

canvas

`canvas` 是 HTML5 的新特性，它允许我们使用 `canvas` 元素在网页上通过 JavaScript 绘制图像。

canvas 标签

`<canvas>` 标签只是图形容器，相当于一个画布，`canvas` 元素本身是没有绘图能力的。所有的绘制工作必须在 JavaScript 内部完成，相当于使用画笔在画布上画画。

注意：必须指定宽高

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

getContext()

`context` 是一个封装了很多绘图功能的对象，我们在页面中创建一个 `canvas` 标签之后，首先要使用 `getContext()` 获取 `canvas` 的上下文环境，目前 `getContext()` 的参数只有 2d，暂时还不支持 3d

`getContext("2d")` 对象是内建的 HTML5 对象，拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

```
const canvas = document.getElementById('charts');  
const context = canvas.getContext('2d');
```

绘制线段

- `moveTo(x, y)`：把路径移动到画布中的指定点，不创建线条
- `lineTo(x, y)`：添加一个新点，然后在画布中创建从该点到最后指定点的线条

```
context.strokeStyle = 'yellowgreen';  
context.moveTo(0, 0);  
context.lineTo(100, 100);  
context.stroke();
```

绘制矩形

- `fillRect(x, y, width, height)` 绘制填充颜色的矩形
- `strokeRect(x, y, width, height)` 绘制线条的矩形

```
context.fillStyle = "pink";
context.strokeStyle = "darkred";
context.fillRect(0, 0, 100, 100);
context.strokeRect(120, 0, 100, 100);
```

绘制圆形

- `arc(x, y, radius, starAngle, endAngle, anticlockwise)`
 - x: 圆心的 x 坐标
 - y: 圆心的 y 坐标
 - radius: 半径
 - starAngle: 开始角度
 - endAngle: 结束角度
 - anticlockwise: 是否逆时针 (true) 为逆时针, (false) 为顺时针

```
context.beginPath();
context.arc(300, 350, 100, 0, Math.PI * 2, true);
//不关闭路径路径会一直保留下去
context.closePath();
context.fillStyle = 'rgba(0,255,0,0.25)';
context.fill();
// 或
// context.stroke(); // 此时就会有问题
```

清除路径

清除绘画的路径，多个图形就不会连接在一起

```
context.beginPath()
context.closePath()
```

清除矩形区域

当在一个画布反复绘制图形，需要将上一次的图形清空

- `clearRect(x, y, width, height)`

绘制文字

- `fillText(text, x, y, maxWidth)`

案例：绘制一个柱状图

::: tip

一个少女心满满的例子带你入门 Canvas

:::