http://webglfundamentals.org/webgl/lessons/webgl-fundamentals.html

WebGL是一个光栅引擎。

WebGL的程序是跑在GPU上面的，可以在GPU上面的运行的CODE需要两个function，一个是vertex shader，一个是fragment shader，这两个function是用GLSL来写的。这一对函数成为一个program。

GLSL是Graphics Library Shader Language的简称，基本的语法与C相似。特殊的地方在于存在vec2, vec3, vec4, mat2, mat3, mat4之类的数据类型，vec用来表示对应的向量，mat表示对应的矩阵。所有的加减法乘法都可以针对向量和矩阵来执行，一些函数也可以针对这些类型来计算，例如sin, cos。

向Shader传值的方法可以分为4种：

1. Attribute和buffer，这个是最经常使用的传值方式，buffers是我们上传到GPU的二进制数组，一般存储坐标位置、纹理坐标、顶点颜色等，当然存储什么都可以。Attribute用来确定怎么从buffer中取数据，例如从哪个buffer中获取数据，获取的数据类型，数据存取的偏移量，每次取几个字节等  
   buffer中的数据是线性访问的，每次会取出下一个值放到attribute里面。
2. Uniforms是shader中的全局变量，不随每个顶点的处理而改变
3. Varings可以从vertex shader中传值给fragment shader。根据画的东西的不同，在执行过程中，会在vertex shader中插值给fragment shader
4. Texture是可以在shader里面随机访问的数组，一般用来存放图像数据。

Vertex shader提供clip space坐标，所谓clip space坐标，就是把整个画布的坐标放到[-1, 1] [-1, 1]中间的坐标。

Fragment shader提供的颜色是rgba的颜色，是[0,1]区间的颜色。

假设我们来画一个三角形，每当我们传给GPU三个点，GPU会生成三角形里面的每一个点，GPU会向Fragment shader询问每个点的颜色。虽然我们只传给vertex shader 3个点，GPU会根据attribute中的值来生成三角形内的所有的点。