

## 🟡 1. three.js Canvas画布布局

学习本节课之前，可以先回顾下第一章节入门部分的6和12两小节关于threejs Canvas画布布局的讲解。

- 网页上局部特定尺寸: [1.6 第一个3D案例—渲染器](#)🔗
- 全屏，随窗口变化:[1.12 Canvas画布布局 and 全屏](#)🔗
- 局部，尺寸随窗口变化: [在线案例](#)🔗

### Canvas元素

Canvas是HTML的元素之一，Canvas元素的CSS布局规律和div、img等其它HTML元素相似，webgl就是依赖于canvas实现，threejs是基于webgl封装的，自然也要依赖canvas，具体说就是把三场场景渲染到canvas画布上。

```
<!-- canvas元素默认是行内块元素 -->                                     html
<canvas style="background-color: #ff0000;" width="300" height="180"></canvas>
```

### threejs渲染结果CSS布局

浏览器控制台通过**元素**选项查看threejs输出的cnavas画布，你可以看到threejs默认把canvas设置为块元素 `display: block;` 。

既然threejs渲染输出的Canvas画布本质上就是一个HTML元素，那么你布局的时候，直接使用你的前端CSS知识即可。

### 案例源码“CSS布局案例.html”

案例源码“CSS布局案例.html”提供了一个CSS的布局案例，你可以把threejs画布插入右下角的div元素中，并完全填充。

案例源码“CSS布局案例.html”的布局效果，你也可以用CSS其它方式实现，不一定和课程中一样，CSS布局是前端知识，这里不展开讲解，咱们把重点放在threejs上面。

html

```
<head>
  <style>
    body {
      overflow: hidden;
      margin: 0px;
    }
  </style>
</head>
<body style="background: #555555;">
  <div style="height: 55px;background-color: #444444;">上面布局</div>
  <!-- 下 -->
  <div>
    <div id="left" style="position: absolute;top: 60px;background-color: #44
    <div id="webgl" style="position: absolute;top: 60px;left: 200px;backgrou
  </div>
  <script>
    const width = window.innerWidth - 200;
    const height = window.innerHeight - 60;
    document.getElementById('left').style.height = height + 'px';
    document.getElementById('webgl').style.width = width + 'px';
    document.getElementById('webgl').style.height = height + 'px';
  </script>
</body>
```

## threejs Canvas画布尺寸设置

设置canvas画布的尺寸，注意在浏览器窗口文档区域整体宽高基础上，减掉其他顶部和左侧div元素和布局间隙的尺寸。

- canvas画布的**宽度**: `window.innerWidth` 减掉**左侧**div元素宽度 195px 和布局间距 5px
- canvas画布的**高度**: `window.innerHeight` 减掉**顶部**div元素高度 55px 和布局间距 5px

js

```
// 200表示左侧div元素宽度195px+间距5px
const width = window.innerWidth - 200; //canvas画布高度
//60表示顶部div元素高度55px+间距5px
const height = window.innerHeight - 60; //canvas画布宽度
...
const camera = new THREE.PerspectiveCamera(30, width / height, 1, 3000);
```

```
renderer.setSize(width, height);
```

js

```
// 200表示左侧div元素宽度195+间距5px
const width = window.innerWidth - 200; //canvas画布高度
//60表示顶部div元素高度55px+间距5px
const height = window.innerHeight - 60; //canvas画布宽度
...
const camera = new THREE.PerspectiveCamera(30, width / height, 1, 3000);
renderer.setSize(width, height);
```

## Canvas随着窗口变化

js

```
// 画布跟随窗口变化
window.onresize = function () {
    const width = window.innerWidth - 200; //canvas画布高度
    const height = window.innerHeight - 60; //canvas画布宽度
    renderer.setSize(width, height);
    camera.aspect = width / height;
    camera.updateProjectionMatrix();
};
```

---

← [7. 三维软件导出PBR材质属性](#)

[2. UI交互界面与Canvas画布叠加](#) →