

第 4 章 FPDF 类库应用

PDF 文档格式是当前流行的电子文档与电子表格的一种标准格式。PDF 提供了完善的压缩处理，无论创建者创建的 PDF 文档使用了什么字体、使用了什么样的图片或者版式设计，浏览者都可以通过免费的 Adobe Reader 对其进行阅读。本章将要介绍的是一个可以用 PHP 创建 PDF 文档的类库 FPDF。

4.1 FPDF 简介

FPDF 是“Free PDF”的缩写，即“免费 PDF”的意思。FPDF 类库提供了基本的 PDF 创建功能，并且其源码和使用授权都是免费的。

FPDF 可以支持汉语等东方语系，但是需要对相应字体进行配置。FPDF 中的几个基本的操作是 Open 函数、AddPage 函数、SetFont 函数、Cell 函数和 Output 函数，这几个基本函数就可以实现简单的 PDF 输出。

4.2 FPDF 的安装与配置

FPDF 的下载可以通过访问 <http://www.fpdf.org/> 来得到，将下载的文件直接解压到某一文件夹即可。为了方便起见，可以将下载的文件解压到 PEAR 的目录下，这样方便使用。

有一个需要配置的地方是解压缩后文件中的 font 文件夹。一般需要将该文件夹移动到用户站点与要调用 FPDF 类库代码的同一目录下，因为调用类库的时候要首先指明 font 文件夹所在位置。

4.3 FPDF 创建 PDF 文档

前面介绍了如何下载和安装 FPDF，以及如何配置 font 文件夹所在位置。本节将介绍一些 FPDF 的基本用法。

4.3.1 FPDF 创建 Hello World

前面介绍过，在 FPDF 中，PDF 文档的创建主要通过 Open 函数、AddPage 函数、SetFont 函数、Cell 函数和 Output 函数来实现的。这些函数都是 FPDF 对象的方法。创建一个 FPDF 对象的语法格式如下所示。

```
$pdf = new FPDF([string page-orientation [, string measure-unit [, string page-format]])
```

其中，page-orientation 用于表示创建的 PDF 文档是横向还是竖向的。page-orientation 可用的值有以下两种。

□ P: 表示放纸方向为竖向。

□ L: 表示放纸方向为横向。

measure-unit 用于表示文档中位置的计量单元。measure-unit 可用的值有以下四种。

□ pt: 点

□ mm: 毫米

□ cm: 厘米

□ in: 英寸

page-format 用于表示创建的 PDF 文档的纸张类型。page-format 可用的值可以是用于表示纸张类型的字符串, 例如“A4”“A5”“Letter”等。也可以是一个包含有两个元素的二维数组来直接指明纸张的大小。

Open 函数用于标示开始创建 PDF 文档, 语法格式如下所示。

```
$pdf->Open()
```

AddPage 函数用于为 PDF 文档添加一个新页, 语法格式如下所示。

```
$pdf->AddPage([string page-orientation])
```

其中, page-orientation 用于表示创建的 PDF 文档是横向还是竖向的。与前面介绍的创建对象时的使用方法相同。

SetFont 函数用于设置当前使用的字体, 语法格式如下所示。

```
$pdf->SetFont(string font [, string style [, float size]]);
```

其中, font 用于表示字体。style 用于表示样式, style 可用的值可以是以下三种。如果没有指定 style 的值, 则默认为普通的形式。

□ B: 粗体

□ I: 斜体

□ U: 下滑线

size 用于表示字体的大小。如果不指定则取默认值, 为 12pt。

Cell 函数用于为当前 PDF 文档增加一个单元格, 常用的语法格式如下所示。

```
$pdf->Cell(float width, float hight, string str, int bolder)
```

其中, width 表示增加的单元格宽度, hight 表示增加的单元格高度, str 表示要放置在单元格中的文本字符串, bolder 表示单元格的边框。需要注意的是如果将 width 或者 height 设置成 0, 并不是表示该单元格不存在, 而是表示没有设置。

Output 函数用于为输出 PDF 文档, 语法格式如下所示。

```
$pdf->Output([string filename [, bool download]])
```

其中, filename 表示要存储的文件名。如果不指定文件名, 则浏览时会在 IE 中直接打开。Download 表示是否一定要用户下载查看, 默认为 false。以下代码输出一个“Hello World”字符串。

```
<?php
define('FPDF_FONTPATH','font/');           //定义 font 文件夹所在路径
require_once('fpdf/fpdf.php');              //包含 fpdf 类库文件
$pdf=new FPDF('P', 'mm', 'A4');             //创建新的 FPDF 对象, 竖向放纸, 单位为毫米, 纸张大小 A4
$pdf->Open();                                //开始创建 PDF
$pdf->AddPage();                              //增加一页
$pdf->SetFont('Courier','I',20);              //设置字体样式
$pdf->Cell(0,0,'Hello World!');               //增加一个单元格
$pdf->Output();                               //输出 PDF 到浏览器
?>
```

运行结果如图 4-1 所示。

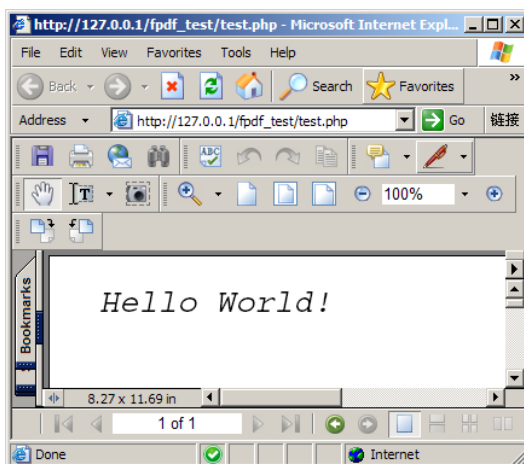


图 4-1 使用 FPDF 输出的 Hello World

在实际应用中，往往将创建的文档保存到服务器并提示用户下载。这样可以将上面代码中的 Output 语句修改如下。

```
$pdf->Output('test.pdf', true);
```

修改后的运行结果如图 4-2 所示。

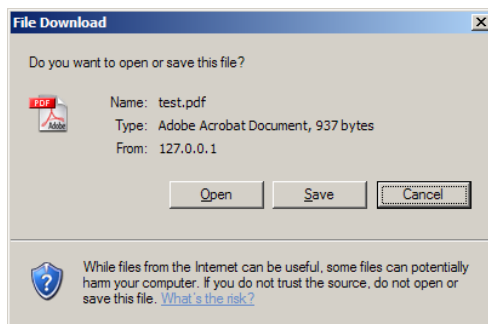


图 4-2 使用 FPDF 创建 Hello World 文档的下载对话框

4.3.2 在 PDF 中插入图片

前面介绍了 Cell 函数，该函数主要用于向 PDF 文档中插入文本。通常，PDF 文档是由文本和图片共同组成的。FPDF 中使用 Image 函数向 PDF 中插入图片，其语法格式如下所示。

```
Image(string file, float x, float y, float width, float height)
```

其中，x 和 y 表示图片所在的坐标，width 和 height 表示图片的宽度和高度。需要注意的是如果想让图片保持原来大小，只需要将 width 和 height 设置成 0 即可。以下代码是一个在 PDF 中插入图片的例子。

```
<?php
define('FPDF_FONTPATH','font/');           //定义 font 文件夹所在路径
require_once('fpdf/fpdf.php');              //包含 fpdf 类库文件
$pdf=new FPDF('P', 'mm', 'A4');             //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小 A4
$pdf->Open();                               //开始创建 PDF
$pdf->AddPage();                            //增加一页
$pdf->SetFont('Courier','I',20);            //设置字体样式
$pdf->Image('sight.jpg',20,20,0,0);         //增加一张图片，文件名为 sight.jpg
$pdf->Output();                             //输出 PDF 到浏览器
?>
```

运行结果如图 4-3 所示。

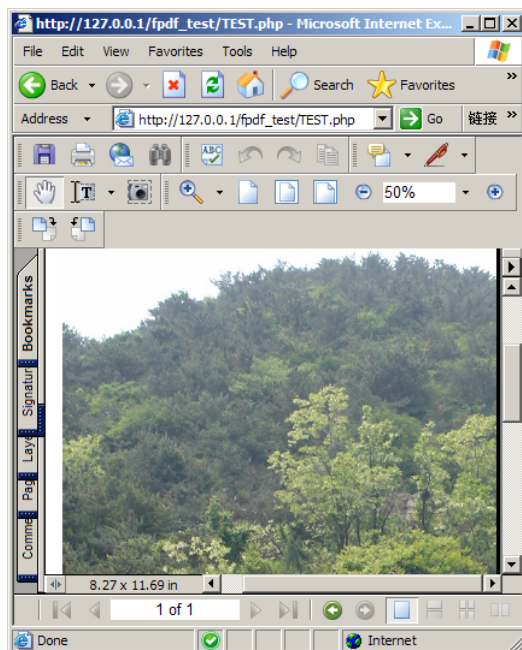


图 4-3 向 PDF 中插入图片

4.3.3 页眉与页脚

页眉与页脚是通过重写 FPDF 类中的 Header 方法和 Footer 方法来实现的。在原有的 FPDF 类中存在这两个方法，但是方法体没有任何内容。在输出 PDF 文档时，这两个方法将被自动调用。

在介绍如何创建页眉与页脚前，先来介绍一个常在页眉或页脚中用到的方法——PageNo()，其语法格式如下所示。

```
int PageNo();
```

该方法返回当前页码。这个方法在页眉或页脚中常被用到，用于显示当前页码。以下代码为前面的“Hello World!”程序产生的文档增加了页眉与页脚。

```
<?php
define('FPDF_FONTPATH','font/');           //定义 font 文件夹所在路径
require_once('fpdf/fpdf.php');              //包含 fpdf 类库文件

class PDF extends FPDF
{
    function Header()                        //设置页眉
    {
        $this->SetFont('Arial','B',15);      //设置页眉字体
        $this->Cell(80);                     //移动单元格
        $this->Cell(30,10,'Title');          //写入页眉文字
        $this->Ln(20);                       //换行
    }

    function Footer()                        //设置页脚
    {
```

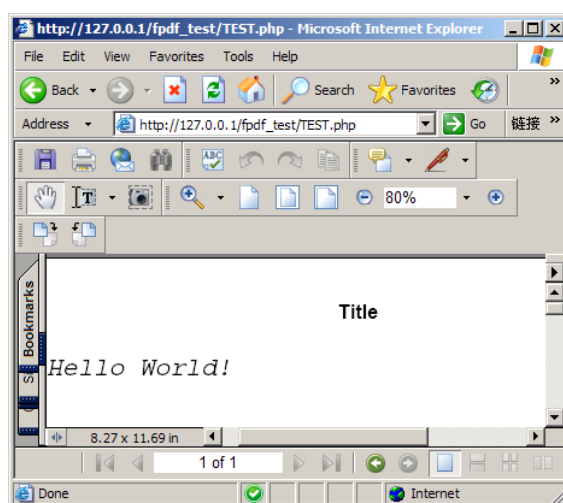
```

$this->SetY(-15);           //设置页脚所在位置
$this->SetFont('Arial','I',8); //设置页脚字体
$this->Cell(0,10,'Page - '.$this->PageNo()); //输出当前页码作为页脚内容
}
}

$pdf=new PDF('P', 'mm', 'A4'); //创建新的 FPDF 对象, 竖向放纸, 单位为毫米, 纸张大小 A4
$pdf->Open();                 //开始创建 PDF
$pdf->AddPage();              //增加一页
$pdf->SetFont('Courier','I',20); //设置字体样式
$pdf->Cell(0,0,'Hello World!'); //增加一个单元格
$pdf->Output();               //输出 PDF 到浏览器
?>

```

运行结果如图 4-4 和 4-5 所示。



4-4 页眉

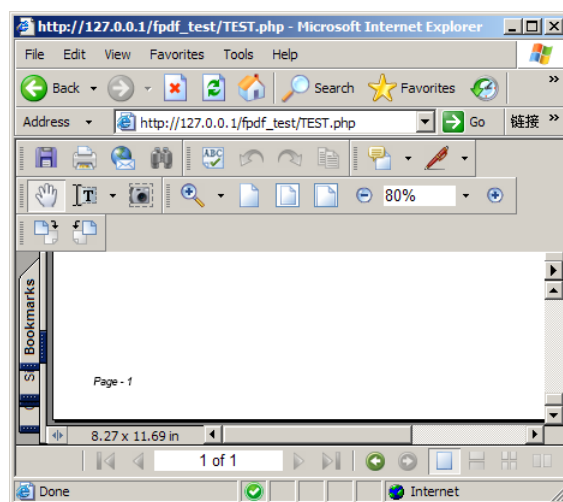


图 4-5 页脚

需要注意得是在设置页脚的时候, 方法 `SetY` 是必不可少的, 其语法格式如下所示。

```
SetY(float y);
```

这里的 `y` 指代页面上的 Y 坐标, 单位为毫米。如果 `y` 为负数, 则表示从页面底部向上的距离。在前

面的例子中，设置 y 为 -15，即页脚所在位置距页面底部 15 毫米。

4.3.4 绘制表格

使用 FPDF 绘制表格的方法与输出文字的方法类似，在输出文字时往往指定 Cell 方法的边框参数为 0，即不输出边框。在绘制表格时往往将其设置成大于 0 的整数，用于表示边框。

以下代码将前面的“Hello World!”程序进行改写，输出边框。

```
<?php
define('FPDF_FONTPATH','font');           //定义 font 文件夹所在路径
require_once('fpdf/fpdf.php');             //包含 fpdf 类库文件
$pdf=new FPDF('P', 'mm', 'A4');            //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小 A4
$pdf->Open();                              //开始创建 PDF
$pdf->AddPage();                            //增加一页
$pdf->SetFont('Courier','I',20);           //设置字体样式
$pdf->Cell(60,10,'Hello World!',1);        //增加一个单元格 边框为 1
$pdf->Output();                             //输出 PDF 到浏览器
?>
```

输出结果如图 4-6 所示。

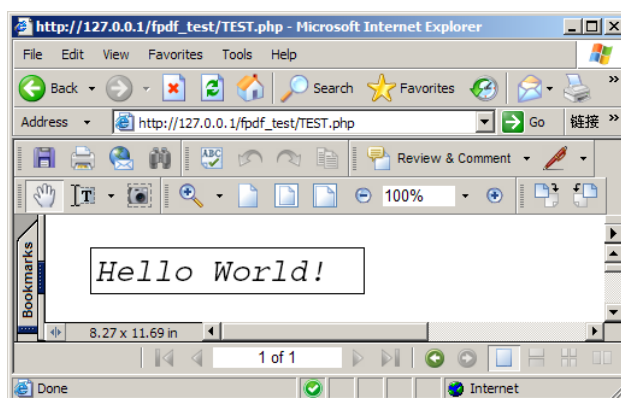


图 4-6 带边框的文字

将带边框的文字很好的组合起来，就构成了表格。以下代码是一个输出表格的例子。

```
<?php
define('FPDF_FONTPATH','font');           //定义 font 文件夹所在路径
require_once('fpdf/fpdf.php');             //包含 fpdf 类库文件
$pdf=new FPDF('P', 'mm', 'A4');            //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小 A4
$pdf->Open();                              //开始创建 PDF
$pdf->AddPage();                            //增加一页

$pdf->SetFont('Arial','',14);              //设置字体样式

$header=array('Name','Age','Sex','Salary'); //设置表头
$data=array();                             //设置表体
$data[0] = array('Simon','24','Male','5,000.00');
$data[1] = array('Elaine','25','Female','6,000.00');
$data[2] = array('Susan','25','Female','7,000.00');
$data[3] = array('David','26','Male','8,000.00');
```

```

$width=array(40,40,40,40);           //设置每列宽度

for($i=0;$i<count($header);$i++)     //循环输出表头
    $pdf->Cell($width[$i],6,$header[$i],1);
$pdf->Ln();

foreach($data as $row)                //循环输出表体
{
    $pdf->Cell($width[0],6,$row[0],1);
    $pdf->Cell($width[1],6,$row[1],1);
    $pdf->Cell($width[2],6,$row[2],1);
    $pdf->Cell($width[3],6,$row[3],1);
    $pdf->Ln();
}

$pdf->Output();                        //输出 PDF 到浏览器
?>

```

输出结果如图 4-7 所示。

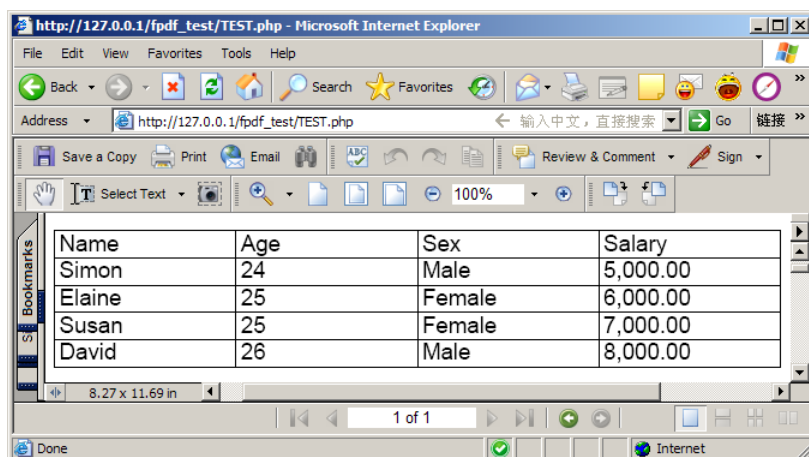


图 4-7 PDF 表格

4.4 为 FPDF 增加中文支持

前面介绍了使用 FPDF 创建 PDF 文档的方法，但是在默认情况下，FPDF 并不支持中文字符的输出，这在实际应用中就会有很大麻烦。以下代码试图使用一种 FPDF 支持的字体输出中文。

```

<?php
define('FPDF_FONTPATH','font/');           //定义 font 文件夹所在路径
require_once('fpdf/fpdf.php');              //包含 fpdf 类库文件
$pdf=new FPDF('P','mm','A4');               //创建新的 FPDF 对象，竖向放纸，单位为毫米，纸张大小 A4
$pdf->Open();                                //开始创建 PDF
$pdf->AddPage();                             //增加一页
$pdf->SetFont('Courier','I',20);             //设置字体样式
$pdf->Cell(0,0,'你好，FPDF');               //增加一个单元格并输出中文

```

```
$pdf->Output();           //输出 PDF 到浏览器
?>
```

运行结果如图 4-8 所示。

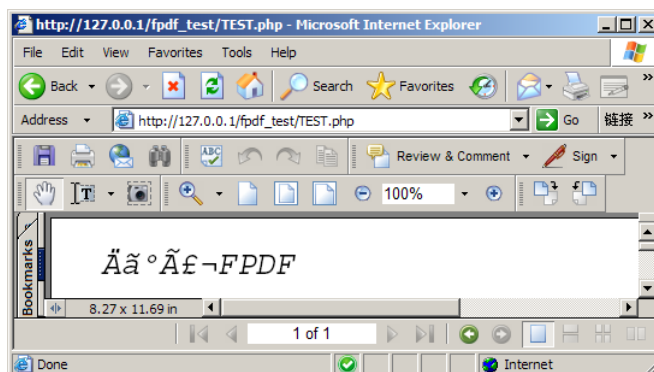


图 4-8 不成功的中文输出

可以看出，英文字符被成功的输出了，但是中文字符却变成了乱码。为了解决这个问题，一个常用的方法是下载一个流行的中文 FPDF 插件。该插件可以从 <http://www.litou.net/download.php?id=25> 下载到。

解压缩下载的文件，并将 `chinese.php` 放到网站的目录下。这时，就可以通过在代码中调用 `PDF_Chinese` 类来实现中文字符的输出，如以下代码所示。

```
<?php
require('chinese.php');
class PDF extends PDF_Chinese
{
function Header()           //设定页眉
{
    $this->SetFont('GB', '', 10);
    $this->Write(10, 'FPDF 中文测试');
    $this->Ln(20);
}

function Footer()           //设定页脚
{
    $this->SetY(-15);
    $this->SetFont('GB', '', 10);
    $this->Cell(0, 10, '第' . $this->PageNo() . '页');
}
}

$pdf=new PDF();              //创建 PDF 文档
$pdf->AddGBFont();
$pdf->Open();
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->SetFont('GB', 'l', 20);
$pdf->Cell(0, 10, '你好，FPDF'); //输出一段中文
$pdf->Output();
?>
```

运行结果如图 4-9 所示。

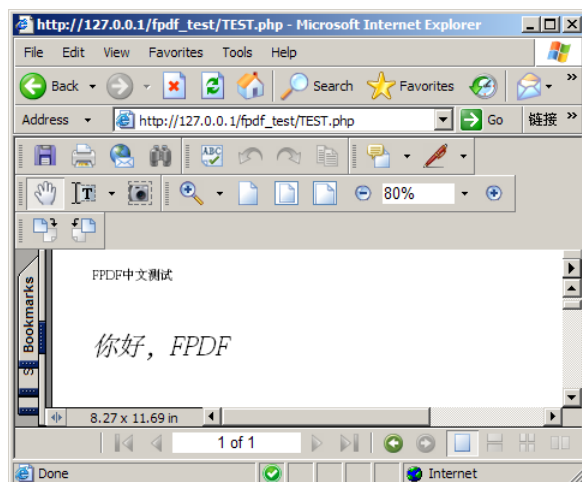


图 4-9 成功的中文输出

可以看到，这次 FPDF 成功的将中文输出了。

4.5 FPDF 的实际应用

前面对 FPDF 的一些使用方法进行了简要介绍，本节将以两个实例来说明 FPDF 在实际项目中的应用。

4.5.1 为文章系统创建 PDF 页面

在文章系统的设计中，往往使用一个专门的页面输出存储在数据库中的文章数据。以下代码通过对数据库的读取显示了文章的具体内容。

```
<?php
$conn = mysql_connect("localhost", "root", "");           //连接数据库
$colname_rs_article = $_GET['id'];                       //获取参数 id

mysql_select_db("cms", $conn);                           //执行 SQL
$query_rs_article = sprintf("SELECT * FROM articles WHERE article_id = %s", $colname_rs_article);
$rs_article = mysql_query($query_rs_article, $conn) or die(mysql_error());
$row_rs_article = mysql_fetch_assoc($rs_article);
$totalRows_rs_article = mysql_num_rows($rs_article);

function conv($Text)                                     //对返回文本进行处理
{
    $Text=htmlspecialchars($Text);                       //转换 HTML 关键字符
    $Text=nl2br($Text);                                   //转换换行符
    return $Text;
}
?>

<p align="center"><B><?php echo $row_rs_article['title']; ?></B></p>
<p align="center"><font size=2><?php echo $row_rs_article['author']; ?> | <a href="showpdf.php?id=<?php echo
```

```
$row_rs_article['article_id']; ?>">下载 PDF 文档</a></font></p>
<HR>
<p><?php echo conv($row_rs_article['content']); ?></p>
```

上面的代码通过获得浏览器地址栏中的传入参数，选择数据库中相应的数据并将结果显示在网页上，运行效果如图 4-10 所示。

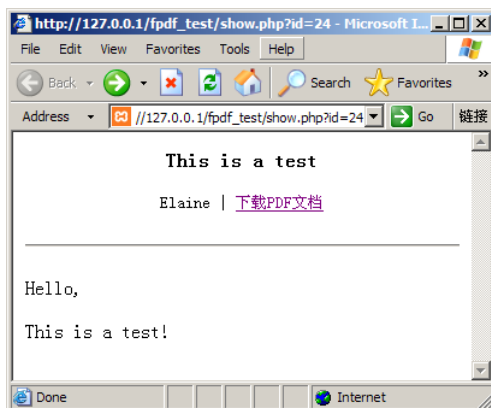


图 4-10 显示文章内容的网页

对于一些大型文章系统，例如论文系统，用户可能更希望将页面以 PDF 格式将文章保存到自己的硬盘上。以下代码使用相同的数据库读取代码将数据库中的数据输出到一个 PDF 文档中。

```
<?php
require('chinese.php');
class PDF extends PDF_Chinese
{
    function Header()                                //设置页眉
    {
        $this->SetFont('GB','',10);
        $this->Write(10,'文章系统 - XX 网站');
        $this->Ln(20);                                //换行
    }
    function Footer()                                //设置页脚
    {
        $this->SetY(-15);
        $this->SetFont('GB','',10);
        $this->Cell(0,10,'第'.$this->PageNo().'页');
    }
}
//主程序开始
$conn = mysql_connect("localhost", "root", "");      //连接数据库
$colname_rs_article = $_GET['id'];                  //获取参数 id

mysql_select_db("cms", $conn);                      //执行 SQL
$query_rs_article = sprintf("SELECT * FROM articles WHERE article_id = %s", $colname_rs_article);
$rs_article = mysql_query($query_rs_article, $conn) or die(mysql_error());
$row_rs_article = mysql_fetch_assoc($rs_article);
$totalRows_rs_article = mysql_num_rows($rs_article);
//开始创建 PDF 文档
$pdf=new PDF();
```

```

$pdf->AddGBFont();
$pdf->Open();
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->SetFont('GB','B',20);
$pdf->Cell(0,10,$row_rs_article['title']);           //输出文章标题
$pdf->Ln();                                           //换行
$pdf->SetFont('GB','',10);
$pdf->Cell(0,10,$row_rs_article['author']);          //输出文章作者
$pdf->Ln();
$pdf->SetFont('GB','',12);
$content = $row_rs_article['content'];
while($content != "")                               //循环逐页将文章内容写入 PDF
{
    $length = strlen($content);                     //获取文章长度
    $output = substr($content, 0, 1024);             //获取本页输出内容，每 1024 个字符为 1 页
    $pdf->Cell(0,10,$output);                        //输出文章内容
    $content = substr($content, 1024, $length);      //获取剩余未输出内容
    $pdf->AddPage();                                  //换页
}
$pdf->Output($row_rs_article['title'].".pdf", true);  //输出 PDF 文件，文件名为文章标题
?>

```

将上面的代码以 showpdf.php 文件名保存，单击图 4-10 中的【下载 PDF 文档】的链接，就可以弹出文件下载的对话框。打开下载后的文件如图 4-11 所示。

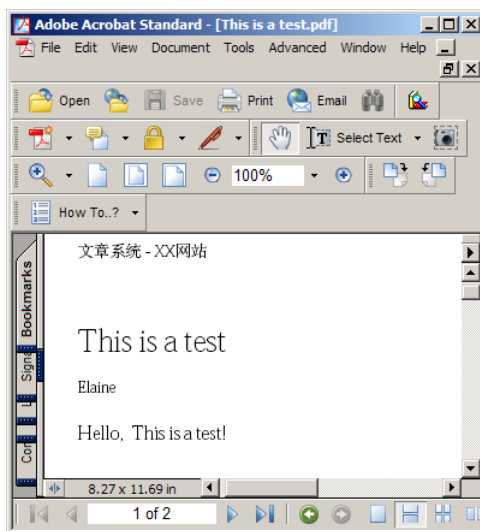


图 4-11 显示文章内容的网页的 PDF 文档

可以看到，图 4-11 中所示的 PDF 文档显示了与图 4-10 中所示网页相同的内容。需要注意的是由于 PDF 文档是按照打印纸张的大小分页显示的，在输出 PDF 文档时，需要根据文章的长度适当的进行分页处理。

4.5.2 动态生成企业产品名录

很多企业都喜欢将其生产的产品名录做成 PDF 文档的方式在网站上发布。这样做，可以方便客户

进行查看和打印。本小节要实现的就是一个使用 FPDF 自动生成企业产品名录的程序，该程序可以在每次用户要进行 PDF 文档下载时获得数据库中的产品名录信息，代码如下所示。

```
<?php
require('chinese.php');
class PDF extends PDF_Chinese
{
    function Header()                                //设置页眉
    {
        $this->SetFont('GB','',10);
        $this->Write(10,'XX 公司产品名录');
        $this->Ln(20);                                //换行
    }
    function Footer()                                //设置页脚
    {
        $this->SetY(-15);
        $this->SetFont('GB','',10);
        $this->Cell(0,10,'第'.$this->PageNo().'页');
    }
}

$conn = mysql_connect("localhost", "root", "");      //连接数据库

mysql_select_db("product", $conn);                  //执行 SQL
$query_rs_prod = "SELECT * FROM product ORDER BY prod_id";
$rs_prod = mysql_query($query_rs_prod, $conn) or die(mysql_error());
$row_rs_prod = mysql_fetch_assoc($rs_prod);
$totalRows_rs_prod = mysql_num_rows($rs_prod);

$pdf=new PDF();                                     //创建新的 FPDF 对象
$pdf->AddGBFont();                                   //设置中文字体
$pdf->Open();                                         //开始创建 PDF
$pdf->AddPage();                                     //增加一页

$pdf->SetFont('GB','',10);                           //设置字体样式

$header=array('产品编号','产品名称','产品类型','产品单价'); //设置表头
$width=array(20,80,40,20);                          //设置每列宽度

for($i=0;$i<count($header);$i++)                   //循环输出表头
    $pdf->Cell($width[$i],6,$header[$i],1);
$pdf->Ln();

do                                                    //循环输出表体
{
    $pdf->Cell($width[0],6,$row_rs_prod['prod_id'],1);
    $pdf->Cell($width[1],6,$row_rs_prod['prod_name'],1);
    $pdf->Cell($width[2],6,$row_rs_prod['prod_type'],1);
    $pdf->Cell($width[3],6,$row_rs_prod['prod_price'],1);
    $pdf->Ln();
} while ($row_rs_prod = mysql_fetch_assoc($rs_prod));
```

```
$pdf->Output("product.pdf", true);  
?>
```

//下载 PDF 文件

运行结果如图 4-12 所示。

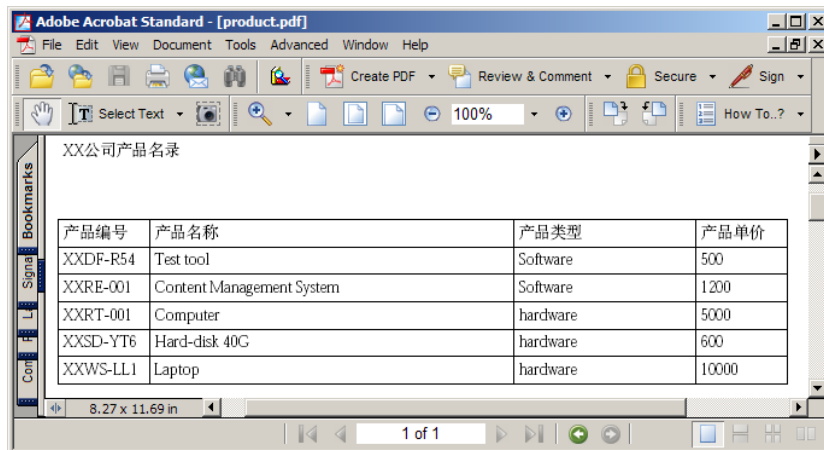


图 4-12 产品名录文档

4.6 小结

本章主要介绍了使用 FPDF 类库创建一些简单的 PDF 文档。PDF 是一种很流行的文件格式，因此在实际应用中，往往会根据实际需要将数据库中的数据制成 PDF 的格式供用户下载、存档。FPDF 在创建 PDF 方面虽然并不是很强大，但是在一般性应用中已经足够了。对于更多的操作，读者可以通过阅读随下载包附带的用户手册来获取。