

# 第3章 构建UI界面

## 第7节 画布组件

谢涛



# 内容提要

## ➤ 画布组件



## 画布组件

- canvas画布组件用来绘制正方形、圆形或一些其他形状
- 该组件默认宽度为300px、高度为150px
- 使用时需要有唯一的标识
- 使用wx.createCanvasContext获取绘图上下文
- canvas画布属性如下表



# canvas画布属性

属性	类型	默认值	说明
canvas-id	string		canvas 组件的唯一标识符，若指定了 type 则无需再指定该属性
disable-scroll	boolean	false	当在 canvas 中移动时且有绑定手势事件时，禁止屏幕滚动以及下拉刷新
bindtouchstart	eventhandle		手指触摸动作开始
bindtouchmove	eventhandle		手指触摸后移动
bindtouchend	eventhandle		手指触摸动作结束
bindtouchcancel	eventhandle		手指触摸动作被打断，如来电提醒，弹窗
bindlongtap	eventhandle		手指长按 500ms 之后触发，触发了长按事件后进行移动不会触发屏幕的滚动
binderror	eventhandle		当发生错误时触发 error 事件，detail = {errMsg}

## 示例代码

```
<!--index.wxml-->  
<view class="container">  
  <canvas style="width: 300px; height: 200px;" canvas-  
    id="firstCanvas"></canvas>  
  <!-- 当使用绝对定位时，文档流后边的 canvas 的显示层级高于前边的 canvas -->  
  <canvas style="width: 400px; height: 500px;" canvas-  
    id="secondCanvas"></canvas>  
  <!-- 因为 canvas-id 与前一个 canvas 重复，该 canvas 不会显示，并会发送一个  
    错误事件到 AppService -->  
  <canvas style="width: 400px; height: 500px;" canvas-id="secondCanvas"  
    binderror="canvasIdErrorCallback"></canvas>  
</view>
```



```
Page({
  canvasIdErrorCallback: function (e) {
    console.error(e.detail.errMsg)
  },
  onReady: function (e) {
    // 使用 wx.createContext 获取绘图上下文
    context

    var context =
    wx.createCanvasContext('firstCanvas')

    context.setStrokeStyle("#00ff00")
    context.setLineWidth(5)
    context.rect(0, 0, 200, 200)
    context.stroke()
    context.setStrokeStyle("#ff0000")

    context.setLineWidth(2)
    context.moveTo(160, 100)
    context.arc(100, 100, 60, 0, 2 *
    Math.PI, true)
    context.moveTo(140, 100)
    context.arc(100, 100, 40, 0, Math.PI,
    false)
    context.moveTo(85, 80)
    context.arc(80, 80, 5, 0, 2 * Math.PI,
    true)
    context.moveTo(125, 80)
    context.arc(120, 80, 5, 0, 2 *
    Math.PI, true)
    context.stroke()
    context.draw()
  }
})
```





## 运行效果

