Three.js的核心分为四个核心点（和所有的3d模型建造工具类似）

1、渲染器（Renderer）

2、场景（Scene）

3、照相机（Camera）

4、你在场景中创建的物体

<https://github.com/mrdoob/three.js/tree/master/build> 该网址为three.js在github上的开源地址，从上面可以找到three.js 。引入它，开始我们的three.js的旅途

1. **Renderer**

1.1Render的新建

var renderer = new THREE.WebGLRenderer({

canvas: document.getElementById('mainCanvas')

});

//通过THREE 新建webgl的一个渲染器将渲染器与页面上指定的一个h5canvas画布联系起来

var renderer = new THREE.WebGLRenderer();

renderer.setSize(400, 300);

document.getElementsByTagName('body')[0].appendChild(renderer.domElement);

//或者通过新建一个渲染器，设置渲染器大小，动态的通过js插入到页面dom流当中

1.2设置canvas窗口的颜色。

renderer.setClearColor(0x000000);

//设置成了黑色 （ox颜色编码，暂时没有试过css的#开头编码）

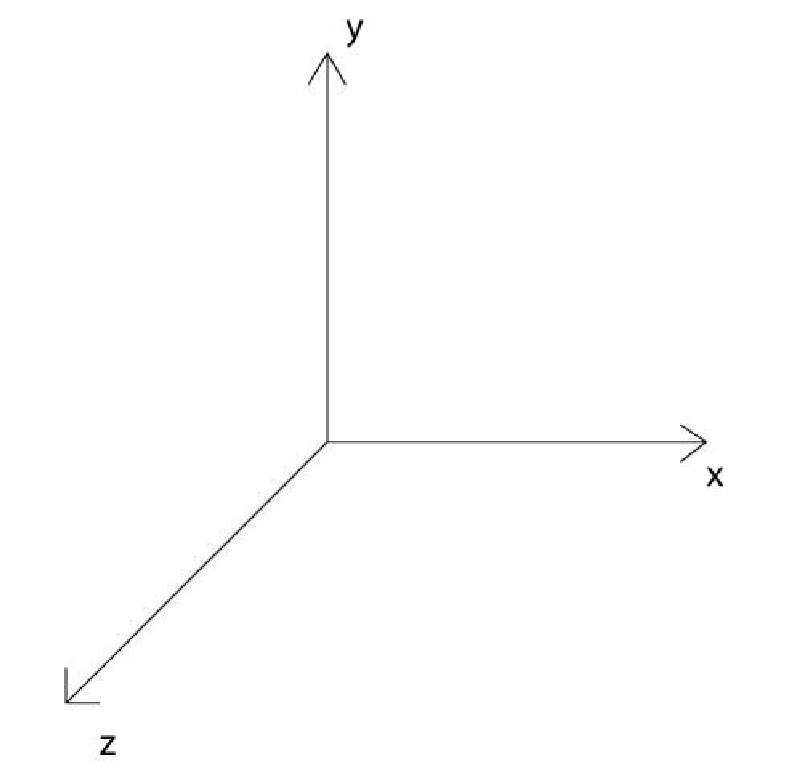
1. **Scene场景**

var scene = new THREE.Scene();

//场景的实例化，此处新建了一个场景。一般在程序最开始的时候实例化。 将我门需要出现在场景中的东西，添加进场景就好了。

1. **Camera照相机**

在webgl的3d模型中坐标系是这个样子



在这我们定义一个透视投影的照相机。它就是我们现在的定义的观测位置。

var camera = new THREE.PerspectiveCamera(45, 4 / 3, 1, 1000);

camera.position.set(0, 0, 5);

camera.lookAt(new THREE.Vector3(0, 0, 0));

scene.add(camera);

此处为透视（张角，长宽比，景深 ，深度）

定义可视范围相对与相机位置（左，右，上，下，深度，距离）正交

设置区域的 朝向

照相机定义好后需要加入场景当中哦。

1. **我们新建的物体**

var cube = new THREE.Mesh(new THREE.CubeGeometry(1, 2, 3),

new THREE.MeshBasicMaterial({

color: 0xff0000

wireframe: true

})

);

scene.add(cube)

cub一个我们新建的three.js标准的3d模型 。

它通过Mesh新建， 包含了 CubeGeometry 新建的轮廓和MeshBasicMaterial新建的材质包

Wireframe 非实心材质

1. **通过render 渲染器 ，将场景渲染出来**

renderer.render(scene, camera);

将我们的场景，和相机放进去。