

## 第 10 章 Jpgraph 创建统计图

上一章介绍了如何使用 GD 库来创建图像，使用 GD 库提供的多种函数可以生成多种图像。但是，如果需要在 PHP 中创建一些统计图，使用 GD 库就显得很麻烦了。本章将要介绍一种专门用于输出统计图的 Jpgraph 库。该库在数据统计等方面有很大优势。在实际应用中，可以很容易的使用 Jpgraph 生成大方得体的统计图。

### 10.1 Jpgraph 简介

Jpgraph 是一个可应用在 PHP 4.3.1 以上版本的图形库，Jpgraph 提供了多种方法创建各种统计图。Jpgraph 是一个完全使用 PHP 语言编写的类库，并可以很容易的应用在任何 PHP 系统中。

Jpgraph 可以创建各种类型的统计图，包括普通 X-Y 坐标图、柱形图、饼状图等多种。Jpgraph 可以用一种很简单的方式创建复杂的图形。

### 10.2 Jpgraph 的安装与配置

Jpgraph 可以从其官方网站 <http://www.aditus.nu/jpgraph/> 下载，目前最新的版本是 2.1.3。本章中所介绍的例子和使用方法均在 2.1.3 版本上调试通过。

#### 10.2.1 Jpgraph 的安装

Jpgraph 的安装非常简单，文件下载后，安装步骤如下所示。

(1) 将压缩包下的全部文件解压缩到一个文件夹中。例如 D:\jpgraph。

(2) 打开 PHP 的安装目录，编辑 php.ini 文件并修改其中的 include\_path 参数，在其后增加前面的文件夹名。例如 include\_path = ".;d:\jpgraph"。

(3) 重新启动 Apache 服务器使改动生效。

如果希望 Smarty 类库仅对当前站点有效，还可以按照以下的简便方法来进行安装。

(1) 将 Smarty 压缩包下 src 文件夹中的全部文件解压缩到网站所在目录的文件夹中。例如 D:\www\htdocs\src。

(2) 修改文件夹名称为自己希望的名字，例如 D:\www\htdocs\jpgraph。

如果在服务器中装有 pear，还可以直接将 src 文件夹放到 pear 所在文件夹下并将其重命名成一个有意义的名字以方便插件的管理。需要注意的是 Jpgraph 需要 GD 库的支持，关于 GD 库的使用方法详见上一章。

## 10.2.2 Jpgraph 的配置

Jpgraph 提供了一个专门用于配置 Jpgraph 类库的文件——`jpg-config.inc.php`。在使用 Jpgraph 之前，可以通过修改本文件来完成 Jpgraph 的配置。

常用的配置主要有以下两项。

- ❑ 中文字体配置：Jpgraph 提供中文标准字体和 BIG5 码字体两种，字体所在文件可以通过修改 `CHINESE_TTF_FONT` 的设置来完成，如下所示。

```
DEFINE('CHINESE_TTF_FONT','bkai00mp.ttf');
```

这里，默认指定该字体文件为 `bkai00mp.ttf`，该文件应该可以在 Windows 安装目录下的 `font` 文件夹中找到。

- ❑ 默认图片格式配置：默认图片格式的配置通过修改 `DEFAULT_GFORMAT` 的设置来完成，如下所示。

```
DEFINE("DEFAULT_GFORMAT","auto");
```

这里，可以根据当前 PHP 环境中支持的图片格式来设定默认的生成图片格式。这里，默认情况下将其设置成“auto”，表示 Jpgraph 将依次按照 PNG、GIF 和 JPEG 的顺序来寻找系统支持的图片格式。

## 10.3 Jpgraph 绘制统计图实例

打开前面介绍的 `src` 文件夹，可以看到多个 PHP 源代码文件。在实际应用中，根据绘制的统计图的样式不同包含不同的 PHP 文件。本节将以几个 Jpgraph 为例来介绍如何使用 Jpgraph 来绘制统计图。

### 10.3.1 简单的 X-Y 坐标图

一个简单的 X-Y 坐标图需要一个 X-Y 坐标和基于坐标轴的曲线。本节将创建一个简单的 X-Y 坐标曲线为例简要说明一下 Jpgraph 类库的使用。该坐标图以月份作为横坐标，以金额单位（兆美元）为纵坐标描述每个月的总开销数。

在创建实例之前，介绍一下实例中将用到的类库。

首先需要使用 Graph 类库，Graph 类库是 Jpgraph 的主要类库，使用 Graph 类库创建的对象可以看作是统计图对象。创建一个新的 Graph 对象的语法格式如下所示。

```
$graph = new Graph($width, $height);
```

其中，`$graph` 是创建的 Graph 对象，`$width` 表示统计图的宽，`$height` 表示统计图的高。

Graph 对象的一个重要方法是 `SetScale`，用于设置统计图的刻度样式，其语法格式如下所示。

```
$graph->SetScale($AxisType);
```

其中 `$graph` 是前面创建的 Graph 对象，`$AxisType` 是刻度的样式。`$AxisType` 的可能值是下面的 X 轴刻度样式与 Y 轴刻度样式的组合。

- ❑ X：直线 `lin`、文本 `text`、对数 `log`、整型数 `int`。

- ❑ Y：直线 `lin`、对数 `log`、整型数 `int`。

例如，`textlin` 就表示 X 坐标为文本样式、Y 坐标为直线样式的 X-Y 刻度。

Graph 对象的另一个重要方法是 `SetMargin`，用于设置统计图的边界范围，其语法格式如下所示。

```
$graph->img->SetMargin($left, $right, $up, $bottom);
```

其中 `$graph` 是前面创建的 Graph 对象，`$left` 是左边距，`$right` 是右边距，`$up` 是上边距，`$bottom` 是

下边距。

统计图往往需要有一个标题来说明统计图的名称，用于设置统计图标题的方法如下所示。

```
$graph->title->Set($title);
```

其中**\$graph**是前面创建的**Graph**对象，**\$title**是统计图的标题。

如果在统计图上绘制曲线，则需要创建**LinePlot**对象。创建一个新的**LinePlot**对象的语法格式如下所示。

```
$lineplot = new LinePlot($data);
```

其中，**\$lineplot**是创建的**LinePlot**对象，**\$data**是一个传入的数组。**Jpgraph**将根据数组中元素的值绘制曲线。

设置曲线颜色的方法如下所示。

```
$lineplot->SetColor($color);
```

其中，**\$lineplot**是创建的**LinePlot**对象，**\$color**是要设置的颜色。例如下面的代码会将曲线的颜色设成蓝色。

```
<?php  
$lineplot->SetColor("blue");  
?>
```

设置曲线相应图例的方法如下所示。

```
$lineplot-> SetLegend ($str);
```

其中，**\$lineplot**是创建的**LinePlot**对象，**\$str**是图例中的文字。

将曲线放置在图像对象的方法如下所示。

```
$graph->Add($lineplot);
```

其中**\$graph**是前面创建的**Graph**对象，**\$lineplot**是创建的**LinePlot**对象。

将图像输出的方法如下所示。

```
$graph->Stroke([$filename]);
```

其中，**\$graph**是前面创建的**Graph**对象，**\$filename**是要生成的文件名。在这里，如果不指定文件名，则将图像直接输出到网页上。

以下代码是实现本节要绘制的统计图的完整代码。

```
<?php  
include ("src/jpgraph.php");           //Graph 类  
include ("src/jpgraph_line.php");      //LinePlot 类  
  
$data = array(19,23,34,38,45,67,71,78,85,87,90,96); //将要用于图表创建的数据存放在数组中  
  
$graph = new Graph(400,300);           //创建新的 Graph 对象  
$graph->SetScale("textlin");           //设置刻度样式  
$graph->img->SetMargin(30,30,80,30);    //设置图表边界  
$graph->title->Set("Year to Date Cost"); //设置图表标题  
  
$lineplot=new LinePlot($data);         //创建新的 LinePlot 对象  
$lineplot->SetLegend("Amount(M dollars)"); //设置图例文字  
$lineplot->SetColor("red");            //设置曲线颜色  
  
$graph->Add($lineplot);                 //在统计图上绘制曲线  
  
$graph->Stroke();                       //输出图像  
?>
```

从上面的代码可以看出，在代码的开头包含了 `jpgraph.php` 文件，用于 `Graph` 对象的创建。包含 `jpgraph_line.php` 文件用于 `LinePlot` 对象的创建。

代码的运行效果如图 10-1 所示。

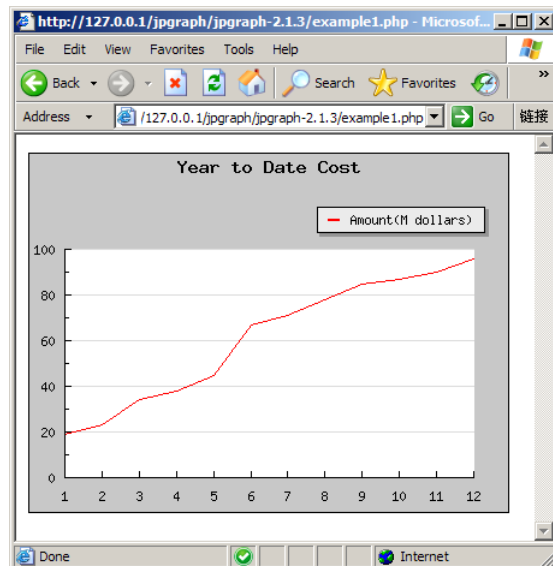


图 10-1 X-Y 统计图

上面的代码完成了如下的功能：

- ☐ 首先定义一个有 12 个元素的数组，分别表示 12 个月中每个月的金额。
- ☐ 创建 `Graph` 对象生成一个 400\*300 的画布，并设置 X 坐标刻度为 `text`，Y 坐标刻度为 `lin`。
- ☐ 设置统计图所在画布的位置，其中，左边距为 30，右边距为 30，上边距为 80，下边距为 30。
- ☐ 设置统计图的标题为“Year to Date Cost”。
- ☐ 创建一个新的 `LinePlot` 对象，并将前面创建的数组传入。
- ☐ 设置图例为“Amount(M dollars)”。
- ☐ 设置曲线的颜色为红色。
- ☐ 最后在网页上输出图像。

从图中可以看出，`Jpgraph` 根据数组中的元素个数创建了 X 坐标的刻度个数，并根据数组元素的实际数值创建了 Y 坐标的刻度范围。

从上面的例子可以看出，使用 `Jpgraph` 可以使用一个简单的方法绘制出非常实用的统计图。

### 10.3.2 改进的 X-Y 坐标图

上一小节的例子创建了一个有一条曲线的 X-Y 坐标图。本节将在上一节例子的基础上添加另一条曲线。

第二条曲线的刻度将放到右侧 Y 轴上，设置右侧 Y 轴刻度样式的方法如下所示。

```
$graph->SetY2Scale($AxisType);
```

其中，`$graph` 是前面创建的 `Graph` 对象，`$AxisType` 是前面介绍过的刻度样式。

将第二条曲线放置到图像的方法如下所示。

```
$graph->AddY2($lineplot);
```

其中 `$graph` 是前面创建的 `Graph` 对象，`$lineplot` 是创建的 `LinePlot` 对象。

完整的实现代码如下所示。

```
<?php
include ("src/jpgraph.php");
include ("src/jpgraph_line.php");

$data1 = array(19,23,34,38,45,67,71,78,85,87,90,96);           //第一条曲线的数组
$data2 = array(523,634,371,278,685,587,490,256,398,545,367,577); //第二条曲线的数组

$graph = new Graph(400,300);                                   //创建新的 Graph 对象
$graph->SetScale("textlin");
$graph->SetY2Scale("lin");
$graph->SetShadow();                                           //设置图像的阴影样式

$graph->img->SetMargin(40,50,20,70);                           //设置图像边距
$graph->title->Set("Year to Date Cost & Revenue");             //设置图像标题

$lineplot1=new LinePlot($data1);                             //创建设置两条曲线对象
$lineplot2=new LinePlot($data2);

$graph->Add($lineplot1);                                       //将曲线放置到图像上
$graph->AddY2($lineplot2);

$graph->xaxis->title->Set("Month");                             //设置坐标轴名称
$graph->yaxis->title->Set("M dollars");
$graph->y2axis->title->Set("M dollars");

$graph->title->SetFont(FF_FONT1,FS_BOLD);                     //设置字体
$graph->yaxis->title->SetFont(FF_FONT1,FS_BOLD);
$graph->y2axis->title->SetFont(FF_FONT1,FS_BOLD);
$graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD);

$lineplot1->SetColor("red");                                  //设置颜色
$lineplot2->SetColor("blue");

$lineplot1->SetLegend("Cost Amount");                         //设置图例名称
$lineplot2->SetLegend("Revenue Amount");

$graph->legend->SetLayout(LEGEND_HOR);                         //设置图例样式和位置
$graph->legend->Pos(0.4,0.95,"center","bottom");

$graph->Stroke();                                             //输出图像
?>
```

运行效果如图 10-2 所示。

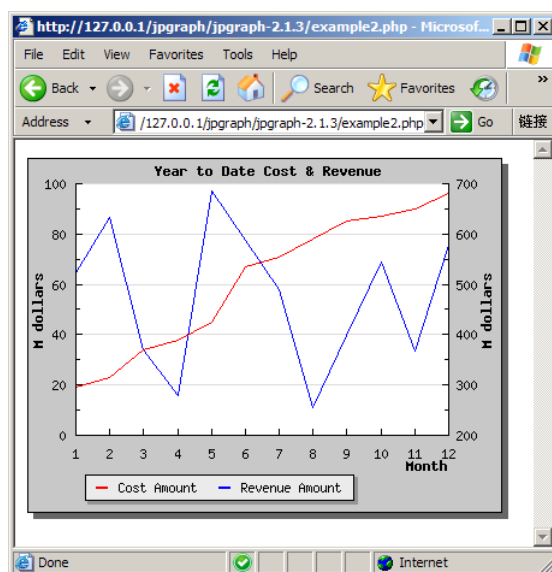


图 10-2 改进的 X-Y 坐标图

这里需要注意的是 SetFont 方法，语法格式如下所示。

```
SetFont($family, [$style,] [$size]);
```

其中，\$family 表示字体，\$style 表示字体样式，\$size 表示字体大小，默认为 10。当前可用的字体可以在 src 文件夹下的 jpgraph.php 文件中找到。对于当前版本的 jpgraph.php，可以看到如下所示的代码。

```
DEFINE("FF_COURIER",10);
DEFINE("FF_VERDANA",11);
DEFINE("FF_TIMES",12);
DEFINE("FF_COMIC",14);
DEFINE("FF_ARIAL",15);
DEFINE("FF_GEORGIA",16);
DEFINE("FF_TREBUCH",17);

// Gnome Vera font
// Available from http://www.gnome.org/fonts/
DEFINE("FF_VERA",18);
DEFINE("FF_VERAMONO",19);
DEFINE("FF_VERASERIF",20);

// Chinese font
DEFINE("FF_SIMSUN",30);
DEFINE("FF_CHINESE",31);
DEFINE("FF_BIG5",31);

// Japanese font
DEFINE("FF_MINCHO",40);
DEFINE("FF_PMINCHO",41);
DEFINE("FF_GOTHIC",42);
DEFINE("FF_PGOTHIC",43);

// Limits for fonts
DEFINE("_FIRST_FONT",10);
```

```

DEFINE("_LAST_FONT",43);

// TTF Font styles
DEFINE("FS_NORMAL",9001);
DEFINE("FS_BOLD",9002);
DEFINE("FS_ITALIC",9003);
DEFINE("FS_BOLDIT",9004);
DEFINE("FS_BOLDITALIC",9004);

//Definitions for internal font, new style
DEFINE("FF_FONT0",1);
DEFINE("FF_FONT1",2);
DEFINE("FF_FONT2",4);

```

其中，“TTF Font styles”一节中为字体样式，“Definitions for internal font, new style”一节中为默认字体。可以看到当前的 Jpgraph 版本支持宋体，以下代码将前面例子中的字体使用 FF\_SIMSUN 来替换原来的默认字体 FF\_FONT1，并将所有的提示翻译成中文。

```

<?php
include ("src/jpgraph.php");
include ("src/jpgraph_line.php");

$data1 = array(19,23,34,38,45,67,71,78,85,87,90,96);           //第一条曲线的数组
$data2 = array(523,634,371,278,685,587,490,256,398,545,367,577); //第二条曲线的数组

$graph = new Graph(400,300);                                   //创建新的 Graph 对象
$graph->SetScale("textlin");
$graph->SetY2Scale("lin");
$graph->SetShadow();                                           //设置图像的阴影样式

$graph->img->SetMargin(40,50,20,70);                             //设置图像边距
$graph->title->Set("年度收支表");                               //设置图像标题

$lineplot1=new LinePlot($data1);                               //创建设置两条曲线对象
$lineplot2=new LinePlot($data2);

$graph->Add($lineplot1);                                       //将曲线放置到图像上
$graph->AddY2($lineplot2);

$graph->xaxis->title->Set("月份");                               //设置坐标轴名称
$graph->yaxis->title->Set("兆美元");
$graph->y2axis->title->Set("兆美元");

$graph->title->SetFont(FF_SIMSUN,FS_BOLD);                     //设置字体
$graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);
$graph->y2axis->title->SetFont(FF_SIMSUN,FS_BOLD);
$graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);

$lineplot1->SetColor("red");                                   //设置颜色
$lineplot2->SetColor("blue");

```

```

$lineplot1->SetLegend("Cost Amount");           //设置图例名称
$lineplot2->SetLegend("Revenue Amount");

$graph->legend->SetLayout(LEGEND_HOR);           //设置图例样式和位置
$graph->legend->Pos(0.4,0.95,"center","bottom");

$graph->Stroke();                                //输出图像
?>

```

运行结果如图 10-3 所示。

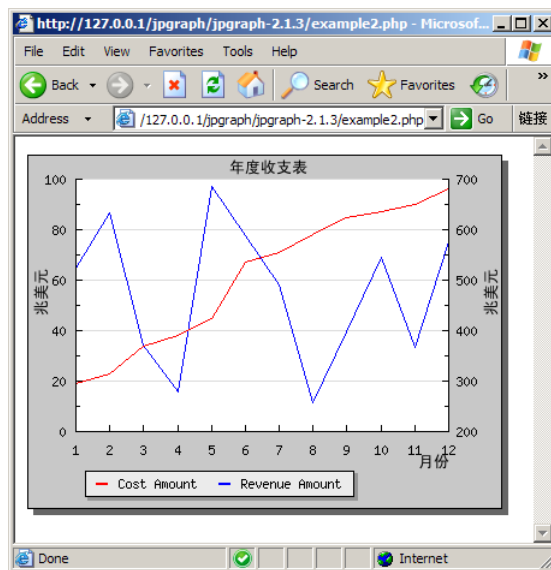


图 10-3 中文字体的显示

### 10.3.3 柱形图

柱形图的绘制与曲线的绘制很相似，不同的是绘制柱形图需要创建 BarPlot 对象，BarPlot 类在 jpgraph\_bar.php 中定义。因此，如果要创建柱形图，必须首先在代码中包含 jpgraph\_bar.php 文件。

BarPlot 对象的创建方法如下所示。

```
$barplot = new BarPlot($data);
```

其中，\$barplot 是创建的 BarPlot 对象，\$data 是一个传入的数组。Jpgraph 将根据数组中元素的值绘制柱形图。

以下代码生成了一个柱形图，例子中所用的数组与前面相同，读者可以直观的与前面的例子进行比较。

```

<?php
include ("src/jpgraph.php");
include ("src/jpgraph_bar.php");

$data = array(19,23,34,38,45,67,71,78,85,87,90,96);           //定义数组

$graph = new Graph(400,300);                                   //创建新 Graph 对象
$graph->SetScale("textlin");

$graph->SetShadow();                                           //设置阴影

```



```

$graph->img->SetMargin(40,30,20,40);           //设置边距

$barplot = new BarPlot($data);                 //创建 BarPlot 对象

$barplot->SetFillColor('blue');                //设置颜色
$barplot->value->Show();                        //设置显示数字
$graph->Add($barplot);                          //将柱形图添加到图像中

$graph->title->Set("年度收支表");               //设置标题和 X-Y 轴标题
$graph->xaxis->title->Set("月份");
$graph->yaxis->title->Set("总金额（兆美元）");

$graph->title->SetFont(FF_SIMSUN,FS_BOLD);       //设置字体
$graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);
$graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);

$graph->Stroke();
?>

```

运行结果如图 10-4 所示。

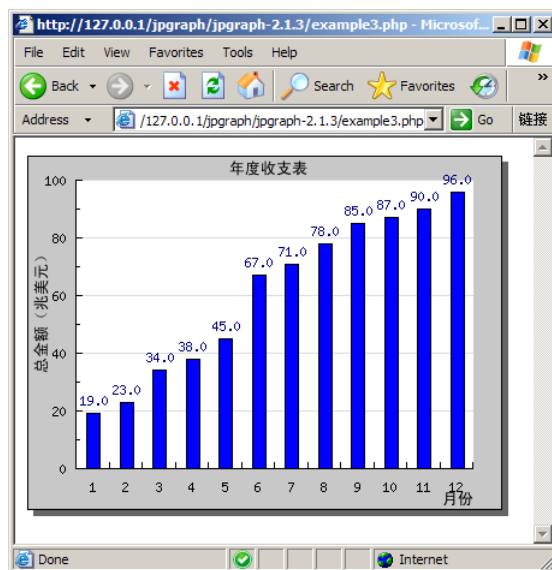


图 10-4 柱形图

比较图 10-4 与图 10-1，可以看到两个统计图完成了同样的作用，但是给访问者的直观效果不同。在实际应用中，应该根据实际需要选择统计图的样式呈现出来。

### 10.3.4 饼图

饼图的绘制与柱形图和曲线的绘制都很相似，不同的是绘制饼图时往往使用 `PieGraph` 对象来替代 `Graph` 对象，`PieGraph` 类是 `Graph` 类的一个子类。同时需要创建 `PiePlot` 对象来设置饼图。`PieGraph` 类和 `PiePlot` 类均在 `jpgraph_pie.php` 中定义。因此，如果要创建柱形图，必须首先在代码中包含 `jpgraph_pie.php` 文件。需要注意的是由于 `PieGraph` 类是 `Graph` 类的子类，因此，在创建饼图时，虽然没有直接用到

jpgraph.php 中的 Graph 类，也许要在程序中包含 jpgraph.php 文件。

PiePlot 对象的创建方法如下所示。

```
$pieplot = new PiePlot($data);
```

其中，\$pieplot 是创建的 PiePlot 对象，\$data 是一个传入的数组。Jpgraph 将根据数组中元素的值绘制饼图。

以下代码生成了一个饼图，例子中所用的数组与前面相同，读者可以直观的与前面的柱形图进行比较。

```
<?php
include ("src/jpgraph.php");
include ("src/jpgraph_pie.php");

$data = array(19,23,34,38,45,67,71,78,85,87,90,96);

$graph = new PieGraph(400,300);           //创建 PieGraph 对象
$graph->SetShadow();                       //设置阴影样式

$graph->title->Set("年度收支表");          //设置统计图标题
$graph->title->SetFont(FF_SIMSUN,FS_BOLD); //设置字体

$pieplot = new PiePlot($data);            //创建 PiePlot 对象
$graph->Add($pieplot);                     //将 PiePlot 对象放置到图像中
$graph->Stroke();
?>
```

运行结果如图 10-5 所示。

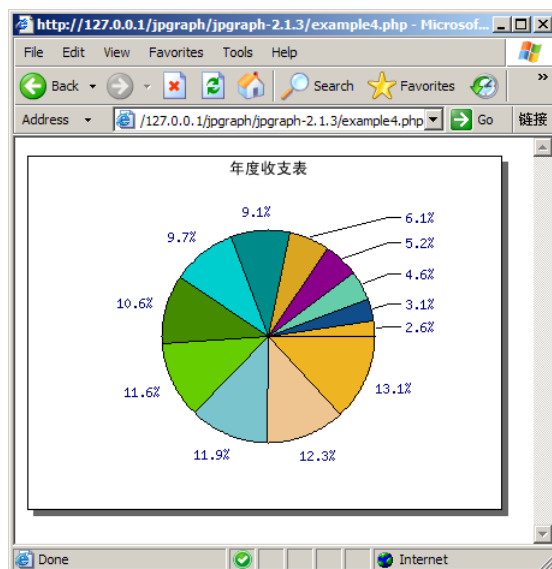


图 10-5 饼图

从饼图中可以看到，各个月份所占百分比在图中被自动计算出来。

### 10.3.5 3D 饼图

上一小节介绍了如何输出一个简单的饼图，本节将要介绍一种更美观的饼图——3D 饼图。3D 饼图

的创建方法与前面的饼图的创建方法相似，不同的是所需要创建的对象为 PiePlot3D 对象。PiePlot3D 类是 PiePlot 类的一个子类，用于创建 3D 饼图。

PiePlot3D 类存放在 jpgraph\_pie3d.php 文件中，因此，绘制 3D 饼图需要包含 jpgraph.php、jpgraph\_pie.php 与 jpgraph\_pie3d.php 三个文件。完整的代码如下所示。

```
<?php
include ("src/jpgraph.php");
include ("src/jpgraph_pie.php");
include ("src/jpgraph_pie3d.php");

$data = array(19,23,34,38,45,67,71,78,85,87,90,96);

$graph = new PieGraph(400,300);                                //创建 PieGraph 对象
$graph->SetShadow();

$graph->title->Set("年度收支表");                                //设置标题
$graph->title->SetFont(FF_SIMSUN,FS_BOLD);

$pieplot = new PiePlot3D($data);                                //创建 PiePlot3D 对象
$pieplot->SetCenter(0.4);                                        //设置饼图中心的位置
$pieplot->SetLegends(array('Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec')); //设置图例
$graph->Add($pieplot);
$graph->Stroke();
?>
```

运行结果如图 10-6 所示。

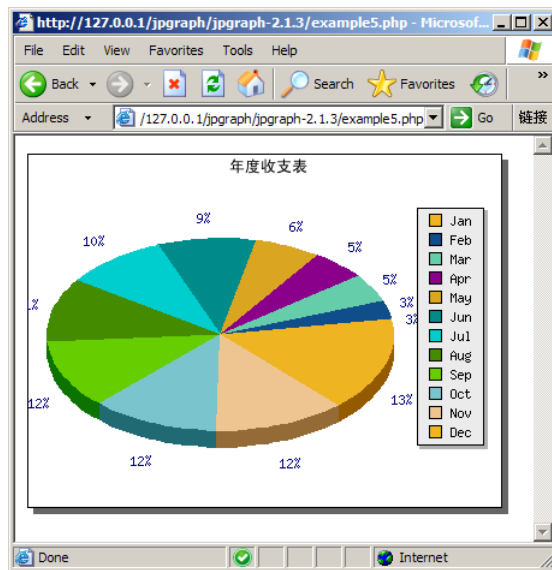


图 10-6 3D 饼图

需要注意的是图例的设置方法，如下所示。

```
$pieplot->SetLegends($array);
```

其中 \$pieplot 是前面定义的 PiePlot3D 对象，\$array 是一个用于传入的数组。在输出图像时，Jpgraph 将根据传入的数组输出一个图例。

## 10.4 Jpgraph 应用实例——工厂销量查询系统

前面介绍了使用 Jpgraph 制作几款不同的统计图，本小节将要综合这几种统计图开发一个工厂销量查询系统。该系统提供了一个浏览统计图的页面，用户可以通过在页面上选择时间范围和要查看的统计图形式动态生成统计图。供用户选择的表单页面如下所示。

```
<html>
<head>
<title>工厂销量查询系统</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<style type="text/css">
<!--
.style1 {
    font-size: 16px;
    font-weight: bold;
}
-->
</style>
</head>

<body>
<form name="form1" method="get" action="example6.php">
  <p align="center" class="style1"> 工厂销量查询系统 </p>
  <table width="300" border="1" align="center" cellpadding="3" cellspacing="3">
    <tr>
      <td width="85"><strong>查询年份</strong></td>
      <td width="188"><select name="acct_yr" id="acct_yr">
        <option value="2006" selected>2006</option>
      </select></td>
    </tr>
    <tr>
      <td><strong>起始月份</strong></td>
      <td><select name="start_mth" id="start_mth">
        <option selected>01</option>
        <option>02</option>
        <option>03</option>
        <option>04</option>
        <option>05</option>
        <option>06</option>
      </select></td>
    </tr>
    <tr>
      <td><strong>终止月份</strong></td>
      <td><select name="end_mth" id="end_mth">
        <option>01</option>
        <option>02</option>
        <option>03</option>
        <option>04</option>
        <option>05</option>
      </select></td>
    </tr>
  </table>
</form>
```



```

case 1:                                     //X-Y 图
    $graph = new Graph(400,300);           //创建新的 Graph 对象
    $graph->SetScale("textlin");            //设置刻度样式
    $graph->img->SetMargin(30,30,80,30);     //设置图表边界
    $graph->title->SetFont(FF_SIMSUN,FS_BOLD); //设置字体
    $graph->title->Set("工厂销量查询表");    //设置图表标题
    $lineplot=new LinePlot($data);
    $lineplot->SetLegend("AMOUNT");
    $lineplot->SetColor("red");
    $graph->Add($lineplot);
    break;

case 2:                                     //柱形图
    $graph = new Graph(400,300);
    $graph->SetScale("textlin");
    $graph->SetShadow();
    $graph->img->SetMargin(40,30,20,40);
    $barplot = new BarPlot($data);         //创建 BarPlot 对象
    $barplot->SetFillColor('blue');        //设置颜色
    $barplot->value->Show();                //设置显示数字
    $graph->Add($barplot);                  //将柱形图添加到图像中
    $graph->title->Set("工厂销量查询表");    //设置标题和 X-Y 轴标题
    $graph->xaxis->title->Set("月份");
    $graph->yaxis->title->Set("总销量");
    $graph->title->SetFont(FF_SIMSUN,FS_BOLD); //设置字体
    $graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);
    $graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);
    break;

case 3:                                     //饼图
    $graph = new PieGraph(400,300);
    $graph->SetShadow();

    $graph->title->Set("工厂销量查询表");    //设置标题
    $graph->title->SetFont(FF_SIMSUN,FS_BOLD);
    $pieplot = new PiePlot($data);
    $pieplot->SetLegends($gDateLocale->GetShortMonth()); //设置图例
    $graph->Add($pieplot);
    break;

case 4:                                     //3D 饼图
    $graph = new PieGraph(400,300);
    $graph->SetShadow();

    $graph->title->Set("工厂销量查询表");    //设置标题
    $graph->title->SetFont(FF_SIMSUN,FS_BOLD);
    $pieplot = new PiePlot3D($data);       //创建 PiePlot3D 对象
    $pieplot->SetCenter(0.4);               //设置饼图中心的位置
    $pieplot->SetLegends($gDateLocale->GetShortMonth()); //设置图例
    $graph->Add($pieplot);
    break;

default:
    echo "graph 参数错误";

```

```
exit;  
}  
$graph->Stroke();  
?>
```

在浏览器中访问表单页面如图 10-7 所示。

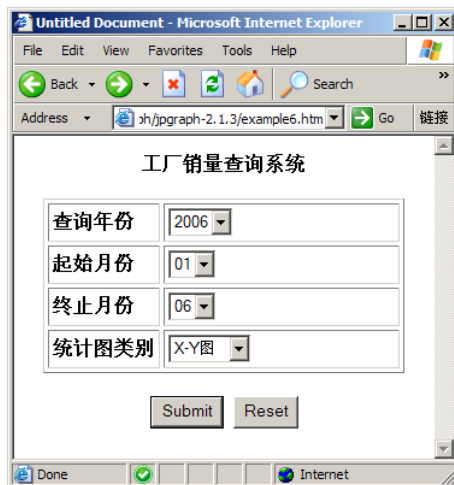


图 10-7 工厂销量查询页面

当用户选择 3D 饼图时，运行结果如图 10-8 所示。

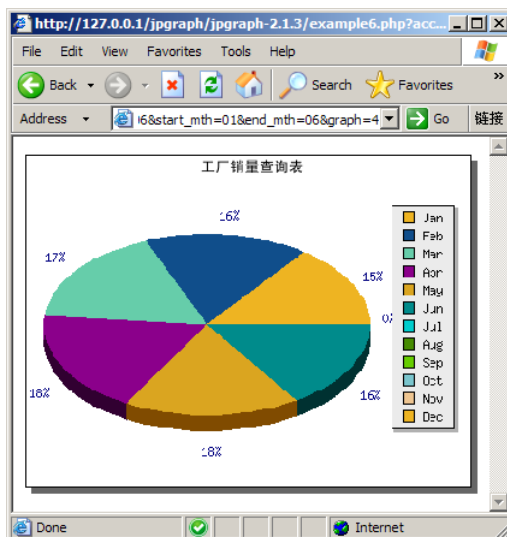


图 10-8 3D 饼图

当用户选择 X-Y 图时，运行结果如图 10-9 所示。



图 10-9 X-Y 图

可以看到，PHP 代码根据用户的选择不同生成了不同类型的统计图。需要注意的是由于页面可能生成很多种统计图类型，在使用 `include` 函数包含类库文件时需要包含所有可能用到的类库文件。

## 10.5 小结

本章介绍了一个专门用于绘制统计图的类库——Jpgraph。使用 Jpgraph 类库可以使用很少的代码绘制一个较复杂的统计图。在实际应用中，如果需要根据数据动态生成统计图，可以使用 Jpgraph 大大减少开发时所需时间。

本章仅对 Jpgraph 进行了一个简要的介绍，在 Jpgraph 的下载压缩包中有大量的实例和类库的说明手册，读者可以参考其中的说明和例子来获取关于 Jpgraph 的更多内容。