

第1章 PHP起步

- 欢迎来到PHP的世界！
- PHP语言是最受欢迎的Web开发语言之一。它以学习简单、开发快速、性能稳定而倍受Web开发人员的青睐。PHP不仅使用人员众多、数以万计的Web站点用它构建，而且有强大的社区支持，使得无论是用PHP开发Web应用，还是学习PHP语言，都会快速有效、事半功倍。在各种Web开发语言、框架、概念纷扰的今天，PHP仍以其独特魅力吸引更多Web开发人员学习和使用。
- 选择PHP，没错的！



1.1 初识PHP

- PHP是一种服务器端的、嵌入HTML的脚本语言。它的语法结构和C语言极为相似。为了能够快速编写动态 Web页面，PHP还加入了自己的某些语言特征，这些特征都非常容易理解和使用。先来看一个嵌入有 PHP代码的HTML文档。
- `<html>`
- `<head><title>Welcome</title></head>`
- `<body>`
- `<?php`
- `echo "Welcome to PHP's world!";`
- `?>`
- `</body>`
- `</html>`



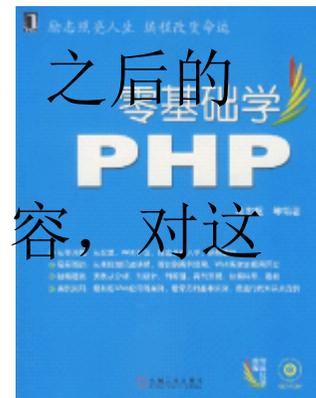
1.2 LAMP——锋利四剑客：Linux、Apache、MySQL和PHP

- 要想学习PHP，就有必要了解和PHP关系密切的其他3种技术：Linux操作系统、Apache网络服务器和MySQL数据库。
- LAMP这个名词最早由Michael Kunze创造，用来代表Linux操作系统、Apache网络服务器、MySQL数据库和PHP（如果可以，Perl和Python也是不错的选择），LAMP正是这4种技术的首字母。
- PHP作为强有力的Web开发语言，和Linux、Apache、MySQL的支持是密不可分的。它们都是开源软件，并且有强有力的社区支持，它们的完美组合构成了当今Web开发世界中不可忽视的一极重要力量。而且，这支力量近年一直在不断地变得更加强大。虽然，这4种技术并不是专门被设计在一起工作的，但多年来，这些软件之间的兼容性不断完善，不仅完改善了个组件之间的协作，扩展出更多的功能，而且在目前几乎所有的Linux版本中都默认包含了这些产品，使得这些产品共同组成了一个强大的Web应用平台。



1.3 PHP Web编程的体系结构和基本内容

- 概括地说，任何Web编程体系，无外乎服务器端和客户端的程序开发。基于PHP的Web应用开发也不例外，其中客户端的开发包括使用HTML设计Web页面，使用样式表控制Web页面的显示效果，还需要客户端的脚本语言来控制浏览器的特效、验证HTML表单数据等，这些脚本语言包括JavaScript、VBScript等。服务器端的开发就需要掌握PHP语言和MySQL数据库的有关内容。
- 本节的各小节将对这些内容做简要说明，之后的各节将较为详细地介绍HTML、样式表和JavaScript，它们都是Web开发的基础内容，对这些内容不了解的PHP初学者有必要掌握。



1.3.1 了解PHP、HTML、层叠样式表（CSS）和JavaScript及其关系

- 作为服务器端的脚本语言，PHP多数情况下都是和HTML相互搭配来使用的。PHP用来完成和逻辑有关的动态内容。PHP程序执行的输出结果，通过HTML文档表现给用户。一般情况下，在HTML页面中需要输出数据的地方内嵌入PHP代码，这个页面也就成为了PHP程序。
- 在创建HTML页面时，都要考虑页面设计。如页面布局、页面颜色、字体、边距空白等。这些设置可以单独在页面中进行，但那样会使HTML页面变得臃肿，也不便于将来的维护。层叠样式表解决了这些问题，它允许页面设计人员在层叠样式表（即CSS）文件里设定页面表现效果，如字体大小、边距控制等。CSS使得HTML页面的表现更加丰富、美观，也更加容易维护。



1.3.2 HTML文档

- HTML文档就是使用HTML标记语言创建的文档。通过浏览器浏览的网页，几乎都是HTML文档，或者是由服务器端程序生成的HTML文档。下面就是一个HTML文档的代码。
- `<html>`
- `<head><title> HTML文档示例</title></head>`
- `<body>`
- `<h1>HTML语言</h1>`
- `<p>First Web Page</p>`
- `</body>`
- `</html>`



1.3.3 使用样式表实现页面效果

- 按照HTML的设计初衷，文档的内容、结构与格式是分离开来的。如果过多地在HTML文档中控制页面的显示效果，会使HTML文档的维护越来越难以维护，也违背了HTML的设计初衷。为了解决这个问题，引入了样式表的概念。一个简单的样式表如下所示。
- 说明：样式表负责控制页面的格式和显示效果，HTML文档负责控制结构和显示内容。
- ```
p{
```
- ```
font-size:10pt;
```
- ```
}
```
- 这个样式表定义的含义是，HTML文档中所有用<p>标签定义的内容，都将使用10pt大小的字体显示。关于样式表语法和如何使用样式表，将在后面做讲述。



## 1.3.4 客户端的响应

- Web页面作为客户端，有时需要直接对用户的请求作出响应。这主要是通过客户端脚本来实现，这些脚本语言包括JavaScript等。
- 可以在HTML页面中加入JavaScript脚本，这些脚本可以直接在浏览器里执行。这样，就可以达到在客户端响应用户请求的需求。一般这样的响应主要是验证表单数据、用户操作提示等。
- 本书将在1.6节介绍JavaScript。



## 1.3.5 嵌入式脚本页面

- PHP代码一般都是嵌入在HTML文档当中，通过服务器解释这些PHP代码，并用代码执行产生的结果替换 PHP代码内容，最后返回给用户的是内嵌代码执行后的 HTML文档。例如内嵌生成“Hello World”的HTML页面代码如下：

- `<html>`
- `<head>`
- `<title>HTML-PHP</title>`
- `</head>`
- `<body>`
- `<h2><?php echo "Hello World"; ?></h2>`
- `</body>`
- `</html>`



## 1.4 Web编程的基础知识之一：HTML

- HTML是创建Web应用的最基本内容，无论是动态还是静态页面，最终都要产生HTML文档。所有的Web开发都要涉及到用HTML设计Web页面。本节将介绍HTML语言及如何使用HTML创建Web页面。



## 1.4.1 HTTP协议简介

- 网络上的计算机之间要进行通信，就必须遵守一定的规则，这种通信规则就是网络协议。协议保证网络上各种不同的计算机之间能够理解彼此传递的消息，好比操不同语言的人们之间，通过翻译来理解对方所说话的含义一样。现在应用最广的Internet使用的是TCP/IP协议，而浏览WWW使用的是HTTP协议，即超文本传输协议（HyperText Transfer Protocol），此协议建立在TCP/IP协议之上。
- 浏览网页的过程，其实就是一系列请求/响应的过程。HTTP协议定义了这个请求/响应过程中请求和响应的格式，及维护HTTP链接的内容。



## 1.4.2 HTML基本知识：标签

- HTML的全称是HyperText Markup Language，即超文本标记语言。它是一种简单、通用的标记语言。之所以叫标记语言，是因为，HTML通过不同的标签，来标记文档的不同部分。读者看到的每个Web页面，都是由HTML通过一系列定义好的标签生成的。
- 从简单的文本编辑器，如Windows的记事本，到专业化的编辑工具，如Dreamweaver，都可以用来编辑HTML文档，编辑好的HTML文档必须按后缀.html或.htm来保存，最后，通过浏览器打开HTML文档，来查看页面效果。
- 在HTML文档中，标签是包含在“<”和“>”之间的部分，如<p>就是一个标签。标签一般是成对使用的，如<b>和</b>同时使用，其中<b>是开始标签，</b>是结束标签。HTML的标签不区分大小写，因此<B>和<b>表示的含义相同。



## 1.4.3 HTML基本知识：元素

- HTML元素由标签定义，标签所定义的内容就叫“元素”，元素包含在开始标签和结束标签之间。
- 每一种HTML元素，一般都会有一个或数个属性，属性用来设置或表示元素的一些特性、名称或显示效果等。属性放在元素标签中，紧跟标签名称之后，它和标签名称之间有一个或数个空格。元素的每个属性都有一个值，属性的值的设定使用“属性=”值””的格式，可以为属性的‘值’加上引号或不加引号。下面的HTML代码为标签<form>设置了name属性，其值为login，表示这个表单的名称为login。
- `<form name=" login" >`



## 1.4.4 HTML基本知识：HTML的基本元素

- 元素是组成HTML文档的关键，本节介绍HTML常用的几种基本元素。
- 1. 标头元素:HTML使用标签<head>定义一个标头，结束标签是</head>。
- 2. 标题元素:这里的标题，是指HTML文档中，内容的标题。标题元素由标签<h1>到<h6>定义。
- 3. 段落元素:HTML中使用标签<p>和</p>定义一个段落。
- 4. 字形元素:使用标签<b>和</b>定义一个粗体字形元素
- 5. 链接:HTML文档中指向其他Web资源，如另一个HTML页面、图片等的链接被称为“锚”。在HTML中使用标签<a>和</a>定义一个锚元素，即链接元素，也就是说在<a>和</a>之间的内容，会成为一个超链接。



## 1.4.4 HTML基本知识：HTML的基本元素

- 6. 图像元素：使用标签<img>定义一个图片元素，在标签中使用属性src来指向一个图片资源，象这样，其中url是指向资源所在位置。这个位置可以是一个URL，也可以是一个相对地址，
- 7. 表格元素：使用标签<table>和</table>定义一个表格元素。一个表格由“行”构成，每一行由数据单元构成。表格的“行”用标签<tr>和</tr>定义，数据单元用标签<td>和</td>定义。
- 8. 列表元素：说明：HTML的列表分为无序列表和有序列表。
- 9. 表单元素：HTML表单是一个包含表单元素的区域，表单元素一般会作为数据，提交给后台服务器做处理。表单域用标签<form>和</form>定义。表单元素是那些定义在表单域里，可以输入信息的元素，如文本框、单选按钮、下拉列表等。



## 1.4.5 创建HTML文档

- 使用任何一款文本编辑器，都可以编辑HTML文档。编辑好的HTML文档，按后缀名.html或htm保存，最后通过浏览器访问HTML文档。打开一个文本编辑器，键入如下HTML代码（代码1-15），



## 1.4.6 创建HTML列表

- HTML最基本的列表有两种：无序列表和有序列表，列表可以嵌套使用，亦可同时使用。
- 创建无序列表，通过设置<ul>标签的type属性，可以更改列表项的标志。type属性的取值可以是disc, square或circle中的任意一个。
- 创建有序列表，通过设置标签<ol>的type属性，可以为有序列表的列表项设置不同的顺序标志。
- 对于无序列表来说，



## 1.4.7 创建页面表格

- 标签<table>和</table>用来创建一个表格。标签<table>主要的常用属性有width、align和border。
- width属性：设置表格的宽度，如下所示。这里定义一个宽度为600px（600点像素）的表格。
- <table width=" 600px" >
- align属性，设置表格相对于浏览器区域的对其方式，可以选取的值有：left（居左）、center（居中）或right（居右）。
- border，设置表格的边框宽度，单位为像素。



## 1.4.8 建立页面表单

- 使用标签<form>和</form>创建一个表单。<form>的主要属性是action和method。
- action属性：用来指定表单数据被提交后，处理这些数据的程序的地址。如下HTML代码，表示当表单提交后，表单的数据将被传到文件login.php，由login.php来处理传入的数据。
- <form action=" login.php" >
- method属性：指定用何种HTTP方式传递数据。
- 有两种传递数据的方式：POST方式和GET方式。POST方式将表单数据放在HTTP数据的正文部分传递。GET方式将表单数据加到action所指的地址之后传递。



## 1.5 Web编程的基础知识之二：层叠样式表（CSS）简介

- 层叠样式表的英文全称是 Cascading Style Sheet（简称 CSS）。HTML最初设计时，只是用来定义文档的内容。比如标签<table>就是用来定义一个HTML文档的表格。页面的布局由浏览器显示，并不由HTML标签控制。
- 随着一个站点HTML文件的越来越多，如果用仍HTML标签排版和控制页面显示效果，它的局限性和困难性的问题会日益突出。甚至，在标签无法满足页面显示效果时，有的设计人员又加入JavaScript来控制页面效果。可以想象，这样HTML文档会变得越发臃肿，并且越来越难以维护。
- 层叠样式表（CSS）的出现解决了这个问题，即使用CSS决定网页内容如何显示，用CSS控制页面显示效果。



## 1.5.1 样式表的基本语法

- 一个样式 (Style) 的语法由3部分构成: Selector (选择器), 属性 (Property), 属性值 (Value)。格式如下:
- selector {property: value}
- 例如下面的例子, p就是selector, color就是属性, blue就是属性值。
- p {color:blue}
- 其中p就是指p标签<p>, 这个样式 (Style) 的含义表示凡是用<p>标签标记的文档内容, 其文本颜色显示为蓝色。HTML中所有的标签都可以作为 selector。
- 如果想为Style添加多个属性, 可以在两个属性之间用分号分隔。下面的Style就包含两个属性, 一个是对齐方式, 其值为居中, 一个字体颜色, 其值为红色, 它们之间用分号分隔开。
- p {text-align:center;color:red}



## 1.5.2 设置页面字体格式

- 使用CSS可以设置字体的名称、大小、显示风格等样式。
- 1. 字体名称属性: font-family
- 用这个属性设定字体的名称, 如Arial, Tahoma, Courier等。
- 2. 字体大小属性: font-size
- font-size属性用来设定字体的大小, 字体大小的单位有多种, 常见的是pt和px。
- 3. 字体风格属性: font-style
- 这个属性有3个值可选: normal (正常显示)、italic (斜体显示)、oblique (斜体显示)。normal是缺省值。



## 1.5.3 设置页面颜色和背景

- 使用CSS背景颜色属性和背景图片属性，可以很方便地设置Web页面的颜色和背景。
- 1. 背景颜色属性：background-color
- 此属性为HTML元素设定背景颜色，示例代码如下。
- ```
body {background-color:#99FF00;}
```
- 上面的代码表示body这个HTML元素的背景颜色值是#99FF00。



1.5.3 设置页面颜色和背景

- 2. 背景图片属性: background-image
- 此属性为HTML元素设定背景图片,



页面背景效果



1.5.4 处理页面的边距和填充

- 边距属性是用来设置页面中一个元素所占空间的边缘到相邻元素之间的距离。
- 左边距属性: `margin-left`: 这个属性用来设定左边距的宽度。示例代码如下。
- `.d1 {margin-left:1cm}`
- 右边距属性: `margin-right`: 这个属性用来设定右边距的宽度。示例代码如下。
- `.d1 {margin-right:1cm}`
- 上边距属性: `margin-top`: 这个属性用来设定上边距的宽度。示例代码如下。



1.5.4 处理页面的边距和填充

- `.d1 {margin-top:1cm}`
- 下边距属性: `margin-bottom`: 这个属性用来设定下边距的宽度。示例代码如下。
- `.d1 {margin-bottom:1cm}`
- 边距属性: `margin`: 这个属性是设定边距宽度的一个快捷的综合写法, 用这个属性可以同时设定上下左右边距属性。还可以为上下左右边距设置相同的宽度。示例代码如下。
- `.d1 {margin:1cm}`



1.5.5 理解HTML层的概念

- HTML中，使用标签<div>和</div>来定义一个层，通过CSS指定不同的属性值，可以定位层，从而实现页面的布局。
- 利用层可以非常灵活地放置内容，例如可以将层前后放置、隐藏某些层而显示其他层、在屏幕上移动层等。可以在一个层中放置背景图像，然后在该层的前面放置第二个层，从而包含带有透明背景的文本。通常，可以把层看成一个容器，在层里可以放置其他更多的HTML元素。



1.5.6 按Web标准建立网页

- 传统HTML布局，一般都是使用表格。表格定位比较简单快捷，但容易出现表格嵌套表格的现象，这样对页面的后期维护很不方便，而且表格越多，越会影响到浏览器解析HTML文档的速度，使页面打开的速度变慢。如今，页面设计人员更多地使用HTML层和CSS，用于Web页面的布局。
- 所谓的Web标准，其实并没有统一的标准。不过，它似乎朝着使用<div>和CSS进行页面布局的方向行进。DIV+CSS只是具体的实现技术手段，并不能涵盖 web标准。web标准不仅仅是布局的问题，更重要的是信息结构清晰、内容与表现相分离，而DIV+CSS技术能较好的实现这种思想。因此，当前看到的多数符合标准的页面都是采用 DIV+CSS制作。



1.5.7 在网页中引入样式表

- 最常用的样式表引入方式有：内嵌样式（Inline Style）、内部样式表（Internal Style Sheet）和外部样式表（External Style Sheet）。
- 内嵌样式：内嵌样式是写在标签里面的。内嵌样式只对所在的标签有效。如下代码所示。
- `<p style="font-size:20pt; color:red">`用Style定义字体`<p>`
- 这里在`<p>`标签内增加样式：`style="font-size:20pt; color:red`，这个样式只对当前`<p>`标签有效。
- 内部样式：内部样式表是写在HTML的`<head></head>`里面的。内部样式表只对所在的网页有效。
- 外部样式：外部样式是指，将样式（Styles）写在一个以.css为后缀的CSS文件里，然后在每个需要用到这些样式（Styles）的网页里引用这个CSS文件。



1.6 Web编程的基础知识之三：JavaScript基础

- JavaScript是一种基于对象和事件驱动并具有安全性能的脚本语言，它是Web页面设计的重要组成部分，它被嵌入在HTML文档中，由浏览器解释执行。可以使用JavaScript设计出更有效果的Web页面、验证Web页面表单数据、创建cookie等。



1.6.1 网页中的JavaScript

- JavaScript可以出现在HTML文档的任何地方，但必须包含在“`<script language="JavaScript">`”和“`</script>`”之间。一般情况下，使用“`<script language="JavaScript">`”和“`</script>`”包含的JavaScript代码。



1.6.2 JavaScript的变量

- JavaScript变量是用来存储数据的地方。这样在需要用这个值的地方就可以用变量来代表，一个变量可以是一个数字、文本等。JavaScript变量的命名区分大小写，例如，名为computer的变量和名为 Computer的是不一样的。而且，变量名必须以字母或下划线开头。
- JavaScript是一种对变量的数据类型要求不太严格的语言，所以不必声明每一个变量的类型，但在使用变量之前先进行声明是一种好习惯。在JavaScript中用var语句声明一个变量，象下面这样：

```
var name = value
```



1.6.3 JavaScript的基本语句

- 这一小节仅介绍JavaScript的几种常用的基本语句，它们是：
- if…else条件语句。
- switch选择语句。
- for循环语句。
- while循环语句。



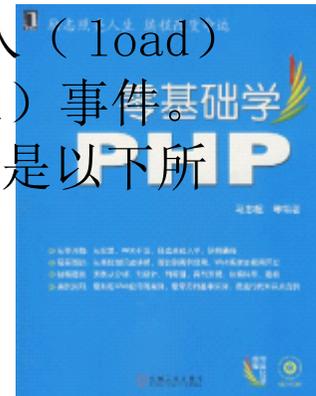
1.6.4 使用简单的对话框

- 通常会使用JavaScript在网页上产生弹出对话框的效果。用JavaScript可以创建的对话框主要有以下所示的两种。
- 警告对话框。
- 确认对话框。
- 下面将通过实例代码，来了解如何使用JavaScript创建这些对话框。使用函数alert()创建警告对话框，它的语法格式如下所示。
- alert(“message”)



1.6.5 JavaScript的基本事件

- 所谓事件，是指用户与Web页面交互时产生的各种操作，例如单击Web页面上的超链接或按钮时，就产生一个单击（click）事件（event）。浏览器为了响应某个事件而进行的处理过程，叫做事件处理。
- 注意：浏览器在程序运行的大部分时间都等待事件的发生，事件能够被Web页面中的JavaScript捕捉到，并在事件发生时，自动调用JavaScript的事件处理函数，完成事件处理。
- 除此之外，浏览器自己的一些动作也可以产生事件，例如，当浏览器载入一个Web页面时，就会发生载入（load）事件，卸载一个页面时，就会发生卸载（unload）事件。本小节将介绍几种常见的JavaScript事件，它们是以下所示的3种。



1.6.5 JavaScript的基本事件

- 鼠标单击事件（onClick）：当用户单击Web页面上的超链接或者按钮时，就会触发onClick事件。
- 页面载入和卸载事件（onload和onunload）：
onload事件发生在Web页面完全下载完成后。
onunload事件发生在用户离开（即关闭）当前Web页面时。onload事件通常用来检测浏览器的一些信息，如浏览器类型、版本等。它们也可以用来处理用户cookie。
- 提交事件（onSubmit）：该事件发生在页面表单的“提交”按钮按下并放开之后。通常使用该事件的处理过程验证表单数据的有效性。



1.6.6 JavaScript的函数

- 函数是一个可以完成特定功能可执行的代码块，它由一条或多条JavaScript语句组成。通常将那些会反复使用，或者用来完成某种功能的代码写成函数，以便利于代码的重用。JavaScript中函数的语法如下所示。
- `function name(var_1, var_2, ..., var_n)`
- `{`
- `statement`
- `}`



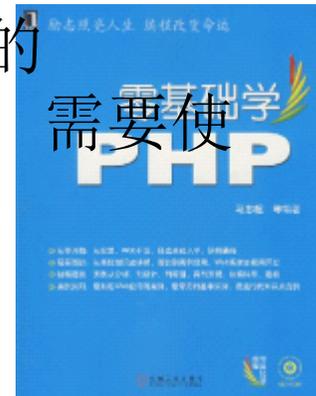
1.6.7 JavaScript的对象

- JavaScript是一种面向对象的编程语言。一段文字、一个图片、一个表单都可以看作是一个对象，每个对象都有自己的属性、方法和事件。属性表示了该对象的某些特征，如字符串的长度、文本框里的文字等。方法可以理解成该对象可以处理的一些事情。
- JavaScript提供了很多对象，这些对象可以直接在JavaScript里使用。本小节介绍JavaScript的一些常用的基本对象。



1.6.7 JavaScript的对象

- 1. String对象
- String对象代表一个字符串，定义该对象的最简单办法就是直接赋值。如下代码所示。此时，变量str就是一个String对象，它的值是“Hello JavaScript”。
- `var str = “Hello JavaScript”`
- 2. 日期对象——Date
- 顾名思义，日期对象就是用来处理时间的JavaScript对象。要定义一个日期对象，需要使用new运算符，如下代码所示。
- `var some_date = new Date()`



1.6.7 JavaScript的对象

- 3. 数组对象——Array
- 数组对象用来存放一组数据（或者说是对象），这些数据可以是不同类型的。数组中的每个对象都有一个“下标”，用来表示它在数组中的位置。数组的下标从0开始算起，所以，第一个对象的位置是0，第二个对象的位置是1，以此类推。JavaScript中定义一个数组的方法如下代码所示。
- ```
var my_arr = new Array()
```



## 1.6.7 JavaScript的对象

- 4. window对象
- 该对象表示的是一个浏览器窗口。引用该对象的方法或属性时，不需要使用“window.xxx”这种形式，而是直接使用该对象的方法或属性。该对象常见的属性如下所示。
- name: 当前窗口的名称。 states: 表示浏览器窗口下方状态栏所显示的内容。通过对改属性发赋值，可以改变浏览器状态栏显示的内容。 self: 指浏览器窗口本身。 history: 浏览器历史访问对象。
- location: 浏览器地址对象。 document: 文档对象。 window对象常见的方法如下所示。
- open(): 打开一个窗口。 close(): 关闭一个已打开的窗口。 alert(): 弹出一个包含“确定”和“取消”按钮的对话框。该方法已经在前面介绍过。



## 1.6.7 JavaScript的对象

- 5. document对象
- document对象代表浏览器窗口内的文档。该对象包含了整个HTML文档，并且可以访问当前页面中的所有元素。该对象最常用的方法就是write，它向当前HTML文档写入数据。前面的内容已多次使用过该方法，这里不再重复举例。document有一个重要属性：cookie。接下来的一小节，就向读者介绍JavaScript中的cookie。



## 1.6.8 JavaScript中的cookie

- cookie是指当用户访问某一个Web站点时，由服务器存储在客户端计算机中的一些变量。它通常用来区别不同的访问用户。当同一台计算机请求访问某个页面时，浏览器也会将cookie发送给服务器。
- cookie一般是这样的形式：cookie名称=cookie值。cookie的名字一般使用字母和数字命名，cookie的值要求是可以URL编码的字符。所有的cookie都有一个失效期，过了失效期，计算机就会将这个cookie删除。JavaScript中通常使用document的cookie属性存储cookie。它的用法如下所示。
- `document.cookie='cookieName'+escape('cookieValue')+';expires='+expirationDateObj.toGMTString();`



## 1.7 开始PHP之旅：搭建PHP开发环境

- 从这节开始，正式进入和PHP有关的内容。本节将介绍Apache、MySQL和PHP的下载、安装及配置，分别讲述在WindowsXP和Linux/UNIX下开发环境的搭建。为了便于初学者理解和掌握，本书将按Windows操作系统下的开发环境讲述PHP语言及其编程。



## 1.7.1 下载Apache、MySQL和PHP

- Apache、MySQL和PHP都是开源产品，均可从其官方网站下载，并且可以免费使用。
- 1. Apache的下载
- Apache服务器的官方网站是<http://httpd.apache.org>。当前，可以通过镜像地址<http://apache.mirror.phpchina.com/httpd/binaries/win32/>来下载Windows的Apache安装程序，这个镜像地址列有各个版本的Apache安装程序，
- 2. MySQL的下载
- MySQL的官方网站是<http://www.mysql.com>。可以通过<http://dev.mysql.com/downloads/mysql/5.0.html>下载最新版本的MySQL，
- 3. PHP的下载
- PHP的官方网站是<http://www.php.net>。可以通过<http://www.php.net/downloads.php>下载最新的PHP版本。对于Windows系统，可以下载安装版本或者PHP压缩包文件，



## 1.7.2 在Windows下配置开发环境

- 下载完需要的软件后，本节介绍如何实现这些软件的安装和配置。
- 1. Apache的安装
- 2. PHP的安装
- 4. 配置开发环境



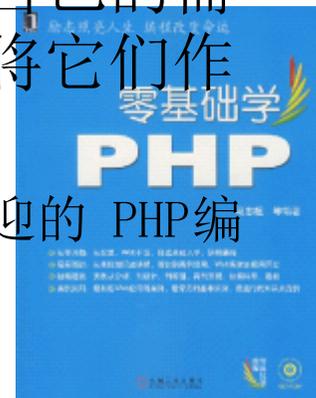
## 1.7.3 在Linux/UNIX下配置开发环境

- 因为在Linux/UNIX下配置PHP的开发环境，对读者的技术要求稍微高一些，而且涉及到一些Linux/UNIX方面的基础知识，如命令的使用、程序编译等，考虑到便于初学者学习和掌握，本节讲述比较简略。另外，本书主要以Windows为平台讲述PHP语言及其开发，因此初学者也可以跳过本节的学习。
- 1. MySQL的安装
- 2. 安装Apache
- 3. PHP的安装



## 1.7.4 善其事利其器——PHP编辑器的选择

- 毋庸置疑，开发人员需要一个功能强大的IDE（Integrated Development Environment，集成开发环境），就好似一个好的猎人，需要一支好的猎枪一样。使用PHP做Web应用的开发，无论是从开发人员的需求上考虑，还是项目的开发效率上考虑，都需要有一个功能强大而且易用的编辑器做支持。如今有许多编辑器可供PHP开发人员选择，它们各有优点，开发人员可以根据自己的需求、使用习惯等方面选择这些编辑器，将它们作为编写代码、开发程序的高效工具。
- 本节将简单介绍三款常用也是比较受开发人员欢迎的PHP编辑器，希望对初学者有所帮助。这三款编辑器是：UltraEdit、Eclipse和Zend Studio。



## 1.8 立竿见影

- 经过上几节的讲述，配置好了PHP的开发环境。从现在开始，通过一些小例子体验一下PHP，看看PHP是怎样一种计算机程序语言，简单了解一下用PHP能做哪些事。从向PHP问好开始，向它说声“Hello, PHP!”！



## 1.8.1 编写第一个PHP程序——“Hello, PHP!”

- 打开最上手的编辑器，来编写第一个PHP程序。这个PHP程序非常简单，
- 在编辑器键入以上代码后，按文件名hello.php保存在Apache安装目录的htdocs目录下。然后打开浏览器，键入地址：  
http://localhost/hello.php，如果一切正确无误的话，将会看到浏览器显示出“Hello, PHP!”字样，



## 1.8.2 使用PHP处理HTML表单

- 这一小节介绍如何使用PHP处理HTML表单数据。在这小节，将编写一个PHP程序，试着用它处理HTML文档提交的数据。



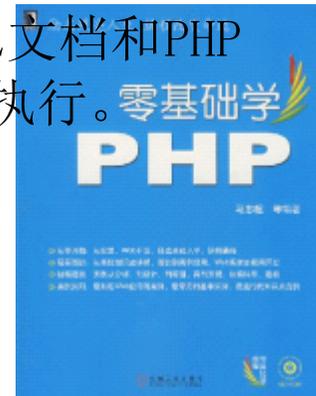
## 1.9 深入了解Apache、PHP和MySQL

- 在正式学习PHP语言之前，还需要对Apache、PHP配置和MySQL的有关方面做一些比较深入的说明，这样即有助于读者对前几节知识加深认识，也有助于读者学习和理解后续内容。以下几小节的内容，主要将以Windows平台下的配置做讲解，Linux/UNIX平台下的情形类似。



## 1.9.1 Apache服务器目录

- Apache安装完成后，有一些目录需要读者进一步了解，如：conf目录、htdocs目录、logs目录和modules目录。
- conf目录：conf目录下存放着一些Apache配置文件，其中最常用到的就是httpd.conf，这是Apache的核心配置文件，Apache服务器的很多重要配置及功能实现都要在这个文件里完成。这个文件也是PHP开发人员需要经常改动的文件，本书将在下一小结讲述httpd.conf，设置一些重要的配置项。
- htdocs目录：这个目录被Apache默认为服务器的根目录。这就是说，在默认情况下，开发人员编写的HTML文档和PHP程序，只有放到这个目录下，才可以被访问或被执行。



## 1.9.1 Apache服务器目录

- logs目录：这个目录下存放着服务器级别的日志文件。如access.log记录用户访问的文件及其访问日期时间、方式等。这个目录下的有些文件，有时可以用来做PHP程序调试之用，因为服务器记录了些错误在这些日志里，开发人员可以通过这些错误，来调试PHP程序。
- modules目录：这个目录下放有Apache执行的核心模块，当Apache启动时，它会根据配置从这个目录里载入需要的模块。一般情况下，PHP开发人员不需要对这个目录了解更多。



## 1.9.2 进行基本的Apache配置

- Apache服务器的很多功能、任务等重要配置，都是通过修改httpd.conf来完成的。如设置服务器根目录、服务器超时时间、监听端口、Apache运行模块的载入、服务器语言字符设置等。下面这段文字就是从httpd.conf摘出的一部分。
- #
- # Timeout: The number of seconds before receives and sends time out.
- #
- Timeout 300



## 1.9.3 Apache的启动与停止

- 对于Windows用户来说，可以通过Windows“服务”来启动和关闭Apache服务器。进入Windows控制面板里的管理工具，打开服务，找到Apache一项，通过“操作”菜单项或图标按钮即可完成Apache的启动与停止。



## 1.9.4 PHP的核心配置——php.ini

- 与Apache相似，PHP也有自己的核心配置文件 php.ini，PHP分析器启动是会读取这个文件。与 httpd.conf 不同，php.ini 以英文分号 “;” 做为注释符本小节主要讲述以下PHP基本的配置项，这些配置项均在 php.ini 中设置。
- doc\_root：该项用来指定PHP页面在服务器的根目录，以本书安装情况为例，该项配置如下所示。
- doc\_root = d:\Apache2\htdocs
- 注意：httpd.conf 配置项与其值之间用空格分割，而 php.ini 使用 “=”。
- extension\_dir：该项用来指定PHP在哪个目录下查找扩展动态库。以本书安装情况为例，该项配置如下所示。表示从PHP安装目录下的 ext 目录里查找扩展动态库。
- extension\_dir = d:\php\ext



## 1.9.5 PHP文件上传功能的配置

- 关于PHP文件上传功能的配置，主要有以下所示的两项。
- `upload_tmp_dir`: 当进行文件上传时，临时文件存放的目录，对于Linux/UNIX用户来说，要特别注意，当前PHP用户必须拥有这个目录的写权限。
- `upload_max_filesize`: 这个配置项指定了允许上传文件大小的最大值，`php.ini`默认值是2M。



## 1.9.6 PHP中session的配置

- 在php.ini中，有关session的基本配置有以下所示的几项。
- session.save\_handler: 这项配置用来设置session的存储方式，一般使用默认值files即可，代表用文件储存。
- session.save\_path: 这项用来设置session的保存路径，以本书为例，将session保存在PHP安装目录的session目录下，如下所示。
- session.save\_path = d:\php\session
- session.use\_cookies: sessionid的传递方式，默认是cookie，推荐使用。



## 1.9.7 PHP中和电子邮件有关的配置

- 在这里PHP的邮件配置目前只需了解一个 `sendmail_path`项即可，其他的配置项将在以后有更多讲述。
- `sendmail_path`这项仅针对Linux/UNIX用户来说，它用来指定sendmail程序的目录位置，通常会为 `/usr/sbin/sendmail`或 `/usr/lib/sendmail`。



## 1.9.8 PHP基本的安全设置

- 这里的安全设置主要是指PHP安全模式方面的内容。基本的配置有以下几项。
- `safe_mode`: 是否允许PHP的安全模式，默认情况此项配置的值是0ff，即关闭安全模式。
- `safe_mode_exec_dir`: 该项表示安全模式下，系统可执行系统程序的目录。这个配置项取决于上一项，如果PHP运行于安全模式下，一些系统函数将会拒绝执行不在该目录下的系统程序。



## 1.9.9 MySQL数据库系统的启动与关闭

- 对于Windows用户，和Apache类似，可以通过Windows服务来关闭和启动MySQL。进入Windows控制面板里的管理工具，打开服务，找到MySQL一项，通过“操作”菜单项或图标按钮即可完成MySQL的启动与停止，



## 1.9.10 MySQL对数据的存储

- 首先了解一下MySQL的目录结构。Windows平台下，
- MySQL会把所有的数据以文件的形式存储在MySQL安装目录的data目录下。开发人员可以通过MySQL创建许多数据库，每创建一个数据库，就会在data目录下生成一个子目录，这个子目录的名字就是数据库的名字，在这些子目录下存放的就是真正的数据文件。关于MySQL更多的概念与内容，将在第12章详细讲述。



## 1.10 实例：使用JavaScript验证HTML表单数据

- 本章介绍的重点内容是HTML、CSS、JavaScript和PHP环境的搭建。其中HTML、CSS已经JavaScript的相关内容又是Web开发的最基础知识，掌握这方面的内容对Web开发至关重要。因为在HTML、CSS及JavaScript三者之间，JavaScript是比较难掌握的，所以，本节以一个JavaScript的应用为实例，作为对Web开发基础知识的一个总结。
- 这个JavaScript实例是用来验证HTML表单数据的简单程序。任何一个Web应用，都离不开数据的提交和处理，这些数据要么被应用程序直接使用，要么被应用程序存入数据库。无论哪种情况，都应该保证浏览器端所提交数据的有效性和正确性。这就要求服务器端程序，在处理数据之前，先对HTML表单所提交的数据进行合法性验证，以保证应用程序执行正常，或者保证存入数据库的数据完整有效。



## 1.11 小结

- 本章较为详细地介绍了Web编程的基础知识和PHP开发环境的配置。Web编程基础部分详细讲述了Web基本体系结构、HTML语言、CSS和JavaScript基础。PHP开发环境的配置讲述了Apache、MySQL和PHP的安装及基本配置，其中重点阐述了如何修改httpd.conf、php.ini等核心配置文件，及其之间的相互关系。最后通过一个实例验证了一个PHP的开发过程，读者也可以通过此例，验证机器的配置是否已经正常使用。

