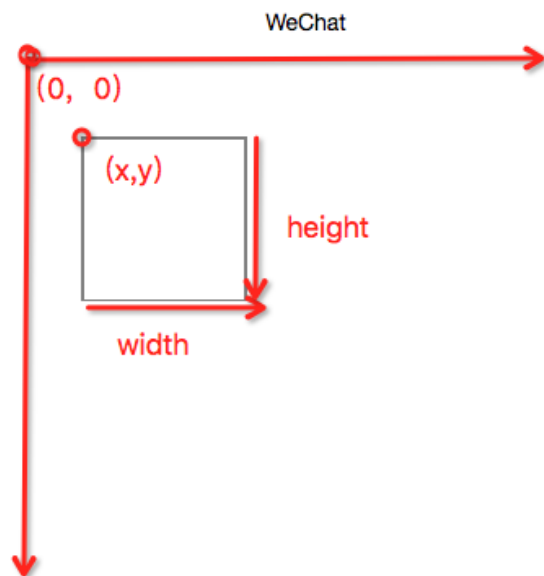


微信小程序把玩《十四》： canvas API



绘图是每个移动应用必备的技术，基本上和 Android，IOS，等移动开发都是相同的，创建个上下文，给你个画布再上画，官网给的小例子都比较全了自己去看吧，drawImage 时没有反应不知道是 BUG 还是电

脑不能测试待定，<http://wxopen.notedown.cn/api/api-canvas.html>

屏幕就像是数学上的坐标轴，且在第四象限，以屏幕左上角为圆点，X 轴向右为正向左为负，Y 轴向下为正向上为负（这点和数学上相反的）以圆点为基点画个距离圆点上下 50 宽高 100 的矩形来演示 canvas 基本用法

微信小程序这里提供了两个 API

- `wx.createContext()` 创建并返回绘图上下文 context 对象
 - `getActions` 获取当前 context 上存储的绘图动作，对应 `wx.drawCanvas(object)` 中的 `actions`
 - `clearActions` 清空当前的存储绘图动作
- `wx.drawCanvas(object)` 绘制
 - `canvasId` 画布标识，传入的 `cavas-id`，这里的标识可以为 `Number`，也可以是 `String`
 - `actions` 绘图动作数组，由 `wx.createContext` 创建的 context，调用 `getActions` 方法导出绘图动作数组。

绘图中可以进行变形，绘制，路径，样式，这些个东西有点多官网有例子，这里通过一个例子引入 wxml

<!--画布

`canvas-id` 为画布标识，当绘制时通过 `canvas-id` 找到画布

-->

<canvas canvas-id="identify"/>

js

```
Page({
  data:{
    text:"Page canvas"
```

```
},
onLoad:function(options){

    // 页面初始化 options 为页面跳转所带来的参数

},
onReady:function(){

    // 页面渲染完成

    //第一步创建个上下文容器

    var context = wx.createContext();

    //第二步绘制这里我们绘制个矩形

    //x, y, width, height
    context.rect(50, 50, 100, 100);

    //绘制的样式进行描边绘制, fill 为填充位置

    context.stroke();
    /**
     * 调用 wx.drawCanvas, 通过 canvasId 指定在哪张画布上绘制, 通过 actions 指定绘制行为
     *
     * 注意 canvasId 可以为数字表示也可以用字符串表示, 就是一个绘制对象的标识, 并且可以指定多个
     * actions 是从 context 上下文中获取的绘制行为, 即为第二步操作
     */

    wx.drawCanvas({
```

```
    //画布标识，传入<canvas/>的 canvas-id
    canvasId: 'identify',

    //获取绘制行为， 就相当于你想做到菜 context.getActions() 就是食材
    actions: context.getActions(),
  })

},
onShow: function() {

    // 页面显示

},
onHide: function() {

    // 页面隐藏

},
onUnload: function() {

    // 页面关闭

}
})
```