

## 🔹 4. 第一个3D案例—创建3D场景

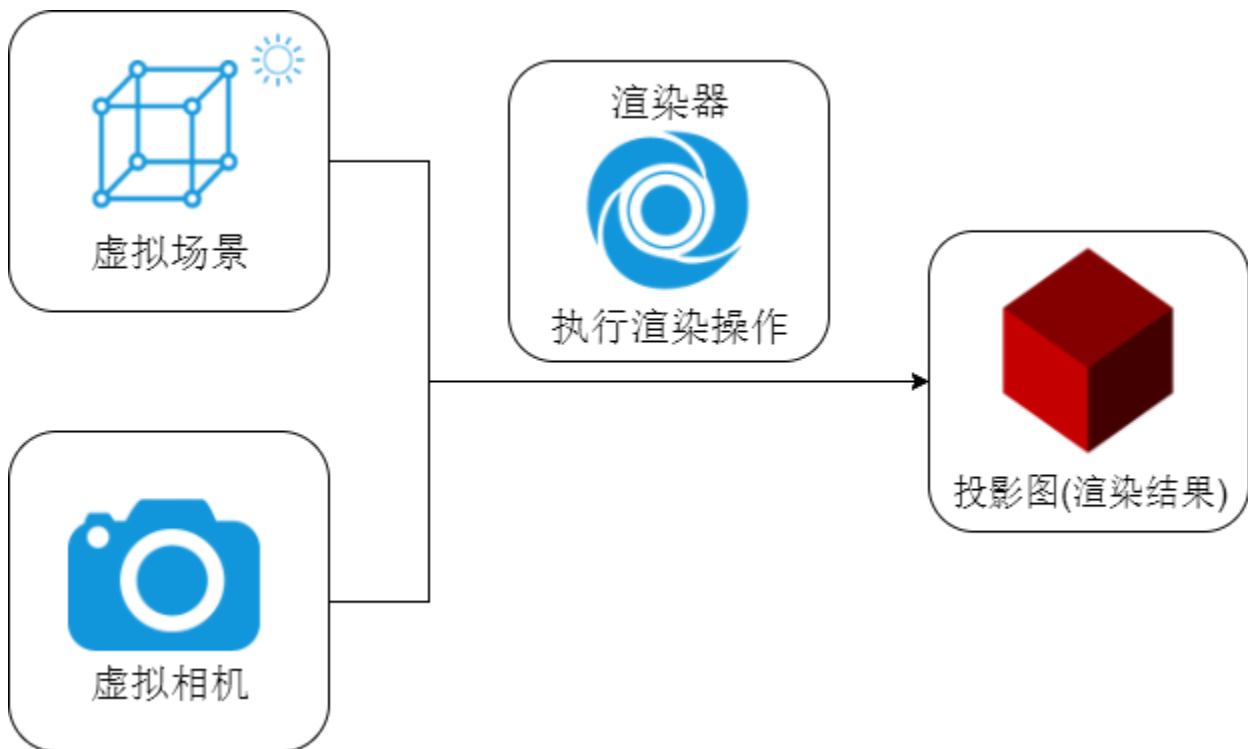
[网易云付费系统课程](#)🔗

[b站免费入门视频](#)🔗

入门Three.js的第一步，就是认识**场景Scene**、**相机Camera**、**渲染器Renderer**三个基本概念，接下来，咱们通过三小节课，大家演示“第一个3D案例”完成实现过程。

学习建议：只要你能把**第一个3D案例**搞明白，后面学习就会非常顺利了。

本节课先完成第一个3D案例的一部分，也就是3D场景Scene的创建。



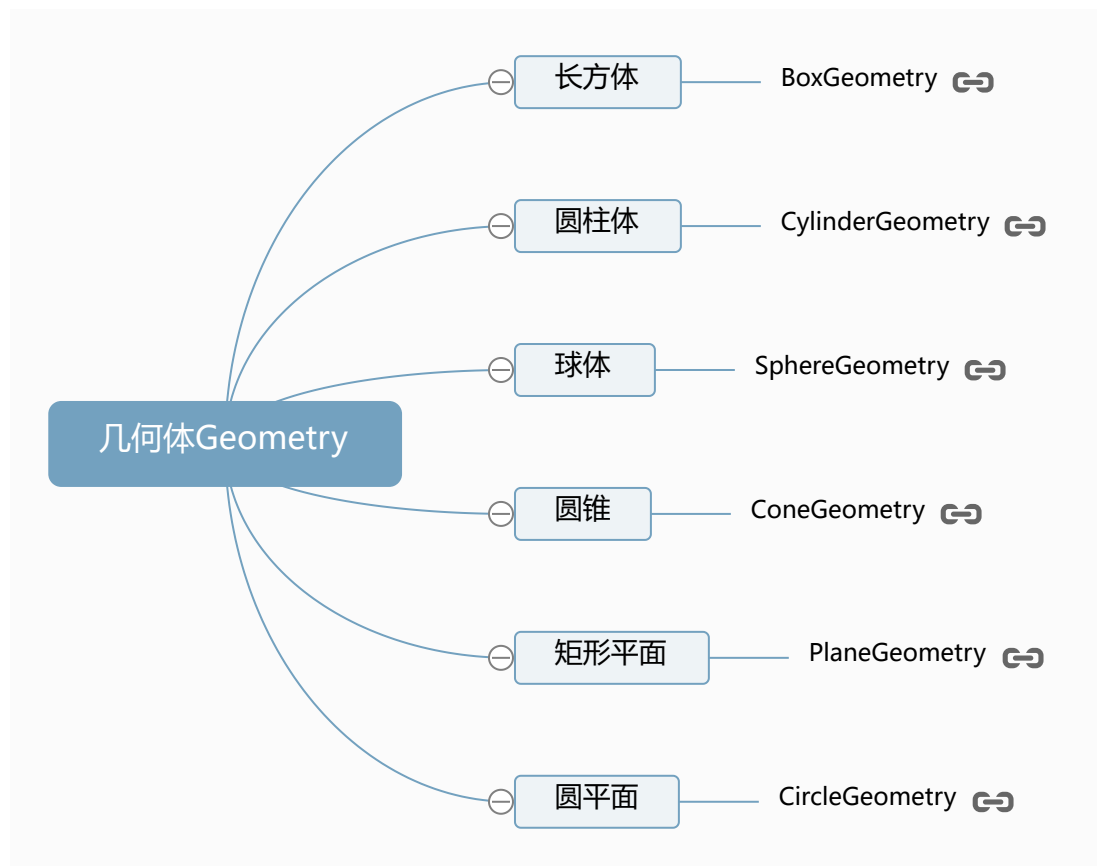
### 三维场景 Scene

你可以把三维场景**Scene**🔗 对象理解为虚拟的3D场景，用来表示模拟生活中的真实三维场景，或者说三维世界。

```
// 创建3D场景对象Scene
const scene = new THREE.Scene();
```

## 物体形状：几何体 Geometry

Three.js提供了各种各样的几何体API，用来表示三维物体的几何形状。



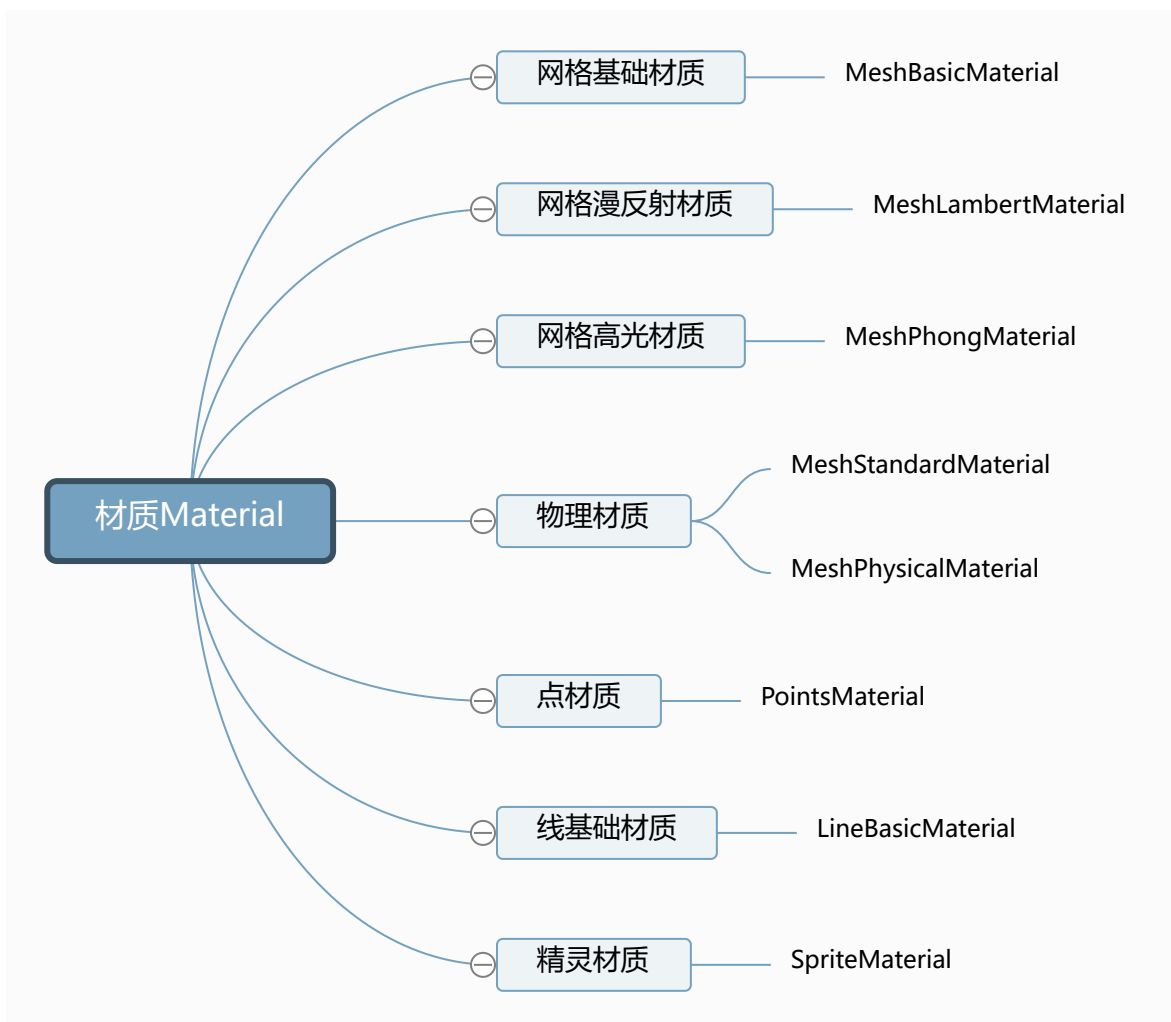
文档搜索关键词 [geometry](#) 你可以看到threejs提供各种几何体相关API，具体使用方法，也可以参考文档。

```
//创建一个长方体几何对象Geometry  
const geometry = new THREE.BoxGeometry(100, 100, 100);
```

## 物体外观：材质 Material

如果你想定义物体的外观效果，比如颜色，就需要通过材质 [Material](#) 相关的API实现。

threejs不同材质渲染效果不同，下面就以threejs最简单的**网格基础材质**[MeshBasicMaterial](#)为例给大家实现一个红色材质效果。



```
//创建一个材质对象Material
const material = new THREE.MeshBasicMaterial({
  color: 0xff0000, //0xff0000设置材质颜色为红色
});
```

## 物体：网格模型 Mesh

实际生活中有各种各样的物体，在threejs中可以通过**网格模型Mesh** 表示一个虚拟的物体，比如一个箱子、一个鼠标。

```
// 两个参数分别为几何体geometry、材质material
const mesh = new THREE.Mesh(geometry, material); //网格模型对象Mesh
```

## 模型位置 .position

实际生活中，一个物体往往是有位置的，对于threejs而言也是一样的，你可以通过位置属性 `.position` 定义网格模型 Mesh 在三维场景 Scene 中的位置。

```
const mesh = new THREE.Mesh(geometry, material); //网格模型对象Mesh
//设置网格模型在三维空间中的位置坐标，默认是坐标原点
mesh.position.set(0,10,0);
```

## .add() 方法

在threejs中你创建了一个表示物体的虚拟对象Mesh，需要通过 `.add()` 方法，把网格模型 `mesh` 添加到三维场景 `scene` 中。

```
scene.add(mesh);
```

## 后续内容

写到这里，我知道你已经迫不及待想执行代码看看渲染效果了，那么你需要看看后面两节课关于**相机Camera**和**渲染器Renderer**的介绍

[本站内容转载和引用声明](#) 

---

← [3. 开发和学习环境，引入threejs](#)

[5. 第一个3D案例—透视投影相机](#) →