1、非零环绕规则：

1. 看一块区域是否填充
2. 从这个区域啦一条直线
3. 看和这条直线相交的轨迹
4. 如果顺时针轨迹+1
5. 如果逆时针轨迹-1
6. 所有轨迹的值计算出来，如果是非零那么填充，否则填充

2、（strokeRect，fillRect），rect的区别：

前者有独立路径，就像画线一样会beginPath，后者不是独立路径

3、渐变：

var ***linearGradient*** = ***ctx***.createLinearGradient(300,100,300,200);  
***linearGradient***.addColorStop(1,'pink');

……可以继续添加0.5，0.6等等，在0-1之间即可  
***linearGradient***.addColorStop(0,'blue');  
***ctx***.fillStyle = ***linearGradient***;  
***ctx***.fillRect(100,100,400,100);

4、饼图：

let ***myCanvas*** = ***document***.querySelector('canvas');  
let ***ctx*** = ***myCanvas***.getContext('2d');  
  
let ***w*** = ***ctx***.canvas.width;  
let ***h*** = ***ctx***.canvas.height;  
let ***x0*** = ***w*** / 2;  
let ***y0*** = ***h*** / 2;  
let ***radius*** = 150;  
let getRandomColor = () => {  
 let r = ***Math***.floor(***Math***.random() \* 256);  
 let g = ***Math***.floor(***Math***.random() \* 256);  
 let b = ***Math***.floor(***Math***.random() \* 256);  
 return `rgb(${r},${g},${b})`;  
};  
let ***data*** = [6, 30, 10, 8];  
let ***angleList*** = [];  
let ***total*** = 0;  
let ***startAngle*** = 0;  
***data***.forEach(item => {  
 ***total*** += item;  
});  
***data***.forEach(item => {  
 let angle = ***Math***.PI \* 2 \* (item / ***total***);  
 ***angleList***.push(angle);  
});  
***angleList***.forEach(item => {  
 let endAngle = ***startAngle*** + item;  
 ***ctx***.beginPath();  
 ***ctx***.moveTo(***x0***, ***y0***);  
 ***ctx***.arc(***x0***, ***y0***, ***radius***, ***startAngle***, endAngle);  
 ***ctx***.fillStyle = getRandomColor();  
 ***ctx***.fill();  
 ***startAngle*** = endAngle;  
})

练习：

let PieChart = function (ctx) {  
 this.ctx = ctx || ***document***.querySelector('canvas').getContext('2d');  
 this.w = this.ctx.canvas.width;  
 this.h = this.ctx.canvas.height;  
 this.x0 = this.w / 2 + 60;  
 this.y0 = this.h / 2;  
 this.radius = 150;  
 this.outLine = 20;  
 this.rectW = 30;  
 this.rectH = 16;  
 this.space = 20;  
};  
PieChart.prototype.init = function (data) {  
 this.drawPie(data);  
};  
PieChart.prototype.drawPie = function (data) {  
 this.transformAngle(data);  
 let startAngle = 0;  
 data.forEach((item, index) => {  
 let endAngle = startAngle + item.angle;  
 let color = this.getRandomColor();  
 this.ctx.beginPath();  
 this.ctx.moveTo(this.x0, this.y0);  
 this.ctx.arc(this.x0, this.y0, this.radius, startAngle, endAngle);  
 this.ctx.fillStyle = color;  
 this.ctx.fill();  
 this.drawTitle(startAngle, item.angle, color, item.title);  
 this.drawDesc(index, item.title);  
 startAngle = endAngle;  
 });  
};  
PieChart.prototype.getRandomColor = () => {  
 let r = ***Math***.floor(***Math***.random() \* 256);  
 let g = ***Math***.floor(***Math***.random() \* 256);  
 let b = ***Math***.floor(***Math***.random() \* 256);  
 return `rgb(${r},${g},${b})`;  
};  
  
PieChart.prototype.drawTitle = function (startAngle, angle, color, title) {  
 let edge = this.radius + this.outLine;  
 let edgeX = ***Math***.cos(startAngle + angle / 2) \* edge;  
 let edgeY = ***Math***.sin(startAngle + angle / 2) \* edge;  
 let outX = this.x0 + edgeX;  
 let outY = this.y0 + edgeY;  
 this.ctx.beginPath();  
 this.ctx.moveTo(this.x0, this.y0);  
 this.ctx.lineTo(outX, outY);  
 this.ctx.strokeStyle = color;  
 this.ctx.font = '14px Microsoft YaHei';  
 let textWidth = this.ctx.measureText(title).width;  
 if (outX > this.x0) {  
 this.ctx.lineTo(outX + textWidth, outY);  
 this.ctx.textAlign = 'left';  
 } else {  
 this.ctx.lineTo(outX - textWidth, outY);  
 this.ctx.textAlign = 'right';  
 }  
 this.ctx.stroke();  
 this.ctx.textBaseline = 'bottom';  
 this.ctx.fillText(title, outX, outY);  
};  
PieChart.prototype.drawDesc = function (index, title) {  
 this.ctx.fillRect(this.space,  
 this.space + index \* (this.rectH + 10),  
 this.rectW, this.rectH);  
 this.ctx.beginPath();  
 this.ctx.textAlign = 'left';  
 this.ctx.textBaseline = 'top';  
 this.ctx.font = '12px Microsoft YaHei';  
 this.ctx.fillText(title, this.space + this.rectW + 10, this.space + index \* (this.rectH + 10));  
};  
PieChart.prototype.transformAngle = data => {  
 let total = 0;  
 data.forEach(item => {  
 total += item.num;  
 });  
 data.forEach(item => {  
 item.angle = item.num / total \* ***Math***.PI \* 2;  
 });  
 ***console***.log(data);  
};  
let ***data*** = [  
 {  
 title: '15-20岁',  
 num: 6  
 }, {  
 title: '20-25岁',  
 num: 30  
 }, {  
 title: '25-30岁',  
 num: 10  
 }, {  
 title: '30岁以上',  
 num: 8  
 }  
];  
let ***pieChart*** = new PieChart();  
***pieChart***.init(***data***);

5、文字对齐方式：

type CanvasTextBaseline = "top" | "hanging" | "middle" | "alphabetic" | "ideographic" | "bottom";

6、