

## 🟡 6. 模型命名(程序与美术协作)

开发一些web3d项目，比如一个小区、工厂的可视化，场景中会有多个模型对象，程序员加载三维模型的时候，通过什么方式才能获取到自己想要的某个模型节点是个问题。

### 三维软件模型命名

课程提供了一个Blender的模型例子，你可以打开查看。

其实模型节点命名可以类比前后端API接口命名，web3d前端和后端对接需要命名接口，和3D美术对接，同样需要给一些模型节点命名。

- 模型命名可以使用汉字、英文、拼音其他语言形式。
- 如果使用汉字注意，有些三维建模软件可能存在导出乱码问题。

### 浏览器控制台查看3D模型树结构

加载gltf模型,通过 `gltf.scene` 可以获取模型的数据，你可以通过浏览器控制打印 `gltf.scene`，然后和你三维建模软件中的模型目录树对比，比较两者的结构是否相同。

- 模型父对象节点可以用 `Object3D` 对象表示，也可以用组对象 `Group` 表示。
- 通过 `.children` 属性可以查看一个父对象模型的所有子对象。
- 通过 `.name` 属性可以查看模型节点的名称

```
loader.load("./简易小区.glb", function (gltf) {
  console.log('场景3D模型树结构', gltf.scene);
  model.add(gltf.scene);
})
```

### `.getObjectByName()` 根据 `.name` 获取模型节点

一般三维建模软件的目录树，都有模型的名称，three.js加载外部模型，外部模型的名称体现为three.js对象的 `.name` 属性,three.js可以通过 `.getObjectByName()` 方法，把模型节点的名

字 `.name` 作为改函数参数，快速查找某个模型对象。

```
// 返回名.name为"1号楼"对应的对象
const nameNode = gltf.scene.getObjectByName("1号楼");
nameNode.material.color.set(0xff0000); //改变1号楼Mesh材质颜色
```

## 分组管理

对于大类，可以进行分组，这样更好管理，比如高层分为一组，洋房分为一组。如果这样做的好处是，程序员可以通过分类名称，快速获取所有模型，然后进行同样的渲染操作，比如洋房批量改变颜色。

```
//获得所有'洋房'房子的父对象
const obj = gltf.scene.getObjectByName('洋房');
console.log('obj', obj); //控制台查看返回结果
console.log('obj.children', obj.children);
// obj.children的所有子对象都是Mesh，改变Mesh对应颜色
obj.children.forEach(function (mesh) {
    mesh.material.color.set(0xffff00);
})
```

---

← [5. gltf不同文件形式\(.glb\)](#)

[7. 递归遍历层级模型修改材质](#) →