

工具

- 基于WebGL的js库three.js
- 贴图来源: https://github.com/mrdoob/three.js/blob/dev/examples/webgl_geometry_cube.html

步骤

- 搭建场景，初始化渲染器
- 创建与设置相机
相机选择透视相机，这样显示的物体符合透视原理，更有真实性
- 创建物体
创建一个正方体，需要指定其geometry与material，并设置其大小，初始位置和初始旋转角
- 给物体增加纹理
创建纹理载入器，纹理载入为异步过程，指定纹理载入器的回调函数，当纹理载入完成后，再完成物体的创建以及添加到场景中
- 添加光照
three.js提供了多种光源供选择，这里使用点光源，合理选择光源的位置和强度，让正方体的各个面显示出明暗区别
- 渲染
通过不断的将render函数加入浏览器的渲染队列中，使浏览器持续显示最新的画面。
在render函数中，可以实现动态改变物体的属性（位置，大小，颜色，旋转角等），以及自动伸缩
- 自动伸缩
保证窗口长宽比，大小发生变化时，物体的比例不发生变化，实现代码如下，在render函数里动态更新透视相机的长宽比

```
//动态伸缩
canvas = renderer.domElement;
camera.aspect = canvas.clientWidth / canvas.clientHeight;
camera.updateProjectionMatrix();
```

- 增加用户控制转动功能
为了方便的使用鼠标控制视角，导入 OrbitControls 模块，配置如下：

```
//user controlled camera
const controller = new OrbitControls(camera, canvas);
controller.target.set(0, 5, 0);
controller.update();
```

结果展示

