

🔗 2. 四元数Quaternion

接着上节课欧拉角 `Euler` 的讲解，本节课给大家介绍threejs一个新的类,四元数 `Quaternion`。

四元数 `Quaternion` 和欧拉角 `Euler` 一样，可以用来计算或表示物体在3D空间中的旋转姿态角度。

Three.js对四元数🔗的数学细节和算法进行了封装，提供了一个四元数相关的类，平时写一些姿态角度的代码，可以使用 `Quaternion` 辅助。本节课，咱们就结合具体的threejs代码科普这个抽象的四元数概念，有了具体代码辅助，这样更容易使用四元数表示物体的姿态角度。

实例化 `Quaternion`

实例化类 `Quaternion`，你可以一个四元数对象 `Quaternion`。

```
const quaternion = new THREE.Quaternion();
```

四元数方法 `.setFromAxisAngle()`

`.setFromAxisAngle()` 是四元数的一个方法，可以用来辅助生成表示特定旋转的四元数。

`.setFromAxisAngle(axis, angle)` 生成的四元数表示绕axis旋转，旋转角度是angle。

`.setFromAxisAngle()` 可以生成一个四元数，绕任意轴，旋转任意角度，并不局限于x、y、z轴。

```
const quaternion = new THREE.Quaternion();  
// 旋转轴new THREE.Vector3(0,0,1)  
// 旋转角度Math.PI/2  
// 绕z轴旋转90度  
quaternion.setFromAxisAngle(new THREE.Vector3(0,0,1),Math.PI/2);
```

接下来用这个生成的四元数进行旋转计算。

需要旋转的A点坐标

```
// A表示3D空间一个点的位置坐标
const A = new THREE.Vector3(30, 0, 0);
```

为了方便观察，可以把旋转A点的位置用一个小球Mesh可视化表示出来

```
// 黄色小球可视化坐标点A
const Amesh = createSphereMesh(0xffff00, 2);
Amesh.position.copy(A);
group.add(Amesh);
// 创建小球mesh
function createSphereMesh(color, R) {
    const geometry = new THREE.SphereGeometry(R);
    const material = new THREE.MeshLambertMaterial({
        color: color,
    });
    const mesh = new THREE.Mesh(geometry, material);
    return mesh;
}
```

四元数旋转A点坐标

threejs三维向量 `Vector3` 具有一个方法 `.applyQuaternion(quaternion)`，该方法的功能就是通过参数 `quaternion` 对`Vector3`进行旋转，比如`Vector3`表示A点的xyz坐标，执行 `A.applyQuaternion(quaternion)`，相当于通过 `quaternion` 表示的四元数旋转A。

```
const quaternion = new THREE.Quaternion();
// 绕z轴旋转90度
quaternion.setFromAxisAngle(new THREE.Vector3(0, 0, 1), Math.PI/2);
// 通过四元数旋转A点：把A点绕z轴旋转90度生成一个新的坐标点B
const B = A.clone().applyQuaternion(quaternion);
console.log('B', B); // 查看旋转后坐标
```

你可以创建一个小球可视化查看B点位置

```
// 红色小球可视化坐标点B
const Bmesh = createSphereMesh(0xff0000,2);
Bmesh.position.copy(B);
group.add(Bmesh);
```

← 1. 欧拉角Euler

3. 四元数表示物体姿态→