

实验 06：使用 Canvas 绘制图表

一、实验目的

- 1、掌握 Canvas 元素的核心特性，能熟练操作 Canvas 进行图形渲染；
- 2、学会用 Canvas API 绘制基础图形及文本，理解图形变换的应用；
- 3、掌握数据到图形的映射逻辑，能实现常见统计图表；
- 4、实现图表的交互功能，提升数据可视化设计与交互能力。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 Visual Studio Code，以及 Edge 浏览器。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

- 1、绘制静态折线图

- 2、绘制动态柱状图
- 3、绘制饼图
- 4、绘制散点图

六、实验内容及步骤

1、绘制静态折线图

步骤 1：创建 HTML 结构

XML/HTML

```
1 <canvas id="lineChart" width="800" height="400"></canvas>
```

步骤 2：实现 JavaScript 代码，示例如下。

```
1 const ctx = document.getElementById('lineChart').getContext('2d');
2 // 1. 定义数据与配置
3 const data = [22, 25, 19, 28, 26, 23, 27]; // 气温数据
4 const labels = ['周一', '周二', '周三', '周四', '周五', '周六', '周日']; // X轴标签
5 const padding = { top:40, right:40, bottom:60, left:60 }; // 内边距
6 const w = ctx.canvas.width, h = ctx.canvas.height;
7 const chartW = w - padding.left - padding.right;
8 const chartH = h - padding.top - padding.bottom;
9
10 // 2. 计算坐标映射 (y轴翻转: Canvas原点在左上角)
11 const maxVal = Math.max(...data);
12 const xStep = chartW / (data.length-1); // X轴点间距
13 const yScale = chartH / maxVal; // Y轴缩放比例
14
15 // 3. 绘制坐标轴与标签
16 ctx.beginPath(); ctx.moveTo(padding.left, h-padding.bottom); ctx.lineTo(
    (w-padding.right, h-padding.bottom));
17 ctx.strokeStyle='#333'; ctx.stroke(); // X轴
18 ctx.beginPath(); ctx.moveTo(padding.left, padding.top); ctx.lineTo(padding.
    left, h-padding.bottom);
19 ctx.stroke(); // Y轴
20 // X轴标签 (星期)
21 ctx.font='14px Arial'; ctx.textAlign='center';
22 labels.forEach((l,i)=>ctx.fillText(l, padding.left+i*xStep, h-padding.b
    ottom+20));
23 // Y轴标签 (温度)
24 ctx.textAlign='right';
25 for(let i=0;i<=maxVal;i+=5) {
26     const y = h-padding.bottom - i*yScale;
27     ctx.fillText(`${i}°C`, padding.left-10, y+4);
28 }
29
30 // 4. 绘制折线与填充
31 ctx.beginPath(); data.forEach((v,i)=> {
32     const x=padding.left+i*xStep, y=h-padding.bottom - v*yScale;
33     i===0?ctx.moveTo(x,y):ctx.lineTo(x,y);
34 }); ctx.strokeStyle='#2196F3'; ctx.lineWidth=2; ctx.stroke();
35 // 填充折线下方
36 ctx.beginPath(); data.forEach((v,i)=> {
37     const x=padding.left+i*xStep, y=h-padding.bottom - v*yScale;
38     i===0?ctx.moveTo(x,h-padding.bottom):ctx.lineTo(x,y);
39     i===data.length-1&&ctx.lineTo(x,h-padding.bottom);
```

```
40 }); ctx.fillStyle='rgba(33,150,243,0.2)'; ctx.fill();
41
42 // 5. 绘制数据点
43 data.forEach((v,i)=> {
44     const x=padding.left+i*xStep, y=h-padding.bottom - v*yScale;
45     ctx.beginPath(); ctx.arc(x,y,4,0,Math.PI*2); ctx.fillStyle='#2196F
46     3'; ctx.fill();
47 });
```

2、绘制动态柱状图

步骤 1：创建 HTML 结构

XML/HTML

```
1 <canvas id="barChart" width="800" height="400"></canvas>
```

步骤 2：实现 JavaScript 逻辑

JavaScript

```
1 let salesData = [120,150,180,90,200,160,220]; // 初始销售额
2 const categories = ['1月','2月','3月','4月','5月','6月','7月'];
3 const config = {
4   padding: {top:40,right:40,bottom:60,left:60},
5   barWidth:40, barSpacing:20,
6   colors: ['#2196F3','#4CAF50','#FFC107','#E91E63','#9C27B0','#00BCD4','#FF5722']
7 };
8 const ctx = document.getElementById('barChart').getContext('2d');
9 const w = ctx.canvas.width, h = ctx.canvas.height;
10 const chartW = w - config.padding.left - config.padding.right;
11 const chartH = h - config.padding.top - config.padding.bottom;
12
13 // 绘制柱状图函数（封装重绘逻辑）
14 function drawBarChart() {
15   ctx.clearRect(0,0,w,h); // 清除画布
16   // 绘制坐标轴（复用任务1逻辑，此处省略）
17   // ...
18   // 绘制柱子
19   const totalSpace = config.barWidth + config.barSpacing;
20   const startX = config.padding.left + totalSpace/2 - config.barWidth/2;
21   salesData.forEach((v,i)=>{
22     const x=startX+i*totalSpace,
23           y=config.padding.top + chartH - (v/Math.max(...salesData))*chartH,
24           height=(v/Math.max(...salesData))*chartH;
25     ctx.fillStyle=config.colors[i]; ctx.fillRect(x,y,config.barWidth,height);
26     // 数据标签
27     ctx.font='14px Arial'; ctx.textAlign='center';
28     ctx.fillText(v,x+config.barWidth/2,y-10);
29     // 分类标签
30     ctx.fillText(categories[i],x+config.barWidth/2,h-config.padding.bottom+20);
31   });
32 }
33 drawBarChart(); // 初始绘制
34 // 动态更新：每3秒替换数据
35 setInterval(()=>{
36   salesData = salesData.map(()=>Math.floor(Math.random()*200)+50);
```

```
37     drawBarChart();  
38 }, 3000);
```

3、绘制饼图

步骤 1：创建 HTML 结构

XML/HTML

```
1 <div style="display:flex;align-items:center;">  
2   <canvas id="pieChart" width="400" height="400"></canvas>  
3   <div id="pieLegend" style="margin-left:20px;"></div>  
4 </div>
```

步骤 2：实现 JavaScript 逻辑

```
1 const productData = [
2   {name:'手机',sales:1200}, {name:'电脑',sales:800},
3   {name:'平板',sales:600}, {name:'耳机',sales:400}
4 ];
5 const ctx = document.getElementById('pieChart').getContext('2d');
6 const centerX=ctx.canvas.width/2, centerY=ctx.canvas.height/2, radius=1
   50;
7
8 // 计算百分比与角度
9 const totalSales = productData.reduce((sum,item)=>sum+item.sales,0);
10 let startAngle=0;
11 productData.forEach(item=>{
12   item.percentage=(item.sales/totalSales)*100;
13   item.startAngle=startAngle;
14   item.endAngle=startAngle + (item.sales/totalSales)*Math.PI*2;
15   startAngle=item.endAngle;
16 });
17
18 // 绘制饼图
19 productData.forEach((item,i)=>{
20   ctx.beginPath(); ctx.moveTo(centerX,centerY);
21   ctx.arc(centerX,centerY,radius,item.startAngle,item.endAngle);
22   ctx.closePath(); ctx.fillStyle=`hsl(${i*90},70%,60%)`; ctx.fill();
23 });
24
25 // 绘制图例
26 const legend=document.getElementById('pieLegend');
27 productData.forEach((item,i)=>{
28   const div=document.createElement('div'); div.style.marginBottom='5p
   x'; div.style.display='flex'; div.style.alignItems='center';
29   // 颜色块
30   const colorBox=document.createElement('div'); colorBox.style.width='1
   4px'; colorBox.style.height='14px';
31   colorBox.style.backgroundColor=`hsl(${i*90},70%,60%)`; colorBox.styl
   e.marginRight='8px';
32   // 文本
33   const text=document.createElement('span'); text.textContent=`${item.n
   ame}:${item.percentage.toFixed(1)}%(${item.sales})`;
34   div.appendChild(colorBox); div.appendChild(text); legend.appendChild
   (div);
35 });
```

4、绘制散点图

步骤 1：创建 HTML 结构

XML/HTML

```
1 <canvas id="scatterChart" width="800" height="400"></canvas>
2 <div id="scatterTooltip" style="position:absolute;display:none;background:
  nd:rgba(0,0,0,0.7);color:white;padding:5px;border-radius:4px;"></div>
```

步骤 2：实现 JavaScript 逻辑

JavaScript

```
1 const scatterData = [
2   {height:160,weight:50},{height:165,weight:55},{height:170,weight:60},
3   {height:175,weight:65},{height:180,weight:70},{height:185,weight:75}
4 ];
5 const ctx=document.getElementById('scatterChart').getContext('2d');
6 const tooltip=document.getElementById('scatterTooltip');
7 const padding={top:40,right:40,bottom:60,left:60};
8 const w=ctx.canvas.width, h=ctx.canvas.height;
9 const chartW=w-padding.left-padding.right, chartH=h-padding.top-paddin
  g.bottom;
10
11 // 坐标映射
12 const maxHeight=Math.max(...scatterData.map(d=>d.height)), maxWeight=Ma
  th.max(...scatterData.map(d=>d.weight));
13 const xScale=chartW/maxWeight, yScale=chartH/maxHeight;
14
15 // 绘制散点图
16 function drawScatter() {
17   ctx.clearRect(0,0,w,h);
18   // 绘制坐标轴 (复用任务1逻辑)
19   // ...
20   // 绘制散点
21   scatterData.forEach(point=>{
22     const x=padding.left+point.weight*xScale, y=h-padding.bottom - poin
      t.height*yScale;
23     ctx.beginPath(); ctx.arc(x,y,6,0,Math.PI*2); ctx.fillStyle='#E91E6
      3'; ctx.fill();
24   });
25 }
26 drawScatter();
27 // hover交互
28 ctx.canvas.addEventListener('mousemove', (e)=>{
29   const rect=ctx.canvas.getBoundingClientRect();
30   const mouseX=e.clientX-rect.left, mouseY=e.clientY-rect.top;
31   let isHover=false;
32   scatterData.forEach(point=>{
33     const x=padding.left+point.weight*xScale, y=h-padding.bottom - poin
      t.height*yScale;
34     const dist=Math.sqrt((mouseX-x)**2+(mouseY-y)**2);
35     if(dist<=6) {
36       isHover=true;
37       tooltip.style.display='block'; tooltip.style.left=`${e.clientX+1
```

```
0}px`; tooltip.style.top=`${e.clientY+10}px`;
38     tooltip.textContent=`身高:${point.height}cm 体重:${point.weight}kg
    `;
39     }
40 });
41 if(!isHover) tooltip.style.display='none';
42 });
43 ctx.canvas.addEventListener('mouseout',()=>tooltip.style.display='none');
```

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

- (1) 学生演示汇报实验内容的完成情况，实验指导老师现场打分。
- (2) 指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。
- (3) 实验考核结束后，进行公布成绩。