HTML5-太阳系

作为HTML5中的新功能，也是一个非常重要的功能,学习HTML5离不开Canvas（画布）。上课的时候学习了canvas，老师也让我们把菜鸟教程里太阳系的例子拷贝了下来运行了一下，下课之后，我又仔细研究了一下这个例子的代码。

html代码中只有一个canvas,js代码中，首先声明三个image对象，然后调用document.getElementById()方法获取canvas组件，然后调用getContext(“2d”)方法获得canvas内置的HTML5实体对象。之后为了显示图片，需要将drawimage()方法放在image对象的onload方法中(我在网上查了一下，大体意思如果不把drawimage()方法放在onload()里面,图片还没加载出来的时候就开始绘制了，当然会绘制失败。)

在绘制方法draw()里面，首先要执行clearRect()方法清空所有的内容，如果不清空的话下一帧会覆盖在上一帧上面。清空了之后就开始绘制。首先绘制太阳，这个比较简单，因为太阳是不动的，然后要保存一下此时的状态（此时的坐标原点为canvas左上角的点），因为后面会改变坐标原点的位置，所以这里先保存下来，方便下一帧画太阳时再使用这个状态，绘制完太阳，再来绘制地球轨道，为了方便，我们把坐标原点移到太阳的圆心，这里就需要用到translate()方法，然后画一个圆心为（0,0）的圆，这就是地球轨道了。画完地球轨道，接下来就是画地球了，与太阳和地球轨道不同，地球是不停转动的，所以我们需要设置一个规则，即地球每秒转多少，为了达到连续性，我们可以设置地球每毫秒转多少，这样在人眼看来就是一直转的了。ctx.rotate(2 \* Math.PI / 60 \* time.getSeconds() + 2 \* Math.PI / 60000 \* time.getMilliseconds())，这是例子中给的代码，观察可以发现，动画上的地球每秒转动6度，每毫秒转动0.006度，这样，经过60秒，地球刚好转一圈。旋转了坐标系之后，再调用translate()方法将坐标原点移到地球轨道上（现在的坐标原点在太阳的圆心），然后就可以绘制地球了。绘制完地球，接下来要绘制月球轨道，月球轨道和地球是相对静止的，所以坐标原点不需要移动，直接绘制月球轨道即可，最后要绘制月球，绘制月球和绘制地球差不多，都是先根据时间旋转坐标系，然后将坐标原点移到月球轨道上，最后画出月球，不同的是月球转动的速度和地球转动的速度不同，ctx.rotate(2 \* Math.PI / 6 \* time.getSeconds() + 2 \* Math.PI / 6000 \* time.getMilliseconds());月球是每秒转动60度，每毫秒转动0.06度。画完月球之后就完成了一帧的绘制，然后调用restore()方法取出之前存的状态，然后开始下一帧的绘制。为了形成动画的效果，需要重复调用draw()方法，requestAnimationFrame(draw)就可以实现重复调用draw()方法，实现动画的效果。