## canvas

```
 <canvas id="mycanvas" width="400px" height="400" style="border: solid 1px
green"/></canvas>
```

## 获取上下文

```
var mycanvas = document.getElementById("mycanvas");
var ctx = mycanvas.getContext("2d");
```

## 画直线

moveTo 移动到某一点准备状态 lineTo线的下一个点坐标 stroke() 绘制信号

```
ctx.moveTo(50,50);
ctx.lineTo(100,100);
ctx.lineTo(50,150);
ctx.lineWidth = "10";
ctx.strokeStyle = "red";
ctx.stroke();
```

## 画折线

beginPath()开始全新的绘制 closePath()绘制结束－闭合

填充与绘制的区别：

fill() stroke() fillStyle strokestyle

```
ctx.beginPath();
ctx.moveTo(100,50);
ctx.lineTo(150,100);
ctx.lineTo(100,150);
ctx.strokeStyle="blue";
ctx.closePath();
ctx.stroke();
```

## 画矩形

1、设置填充样式或者边框样式(ctx.fillStyle = "color";或者ctx.strokeStyle = "color";)

2、绘制矩形区域(如果采用fillRect或者strokeRect绘制矩形，可以省略第3步);

3、填充或者加上边框(ctx.fill();或者ctx.stroke();)

```
ctx.rect(50,50,100,100);
ctx.fillStyle = "green";
ctx.strokeStyle = "blue";
ctx.fill();
ctx.stroke();
```

自定义一组qq表情 画一个五角星

## 画圆弧

1、设置填充样式或者边框样式(ctx.fillStyle = "color";或者ctx.strokeStyle = "color";)

2、绘制圆形区域

ctx.arc(x,y,r,startAngle,endAngle,boolean);

r代表半径，startAngle,endAngle分别代表开始角度和结束角度，最后一个参数表示true(逆时针),false(顺时针)

3、填充或者加上边框(ctx.fill();或者ctx.stroke();)

注：1、在绘制圆弧的时候，必须使用ctx.beginPath()开始和ctx.closePath()结束，当然，前提条件时你要画规则的图形

```
    2、这里的开始角度和结束角度不是deg，而是Math.PI*(相应的值)。

ctx.beginPath();
ctx.arc(200,200,80,0,Math.PI*2,false);
ctx.fillStyle="yellow";
ctx.fill();
ctx.closePath();
ctx.beginPath();
ctx.arc(160,200,70,0,Math.PI*2,false);
ctx.fillStyle="white";
ctx.fill();
```

## 绘制文字：

ctx.fillStyle = "color";或者ctx.strokeStyle = "color";

ctx.textBaseline = "";设置垂直对齐方式

ctx.textAlign = "";设置水平对齐方式

ctx.fillText("文本",x,y)（实心字）或者ctx.strokeText("文本",x,y)（字的轮廓）；

textBaseLine 文本基线

alphabetic top hanging middle ideographic bottom

textAlign为当前对齐方式

start end center left right

```
 ctx.textBaseline = "top";
ctx.textAlign = "center";
ctx.font = "italic 30px 微软雅黑";
ctx.fillText("青云",100,100);
```

## 绘制贝塞尔曲线

http://www.rgraph.net/blog/2013-january-an-example-of-the-html5-canvas-quadraticcurveto-function.html

http://blog.chinaunix.net/uid-20622737-id-3161025.html

context.bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y)

cp1x:第一个控制点x坐标 cp1y:第一个控制点y坐标 cp2x:第二个控制点x坐cp2y:第二个控制点y坐标x:终点x坐标 y:终点y坐标

绘制二次样条曲线 context.quadraticCurveTo(qcpx,qcpy,qx,qy) qcpx:二次样条曲线控制点x坐标 qcpy:二次样条曲线控制点y坐标

```
qx:二次样条曲线终点x坐标   qy:二次样条曲线终点y坐标



ctx.moveTo(20,200);
ctx.quadraticCurveTo(200,0,300,200);
ctx.stroke();
```

## 渐变

线性渐变： var lg= context.createLinearGradient(xStart,yStart,xEnd,yEnd) lg.addColorStop(offset,color);

径向渐变:

var rg=context.createRadialGradient(xStart,yStart,radiusStart,xEnd,yEnd,radiusEnd) 径向渐变（发散）颜色rg.addColorStop(offset,color)

xStart:发散开始圆心x坐标yStart:发散开始圆心y坐标 radiusStart:发散开始圆的半径 xEnd:发散结束圆心的x坐标 yEnd:发散结束圆心的y坐标

radiusEnd:发散结束圆的半径 offset:设定的颜色离渐变结束点的偏移量(0~1)

color:绘制时要使用的颜色

图片背景

运用createPattern方法 createPattern(image, repetitionStyle) "repeat" - 在各个方向上都对图像贴图。默认值。 "repeat-x" - 只在 X 方向上贴图。 "repeat-y" - 只在 Y 方向上贴图。 "no-repeat" - 不贴图，只使用它一次。

```
var lg=ctx.createLinearGradient(0,0,200,0);
    lg.addColorStop(0,"red");
    lg.addColorStop(0.5,"green");
    lg.addColorStop(1,"blue");
    ctx.fillStyle = lg;
    ctx.fillRect(0,0,200,200);

 var backgroundImg = new Image();
 backgroundImg.src = "borderimage.png";
 backgroundImg.onload = function (){
    var bakImg = ctx.createPattern(backgroundImg,"repeat");
    ctx.fillStyle = bakImg;
    ctx.fillRect(0,0,400,400);
}
```

## 图形阴影

shadowOffsetX阴影距形状的水平距离

ctx.shadowOffsetY 阴影距形状的垂直距离

ctx.shadowBlur阴影的模糊级别

ctx.shadowColor = "red";

```
ctx.arc(200,200,50,0,Math.PI*2,false);
ctx.fillStyle="red";
ctx.shadowOffsetX = 15;
ctx.shadowOffsetY = 15;
ctx.shadowBlur = 10;
ctx.shadowColor = "gray";
ctx.fill();
```

## 图形变形

平移 缩放 旋转 ralate scale rotate

```
ctx.translate(100,100);
ctx.rotate(Math.PI/6);
ctx.fillStyle="red";
ctx.fillRect(0,0,150,100);
```

## 图形组合：

在绘制完成第一个图形之后，加

context.globalCompositeOperation=type

再接着绘制第二个图形

type：

source-over（默认值）:在原有图形上绘制新图形

destination-over:在原有图形下绘制新图形

source-in:显示原有图形和新图形的交集，新图形在上，所以颜色为新图形的颜色

destination-in:显示原有图形和新图形的交集，原有图形在上，所以颜色为原有图形的颜色

source-out:只显示新图形非交集部分

destination-out:只显示原有图形非交集部分,是将交集的部分转化为透明

source-atop:显示原有图形和交集部分，新图形在上，所以交集部分的颜色为新图形的颜色

destination-atop:显示新图形和交集部分，新图形在下，所以交集部分的颜色为原有图形的颜色

lighter:原有图形和新图形都显示，交集部分做颜色叠加

xor:重叠飞部分不现实

copy:只显示新图形

## 保存恢复及清空画布：

ctx.save();//保存当前绘制的图形,

ctx.restore();//恢复到离他最近的那个save的状态

像刚交完工的房子，有很多垃圾，你需要清理，清理除一块区域，以后将作为你的卧室。

ctx.clearRect(x,y,w,h);