

## 🎯 3. 开发和学习环境，引入threejs

视频讲解 [🔗](#)

本节课主要给大家说下threejs怎么引入的问题。

1. 开发环境：项目开发引入threejs，比如vue或react脚手架引入threejs。
2. 学习环境：入门threejs阶段，.html文件中直接引入threejs

### 项目的开发环境引入threejs

比如你采用的是Vue + threejs或React + threejs技术栈，这很简单，threejs就是一个js库，直接通过npm命令行安装就行。

npm安装特定版本three.js(注意使用哪个版本，查文档就查对应版本)

```
// 比如安装148版本  
npm install three@0.148.0 --save
```

### npm安装后，如何引入three.js

执行 `import * as THREE from 'three';` ,ES6语法引入three.js核心。

```
// 引入three.js  
import * as THREE from 'three';
```

如果你没有前端ES6基础，可以参考我分享的前端公开课[ES6入门教程](#) [🔗](#)。

### npm安装后，如何引入three.js其他扩展库

除了three.js核心库以外，在threejs文件包中examples/jsm目录下，你还可以看到各种不同功能的扩展库。

一般来说，你项目用到那个扩展库，就引入那个，用不到就不需要引入。

```
// 引入扩展库OrbitControls.js
import { OrbitControls } from 'three/addons/controls/OrbitControls.js';
// 引入扩展库GLTFLoader.js
import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';

// 扩展库引入—旧版本，比如122，新版本路径addons替换了examples/jsm
import { OrbitControls } from 'three/examples/jsm/controls/OrbitControls.js';
```

## 学习环境：.html文件中直接引入threejs

如果不是正式开发Web3D项目，只是学习threejs功能，完全没必要用webpack或vite搭建一个开发环境。

学习使用的环境，只要创建一个.html文件，编写threejs代码，最后通过**本地静态服务**打开.html文件就行。

## script标签方式引入three.js

你可以像平时开发web前端项目一样，通过script标签把three.js当做一个js库引入你的项目。

three.js库可以在threejs官方文件包下面的build目录获取到。

```
<script src="./build/three.js"></script>
```

html

```
//随便输入一个API，测试下是否已经正常引入three.js
console.log(THREE.Scene);
```

js

## ES6 import方式引入

给script标签设置 `type="module"` ,也可以在.html文件中使用 `import` 方式引入three.js。

```
<script type="module">
// 现在浏览器支持ES6语法，自然包括import方式引入js文件
import * as THREE from './build/three.module.js';
</script>
```

html

## type="importmap"配置路径

学习环境中，.html文件引入three.js，最好的方式就是参考threejs官方案例，通过配置 `<script type="importmap">` ,实现学习环境.html文件和vue或react脚手架开发环境一样的写法。这样你实际项目的开发环境复制课程源码，不用改变threejs引入代码。

下面配置的 `type="importmap"` 代码具体写法不用掌握记忆，复制粘贴后，能修改目录就行，你可以去电子书课件或者课件源码中复制。

```
<!-- 具体路径配置，你根据自己文件目录设置，我的是课件中源码形式 -->
<script type="importmap">
  {
    "imports": {
      "three": "../..../three.js/build/three.module.js"
    }
  }
</script>
```

html

```
<!-- 配置type="importmap",.html文件也能和项目开发环境一样方式引入threejs -->
<script type="module">
  import * as THREE from 'three';
  // 浏览器控制台测试，是否引入成功
  console.log(THREE.Scene);
</script>
```

html

## type="importmap"配置——扩展库引入

通过配置 `<script type="importmap">` ，让学习环境.html文件，也能和vue或react开发环境中一样方式引入threejs扩展库。

配置 `addons/` 等价于 `examples/jsm/` 。

```
<script type="importmap">
  {
    "imports": {
      "three": "../three.js/build/three.module.js",
      "three/addons/": "../three.js/examples/jsm/"
    }
  }
</script>
```

html

```
}  
</script>
```

html

```
<script type="module">  
  // three/addons/路径之后对应的是three.js官方文件包`/examples/jsm/`中的js库  
  // 扩展库OrbitControls.js  
  import { OrbitControls } from 'three/addons/controls/OrbitControls.js';  
  // 扩展库GLTFLoader.js  
  import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';  
  console.log(OrbitControls);  
  console.log(GLTFLoader);  
</script>
```

---

← 2. 学习环境-编辑器和静态服务器

4. 第一个3D案例—创建3D场景 →