# 第四章 HTML5中的canvas

1.canvas的基础知识

2.canvas绘制矩形

3.canvas绘制圆形

4.使用moveTo()和lineTo()绘制路径

5.北赛尔和二次方曲线

6.绘制变形图形

一) canvas基础

canvas是HTML5中新增的一个元素，专门用来绘制图形。在页面放置一个canvas元素，就相当于放置了一个画布，可以在其中进行图形的绘制。

但绘制图形，并不是指使用鼠标作画，而是需要javascript进行配合。创建canvas的时候，他默认的宽高为300px\*150px。

|  |
| --- |
| <canvas id="canvas">我是一个画布</canvas> |

二) 绘制矩形

我们在HTML页面中直接插入canvas标签，也可以对它设置宽度和高度，切记设置canvas的时候不可以利用css样式进行设置。

|  |
| --- |
| <canvas id="canvas" width="400" height="300">我是一个画布</canvas> |

那么，我们想在canvas上面画出矩形，我们就需要认识以下几种api

|  |  |
| --- | --- |
| 方法 | 描述 |
| getContext() | 获取绘图环境，可选参数"2d" |
| fillRect(x,y,width,height) | 绘制一个填充的矩形 |
| strokeRect(x,y,width,height) | 绘制一个矩形的边框 |
| clearRect(x,y,width,height) | 清除指定矩形区域。 |

|  |
| --- |
| <script>  var canvas = document.getElementById("canvas")  var ctx = canvas.getContext("2d")//获取绘制图片的"2d"环境  ctx.fillRect(50,50,100,100)//绘制填充的"黑色"的矩形  </script> |

|  |
| --- |
| <script>  var canvas = document.getElementById("canvas")  var ctx = canvas.getContext("2d")//获取绘制图片的"2d"环境  ctx.strokeRect(50,50,100,100)//绘制填充的"黑色"的矩形  </script> |

|  |
| --- |
| <script>  var canvas = document.getElementById("canvas")  var ctx = canvas.getContext("2d")//获取绘制图片的"2d"环境  ctx.fillRect(50,50,100,100)//绘制填充的"黑色"的矩形  ctx.clearRect(50,50,10,10)//清除"x=50,y=50,宽度为10,高度为10的区域"  </script> |

但有些时候，我们希望能够修改矩形的样式，又不可以直接通过修改css方式来修改。那么在canvas中也提供了api为我们来修改样式。

|  |  |
| --- | --- |
| 方法 | 描述 |
| fillStyle | 填充背景颜色 |
| strokeStyle | 设置边框颜色 |
| lineWidth | 设置边框的宽度 |

|  |
| --- |
| var canvas = document.getElementById("canvas")  var ctx = canvas.getContext("2d")//获取绘制图片的"2d"环境  ctx.fillStyle = "red"//绘制填充的"红色"的矩形  ctx.fillRect(50,50,100,100) |

|  |
| --- |
| var canvas = document.getElementById("canvas")  var ctx = canvas.getContext("2d")//获取绘制图片的"2d"环境  ctx.strokeStyle = "red"//绘制边框的"红色"的矩形  ctx.lineWidth = "10"//设置边框宽度  ctx.strokeRect(50,50,100,100) |

注意:在绘制矩形的过程中，必须是绘制样式在绘制图形之前。

**练习1：绘制一个简单的矩形。**

**练习2：在页面上绘制3个水平的矩形**

二) canvas绘制圆形

我们来看下如何在canvas中绘制一个圆形，绘制圆形分为4个步骤。

1) 开始绘制路径(beginPath)

2) 创建图像路径

3) 路径创建完成后，关闭路径

4) 设定绘制样式，调用绘制方法，绘制路径(closePtah)

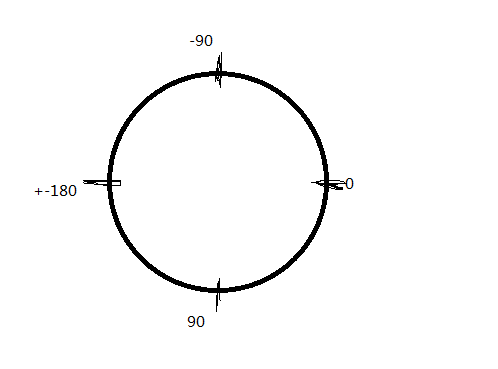
绘制圆形,canvas也为我们提供了arc这个方法,

|  |
| --- |
| ctx.arc(x,y,radius,startAngle,endAngle,anticlockwise) |

该方法有6个参数，x为绘制圆形的起点横坐标，y表示绘制圆形的起点的纵坐标,radius表示圆的半径，startAngle为开始角度，endAngle为结束的角度, anticlockwise是否按照顺时针方向进行绘制。false表示顺时针。（默认为顺时针）。

|  |
| --- |
| var canvas = document.getElementById("canvas")  var ctx = canvas.getContext("2d")//获取绘制图片的"2d"环境  ctx.beginPath()  ctx.arc(100,100,50,0,90\*Math.PI/180,false)  ctx.stroke()  ctx.closePath() |

其中有个弧度转换的公式：degreens\*Math.PI/180，绘制圆形默认按照以下的位置进行切换。



四) 使用moveTo()和lineTo()绘制路径

我们可以使用moveTo()和lineTo()绘制直线，或者是自己想要的图形，不局限于矩形。

|  |  |
| --- | --- |
| 方法 | 描述 |
| moveTo(x,y) | 不绘制，只是将当前位置移动到新的目标点 |
| lineTo(x,y) | 不仅将当前位置移动到新的目标点(x,y)，而且在两个坐标之间画一条直线。 |
| linejoin | 设置两线段连接处所显示的样子。round,bevel,miter |
| lineCap | 线段端点显示的样式 |
| save | 保存绘画路径 |
| restore | 销毁路径 |

注意:我们使用两个方法的时候，要注意使用closePath()闭合。因为它会通知canvas当前绘制的图形已经闭合或者形成了完全封闭。(linejoin)

练习3：绘制一个三角形

练习4：你画我猜小游戏

练习5：小方块运动

练习6：绘制笑脸火柴人

练习7：走动的时钟

五) 贝赛尔和二次方曲线（非重点）

贝塞尔曲线可以是二次方或三次方的形式，常用于绘制复杂而有规律的形状。

在canvas中贝塞尔曲线有两种方法

1) quadraticCurveTo(x1,y1,x2,y2)

第一组参数为控制点，第二组的参数则为结束坐标点。

1. bezierCurveTo(x1,y1,x2,y2,x,y)

第一组参数为控制点，第二组参数也是控制点，第三组参数则为结束坐标点。

6) 绘制变形图形

绘制图形的时候，经常会对绘制的图形进行变化，例如旋转。使用canvas的API的坐标轴变换处理，可以实现这种效果。

1. translate(平移)

移动图形的绘制主要通过translate实现。

Context.translate(x,y)

translate方法使用两个参数，x表示将坐标点原点向左移动若干个单位，y表示将坐标原点向下移动若干个单位。

1. scale(缩放)

使用图形上下文对象的scale方法将图形缩放。

Context.scale(x,y)

scale方法使用两个参数，x是水平方向的放大倍数，y是垂直方向放大的倍数。例如0.5则是将图形缩小到一半。

1. 旋转(rotate)

rotage方法只有一个参数-angle，即旋转的角度，旋转的中心点是坐标轴的原点。旋转以顺时针方向进行，要想逆时针，将angle设为负数即可。

**练习8：旋转的小方块**

**练习9：旋转加缩放的小方块**

七 总结

讲解了HTML5新增的一个画布---canvas功能以及伴随着这个元素而来的一套编程接口canvasAPI,以及如何使用canvasAPI绘制各种图形。