



普通高等教育“十三五”规划教材
高等学校**计算机**教材

PHP

实用教程 (第3版)

郑阿奇 / 主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

“十三五”规划教材

教材

PHP 实用教程

(第3版)

郑阿奇 主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书以 PHP 7 为平台, 介绍内容包含实用教程、实验指导、综合应用实习和附录 4 个部分, 涵盖了理论和实践教学的全过程。实用教程部分包括 HTML+CSS 基础知识、PHP 开发环境、PHP 基础语法、PHP 数组与字符串、PHP 常用功能模块、PHP 面向对象程序设计、构建 PHP 互动网页、数据库基础、使用 PHP 操作数据库、使用 PDO 访问数据库、PHP 与 AJAX 等; 实验指导部分着重训练学生的动手能力; 综合应用实习部分介绍 PHP/MySQL 学生成绩管理系统。本书配有电子课件、书中所有实例程序源代码、综合应用实习源文件, 可从华信教育资源网 <http://www.hxedu.com.cn> 免费注册下载。书中所有实例程序均上机调试通过。

本书可作为大学本科和高职高专相关课程教材和教学参考书, 也可供从事 PHP 应用系统开发的用户学习和参考。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

PHP 实用教程 / 郑阿奇主编. —3 版. —北京: 电子工业出版社, 2019.1
ISBN 978-7-121-34882-2

I. ①P… II. ①郑… III. ①PHP 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2018) 第 185818 号

策划编辑: 戴晨辰

责任编辑: 张 京 特约编辑: 张燕虹

印 刷: 北京京师印务有限公司

装 订: 北京京师印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 25 字数: 739 千字

版 次: 2009 年 8 月第 1 版

2019 年 1 月第 3 版

印 次: 2019 年 1 月第 1 次印刷

定 价: 62.00 元



凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: dcc@phei.com.cn。

前 言

PHP 语言具有简单性、开放性、低成本、安全性和开源免费等诸多优点, 适用 Linux 和 Windows 平台, 是当今最流行的 Web 开发语言。

2009 年, 我们编写了《PHP 实用教程》, 2014 年出了第 2 版, 针对当时的 PHP 5 进行了系统的修改和完善, 一直受到读者的广泛好评, 累计重印 11 次。如今, PHP 已升级至 PHP 7, 我们也与时俱进地推出了《PHP 实用教程》(第 3 版), 以反映新时代 PHP 的崭新面貌。

《PHP 实用教程》(第 3 版) 在前两版的基础上, 主要进行了如下修改和完善:

(1) 仍然保持原书的基础性和系统性的优点, 基础知识部分程序全面使用当下流行的 HTML 5+CSS 3, 在 Windows 7 及以上平台上运行。

(2) PHP 学习和开发环境介绍分立组件和集成 WAMP 两种方式: ①以分立组件方式搭建 PHP 环境; ②采用最新 Apache+PHP+MySQL+Eclipse Oxygen 集成方式搭建 PHP 环境。WAMP (内含 Apache+MySQL+PHP) 也很好体现了 PHP 的最新成果。

(3) PHP 7 对 MySQL 的操作完全废弃了原生函数库, 转而使用改进后的 Mysqli 库取而代之, 本书针对 PHP 7 新的函数库接口全面更新了书中所有涉及连接和操作数据库的程序。上一版 PHP 5 对 MySQL 的操作方法仍然通过出版社网站以电子文档的方式提供。

(4) 除系统介绍使用 PHP 操作 MySQL 数据库的一般方法外, 还介绍了使用 PHP 操作当前主流的另外两大数据库——甲骨文的 Oracle、微软的 SQL Server。

(5) 介绍使用 PDO 访问各种异构数据库的统一方法, 并且通过 PDO 访问 SQL Server 2014 实例进行说明, 其他数据库模仿 PDO 方法即可。

(6) 对 PHP 其他内容及实例的编排进行了重新组织, 加入了反映 PHP 7 新特性的变化, 以及系统介绍了面向对象程序设计的方法。

本书包含实用教程、实验指导、综合应用实习和附录 4 个部分, 涵盖了理论和实践教学的全过程。

实用教程部分包括: HTML+CSS 基础知识、PHP 开发环境、PHP 基础语法、PHP 数组与字符串、PHP 常用功能模块、PHP 面向对象程序设计、构建 PHP 互动网页、数据库基础、使用 PHP 操作数据库、使用 PDO 访问数据库 (包括 SQL Server、Oracle、Sybase、PostgreSQL、DB2、SQLite 等)、PHP 与 AJAX 等。

实验指导部分是对实用教程内容的消化和实例的操作实践。综合应用实习旨在培养学生解决问题的能力, 读者可以进行模仿, 也可作为实际的应用参考。书中所有实例程序均上机调试通过, 通过阅读本书, 结合实验和综合应用实习, 读者就能在较短的时间内基本掌握 PHP 及其应用技术。

本书配有电子课件, 书中所有实例程序源代码、综合应用实习源文件, PHP 操作数据库实例包括以常规方法操作 MySQL、Oracle 和 SQL Server, 以及使用 PDO 操作 SQL Server, 可从华信教育资源网 <http://www.hxedu.com.cn> 免费注册下载。

本书由郑阿奇 (南京师范大学) 担任主编。参加本书编写的还有丁有和、曹弋、徐文胜、周何骏、孙德荣、樊晓青、郑进、刘建、刘忠、郑博琳等。还有其他一些同志对本书的编写提供了许多帮助, 在此一并表示感谢!

由于我们的水平有限, 疏漏和错误在所难免, 敬请广大师生、读者批评指正。

E-mail: easybooks@163.com。

编 者

目 录

第 1 部分 实用教程

第 1 章 HTML+CSS 基础知识 (1)

1.1 HTML 和 PHP 介绍 (1)

1.1.1 HTML 基本结构 (1)

1.1.2 HTML 基本说明 (4)

1.1.3 HTML、XHTML 和

HTML 5 (5)

1.1.4 PHP 语言简介 (5)

1.2 格式化 HTML 页面 (8)

1.2.1 设置文本格式 (8)

1.2.2 多媒体标记 (11)

1.2.3 表格的设置 (13)

1.3 表单 (15)

1.3.1 表单标记 (15)

1.3.2 表单输入控件 (16)

1.3.3 其他表单控件 (19)

1.4 超链接 (21)

1.5 设计框架 (23)

1.6 层叠样式表 (CSS) (25)

1.6.1 样式表定义 (26)

1.6.2 样式的继承和作用顺序 (30)

1.6.3 CSS 属性 (30)

1.6.4 CSS、CSS 3 (33)

1.6.5 HTML 5+CSS 3 (34)

习题 1 (35)

第 2 章 PHP 开发环境 (36)

2.1 PHP 分立组件环境搭建 (36)

2.1.1 组件的选择 (36)

2.1.2 环境的搭建 (37)

2.1.3 创建 PHP 项目 (47)

2.2 PHP 集成软件环境搭建 (50)

2.2.1 WampServer 简介 (50)

2.2.2 WampServer 安装 (51)

2.3 典型 PHP 程序实例 (53)

2.3.1 最简单的 PHP 程序 (53)

2.3.2 同一页面上的 PHP 交互 (54)

2.3.3 不同页面上的 PHP 交互 (55)

2.4 PHP 项目导入 (56)

习题 2 (57)

第 3 章 PHP 基础语法 (58)

3.1 PHP 入门 (58)

3.1.1 PHP 标记风格 (58)

3.1.2 PHP 程序注释 (59)

3.1.3 PHP 页面输出 (59)

3.1.4 HTML 嵌入 PHP (60)

3.1.5 PHP 使用 JavaScript (60)

3.2 数据类型 (61)

3.2.1 整型 (61)

3.2.2 浮点型 (62)

3.2.3 字符串 (62)

3.2.4 布尔型 (64)

3.2.5 数组和对象 (65)

3.2.6 NULL 类型 (65)

3.2.7 资源类型 (66)

3.2.8 伪类型 (66)

3.2.9 类型转换 (66)

3.3 变量与常量 (68)

3.3.1 自定义变量 (68)

3.3.2 可变变量 (71)

3.3.3 预定义变量 (71)

3.3.4 外部变量 (73)

3.3.5 常量 (75)

3.4 运算符与表达式 (76)

3.4.1 算术运算符 (76)

3.4.2 赋值运算符 (76)

3.4.3 位运算符 (77)

3.4.4 比较运算符 (78)

3.4.5	错误控制运算符	(78)	4.1.7	PHP 5.4 之后版本新增特性	(117)
3.4.6	执行运算符	(79)	4.1.8	实例——处理表格数据	(119)
3.4.7	递增/递减运算符	(79)	4.2	字符串操作	(121)
3.4.8	逻辑运算符	(80)	4.2.1	字符串的定义与显示	(121)
3.4.9	字符串运算符	(80)	4.2.2	字符串的格式化	(121)
3.4.10	其他运算符	(81)	4.2.3	常用的字符串函数	(122)
3.4.11	运算符优先级和结合性	(81)	4.2.4	字符串的比较	(124)
3.4.12	表达式	(82)	4.2.5	字符串的替换	(124)
3.5	程序流程控制	(83)	4.2.6	字符串与 HTML	(125)
3.5.1	条件控制语句	(83)	4.2.7	其他字符串函数	(126)
3.5.2	循环控制语句	(87)	4.2.8	实例——留言簿内容处理	(128)
3.5.3	流程控制符	(90)	4.3	正则表达式	(129)
3.5.4	流程控制的替代语法	(92)	4.3.1	基础知识	(129)
3.5.5	包含文件操作	(93)	4.3.2	PHP 中正则表达式的应用	(133)
3.5.6	declare 结构	(93)	4.3.3	实例——验证表单内容	(136)
3.6	PHP 函数	(94)	习题 4		(137)
3.6.1	用户自定义函数	(95)	第 5 章	PHP 常用功能模块	(138)
3.6.2	参数的传递	(95)	5.1	目录与文件操作	(138)
3.6.3	函数变量的作用域	(96)	5.1.1	目录操作	(138)
3.6.4	函数的返回值	(96)	5.1.2	文件的打开与关闭	(140)
3.6.5	函数的调用	(97)	5.1.3	文件的写入	(142)
3.6.6	递归函数	(98)	5.1.4	文件的读取	(144)
3.6.7	变量函数	(98)	5.1.5	文件的上传与下载	(148)
3.6.8	系统函数	(99)	5.1.6	其他常用文件函数	(151)
3.6.9	匿名函数	(99)	5.1.7	实例——投票统计	(153)
3.7	综合实例	(99)	5.2	图形处理	(155)
3.7.1	实例一——多项选择题	(99)	5.2.1	安装 PHP 图像库	(155)
3.7.2	实例二——计算器程序	(100)	5.2.2	创建图形	(156)
习题 3		(102)	5.2.3	绘制图形	(158)
第 4 章	PHP 数组与字符串	(103)	5.2.4	图形的处理	(166)
4.1	数组及处理	(103)	5.2.5	其他的图形函数	(170)
4.1.1	数组的创建和初始化	(103)	5.2.6	实例——自动生成验证码	(171)
4.1.2	键名和值的操作	(106)	5.3	日期和时间	(172)
4.1.3	数组的遍历和输出	(109)	5.3.1	UNIX 时间戳	(172)
4.1.4	数组的排序	(112)	5.3.2	时间转化为时间戳	(173)
4.1.5	数组的集合操作	(115)			
4.1.6	其他操作	(116)			

5.3.3 获取日期和时间	(173)	7.1.2 接收表单数据	(197)
5.3.4 其他日期和时间函数	(176)	7.1.3 常用表单数据的验证方法	(198)
5.3.5 实例——生成日历	(176)	7.1.4 实例——使用 PHP 处理表单数据	(198)
习题 5	(178)	7.2 URL 处理	(201)
第 6 章 PHP 面向对象程序设计	(179)	7.2.1 获取 URL 参数	(201)
6.1 基本概念	(179)	7.2.2 解析 URL	(201)
6.2 PHP 中的类	(180)	7.2.3 URL 编解码	(202)
6.2.1 创建类	(180)	7.3 页面跳转	(203)
6.2.2 类的属性和方法	(180)	7.3.1 使用 header() 函数	(203)
6.2.3 类的实例化	(181)	7.3.2 使用 HTML 标记	(203)
6.2.4 类的访问	(182)	7.3.3 使用客户端脚本	(204)
6.2.5 静态属性和方法	(182)	7.4 会话管理	(204)
6.2.6 构造函数和析构函数	(183)	7.4.1 会话的工作原理	(204)
6.3 类的继承	(184)	7.4.2 实现会话	(204)
6.3.1 子类访问父类	(184)	7.4.3 Cookie 技术	(207)
6.3.2 方法重载	(185)	7.5 实例——制作一个 PHP 互动网页	(211)
6.3.3 使用 final 关键字	(185)	习题 7	(216)
6.3.4 Traits 机制	(186)	第 8 章 数据库基础	(217)
6.4 抽象类与接口	(186)	8.1 数据库的基本概念	(217)
6.4.1 抽象类	(186)	8.1.1 数据库与数据库管理系统	(217)
6.4.2 接口	(187)	8.1.2 关系数据库管理系统	(218)
6.5 类的魔术方法	(189)	8.1.3 关系数据库语言	(220)
6.5.1 复制对象	(189)	8.2 MySQL 基础知识	(221)
6.5.2 方法重载	(189)	8.2.1 MySQL 简介	(221)
6.5.3 属性重载	(190)	8.2.2 MySQL 数据对象	(222)
6.5.4 字符串转换	(191)	8.2.3 MySQL 表结构	(223)
6.5.5 自动加载对象	(191)	8.2.4 MySQL 数据类型	(223)
6.5.6 对象序列化	(192)	8.3 数据库和表的建立与管理	(227)
6.5.7 对象调用	(192)	8.3.1 创建数据库	(227)
6.6 类型判断与引用	(193)	8.3.2 创建表	(228)
6.6.1 对象类型的判断	(193)	8.4 表数据操作	(231)
6.6.2 通过变量引用类	(193)	8.4.1 插入表数据	(231)
6.6.3 引用静态调用的类	(194)	8.4.2 修改表数据	(232)
6.7 实例——设计一个学生管理类	(195)	8.4.3 删除表数据	(233)
习题 6	(196)		
第 7 章 构建 PHP 互动网页	(197)		
7.1 PHP 与表单	(197)		
7.1.1 提交表单数据	(197)		

8.5 数据查询.....	(234)	9.5.1 使用 PHP 操作 Oracle.....	(266)
8.5.1 选择列.....	(234)	9.5.2 使用 PHP 操作 SQL Server.....	(273)
8.5.2 选择查询对象.....	(235)	习题 9.....	(281)
8.5.3 指定查询条件.....	(236)	第 10 章 使用 PDO 访问数据库.....	(282)
8.5.4 数据分组.....	(243)	10.1 PDO 的基本概念.....	(282)
8.5.5 排序.....	(246)	10.2 使用 PDO 访问异构数据库的 方法.....	(283)
8.5.6 限制返回行数.....	(247)	10.2.1 安装 PDO.....	(284)
8.6 视图.....	(247)	10.2.2 创建 PDO 对象.....	(284)
8.6.1 视图的概念.....	(247)	10.2.3 使用 PDO.....	(284)
8.6.2 创建视图.....	(248)	10.3 使用 PDO 访问数据库实例.....	(285)
8.6.3 查询视图.....	(249)	10.3.1 SQL Server 2014 的 PDO 驱动.....	(285)
8.6.4 删除视图.....	(249)	10.3.2 实例——实现简单的登录 系统.....	(285)
8.7 过程式数据库对象.....	(250)	习题 10.....	(293)
8.7.1 存储过程.....	(250)	第 11 章 PHP 与 AJAX.....	(294)
8.7.2 存储函数.....	(252)	11.1 AJAX 基础.....	(294)
8.7.3 触发器.....	(253)	11.1.1 AJAX 的工作原理.....	(294)
习题 8.....	(254)	11.1.2 AJAX 初始化.....	(295)
第 9 章 使用 PHP 操作数据库.....	(256)	11.1.3 发送 HTTP 请求.....	(296)
9.1 使用 PHP 访问 MySQL 的基本 流程.....	(256)	11.1.4 指定响应处理函数.....	(296)
9.2 连接 MySQL 服务器.....	(257)	11.1.5 处理服务器返回的信息.....	(296)
9.2.1 建立连接.....	(257)	11.2 PHP 与 AJAX 交互.....	(297)
9.2.2 选择数据库.....	(257)	11.2.1 使用 GET 方法.....	(297)
9.2.3 关闭连接.....	(258)	11.2.2 使用 POST 方法.....	(299)
9.3 操作 MySQL 数据库.....	(258)	11.2.3 简单实例——AJAX 的 应用.....	(302)
9.3.1 执行 SQL 语句.....	(258)	习题 11.....	(304)
9.3.2 处理结果集.....	(259)	第 2 部分 实验指导	
9.3.3 其他 MySQL 函数.....	(261)	实验 1 使用 HTML 制作网页.....	(305)
9.4 实例——操作课程表.....	(262)	实验 2 PHP 的开发环境.....	(310)
9.5 使用 PHP 操作其他数据库.....	(266)	实验 3 PHP 基础语法.....	(313)
		实验 4 PHP 数组与字符串.....	(318)
		实验 5 PHP 常用功能模块.....	(324)
		实验 6 PHP 面向对象.....	(329)
		实验 7 PHP 互动网页.....	(331)
		实验 8 MySQL 数据库.....	(336)
		实验 9 使用 PHP 操作 MySQL.....	(348)
		实验 10 PHP 与 AJAX.....	(352)

第 3 部分 综合应用实习

实习 PHP/MySQL 学生成绩管理系统.....	(356)	P.4 学生信息查询	(368)
P.1 数据库的创建	(356)	P.5 成绩信息录入	(374)
P.2 程序主界面	(359)	P.6 学生成绩查询	(379)
P.3 学生信息录入	(362)		

第 4 部分 附 录

附录 A PHP 程序调试与异常处理.....	(383)	A.1.3 PHP 错误调试方法.....	(385)
A.1 程序调试	(383)	A.2 异常处理.....	(385)
A.1.1 常见的编程错误	(383)	A.2.1 异常处理类	(385)
A.1.2 PHP 错误报告管理.....	(384)	A.2.2 PHP 的异常处理方法.....	(386)

第1部分 实用教程

第 1 章

HTML+CSS 基础知识

PHP 是开发网页的语言，故在学习它之前，读者必须先具备一定的网页设计和开发的基础知识，当前的网页开发主要使用 HTML 与 CSS 样式相结合的方式来进行。

HTML 的英文全称是 Hypertext Marked Language，译为中文为“超文本标记语言”。

和一般文本不同的是，一个 HTML 文件不仅包含文本内容，还包含一些 Tag，中文称为“标记”。标记是描述性的，用一对中间包含若干字符的“<>”表示，通常是成对出现的，前一个是起始标记，后一个为结束标记。

一个 HTML 文件的后缀名是.htm 或者.html。用文本编辑器就可以编写 HTML 文件。

1.1 HTML 和 PHP 介绍

1.1.1 HTML 基本结构

HTML 文档包括文档头和文档主体，其基本结构如下：

```
<html>
```

```
<head>
```

文档头部分

```
</head>
```

```
<body>
```

文档主体部分

```
</body>
```

```
</html>
```

基本 HTML 页面从<html>标记开始，到</html>标记结束。它们之间是文档头部分和文档主体部分。文档主体部分用<head>...</head>标记界定，一般包含网页标题、文档属性参数等不在页面上显示的元素。文档主体部分是网页的主体，内容均会反映在页面上，用<body>....</body>标记来界定，主要包括描述网页的文字、表格、图像、动画、超链接等内容。

【例 1.1】 使用 HTML 设计一个简单的网页。

(1) 打开 Windows 的附件中的记事本功能, 输入下列内容, 以 `hello.htm` 作为文件名保存到 `C:\wamp\www` 文件夹中。本书的所有实例文件均保存在该文件夹中。

```
<html>
<head>
<title>一个 Hello 网页</title>
<script language="JavaScript">
    function myp()
    {
        alert("大家好!");
    }
</script>
</head>
<body bgcolor="#8888FF" onload="myp()">
<div align=center>
<br/><br/>
<h2>
学生成绩管理系统
</h2>
</div>
</body>
</html>
```

(2) 在浏览器地址栏中输入 `C:\wamp\www\hello.htm`; 或者配置 IIS 后将当前的虚拟根目录设置为 `"C:\wamp\www"`, 然后在浏览器地址栏中输入 `http://127.0.0.1/hello.htm`。其中, `127.0.0.1` 代表本机虚拟根目录。

用浏览器打开文档, 将显示如图 1.1 所示的页面。

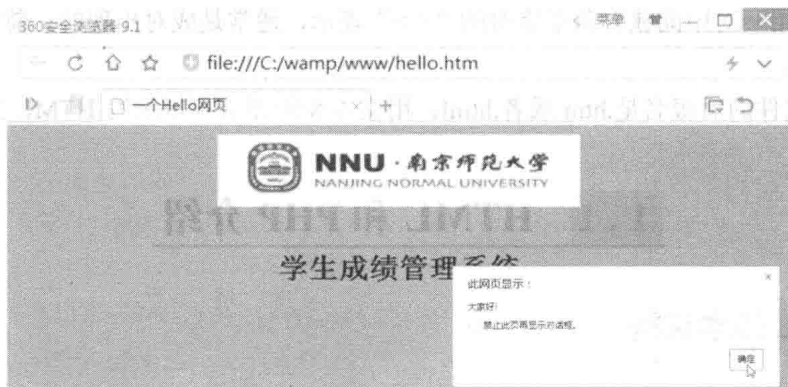


图 1.1 浏览显示结果

在 HTML 的所有标记中, 许多标记还有若干属性, 通过设置属性值, 可对标记内的内容进行控制 (如上例中的 `<script>`、`<body>`、`<div>`、`<img>` 标记)。如果不设置这些标记的属性值 (如上例中的 `<h2>` 标记), 则使用系统的默认属性值。

有些标记 (如上例中的 `<body>` 标记) 还有一些事件, 通过设置事件代码, 当该事件产生时, 事件代码便被执行。事件代码用脚本语言编写, 目前, 常用的脚本语言为 JavaScript。用脚本语言编写的程序以 `<script>` 标记括起, `language` 属性告知浏览器以 `<script>` 标记括起的脚本是用什么脚本语言编写的。

1. 文档头描述

文档头部分处于 `<head>` 与 `</head>` 标记之间, 在文档头部分一般可以使用以下几种标记。

(1) 指定网页的标题: `<title>` 和 `</title>`。

指定的网页标题在浏览器顶端的标题栏中显示, 搜索引擎通过标题能够搜索到该网页。

(2) 指定文档内容的样式表: `<style>`和`</style>`。

样式包括字体、颜色、格式等。在文档头部分定义了样式表后,就可以在文档主体部分引用样式表。

(3) 注释: `<!--`和`-->`。

这两个标记之间的内容为 HTML 的注释部分,是网页设计人员的说明内容,浏览器不做任何处理。

(4) 描述网页文档的属性: `<meta>`。

描述标记的格式为`<meta 属性="值"...>`,常用的属性有 `name`、`content` 和 `http-equiv`。

`name` 为 `meta` 的名字; `content` 为页面的内容; `http-equiv` 为 `content` 属性的类别, `http-equiv` 取不同值时, `content` 表示的内容也不一样。

(5) 脚本语言程序: `<script>`和`</script>`。

在这两个标记之间可以插入客户端脚本语言程序,例如:

```
<script language="JavaScript">
    alert("大家好!");
</script>
```

以上代码表示插入的是 JavaScript 脚本语言。

2. 文档正文标记

`<body>`和`</body>`是文档正文标记,文档的主体部分就处于这两个标记之间。`<body>`标记中还可以定义文档主体的一些属性,格式如下:

```
<body 属性="值"... 事件="执行的程序"...>
```

(1) `<body>`标记属性。

`<body>`标记常用的属性如下。

- 文档背景图片: `background`。

例如:

```
<body background="back-ground.gif">
```

表示文档背景图片名称为 `back-ground.gif`,上面代码中没有给出图片所在的位置,则表示图片和文档文件在同一文件夹下,如果图片和文档文件不在同一位置,则需要给出图片的路径,例如:

```
<body background=" C:/wamp/www/image/back-ground.gif">
```

说明:在指定文件位置时,为防止与转义符“\”混淆,一般使用“/”来代替“\”。

- 文档的背景颜色: `bgcolor`。

例如:

```
<body bgcolor="red">
```

表示文档的背景颜色为红色。

- 文档中文本的颜色: `text`。

例如:

```
<body text="blue">
```

表示文档中文字的颜色都为蓝色。

- 文档中链接的颜色: `link`。

- 文档中已被访问过的链接的颜色: `vlink`。

- 文档中正在被选中的链接的颜色: `alink`。

(2) `<body>`标记事件。

`<body>`标记中常用事件有 `onload` 和 `onunload`。

- `onload` 表示文档首次加载时调用的事件处理程序。

- `onunload` 表示文档卸载时调用的事件处理程序。

1.1.2 HTML 基本说明

HTML 中常用下列描述。

1. 颜色

许多标记也用到了颜色属性，颜色值一般用颜色名称或十六进制数来表示。

(1) 使用颜色名称来表示。例如，红色、绿色和蓝色分别用 red、green 和 blue 表示。

(2) 使用十六进制格式数值#RRGGBB 来表示，RR、GG 和 BB 分别是表示颜色中的红、绿、蓝三原色的两位十六进制数。例如，红色、绿色和蓝色分别用#FF0000、#00FF00 和#0000FF 表示。表 1.1 列出了 16 种标准颜色的名称及其十六进制数。

表 1.1 16 种标准颜色的名称及其十六进制数

颜 色	名 称	十六进制数	颜 色	名 称	十六进制数
淡蓝	aqua(cyan)	#00FFFF	海蓝	navy	#000080
黑	black	#000000	橄榄色	olive	#808000
蓝	blue	#0000FF	紫	purple	#800080
紫红	fuchsia(magenta)	#FF00FF	红	red	#FF0000
灰	gray	#808080	银色	silver	#C0C0C0
绿	green	#008000	淡青	teal	#008080
橙	lime	#00FF00	白	white	#FFFFFF
褐红	maroon	#800000	黄	yellow	#FFFF00

2. HTML 字符实体

有些字符在 HTML 里有特别的含义，如小于号 (<) 就表示 HTML 标记的开始，这个小于号是不显示在网页中的。如果我们希望在网页中显示一个小于号，就要涉及 HTML 字符实体。

一个字符实体以&符号打头后跟实体名字或者是#加上实体编号，最后是一个分号。最常用的字符实体如表 1.2 所示。

表 1.2 最常用的字符实体

显示结果	说 明	实 体 名	实 体 号
	显示一个空格	 	
<	小于	<	<
>	大于	>	>
&	&符号	&	&
"	双引号	"	"
©	版权	©	©
®	注册商标	®	®
×	乘号	×	×
÷	除号	÷	÷



注意：

并不是所有的浏览器都支持最新的字符实体名字。而采用字符实体编号后，各种浏览器都能处理。

字符实体是区分大小写的。

更多字符实体请参见 ISO Latin-1 字符集。

3. HTML 标记常用属性

有些属性在 HTML 许多标记中出现, 下面分别说明。

(1) 类名: class。

(2) 唯一标志: id。

(3) 内样式: style。

(4) 提示信息: title。

4. HTML 标记常用事件

事件处理描述是一个或一系列以分号隔开的 JavaScript 表达式、方法和函数调用, 并用引号引起来。当事件发生时, 浏览器会执行这些代码。

事件包括窗口事件、表单及其元素事件、键盘事件、鼠标事件。

1.1.3 HTML、XHTML 和 HTML 5

W3C 成立之前, 在 HTML 的早期发展中, 很多标准的制定都是在浏览器的开发者们互相讨论的情况下完成的, 如从 HTML 2.0、3.2 到 4.0、4.01, 这些标准大部分都是先有实现后有标准。在这种情况下, HTML 标准不是很规范的, 浏览器也对 HTML 页面中的错误相当宽容。

W3C 随后意识到了这个问题, 并认为这是互联网的一个基础性问题, 应该加以解决。为了规范 HTML, W3C 结合 XML 制定了 XHTML 1.0 标准, 这个标准没有增加任何新的标记, 只是按照 XML 的要求来规范 HTML, 并定义了一个新的 MIME 类型。W3C 的初衷是对这个 MIME 类型浏览器要实行强错误检查, 即如果页面有 HTML 错误, 就要显示错误信息。但是由于 Web 页面中已经有了大量的错误, 很多开发者拒绝使用新的 MIME 类型。XHTML 2.0 竟然与原先的 HTML 不再兼容, 浏览器厂商终于愤怒了, Mozilla 和苹果公司牵头成立了 WHATWG 小组。失去了厂商支持的 W3C 的 XHTML 2 标准很快就成为一个笑话。

有了 XHTML 的教训, WHATWG 和 W3C 在制定下一代 HTML 标准即 HTML 5 时, 就将向后兼容作为一个很重要的原则。虽然 HTML 5 引入了许多新的特性, 但仍可以将任何已有的网页的第一行改成 `<!DOCTYPE html>`, 它就成为一个 HTML 5 页面, 并且可以在浏览器里正常展示。

例如, 以下代码可以认为是 HTML 5 文档。

```
<!DOCTYPE html>
<html>
<body>
    HTML 文本框
    <input type="text" value="输入初值">
</body>
</html>
```

1.1.4 PHP 语言简介

1. PHP 发展史

PHP 是“PHP: Hypertext Preprocessor”的缩写。

PHP 语法与 C 语言相似, 可运行在 Apache、Microsoft Internet Information Server (IIS) 和 iPlanet 等 Web 服务器上。PHP 作为一种工具, 可以创建动态 Web 页面。应用 PHP 的网页与常规的 HTML 页面并无不同, 用户可以用同样的方式来创建和编辑它们。PHP 允许直接在 HTML 代码中写入简单

的脚本,这一点与 JavaScript 非常相似。而不同的是,PHP 不依赖于浏览器,它是服务器端的语言,而 JavaScript 却是一种客户端的嵌在 HTML 中的语言。概念上,PHP 与 Netscape 的 LiveWire Pro 产品、Microsoft 的 ASP 及原 Sun 公司的 JSP 相似。

PHP 最初是由丹麦的 Rasmus Lerdorf 创建的。刚开始,它只是一个简单的用 Perl 语言编写的程序,用来统计网站的访问量;后来又用 C 语言重新编写,添加访问数据库的功能。1995 年,Lerdorf 以 Personal Home Page Tools (PHP Tools) 开始对外发布第一个版本,Lerdorf 写了一些介绍此程序的文档,并且发布了 PHP 1.0。在早期的 PHP 版本中,它提供了访客留言本、访客计数器简单功能。以后越来越多的网站使用了 PHP,并且强烈要求增加一些特性,比如循环语句和数组变量等。

1995 年,PHP 2.0 发布。第 2 版定名为 PHP/FI (Form Interpreter)。PHP/FI 加入了对 MySQL 数据库的支持,从此建立了 PHP 在动态网页开发上的地位。到 1996 年年底,有 15 000 个网站使用了 PHP/FI。而到了 1997 年,使用 PHP/FI 的网站超过 50 000 个。

1997 年,Rasmus Lerdorf 开始了第 3 版的开发计划,Zeev Suraski 和 Andi Gutmans 加入了开发小组,第 3 版定名为 PHP 3。

2000 年,PHP 4.0 问世了,它增加了许多新的特性。它使用了 Zend 引擎,提供了更高的性能,同时还包含了其他一些关键功能,如支持更多的服务器、支持 HTTP Session、输出缓存、更安全的处理用户输入的方法及一些新的语言结构。

2004 年,PHP 5.0 发布,它是一个里程碑式的版本,相比之前的 PHP 做了以下革命性的升级:首先,完全实现面向对象,提供名为 PHP 兼容模式的功能;其次是 XML 功能,支持可直观地访问 XML 数据、名为 SimpleXML 的 XML 处理界面,同时还强化了 XML Web 服务支持,且支持 SOAP 扩展模块;在数据库方面,PHP 5.0 提供旨在访问 MySQL 的新界面——MySQLi。除之前的界面外,用户还可以使用面向对象界面和预处理语句 (Prepared Statement) 等 MySQL 的新功能。另外,PHP 5.0 上还捆绑有小容量 RDBMS——SQLite。

2007 年在巴黎举行的 PHP 开发者大会提出了开发 PHP 6 的草案,原计划将全面支持 Unicode,并移除旧版 PHP 中很多过时的功能和特性,但由于解决 Unicode 问题的难度远超当初的设想,PHP 6.0 迟迟未能发布,而草案中所述 PHP 6 的其他主要特性在随后的几年中陆续皆由 PHP 5.3~5.6 提前实现了,最终导致 PHP 6 “流产”。因此,PHP 版本的升级由 5 系列直接跨越到 7 (跳过了 6)。

2015 年 6 月 11 日,PHP 官网正式公开发布 PHP 7 第 1 版的 alpha 版本,它使用新版的 ZendEngine 引擎,带来了许多新的特性:首先是性能上的提升,PHP 7.0 比 PHP 5.6 性能提升了 2 倍;其次是提供了全面一致的 64 位支持;另外还丰富了一些新的语言特性。目前最新的版本为 PHP 7.2.3,本书的所有程序都在这个版本上编写实现。

2. PHP 语言功能

PHP 作为一种被广泛使用的开放源代码多用途脚本语言,尤其适用于 Web 开发,并可以嵌入到 HTML 中。其语法利用 C、Java 和 Perl,非常容易学习。该语言的主要目标是让 Web 开发人员可以很快地写出动态生成的网页,但 PHP 的功能远不止如此。

PHP 与 HTML 语言有着非常好的兼容性,用户可以直接在 PHP 脚本代码中加入 HTML 标记,或者在 HTML 语言中嵌入 PHP 代码,从而更好地实现页面控制。PHP 提供了标准的数据接口,数据库连接十分方便,兼容性好,扩展性好,可以进行面向对象编程。

PHP 脚本主要用于以下 3 个领域。

- 服务端脚本。这是 PHP 最传统、最主要的目标领域。开展这项工作需要具备以下 3 个设备:PHP 解析器 (CGI 或服务模块)、Web 服务器和 Web 浏览器。需要在运行 Web 服务器时,安装并配置 PHP,然后可以用 Web 浏览器来访问 PHP 程序的输出,即浏览服务端的 PHP 页面。

- 命令行脚本。用户可以编写一段 PHP 脚本，并且不需要任何服务器或浏览器来运行它。通过这种方式，只需要用 PHP 解析器来执行。这种用法对于依赖 cron（UNIX 或 Linux 环境）或者 Task Scheduler（Windows 环境）的脚本来说是理想的选择。这些脚本也可以处理简单的文本。
- 编写桌面应用程序。对于有着图形界面的桌面应用程序来说，PHP 或许不是一种最好的语言，但是如果用户非常精通 PHP，并且希望在客户端应用程序中使用 PHP 的一些高级特性，可以利用 PHP-GTK 来编写这些程序。用这种方法还可以编写跨平台的应用程序。PHP-GTK 是 PHP 的一个扩展，在通常发布的 PHP 包中并不包含它。

3. PHP 语言特点

PHP 作为一种服务器端的脚本语言，它的特点主要有以下 8 个方面。

（1）开放源代码。

PHP 属于自由软件，是完全免费的，用户可以从 PHP 官方网站（<http://php.net/>）自由下载，而且可以不受限制地获得源码，甚至可以从中加进自己需要的特色。

（2）基于服务端。

PHP 是运行在服务器上的，充分利用了服务器的性能，PHP 的运行速度只与服务器的速度有关，因此它的运行速度可以非常快；PHP 执行引擎还会将用户经常访问的 PHP 程序驻留在内存中，其他用户再一次访问这个程序时就不需要重新编译了，只要直接执行内存中的代码即可，这也是 PHP 高效性的体现之一。

（3）数据库支持。

PHP 能够支持目前绝大多数的数据库，如 MySQL、Microsoft SQL Server、Oracle、PostgreSQL 等，并完全支持 ODBC，即 Open Database Connection Standard（开放数据库连接标准），因此可以连接任何支持该标准的数据库。其中，PHP 与 MySQL 是绝佳的组合，它们的组合可以跨平台运行。

（4）跨平台。

PHP 可以在目前所有主流的操作系统上运行，包括 Linux、UNIX 的各种变种、Microsoft Windows、Mac OS X、RISC OS 等。正是由于这个特点，使 UNIX/Linux 操作系统上有了一种与 ASP 媲美的开发语言。另外，PHP 已经支持了大多数的 Web 服务器，包括 Apache、IIS、iPlanet、Personal Web Server（PWS）、Oreilly Website Pro Server 等。对于大多数服务器，PHP 均提供了一个相应模块。

（5）易于学习。

PHP 的语法接近 C、Java 和 Perl，学习起来非常简单，而且拥有很多学习资料。PHP 还提供数量巨大的系统函数集，用户只要调用一个函数就可以完成很复杂的功能，编程时十分方便。因此，用户只需要很少的编程知识就能够使用 PHP 建立一个交互的 Web 站点。

（6）网络应用。

PHP 还提供强大的网络应用功能，支持 LDAP、IMAP、SNMP、NNTP、POP3、HTTP、COM（Windows 环境）等协议服务。它还可以开放原始端口，使任何其他协议能够协同工作，PHP 也可以编写发送电子邮件、FTP 上传/下载等网络应用程序。

（7）安全性。

由于 PHP 本身的代码开放，所以它的代码由许多工程师进行了检测，同时它与 Apache 编译在一起的方式也让它具有灵活的安全设定。因此，到现在为止，PHP 具有公认的安全性。

（8）其他特性。

PHP 还提供其他编程语言所能提供的功能，如数字运算、时间处理、文件系统、字符串处理等。除此之外，PHP 还提供更多的支持，包括高精度计算、公历转换、图形处理、编码与解码、压缩文件处理及有效的文本处理功能（如正则表达式、XML 解析等）。

4. PHP 语言发展趋势

PHP 目前在开发语言排行榜 (TIOBE) 排名第 6 位, 仅次于 Java、C、C++、C# 和 Python。同时, PHP 也是世界上使用率最高的网页开发语言。

目前, 新版 PHP 7 依然在稳步发展中, 越来越完善, 功能也更加强大。跨平台的使用特性将使它在 Linux 平台上大放异彩。可以直接使用 Java 的类库, 可以直接调用以 Perl、C 等语言编写的程序, 这增强了它的可扩展性。越来越成熟的 MVC 开发框架使它能适应企业级的大型应用开发, 再加上它天生强大的数据库支持能力, PHP 将会得到更多 Web 开发者的青睐。

1.2 格式化 HTML 页面

HTML 页面中显示的内容都是在文档的主体部分即 `<body>` 和 `</body>` 标记之间定义的。文档主体部分能够定义文本、图像、表格、表单、超链接和框架等。

1.2.1 设置文本格式

文本是 HTML 网页的重要内容。编写 HTML 文档时, 可以将文本放在标记之间来设置文本的格式。文本格式包括分段与换行、段落对齐方式、字体、字号、文本颜色及字符样式等。

1. 分段标记

格式如下:

```
<p 属性="值"...>...</p>
```

段落是文档的基本信息单位, 在 HTML 文档中原有的回车和换行均被忽略, 利用段落标记可以定义一个新段落, 或换行并插入一个空格。

单独用 `<p>` 标记时会空一行, 使后续内容隔行显示。同时使用 `<p>` 和 `</p>` 标记则将段落包围起来, 表示一个分段的块。

`align` 属性定义段落的水平对齐方式。其取值可以是 `left` (左对齐)、`center` (居中)、`right` (右对齐) 和 `justify` (两端对齐)。当该属性省略时则使用默认值 `left`。例如:

```
<p align="center">分段标记演示</p>
```

在下面的标记中也会经常使用到 `align` 属性。

2. 换行标记

换行标记为 `<br/>`, 该标记将强行中断当前行, 使后续内容在下一行显示。

3. 标题标记

格式如下:

```
<h1 属性="值">...</h1>
```

其中, `h1` 可以是 `h1`、`h2`、`h3`、`h4`、`h5` 和 `h6`, 都表示黑体, `h1` 表示字体最大, `h6` 表示字体最小。标题标记的常用属性也是 `align`, 与分段标记类似。

4. 对中标记

格式如下:

```
<center>...</center>
```

对中标记的作用是将标记中间的内容全部居中。

5. 块标记

格式如下: