首先使用我们先用three.js创建一个立方体的hello world类型的案例。

案例查看地址：<http://www.wjceo.com/blog/threejs/2018-02-09/4.html>

1. <!doctype html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
6. <meta http-equiv="X-UA-Compatible" content="ie=edge">
7. <title>Document</title>
8. <script src="build/three.js"></script>
9. <style>
10. body{margin:0;}
11. canvas{width: 100%; height:100%; display: block;}
12. </style>
13. </head>
14. <body>
15. <script>
16. *//创建场景*
17. var scene = new THREE.Scene();
18. *//设置相机（视野，显示口的宽高比，近裁剪面，远裁剪面）*
19. var camera = new THREE.PerspectiveCamera( 75, window.innerWidth/window.innerHeight, 0.1, 1000 );
20. *//渲染器*
21. var renderer = new THREE.WebGLRenderer();
22. *//设置渲染器的高度和宽度，如果加上第三个值 false，则按场景大小显示，等比例缩放*
23. renderer.setSize( window.innerWidth, window.innerHeight,false);
24. *//将渲染器添加到html当中*
25. document.body.appendChild( renderer.domElement );
27. *//盒子模型（BoxGeometry），这是一个包含立方体所有顶点和填充面的对象。*
28. var geometry = new THREE.BoxGeometry( 1, 2, 1 );
29. *//使用网孔基础材料（MeshBasicMaterial）进行着色器，这里只绘制了一个绿色*
30. var material = new THREE.MeshBasicMaterial( { color: 0x00ffff } );
31. *//使用网孔(Mesh)来承载几何模型*
32. var cube = new THREE.Mesh( geometry, material );
33. *//将模型添加到场景当中*
34. scene.add( cube );
35. *//将相机沿z轴偏移5*
36. camera.position.z = 5;
38. *//设置一个动画函数*
39. var animate = function () {
40. *//一秒钟调用60次，也就是以每秒60帧的频率来绘制场景。*
41. requestAnimationFrame( animate );
43. *//console.log(cube.rotation);*
44. *//每次调用模型的沿xy轴旋转0.01*
45. cube.rotation.x += 0.01;
46. cube.rotation.y += 0.01;
47. *//使用渲染器把场景和相机都渲染出来*
48. renderer.render(scene, camera);
49. };
51. animate();
52. </script>
53. </body>
54. </html>

上面的代码案例解析：

（1）首先引入了three.js的库文件，就和引入jq一样。

（2）创建场景（17行）

（3）创建相机，并设置视野，显示的宽高比，近裁剪面，远裁剪面（19行）

（4）创建渲染器，并设置属性，放置到dom中（21-25行）

（5）创建一个立方体模型，并放入到场景（28-34）

（6）设置相机的位置（36行）

（7）设置一个动画函数，并使用渲染器把场景和相机渲染出来，每秒60帧，显示出来，就变成了动画。