# 创建一个基本的threejs程序一般需要如下步骤

## 0.安装three库并且导入

import \* as Three from 'three'

## 1.创建场景

**let secen = new Three.Scene()**

## 2.创建相机，注意需要使用透视相机，搞错了没有任何效果

**let camera = new Three.PerspectiveCamera(45,window.innerWidth/window.innerHeight,1,1000)**

## 3.设置相机的z位置，距离越小物体显示的越大

**camera.position.z = 6**

## 4.创建渲染器，需要画布作为参数

let canvas = document.getElementById('mycanvas')

let renderer = new Three.WebGLRenderer({

  canvas,

  antialias:true

})

## 5.把渲染器添加到网页的body中，这一点需要记住，否则没有效果

document.body.appendChild(renderer.domElement)

## 6.设置渲染器的渲染区域大小，不设置的话，显示的区域非常小

renderer.setSize(window.innerWidth,window.innerHeight)

## 7.创建物体，如立方体，需要先创建几何体，任何创建材质，再以它们作为参数创建立方体

let cubeGeo = new Three.BoxGeometry(1,1,1)

let cubMatr = new Three.MeshBasicMaterial({color:0xffff00})

let cube = new Three.Mesh(cubeGeo,cubMatr)

## 8.设置物体位置

cube.position.x = 0

cube.position.y = 0

cube.position.z = 0

## 9.添加物体到场景，这一步非常重要，没有这一步，就看不见物体

scene.add(cube)

## 10.渲染场景

render.render(scene,camera)

## 11.此时物体是静态的，我们需要添加动画效果，

### (1)定义动画效果函数

function animate(){

  window.requestAnimationFrame(animate)

  cube.rotation.x += 0.01

  cube.rotation.y += 0.01

  renderer.render(scene,camera)

}

### (2)调用动画效果函数

animate()

## 此时，一个基本threejs程序就完成了，你可以在浏览器窗口里面看到一个旋转的黄色立方体

## 12.但是此时窗口不是响应式的，我们需要给window添加一个resize监听事件

window.addEventListener('resize',()=>{

  camera.aspect = window.innerWidth/window.innerHeight

  renderer.setSize(window.innerWidth,window.innerHeight)

  camera.updateProjectionMatrix()

})

## 扩展，这个程序我们无法调试任何效果，它不受我们控制，如果需要调整效果，需要使用three自带的lil-gui,它不是three的核心功能，需要从examples里面引入

|  |
| --- |
|  |

## 13.创建gui对象

let gui = new GUI()

## 14.把cube对应的属性添加到gui对象中

gui.add(cube.material,'wireframe')

gui.addColor(colorSetting,'color').onChange((value)=>cube.material.color.set(value))

### 效果：

|  |
| --- |
|  |
|  |
|  |

# 这一节只要学习最基本的three代码。必须记住编写流程和一些需要注意的地方，如物体添加到场景，渲染器添加到body，需要使用透视相机等等