**canvas**：

就是HTML5的一个标签，提供一个空白的图形区域，起到画布的作用，用来展示，本身不具有绘图功能

需要通过**js中的绘制API**来绘制

clipboard.png

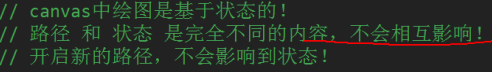
OpenGL专门处理3D图形

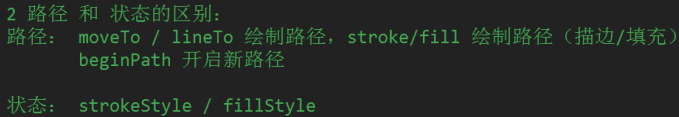
webgl浏览器端处理3D效果的

**区分rect和fillRect**



fillRect不会影响之前的红色，rect会影响





canvas逐渐取代Flash，比Flash更加立体，更加精巧，**canvas游戏在流畅度和跨平台方面更牛**

**可视化数据（数据图表化）： 百度的 echarts或者**

**3D动态效果： 可以通过 three.js官网**



canvas默认尺寸为：300（宽） \* 150（高）

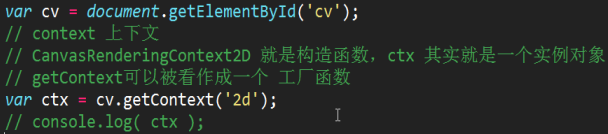


**canvas使用步骤：**

获取到canvas标签

获取到 当前canvas标签的绘制上下文

调用 上下文（对象）中的绘图API进行绘制

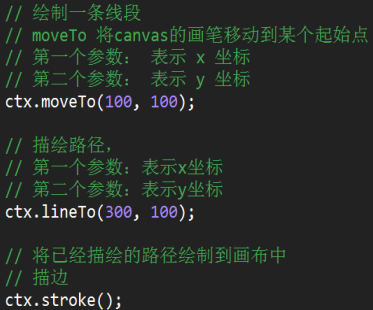


？？？？ DOM.$('')

**方法：**

ctx.moveTo(x,y) ：将画笔移到起始点

ctx.lineTo(x,y) ： 描绘路径，如果没有moveTo，第一个lineTo就相当于moveTo



ctx.stroke() ：将已经描绘的路径 绘到画布中

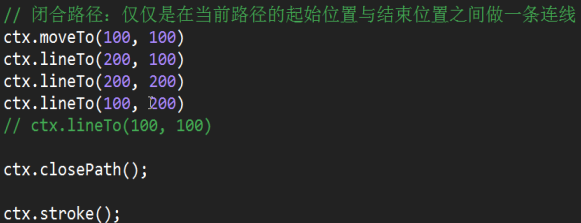
ctx.fill() ：填充，如果描绘的路径没有闭合，会自动闭合，再填充

ctx.beginPath() ：**开启新路径**，相当于将之前绘制上下文中描绘的路径清除掉

**调用该方法后，再 调用fill或stroke方法，就不会描绘以前的路径了**，只描绘新路径的

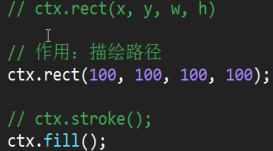
**路径的操作不会影响绘制状态（strokeStyle或fillStyle），也就是beginPath之前绘制的状态，后面再 fill/stroke 的时候也会拥有此状态**

ctx.closePath() ：闭合路径,比如画正方形，只画3条边，然后调用ctx.closePath()会自动连最后一条闭合



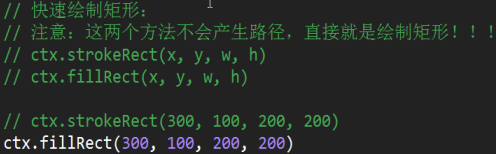
ctx.rect（x,y,w,h）： x坐标，y坐标，宽，高

绘制矩形的路径 ，还需要配合stroke或者fill



ctx.strokeRect（x,y,w,h）：

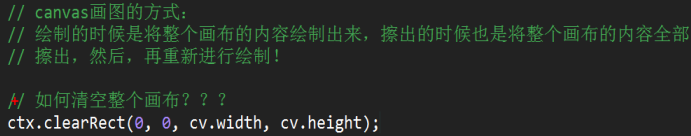
ctx.fillRect（x,y,w,h）：



不产生路径，直接绘制

ctx.clearRect（x,y,w,h）：canvas的API只提供了 一个 清除绘制内容的方法

作用：清除某个矩形区域里的内容，仅仅是擦除展示的内容，不会影响路径



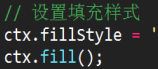
ctx.clearRect（x,y,w,h）

**属性：**

ctx.strokeStyle：



ctx.fillStyle：



颜色同上

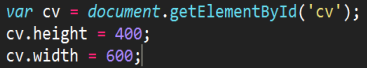
注意：一定先写 fillStyle= 之后再写 fill（）

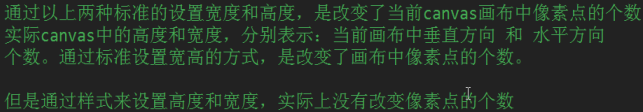
**设置canvas宽高：**

第一种

clipboard.png

第二种





注意：不能通过css设置宽高，因为通过css设置的宽高，相当于把canvas拉伸了，绘制图形时 是变形的

**canvas坐标：**

左上角是原点

向右 X正方向

向下 Y正方向

**canvas中的上下文（说白了就是对象）：**

相当于画图软件的工具栏，里面提供了很多与绘图相关的工具

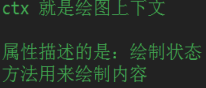
clipboard.png

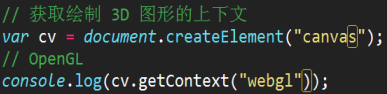
获取绘图平面图形的上下文

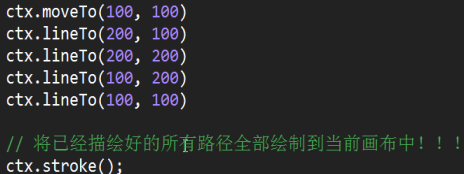
**var cv = document.getElementById('dv');**

ctx就是绘图上下文，里面包含与绘图相关的各种API（颜色，绘制线等方法）

**var ctx = cv.getContext('2d');**

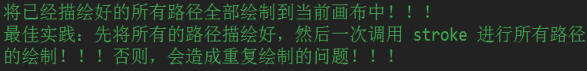






画多个图形，也最后只写一个stroke，此方法绘制所有路径，多写会重复，线会变粗

**stroke（）：**

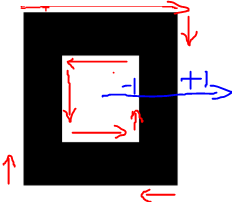


ctx.beginPath（）



非零环绕原则：用来查看某个区域是否被填充的一个规则

从要查看的区域引出一条射线，穿过整个图形，使射线与图形相交，并且以射线起点为圆心，顺着线绘制方向画圆，**顺时针**形成的圆记为**+1**，**逆时针-1**，所有结果相加，结果为0，射线原点的区域不绘制，**结果不为0，绘制**



**边擦边画，运动的矩形（**不会很影响性能，比如做游戏也相同道理**）**

