

11. 动画渲染循环

视频讲解 [🔗](#)

threejs可以借助HTML5的API请求动画帧 `window.requestAnimationFrame` 实现动画渲染。

请求动画帧 `window.requestAnimationFrame`

```
// requestAnimationFrame实现周期性循环执行
// requestAnimationFrame默认每秒钟执行60次，但不一定能做到，要看代码的性能
let i = 0;
function render() {
  i+=1;
  console.log('执行次数'+i);
  requestAnimationFrame(render);//请求再次执行函数render
}
render();
```

js

备注说明：对于部分高刷新率的电脑硬件设备，`.requestAnimationFrame` 每秒钟默认调用函数执行次数也是有可能超过60次的，比如你的电脑显卡、显示器等硬件能够支持144hz刷新频率，`.requestAnimationFrame` 的每秒执行上限，也可以接近144帧率。

threejs旋转动画

动画说白了就是一张张照片，连起来依次展示，这样就形成一个动画效果，只要帧率高，人的眼睛就感觉不到卡顿，是连续的视频效果。

```
const renderer = new THREE.WebGLRenderer();
renderer.setSize(width, height);
// renderer.render(scene, camera); //执行渲染操作
document.body.appendChild(renderer.domElement);

// 渲染函数
function render() {
  renderer.render(scene, camera); //执行渲染操作
```

js

```
mesh.rotateY(0.01); //每次绕y轴旋转0.01弧度
requestAnimationFrame(render); //请求再次执行渲染函数render，渲染下一帧
}
render();
```

计算两帧渲染时间间隔和帧率

```
// 渲染循环
const clock = new THREE.Clock();
function render() {
  const spt = clock.getDelta()*1000; //毫秒
  console.log('两帧渲染时间间隔(毫秒)', spt);
  console.log('帧率FPS', 1000/spt);
  renderer.render(scene, camera); //执行渲染操作
  mesh.rotateY(0.01); //每次绕y轴旋转0.01弧度
  requestAnimationFrame(render); //请求再次执行渲染函数render，渲染下一帧
}
render();
```

js

渲染循环和相机控件OrbitControls

设置了渲染循环,相机控件OrbitControls就不用再通过事件 `change` 执行

```
renderer.render(scene, camera); , 毕竟渲染循环一直在执行
renderer.render(scene, camera); 。
```

← 10. 平行光与环境光

12. Canvas画布布局 and 全屏 →