利用canavs绘制动态柱形图

最近，闲来无事。看了一点canvas的API，写了一点小东西，希望各位大神多多指教。

注释：可以直接复制，就能跑起来

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Document</title>

<style type="text/css"media="screen">

.columnMark{

width:600px;

padding-left:60px;

height:5rem;

text-align:left;

font-size:1.2rem;

margin:0 auto;

}

.queryMark{

margin-top:6px;

height:2rem;

line-height:2rem;

float:left;

margin-right:1rem;

margin-left:1rem;

}

.colorMark{

display:block;

width:2rem;

height:1.5rem;

background:blue;

float:left;

}

</style>

</head>

<body>

<script type="text/javascript"src="../js/jQuery-2.0.2.js"></script>

<canvas width="647"height="500"id='canvas'style="display:block;margin:0 auto;"></canvas>

<div class="columnMark"id="columnMark">

</div>

<script type="text/javascript">

window.onload=function(){

/\*canvas绘制柱形图\*/

var canvas,context,start,end,showcanvas,showcontext;

start=50;

//图表属性

var cWidth,cHeight,cMargin,cSpace,rSpace;

var cMarginSpace,cMarginHeight;

//栏属性

var bWidth,bMargin,totalBars,maxDataValue,markerVal,markerValHt;

var bWidthMargin;

//属性

var ctr,numctr,speed;

//轴属性

var totLabelsOnYAxis,imgURI,image;

var json=[{

'num':120,

'year':2011,

'color':'#9A32CD'

},{

'num':130,

'year':2012,

'color':'#98F5FF'

},{

'num':150,

'year':2013,

'color':'#C71585'

},{

'num':90,

'year':2014,

'color':'#BCEE68'

},{

'num':150,

'year':2015,

'color':'#B4CDCD'

},{

'num':150,

'year':2016,

'color':'#B22222'

},{

'num':150,

'year':2017,

'color':'#98FB98'

}];

function data(){

var data=json;

for(var i=0;i<data.length;i++){

data[i].num=Math.random()\*100;

}

transactionType(data);

}

setInterval(data,1000);

//图表显示的数值；

function transactionType(json){

var transactionData;

transactionData=json;

//创建图表函数调用；

barChart(transactionData);

//创建图表标记函数调用；

transactionName(transactionData);

}

//柱形图执行函数封装；

function barChart(json){

//取得要是显示的canvas画布；

showcanvas=document.getElementById('canvas');

//获取2D上下文；

showcontext=showcanvas.getContext('2d');

//动态创建离屏canvas;

canvas=document.createElement('canvas');

context=canvas.getContext('2d');

//把离屏canavs的宽高和要显示的canavs一样；

canvas.height=showcanvas.height;

canvas.width=showcanvas.width;

//获得的交易数据；

var data=json;

//绘制网格线函数调用；

gridLine(context,'lightgray',10,10,start,end);

//初始化图表和数组值函数调用；

chartSettings(data);

//绘制X轴

drawAxisLabelMarkers();

//创建柱形图函数调用

drawChartWithAnimation(data);

//柱形图上显示的数据函数调用；

axisdataY(data);

//绘制显示的图片函数调用；

drawImage();

}

//绘制柱形图网格线；

function gridLine(context,color,stepx,stepy,start){

context.save();

context.strokeStyle=color;

context.lineWidth=0.5;

//绘制纵轴线段；

for(var i=stepx+0.5;i<context.canvas.width-77;i+=stepx){

context.beginPath();

context.moveTo(start+i,30);

context.lineTo(start+i,context.canvas.height-20);

context.stroke();

context.closePath();

}

//绘制横向线段；

for(var i=stepy+0.5+20;i<context.canvas.height-10;i+=stepy){

context.beginPath();

context.moveTo(60,i);

context.lineTo(context.canvas.width-37,i);

context.stroke();

context.closePath();

}

context.restore();

}

//初始图表和数组值；

function chartSettings(data){

//X轴距离画布左边的距离

cMargin=30;

//Y轴距离画布下边的距离；

cSpace=20;

//X图表距离画布右边的距离；

rSpace=47;

//X轴的长度；

cWidth=canvas.width-cMargin-rSpace-20;

//Y轴的长度；

cHeight=canvas.height-cMargin-cSpace;

cMarginSpace=cWidth+cMargin;

cMarginHeight=cMargin+cHeight;

//

bMargin=15;

totalBars=data.length;

if(totalBars<=5){

bWidth=40;

}else{

bWidth=(cWidth/totalBars)-bMargin;

}

//在图表上找到最大值

maxDataValue=0;

for(var i=0;i<totalBars;i++){

var arrVal=data[i].num;

var barVal=parseInt(arrVal);

if(parseInt(barVal)>parseInt(maxDataValue))

maxDataValue=barVal;

}

totLabelsOnYAxis=10;

context.font="bolder 10px Arial";

//初始化动画变量

ctr=0;

numctr=100;

speed=10;

}

//绘制图表轴，标签和标记

function drawAxisLabelMarkers(){

context.lineWidth="2.0";

context.fillStyle="#333";

//绘制Y轴

drawAxis(cMargin+cMargin,cMarginHeight,cMargin+cMargin,cMargin-10);

//绘制X轴；

drawAxis(cMargin+cMargin,cMarginHeight,cMarginSpace+40,cMarginHeight);

context.lineWidth="1.0";

drawMarkers();

}

//绘制Y轴和X轴函数封装；

function drawAxis(x,y,X,Y){

context.beginPath();

context.moveTo(x,y);

context.lineTo(X,Y);

context.closePath();

context.stroke();

}

//X和Y轴上的绘制图表标记

function drawMarkers(){

//对最大值除以一个整数totLabelsOnYAxis=10；并进行取整；

var numMarkers=parseInt(maxDataValue/totLabelsOnYAxis);

context.textAlign="right";

context.fillStyle="#333";

context.textAlign='center';

//添加Y轴标题

context.fillText('数值',cMargin+cMargin,cMargin-15);

//添加X轴标题

context.fillText('年份',canvas.width-12,cMarginHeight+cSpace/3);

}

function drawChartWithAnimation(data){

//通过循环创建出矩形柱子

for(var i=0;i<totalBars;i++){

var arrVal=data[i].num;

var bVal=parseInt(arrVal);

var bHt=(bVal\*cHeight/maxDataValue)/numctr\*ctr;

var bX=bMargin+bMargin+(i\*(bWidth+bMargin))+bMargin+cMargin;

var bY=cMarginHeight-bHt-2;

var color=data[i].color;

drawRectangle(bX,bY,bWidth,bHt,color);

}

//超时运行，检查是否已达到

//所需高度；如果没有，保持生长;

if(ctr<numctr){

ctr=ctr+1;

setTimeout(drawChartWithAnimation(data),10);

}

}

//显示每个柱形图的具体数据；

function axisdataY(data){

//循环创建矩形

context.textAlign="center";

context.fillStyle="#333";

for(var i=0;i<totalBars;i++){

var arrVal=data[i].num;

var bVal=parseInt(arrVal);

var bHtext=(bVal\*cHeight/maxDataValue);

var bX=bMargin+bMargin+(i\*(bWidth+bMargin))+bMargin+bWidth/2+cMargin;

var bYtext=cMarginHeight-bHtext-10;

context.fillText(bVal,bX,bYtext);

}

}

//生成柱形函数封装；

function drawRectangle(x,y,w,h,color){

context.beginPath();

context.rect(x,y,w,h);

context.fillStyle=color;

context.strokeStyle=color;

context.fill();

context.closePath();

context.stroke();

}

//把离屏的canvas绘制到实际显示画布上；

function drawImage(){

//取得图像的数据URI

imgURI=canvas.toDataURL("image/png");

//显示图像

image=document.createElement("img");

image.src=imgURI;

//加载生成的图片；

image.onload=function(){

//每次图片加载完成以后首先清空原来的要显示的showcanavs画布；

showcontext.clearRect(0,0,canvas.width,canvas.height);

//然后在所要生成的图片绘制到showcanavs画布上；

showcontext.drawImage(image,0,0,canvas.width,canvas.height);

//最后再清空动态生成的canvas.

context.clearRect(0,0,canvas.width,canvas.height);

}

}

//每个柱形图表示函数封装；

function transactionName(data){

//先清空原来的内容；

$('#columnMark').html('');

var queryMark="";

for(var i=0;i<data.length;i++){

var colName=data[i].year;

console.log(data[i].color);

queryMark=$('<div class="queryMark">'+

'<span class="colorMark"style="background-color:'+data[i].color+'"></span>'+

'<span class="textMark">'+colName+'</span>'+

'</div>');

$('#columnMark').append(queryMark);

}

}

}

</script>

</body>

</html>