静态.txt	2009-4-14 23:11	文本文档	1 KB
升级方法.txt	2009-2-4 9:05	文本文档	1 KB

图 5-8 PHPnow 目录结构

在 htdocs 目录下,新建 test.php,然后在其中输入 PHP 代码。

```
<?php
echo "hello";
?>
```

随后在浏览器中输入 `http://127.0.0.1/test.php`,如果能在页面中看到 hello 字样,则运行成功(见图 5-9)。在学习 PHP 语法时可以直接将本节例子复制到 test.php 中保存,然后刷新浏览器,就可以看到执行结果。

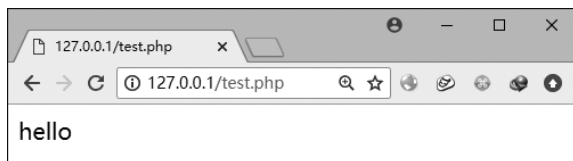


图 5-9 test.php 执行结果

如果使用的是其他 PHP 套件,可以尝试在浏览器中输入 `http://127.0.0.1:8080/test.php`,其中 8080 代表端口号,Apache 有可能不在 80 端口监听,而在 8080 端口监听。

如果新建的文件名不是 test.php,可以根据文件名修改上面的 URL。例如,文件为 abc.php,则 URL 应为 `http://127.0.0.1/abc.php`。

PHP 页面与普通的 HTML 页面的不同是,PHP 页面包含一个 `<?php ?>` 标签,在这个标签中包含 PHP 代码,代码中使用 echo 输出一个文本。这个代码会在服务器端执行,发送给浏览器的页面如下:

```
<!DOCTYPE html>
<html>
<body>
hello world!
</body>
</html>
```

可以看到,PHP 代码并没有泄露给用户,用户得到的是 PHP 代码的执行结果。

5.2.2 PHP 变量与流程控制

PHP 中的变量是隐式类型的变量,变量不需要声明就可以使用,它们必须以符号 \$ 开头,\$ 后必须以字母或下画线字符开始,并且只能包含字母、下画线和数字。除了符号 \$,其他规则与 C 语言相同。下面给出几个变量例子。

```
$a = 12;           //正确
$12 = 3;          //有语法错误
$_ = "ss";        //正确
$a_3 = 3.5;       //正确
```

注意: 在 PHP 文件中,所有的 PHP 代码均必须包裹在 `<?php ?>` 标签中,否则会被认为是纯文本而不被执行。例如:

```
<html>
<body>
echo 'hello';
</body>
</html>
```

这个程序会直接在浏览器页面输出 “echo 'hello;'”,而不是输出执行结果 hello。

PHP 变量的类型由赋值语句的右侧决定,而且可以随时变化。例如:

```
$a = 12;
$a = "hello";
```

在执行完第一条语句后,\$a 的类型为整数类型;在执行完第二条语句后,它的类型变为字符串类型。

PHP 的表达式语法与 C 语言一致,这是因为 PHP 演化自 Perl 语言,Perl 语言演化自 C 语言,与 Java、JavaScript、C# 等语言一样都从属于 C 家族语言。这种相似性大大降低了开发者的学习成本。

PHP 中的分支语句写法与 C 语言完全相同,也包含 if 语句和 switch 语句。下面看一下 if 语句:

```
<?php
$t=date("H");
if($t<20) {
    echo "Good day!";
}else{
    echo "Good night!";
}
?>
```

这段代码使用 date 函数获取了当前的 24 小时制的小时,放到变量 \$t 中,随后判断

\$t 是否在 20 点之前,如果是,则显示“Good day!”,否则显示“Good night!”。

switch 语句与 if 语句类似,只不过 C 语言中的 switch 语句仅可以支持整数和字符条件,而 PHP 中的 switch 语句可以支持字符串条件,下面举例说明。

```
$favcolor="red";
switch($favcolor)
{
case "red":
    echo "你喜欢红色!";
    break;
case "blue":
    echo "你喜欢蓝色!";
    break;
case "green":
    echo "你喜欢绿色!";
    break;
default:
    echo "你喜欢的颜色不是红色、蓝色或绿色!";
}
```

例子中的 switch 条件为变量 \$favcolor,这个变量是字符串类型的。case 语句的条件也是字符串类型的。

PHP 中的循环同 C 语言一样,也支持 while 语句、do-while 语句和 for 语句。

```
<html>
<body>
<?php
$i=1;
while($i<=5){
    echo "$i<br>";
    $i++;
}
?>
</body>
</html>
```

5.2.3 PHP 数组

PHP 中的数组与 C 语言中的数组在语法上有较大不同,可以用以下两种方法构建数组。

第一种方法:使用 array 函数。

```
$cp = array("apple","xiaomi","thinkpad");
echo "I like $cp[1]";
```

上述代码中,\$cp 为一个长度为 3 的数组,\$cp[1]的值为 xiaomi,页面运行结果如图 5-10 所示。

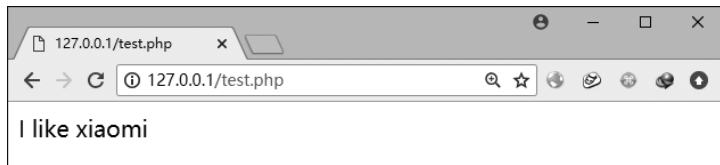


图 5-10 PHP array 示例

第二种方法：在 PHP 5.4 之后的版本中也可以使用方括号来构建。

```
$cp = ["apple", "xiaomi", "thinkpad"];
```

低版本 PHP 运行此代码会报如下错误：

```
Parse error: syntax error, unexpected '[' in C:\dev\PHPnow\htdocs\test.php on line 1
```

使用 count 函数可以获取数组长度,方便对数组进行遍历。例如：

```
$cp=array("thinkpad","xiaomi","apple");  
$cp[3]='aaa';  
$n=count($cp);  
for($i=0;$i<$n;$i++){  
    echo $cp[$i];  
    echo ",";  
}
```

运行结果如下：

```
thinkpad,xiaomi,apple,aaa,
```

可以看到,使用 array 函数创建了一个长度为 3 的数组,第二行使用赋值语句对 \$cp[3]进行了赋值,导致数组长度加 1,在下面的遍历中输出了 4 个元素。这种写法在 C 语言这类强类型语言中是不被允许的,会导致语法错误。

另外,PHP 中的数组还可以以字符串作为下标,这时数组更像哈希表(Hash Table)。符号=>的前面为键,后面为值。后面代码中\$age['Tom']的值为 35。

```
$age=array("Tom"=>"35","Ben"=>"37","Joe"=>"43");  
echo "Tom is " . $age['Tom'] . " years old.";
```

代码输出如下：

```
Tom is 35 years old.
```

另外,可以使用以下语法对数组进行遍历：