

第32章 生成PDF格式的个性化文档

在以服务为驱动的站点中，我们有时候需要发布个性化文档，这些文档是根据访问者的输入而生成的。这个特性可以用来实现表单的自动填写或用来生成个性化文档，例如法律文档、信件或证书等。

在本章中，我们所介绍的例子将为用户提供一个在线技术评估页面并生成一个证书。

在本章中，我们主要介绍以下内容：

- 如何使用PHP字符串处理功能将用户数据与模板结合起来，从而创建一个RTF格式的文档。
- 如何使用类似的方法生成PDF格式的文档。
- 如何使用PHP的PDFlib函数来生成类似的PDF文档。

32.1 项目概述

对于这个项目，我们希望能为访问者提供一个包括许多问题的测验。如果他们正确回答了足够多的问题，则给他们一个证书证明通过了该测验。

因为计算机能够比较容易地对问题进行打分，所以问题采用多项选择形式，它由一个问题或多个可能的答案组成。每个问题仅仅有一个答案是正确的。

如果用户在这些问题上获得了及格的分，将获得一个证书。

在理想状况下，我们所颁布的证书文件格式应该是：

- 容易设计。
- 能够包含多种不同的元素，例如位图和向量图。
- 能产生一个高质量的打印输出。
- 只需要下载一个小文件。
- 几乎立即就能生成结果。
- 成本低。
- 可以在多种操作系统上工作。
- 难以伪造及更改。
- 不需要任何特殊的软件来浏览或打印。
- 对所有接收者都能够一致地显示和打印。

像我们有时候做很多决定一样，当选择一种传送格式来满足以上特性时，需要进行一些折中。

评估文档格式

需要做出的最重要决定是用什么格式来传送证书。可选项包括纸张、ASCII文本、HTML、Microsoft Word或其他字处理格式，诸如：RTF、PostScript和PDF等。根据上面列出的特性，

我们可以考虑并比较其中的一些选项。

1. 纸张

用纸张传送证书具有许多明显的优点。我们可以完全控制处理过程。在将证书发送出去之前，可以很清楚地看到输出证书的外观，从而不必担心软件或带宽，并且证书可以使用反伪造技术进行印制。

除了“几乎立即就能生成结果”和“成本低”项要求外，它可以适应其他所有要求。但是纸质证书不能快速制造并传送。根据我们及接收者邮政投递系统的不同，这可能需要花费几天或几周的时间。

每一张证书都要花费几分到几美元的印刷和邮寄费用，而在实际处理中可能需要更多费用。相比较，自动电子投递可能要便宜些。

2. ASCII

以ASCII或纯文本格式来投递文档具有一些优势。首先，兼容性就不会成为问题。带宽要求也会相当小，因此费用会相当低。最终结果的简单性使得设计起来非常简单，而且可以使用脚本快速生成。

但是，如果以ASCII文件的形式向访问者提供证书，那么对证书的外观几乎没有任何控制。我们不能控制字体或者分页。我们仅能包含文本，而几乎对格式没有任何控制。我们也不能控制接收者对文档的复制和更改。这种方法使得接收者易于欺骗性地更改证书。

3. 超文本置标语言 (HTML)

在Web上传送文档，HTML是一个显而易见的选择。超文本置标语言就是专门为该目的而设计的。毫无疑问，HTML包括格式控制，支持包括多种对象（比如图像）的语法，并且与多种操作系统和软件是兼容的（包含一些变化）。它非常简单，因此既易于设计又可以快速地由脚本程序生成和传送。

在本程序中使用HTML的缺陷包括：对打印相关格式的有限支持，比如分页、在不同的平台及程序上的输出不一致性，以及不稳定的打印质量。此外，虽然HTML可以包含任何外部元素的类型，但是对于不常见的类型，浏览器显示或使用这些元素的能力不能得到保证。

4. 字处理器格式

对于企业内部网的项目，提供字处理器格式的文档是尤其有意义的。但是，对于Internet项目，使用专有的字处理器格式会排斥某些访问者，但由于其市场优势，对Microsoft Word来说这样做可能有意义。多数用户会用Word或者某种字处理器（例如，OpenOffice Writer）来阅读Word文件。

没有Word的Windows用户可以从如下网址下载免费软件WordViewer：<http://office.microsoft.com/en-us/downloads/ha010449811033.aspx>。

生成像Microsoft Word文档一样的文档有几点好处。只要有一份Word的副本，设计一个文档就变得相当容易。我们可以很好地控制文档的打印效果，文档内容也极具灵活性。当然，也可以通过要求密码的方式来使得接收者难以修改其内容。

不幸的是，Word文件可能会很大，特别是当它们包含图像或其他复杂元素时。而且没有很好的办法使用PHP动态生成这种文档。这种格式是经过文档化的，但是它是一种二进制格式，

而且这种格式的文档化要求得到许可条件。使用COM对象生成Word文档是可能的，但是这相当复杂。

可以考虑的另一个可能方法是使用OpenOffice Writer软件，这个软件具有非专利软件和使用基于XML的文件格式的双重优点。

如今，Office 2003和2007也支持自身定义的XML文件格式。Word和其他Office产品的文档类型定义（DTD）可以从Microsoft.com站点下载。请查阅“Office XML参考模式”（Office XML Reference Schemas）。这是一个不错的选择，但不是最简单的。

5. RTF格式

RTF（Rich Text Format）具有Word的大多数优点，但是RTF文件更容易生成。在打印页的排列和格式方面仍然具有灵活性，仍然可以包含诸如向量图或位图之类的元素。此外还可以确保当用户浏览或打印文档时，他们看到的结果与我们看到的会十分相似。

RTF是Microsoft Word的文本格式。它是为一种在不同程序间传送文档所使用的交互格式而设计的。在某种程度上，它与HTML类似。它使用语法和关键字而不用二进制数据来表达格式化信息，因此相对更容易阅读一些。

这种格式的文档非常丰富和翔实，其规范可以免费获得，用关键字“RTF Specification”搜索Microsoft.com就可找到。

生成一个RTF文档的最简单的方法是在字处理器中选择Save As RTF选项。由于RTF文件仅包含文本，因此直接生成和修改已有的文件也十分简单。

因为这种格式的文档非常丰富而且可以免费使用，与Word的二进制格式比起来，RTF可以被更多的软件读取。请注意，用户如果使用旧版本的Word或其他字处理器软件打开一个复杂的RTF文件，经常会产生不同的显示结果。每种新版本的Word都为RTF引进新的关键字，因此旧版本经常忽略它们无法解释的控件或那些已经没有选择实现的控件。

从我们最初的要求列表来看，RTF证书易于使用Word或其他字处理器设计；可以包含多种不同的元素，如向量图和位图；具有高质量的打印输出；可以简单且快速地生成；并且以低廉的费用通过电子化方式传送。

它可以在多种应用程序及操作系统上使用，尽管结果略微有些变化。不利的一面是：RTF文档可以轻易地被任何人随时随意更改，这对于证书和某些其他文档来说是一个问题。对于复杂文档，其文件可能稍微大一些。

RTF对于许多文档分发程序来说是一个不错的选择，因此我们这里把它作为一个可选项。

6. PostScript

Adobe公司的PostScript是一种页面描述语言。它是一种强大且复杂的编程语言，试图以与设备无关的方式表示文档，也就是说，一段描述可以在不同的设备，比如打印机、显示器上产生一致的输出结果。该格式的文档也非常丰富。除了很多Web站点外，至少有3本图书全面地介绍了这种格式。

PostScript文档可以包含非常精确的格式、文本、图像、嵌入式字体和其他元素。可以通过在应用程序中将PostScript文档打印到PostScript打印机驱动程序，很容易地生成一个PostScript文档。如果对这些感兴趣，可以学习用它直接编辑。

PostScript文档具有很好的可移植性。在不同的设备及操作系统下，它们都可以给出一致的高质量打印输出。

用PostScript发布文档有两个重大缺陷：文件大小可能很大以及许多用户必须下载额外的软件来使用它。

大多数UNIX用户可以处理PostScript文件，但Windows用户通常需要下载一个浏览器，比如GSview，该浏览器使用Ghostscript PostScript解释器。该软件适合多种类型的平台。虽然它是自由软件，但我们不希望强迫用户下载更多的软件。

可以在如下网址获得更多关于Ghostscript的文章：<http://www.ghostscript.com/>。

也可以在如下网址下载：<http://www.cs.wisc.edu/~ghost/>。

对于我们当前的程序，PostScript在具有一致性和高质量的输出方面表现十分出色，而在其他要求方面却表现得相当糟糕。

7. 可移植文档格式 (PDF)

可喜的是，有一种格式既具有PostScript的大多数功能，又具有很多其他优点。PDF (Portable Document Format，也出自Adobe公司) 是出于以下目的来设计文档发布方法的：这种文档在不同的平台表现一致，且可在屏幕或纸上产生可预见的高质量输出。

Adobe公司将PDF称为“面向世界范围开放的电子文档发布的事实标准”。Adobe的PDF是一种统一文件格式，它预留了所有字体、格式、颜色和任何源文档的图形，不论它是在何种应用程序和平台上创建的。PDF文件是简洁的，拥有免费Adobe Acrobat阅读器的人可以随心所欲地共享、浏览、操作和打印。

PDF是一种开放格式，除了其他许多Web站点和正式的手册外，也可以在如下网址找到相关文档：<http://partners.adobe.com/asn/tech/pdf/specifications.jsp>。

从我们要求的特性来看，PDF非常令人满意：PDF可以产生一致的、高质量的输出，可以包含诸如位图或向量图之类的元素，可以使用压缩方法创建较小的文件，可以廉价地通过电子方式进行传送，可以在多数操作系统上使用，而且还包括安全控制。

依靠PDF来工作是基于这样一个事实，那就是大多数用来创建PDF文档的软件都是商业性的。浏览PDF文件需要一个阅读器，但Adobe公司发布的Acrobat Reader的Windows、UNIX及Macintosh版本都是可以免费使用的。我们站点的许多访问者应该已经相当熟悉.pdf的扩展名，而且他们可能已经安装了阅读器。

PDF文件是一种易于发布的、可打印文档的好方法，特别是对那些不希望接收者轻易更改的情况。

接下来，我们将要通过两种不同的方法来生成PDF证书。

32.2 解决方案的组成

为了使系统能够工作，我们需要测试用户已经掌握的知识（假设他们通过了测试），并生成一个证书来报告成绩。我们通过3种方法来原因证书的生成：使用RFT模板，使用PDF模板以及通过程序创建新的PDF。

下面，我们仔细了解各组成部分的要求。

32.2.1 问题与回答系统

提供一个灵活的在线评估系统，用以支持不同的问题类型，支持信息的不同媒体类型，对错误回答给出有效反馈，以及支持智能的统计信息搜集和报告，这本身就是一个非常复杂的任务。

在本章中，我们主要兴趣在于生成可以通过Web发布的用户自定义文档，因此只创建一个非常简单的测试系统。这个测试系统不依赖于任何特殊的软件，它使用HTML表单来提出问题，使用PHP脚本来处理用户的回答。事实上，我们从第1章就开始了PHP脚本的编写。

32.2.2 文档生成软件

在Web服务器上，我们不需要额外的软件就可以从模板中生成RTF或PDF格式的文档。但是需要使用某些软件来建立模板。为了使用PHP PDF 创建函数，必须将PDF支持编译到PHP中。（我们将在稍后的内容详细讨论。）

1. 创建RTF模板的软件

可以使用我们选择的字处理器软件来生成RTF文件。我们使用Microsoft Word来创建证书模板。证书模板包含在附带文件的第32章的目录下。

如果希望使用其他字处理软件，最好是先在Word下测试其输出效果，因为这是网站访问者最经常使用的软件。

2. 创建PDF模板的软件

PDF文档生成起来稍微难些。最简单的方法就是购买Adobe Acrobat软件。该软件允许我们从各种应用程序中创建高质量的PDF。对于该项目，我们将使用Acrobat来创建模板文件。

为了创建该文件，可以使用Microsoft Word来设计一个文档。Acrobat包中的一个工具是Adobe Distiller。在Distiller中，需要选择一些非默认的选项。该文件必须以ASCII格式保存，而压缩选项必须关闭。设置完这些选项后，创建一个PDF文件就同打印一样简单。

可以在如下网址找到更多关于Acrobat的内容：<http://www.adobe.com/products/acrobat/>。

我们可以通过在线方式或者从普通软件零售商那里购买该软件。

创建PDF的另一个方法是使用转换程序ps2pdf，正如其名所示，该程序将PostScript文件转换成PDF文件。该程序是免费的，但是对于那些包含图像或非标准字体的文档并不总是具有好的输出。ps2pdf转换器包是前面提到的Ghostscript包附带的。

很明显，如果想通过这种方法创建PDF文件，必须先创建一个PostScript文件。通常，UNIX用户会使用a2ps或dvips工具来创建PostScript文件。

如果在Windows环境下工作，也可以在不使用Adobe Distiller的情况下创建PostScript文件，虽然创建过程可能稍复杂一些。需要安装一个PostScript打印机驱动程序。例如，可以使用Apple LaserWriter IINT驱动程序。如果没有安装PostScript驱动程序，可以从Adobe公司的网址下载一个：<http://www.adobe.com/support/downloads/product.jsp?product=44&platform=Windows>。

要创建PostScript文件，必须选择该打印机和“打印到文件”选项，通常，可以在打印对话框里找到该选项。

大多数Windows应用程序将产生一个.prn扩展名的文件。这应该是一个PostScript文件。需要将它重命名为.ps文件。使用GSview或其他PostScript浏览器，可以浏览它，或者可以通过ps2pdf工具创建一个PDF文件。

请注意，不同的打印机驱动程序将产生不同质量的PostScript输出。我们可能会发现生成的某些PostScript文件在ps2pdf工具中运行时会产生错误。遇到这种情况，我们建议使用不同的打印机驱动程序。

如果只想创建少量的PDF文件，Adobe在线服务能够满足要求。每个月只需要9.99美元，可以上传多种格式的文件并下载转换后的PDF文件。这个服务很适合我们的证书，但是，它却不允许我们选择某些对项目很重要的选项。生成的PDF将作为一个二进制文件保存并要经过压缩。这使得修改它变得非常困难。

该服务可以在如下网址找到：<https://createpdf.adobe.com/>。

如果希望试用这个服务，也可以找到该服务的免费试用选项。此外，如果你要创建的PDF文件数量少于5个，你也可以Adobe的免费服务：<http://www.acrobat.com>。

在Net Distillery网站上，还有一个基于FTP的ps2pdf下载界面：<http://www.babinszki.com/distiller/>。

最后一个选项就是以XML格式对证书进行编码，然后使用XML风格样式单转换器(XSLT)将其转换成PDF格式和任何其他所需的格式。这种方法需要对XSLT的很好理解，在这里不做介绍。

3. 用软件通过编程方式创建PDF

PHP本身就提供了对创建PDF文档的支持。PHP的PDFlib函数使用PDFlib函数库。该函数库可以从<http://www.pdflib.com/products/pdflibfamily/>下载。PDFlib提供了生成PDF文档所需的API。

PDFlib并不是免费的；它要求商业许可。PDFlib Lite是免费的，而且是开放源代码的，但是只有在特定条件下，例如用于非商业用途。

此外，还有一些免费的函数库，例如FPDF。FPDF使用PHP语言编写（而不是用C语言编写的PHP扩展），它的性能要慢于其他两个库。你可以从<http://www.fpdf.org/>下载FPDF。

在本章中，我们将使用PDFlib，因为它可能是使用的最为广泛的创建PDF的扩展。

检查phpinfo()函数的输出，你可以检查系统是否已经安装PDFlib。在PDF标题部分，你可以找到PDFlib支持是否已经启用以及所使用的PDFlib版本。

如果要在PDF文档使用TIFF或JPEG图像，你还需要安装TIFF函数库（可以从<http://www.libtiff.org/>下载），以及JPEG函数库（可以从[ftp://ftp.uu.net/graphics/jpeg/](http://ftp.uu.net/graphics/jpeg/)下载）。

PDFlib扩展并没有内置在PHP中；你必须从PECL（PHP扩展组织函数库）获得这些文件并且手动安装这些扩展。

在非Windows系统中，你可以<http://pecl.php.net/package/pdflib>处下载这些扩展文件并使用pecl命令安装。请参阅<http://www.php.net/manual/en/install.pecl.pear.php>处的说明。

在Windows系统中，从http://pecl4win.php.net/ext.php/php_pdflib.dll下载预编译扩展（php_pdf.dll）或者从PHP.net下载页面下载完整的已编译PECL扩展。下载后，将php_pdflib.dll文件保存在PHP扩展目录（通常是PHP安装路径下的ext目录），并且在php.ini文

件添加如下代码：extension=php_pdf.dll。

32.3 解决方案概述

在这个项目中，我们将设计一个具有3种可能输出的系统。如图32-1所示，我们将提出测验问题，评估用户的回答，然后用以下3种方法之一来生成证书：

- 用一个空白模板生成RTF文档。
- 用一个空白模板生成PDF文档。
- 通过PDFlib库用程序生成PDF文档。

表32-1给出了该证书项目将用到的所有文件。

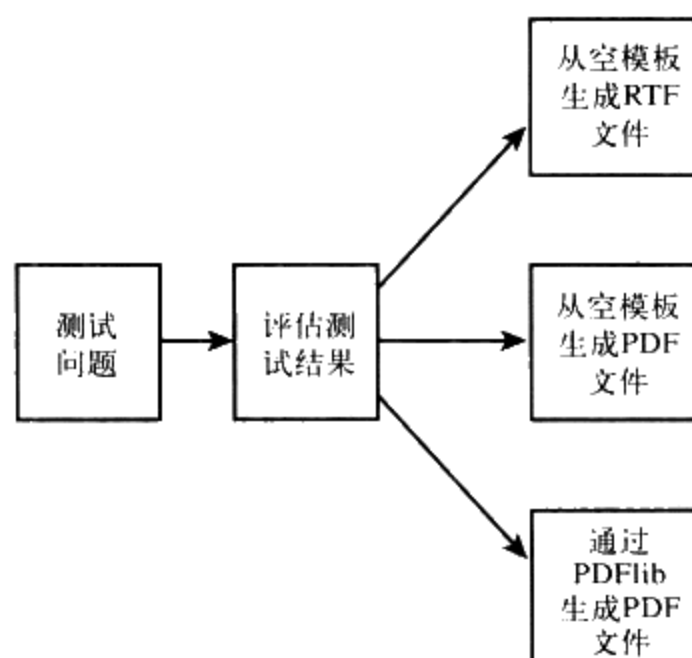


图32-1 证书系统将生成3种不同证书中的一种

表32-1 证书应用程序将用到的所有文件

文件 名	类 型	描 述
index.html	HTML页	包括测验题目的HTML表单
score.php	应用程序	评估用户回答的脚本程序
rtf.php	应用程序	用模板生成RTF的脚本程序
pdf.php	应用程序	用模板生成PDF的脚本程序
pdflib.php	应用程序	用PDFlib库生成PDF证书的脚本程序
signature.png	图像	包含在PDFlib证书中的签名的位图图像
PHPCertification.rtf	RTF	RTF证书模板
PHPCertification.pdf	PDF	PDF证书模板

下面，我们继续学习该程序。

32.3.1 提问

index.html文件非常直观。它必须包含一个HTML表单来询问用户姓名及一些问题的答案。在一个真正的评估应用程序中，很可能是从数据库中取出这些问题。在这里，我们将集中讨论

产生证书的问题，因此只随意编造几个题目“硬编码”在HTML中。

姓名域是一个文本输入框。每个题目有3个单选按钮以供用户选择答案。该表单还有一个带有图像的提交按钮。

该页的代码如程序清单32-1所示。

程序清单32-1 index.html——包含测验问题的HTML页

```
<html>
  <body>
    <h1><p align="center">
      <img src='rosette.gif' alt="">
      Certification
      </p></h1>
    <p>You too can earn your highly respected PHP certification
      from the world famous Fictional Institute of PHP Certification.</p>
    <p>Simply answer the questions below:</p>

    <form action="score.php" method="post">

      <p>Your Name <input type="text" name="name" /></p>

      <p>What does the PHP statement echo do?</p>
      <ol>
        <li><input type="radio" name="q1" value="1" />
          Outputs strings.</li>
        <li><input type="radio" name="q1" value="2" />
          Adds two numbers together.</li>
        <li><input type="radio" name="q1" value="3" />
          Creates a magical elf to finish writing your code.</li>
      </ol>

      <p>What does the PHP function cos() do?</p>
      <ol>
        <li><input type="radio" name="q2" value="1" />
          Calculates a cosine in radians.</li>
        <li><input type="radio" name="q2" value="2" />
          Calculates a tangent in radians. </li>
        <li><input type="radio" name="q2" value="3" />
          It is not a PHP function. It is a lettuce. </li>
      </ol>

      <p>What does the PHP function mail() do?</p>
      <ol>
        <li><input type="radio" name="q3" value="1" />
          Sends a mail message.
        <li><input type="radio" name="q3" value="2" />
          Checks for new mail.
        <li><input type="radio" name="q3" value="3" />
```



```

        'Toggles PHP between male and female mode.'
    </ol>

    <p align="center"><input type="image" src="certify-me.gif" border="0"></p>

</form>
</body>
</html>

```

如图32-2所示的是在浏览器中打开index.html的结果。

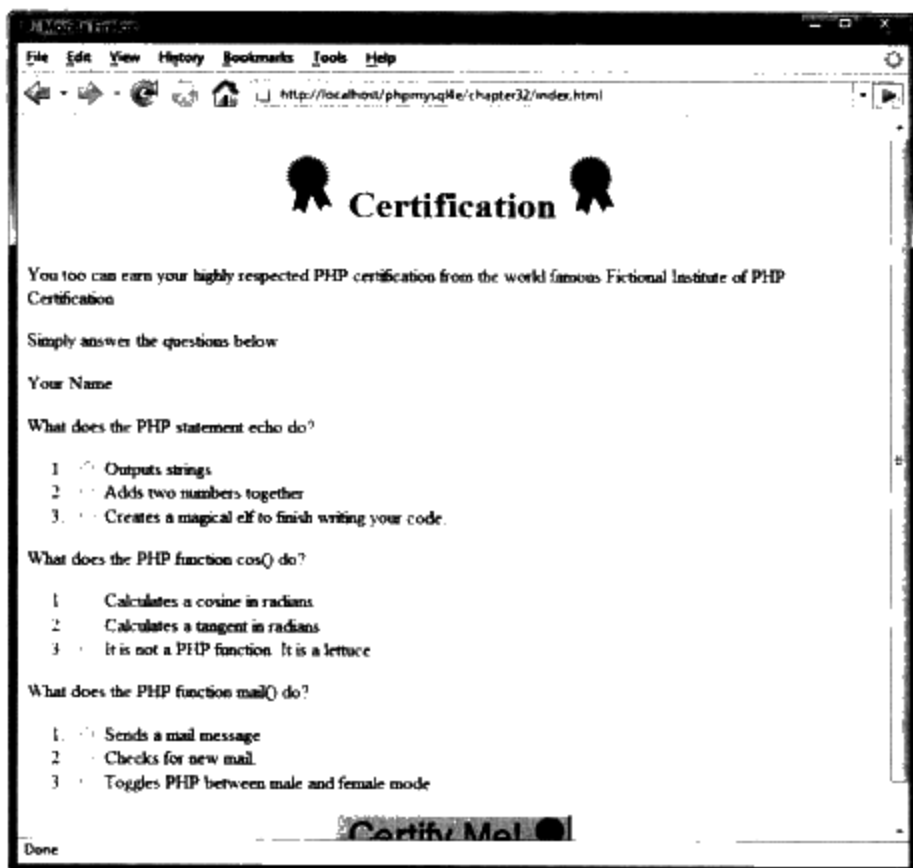


图32-2 index.html 要求用户回答测试问题

32.3.2 给答题评分

当用户在index.html中提交了题目的答案后，我们需要对答案进行评估，并且计算得分。这项工作是由score.php脚本程序完成的。该脚本代码如程序清单32-2所示。

程序清单32-2 score.php——评阅测验的脚本

```

<?php
    //create short variable names
    $q1 = $_POST['q1'];
    $q2 = $_POST['q2'];
    $q3 = $_POST['q3'];
    $name = $_POST['name'];

    // check that all the data was received
    if (($q1==='') || ($q2==='') || ($q3==='') || ($name=='')) {

```

```

echo '<h1>
    <p align=\\"center\\">
    <img src=\\"rosette.gif\\" alt=\\"\\" />
    Sorry:
    <img src=\\"rosette.gif\\" alt=\\"\\" /></p></h1>
    <p>You need to fill in your name and answer all questions.</p>";
} else {
    //add up the scores
    $score = 0;
    if ($q1 == 1) {
        // the correct answer for q1 is 1
        $score++;
    }
    if ($q2 == 1) {
        // the correct answer for q2 is 1
        $score++;
    }
    if ($q3 == 1) {
        // the correct answer for q3 is 1
        $score++;
    }

    //convert score to a percentage
    $score = $score / 3 * 100;

    if ($score < 50) {
        // this person failed
        echo '<h1>
            <p align=\\"center\\">
            <img src=\\"rosette.gif\\" alt=\\"\\" />
            Sorry:
            <img src=\\"rosette.gif\\" alt=\\"\\" /></p></h1>
            <p>You need to score at least 50% to pass the exam.</p>";
    } else {
        // create a string containing the score to one decimal place
        $score = number_format($score, 1);
        echo "<h1 align=\\"center\\">
            <img src=\\"rosette.gif\\" alt=\\"\\" />
            Congratulations!
            <img src=\\"rosette.gif\\" alt=\\"\\" /></h1>

            <p>Well done ".$name.", with a score of ".$score."%,
            you have passed the exam.</p>";

        // provide links to scripts that generate the certificates
        echo '<p>Please click here to download your certificate as
            a Microsoft Word (RTF) file.</p>
            <form action=\\"rtf.php\\" method=\\"post\\">

```

```

<div align=\"center\">
<input type=\"image\" src=\"certificate.gif\" border=\"0\">
</div>
<input type=\"hidden\" name=\"score\" value=\"\".$score.\"\"/>
<input type=\"hidden\" name=\"name\" value=\"\".$name.\"\"/>
</form>

<p>Please click here to download your certificate as
a Portable Document Format (PDF) file.</p>
<form action=\"pdf.php\" method=\"post\">
<div align=\"center\">
<input type=\"image\" src=\"certificate.gif\" border=\"0\">
</div>
<input type=\"hidden\" name=\"score\" value=\"\".$score.\"\"/>
<input type=\"hidden\" name=\"name\" value=\"\".$name.\"\"/>
</form>

<p>Please click here to download your certificate as
a Portable Document Format (PDF) file generated with PDFlib.</p>
<form action=\"pdflib.php\" method=\"post\">
<div align=\"center\">
<input type=\"image\" src=\"certificate.gif\" border=\"0\">
</div>
<input type=\"hidden\" name=\"score\" value=\"\".$score.\"\"/>
<input type=\"hidden\" name=\"name\" value=\"\".$name.\"\"/>
</form>";
    }
}
?>

```

如果用户没有回答完所有问题，或者得分低于规定的及格线，该脚本显示一条消息。

如果用户成功地回答了问题，可以获得一个证书。一个成功测试的输出如图32-3所示。

在这里，用户有3个选项。可以得到一个RTF证书，或者两个PDF证书之一。下面，我们分别介绍相应的脚本。

32.3.3 生成RTF证书

我们可以通过将ASCII文本写入文件或字符串变量的办法来生成一个RTF文档，但这意味着我们要学习另一套语法。

如下所示的是一个非常简单的RTF文档示例：

```

{\rtf1
{\fonttbl {\f0 Arial;}{\f1 Times New Roman;}}
\f0\fs28 Heading\par
\f1\fs20 This is an rtf document.\par
}

```

该文档设置一个包含两种字体的字体表：Arial，对应于f0，Times New Roman，对应于f1。

然后以28号字（14 point），f0（Arial）字体书写“Heading”。\par控制符表示一个段落分隔符。接着，用20号字（10 point）的f1（Times New Roman）字体书写“This is an rtf document”。

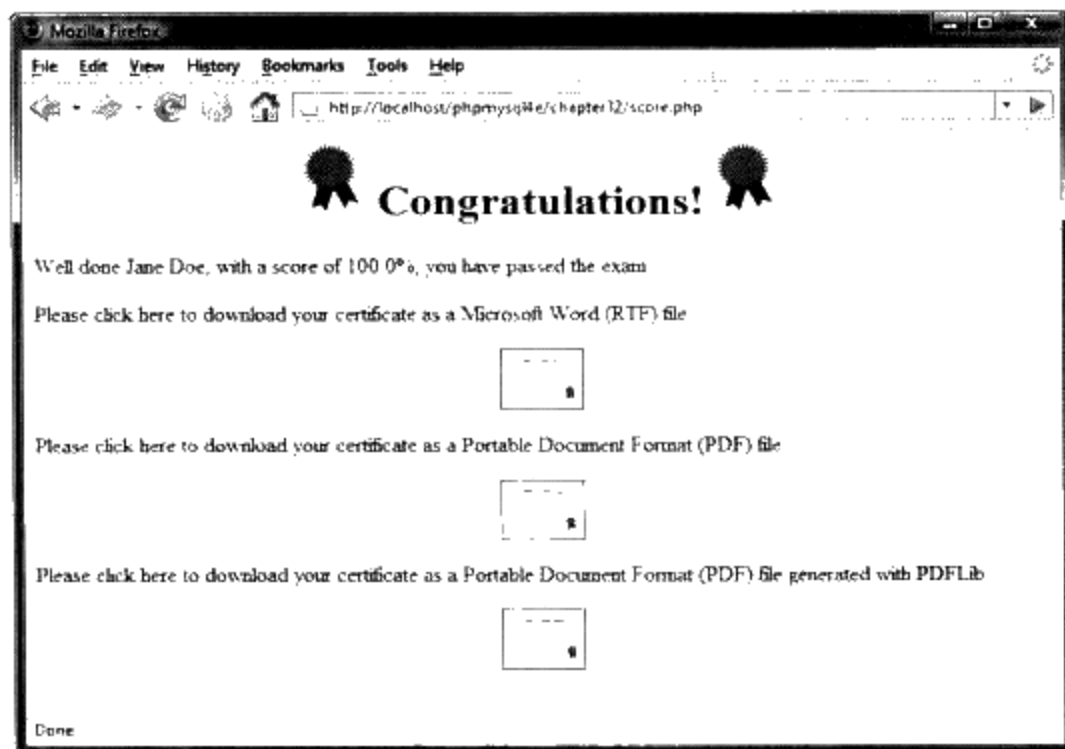


图32-3 score.php向成功回答问题的用户展示3种可供选择证书的选项

我们可以像这样手动生成一个RTF文档，但在PHP中，并没有任何简便的函数来简化这些困难的工作，比如，合并图形。幸运的是，在许多文档中，结构、类型和大部分文本都是静态的，仅有一小部分存在个体差异。使用模板是生成文档的一种更高效的方法。

我们可以使用字处理器方便地建立一个复杂的文档，如图32-4所示。

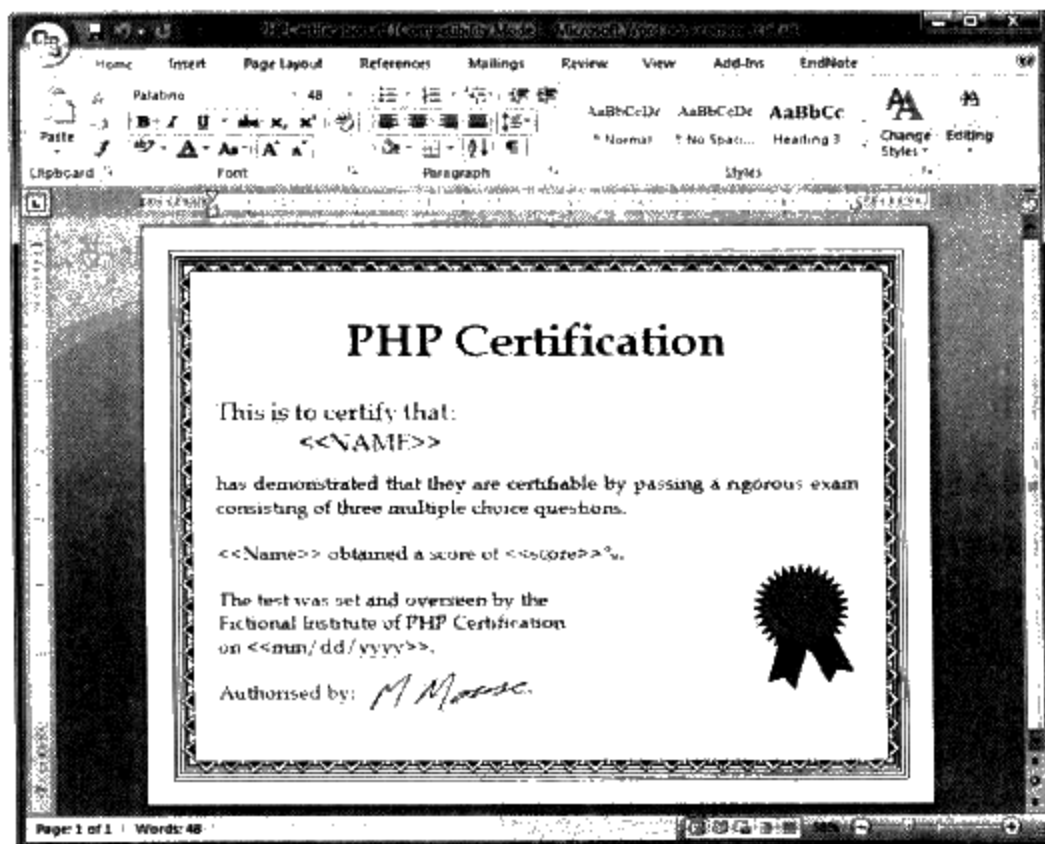


图32-4 使用字处理器，可以容易地创建一个复杂而又有用的模板

模板包括占位符，如<<NAME>>，用来标记该处将插入动态数据。这些占位符是什么样子并不重要。我们在两组尖括号之间使用一个有意义的字符串来描述它。选择一个不可能在文档其余部分显示的占位符是相当重要的。当占位符与替代它们的数据的长度大致相同时，它有助于排列模板格式。

这个文档中的占位符包括<<NAME>>、<<Name>>、<<score>>和<<mm/dd/yyyy>>。请注意，我们既使用了NAME又使用了Name，因为我们想用区分大小写的方法来替换它们。

现在我们已经有了模板，需要一个脚本来让它个性化。该脚本是rtf.php，代码如程序清单32-3所示。

程序清单32-3 rtf.php——产生一个个性化RTF证书脚本

```
<?php
//create short variable names
$name = $_POST['name'];
$score = $_POST['score'];
// check we have the parameters we need

if(!$name || !$score) {
    echo "<h1>Error:</h1>";
    <p>This page was called incorrectly</p>";
} else {
    //generate the headers to help a browser choose the correct application
    header('Content-type: application/msword');
    header('Content-Disposition: inline, filename=cert.rtf');

    $date = date('F d, Y');

    // open our template file
    $filename = 'PHPCertification.rtf';
    $fp = fopen ($filename, 'r');

    //read our template into a variable
    $output = fread( $fp, filesize($filename));

    fclose ($fp);

    // replace the place holders in the template with our data
    $output = str_replace('<<NAME>>', strtoupper($name), $output);
    $output = str_replace('<<Name>>', $name, $output);
    $output = str_replace('<<score>>', $score, $output);
    $output = str_replace('<<mm/dd/yyyy>>', $date, $output);

    // send the generated document to the browser
    echo $output;
}
?>
```

该脚本将执行一些基本的错误检查，从而确保已提交了用户的所有资料，然后转而进行生成证书的工作。

该脚本的输出是一个RTF文件而不是HTML文件，因此我们必须提醒用户浏览器知道这个事实。这个步骤很重要，这样浏览器才能以正确的应用程序打开这个文件，或者在不能辨认.rtf扩展名的情况下能给出一个Save As...类型的对话框。

使用PHP的header()函数，可以指定将要输出文件的MIME类型以发送相应的HTTP页头，如下代码所示：

```
header('Content-type: application/msword');
header('Content-Disposition: inline; filename=cert.rtf');
```

第一个页头通知浏览器正在发送一个Microsoft Word文件（其实并不完全正确，但是它是最有可能打开该RTF文件的应用程序）。

第二个页头通知浏览器自动显示文件内容，并告知推荐文件名为cert.rtf。如果用户想在浏览器中保存该文件，这个名字是用户看到的默认文件名称。

在发送页头之后，我们打开并将该模板RTF文件读入到\$output变量中，通过str_replace()函数，用我们希望显示的实际数据替换占位符。该行代码如下所示：

```
$output = str_replace('<<Name>>', $name, $output);
```

用变量\$name的内容替换任何出现占位符<<Name>>的地方。

替换完成之后，接下来就只需将输出回送到浏览器中。该脚本的示例输出结果如图32-5所示。

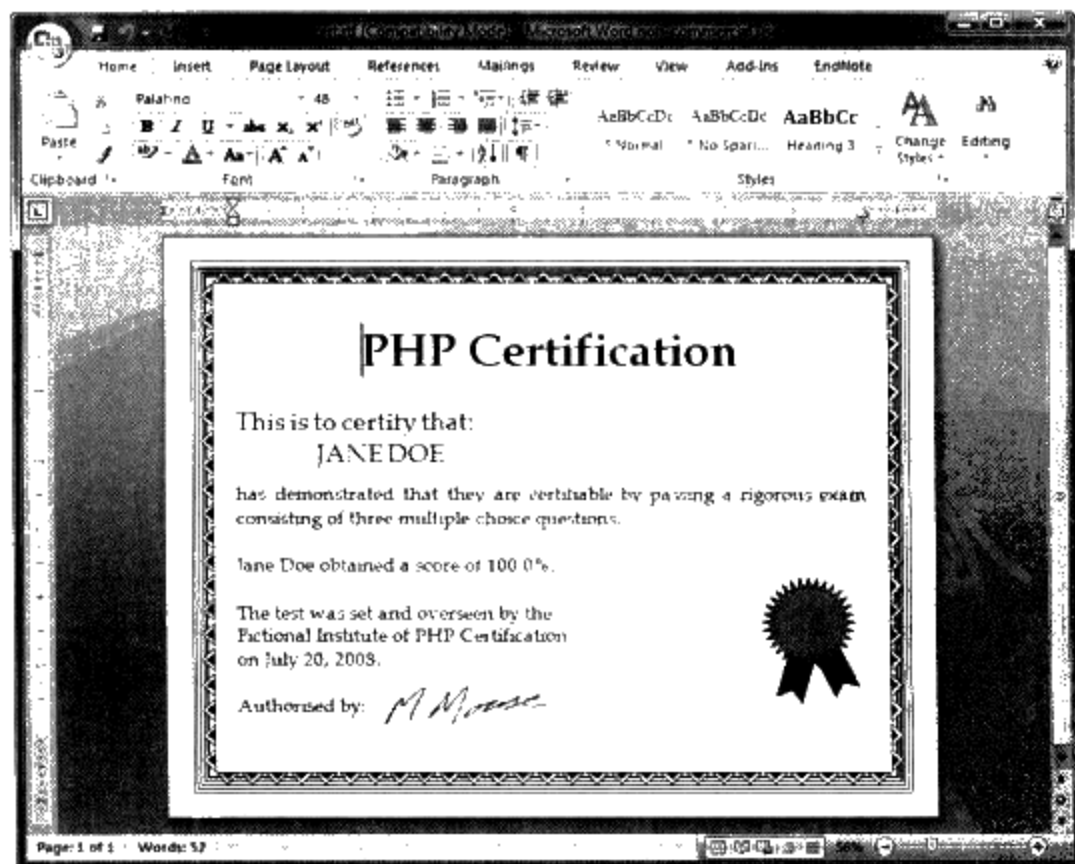


图32-5 rtf.php根据RTF模板生成一个证书

这个方法效果很好。即使在模板和\$output变量的内容非常长的情况下，函数

`str_replace()`的调用执行速度也很快。从这个应用程序的角度来看，主要的问题就是用户将在他自己的字处理器软件中载入证书并打印。这可能会导致用户能够修改证书。RTF不允许我们创建一个只读文档。

32.3.4 从模板生成PDF证书

从模板生成PDF证书的过程非常相似。主要的差异在于当创建PDF文件时，某些占位符可能会被解释为一些格式化代码，这将根据所使用的Acrobat版本不同而不同。例如，如果我们查看所创建的证书模板（使用一个文本编辑器），可以看到占位符将变成：

```
<<N1-13(AME)-10(>)-6(>
<<Na)-9(m)0(e)-12(>>
<)-11(<)1(ac)-17(or)-6(e)-6(>)-11(>
<)-11(<)1(m)-12(m)0(d)-6(d)-19(/)1(yy)-13(yy)-13(>>
```

如果阅读整个文件，将发现与RTF不同，这不是一个我们能够轻易理解的格式。

提示 根据所使用的Acrobat版本或其他PDF生成工具的不同，PDF模板文件也会有所不同。在以上示例所提供的代码可能无法适用于你所生成的模板。请检查模板并且相应地修改其中的代码。如果还存在问题，可以使用本章稍后给出的PDFlib示例。

对于这种情况，我们可以使用几种不同的办法来解决问题。可以检查每个占位符并删除这些格式化符号。事实上，删除这些格式化符号最终将不会太多改变文档外观，因为嵌入到前面模板中的代码表示了我们将要替代的占位字符之间需要留出多少空间。然而，如果采取这种方法，必须在每次修改或更新文件的时候，仔细检查并手动编辑PDF文件。如果仅有4个占位符的时候这也许并不是大问题，但是当需要处理有许多占位符的多个文档，并决定要修改所有文档的信头时，这将会是一件非常可怕的事情。

我们可以通过使用其他技巧来避免此问题。可以用Adobe Acrobat来创建一个PDF表单——与带有空白命名域的HTML表单相似。然后可以用PHP脚本创建所谓的FDF（Forms Data Format）文件，该文件主要是一些将要合并到模板中的数据。可以用PHP的FDF函数库创建FDF：`fdf_create()`函数创建一个文件；`fdf_set_value()`函数设置表单域的值；`fdf_set_file()`函数设置相关的模板表单文件。可以用适当的MIME类型将该文件返回到浏览器中（在这个例子中，MIME类型是vnd.fdf），并且用浏览器的Acrobat Reader插件程序来替换数据并填充到表单中。

这是一个非常巧妙的方法，但是，它也有两点局限性。首先，它需要一份Acrobat专业版的副本（完整版本，不是免费的阅读器，也不是标准版）。其次，在行内的文本而非表单域中进行文本替换是相当困难的。这可能是一个问题，但也可能不是问题，这是由具体操作来决定的。我们已在需要成行地替换许多文本的地方大量使用PDF生成功能来生成字符。FDF不能很好地完成这个功能。例如，如果自动填充一个在线税收表单，这不会成为一个问题。

在Adobe站点，可以看到更多关于FDF格式的介绍：<http://www.adobe.com/devnet/acrobat/fdftoolkit.html>。

如果打算使用这种方法的话，还应该查看PHP手册中的FDF文档：<http://www.php.net/>

manual/en/ref.fdf.php。

现在，我们回到解决前面问题的PDF方案。

如果我们确认附加的格式代码只是由单个的连字符、阿拉伯数字及圆括号组成，并且能够通过正则表达式进行匹配，那么我们仍然可在PDF文件中找出并替换这些占位符。我们已经写了一个pdf_replace()函数来自动地生成某个占位符的匹配正则表达式，并能用适当的文本替代该占位符。

请注意，在Acrobat的某些版本中，占位符以纯文本格式出现。正如前面所介绍的，可以用str_replace()函数来替换它们。

除了这点外，通过PDF模板生成证书的代码与RTF版本的也非常相似，该脚本程序如程序清单32-4所示。

程序清单32-4 pdf.php——通过模板产生个性化PDF证书的脚本程序

```
<?php
    set_time_limit(180); // this script can be slow

    //create short variable names
    $name = $_POST['name'];
    $score = $_POST['score'];

    function pdf_replace($pattern, $replacement, $string) {
        $len = strlen( $pattern );
        $regexp = '';

        for ($i = 0; $i<$len; $i++) {
            $regexp .= $pattern[$i];
            if ($i<$len-1) {
                $regexp .= '(\)\-{\0,1}[0-9]*\(\){0,1}";
            }
        }
        return ereg_replace ( $regexp, $replacement, $string );
    }

    if(!$name || !$score) {
        echo "<h1>Error:</h1>
            <p>This page was called incorrectly</p>";
    } else {
        //generate the headers to help a browser choose the correct application
        header('Content-Disposition: filename=cert.pdf');
        header('Content-type: application/pdf');

        $date = date('F d, Y');

        // open our template file
        $filename = 'PHPCertification.pdf';
        $fp = fopen ($filename, 'r');
```

```

//read our template into a variable
$output = fread($fp, filesize($filename));

fclose ($fp);

// replace the place holders in the template with our data
$output = pdf_replace('<<NAME>>', strtoupper($name), $output);
$output = pdf_replace('<<Name>>', $name, $output);
$output = pdf_replace('<<score>>', $score, $output);
$output = pdf_replace('<<mm/dd/yyyy>>', $date, $output);
// send the generated document to the browser

echo $output;
}
?>

```

该脚本产生PDF文档的一个自定义版本。如图32-6所示，该文档在多数系统中可以稳定地打印出来，且接收者无法修改其内容。我们会发现图32-6中的PDF文档看起来很像图32-5中的RTF文档。

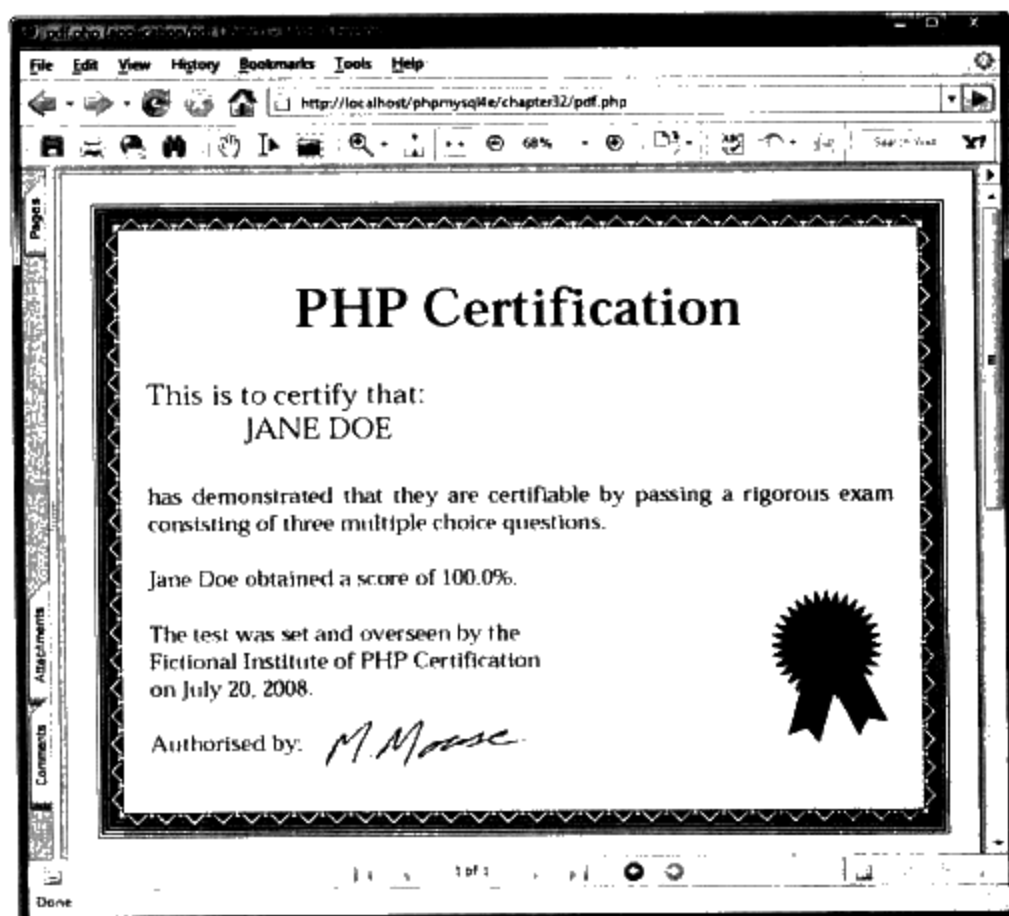


图32-6 pdf.php用PDF模板生成证书

该方法的一个问题是由于正则表达式的匹配使得该代码的运行速度非常慢。正则表达式比在RTF版本中用到的`str_replace()`函数要慢得多。

如果要在相同的服务器上匹配大量的占位符，或者试图生成许多这样的文档，可能需要寻求其他方法。而对于简单的模板，这个问题稍小一些。该文件中的大部分数据都是表示图像。

32.3.5 使用PDFlib生成PDF文档

PDFlib是为能通过Web生成动态PDF文档而设计的。严格地说，它并不是PHP的一部分，而是一个单独的库，它提供了大量函数供多种编程语言调用。绑定的编程语言有C、C++、Java、Perl、Python、Tcl和ActiveX/COM。

PDFlib由PDFlib GmbH正式支持。这就意味着你可以参考<http://www.php.net/en/manual/ref.pdf.php>处的PHP文档或者从<http://www.pdflib.com>下载正式文档。

32.3.6 使用PDFlib的一个“Hello World”程序

在拥有并以启用PDFlib的方式安装了PHP之后，可以用一个简单的程序（例如Hello World）来测试它，如程序清单32-5所示。

程序清单32-5 testpdf.php——通过PHP使用PDFlib编写的Hello World典型示例

```
<?php

// create a pdf document in memory
$pdf = pdf_new();
pdf_open_file($pdf, "");

pdf_set_info($pdf, "Author", "Luke Welling and Laura Thomson");
pdf_set_info($pdf, "Title", "Hello World (PHP)");
pdf_set_info($pdf, "Creator", "testpdf.php");
pdf_set_info($pdf, "Subject", "Test PDF");

// US letter is 11" x 8.5" and there are 72 points per inch
pdf_begin_page($pdf, 8.5*72, 11*72);

// add a bookmark
pdf_add_bookmark($pdf, "Page 1", 0, 0);

$font = pdf_findfont($pdf, "Times-Roman", "host", 0);
pdf_setfont($pdf, $font, 24);
pdf_set_text_pos($pdf, 50, 700);

// write text
pdf_show($pdf, "Hello, world!");
pdf_continue_text($pdf, "(says PHP)");

// end the document
pdf_end_page($pdf);
pdf_close($pdf);
$data = pdf_get_buffer($pdf);

// generate the headers to help a browser choose the correct application
header("Content-Type: application/pdf");
```

```

header('Content-Disposition: inline; filename=testpdf.pdf');
header('Content-Length: ' . strlen($data));

// output PDF
echo $data;

?>

```

如果该脚本执行失败，最有可能看到的出错信息如下所示：

```

Fatal error: Call to undefined function pdf_new()
in C:\Program Files\Apache Software
Group\Apache2.2\htdocs\phpmysql4e\chapter32\testpdf.php on line 4

```

这表明我们没有将PDFlib扩展库的支持编译到PHP中。

安装过程相当直观，但一些细节的修改将根据所使用的PHP和PDFlib的不同而有所不同。一个寻求具体细节意见的好地方就是注释过的PHP手册中PDFlib页上用户添加的注释。

当准备好了该脚本并在系统中运行后，可以开始了解它如何工作了。如下所示的代码：

```

$pdf = pdf_new();
pdf_open_file($pdf, "");

```

将在内存中初始化一个PDF文档。

pdf_set_info()函数允许使用一个“主题；标题；创建者；作者”这些关键字列表以及对应的用户自定义域来标记一个文档。

在这里，我们设置了创建者、作者和标题以及主题。注意这6个信息字段都是可选的：

```

pdf_set_info($pdf, "Author", "Luke Welling and Laura Thomson");
pdf_set_info($pdf, "Title", "Hello World (PHP)");
pdf_set_info($pdf, "Creator", "testpdf.php");
pdf_set_info($pdf, "Subject", "Test PDF");

```

一个PDF文档由很多页组成。要开始一个新页面，必须调用pdf_begin_page()函数。pdf_begin_page()函数的参数除了要求有pdf_open()函数返回的标识符外，还要求有页面大小的参数。文档中的每一页都可以有不同的尺寸，但是应该尽量使用相同的页面大小，除非有充足的理由使用不同大小的页面。

PDFlib是以点(point)为度量单位的，无论是对于页面大小还是在页面中定位坐标位置都是这样的。作为参考，A4纸的大小大概是595×842点，美国信函纸大小为612×792。如下代码行：

```
pdf_begin_page($pdf, 8.5*72, 11*72);
```

表示在文档中创建了一个美国信函大小的页面。

一个PDF文档不只限于具有打印功能的文档。在文档中，还可以包含许多PDF特征，比如超链接和书签等。pdf_add_outline()函数可以将书签添加到文档提纲中。文档中的书签将以单独的窗格格式出现在Acrobat阅读器中，这样允许我们直接跳到重要的章节。

如下代码行：

```
pdf_add_bookmark($pdf, 'Page 1', 0, 0);
```

将增加一条带有标签Page1的大纲记录，该记录指向当前页。

系统中可用字体将根据操作系统不同而不同，甚至对于不同的机器也不一样。为了保证结果的一致性，PDF阅读器包含了一组核心字体。这14个核心字体分别是：

- Courier
- Courier-Bold
- Courier-Oblique
- Courier-BoldOblique
- Helvetica
- Helvetica-Bold
- Helvetica-Oblique
- Helvetica-BoldOblique
- Times-Roman
- Times-Bold
- Times-Italic
- Times-BoldItalic
- Symbol
- ZapfDingbats

文档中也可以包含以上字体集合以外的字体。但这可能会增加文件的大小，而且可能不被某些特殊的字体许可所支持。我们可以选择字体、大小和字符编码方式，如下所示：

```
$font = pdf_findfont($pdf, 'Times-Roman', 'host', 0);
pdf_setfont($pdf, $font, 24);
```

字体大小是按点计算的。在这个例子中，我们选择了主机字符编码。这种编码允许的取值有winansi、buitin、macroman、ebcdic或host。不同取值的含义如下所示：

- winansi——由Microsoft在ISO8859-1字符集的基础上再添加的特殊符号，比如欧元符号。
- buitn——使用字体内置的编码。通常，用于非拉丁字体和符号。
- macroman——Mac Roman编码。默认的Macintosh字符集。
- ebcdic——用在IBM AS/400系统中的EBCDIC。
- host——基于Mac的系统自动选择macroman，基于EBCDIC的系统选择ebcdic，所有其他系统选择winansi。

如果不需要包含特殊字符，那么这个编码选项并不重要。

PDF文档不像HTML文档或字处理器文档。在默认情况下，文本并不是简单地从左上方开始，接下来是其他行。我们必须选择在哪里放置每一行文本。正如前面已经提到的，PDF用点来指定位置。原点（X，Y坐标为[0，0]）位于页面的左下角。

假定页面大小是612×792点，点（50，700）大约距离页面的左边2/3英寸，距离顶部1/3英寸。要将文本位置设置到该点，可以使用如下所示的代码：

```
pdf_set_text_pos($pdf, 50, 700);
```

最后，设置好页面后，可以在上面书写文本了。我们用pdf_show()函数在当前点以当前字体添加文本。

如下代码行：

```
pdf_show($pdf, 'Hello,world!');
```

将“Hello world!”添加到文档中。

要移到下一行并接着写入更多文本，可以使用pdf_continue_text()函数。要在文档中增加字符串“(says PHP)”，可以使用如下所示语句：

```
pdf_continue_text($pdf, '(says PHP)');
```

该字符串的确切显示位置将由所选的字体及大小来确定。

如果还要插入其他行、短语和段落，pdf_show_boxed()函数非常有用。它允许声明一个文本框并在其中输入文本。

完成将元素添加到页面后，要以如下方式调用pdf_end_page()函数：

```
pdf_end_page($pdf);
```

生成整个PDF文档后，还需要调用pdf_close()函数关闭该文档。当要生成一个文件时，也需要关闭这个文件。

如下代码行：

```
pdf_close($pdf);
```

将完成Hello World整个文档。

现在，我们可以将所完成的PDF文档发送给浏览器：

```
$data = pdf_get_buffer($pdf);

// generate the headers to help a browser choose the correct application
header('Content-Type: application/pdf');
header('Content-Disposition: inline; filename=testpdf.pdf');
header('Content-Length: ' . strlen($data));

// output PDF
echo $data;
```

我们也可以将该数据写入磁盘。通过以文件名称作为pdf_open_file()函数的第二个参数，PDFlib允许将数据写入磁盘中。

请注意，在PDFlib的某些版本中，PHP手册中给出的某些可选的PDFlib函数参数是必需的。关于证书的文档相对要复杂一些，它包含边界、一幅向量图和一幅位图。使用其他两种技巧，我们可以用字处理器来添加这些元素。如果使用PDFlib，我们则必须手动添加。

32.3.7 用PDFlib生成证书

在这个项目中，为了使用PDFlib，我们需要做一些折中。虽然精确复制前面用到的证书是肯定可以的，但仍需要努力地手动生成及定位每一个元素，而不能使用像Microsoft Word这样

的工具来协助编排文档。

我们使用与前面相同的文本，包括红色的圆花饰及位图签名，但我们不打算复制复杂的边界。该脚本的完整代码如程序清单32-6所示。

程序清单32-6 pdflib.php——用PDFlib生成证书

```
<?php
// create short variable names
$name = $_POST['name'];
$score = $_POST['score'];

if(!$name || !$score) {
    echo "<h1>Error:</h1>";
    <p>This page was called incorrectly</p>";
} else {
    $date = date('F d, Y');

    // create a pdf document in memory
    $pdf = pdf_new();
    pdf_open_file($pdf, "");

    // set up name of font for later use
    $fontname = "Times-Roman";

    // set up the page size in points and create page
    // US letter is 11" x 8.5" and there are approximately
    // 72 points per inch
    $width = 11*72;
    $height = 8.5*72;
    pdf_begin_page($pdf, $width, $height);

    // draw our borders
    $inset = 40; // space between border and page edge
    $border = 10; // width of main border line
    $inner = 2; // gap within the border

    //draw outer border
    pdf_rect($pdf, $inset-$inner,
              $inset-$inner,
              $width-2*($inset-$inner),
              $height-2*($inset-$inner));
    pdf_stroke($pdf);

    //draw main border $border points wide
    pdf_setlinewidth($pdf, $border);
    pdf_rect($pdf, $inset+$border/2,
              $inset+$border/2,
              $width-2*($inset+$border/2),
```



```

        $height-2*($inset+$border/2));
pdf_stroke($pdf);
pdf_setlinewidth($pdf, 1.0);

// draw inner border
pdf_rect($pdf, $inset+$border+$inner,
        $inset+$border+$inner,
        $width-2*($inset+$border+$inner),
        $height-2*($inset+$border+$inner));
pdf_stroke($pdf);

// add heading
$font = pdf_findfont($pdf, $fontname, 'host', 0);
if ($font) {
    pdf_setfont($pdf, $font, 48);
}
$startx = ($width - pdf_stringwidth($pdf, 'PHP Certification',
        $font, 12))/2;
pdf_show_xy($pdf, 'PHP Certification', $startx, 490);

// add text:
$font = pdf_findfont($pdf, $fontname, 'host', 0);
if ($font) {
    pdf_setfont($pdf, $font, 26);
}
$startx = 70;
pdf_show_xy($pdf, 'This is to certify that:', $startx, 430);
pdf_show_xy($pdf, strtoupper($name), $startx+90, 391);

$font = pdf_findfont($pdf, $fontname, 'host', 0);
if ($font) {
    pdf_setfont($pdf, $font, 20);

pdf_show_xy($pdf, 'has demonstrated that they are certifiable',
        $startx, 340);
pdf_show_xy($pdf, 'by passing a rigorous exam', $startx, 340);
pdf_show_xy($pdf, 'consisting of three multiple choice questions.',
        $startx, 310);

pdf_show_xy($pdf, "$name obtained a score of $score".'%', $startx, 260);

pdf_show_xy($pdf, 'The test was set and overseen by the ', $startx, 210);
pdf_show_xy($pdf, 'Fictional Institute of PHP Certification',
        $startx, 180);
pdf_show_xy($pdf, 'on $date.', $startx, 150);
pdf_show_xy($pdf, 'Authorised by:', $startx, 100);

// add bitrap signature image

```

```
$signature = pdf_load_image($pdf, 'png', '/Program Files/Apache Software
Foundation/Apache2.2/htdocs/phpmysql4e/chapter32/signature.png', '');
pdf_fit_image($pdf, $signature, 200, 75, '');
pdf_close_image($pdf, $signature);

// set up colors for rosette
pdf_setcolor ($pdf, 'both', 'cmyk', 43/255, 49/255, 1/255, 67/255); // dark
blue
pdf_setcolor ($pdf, 'both', 'cmyk', 1/255, 1/255, 1/255, 1/255); // black

// draw ribbon 1
pdf_moveto($pdf, 630, 150);
pdf_lineto($pdf, 610, 55);
pdf_lineto($pdf, 632, 69);
pdf_lineto($pdf, 646, 49);
pdf_lineto($pdf, 666, 150);
pdf_closepath($pdf);
pdf_fill($pdf);

// outline ribbon 1
pdf_moveto($pdf, 630, 150);
pdf_lineto($pdf, 610, 55);
pdf_lineto($pdf, 632, 69);
pdf_lineto($pdf, 646, 49);
pdf_lineto($pdf, 666, 150);
pdf_closepath($pdf);
pdf_stroke($pdf);

// draw ribbon 2
pdf_moveto($pdf, 660, 150);
pdf_lineto($pdf, 680, 49);
pdf_lineto($pdf, 695, 69);
pdf_lineto($pdf, 716, 55);
pdf_lineto($pdf, 696, 150);
pdf_closepath($pdf);
pdf_fill($pdf);

// outline ribbon 2
pdf_moveto($pdf, 660, 150);
pdf_lineto($pdf, 680, 49);
pdf_lineto($pdf, 695, 69);
pdf_lineto($pdf, 716, 55);
pdf_lineto($pdf, 696, 150);
pdf_closepath($pdf);
pdf_stroke($pdf);
pdf_setcolor ($pdf, 'both', 'cmyk', 1/255, 81/255, 81/255, 20/255); // red
```

```

//draw rosette
draw_star(665, 175, 32, 57, 10, $pdf, true);

//outline rosette
draw_star(665, 175, 32, 57, 10, $pdf, false);

// finish up the page and prepare to output
pdf_end_page($pdf);
pdf_close($pdf);
$data = pdf_get_buffer($pdf);

// generate the headers to help a browser choose the correct application
header('Content-type: application/pdf');
header('Content-disposition: inline; filename=test.pdf');
header('Content-length: ' . strlen($data));

// output PDF
echo $data;
}

function draw_star($centerx, $centery, $points, $radius,
                  $point_size, $pdf, $filled) {
    $inner_radius = $radius-$point_size;

    for ($i = 0; $i<-$points*2; $i++) {
        $angle= ($i*2*pi())/($points*2);

        if($i%2) {
            $x = $radius*cos($angle) + $centerx;
            $y = $radius*sin($angle) + $centery;
        } else {
            $x = $inner_radius*cos($angle) + $centerx;
            $y = $inner_radius*sin($angle) + $centery;
        }

        if($i==0) {
            pdf_moveto($pdf, $x, $y);
        } else if ($i==-$points*2) {
            pdf_closepath($pdf);
        } else {
            pdf_lineto($pdf, $x, $y);
        }
    }

    if($filled) {
        pdf_fill_stroke($pdf);
    } else {
        pdf_stroke($pdf);
    }
}

```

使用以上脚本生成的证书如图32-7所示。可以看到，它与其他证书非常相似，只是边界简单一些，星状图案看起来稍有不同。这是因为我们这里是将它绘制到文档中的，而不是使用一个已存在的剪切美术文件。

现在，我们介绍一下该脚本与前面的例子不同的部分。

由于访问者需要在证书中获得关于自己的资料，因此我们将在内存中创建该文档而不是在文件中。如果将它写入一个文件，需要考虑能够生成唯一文件名的机制，并且能够防止人们偷偷进入其他人的证书，以及选择一种删除旧文件以释放服务器上硬盘空间的机制。

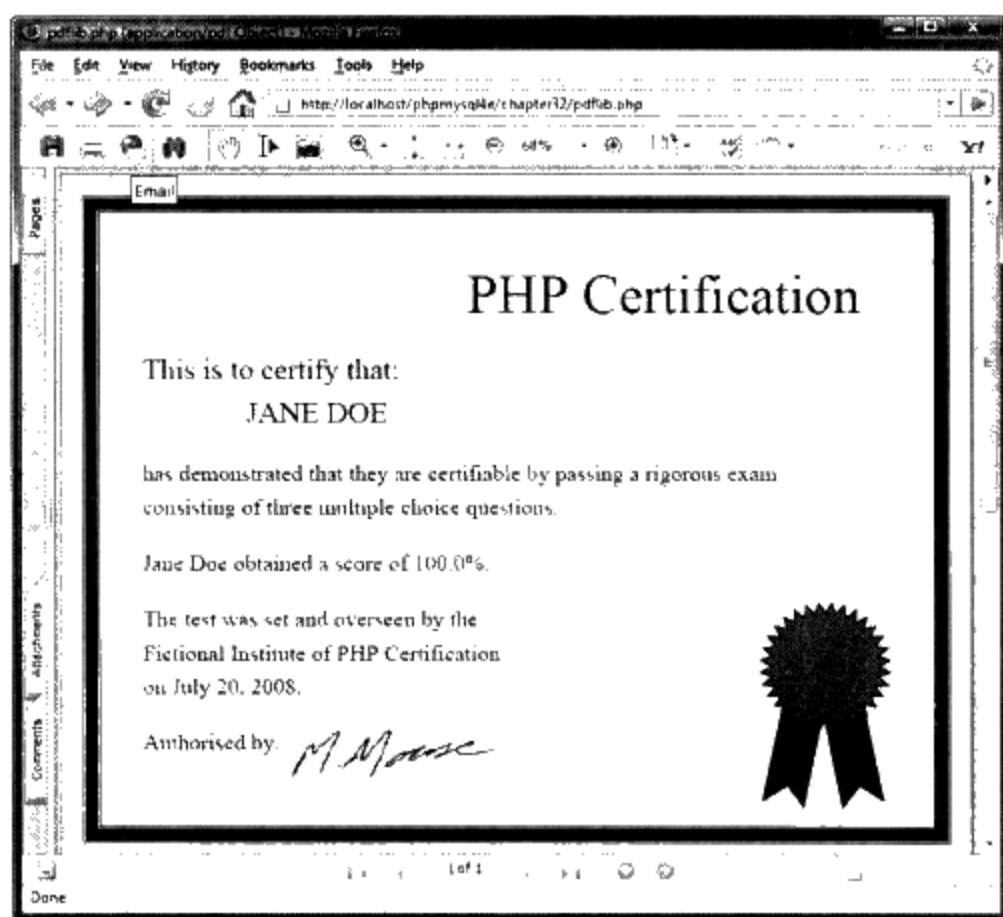


图32-7 pdflib.php将证书画到一个PDF文档中

为了在内存中创建一个文档，我们在调用了`pdf_open_file()` 函数后再调用不带参数的`pdf_new()`函数。如下所示：

```
$pdf = pdf_new();
pdf_open_file($pdf, "");
```

简化的边界将包含3个条纹：一条粗边框和两条细边框，其中一个在主边框里面，另一个在外面。我们将以矩形方式绘制它们。

为了以这种方法确定边界的位置以便轻易调整页面的大小或边界的外观，我们把所有边界位置建立在已有的变量`$width`和`$height`，以及新变量`$inset`、`$border`和`$inner`的基础上。用`$inset`指定位于页面边缘的边界有多少点宽，`$border`来指定主边框的厚度，以及

使用`$inner`来指定主边框与两条细边框之间的缝隙有多宽。

如果我们使用过其他的图形API来绘制图形，在使用PDFlib绘制图形时，就不会有太多的惊讶。第22章对于这项工作是有帮助的，因为用gd库绘制图形与用PDFlib绘制图形非常相似。

细边框是比较简单的。为了创建一个长方形，可以使用`pdf_rect()`函数。该函数要求提供如下参数：PDF文档标识符、长方形的左下角的X和Y坐标以及宽和高。因为我们希望排列能够灵活些，因此用已设置的变量来计算这些：

```
pdf_rect($pdf, $inset-$inner,
        $inset-$inner,
        $width-2*($inset-$inner),
        $height-2*($inset-$inner));
```

对`pdf_rect()`函数的调用可以设置一块长方形的区域。为了绘制出这个形状，需要调用`pdf_stroke()`函数。如下所示：

```
pdf_stroke($pdf);
```

为了绘制主边框，需要指定线宽。在默认的情况下，线宽为1个点。如下代码中，对`pdf_setlinewidth()`函数的调用将它设为`$border`（在这个例子中，设置为10）点：

```
pdf_setlinewidth($pdf, $border);
```

在设置好宽度后，我们再次调用`pdf_rect()`函数来创建一个矩形并调用`pdf_stroke()`函数将它绘制出来：

```
pdf_rect($pdf, $inset+$border/2,
        $inset+$border/2,
        $width-2*($inset+$border/2),
        $height-2*($inset+$border/2));
pdf_stroke($pdf);
```

在绘制好一条宽线后，我们必须记住用如下所示代码将宽度重新设置为1：

```
pdf_setlinewidth($pdf, 1.0);
```

我们使用`pdf_show_xy()`函数来定位证书中每一行文本的位置。对于大多数文本行，我们使用可配置的左边距（`$startx`）作为X坐标。如果想要将标题置于页面的中央，那么需要知道它的宽度来确定左面的位置。我们可以使用`pdf_stringwidth()`函数来得到其宽度。如下所示的函数调用：

```
pdf_stringwidth($pdf, 'PHP Certification', $font, 12);
```

将以当前的字体与大小返回字符串“PHP Certification”的宽度。

对于其他版本的证书，需要嵌入一个扫描后的位图签名。如下所示的3个语句：

```
$signature = pdf_load_image($pdf, 'png', '/Program Files/Apache Software
Foundation/Apache2.2/docs/phpmysql4e/chapter32/signature.png', '');
pdf_fit_image($pdf, $signature, 200, 75, '');
pdf_close_image($pdf, $signature);
```

将打开一个包含签名的GIF文件，将图像添加到页面的指定位置并关闭该GIF文件。我们也

可以使用其他类型的文件。

提示 当通过pdf_load_image()函数载入一个图像时,使用该文件的完整路径。在这个例子中,signature.png的完整路径是Windows系统的路径。

使用PDFlib添加到证书中的元素最困难的是那个圆饰物。我们不能自动地打开已有的包含该圆饰物的Windows meta文件,但可以任意绘制任何想要的图形。

为了绘制一个填充的图形,例如其中一个锯齿,我们编写了如下所示的代码。在这里,条纹和直线的颜色都设为黑色,填充或内部颜色设为暗蓝色:

```
pdf_setcolor($pdf, 'fill', 'rgb', 0, 0, .4, 0); // dark blue
pdf_setcolor($pdf, 'stroke', 'rgb', 0, 0, 0, 0); // black
```

这里,我们建立了一个五边形作为其中一个锯齿并填充它:

```
pdf_moveto($pdf, 630, 150);
pdf_lineto($pdf, 610, 55);
pdf_lineto($pdf, 632, 69);
pdf_lineto($pdf, 646, 49);
pdf_lineto($pdf, 666, 150);
pdf_closepath($pdf);
pdf_fill($pdf);
```

就像多边形的轮廓一样,我们需要用相同的方法再一次设置,但是这次是调用pdf_stroke()函数而不是pdf_fill()函数。

由于多点星形是一种复杂形状,我们编写了一个函数来计算该形状沿途路径的位置。该函数叫做draw_star(),它要求中点的X和Y坐标、角点的数目、半径、角点的长度、PDF文件标识符和一个布尔值来表示该星形应填充还是仅绘制其轮廓。

draw_star()函数用到了一些基本的三角学来计算形成一颗星的一系列点的位置。对于星形具有的每个点,我们在星形的半径上找出一个点,在外圆周的更小的圆\$point_size上再找出一个点,并在它们之间画一条直线。一个值得注意的地方是PHP的三角函数如cos()和sin()是以弧度而不是以度为单位的。

利用一个函数和一些数学知识,我们可以精确地生成一个复杂的重复性图形。如果希望以复杂的模式画页面边界,也可以使用相似的方法。

当产生了所有页面元素后,需要关闭该页面及文档。

32.4 处理标题的问题

在所有这些脚本中,需要注意的一个细小问题是需要告诉浏览器将发送给它什么类型的数据。我们通过发送一个内容类型的HTTP标题来实现该操作,如下所示:

```
header('Content-type: application/msword');
```

或者

```
header('Content-type: application/pdf');
```

一个值得注意的地方是浏览器处理这些标题的方法并不一致。特别是Internet Explorer经常

选择忽略MIME类型，而试图自动检测文件类型（这个特殊的问题似乎已经在Internet Explorer的最新版本中得到解决，因此，如果还遇到这样的问题，最简单的解决方法是升级浏览器）。

一些标题似乎会在会话控制标题中出现。如今，对于这个问题，已经有几种解决方法。我们发现通过使用GET参数而不是POST或会话变量参数可以避免该问题。

另一种方法，正如Hello World PDFlib示例中一样，不使用内嵌的PDF，而让用户下载它。

编写两种不同版本的代码也可以避免此类问题，一个适用于Netscape浏览器，另一个适用于Internet Explorer浏览器。

32.5 扩展该项目

显然，在测试中添加一些更现实的评测任务可以扩展该项目，但是，在本书中，该项目只是作为一个发布自己文档的方法示例。

通过在线传送方式发布的自定义文档可以包括法律文档、部分填充的订单或申请表单，以及政府部门需要的表单。

32.6 下一章

在下一章中，我们将介绍PHP 5的XML新功能以及使用PHP通过REST和SOAP协议访问Amazon的Web服务API。