第1章 准备工作

PHP(Pre Hypertext Preprocessor,超文本预处理器)是一种跨平台开源脚本语言,同时 也是一种流行的Web服务器开发技术。PHP代码可以很方便地嵌入HTML页面,并通过大 量的内置和第三方模块实现各种功能,如文件处理、数据库操作、图形图像处理、邮件发 送、Excel文档处理等。

本章将介绍 PHP 网站开发与测试环境的搭建,以及 MariaDB 数据库的安装与配置。

1.1 PHP 开发与测试环境

虽然 PHP 网站大多数会运行在 Linux 系统中,但本书介绍的是 PHP 网站所需要的代码 实现,为便于写作和测试,书中的示例会在 Windows 环境中完成,这些代码可以直接或稍 作修改后在 Linux 环境下运行。

本节介绍如何在 Windows 系统下搭建 PHP 网站的开发与测试环境。

1.1.1 获取 PHP

首先,可以从 php.net 网站下载需要的 PHP 版本,如果使用 IIS (Internet Information Services,互联网信息服务)和 FastCGI 方式运行 PHP,需要下载无线程安全的版本,PHP 7.4.10 的文件名为 php-7.4.10-nts-Win32-vc15-x64.zip (64 位系统),本书示例解压路径为 d:\php7nts 目录,如果解压在不同的目录,配置参数时注意使用正确的路径。

如果需要使用 Apache HTTP Server 作为 Web 服务器,需要下载线程安全版本,PHP 7.4.10 的文件名为 php-7.4.10-Win32-vc15-x64.zip(64 位系统),本书示例解压路径为 d:\php7ts 目录。

Windows 系统下的 PHP 配置文件为 php.ini。在 PHP 目录中包含两个配置文件模板,分别是 php.ini-development 和 php.ini-production 文件。开发和测试工作中,可以将 php.ini-development 文件复制一份,重新命名为 php.ini。配置文件中,以分号(;)开始的内容都是注释,如果需要启用参数,需要删除分号。

首先是扩展资源的路径,在 php.ini 文件中找到 extension_dir 参数,确保删除了以下代码前的分号。

extension_dir ="ext"

此参数设置了 Windows 系统下默认的扩展模块存放路径,除了使用相对路径,还可以 使用绝对路径,如下面的代码。

extension_dir="d:/php7ts/ext/"

默认情况下,扩展资源位于 PHP 目录下的 ext 目录中,本书需要的扩展模块包括: ● gd2,绘图功能。请注意,在 PHP 8 中,此模块更名为 gd。

- mbstring, 处理多字节文本。
- mysqli, MySQL 和 MariaDB 等数据库操作。
- openssl, OpenSSL 支持。

在 php.ini 文件中找到相应的 extension 参数,例如,需要启用 mbstring 模块就要删除 "extension=mbstring" 前面的分号 (;)。

此外,通过命令行执行 PHP 代码时,可以使用 php.exe 命令。首先,创建 d:\test.php 文 件,并使用"记事本"应用程序打开,修改内容如下。

```
<?php
echo "Hello PHP";
?>
```

2

通过 cmd.exe 打开命令行窗口,按顺序执行如下命令。

```
d:
cd php7nts
php d:\test.php
```

执行成功时,可以在命令行窗口中看到输出"Hello PHP"字样。

1.1.2 IIS 和 FastCGI

Windows 系统中, PHP 网站可以通过 FastCGI 方式在 IIS 中运行。在 Windows 7 或 Windows 10 系统中通过"控制面板"打开"程序和功能",并在"启动或关闭 Windows 功能" 中修改 Windows 组件,见图 1-1,需要启用 Internet Information Services \rightarrow "万维网服务" \rightarrow "应用程序开发功能" \rightarrow CGI 等功能。



图 1-1

接下来,通过运行 inetmgr.exe 打开 IIS 管理器。这里可以新建一个网站,并指定网站的端口为 10001,路径为 PHP 网站的测试路径,如本书使用的 d:\chy\cpl\site,见图 1-2。

第1章 准备工作 3



图 1-2

创建网站后,打开网站的"处理程序映射",在右上角选择"添加模块映射",见图 1-3。

Signature Services (IIS)管理器	- 🗆 🗙	
(← →) ● ▶ DESKTOP-7S1D3N1 ▶ 网站 ▶ phpsite ▶	📅 🖂 🔂 🔹	
文件(F) 视图(V) 報助(H)		
选援 phpsite 主页	操作	操作
● ●	浏览 编辑权限 編輯网站 鄧 基本设置 查看应用程序	添加托蓉大理程序 添加基本染射 添加基本染射 添加提快染射 编辑功能仅限 伏复为公规 音響经计结束的列表
> - 11 demo CGI HTTP 响应标 ISAPI 筛选器 MIME 类型 SSL 设置 处理程序映 > - 11 fonts 头 射	管理网站	2 帮助
→ 二 ing → 二 ib → 二 questionnc → 二 user → user → 二 user →	 ◇ 重新回説 >> 回説 今止 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
□ □ □ ● ● ● ● ● ● ● ● ● ● ● ●	Prota HSTS ② 帮助	
就绪	¶.:	



接下来参考图 1-4 中的参数进行设置。

编辑模块映射	? ×
请求路径(2):	
t.php	
示例: *.bas, wsvc.axd	
模块(<u>M</u>):	
FastCgiModule	~
可执行文件(可选)(E):	
D:\php7nts\php-cgi.exe	
名称(<u>N</u>):	
php	
请求限制(<u>B</u>)	
确定	取消
图 1-4	

网站全栈开发指南: HTML+CSS+JavaScript+PHP 四校 正文.indd 3

这里设置的参数主要如下。

- 请求路径:这里指定 PHP 文件的扩展名,即*.php。
- 模块: 在列表中选择"FastCgiModule"。
- 可执行文件:这里需要选择 PHP 无线程安全版本中的 php-cgi.exe 文件,如本书示例 中的 D:\php7nts\php-cgi.exe。
- 名称:这里指定为 php。

网站创建后,如果需要修改网站的主目录,可以在网站的"Internet Information Services(IIS)管理器"中选择网站,打开"基本设置"后设置。如图 1-5 所示,在"编辑网站"对话框的"物理路径"文本框中调整具体路径。

编辑网站	?	×
网站名称(S): 应用程序池(L): php-site	选择(E)	
物理路径(D): D:\chy\cp\site		
确定	取消	

图 1-5

IIS 网站配置的最后一步,打开网站设置中的"默认文档",确认其中包含 index.php 文件,如果没有可以添加,并将其移动到列表的第一位,见图 1-6。

💐 Internet Information Servi	ces (IIS)管理器			_		\times
	P-7S1D3N1) 网站) phpsite)			1	6	• 1
文件(F) 视图(V) 帮助(H)					/	
连接 21	● 默认文档		操作 添加			
✓ - ● DESKTOP-7S1D3N1 - 应用程序池	使用此功能指定当客户端未请求特定 默认文件。按优先级顺序设置默认文	文件名时返回的 档。	禁用 恢复为父	项		
 Mpike Default Web S Physite map demo fonts fonts fonts for pysice questionne mains 	名称 index.php Default.tasp index.htm index.html iisstart.htm default.aspx	会目 <u>类型</u> 本地 继承 继承 继承 继承 继承 继承 继承 继承	❷ 帮助			
< >> >> 配署:"phpsite" web config	🔟 功能视图 👫 内容视图					G E



本书网站位于源代码中的 site 目录,测试时可以将此目录设置为网站路径,也可以创建 自己的网站目录,并在其中创建 index.php。接下来,简单的代码测试会通过此文件完成。

1.1.3 Apache

开源的江湖上,有 Linux+Apache+PHP+MySQL(简称 LAPM)黄金组合的传说,所以, 了解一下 Apache 的应用是很有必要的。如果使用 Apache HTTP Server 环境运行 PHP 网站,



需要下载 PHP 的线程安全版本,文件名类似 php-7.4.10-Win32-vc15-x64.zip,本书解压的位置是 d:\php7ts,请注意配置其中的 php.ini 文件。

本书以 Apache 2.4 为例,可以在 http://httpd.apache.org/download.cgi 下载所需要的文件, 64 位版本文件名为 httpd-2.4.46-o111h-x64-vc15.zip,下载后将文件解压到 d:\Apache24 目录。

解压后的 Apache 资源中, conf 目录包含一些配置文件, 这里重点关注 httpd.conf 文件。 该文件可以使用"记事本"应用程序进行编辑,下面是一些常用的配置项。

- Define SRVROOT "d:/Apache24",设置 Apache 的主目录。
- Listen 10002,设置侦听的端口,这里指定为 10002。
- ServerName localhost:10002,指定访问服务名称。
- DocumentRoot "d:/chy/cpl/site" 和 <Directory "d:/chy/cpl/site" >, 指定 PHP 网站的根 目录路径等相关参数。
- DirectoryIndex index.php index.html,指定网站中的默认页面文档。

下面的配置用于支持 PHP 7,可以将这些内容添加到 httpd.conf 文件的最后。

```
LoadModule php7_module d:/php7ts/php7apache2_4.dll
AddType application/x-httpd-php .php .html .htm
PHPIniDir "d:/php7ts/"
```

此外,httpd.conf 配置文件中以#符号开始的是注释内容。

Apache 配置完成后,需要安装和启动 Apache 服务,可以使用 bin 目录中的 httpd.exe 命 令和 -k 参数执行一系列操作,如:

- install, 将 Apache 安装为 Windows 系统服务。
- uninstall, 卸载 Apache 服务。
- start, 启动 Apache 服务。
- restart, 重启 Apache 服务。
- stop, 停止 Apache 服务。

以管理员方式启动 cmd.exe, 然后进入 d:\apache24\bin 目录, 下面的命令会将 Apache 安装为 Windows 系统服务。

d:\apache24\bin>httpd -k install

然后,通过下面的代码启动 Apache 服务。

d:\apache24\bin>httpd -k start

开发和测试过程中,也可以同时运行多个 PHP 网站,例如,在 IIS 中运行 PHP 7 网站, 在 Apache 中运行 PHP 8 网站,网站的目录可以设置为相同的路径,这样就可以观察代码在 不同 PHP 版本下的运行效果。

1.2 开发工具

实际上,本书的代码编辑工作只需要一个文本编辑器就可以了,当然,如果支持语法高 亮更好。Notepad++就是一款不错的文本编辑器,下面简单了解一下它的使用方法。

6

首先,可以从 https://notepad-plus-plus.org/ 获取 Notepad++ 的最新版本,下载后按提示 安装即可。如果下载的是 zip 压缩包,解压后可直接使用。

启动 Notepad++,如果界面显示的是英文,可以通过菜单 Settings \rightarrow Preferences 打开偏好设置窗口,在 Localization 列表中选择语言,如"中文简体"。

通过菜单项"文件"→"打开文件夹作为工作区"打开网站所在的根目录,见图 1-7。

📔 new 1 - Notepad++	
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N)	语言(L)设置(T)工具(O)宏(M)运行(R)插件(P)窗口(W)?
新建(N)	Ctrl+N 🤏 🔫 🖫 🔤 🖿 11 🧱 🐺 🎉 🖾 🜰 🔍 🗉 🕨 🕨 🔤
打开(O)	Ctrl+O
打开所在文件夹	>
使用默认查看器打开	
打开文件夹作为工作区	
重新读取文件(L)	Ctrl+R

图 1-7

默认情况下,Notepad++可以根据文件的扩展名判断文件类型,并使用不同的颜色显示 关键字等内容。使用过程中,也可以切换代码文件的语言类型,通过菜单项"语言"选择相 应的编程语言类型;对于 PHP,可以通过菜单"语言"→P→PHP 选择。对于新建的文件, 在保存之后就可以识别文件类型,例如,PHP 代码保存为.php 文件后就会自动识别并高亮 显示关键字等内容。

1.3 安装 MariaDB 数据库

MariaDB 是 MySQL 数据库的一个重要分支,本书主要以 MariaDB 数据库为例,但绝大部分操作同样适用于 MySQL 数据库。

MariaDB 数据库的网站为 https://mariadb.org/,可以从这里下载与操作系统相对应的版本,如 64 位 Windows 系统,下载的文件类似 mariadb-10.5.5-winx64.zip。本书示例中,会将下载的 zip 文件解压到 d:\mariadb10 目录。

如果需要简化安装过程,也可以直接下载安装文件,如 mariadb-10.4.14-winx64.msi。请 注意,如果使用 Windows 7 操作系统,需要下载 10.4 版本。

安装过程中,需要注意几个配置项,见图 1-8。



图 1-8



这里,第一个选项是 root 用户的密码,本书使用 DEV_Test123456,在连接数据库时注 意使用正确的密码。第二个选项,是否允许 root 用户远程登录,如果只是在本机学习和测 试,可以不选择。第三个选项是在 MariaDB 服务器中默认使用 UTF-8 字符集。设置完成后, 单击 Next 按钮继续,见图 1-9。

👘 Database settings			-	
Default instance p	roperties		MariaD	BR
MariaDB 10.4 (x64)	latabase configuratior	1	Serve	er er un
✓ Install as service				
Service Name:	mariadb			
Enable networking				
TCP port:	3333			
Innodb engine set	tings			
Buffer pool size:	1005 MB			
Page size:	16 ▼ KB			
		Back	Next	Cancel

图 1-9

这里, MariaDB 数据库的系统服务名为 mariadb。请注意,本书测试环境中,TCP 端口 修改为 3333, 避免与 MySQL 数据库默认的 3306 端口冲突。最后,按提示完成安装。

1.3.1 安装和卸载 MariaDB 服务

接下来介绍在 Windows 10 中如何通过命令行安装和配置 MariaDB 数据库。首先,下载的压缩文件中没有包含 data 目录,需要通过初始化操作创建。

通过管理员身份执行 cmd.exe 命令,并通过如下命令进入 bin 目录。

```
d:
cd mariadb10\bin
```

在 bin 目录中执行下面的命令,对 MariaDB 数据库进行初始化操作。

mariadb-install-db

接下来,通过下面的命令将 MariaDB 安装为 Windows 系统服务。

mysqld --install mariadb

默认情况下,服务名为 MySQL,这里指定服务名为 mariadb,使用不同的服务名,可以在一台计算机中安装多个版本的 MySQL 或 MariaDB 数据库。

除了不同的服务名,还需要区分服务的端口,MySQL数据库服务的默认端口为3306,本书示例中,可以通过 d:\mariadb10\data\my.ini 文件修改 MariaDB 配置参数。下面就是本书使用的配置内容。

```
[mysqld]
datadir=D:/mariadb10/data
port=3333
character_set_server=utf8
character_set_client=utf8
```

[client]
plugin-dir=D:/mariadb10/lib/plugin

这里将 mariadb 服务的端口设置为 3333,连接数据库时,需要设置正确的服务器地址 和端口。此外,这里还将 MariaDB 数据库服务器端和客户端字符集设置为 utf-8,这样就可 以有效地传递汉字等多字节文本信息。

最后,启动数据库服务,如下面的命令。

net start mariadb

8

卸载 MariaDB 服务器时,首先需要停止 mariadb 服务,同样使用管理员身份运行 cmd. exe,然后执行如下命令。

net stop mariadb

卸载 MariaDB 服务时可以在 d:\mariadb10\bin 目录中执行如下命令。

mysqld --remove

也可以通过 Windows 命令来删除系统服务,如下面的命令。

sc delete mariadb

1.3.2 修改 root 用户密码

修改 MariaDB 数据库用户的登录密码,可以使用 bin 目录中的 mysqladmin 工具,在命令行环境中执行如下命令。

mysqladmin -u root -p -P 3333 password

执行命令后,根据提示输入旧的密码(默认为空),然后输入两次新的密码,操作无误 后就会修改 root 用户的密码。这里使用的参数如下:

- ●-u,小写字母u,指定登录的用户名。
- -p, 小写字母 p, 登录时需要输入密码。
- -P, 大写字母 P, 指定数据库服务的端口。不使用此参数时, 默认端口为 3306, 如 果改变了数据库服务的端口, 就需要在登录时指定正确的端口。

● password,执行修改密码操作。

修改密码后请牢记,本书示例中,连接数据库时应指定为实际的登录密码。

此外,远程登录 MariaDB 服务器时,还需要使用 -h 参数指定服务器地址,如下面的代码就是使用 root 用户登录本机的 MariaDB 服务。

mysql -h 127.0.0.1 -P 3333 -u root -p

本例包含了登录数据库的主要信息,包括服务器地址、服务端口号、用户名和密码,其 中,服务器地址指定的是本机(127.0.0.1)。

1.3.3 命令行操作界面

操作 MariaDB 数据时,可以使用 mysql 命令启动客户端命令行工具,如下面的命令。

mysql -P 3333 -D mysql -u root -p

第1章 准备工作

9

命令中的 -D 参数指定连接后的默认数据库,本例就是 mysql,见图 1-10。



图 1-10

退出客户端命令行环境时,可以使用 quit 命令。

1.3.4 使用 HeidiSQL

Windows 下操作 MariaDB 数据库,使用图形化工具是一个不错的选择,如 HeidiSQL, 下载网址为 www.heidisql.com。通过安装包安装 MariaDB 数据库时,其中包含了 HeidiSQL, 可以根据需要选择安装,也可以从 HeidiSQL 网站获取最新的版本。

打开 HeidiSQL,可以创建新的数据库连接,并保存连接信息,见图 1-11。



图 1-11

这里,需要注意 MariaDB 数据库服务器的连接信息,主要包括:

- 主机名/IP。指定 MariaDB 数据库服务器的计算机名或 IP 地址。图 1-11 中的 127.0.0.1 指本机。
- 用户。登录数据库的用户名。
- 密码。登录密码。
- 端口。数据库服务端口,3306 为默认端口,本书中,MariaDB 数据库服务端口使用 了 3333。

正确连接 MariaDB 数据库后, 主界面见图 1-12。后续内容中会介绍在此环境下操作 MariaDB 数据库。

MariaDB(localhost)\ - HeidiSQL 11.0.0.5919	- 🗆 X
文件 编辑 搜索 工具 转到 帮助	
🚿 • 💉 📭 🖻 to 🖶 🔕 • 🕫 🖬 🐋 🕗 🖊 🕨 🔇	🕽 🛇 🖂 🕨 ד 🖿 🗕 🕶 💾 📖 🔍 💭 🈼 👫
🐛 数据库过滤器 🔍 表过滤器 🔶	
✓	📃 数据库 (4) 🔹 变量 🔸 状态 🕨 进程 📊 命令统计
 information_schema mysql performance_schema test 	数据库 ~ 大小 项目 上次 information_schema mysql performance_sche test
	× 过滤: 正则表达式
24/* 连接到 127.0.0.1 (经由 MariaDB (TCP/I 25 SELECT CONNECTION ID();	P)),用户名 root,密码: Yes */
26/* <i>已连接,线程ID: 9 */</i>	
27 /* 字符集: utf8mb4 */	
🕓已连接: 0(🚄 MariaDB 10	.5.5 运行时间: 00:54 h 🕓 服务器时 🔵 空闲。

图 1-12

这里,我们已经在 Windows 系统下创建了 PHP 网站和 MariaDB 数据库的测试环境。第 2 章将开始学习服务器端 PHP 开发的基础知识。

第2章 PHP开发基础

PHP 代码文件一般以.php 为扩展名,作为文本文件,可以使用自己喜欢的文本编辑器进行编辑。本章将介绍如何编写 PHP 代码,并了解变量、常量和注释的应用。

2.1 页面中添加 PHP 代码

页面文件中, PHP 和 HTML 等代码可以混合编写, 而 PHP 代码应该定义在 <?php 和 ?> 之间。例如修改网站根目录下 index.php 文件的代码如下。

```
<?php
phpinfo();
?>
```

页面会显示 PHP 环境的配置信息,按照本书 IIS 网站的配置,在浏览器中可以通过以下地址访问网站:

- http://127.0.0.1:10001/
- http://127.0.0.1:10001/index.php
- http://localhost:10001/
- http://localhost:10001/index.php

如果 PHP 及 IIS 配置正确, 会看到如图 2-1 所示的页面。

PHP 7.4.10 - phpinfo() × +	- 🗆 ×	
← → C (i) 127.0.0.1:10001	🛚 \star 😁 :	
PHP Version 7.4.10	php	*
System	Windows NT DESKTOP-7S1D3N1 10.0 build 19042 (Windows 10) AMD64	
Build Date	Sep 1 2020 16:45:02	
Compiler	Visual C++ 2017	
Architecture	x64	
Configure Command	cscript /nologo /ejscript configure.js "enable-snapshot-build" *enable-debug-pack" *disable-zts" * with-pdo-oci=c\php-snap-build\deps aux\oracle\x64\instantClient_12 \1x4k,shared "with-oci8- 12c=c\php-snap-build\deps aux\oracle\x64\instantClient_12 \1x4k,shared" *with-oci8- 12c=c\php-snap-build\deps aux\oracle\x64\instantClient_12 \1x4k,shared "enable-object-out- dir=./obj/" *enable-com-dotnet=shared "without-analyzer" *with-pgo"	
Server API	CGI/FastCGI	
Virtual Directory Support	disabled	
Configuration File (php.ini) Path	no value	
Loaded Configuration File	D:\php7nts\php.ini	
Scan this dir for additional .ini files	(none)	
Additional .ini files parsed	(none)	
PHP API	20190902	
PHP Extension	20190902	
Zend Extension	320190902	•

图 2-1



请注意图 2-1 中箭头所指的位置,这是当前 PHP 环境使用的配置文件的路径(Loaded Configuration File),需要改变 PHP 环境参数时,应修改此文件并重启 PHP 网站,必要时可能还需要重启计算机,然后才会应用新的配置参数。

需要在页面中输出文本内容时,可以使用 print 或 echo 语句,它们的区别在于, print 语 句只使用一个参数, echo 语句可以使用一个或逗号分隔的多个参数,如下面的代码。

```
<?php
echo "aaa","bbb","ccc","<br>";
print "ddd";
?>
```

代码执行结果见图 2-2, 其中的
 是 HTML 中的换行标记。



图 2-2

另一个显示信息的常用语句是 print_r 语句结构,它适合于显示对象的信息。如下面的 代码使用 print_r 语句显示一个数组的成员。

```
<?php
$arr = range(1,3);
print_r($arr);
?>
```

代码执行结果见图 2-3。



图 2-3

显示结果中, => 符号左侧一对方括号 [] 中的数据是数组成员的索引值, => 符号右侧是 数组成员的数据。稍后会详细介绍数组的应用。

此外,定义 PHP 代码还允许一种简化格式,也就是将 PHP 代码定义在 <? 和 ?>之间。 Windows 系统下,可以将 php.ini 配置文件中的 short_open_tag 参数设置为 1 来启用简化格 式。默认情况下,同时也是推荐的方式,应使用完整的 <?php 和 ?> 标记定义 PHP 代码。

2.2 语句和函数

PHP 代码与 C 或 C++ 比较相似,代码的执行以"语句"作为基本单位,其中,简单的 语句以分号(;)作为结束,而复合语句(语句块、语句结构)使用一对花括号定义,即代码 定义在 {和}之间,其中可能包含一条或多条简单语句,也可能包含其他的复合语句。 给变量赋值就是一条简单语句,如下面的代码就是将整数10赋予 \$x 变量。

```
x = 10;
```

if 语句结构就是复合语句,如下面的代码用于判断一个数值是否为偶数。

```
$x = 10;
if($x % 2 == 0) {
    echo "{$x} 是偶数";
} else {
    echo "{$x} 不是偶数";
```

代码中,首先将 \$x 变量赋值为整数 10,然后判断 \$x 的值是否为偶数,可以修改 \$x 变量的值来观察代码执行结果。

PHP 文件中,除了主流程代码,很多功能代码会进行封装。封装的基本形式有两种, 一种是函数 (function),另一种是类 (class)。

PHP 已经内置了大量的函数,并提供大量的扩展资源,下面了解函数的基本应用,稍 后还会有更多关于代码组织、管理和应用的内容。

函数的组成要素包括:

- 函数名,即函数的名称。
- 参数列表, 参数列表定义在函数名后的一对圆括号中, 用于向函数内部传递数据。
- 返回值类型, PHP 函数的返回值是比较灵活的, 函数可以根据需要返回不同类型的 数据, 或者没有返回数据。
- 函数体,即函数的执行部分,执行代码定义在一对花括号中;如果有返回值,在函数体中使用 return 语句返回。

下面的代码定义了一个简单的 add() 函数,其功能是返回两个数值相加的结果,其中,两个数据分别由参数 \$a 和 \$b 带入函数。

```
<?php
// 定义函数
function add($a,$b)
{
    return $a+$b;
}
// 调用函数
echo add(10,99);
echo "<br>";
echo add(1.2,4.9);
?>
```

代码执行结果见图 2-4。



图 2-4

本例使用 function 关键字定义函数,其中可以看到函数应用的一些要素,如:

- 函数名为 add。
- 函数包括两个参数,即 \$a 和 \$b。
- 函数主体中,使用 return 语句返回 \$a 和 \$b 相加的结果。
- 这里并没有定义函数的返回值类型,那么它是混合型 (mixed)。

代码中调用了两次 add() 函数,分别计算了两个整数和两个浮点数的和,这也体现了 PHP 中数据和函数应用的灵活性。

2.3 变量

变量是指在代码执行过程中,可以根据需要随时改变其数据的标识,在 PHP 中,变量 名使用 \$ 符号开始,第二个字符必须为字母或下画线,然后可以使用字母、下画线和数字的 组合。

PHP 中,变量在使用前并不需要声明,而是在需要的地方直接赋值后使用,如下面的 代码。

```
<?php
$x = 10;
echo $x;
?>
```

执行代码会显示数字 10。可以看到,使用变量时,并没有指定数据类型,这和 C、 C++、Java、C#等编程语言有很大的区别。不过,PHP 中的变量还是包含了数据及其类型信 息的,可以使用 var_dump()函数来查看,其功能是同时显示参数的数据类型和值,如下面 的代码。

```
<?php
$x = 10;
var_dump($x);
?>
```

代码执行结果见图 2-5, 说明 \$x 变量的数据是 10, 其类型是整数 (int)。



图 2-5

2.4 常量

常量是指定义时赋值,但在代码执行过程不能改变其数据的标识。PHP中,可以使用 define()函数定义常量,函数定义如下。

define(string \$name, mixed \$value[, bool \$case_insensitive = false]) : bool

其中,

- \$name 参数使用字符串指定常量名称,可以使用一对双引号或一对单引号定义,如 "MAX_SIZE"。习惯上,常量名称会使用大写字母,单词之间使用下画线(_)连接的形式。
- \$value 参数指定常量的值,使用混合类型(mixed),可以指定不同类型的数据。不过,建议常量还是使用基本的数据类型,如数值、字符串、布尔类型等。
- \$case_insensitive 参数指定是否忽略常量名中字母的大小写,默认为 false,即常量名 区分字母的大小写。请注意,在 PHP 8 中不再允许此参数设置为 true,也就是说,未 来的 PHP 常量名将严格区分字母的大小写,例如, Max_Size 和 MAX_SIZE 表示不 同的常量。

读取常量的数据时,可以使用 constant() 函数,其定义如下。

constant(string \$name) : mixed

constant() 函数只需要一个参数,即常量名。下面的代码演示了如何定义常量并读取它的数据。

```
<?php
define("MAX_SIZE",255);
echo constant("MAX_SIZE");
?>
```

页面会显示 255。此外,也可以直接使用常量名调用,如下面的代码同样会显示 255。

```
<?php
define("MAX_SIZE",255);
echo MAX_SIZE;
?>
```

判断一个常量是否已经定义,可以使用 defined()函数,其定义如下。

```
defined(string $name) : bool
```

参数 \$name 指一个常量名,如果常量已定义,函数将返回 true,否则返回 false。下面 的代码演示了 defined()函数的使用。

```
<?php
define("MAX_SIZE",255);
var_dump(defined("MAX_SIZE"));
echo "<br>";
var_dump(defined("MAX_BUFFER"));
?>
```

代码执行结果见图 2-6。



图 2-6

第一个输出,由于已经定义了常量 MAX_SIZE,所以 defined()函数返回 true;第二个输出,由于没有定义常量 MAX_BUFFER,则返回 false。

true 和 false 是布尔类型 (bool)数据,直接显示它们的值时需要注意,true 会显示 1,而 false 会显示为空,在页面中看不到内容,所以,对于布尔类型的值,可以使用 var_ dump()函数显示。

2.5 注释

注释是在代码中的说明性内容,它们并不是可执行代码,但可以对代码添加必要的说 明,帮助开发人员和维护人员更有效地理解代码的功能。

PHP 代码中可以使用三种风格的注释,分别是:

- 使用 // 开始的行注释。
- 使用 # 开始的脚本风格的行注释。
- 使用包含在 /* 和 */ 之间的块注释。

下面的内容演示了不同注释风格的应用。

```
<?php
// 定义两个数据
$x = 10; // 数据 1
$y = 99; # 数据 2
/*
运算结果
*/
// 加法运算结果
echo "$x + $y = ",$x+$y,"<br>";
# 乘法运算结果
echo "$x * $y = ",$x*$y;
?>
```

代码执行结果见图 2-7。



第3章将介绍 PHP 中的基本数据类型,以及它们的运算和类型转换等操作。

第3章 数据类型及操作

软件开发中,数据处理是最基础的工作,本章将介绍 PHP 中支持的基本数据类型,以 及它们的运算特点、类型判断和转换,数据处理的常用资源等内容。

3.1 整数

PHP 中可以处理有符号整数,包括负整数、零和正整数。整数的处理范围与 long 类型 (C语言类型)相同。32 位平台中,整数的处理范围是 -2 147 483 648 ~ +2 147 483 647,即 -2³² ~ +2³²-1;64 位平台中,整数的处理范围是 -9 223 372 036 854 775 808 ~ +9 223 372 036 854 775 807,即 -2⁶⁴ ~ +2⁶⁴-1。

在 PHP 中, 数字字面量除了常用的十进制, 还可以使用其他进制的整数, 如:

- 十六进制, 使用 0x 开始的数值表示十六进制数, 如 0x1F 表示十进制的 31。
- 八进制, 数字 0 开始的数值表示八进制数, 如 011 表示十进制的 9。

● 二进制,使用 0b 开始的数值表示二进制数,如 0b1111 表示十进制的 15。 下面的代码演示了不同进制数值的字面量。

<?php echo 15,"
"; echo 0x1F,"
"; echo 011, "
"; echo Ob1111; ?>

代码执行结果见图 3-1,所有数字都以对应的十进制数输出。



图 3-1

PHP 中的算术运算包括:

- 加法运算, 使用+运算符。
- 减法运算,使用 -运算符。
- 乘法运算, 使用*运算符。
- 除法运算, 使用 / 运算符。如果不能整除, 会返回包含小数部分的结果 (浮点数)。
- 模运算,又称为取余数运算。使用%运算符,用于计算整数相除的余数,如果运算数不是整数,则自动转换为整数后计算。
- ** 运算符, 求某数的次方, 如 2**3 的结果为 8。这是 PHP 5.6 新增的运算符。

18

网站全栈开发指南:HTML+CSS+JavaScript+PHP

下面的代码演示了这些算术运算的应用。

 \rightarrow

```
<?php

$x = 5;

$y = 3;

echo "$x + $y = ",$x+$y,"<br>";

echo "$x - $y = ",$x-$y,"<br>";

echo "$x * $y = ",$x*$y,"<br>";

echo "$x / $y = ",$x/$y,"<br>";

echo "$x % $y = ",$x$$y,"<br>";

echo "$x ** $y = ",$x*$y,"<br>";

echo "$x ** $y = ",$x*$y,"<br>";
```

代码执行结果见图 3-2。



图 3-2

数值在运算过程中,可能会出现无效的结果,此时,需要关注一个特殊值,即 NaN, 它表示"不是一个数值"(not a number)。判断 NaN 值时可以使用 is_nan()函数,如 is_ nan(\$x) 返回 true,表示 \$x 不是一个数值,并且不能转换为数值。

下面的代码使用 is_nan() 函数测试字符串内容是否可以转换为数值, 然后使用 intval() 函数将字符串转换为整数。

```
<?php

$s = "123";

var_dump(is_nan($s));

echo "<br>";

$x = intval($s);

var_dump($x);

?>
```

代码执行结果见图 3-3。



图 3-3

接下来介绍几种整数特有的操作。

3.1.1 增量与减量运算

和很多 C 风格编程语言一样, PHP 中也包括增量运算和减量运算, 其原理也是相同的, 下面先来看增量运算。

增量运算又分为前增量和后增量,它们的区别在于表达式的值。前增量运算时,变量会先执行加1操作,然后返回表达式的值,这样,表达式和变量的值都是原值加1,如下面的代码。

```
<?php
$i = 1;
echo ++$i, "<br />", $i;
?>
```

执行代码会显示两个 2,即表达式和变量的值都是 \$i 加 1 后的结果。 后增量运算时,表达式会先返回变量的值,然后进行变量加 1 的操作,如下面的代码。

```
<?php
$i = 1;
echo $i++, "<br />", $i;
?>
```

执行代码会显示1和2,即表达式先返回 \$i 变量的值1,然后进行加1操作,最终 \$i 变量的值是2。

减量运算同样分为前减量运算和后减量运算,只不过它执行的是减1的操作,如下面的 代码演示了后减量运算的应用。

```
<?php
$i = 1;
echo $i--,"<br>";
echo $i;
?>
```

执行代码会显示1和0,即\$i--表达式会先返回1,然后进行\$i变量减1的操作。

3.1.2 位运算

位运算是对整数二进制位进行运算。PHP 中的位运算包括:

- 按位与运算,使用 & 运算符,两个二进制位上的值都是 1 时,运算结果为 1,否则 为 0。
- 按位或运算,使用 | 运算符,两个二进制位上的值有一个是1时,运算结果为1,两 个数据都是0时,运算结果为0。
- 按位取反运算, 使用~运算符, 1取反得0, 0取反得1。
- 按位异或运算,使用 ^ 运算符,两个二进制位上的值一个为 1,另一个为 0 时,结果 为 1,两个数据相同时,结果为 0。
- 位左移运算,使用 << 运算符,如 16<<2 就是将 16 的二进制位向左(高位)移动 2
 位,低位补 0,相当于执行 16*2*2 的操作,可以理解为 x<<n 执行了 x=x × 2ⁿ运算。
- 位右移运算,使用 >> 运算符,将数据的二进制位向右(低位)移动,高位补 0。如 x>>n 执行的就是 x=x ÷ 2ⁿ运算。

下面的代码显示了这些位运算的操作。

```
<?php
$x = 0b0110;
$y = 0b1010;
echo decbin(($x & $y)),"<br />";
```

```
20 网站全栈开发指南: HTML+CSS+JavaScript+PHP
```

```
echo decbin(($x | $y)),"<br />";
echo decbin((~$y)),"<br />";
echo decbin(($x ^ $y)),"<br />";
echo decbin(($x << 2)),"<br />";
echo decbin(($x >> 2));
?>
```

代码执行结果见图 3-4。



图 3-4

结果并不难理解,只是第三行按位取反的结果需要解释下,大家可以数数有几位数。64 位!暴露作者使用的平台了。在64位平台上,整数是64位的,那么1010前面的60位是 60个0,取反后是60个1,如图3-4中显示的结果。

3.1.3 进制转换

需要整数不同进制的形式时,可以使用数学模块中的相关函数,如:

- base_convert() 函数,任意进制之间的转换。参数一指定需要转换的数据(字符串类型);参数二指定源数据进制(整数类型);参数三指定目标进制(整数类型)。函数会返回转换后的结果(字符串类型)。
- decbin() 函数, 十进制数转换为二进制数。
- decoct() 函数, 十进制数转换为八进制数。
- dechex() 函数, 十进制数转换为十六进制数。
- bindec() 函数,二进制数转换为十进制数。
- octdec() 函数, 八进制数转换为十进制数。
- hexdec() 函数, 十六进制数转换为十进制数。
- bin2hex() 函数,二进制数转换为十六进制数。
- hex2bin() 函数,十六进制数转换为二进制数。

下面的代码演示了 base_convert() 函数的使用。

```
<?php
echo base_convert("f1",16,2);
?>
```

本例中,将十六进制数的 f1 转换为二进制数形式,代码执行结果见图 3-5。



图 3-5

下面的代码演示了十进制数和十六进制数之间的转换,其中使用了 dechex()和 hexdec()函数。

```
<?php
echo dechex(128),"<br>";
echo hexdec("1F");
?>
```

代码执行结果见图 3-6。



图 3-6

3.2 组合运算符

组合运算符就是将一些基本的二元运算符与赋值运算符(=)组合使用,如下面的代码。

```
<?php
$x = 10;
$x += 99;
echo $x;
?>
```

执行代码会显示 109。

本例中使用了 += 运算符, \$x += 99 的含义是 \$x = \$x + 99。对于算术运算和位运算,都可以使用这种形式的组合运算符,如 +=、-=、*=、/=、%=、&=、|=、^=、<<<>>>=。

3.3 浮点数

浮点数(float)可以处理带有小数部分的数值。PHP中的浮点数采用 IEEE 754 标准的双 精度格式,使用 64 位存储。

浮点数同样可以进行基本的加、减、乘、除等算术运算,下面的代码演示了相关运算。

```
<?php
$x = 5.1;
$y = 2.0;
echo "$x + $y = ",$x+$y,"<br>";
echo "$x - $y = ",$x-$y,"<br>";
echo "$x * $y = ",$x*$y,"<br>";
echo "$x / $y = ",$x/$y,"<br>";
echo "$x % $y = ",$x%$y,"<br>";
echo "$x ** $y = ",$x*$y,"<br>";
echo "$x ** $y = ",$x**$y,"<br>";
echo "$x ** $y = ",$x**$y,"<br<";
echo "$x ** $y = ",$x** $y = ",$x** $y,"<br<";
echo "$x ** $y = ",$x** $y = ",$x**
```

代码执行结果见图 3-7。