

PHP 程序设计

教程实验及课程设计

PHP CHENGXU SHEJI JIAOCHENG SHIYAN
JI KECHENG SHEJI

主 编 黎远松 曾静楣



西南交通大学出版社

PHP 程序设计教程

实验及课程设计

主 编 黎远松 曾静楣
副主编 何海涛 夏以波

西南交通大学出版社
· 成 都 ·

内容简介

本书以一种清晰而简练的风格介绍了 PHP 语言的语法和程序设计技术,并通过大量的示例演示了它们的正确使用方法和习惯用法。每一章在讲解了 PHP5 最基本的知识点后,都配有实例,带领读者走进 PHP5 殿堂。本书分 3 部分,第 1 部分是教程,从 Web 服务器基本原理开始,讲解了 PHP 应用开发环境安装与配置、PHP5 语言编程基础知识及 PHP5 面向对象知识,介绍了 PHP 相关应用,包括 PHP 操作文件系统、与表单交互等,还介绍了 PHP5 与数据库交互操作知识及 PHP5 一些高级应用。第 2 部分是实验,包括 7 个实验。第 3 部分是课程设计,由实例详细讲解 PHP5 在实际开发时的应用。本书适合广大 Web 网站开发人员、网站管理维护人员和大专院校学生作为参考书与教材使用,尤其适合有一定网络编程经验的开发人员使用。

图书在版编目(CIP)数据

PHP 程序设计教程实验及课程设计 / 黎远松, 曾静楣

主编. —成都: 西南交通大学出版社, 2017.6

ISBN 978-7-5643-5539-5

I. ①P… II. ①黎… ②曾… III. ①PHP 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2017) 第 151947 号

PHP 程序设计教程实验及课程设计

主 编 / 黎远松 曾静楣

责任编辑 / 穆 丰

封面设计 / 米迦设计工作室

西南交通大学出版社出版发行

(四川省成都市二环路北一段 111 号西南交通大学创新大厦 21 楼 610031)

发行部电话: 028-87600564

网址: <http://www.xnjdcbs.com>

印刷: 四川森林印务有限责任公司

成品尺寸 185 mm × 260 mm

印张 20.25 字数 491 千

版次 2017 年 6 月第 1 版 印次 2017 年 6 月第 1 次

书号 ISBN 978-7-5643-5539-5

定价 42.00 元

课件咨询电话: 028-87600533

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

前 言

最好的编程图书应该不是纯粹地讲述理论，而是要切合实际。在写书过程中，编者一直在努力让所写的内容能用到实处。如果你希望获得 PHP 编程语言的实践经验，对它们有全面的了解，并且想知道如何结合这些卓越的技术创建数据库驱动的动态 Web 应用程序，那么本书正合你所需。

如果你是初学者，强烈推荐从第 1 章开始阅读，因为首先要掌握 PHP 的基础知识，这对于阅读后面的章节很有好处。本书在编写时尽量合理地划分各章的内容，让读者能很快地了解各章的主题，而无需先掌握其他章节的内容。

另外，不论是新手还是经验丰富的 PHP 开发人员，都能从本书中获益。要理解本书介绍的概念，最有效的办法就是使用书中的代码亲自尝试。编写非常希望能收到读者的来信，并提出建议、意见和问题，可以随时给编者发电子邮件：lys700620@yeah.net。

本书获四川理工学院教材建设基金资助，四川省自贡市富顺县琵琶镇人民政府曾静楣同志参与实验和课程设计的编写工作，在此表示衷心的感谢。

编 者
2017 年 5 月

目 录

第 1 部分 教 程

1 基础教程.....	002
1.1 简 介.....	002
1.2 Wamp Server 安装.....	003
1.3 PHP 语法.....	004
1.4 变 量.....	006
1.5 echo 和 print 语句.....	010
1.6 数据类型.....	012
1.7 字符串函数.....	016
1.8 常 量.....	016
1.9 运算符.....	017
1.10 条件语句.....	022
1.11 Switch 语句.....	024
1.12 循环语句.....	025
1.13 函 数.....	028
1.14 数 组.....	030
1.15 数组排序.....	033
1.16 超全局变量.....	038
1.17 正则表达式.....	041
1.18 习 题.....	048
2 PHP 和表单.....	073
2.1 表单处理.....	073
2.2 PHP 表单验证.....	075
2.3 必填字段.....	080
2.4 验证名字、E-mail 和 URL.....	085
2.5 完整的表单实例.....	090
2.6 习 题.....	095
3 高级教程.....	098
3.1 多维数组.....	098

3.2	日期和时间	100
3.3	include 文件	103
3.4	文件处理	106
3.5	文件打开/读取/关闭	107
3.6	文件创建/写入	110
3.7	文件上传	111
3.8	Cookies	114
3.9	Sessions	116
3.10	发送电子邮件	117
3.11	安全的电子邮件	120
3.12	错误处理	123
3.13	异常处理	127
3.14	过滤器 (Filter)	133
3.15	习 题	137
4	PHP 和 MySQL 数据库	150
4.1	MySQL 简介	150
4.2	MySQL 连接数据库	150
4.3	创建数据库和表	151
4.4	insert into 语句	153
4.5	select 语句	155
4.6	where 子句	156
4.7	order by 子句	157
4.8	update 语句	157
4.9	delete from 语句	158
4.10	ODBC	158
4.11	习 题	161
5	PHP 和 XML	170
5.1	Expat 解析器	170
5.2	DOM 解析器	173
5.3	SimpleXML	175
5.4	XML 应用实例——留言本	176
5.5	习 题	179
6	PHP 和 AJAX	182
6.1	AJAX 简介	182
6.2	XMLHttpRequest 对象	183
6.3	AJAX 请求	184
6.4	AJAX XML 实例	187

6.5	AJAX MySQL 数据库实例	197
6.6	ResponseXML 实例	201
6.7	Live Search	206
6.8	RSS 阅读器	211
6.9	AJAX 投票	214
6.10	习 题	217

第 2 部分 实 验

实验 1	PHP 开发环境安装	222
实验 2	PHP 基础（一）	224
实验 3	PHP 基础（二）	227
实验 4	PHP 数据处理	232
实验 5	PHP Web 项目实践——编写 PHP 互动网页	244
实验 6	PHP 和数据库	251
实验 7	PHP 和 Ajax 技术	257

第 3 部分 课程设计

网络在线考试系统	262
参考文献	316

2.1 Wamp Server 安装

Wamp Server 是 Apache、MySQL 和 PHP 集成安装。Wamp 是 Windows、Apache、MySQL、PHP 的缩写。它提供了在 Windows 上安装和运行 Apache、MySQL 和 PHP 的简单方法。安装 Wamp Server 后，你可以立即使用它来开发网站。Wamp Server 的安装和配置非常简单，即使是初学者也可以轻松完成。

1.1.1 安装方法和步骤

第 1 部分

教程

1.1.1 安装方法和步骤

1. 下载 Wamp Server 安装包。你可以从 Wamp Server 官方网站（<http://www.wampserver.com/>）下载最新的安装包。Wamp Server 提供了 Windows、Linux 和 Mac OS X 三个版本。这里我们以 Windows 版本为例进行介绍。

2. 运行 Wamp Server 安装包。双击下载的 Wamp Server 安装包，启动安装向导。在安装向导中，你可以选择安装的路径和组件。默认情况下，Wamp Server 会安装在 C:\wamp 目录下。你可以选择其他路径，但需要注意路径中不能包含中文字符。在“组件”选项卡中，你可以选择要安装的组件。Wamp Server 提供了 Apache、MySQL 和 PHP 三个组件。你可以选择全部安装，也可以选择只安装部分组件。这里我们选择全部安装。

3. 配置 Wamp Server。安装完成后，Wamp Server 会自动启动。你可以看到 Wamp Server 的图标出现在任务栏上。你可以右键单击 Wamp Server 图标，打开 Wamp Server 的配置文件。在 Wamp Server 的配置文件（wamp.conf）中，你可以配置 Wamp Server 的各项参数。例如，你可以配置 Apache 的监听端口、MySQL 的数据库名和 PHP 的扩展模块等。配置完成后，你可以重启 Wamp Server 以使配置生效。

4. 测试 Wamp Server。安装和配置完成后，你可以测试 Wamp Server 是否正常工作。你可以打开浏览器，访问 <http://localhost/>。如果你看到 Wamp Server 的欢迎页面，说明 Wamp Server 已经安装成功并正常运行。

1.1.2 使用方法和步骤

1.1.2 使用方法和步骤

1. 创建新项目。在 Wamp Server 的图标上右键单击，选择“创建新项目”。在弹出的对话框中，你可以选择项目的名称和路径。默认情况下，Wamp Server 会将项目创建在 C:\wamp\www 目录下。你可以选择其他路径，但需要注意路径中不能包含中文字符。创建项目后，Wamp Server 会自动创建项目的配置文件（如 index.php、index.html 等）。

2. 运行项目。在 Wamp Server 的图标上右键单击，选择“运行项目”。Wamp Server 会自动启动 Apache 和 MySQL 服务，并打开浏览器访问 <http://localhost/>。你可以看到项目的欢迎页面。如果你看到项目的欢迎页面，说明项目已经创建成功并正常运行。

3. 配置项目。在 Wamp Server 的图标上右键单击，选择“配置项目”。在弹出的对话框中，你可以配置项目的各项参数。例如，你可以配置项目的监听端口、数据库名和扩展模块等。配置完成后，你可以重启 Wamp Server 以使配置生效。

4. 测试项目。安装和配置完成后，你可以测试项目是否正常工作。你可以打开浏览器，访问 <http://localhost/>。如果你看到项目的欢迎页面，说明项目已经安装成功并正常运行。

1.1.3 为什么使用 PHP

1.1.3 为什么使用 PHP

1. PHP 支持运行于多种平台（Windows、Linux、Mac OS X 等）。PHP 是一种跨平台的脚本语言，可以在多种操作系统上运行。这使得 PHP 具有广泛的适用性，可以满足不同用户的需求。

2. PHP 兼容几乎所有数据库（MySQL、MySQLi、PDO、SQLite 等）。PHP 提供了丰富的数据库接口，可以轻松地与各种数据库进行交互。这使得 PHP 具有强大的数据处理能力，可以满足不同用户的需求。

3. PHP 支持多种数据类型（字符串、整数、浮点数、布尔值、数组等）。PHP 提供了丰富的数据类型，可以满足不同用户的需求。这使得 PHP 具有强大的表达能力，可以满足不同用户的需求。

4. PHP 是免费的。PHP 是一种开源的脚本语言，任何人都可以自由地使用和修改。这使得 PHP 具有强大的生命力，可以满足不同用户的需求。

5. PHP 易于学习。PHP 的语法简单易懂，易于学习和使用。这使得 PHP 具有强大的吸引力，可以满足不同用户的需求。

1 基础教程

1.1 简介

PHP 是一种创建动态交互性站点的强有力的服务器端脚本语言,对于像微软 ASP 这样的竞争者来说,PHP 无疑是另一种高效的选择。

1.1.1 什么是 PHP

PHP 是“PHP Hypertext Preprocessor”的首字母缩略词,是一种被广泛使用的开源脚本语言,PHP 脚本在服务器上执行,可供免费下载和使用。PHP 是一门令人惊叹的流行语言,它强大到足以成为在网络上最大的博客系统的核心(WordPress),它深邃到足以运行最大的社交网络(facebook),而它的易用程度足以成为初学者的首选服务器端语言。

PHP 文件能够包含文本、HTML、CSS 以及 PHP 代码,在服务器上执行,而结果以纯文本返回浏览器,其后缀是“.php”。

1.1.2 PHP 能够做什么

- (1) PHP 能够生成动态页面内容;
- (2) 创建、打开、读取、写入、删除以及关闭服务器上的文件;
- (3) 接收表单数据;
- (4) 发送并取回 cookies;
- (5) 添加、删除、修改数据库中的数据;
- (6) 限制用户访问网站中的某些页面;
- (7) 对数据进行加密。

通过 PHP,你可以不受限于只输出 HTML,还能够输出图像、PDF 文件,甚至 Flash 影片,也可以输出任何文本,比如 XHTML 和 XML。

1.1.3 为什么使用 PHP

- (1) PHP 支持运行于各种平台(Windows, Linux, Unix, Mac OS X 等);
- (2) 兼容几乎所有服务器(Apache, IIS 等);
- (3) 支持多种数据库;
- (4) PHP 是免费的;
- (5) 易于学习,并可高效地运行在服务器端。

1.2 Wamp Server 安装

Wamp Server 是 Apache、MySQL 和 PHP 集成的开发环境，Wamp 集成了 PHP 以及 phpMyAdmin，集成了管理 MySQL 数据库的图形工具 SQLiteManager 和 phpMyAdmin 两种管理工具，既方便又快捷，可以说，能满足大多数 PHP 开发人员的需求。

1.2.1 安装方法和步骤

- (1) 下载 wamp5，假设文件名为：wamp5_1.7.4.exe；
- (2) 下载完毕，双击 wamp5_1.7.4.exe 文件；
- (3) 出现安装界面后，点击“Next”按钮；
- (4) 然后选中“I accept the agreement”选项，并点击“Next”按钮；
- (5) 选择默认安装路径，点击“Next”按钮；
- (6) 在弹出的对话框中，点击是“(Y)”按钮；
- (7) 点击“Next”按钮；
- (8) 点击“Next”按钮；
- (9) 点击“Install”按钮进行安装；
- (10) 点击“Next”按钮；
- (11) 点击“Next”按钮；
- (12) 点击是“(Y)”按钮；
- (13) 点击“finish”按钮，完成安装。

1.2.2 使用方法和步骤

- (1) 点击任务栏右下角图标；
- (2) 点击 www directory，打开 www 目录；
- (3) 在空白处右击，选择“新建文本文档”；
- (4) 修改文本文档文件名为 index.php（为了方便起见，默认主页）；
- (5) 双击 index.php，打开文件（打开方式，用记事本打开）；
- (6) 输入 index.php 源程序并保存：

```
<?php
echo "Hello, world!";
?>
```

点击 Localhost 运行，结果如图 1-1-1 所示。

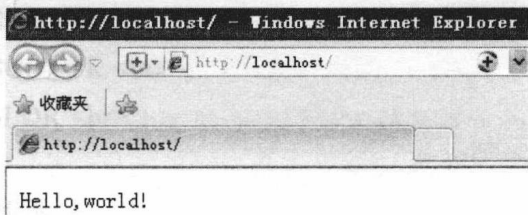


图 1-1-1 运行结果

1.3 PHP 语法

1.3.1 基础语法

PHP 脚本以 “<?php” 开头，以 “?>” 结尾。PHP 文件通常包含 HTML 标签以及一些 PHP 脚本代码。PHP 语句以分号 “;” 结尾，在 PHP 代码块的最后一行不必使用分号。PHP 默认文件扩展名是 “.php”。

下面的例子是一个简单的 PHP 文件，其中包含了使用内建 PHP 函数 “echo” 在网页上输出文本 “Hello World!” 的一段 PHP 脚本 (index.php):

```
<!DOCTYPE html>
<html>
<body>
<h1>我的第一张 PHP 页面</h1>
<?php
    echo "Hello World!";
?>
</body>
</html>
```

运行结果如图 1-1-2 所示。

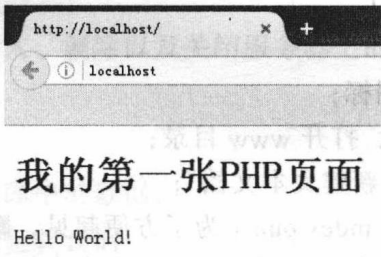


图 1-1-2 运行结果

1.3.2 PHP 中的注释

PHP 支持三种注释:

```
<?php
//这是单行注释
#这也是单行注释
/*
这是多行注释块
它横跨了
```

多行

```
*/
```

```
?>
```

注释用于：

(1) 使其他人理解你正在做的工作。注释可以让其他程序员了解你在每个步骤进行的工作；

(2) 提醒自己做过什么。大多数程序员都曾经历过一两年后对项目进行返工，然后不得不重新考虑他们做过的事情，注释可以记录你在写代码时的思路；

(3) 调试程序时，根据需要在注释掉不调试的程序。

通常在文件头注释文件名，修改时间，作者，电子邮件等信息：

```
/*
 * filename:  index.php
 * modify:    2012-05-22 14:15
 * author:    luker
 * e_mail:    service@pcboy.me
 */
```

1.3.3 大小写敏感问题

在 PHP 中，所有用户定义的函数、类和关键词（例如 if、else、echo 等）都对大小写不敏感。在下面的例子中，所有这三行 echo 语句都是合法的（等价的）：

```
<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>
```

不过，在 PHP 中，所有变量都对大小写敏感。在下面的例子中，只有第一条语句会显示 \$color 变量的值（这是因为 \$color、\$COLOR 以及 \$coLOR 被视作三个不同的变量）：

```
<?php
$color="红色的";
echo "我的汽车是$color <br>";
echo "我的房子是$COLOR <br>";
echo "我的船是$coLOR <br>";
?>
```

运行结果如图 1-1-3 所示。

我的汽车是红色的
我的房子是
我的船是

图 1-1-3 运行结果

\$color 的值为“红色的”，\$COLOR 和 \$coLOR 的值为 NULL。

1.4 变 量

变量是存储信息的容器。PHP 变量命名规则：

- (1) 变量以\$符号开头，其后是变量的名称。
- (2) 变量名称对大小写敏感（\$Submit 与 \$submit 是两个不同的变量）。
- (3) 变量名称必须以字母或下划线开头。
- (4) 变量名称不能以数字开头。
- (5) 变量名称只能包含字母数字字符和下划线（A~z、0~9 以及_）。

1.4.1 创建变量

PHP 没有创建变量的命令，变量会在首次为其赋值时被创建：

```
<?php
$txt="Hello world!";
$x=5;
$y=10.5;
?>
```

以上语句执行后，变量 txt 会保存值“Hello world!”，变量 x 会保存值 5，变量 y 会保存值 10.5。

PHP 是一门类型松散的语言，在上面的例子中，不必告知 PHP 变量的数据类型，PHP 根据它的值，自动把变量转换为正确的数据类型，在诸如 C，C++ 以及 Java 之类的语言中，程序员必须在使用变量之前声明它的名称和类型。

1.4.2 变量的作用域

在 PHP 中，可以在脚本的任意位置对变量进行声明，变量的作用域指的是变量能够被引用/使用的那部分脚本，PHP 有局部、全局和静态三种不同的变量作用域。

1. 局部和全局作用域

函数之外声明的变量拥有全局作用域，只能在函数以外进行访问，函数内部声明的变量拥有局部作用域，只能在函数内部进行访问，下面的例子测试了带有局部和全局作用域的变量：

```
<?php
    $x=5;//全局作用域
    function myTest()
    {
        $y=10;//局部作用域
        echo "<p>测试函数内部的变量: </p>";
        echo "变量 x 是: $x";
        echo "<br>";
        echo "变量 y 是: $y";
    }
    myTest();
    echo "<p>测试函数之外的变量: </p>";
    echo "变量 x 是: $x";
    echo "<br>";
    echo "变量 y 是: $y";
?>
```

运行结果如图 1-1-4 所示。

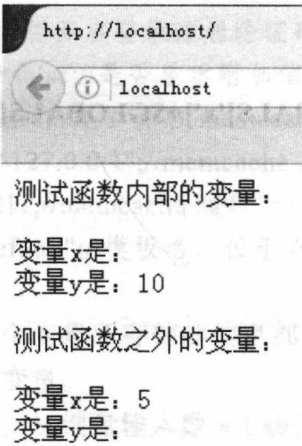


图 1-1-4 运行结果

在上例中，有两个变量\$*x* 和\$*y* 以及一个函数 *myTest()*，\$*x* 是全局变量，因为它是在函数之外声明的，而\$*y* 是局部变量，因为它是在函数内声明的。如果在 *myTest()*函数内部输出两个变量的值，\$*y* 会输出在本地声明的值，但是无法输出\$*x* 的值，因为它在函数之外创建，如果在 *myTest()*函数之外输出两个变量的值，那么会输出\$*x* 的值，但是不会输出\$*y* 的值，因为它是局部变量，在 *myTest()*内部创建。你可以在不同的函数中创建名称相同的局部变量，因为局部变量只能被在其中创建它的函数识别。

2. global 关键词

global 关键词用于访问函数外的全局变量。要做到这一点，请在（函数内部）变量前面使用 global 关键词。

```

<?php
    $x=5;
    $y=10;
    function myTest()
    {
        global $x, $y;
        $y=$x+$y;
    }
    myTest();
    echo $y; //输出 15
?>

```

PHP 同时在名为**\$GLOBALS** 的数组中存储了所有的全局变量，下标为变量名，这个数组在函数内也可以访问，并能够用于直接更新全局变量，上面的例子可以这样重写：

```

<?php
    $x=5;
    $y=10;
    function myTest()
    {
        $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
    }
    myTest();
    echo $y; //输出 15
?>

```

或者通过参数传递：

```

<?php
    $x=5;
    $y=10;
    function myTest($x, &$y)
    {
        $y=$x+$y;
    }
    myTest($x, $y);
    echo $y; //输出 15
?>

```

3. static 关键词

通常，当函数完成/执行后，会删除所有变量，如果要保留某个局部变量，声明变量时要使用 **static** 关键词。

```
<?php
```

```
@session_start();
ini_set("display_errors", "On");
static $DB_HOST="localhost";//数据库服务器
static $DB_NAME="acm";//数据名
static $DB_USER="root";//数据库用户
static $DB_PASS="root";//数据库密码
static $OJ_NAME="SUSE ACM";//系统名
static $OJ_HOME=".";//系统根目录
static $OJ_ADMIN="service@pcboy.me";//管理员邮箱
static $OJ_DATA="/home/judge/data";//测试数据在服务器上的位置
//static $OJ_LANG="en";//语言系统 cn->中文 en->英文
static $OJ_SIM=false;//相似度检查(此系统暂时没有用到)
static $OJ_DICT=false;//开启划词翻译插件(仅在中文模式下可用)
static $OJ_LANGMASK=1008;//支持编程语言 1->c 2->c++
static $OJ_EDITE_AREA=true;//是否开启代码高亮编辑框
static $OJ_AUTO_SHARE=false;//是否共享代码,开启后其他人可以查看用户提交的
```

的代码

```
static $OJ_VCODE=true;//是否开启登录注册验证码
static $OJ_APPENDCODE=false;//是否显示附加信息
static $OJ_MEMCACHE=false;//是否开启 memcache 缓存
static $OJ_MEMSERVER="127.0.0.1";//memcache 服务器 ip
static $OJ_MEMPORT=11211;//memcache 端口
static $OJ_TEMPLATE="default";//模板名,位于./template/下
```

```
?>
```

然后,每当函数被调用时,这个变量所存储的信息都是函数最后一次被调用时所包含的信息,但该变量仍然是函数的局部变量。

我们可以利用静态变量求阶乘。从键盘输入数 n (确保 $n > 0$),阶乘用函数实现,要求函数能“记忆”上一次阶乘的结果,即第一次调用时,函数计算的结果是 $1!$,第二次调用时,函数记住了上次函数调用的结果 ($1!$),在此基础上计算出 $2!$ 。

```
<?php
```

```
function fac($n)
{
    static $f=1;
    $f=$f*$n;
    return($f);
}
for($i=1;$i<=5;$i++)
    printf($i."!=".fac($i)."<br>");
```

```
?>
```

1.5 echo 和 print 语句

在 PHP 中，有两种基本的输出方法：`echo` 和 `print`。`echo` 能够输出一个以上的字符串，`print` 只能输出一个字符串，并始终返回 1。`echo` 比 `print` 执行稍快，因为它不返回任何值。

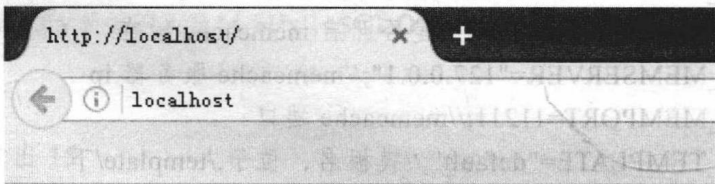
在本教程中，我们几乎在每个例子中都会用到 `echo` 和 `print`，因此，本节将讲解更多关于这两条输出语句的知识。

1.5.1 echo 语句

`echo` 是一个语言结构，有无括号均可使用，但 `echo()` 只能输出一个字符串。下面的例子展示如何用 `echo` 命令来显示不同的字符串（同时请注意字符串中能包含 HTML 标记）：

```
<?php
echo "<h2>PHP is fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This", " string", " was", " made", " with multiple parameters.";
//echo 能够输出一个以上的字符串，此时能用 echo，不能用 echo()。
?>
```

运行结果如图 1-1-5 所示。



PHP is fun!

Hello world!

I'm about to learn PHP!

This string was made with multiple parameters.

图 1-1-5 运行结果

下面的例子展示如何用 `echo` 命令来显示字符串和变量：

```
<?php
$txt1="Learn PHP";
$txt2="W3School.com.cn";
$scars=array("Volvo", "BMW", "SAAB");
echo $txt1, "<br>";
echo "Study PHP at $txt2 <br>";
echo "My car is a $scars[0]";
?>
```