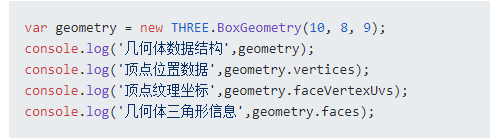
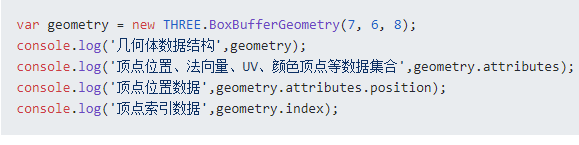
**BufferGeometry和Geometry有什么不同**

1. BufferGeometry和Geometry对象的数据结构不同，但是都可以用来描述几何体的顶点信息。





1. 缓冲类型几何体BufferGeometry相比普通几何体Geometry性能更好。

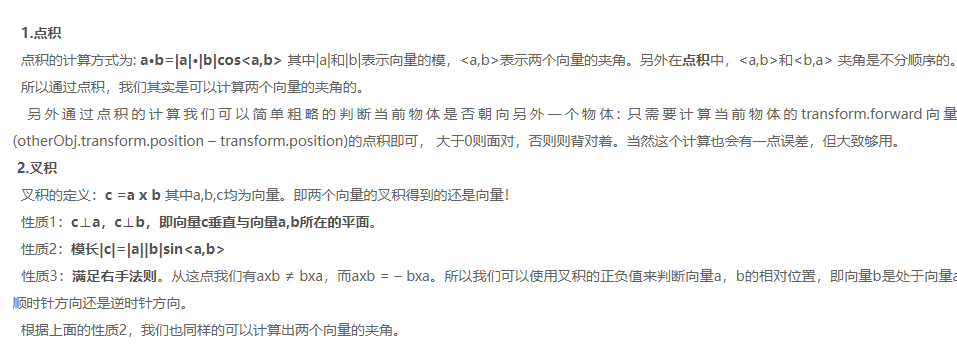
数据结构

渲染过程

Three.js渲染器在解析几何体对象的时候，如果几何体对象是普通几何体对象Geometry，Three.js的WebGL渲染器会把普通几何体对象Geometry转化为缓冲类型几何体对象BufferGeometry再提取BufferGeometry包含的顶点信息，所以直接使用BufferGeometry解析的时候相对Geometry少了一步，所以性能更高一些

Vector3.Dot表示求两个向量的点积; 计算的结果为数值

Vector3.Cross表示求两个向量的叉积。计算的结果为向量



a.normalized 和 b.normalized 表示的是两个向量的**单位向量**

因为在公式里，有向量和模的除法，得出来的结果就是单位向量，所以我们这里和后面都直接用单位向量来计算，省去不少麻烦

属性属于顶点，三角形被创建为三个顶点的合集，只储存索引信息



这种情况中间两个顶点有不同颜色(黄色和橙色)，所以我们要把这两个顶点拆分成4个顶点。这样看起来优点浪费和麻烦但是只有一个地方放置"所有"信息（所有属性都在顶点上）的简单性使许多事情变得更简单。这就是图形硬件的工作原理（我们将在本课程中稍后了解更多信息）。所以它实际上更有效率。

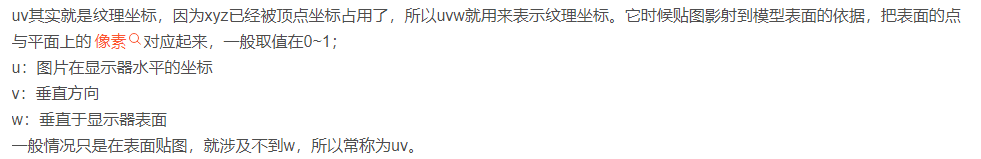
由于所有顶点都是相同的，因此我们可以制作简单的内存块，将所有顶点的值存储为一维数组。例如，顶点的位置只是一长串数字（每个顶点 3 个）。我们将内存块称为"缓冲区"。我们将用于存储属性列表（每个顶点）的内存块称为 AttributeBuffer

它们只是简单的值数组 - 以稍微特殊的方式存储，以允许它们有效地传输到图形硬件

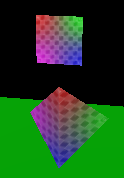
three Face3对象定义Geometry的三角面

<https://blog.csdn.net/ok2547881370/article/details/109903075>

**纹理坐标系统uv**

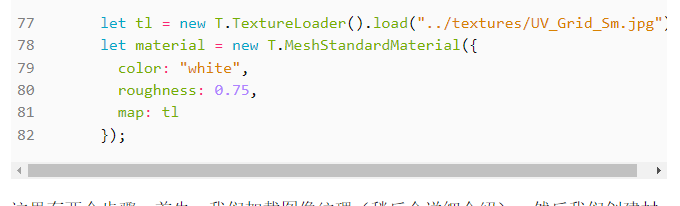






uvs里每2个数字代表一个顶点的颜色的位置，

比如uvs里第三和第四数字：1，0代表vertices里0，0，0这个点的颜色是纹理里坐标1，0的，也就是蓝色。0，1在纹理图代表左上角，也就是红色，1，1代表绿色



一般把颜色设置为白色这样不会干扰到纹理的颜色