



新起点

电脑教程

第 1 章

PHP 开发基础

本章要点

- ▣ 什么是 PHP
- ▣ 搭建 PHP 7 开发环境
- ▣ 使用 Dreamweaver CS
- ▣ 运行第一个 PHP 程序

本章主要内容

PHP 是一门优秀的网络编程语言，有着独立运行的环境，在学习 PHP 之前，首先需要明白什么是 PHP，它在怎样一个环境下运行。在本章的内容中，将向大家详细讲解学习 PHP 必须具备的基础知识，介绍自定义搭建 PHP 开发环境的过程，为大家步入本书后面知识的学习打下基础。



1.1 什么是 PHP



PHP 是 Hypertext Preprocessor(超文本预处理器)的缩写,是一种服务器端、跨平台、HTML 嵌入式的脚本语言。PHP 语言的语法结构比较独特,混合了 C 语言、Java 语言和 Perl 等编程语言的特点,是一种被广泛应用的开源式的多用途脚本语言,尤其适合 Web 开发。

↑ 扫码看视频

1.1.1 PHP 的地位

在动态 Web 开发世界,PHP 的占有率一直十分稳定,保持了三分之一份额的市场占有率。根据最新权威数据统计,全世界有超过 1.32 亿的网站和 12.5 万家公司在使用 PHP 语言,包括百度、雅虎、Google、YouTube、Digg 等著名网站,也包括汉莎航空电子订票系统、德意志银行的网上银行、华尔街在线的金融信息发布系统等,甚至军队系统这类苛刻的环境也选择使用 PHP 语言。除此之外,PHP 也是企业用来构建服务导向型和服务创新型混合于一体的、新一代的综合性商业所使用的语言,成为开源商业应用发展的方向。

1.1.2 PHP 的特点

PHP 语言具有以下特点。

- 快速:这是最突出的特点,PHP 是一种强大的 CGI 脚本语言,语法混合了 C、Java、Perl 和 PHP 式的新语法,执行网页的速度比 CGI、Perl 和 ASP 等语言更快。
- 开放性和可扩展性:PHP 是自由软件,其源代码完全公开,任何程序员都可以非常容易地为 PHP 扩展附加功能。
- 数据库支持功能强大:PHP 支持多种主流与非主流的数据库,如 MySQL、Microsoft SQL Server、Solid、Oracle 和 PostgreSQL 等。其中 PHP 与 MySQL 是绝佳组合,可以实现跨平台运行。
- 功能丰富:从对象式的设计、结构化的特性、数据库的处理、网络接口应用到安全编码机制等,PHP 几乎涵盖了所有网站的一切功能。
- 易学好用:只需要了解一些基本的语法和语言特色,就可以开始你的 PHP 编码之旅。如果在编码的过程中遇到了什么麻烦,可以去翻阅相关文档,如同查找词典一样,十分方便。
- 学习速度快:只需 30 分钟就可以熟练掌握 PHP 的核心语言,PHP 代码通常嵌入 HTML 中,在设计和维护站点的同时,可以轻松地加入 PHP,使站点更加具有动态特性。

- 功能全面：PHP 包括图形处理、编码与解码、压缩文件处理、XML 解析、支持 HTTP 的身份认证、Cookie、POP3、SNMP 等。可以利用 PHP 连接包括 Oracle、MS-Access、MySQL 在内的大部分数据库。

1.2 搭建 PHP 开发环境



截至目前，PHP 的最流行版本是 PHP 7。在接下来的内容中，将首先详细讲解搭建 PHP 开发环境的基本知识，为读者步入本书后面知识的学习打下基础。

↑ 扫码看视频

1.2.1 使用 AppServ 组合包

组合包，就是将 Apache、PHP、MySQL 等服务器软件和工具安装配置完成后打包处理。开发人员只要将已配置的套件解压到本地硬盘中即可使用，无须再另行配置。组合包实现了 PHP 开发环境的快速搭建。对于刚开始学习 PHP 语言的读者来说，建议采用组合包方法搭建 PHP 的开发环境。虽然组合包在灵活性上要差很多，但是具备安装简单、安装速度较快和运行稳定的优点，所以比较适合初学者使用。

目前网上流行的组合包有十几种，安装步骤基本上大同小异。其中比较常用的组合包有 EasyPHP、AppServ 和 XAMPP。笔者在此建议新手使用 EasyPHP 或 AppServ，这两个组合包都对 Apache+MySQL+PHP 开发环境进行了集成。几个组合包的下载地址如下。

- EasyPHP: <http://www.easyphp.org/>。
- AppServ: <http://www.AppServnetwork.com/>。
- XAMPP: <http://www.Apachefriends.org/>。



智慧锦囊

要想使用 PHP 开发语言，必须先搭建 PHP 开发环境，可以通过如下步骤来完成开发环境的搭建工作。

- (1) 安装 Apache 服务器：用户要想运行 PHP 程序，必须在电脑上安装 Apache 服务器软件。
- (2) 安装 PHP：最流行版本是 PHP 7。
- (3) 安装 MySQL：MySQL 是数据库工具，是 PHP 语言的黄金伙伴。
- (4) 安装 PHPMyAdmin：PHPMyAdmin 是一款管理 MySQL 数据库的软件，功能很强大，能完成其他 MySQL 管理软件都能完成的功能。



1.2.2 搭建 AppServ 开发环境

在 Windows 10(64 位)环境中, 搭建 AppServ 开发环境的具体流程如下所示。

第 1 步 登录 AppServ 的官方下载地址: <https://www.appserv.org/en/download/>, 其页面如图 1-1 所示。



图 1-1 AppServ 官方下载页面

第 2 步 单击下方红色的 DOWNLOAD 按钮开始下载, 下载后将得到一个 .exe 文件。(用鼠标)右击这个文件, 在弹出的快捷菜单中选择“以管理员身份运行”命令, 在弹出的安装向导的欢迎界面中单击 Next 按钮, 如图 1-2 所示。

第 3 步 在弹出的如图 1-3 所示的界面中单击 I Agree 按钮, 同意安装协议。



图 1-2 欢迎安装界面

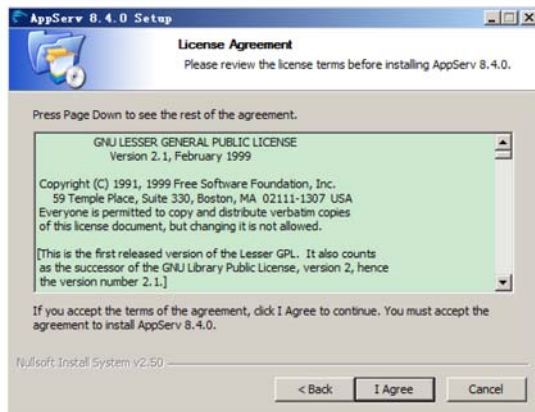


图 1-3 许可协议界面

第 4 步 在弹出的如图 1-4 所示的选择安装位置界面中设置程序的安装路径, 建议不要安装在系统盘 C 上, 然后单击 Next 按钮。

第 5 步 在弹出的如图 1-5 所示的选择组件界面中显示四个复选框, 分别代表要安装的功能。在此全部选中这 4 个复选框, 然后单击 Next 按钮。

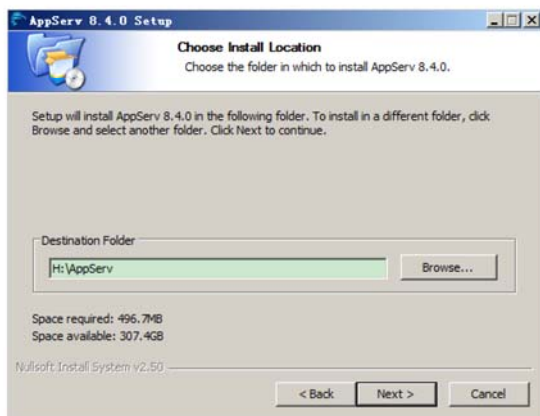


图 1-4 安装路径设置界面



图 1-5 安装功能选择界面

第 6 步 在弹出的如图 1-6 所示的界面中设置网站信息，在第一个文本框中输入自己网站的域名，在第二个文本框中输入网站管理员的邮箱地址，在第三个文本框中输入想使用的端口，默认是 80。在此只需使用图 1-6 中的默认设置即可，然后单击 **Next** 按钮。

第 7 步 在弹出的如图 1-7 所示的数据库服务器配置界面中设置网站数据库的密码及使用的编码。在第一个文本框中输入想设置的数据库的密码(密码至少 8 位)，在第二个文本框中输入刚才输入的密码进行确认。然后选择设置一个编码，默认为 UTF-8。最后单击 **Install** 按钮。

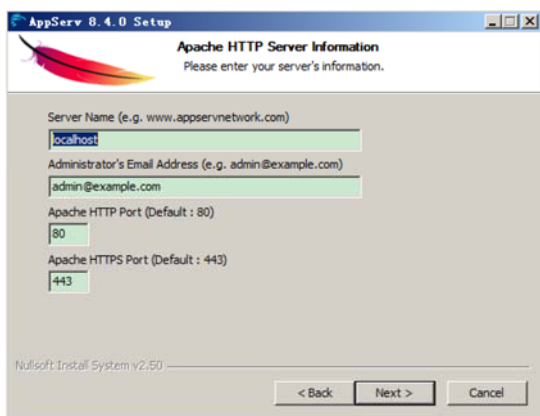


图 1-6 网站信息设置界面



图 1-7 密码及编码设置界面



注意：这里设置的是组合包中安装的 MySQL 数据库的密码。在设置时必须牢记设置的密码，因为以后在程序中连接数据库时要使用到。建议读者将密码设置为 66688888，因为这是本书所使用的数据库密码。

第 8 步 在弹出的如图 1-8 所示的界面中显示安装进度。

第 9 步 进度完成后弹出如图 1-9 所示的界面，单击 **Finish** 按钮完成安装。

第 10 步 在安装好 AppServ 之后，整个目录默认安装在“C:\AppServ”路径下，笔者的安装目录是“H:\AppServ”，在此目录下包含 5 个子目录，如图 1-10 所示，读者可以将



所有 PHP 网页文件存放到“ www” 目录下进行调试。



图 1-8 安装进度界面



图 1-9 安装完成界面

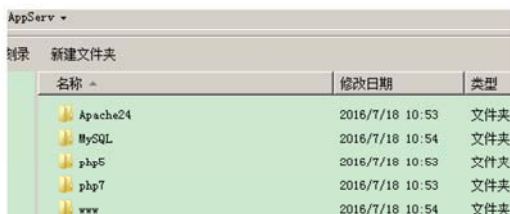


图 1-10 AppServ 的安装目录

- Apache24: Apache 的存储目录。
- MySQL: MySQL 数据库的存储目录。
- php5: PHP 5 的存储目录。
- php7: PHP 7 的存储目录。
- www: PHP 网页文件和 phpMyAdmin 的存储目录。

第 11 步 在浏览器中输入“ <http://127.0.0.1:8080/>” 或“ <http://localhost:8080/>” 会显示如图 1-11 所示的界面，这说明 AppServ 组合包已经安装完成。在此既可设置以 PHP 5 运行程序，也可以设置使用 PHP 7 运行程序。



图 1-11 AppServ 组合包安装完成

1.2.3 选择 PHP 7 运行环境

因为 AppServ 组合包同时集成了 PHP 5 和 PHP 7，并且安装完成后会默认运行 PHP 5 环境，具体如图 1-11 所示。本书讲解的是 PHP 7，所以需要设置 PHP 7 环境。具体方法是：单击“开始”按钮，选择“所有应用”→AppServ 命令，在其子菜单中选择 PHP Version Switch 命令，如图 1-12 所示。然后在打开的如图 1-13 所示的界面中输入数字“7”，并按 Enter 键，系统将自动设置为 PHP 7 环境。

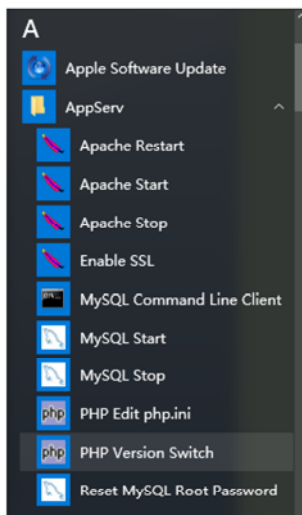


图 1-12 选择 PHP Version Switch 命令

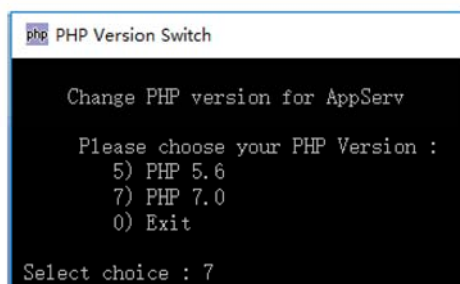


图 1-13 PHP Version Switch 界面

设置完成后，在浏览器中输入“<http://127.0.0.1:8080/>”或“<http://localhost:8080/>”后会显示 PHP 7 环境界面，具体如图 1-14 所示。



图 1-14 运行 PHP 7 环境



知识精讲

问：听说 HHVM 是开发 PHP 的一款利器，其效率真有大家说的那么牛吗？

答：HHVM 是脸谱公司为提高 PHP 性能而开发出来的工具，使用了 Just-In-Time (JIT)编译方式将 PHP 代码转换成某种字节码。在实际测试过程中，HHVM 对于 PHP 的性能提高是一个质的飞跃，高效的 PHP 运行环境能提升 PHP 性能 9 倍以上。在 PHP 7 正式发布后，在 PHP 性能方面得到了非常大的改善，实际测试发现，在部分场合 PHP 7 性能超过了 HHVM。

1.3 使用 Dreamweaver



在当今市面中，Dreamweaver 是最为常用的网页设计工具之一，也是 PHP 开发人员的最佳开发工具之一。在本书后面的内容中，将使用 Adobe Dreamweaver CS6 作为开发工具进行讲解。

↑ 扫码看视频

1.3.1 安装 Dreamweaver

在获取 Dreamweaver CS6 安装包后，接下来按照如下所示的步骤进行安装。

第 1 步 下载完安装文件后双击安装图标，弹出解压缩界面，在此选择一个保存解压缩安装文件的路径，如图 1-15 所示。

第 2 步 单击“下一步”按钮，弹出“正在准备文件”界面，如图 1-16 所示。

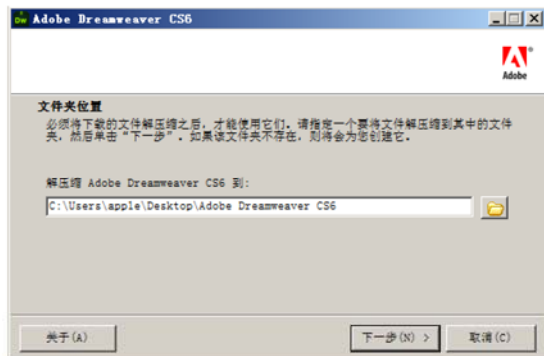


图 1-15 选择一个保存解压缩安装文件的路径

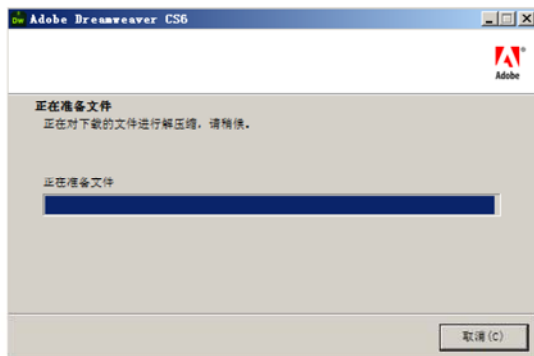


图 1-16 “正在准备文件”界面

第 3 步 进度完成后弹出“欢迎”界面，在此选择“安装”选项，如图 1-17 所示。

第 4 步 弹出“Adobe 软件许可协议”界面，在此单击“接受”按钮，如图 1-18 所示。



图 1-17 “欢迎”界面



图 1-18 “Adobe 软件许可协议”界面

第5步 弹出“序列号”界面，在此输入合法的序列号，并单击“下一步”按钮，如图 1-19 所示。

第6步 在弹出的“选项”界面中设置安装目录，然后单击“安装”按钮，如图 1-20 所示。

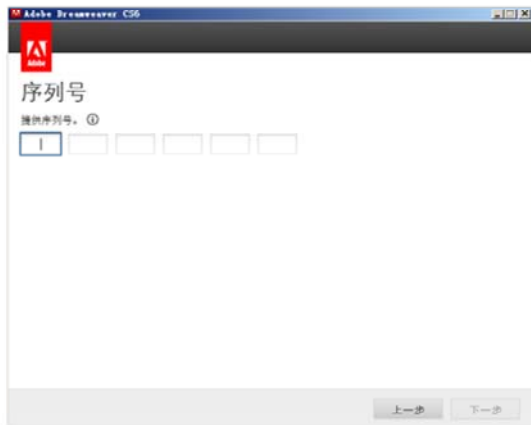


图 1-19 “序列号”界面



图 1-20 “选项”界面

第7步 在弹出的“安装”界面中显示安装进度条，如图 1-21 所示。

第8步 进度完成后弹出“安装完成”界面，如图 1-22 所示。

第9步 当第一次打开 Dreamweaver CS6 的时候，会弹出“默认编辑器”对话框，在此可以设置我们常用的文件类型，如图 1-23 所示。

第10步 单击图 1-23 中的“确定”按钮后进入启动界面，如图 1-24 所示。

第11步 启动 Dreamweaver CS6 后的界面效果如图 1-25 所示，在此可以设置新建页面的类型。



注意：在安装 Dreamweaver CS6 之前，一定要确保在本地机器上没有安装过 Dreamweaver。如果已经安装过其他的版本，请务必确保卸载干净，否则会安装失败。



图 1-21 “安装”界面



图 1-22 “安装完成”界面



图 1-23 “默认编辑器”对话框

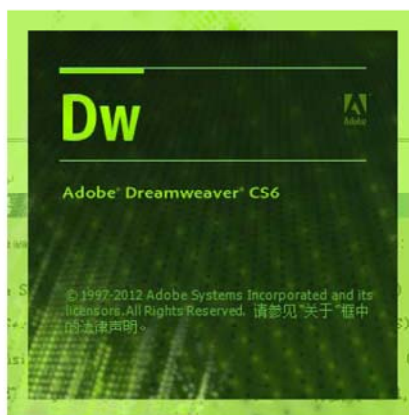


图 1-24 启动界面

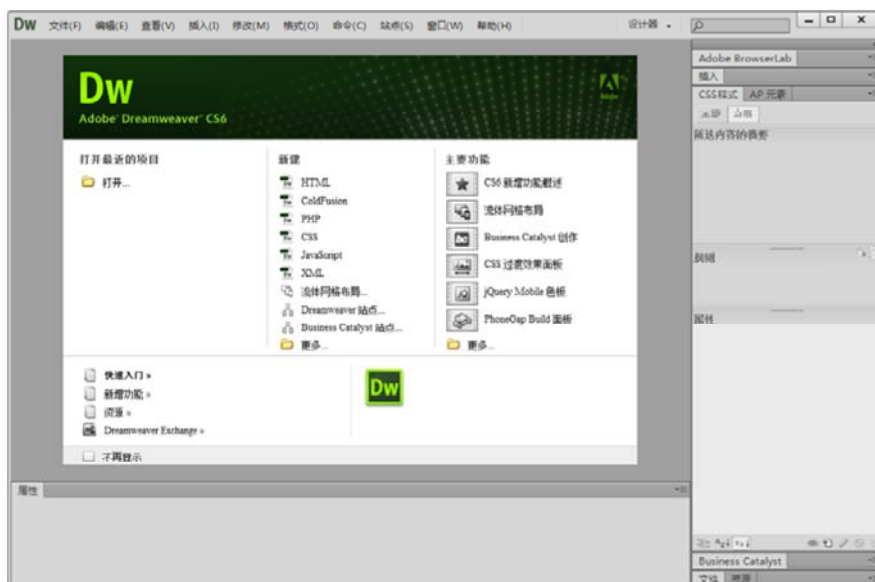


图 1-25 启动 Dreamweaver CS6 后的界面



1.3.2 使用 Dreamweaver 建立 PHP 站点

第 1 步 打开 Dreamweaver CS6，在菜单栏中选择“站点”→“管理站点”命令，打开如图 1-26 所示的“管理站点”对话框。

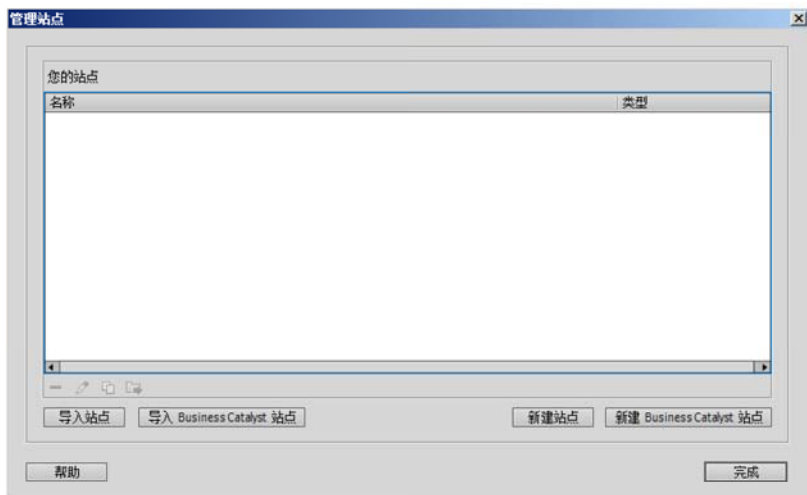


图 1-26 “管理站点”对话框

第 2 步 单击“管理站点”对话框中的“新建站点”按钮，在弹出的“站点设置对象”对话框中设置“站点名称”为“PHPBook”，设置“本地站点文件夹”为“H:\AppServ\www\book\”，如图 1-27 所示。



图 1-27 “站点设置对象”对话框

这样即成功将本地文件夹“H:\AppServ\www\book\”设置为一个站点。



1.4 实践案例与上机指导



通过对本章内容的学习，读者基本可以掌握搭建 PHP 语言开发环境和使用开发工具 Dreamweaver 的知识。其实搭建 PHP 语言开发环境的知识还有很多，这需要读者通过课外渠道来加深学习。下面通过练习操作，以达到巩固学习、拓展提高的目的。

↑ 扫码看视频

1.4.1 运行第一个 PHP 程序

PHP 开发环境搭建完毕后，下面开始运行第一个 PHP 程序，测试一下我们的开发环境是否搭建成功。



实例 1-1：第一个 PHP 程序

源文件路径：daima\1\1-1

实例文件 index.php 的主要实现代码如下：

```
<body>
<?php
    echo "欢迎进入 PHP 的世界！！";
?>
</body>
```

在上述 PHP 代码中，“echo”是 PHP 语言的输出语句，能够将后面的字符串或值显示在页面中，每行 PHP 代码都以分号“;”结束。将上述代码保存在“AppServ\www\daima\1\1-1\”目录下，命名为“index.php”。在浏览器中输入“http://127.0.0.1:8080/daima/1/1-1/index.php”进行测试，执行效果如图 1-28 所示。这说明我们的 PHP 程序已经正确运行了，标志着我们的 PHP 开发环境搭建成功。



图 1-28 执行效果

1.4.2 输出显示当前时间

通过使用 PHP 程序，可以输出显示服务器的当前时间。



实例 1-2：输出显示当前时间

源文件路径：daima\1\1-2

实例文件 index.php 的主要实现代码如下：

```
<body style="font-family:'华文彩云'; color:#0000CC; font-size:16px">
系统的当前时间是:
<?php
    date_default_timezone_set('Asia/ShangHai');
    echo date('Y-m-d H:i:s');
?>
</body>
```

将上述代码保存在“AppServ\www\daima\1\1-2\”目录下，命名为“index.php”。在浏览器中输入“http://127.0.0.1:8080/daima/1/1-2/index.php”进行测试，执行效果如图 1-29 所示。

系统的当前时间是：2019-01-25 11:52:51

图 1-29 执行效果

1.5 思考与练习

本章首先介绍了什么是 PHP，然后详细阐述了搭建 PHP 7 开发环境的知识，最后讲解了使用 Dreamweaver CS 并调试运行第一个 PHP 程序的知识。通过对本章内容的学习，读者应该初步了解 PHP 语言的基础知识，掌握其使用方法和技巧。

1. 选择题

- (1) () 是 PHP 初学者的最佳开发工具。
A. Notepad B. Editplus C. Dreamweaver
- (2) 被认为和 PHP 是最佳拍档的数据库工具是()。
A. MySQL B. SQL Server C. Oracle D. Access

2. 判断对错

- (1) “TIOBE 编程语言社区排行榜”只是反映某个编程语言的热门程度，并不能说明一门编程语言好不好，或者一门语言所编写的代码数量多少。这个排行榜可以考查大家的编程技能是否与时俱进，也可以在开发新系统时作为一个语言选择依据。 ()
- (2) MySQL 是一个小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司。在 2008 年 1 月 16 日被 SUN 公司收购。而 2009 年，SUN 又被 Oracle 公司收购。目前 MySQL 被广泛地应用在 Internet 上的中小型网站中。 ()

3. 上机练习

- (1) 下载并安装 Editplus，并尝试调试本章中的实例 1-1。
- (2) 下载并安装 Notepad，并尝试调试本章中的实例 1-1。



新起点

第 2 章

电脑教程

PHP 语言的基础语法

本章要点

- 语法结构
- 页面注释
- 变量
- 常量
- 数据类型
- 运算符和表达式

本章主要内容

无论是初出茅庐的“菜鸟”，还是资历深厚的“高手”，没有扎实的基础做后盾是不行的。在本章的内容中，将向大家详细讲解 PHP 语言基本语法的知识。只要掌握了这些基础语法，就等于有了坚固的地基，由此可以“万丈高楼平地起”，为步入本书后面知识的学习打下基础。



2.1 语 法 结 构



在 PHP Web 项目中,PHP 代码是嵌套在 HTML 程序中的,本节将通过语法结构的知识讲解哪些是 HTML 语句,哪些是 PHP 语句,一个完整的 PHP 语句的构成元素等知识。

↑ 扫码看视频

2.1.1 PHP 文件构成

PHP 文件实际上是一个十分简单的文本文件,用户可以使用任何文本工具对它进行编写,如记事本、Dreamweaver 等工具,然后将其保存为.php 文件。当编辑好 PHP 文件后,开发者只需要将文件复制到本书第 1 章所配置的环境目录中即可运行,然后就可以通过浏览器浏览运行后的 PHP 文件。一个 PHP 文件通常由以下元素构成。

- HTML 代码。
- PHP 标记。
- PHP 代码。
- 注释。
- 空格。
- 其他元素。



实例 2-1: 在网页中显示文字

源文件路径: daima\2\2-1

本实例的功能是在浏览器中显示“欢迎进入 PHP 的世界!!”,主要实现代码如下:

```
<body>
<?php
    echo "欢迎进入 PHP 的世界!!";
?>
</body>
```

将上述代码文件保存到服务器环境下,运行后得到如图 2-1 所示的结果。



图 2-1 在网页中显示文字



智慧锦囊

经过上面的实例学习，读者可能会有一个疑问：通过第 1 章的学习，也可以制作出同样的效果，何必这样麻烦？实际上两者之间是有很大的不同的，用 HTML 语言写出同样功能的语句，只能算是静态网页，双击打开后也会得到同样的效果。两个实例不同的是，前者是显示出来的，而后者是输出出来的，它只能在符合 PHP 运行条件时才能得到上面的效果图，双击是无法打开看到效果的，这就是两者之间的区别。

2.1.2 PHP 标记

通常 PHP 代码被嵌套在 HTML(超文本标记语言，标准通用标记语言下的一个应用。“超文本”就是指页面内可以包含图片、链接，甚至音乐、程序等非文字元素)代码中，在 HTML 中有多种元素，例如文本、表单和按钮等。PHP 用什么标记和各种 HTML 标记进行区分呢？在通常情况下，可以用以下几种标记来标识 PHP 代码：

- `<?php...?>`
- `<?...?>`
- `<script language="php">...</script>`
- `<%...%>`

在大多数情况下，开发者会使用第一种标记来声明这部分是 PHP 代码。`<?...?>`是 XML 标记，有时可能会和 XML 冲突，`<script language="php">...</script>`是脚本语言的标记，`<%...%>`是 ASP 的语法风格。所以在此建议读者使用`<?php...?>`，这是 PHP 语言的标准标记。例如在下面的实例中，使用 4 种标记方式在页面中显示了一段文字。

**实例 2-2：**使用 4 种标记方式显示一段文字

源文件路径：daima\2\2-2

实例文件 `index.php` 的主要实现代码如下：

```
<?php echo("第一种书写方法!\n");
?>
<script language="php">
    echo("script 书写方法!");
</script>
<%
    echo("这是 ASP 的标记输出");
%>
<?
    echo("这是 PHP 的简写标记输出");
?>
```

将上述代码文件保存到服务器的环境下，浏览运行后的效果如图 2-2 所示。

第一种书写方法! script书写方法!<% echo("这是ASP的标记输出"); %> 这是PHP的简写标记输出

图 2-2 PHP 的 4 种标记方式



智慧锦囊

在运行本实例时，有的读者的电脑可能不能正确显示 ASP 方式的代码，并没有输出任何东西。这是因为用户没有对 `php.ini` 进行设置，如果用户要用 ASP 分隔符风格编写 PHP，必须修改 `php.ini` 文件的内容，将 `php.ini` 的如下文本：

```
;Allow ASP-style<% %> tags.  
Asp_tags=off
```

修改为：

```
;Allow ASP-style<% %> tags.  
Asp_tags=ON
```

然后重新启动服务器软件，即可得到前面正确的运行结果。

2.2 页面注释



注释是每一种编程语言都离不开的元素，它是对代码的解释和说明。JSP、ASP.NET 和 PHP 等程序都离不开注释，注释能帮助开发者进行后期维护。良好的代码注释对后期维护、升级能够起到非常重要的作用。PHP 是一种优秀的网络程序设计语言，其注释风格和经典程序设计语言 C 大致相同。

↑ 扫码看视频

到目前，PHP 语言支持以下 3 种注释方式。

- `//`：C++风格的单行注释，单行注释以“`//`”开始，到该行结束或者 PHP 标记结束之前的内容都是注释。
- `#`：Shell 脚本风格的注释，Shell 脚本注释以“`#`”开始，到该行结束或者 PHP 标记结束之前的内容都是注释。
- `/*`和`*/`：C 风格的多行注释。

读者可以根据自己的喜好和习惯来选择一种方式，例如在下面的实例中使用了上述三种注释风格。



实例 2-3：使用注释

源文件路径：daima\2\2-3

实例文件 `index.php` 的主要实现代码如下：

```
<?php  
echo "我是 C++语言注释的方法 // <br>"; // 采用 C++的注释方法  
/* 多行注释
```

```
对于大段的注释很有用的哦 */
echo "我是C语言注释的方法,对多行注释十分有用 /*...*/ <br>";
echo "我是Unix的注释方法 # <br>"; # 使用 UNIX Shell 语法注释
?>
```

将上述代码文件保存到服务器环境下,运行浏览后得到如图 2-3 所示的结果。

```
我是C++语言注释的方法 //
我是C语言注释的方法,对多行注释十分有用 /*...*/
我是Unix的注释方法 #
```

图 2-3 几种注释方法

2.3 变 量



在 PHP 语言程序中,变量是指在程序执行过程中数值可以变化的量,通过一个名字(变量名)来标识。PHP 的变量与其他语言有所不同,例如在 PHP 中使用变量之前不需要声明,只需为变量赋值后即可使用。

↑ 扫码看视频

2.3.1 变量的定义

PHP 中的变量名称用“\$”和标识符表示,变量名是区分大小写的。变量赋值,是指给变量一个具体的数据值,对于字符串和数字类型的变量,可以通过“:”来实现赋值。

变量赋值是指赋予变量具体的数据,自从 PHP 4.0 开始,PHP 不但可以对变量赋值,还可以对变量赋予一个变量地址,即引用赋值。定义并赋值 PHP 变量的语法格式如下:

```
<?php $name = value; ?>
```

请看下面的演示代码,运行后会输出两次“My name is Bob”,因为在第 3 行代码中变量\$bar 通过引用赋值得到了变量\$foo 的内存地址,所以当在第 4 行改变\$bar 的值时,\$foo 的值也会发生变化。

```
<?php
$foo = 'Bob';           //变量$foo 赋值为 Bob
$bar = &$foo;           //变量$bar 赋值得到了变量$foo 的内存地址
$bar = "My name is $bar"; //改变变量$bar 的值
echo $bar;              //输出变量$bar 的值
echo $foo;              //输出变量$foo 的值
?>
```

PHP 语言规定:变量名用“\$”和标识符表示,并且需要遵循以下规则。

- 在 PHP 中的变量名是区分大小写的。



- 变量名必须以美元符号“\$”开始。
- 变量名开头可以以下划线开始。
- 变量名不能以数字字符开头。
- 变量名可以是中文。
- 变量名可以包含一些扩展字符(如重音拉丁字母)，但不能包含非法扩展字符(如汉字字符和汉字字母)。

2.3.2 变量的作用域

在 PHP 程序中使用变量时一定要符合使用变量的规则，变量必须在有效范围内使用，如果变量超出有效范围，变量也就失去其意义。PHP 变量有以下三种使用范围。

(1) 局部变量：即在函数的内部定义的变量，其作用域是所在函数。

(2) 全局变量：即被定义在所有函数以外的变量，其作用域是整个 PHP 文件，但是在用户自定义函数内部是不可用的。要想在用户自定义函数内部使用全局变量，需要使用 `global` 关键字进行声明。

(3) 超级变量：在任何位置都可用的特定数量的变量，并且可以从脚本的任何位置访问它们。

请看下面的演示代码，考虑为什么下面的代码没有输出任何结果。

```
<?php
$a = 1;           //定义变量 a，并赋值为 1
function Test()   //定义函数 Test()
{
    echo $a;       //输出变量 a 的值
}
Test();           //执行函数 Test()
?>
```

这是因为 `echo` 语句引用了一个局部版本的变量“\$a”，而且在这个范围内它并没有被赋值。读者可能注意到 PHP 的全局变量和 C 语言有一点点不同。因为在 C 语言中，全局变量在函数中自动生效，除非被局部变量覆盖。这可能引起一些问题，有些人可能会漫不经心地改变一个全局变量。PHP 中的全局变量在函数中使用时必须声明为全局变量。

2.3.3 可变变量

在 PHP 程序中，可变变量是一种独特的变量，允许动态地改变一个变量名称。可变变量的工作原理是，该变量的名称由另外一个变量的值来确定，一个普通的变量通过声明来设置，一个可变变量获取一个普通变量的值作为这个可变变量的变量名。例如在下面的代码中，“hello”使用了两个美元符号“\$”以后，就可以作为一个可变变量的变量了。

```
<?php
$a = 'hello';
?>
```

紧接着上述代码，在下面的代码中，这时两个变量都被定义了，其中 \$a 的内容是

“hello”，并且 \$hello 的内容是“world”。

```
<?php
$$a = 'world';
?>
```



智慧锦囊

在 PHP 程序中规定，变量不可以与已有的变量重名，否则将引起冲突。在给变量命名的时候，最好让变量有一定的含义，因为这样有利于阅读代码，同时也有利于对变量名的引用。

2.4 常 量



在 PHP 程序中，常量是指其值在程序的运行过程中不发生变化的量，常量值被定义后，在脚本的其他任何地方都不能改变。在本节的内容中，将详细讲解 PHP 常量的基本知识和用法。

↑ 扫码看视频

2.4.1 定义并使用常量

在 PHP 程序中，定义常量的语法格式如下：

```
bool define ( string name, mixed value [, bool case_insensitive] ) ;
```

上述函数 `define` 有如下所示的 3 个参数。

- 第一个参数为常量名称，即标识符。
- 第二个参数为常量的值。
- 第三个参数指定是否大小写敏感，设定为 `true`，表示不敏感。

在 PHP 程序中，可以通过指定其名字来取得常量的值，切记不要在常量前面加“\$”符号。如果要在程序中动态获取常量值，可以使用 `constant()` 函数，该函数要求一个字符串作为参数，并返回该参数的值。如果要判断一个常量是否已经定义，可以使用 `defined()` 函数，该函数也需要一个字符串参数，该参数为需要检测的常量名称，若该常量已经定义，则返回 `true`；如果想获取所有当前已经定义的常数列表，可以使用 `get_defined_constants()` 函数来实现。请看下面的实例，演示了定义并使用常量的过程。



实例 2-4：定义并使用常量

源文件路径：daima\2\2-4



实例文件 index.php 的主要实现代码如下：

```
<?php
define ("MESSAGE","能看到一次");           //设置常量 MESSAGE 的值
echo MESSAGE."<BR>";                          //输出常量 MESSAGE
echo Message."<BR>";                          //输出 "Message"，表示没有该常量
define ("COUNT","能看到多次",true);
echo COUNT."<BR>";                            //输出常量 COUNT
echo Count."<BR>";                            //输出常量 COUNT，因为设定大小写不敏感
$name = "count";
echo constant ($name)."<BR>";                //输出常量 COUNT
echo (defined ("MESSAGE"))."<BR>";          //如果定义返回 True，使用 echo 输出显示 1
?>
```

在上述代码中使用了 define()、constant() 和 defined() 共 3 个函数。其中使用 define() 函数来定义一个常量，使用 constant() 函数来动态获取常量的值，使用 defined() 函数来判断常量是否被定义。执行效果如图 2-4 所示。

能看到一次
Message
能看到多次
能看到多次
能看到多次
1

2.4.2 预定义常量

图 2-4 实例 2-4 的执行效果

在 PHP 程序中，可以使用预定义常量获取 PHP 中的信息，常用的预定义常量如表 2-1 所示。

表 2-1 PHP 常用的预定义常量

常 量 名	功 能
FILE	默认常量，PHP 程序文件名
LINE	默认常量，PHP 程序行数
PHP VERSION	内置常量，PHP 程序的版本，如 3.0.8_dev
PHP_OS	内置常量，执行 PHP 解析器的操作系统名称，如 Windows
TRUE	该常量是一个真值(true)
FALSE	该常量是一个假值(false)
NULL	一个 null 值
E_ERROR	该常量指到最近的错误处
E_WARNING	该常量指到最近的警告处
E_PARSE	该常量指到解析语法有潜在问题处
E_NOTICE	该常量为发生不寻常处的提示，但不一定是错误处



实例 2-5：定义并使用预定义常量

源文件路径：daima\2\2-5

实例文件 index.php 的主要实现代码如下：

```
<?php
echo "当前文件路径： ".__FILE__;           //显示当前文件路径
```

```
echo "<br>当前行数: " . __LINE__ ; //显示当前文件的行数
echo "<br> 当前操作系统: " . PHP_OS ; //显示当前的操作系统
?>
```

本实例的执行效果如图 2-5 所示。

```
当前文件路径: H:\AppServ\www\book\2\2-8\index.php
当前行数: 8
当前操作系统: WINNT
```

图 2-5 实例 2-5 的执行效果

2.5 数据类型



无论是变量、常量，还是在以后要学习的数组，都有属于自己的数据类型。在 PHP 程序中支持 8 种数据类型，这 8 种数据类型又可以分为三类，分别是简单类型、复合类型和特殊类型，下面对它们进行详细讲解。

↑ 扫码看视频

2.5.1 布尔型

布尔变量是 PHP 中最简单的，它保存了一个 true 或者 false 值。其中 true 或者 false 是 PHP 的内部关键字。设定一个布尔型的变量后，只需将 true 或者 false 赋值给该变量即可，并不区分大小写。



实例 2-6：使用布尔型

源文件路径: daima\2\2-6

实例文件 index.php 的主要实现代码如下：

```
<?php
$boo = true; //变量 boo 的初始值为 true
if($boo == true) //如果变量 boo 的值为 true
    echo '变量$boo 为真!'; //输出对应的内容
else //如果变量 boo 的值不为 true
    echo '变量$boo 为假!!'; //输出对应的内容
?>
```

上述代码中，在 if 条件控制语句中判断变量“\$boo”中的值是否为 true，如果为 true，则输出“变量\$boo 为真！”，否则输出“变量\$boo 为假！！”，执行效果如图 2-6 所示。

变量\$boo为真!

图 2-6 实例 2-6 的执行效果



2.5.2 整型

在 PHP 程序中,整数数据类型只能包含整数。这些数据类型可以是正数或负数。在 32 位的操作系统中,有效的范围是 $-2\,147\,483\,648 \sim +2\,147\,483\,647$ 。在 64 位系统下,无符号整型的最大值是 $2^{64}-1=18446744073709551615$,最小值为 0;有符号整型数的最大值是 $2^{63}-1=9223372036854775807$,最小值是 $-2^{63}=-9223372036854775808$ 。

在给一个整型变量或者常量赋值时,可以采用十进制、十六进制或者八进制。例如下面的实例演示了使用整型的过程。



实例 2-7: 使用整型

源文件路径: daima\2\2-7

实例文件 index.php 的主要实现代码如下:

```
<?php
    $int_D=2009483648;           //十进制赋值
    echo($int_D);                //输出整型变量值
    echo("<br>");                 //换行
    $int_H=0x7AAAFFFFAA;        //进行十六进制赋值
    echo($int_H);                //输出十六进制变量值
    echo("<br>");                 //换行
    $int_O=016666667766;        //八进制赋值
    echo($int_O);                //输出八进制变量值
    echo("<BR>");                 //换行
?>
```

本实例的执行效果如图 2-7 所示。

```
2009483648
526854913962
1994092534
```

图 2-7 使用整型数据类型的结果

2.5.3 浮点型

浮点型数据类型是用来存储数字的,也可以用来保存小数,它提供的精度比整型数据大得多。在 32 位的操作系统中,有效的范围是 $1.7E-308 \sim 1.7E+308$,给浮点数据类型赋值的方法也很多。在 PHP 4.0 以前的版本中,浮点型的标识为 double,也称为双精度浮点数。请看下面的实例,演示了使用浮点型数据类型的过程。



实例 2-8: 使用浮点型

源文件路径: daima\2\2-8

实例文件 index.php 的主要实现代码如下:

```
<?php
    $float_1=900000000000;      //定义浮点型变量 float_1
```

```

echo($float_1);           //输出变量 float_1 的值
echo("<br>");              //换行
$float_2=9E10;            //定义浮点型变量 float_2
echo($float_2);           //输出变量 float_2 的值
echo("<br>");              //换行
$float_3=9E+10;           //定义浮点型变量 float_3
echo($float_3);           //输出变量 float_3 的值
?>

```

本实例的执行效果如图 2-8 所示。

```

90000000000
90000000000
90000000000

```

图 2-8 使用浮点型数据类型的结果

2.5.4 字符串

字符串是一个连续的字符序列，字符串中的每个字符只占用一个字节。在 PHP 程序中，有如下 3 种定义字符串的方式。

- 单引号方式。
- 双引号方式。
- Heredoc 方式。



实例 2-9：使用字符串

源文件路径：daima\2\2-9

实例文件 index.php 的主要实现代码如下：

```

<?php
$single_str='我被单引号括起来了!<br>'; //定义字符串变量 single_str
echo $single_str; //输出变量 single_str 的值
$single_str='输出单引号： \'嘿嘿，我在单引号里面\'<br>';
echo $single_str; //输出变量 single_str 的值
$single_str='输出双引号： "我在双引号里面"<br>';
print $single_str; //输出变量 single_str 的值
$single_str='输出美元符号： $'; //定义字符串变量 single_str
print $single_str;
$single_str="输出单引号： '嘿嘿，我在单引号里面'<br>";
//定义字符串变量，不需要转义符
echo $single_str;
$single_str="输出双引号： \"我在双引号里面\"<br>"; //定义字符串变量，需要转义符
print $single_str;
$single_str="输出美元符号： \$ <br>"; //定义字符串变量，需要转义符
print $single_str;
$single_str="输出反斜杠： \\ <br>"; //定义字符串变量，需要转义符
print $single_str;
$heredoc_str =<<<heredoc_mark //Heredoc 方式定义变量
你好<br>
美元符号 $ <br>

```




```
反斜杠  \<br>
"我爱你"<br>
'我恨你'
heredoc_mark;
    echo $heredoc_str;           //输出 Heredoc 方式定义的变量
?>
```

本实例的执行效果如图 2-9 所示。

```
我被单引号括起来了!
输出单引号：'嘿嘿，我在单引号里面'
输出双引号："我在双引号里面"
输出美元符号：$输出单引号：'嘿嘿，我在单引号里面'
输出双引号："我在双引号里面"
输出美元符号：$
输出反斜杠：\
你好
美元符号 $
反斜杠 \
"我爱你"
'我恨你'
```

图 2-9 使用字符串数据类型的结果



知识精讲

在进行类型转换的过程中应该注意，当转换成 `boolean` 类型时，`null`、`0` 和未赋值的变量或数组会被转换为 `false`，其他的为 `true`。当转换成整型时，布尔型的 `false` 转换为 `0`，`true` 转换为 `1`，浮点型的小数部分被舍去，字符型如果以数字开头就截取到非数字位，否则输出 `0`。

2.6 运算符和表达式



在 PHP 语言中，运算符是对变量、常量或者数据进行计算的符号，它对一个值和一组值执行一个指定的操作。PHP 语言中的运算符包括算术运算符、赋值运算符、逻辑运算符、比较运算符、字符串运算符、逻辑运算符、自增/自减运算符、位运算符、执行运算符和错误控制运算符。而表达式是由运算符和变量或常量组成的式子。

↑ 扫码看视频

2.6.1 表达式

表达式是 PHP 语言中最重要基石，几乎所有的 PHP 代码都是一个表达式。最基本的



表达式形式是常量和变量。当键入表达式“\$a = 5”时，表示将值“5”分配给变量 \$a。\$a 的值为 5。换句话说，\$a 是一个值为 5 的表达式(在这里，“5”是一个整型常量)。赋值之后，所期待情况是 \$a 的值为 5，因而如果写下 \$b = \$a，期望的是它犹如 \$a = 5 一样。换句话说，\$b 是一个值也为 5 的表达式。如果一切运行正确，那这正是将要发生的正确结果。

在 PHP 程序中，表达式通过具体的代码来实现，是一个个符号组合起来形成的代码。而这些符号只是一些对 PHP 解释程序有具体含义的原子单元，它们可以是变量名、函数名、运算符、字符串、数值和括号等。例如在下面的代码中，是由两个表达式组成的一个 PHP 代码，即“fine”和“\$a=“word””。

```
<?php
"fine" ;
$a = "word" ;
?>
```

在 PHP 程序代码中，使用分号“;”来区分表达式，表达式也可以包含在括号内。我们可以这样理解，一个表达式加上一个分号后就是一条 PHP 语句。



提示：在编写 PHP 程序时，应该注意不要漏写表达式后面的分号“;”，这是一个出现频率很高的错误。

2.6.2 算术运算符

算术运算符是处理四则运算的符号，是最简单的，也是使用频率最高的运算符，尤其是对数字的处理。PHP 语言中常用的算术运算符如表 2-2 所示。

表 2-2 算术运算符

名 称	操 作 符	示 例
加法运算	+	\$a + \$b
减法运算	-	\$a-\$b
乘法运算	*	\$a * \$b
除法运算	/	\$a / \$b
取余数运算	%	\$a % \$b
累加	++	\$a ++
自减	--	\$a --



实例 2-10：使用算术运算符
源文件路径：daima\2\2-10

实例文件 index.php 的主要实现代码如下：

```
<?php
$a= 21; //定义变量 a
```



```
$b= 22;           //定义变量 b
$c= 23;           //定义变量 c
echo $a+$b . "<br>"; //加
echo $a-$b . "<br>"; //减
echo $a*$b . "<br>"; //乘
echo $a/$b . "<br>"; //除
echo$a%$b. "<br>"; //取余数
?>
```

本实例的执行效果如图 2-10 所示。

```
43
-1
462
0.95454545454545
21
```

图 2-10 使用算术运算符的执行效果

2.6.3 赋值运算符

赋值运算符是把基本赋值运算符“=”右边的值赋给左边的变量或者常量，PHP 中常用的赋值运算符如表 2-3 所示。

表 2-3 赋值运算符

操 作	符 号	示 例	展开形式	意 义
赋值	=	\$a=b	\$a=3	将右边的值赋给左边
加	+=	\$a+= b	\$a=\$a + b	将右边的值加到左边
减	-=	\$a-= b	\$a=\$a-b	将右边的值减到左边
乘	*=	\$a*= b	\$a=\$a * b	将左边的值乘以右边
除	/=	\$a/= b	\$a=\$a / b	将左边的值除以右边
连接字符	.=	\$a.= b	\$a=\$a. b	将右边的字串加到左边
取余数	%=	\$a%= b	\$a=\$a % b	将左边的值对右边取余数

基本的赋值运算符是“=”，它的含义不是“等于”。它实际上意味着把右边表达式的值赋给左边的运算数。赋值运算表达式的值也就是所赋的值，例如“\$a=3”的值是 3。



实例 2-11：使用赋值运算符

源文件路径：daima\2\2-11

实例文件 index.php 的主要实现代码如下：

```
<?php
$a = ($b = 4) + 5; // $a 现在成了 9，而 $b 成了 4
echo $a;           //输出变量 a 的值
echo "和";         //输出字符“和”
echo $b;           //输出变量 b 的值
?>
```



本实例的执行效果如图 2-11 所示。

9和4

图 2-11 实例 2-11 的执行效果

2.6.4 自增/自减运算符

自增/自减运算符有两种使用方法，一种是先将变量增加或者减少 1，再将值赋给原变量，称为前置递增或递减运算符；另一种是将运算符放在变量后面，即先返回变量的当前值，然后变量的当前值增加或者减少 1，称为后置递增或递减运算符。PHP 中的自增/自减运算符的说明如表 2-4 所示。

表 2-4 PHP 中的自增/自减运算符

操 作	符 号	示 例	展开形式	意 义
前加加	++	++\$a	\$a=++\$a+1	\$a 的值加 1，然后返回 \$a
后加加	++	\$a++	\$a=(\$a++)-b	返回 \$a，然后将 \$a 的值加 1
前减减	--	--\$a	\$a=--\$a-b	\$a 的值减 1，然后返回 \$a
后减减	--	\$a--	\$a=(\$a--) * b	返回 \$a，然后将 \$a 的值减 1



实例 2-12：使用自增/自减运算符

源文件路径：daima\2\2-12

实例文件 index.php 的主要实现代码如下：

```
<?php
echo "<h3>Postincrement</h3>";
$a = 5;
echo "Should be 5: " . $a++ . "<br />\n";           //使用后++，先返回原值
echo "Should be 6: " . $a . "<br />\n";           //使用后++，后返回加 1
echo "<h3>Preincrement</h3>";
$a = 5;
echo "Should be 6: " . ++$a . "<br />\n";         //使用前++，先返回加 1
echo "Should be 6: " . $a . "<br />\n";         //使用前++，后返回也是加 1
echo "<h3>Postdecrement</h3>";
$a = 5;
echo "Should be 5: " . $a-- . "<br />\n";         //使用后--，先返回原值
echo "Should be 4: " . $a . "<br />\n";         //使用后--，后返回减 1
echo "<h3>Predecrement</h3>";
$a = 5;
echo "Should be 4: " . --$a . "<br />\n";         //使用前--，先返回减 1
echo "Should be 4: " . $a . "<br />\n";         //使用前--，后返回也是减 1
?>
```

本实例的执行效果如图 2-12 所示。



Postincrement

Should be 5: 5
Should be 6: 6

Preincrement

Should be 6: 6
Should be 6: 6

Postdecrement

Should be 5: 5
Should be 4: 4

Predecrement

Should be 4: 4
Should be 4: 4

图 2-12 实例 2-12 的执行效果



知识精讲

单引号和双引号的区别

在使用单引号时，只要对单引号“'”进行转义即可，但是在使用双引号时，还要注意“'”“\$”等字符的使用。这些特殊字符都要通过转义符“\”来显示。

2.7 实践案例与上机指导



通过对本章内容的学习，读者基本可以掌握 PHP 语言的基本语法知识。其实 PHP 语言的语法知识还有很多，这需要读者通过课外渠道来加深学习。下面通过练习操作，以达到巩固学习、拓展提高的目的。

↑ 扫码看视频

2.7.1 使用特殊类型

在 PHP 程序中，特殊数据类型包括 Resource(资源)和 Null(空值)两种，具体说明如下所示。

- **Resource:** 是 PHP 内的几个函数所需要的特殊数据类型，由编程人员来分配。
- **Null:** 是最简单的数据类型，表示没有为该变量设置任何值，并且是空值，不区

分大小写。

1) 资源

当在 PHP 程序中使用资源时，系统会自动启用垃圾回收机制，释放不再使用的资源，避免内存消耗殆尽。因此，资源很少需要手工释放。

2) 空值

空值，顾名思义，表示没有为该变量设置任何值。另外，空值不区分大小写，`null` 和 `NULL` 效果是一样的。被赋予空值的情况有以下 3 种：还没有赋任何值、被赋值 `null`、被 `unset()` 函数处理过的变量。



实例 2-13：使用空值

源文件路径：daima\2\2-13

实例文件 `index.php` 的主要实现代码如下：

```
<?php
echo "变量(\$string1)直接赋值为 null: ";
$string1 = null;                                //变量$string1 被赋空值
$string3 = "str";                                //变量$string3 被赋值 str
if(is_null($string1))                            //判断$string1 是否为空
    echo "string1 = null";
echo "<p>变量(\$string2)未被赋值: ";
if(is_null($string2))                            //判断$string2 是否为空
    echo "string2 = null";
echo "<p>被 unset() 函数处理过的变量(\$string3): ";
unset($string3);                                //释放$string3
if(is_null($string3))                            //判断$string3 是否为空
    echo "string3 = null";
?>
```

在上述代码中，字符串 `string1` 被赋值为 `null`，`string2` 根本没有声明和赋值，所以也输出 `null`，最后的 `string3` 虽然被赋予了初值，但被 `unset()` 函数处理后，也变为 `null` 型。`unset()` 函数的作用就是从内存中删除变量。其执行效果如图 2-13 所示。

变量(\$string1)直接赋值为null : string1 = null

变量(\$string2)未被赋值 : string2 = null

被unset()函数处理过的变量(\$string3) : string3 = null

图 2-13 实例 2-13 的执行效果

2.7.2 检测数据类型

在 PHP 程序中内置了检测数据类型的系列函数，可以对不同类型的数据进行检测，判断其是否属于某个类型。如果符合则返回 `true`，否则返回 `false`。PHP 中检测数据类型的函数如表 2-5 所示。



表 2-5 PHP 中的检测数据类型函数

函 数	检测类型	举 例
is_bool	检查变量是否是布尔类型	is_bool(true)、is_bool(false)
is_string	检查变量是否是字符串类型	is_string('string')、is_string(1234)
is_float/is_double	检查变量是否为浮点类型	is_float(3.1415)、is_float('3.1415')
is_integer/is_int	检查变量是否为整数	is_integer(34)、is_integer('34')
is_null	检查变量是否为 null	is_null(null)
is_array	检查变量是否为数组类型	is_array(\$arr)
is_object	检查变量是否是一个对象类型	is_object(\$obj)
is_numeric	检查变量是否为数字或由数字组成的字符串	is_numeric('5'). is_numeric('bcd110')



实例 2-14：检测指定的变量是否是数字

源文件路径：daima\2\2-14

实例文件 index.php 的主要实现代码如下：

```
<?php
$boo = "043112345678";           //声明一个全由数字组成的字符串变量
if(is_numeric($boo))              //判断该变量是否由数字组成
    echo "Yes,the \$boo a phone number: $boo!"; //如果是，输出该变量
else
    echo "Sorry,This is an error!"; //否则，输出错误语句

?>
```

在上述代码中，使用函数 is_numeric()检测了在变量中的数据是否是数字，从而了解并掌握 PHP 中 is 系列函数的用法。其执行效果如图 2-14 所示。

Yes,the \$boo a phone number: 043112345678!

图 2-14 实例 2-14 的执行效果

2.8 思考与练习

本章首先介绍了 PHP 语言的基本语法结构，接着详细阐述了 PHP 注释的知识，然后讲解了常量、变量和数据类型的用法，最后讲解了运算符和表达式的知识。在讲解过程中，通过具体实例介绍了 PHP 基础语法的使用方法，应掌握其使用方法和技巧。

1. 选择题

(1) PHP 语言中的三元运算符是()。

- A. ?: B. ? C. :? D. .?



(2) 假设变量 $a=23$, $b=124$, 则 $\$a \& \b 的结果是()。

- A. 20 B. 127 C. 147 D. -121

2. 判断对错

(1) 基本的赋值运算符是“ $=$ ”，它的含义不是“等于”。它实际上意味着把右边表达式的值赋给左边的运算数。 ()

(2) PHP 是一门灵活优秀的开发语言，在编写 PHP 程序时，可以漏写表达式后面的分号“ $;$ ”，程序会正确执行。 ()

3. 上机练习

(1) 在 PHP 程序中动态输出 JavaScript 代码。

(2) 在 PHP 程序中输出数字和字符串。