

# HTML5 作图函数库

## Canvas 函数库

作者：[金海龙\(bigengineer@163.com\)](mailto:bigengineer@163.com)

前不久，我发布了一个 HTML5 Canvas 初等作图函数库，很多个人开发者关注了我，所以我打算完善这个用 JavaScript 写成的函数库。

下面详细介绍新增的函数，需要源代码的 Web 开发者，可在页底查看下载链接。要运行下面的源代码，需要使用 Google 浏览器或者 IE 9

注：下列绘图函数都是“[bigengineer.js](#)”中的内容，这个文档可以在如下网址下载：

[http://www.filefactory.com/file/b3d9992/n/bigengineer\\_rar](http://www.filefactory.com/file/b3d9992/n/bigengineer_rar)

也可以网络浏览：

[http://www.docstoc.com/docs/document-preview.aspx?doc\\_id=56753328](http://www.docstoc.com/docs/document-preview.aspx?doc_id=56753328)

1、function DrawRoundRect(Canvas,P1,P2,Radius)

功能：画正圆角矩形

参数：P1：矩形左上角坐标；P2：矩形右下角坐标；

Radius：圆角大小

实例：

```
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");

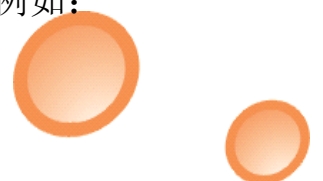
var A=new Array(55,55);
var C=new Array(150,150);

hb.beginPath();
hb.strokeStyle="#000000";
DrawRoundRect(hb,A,C,10);
hb.stroke();
hb.endPath();
</script>
</body>
</html>
```

用 **Google** 浏览器浏览的效果如下：



如果 Radius 设置的太大，那么就能画出意想不到的图片，例如：



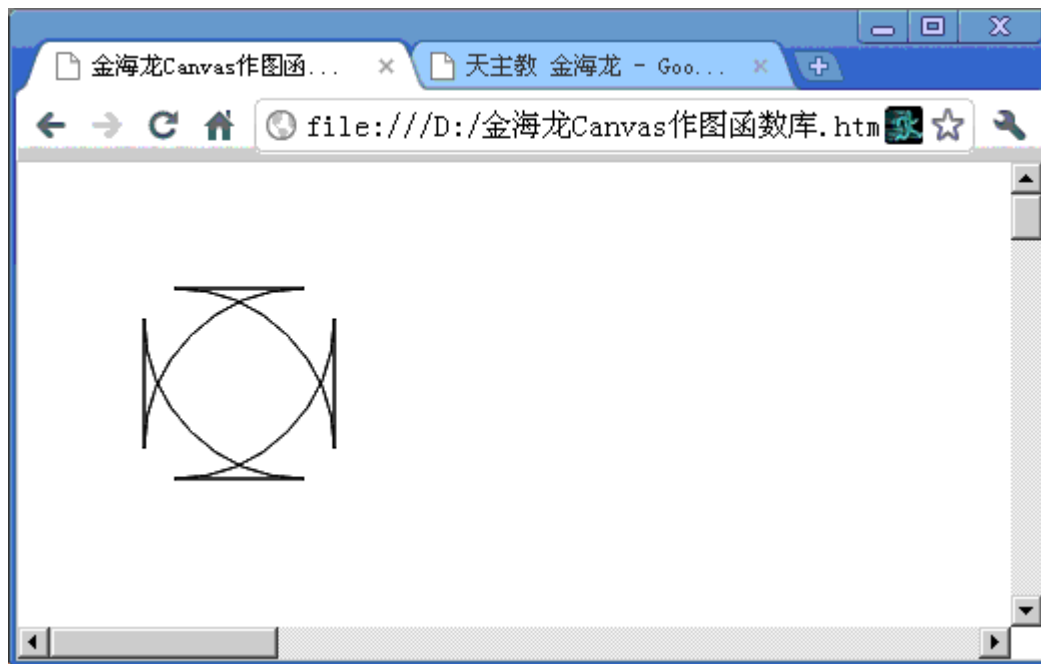
.....

```
DrawRoundRect(hb,A,C,80);//把 Radius 设置成 80
```

```
hb.stroke();
```

.....

用 Google 浏览器浏览的效果如下：



.....

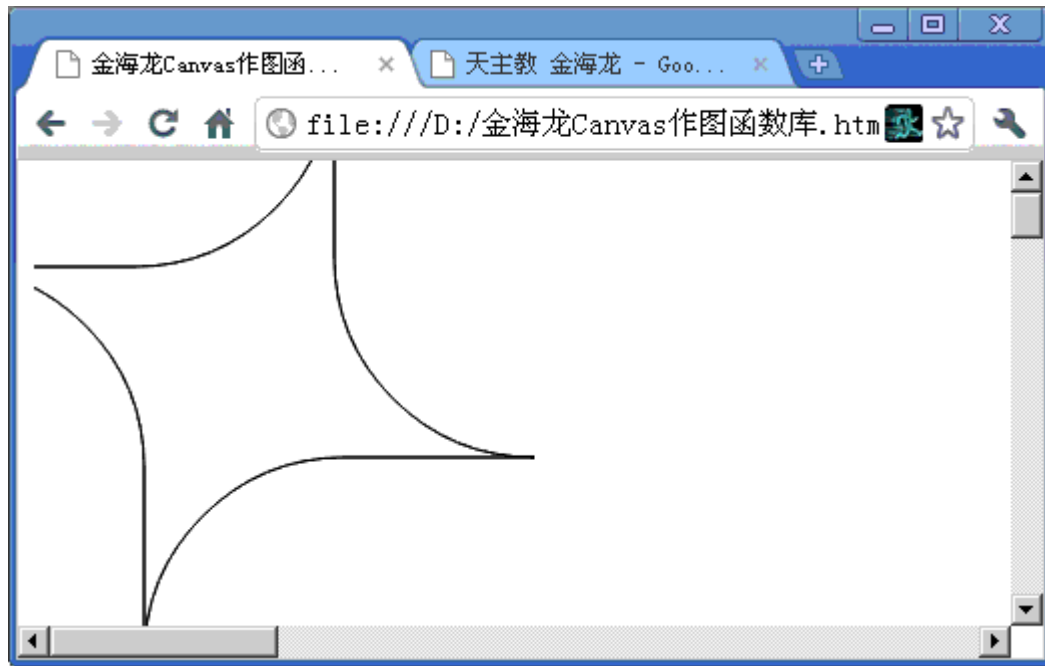
```
DrawRoundRect(hb,A,C,100);//把 Radius 设置成 100
```

```
hb.stroke();
```

.....

用 Google 浏览器浏览的效果如下：





## 2、function DrawAOBArc(Canvas,A,O,B,Radius)

功能：画圆角

参数：Radius：圆角的大小；A，B：边上两点；

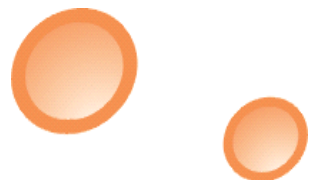
实例：

```
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");

var A=new Array(5,5);
var O=new Array(130,130);
var B=new Array(5,150);

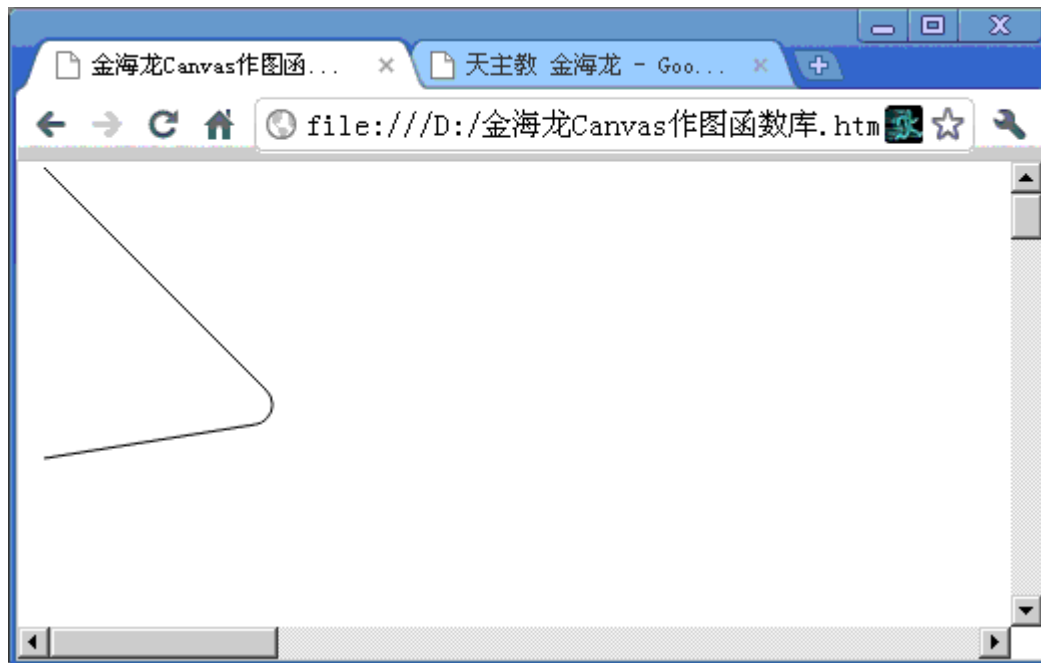
hb.beginPath();
hb.strokeStyle="#000000";

DrawAOBArc(hb,A,O,B,10);
hb.stroke();
```



```
hb.endPath();
</script>
</body>
</html>
```

用 **Google** 浏览器浏览的效果如下:



### 3、function DrawQuadraticCurveTo(Canvas,A,O,B)

功能：画二次样条曲线路径

参数：A:起点；B:终点；O:控制点

### 4、function DrawbezierCurveTo(Canvas,A,O1,O2,B)

功能：画贝塞尔曲线路径

参数：O1 和 O2 是控制点

实例：

```
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");
```



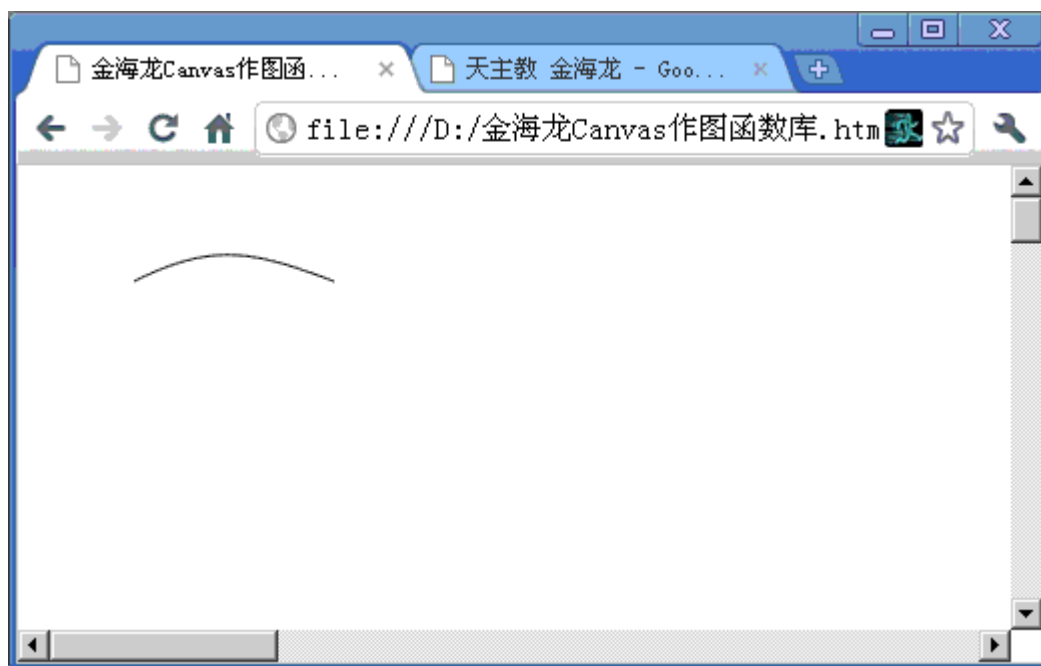
```
var A=new Array(50,50);
var O1=new Array(90,30);
var O2=new Array(110,35);
var B=new Array(150,50);

hb.beginPath();
hb.strokeStyle="#000000";

DrawbezierCurveTo(hb,A,O1,O2,B);
hb.stroke();

hb.endPath();
</script>
</body>
</html>
```

用 **Google** 浏览器浏览的效果如下：



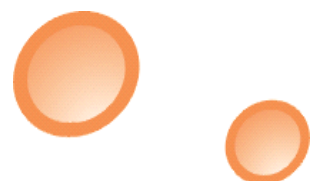
## 5、function DrawArcTo(Canvas,A,O,B,Radius)

功能：画段弧线

参数：Radius：弧的大小

实例：

```
<html>
```



```

<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");

var A=new Array(50,50);
var O=new Array(90,10);
var B=new Array(150,50);

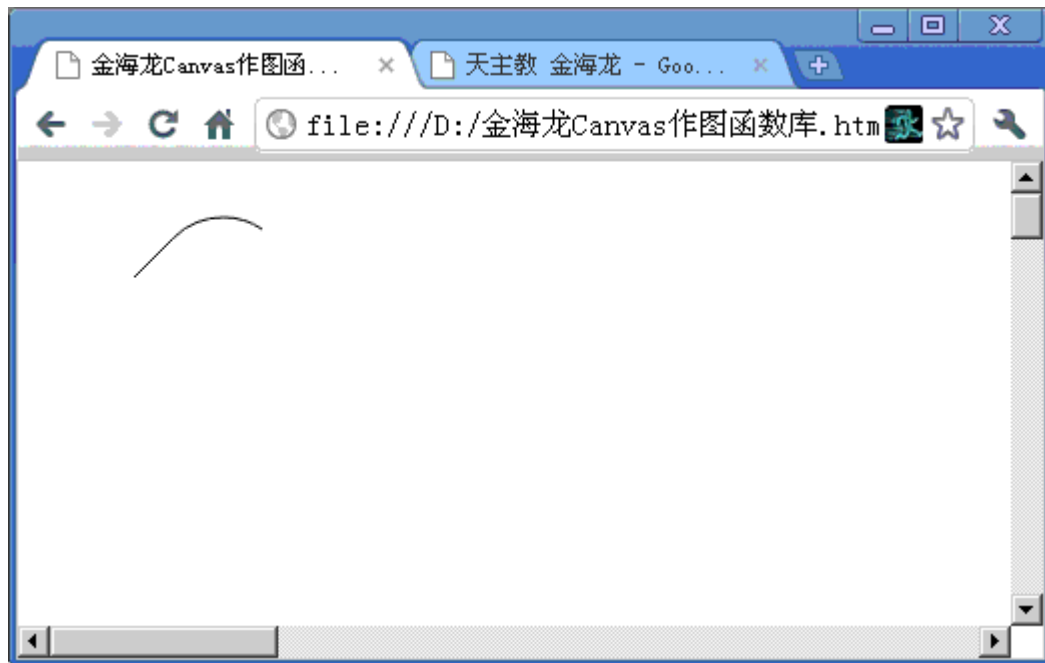
hb.beginPath();
hb.strokeStyle="#000000";

DrawArcTo(hb,A,O,B,35);
hb.stroke();

hb.endPath();
</script>
</body>
</html>

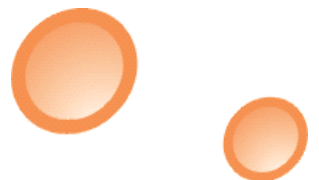
```

用 **Google** 浏览器浏览的效果如下：



6、function DrawArc(Canvas,O,Radius,startAngle, endAngle,  
anticlockwise)

功能：画段弧线



参数: **startAngle**:起始角度; **endAngle**: 终止角度;

**anticlockwise**: 是否按照逆时针方向画弧;

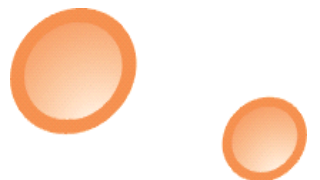
实例:

Example 1: DrawArc(hb,B,50,0,-90,true);//逆时针画弧线

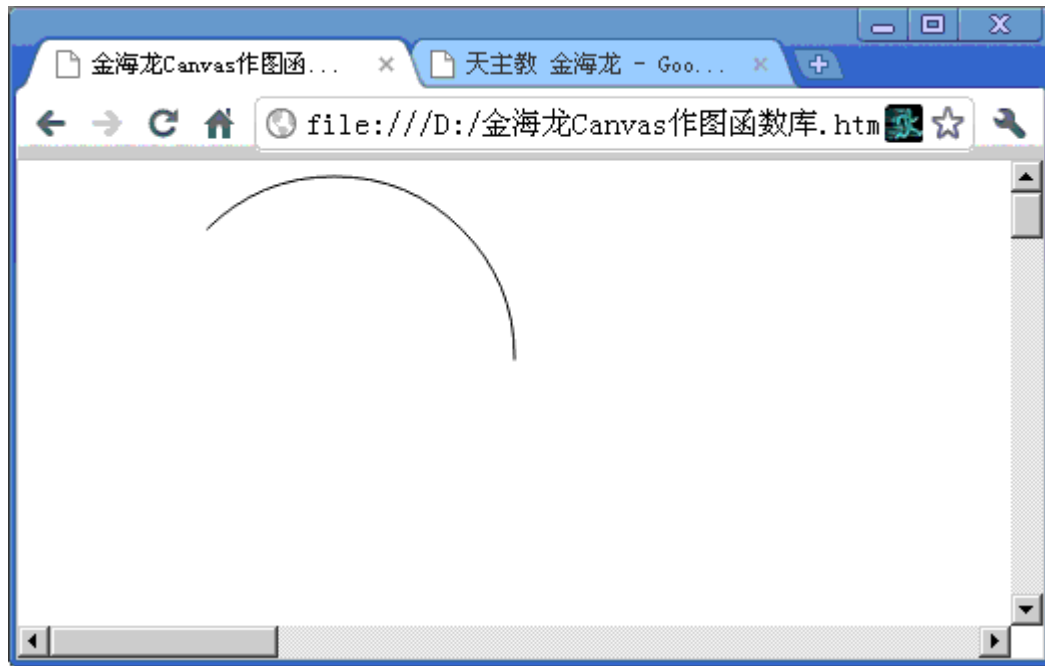
Example 2: DrawArc(hb,B,50,0,-90,false);//顺时针画弧线

```
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");
var B=new Array(150,100);
hb.beginPath();
hb.strokeStyle="#000000";
DrawArc(hb,B,90,-135,true);
hb.stroke();
hb.endPath();
</script>
</body>
</html>
```

用 **Google** 浏览器浏览的效果如下:







## 7、function DrawCircle(Canvas,O,Radius)

功能：画圆

参数：O：圆心； Radius：半径；

实例：

```
var B=new Array(150,150);
```

```
DrawCircle(hb,B,50);
```

下面是一个完整的例子：

```
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");

var B=new Array(150,100);

hb.beginPath();
hb.strokeStyle="#000000";

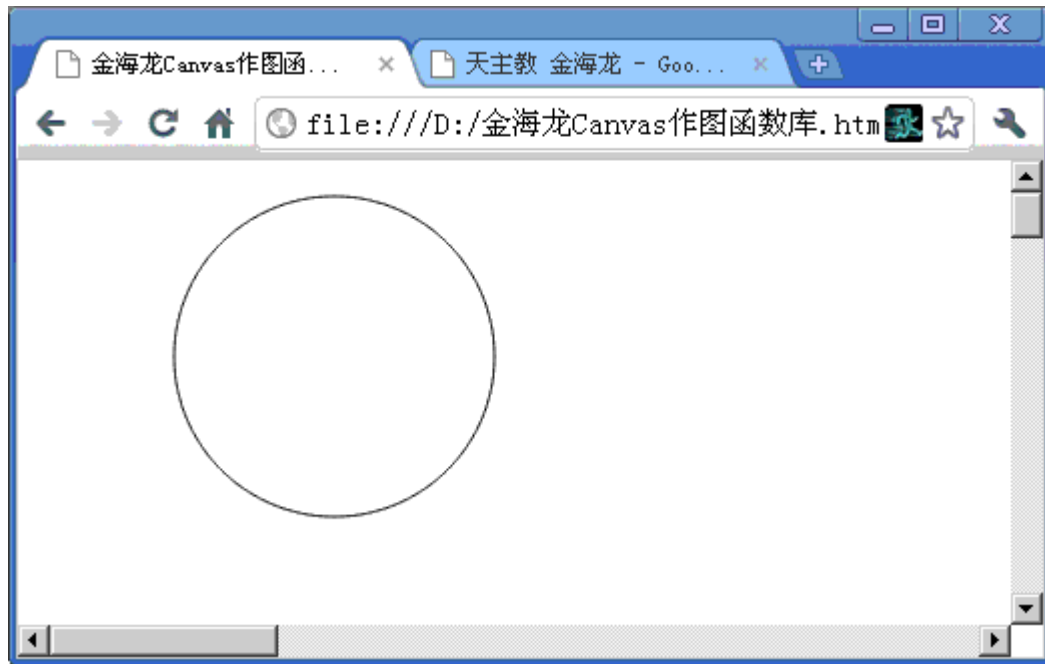
DrawCircle(hb,B,80);
```



```
hb.stroke();

hb.endPath();
</script>
</body>
</html>
```

用 **Google** 浏览器浏览的效果如下：



## 8、function DrawEllipse(Canvas,O,OA,OB)

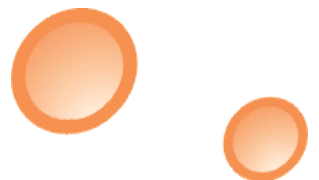
功能：画椭圆

参数：O：椭圆中心的坐标； OA，OB：长轴、短轴的长度

实例：

```
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");

var A=new Array(50,50);
var O=new Array(90,10);
var B=new Array(150,100);
```



```

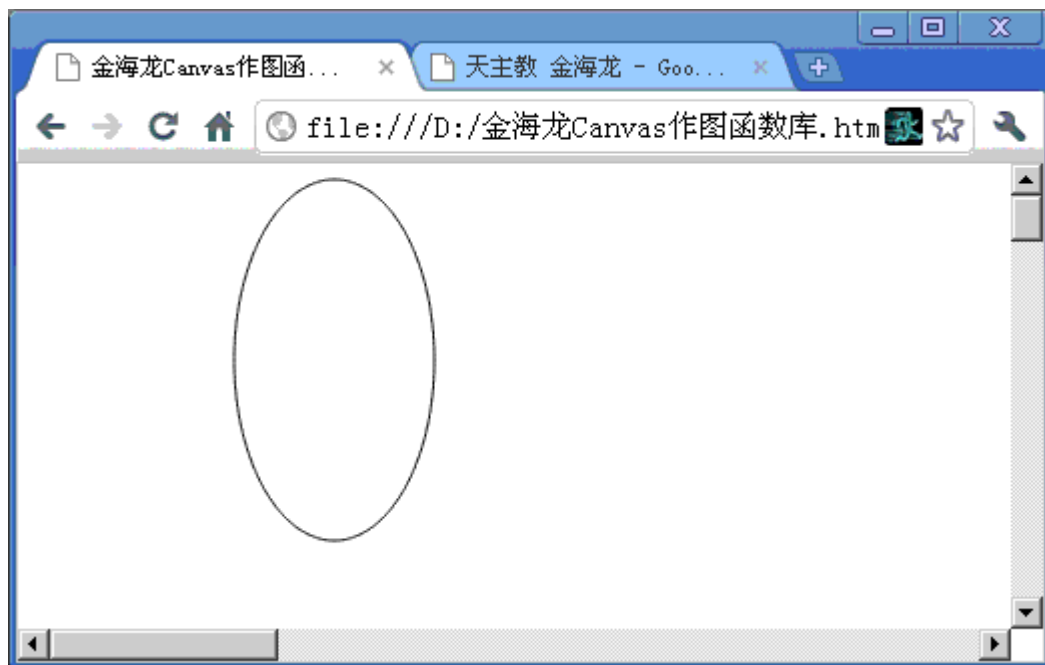
hb.beginPath();
hb.strokeStyle="#000000";

DrawEllipse(hb,B,50,90);
hb.stroke();

hb.endPath();
</script>
</body>
</html>

```

用 **Google** 浏览器浏览的效果如下：



## 9、function DrawPolyline(Canvas,P)

功能：绘制不闭合的线

参数：P 中是多个点的坐标，P 的长度必须是偶数

实例：

```

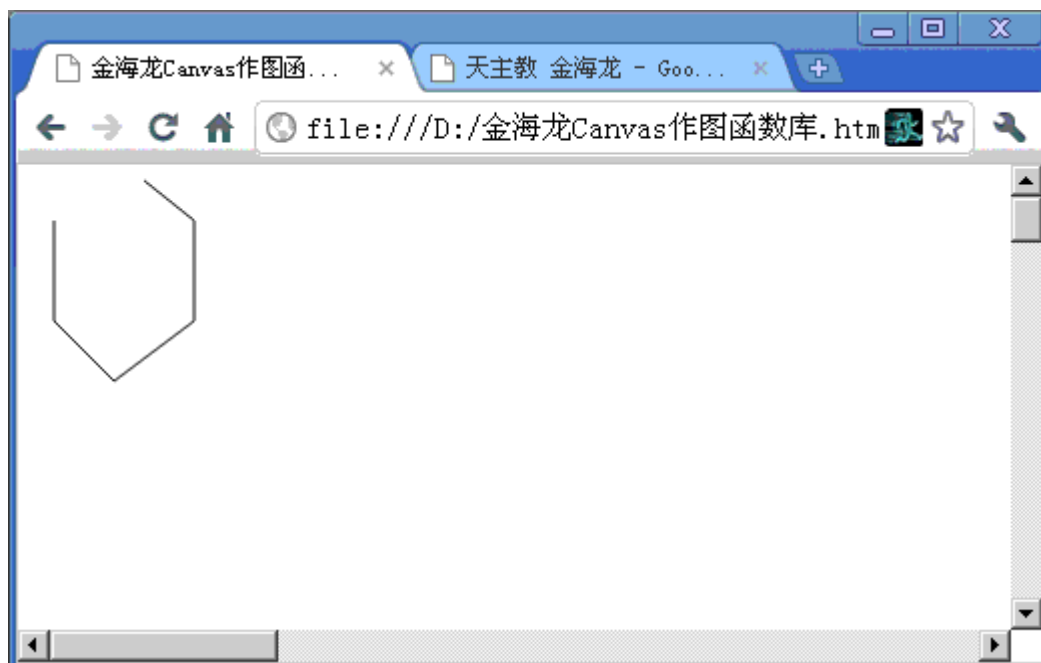
<html>
<script type="text/javascript" src="bigengineer.js"></script>
<body><canvas id="cc" width="2000" height="2000"></canvas>
<script type="text/javascript">
var c=document.getElementById("cc");
var hb=c.getContext("2d");

```



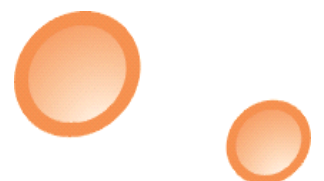
```
var P=new Array(12);
P[0]=10;
P[1]=30;
P[2]=10;
P[3]=80;
P[4]=40;
P[5]=110;
P[6]=80;
P[7]=80;
P[8]=80;
P[9]=30;
P[10]=55;
P[11]=10;
hb.beginPath();
DrawPolyline(hb,P);
hb.stroke();
hb.endPath();
</script>
</body>
</html>
```

用 **Google** 浏览器浏览的效果如下：



## 10、function DrawPolygon(Canvas,P)

功能：画闭合的多边形



参数：P 是一个包含各个顶点坐标的数组

实例：

```
<html>
```

```
<script type="text/javascript" src="bigengineer.js"></script>
```

```
<body><canvas id="cc" width="2000" height="2000"></canvas>
```

```
<script type="text/javascript">
```

```
var c=document.getElementById("cc");
```

```
var hb=c.getContext("2d");
```

```
var P=new Array(12);
```

```
P[0]=10;
```

```
P[1]=30;
```

```
P[2]=10;
```

```
P[3]=80;
```

```
P[4]=40;
```

```
P[5]=110;
```

```
P[6]=80;
```

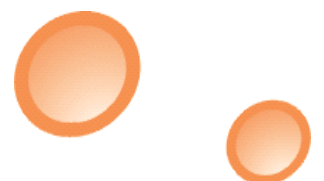
```
P[7]=80;
```

```
P[8]=80;
```

```
P[9]=30;
```

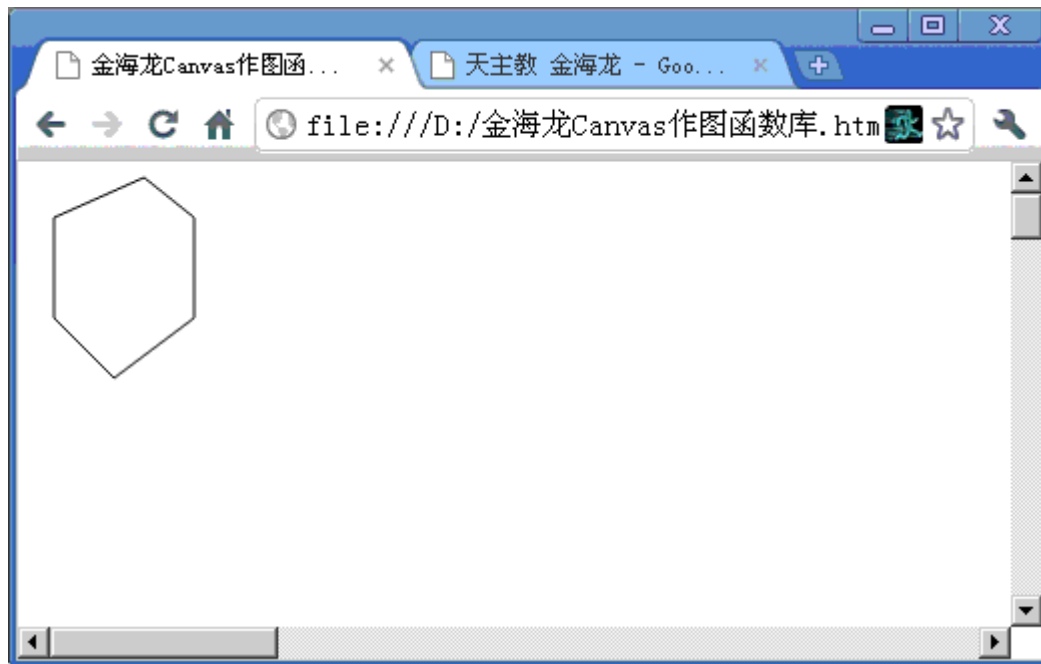
```
P[10]=55;
```

```
P[11]=10;
```



```
hb.beginPath();  
  
DrawPolyline(hb,P);  
  
hb.stroke();  
  
hb.endPath();  
  
</script>  
  
</body>  
  
</html>
```

用 **Google** 浏览器浏览的效果如下：



记住我的名字，用 Google 搜索就能找到我，**Google 搜索：**[金海龙 天主教](#)

所有的函数都写在“bigengineer.js”这个文件中，下载地址如下：

[http://www.filefactory.com/file/b3d9992/n/bigengineer\\_rar](http://www.filefactory.com/file/b3d9992/n/bigengineer_rar)

提示：在 FileFactory 下载这个文件，下载链接在底部，要等待 30 秒。

<http://www.docin.com/p-86276039.html>



如果大家觉得有必要谢谢函数编制的原理，欢迎在[我的博客](#)留言。

