

BaseObject3D

该对象继承自 [threejs](#) `Object3D` 对象。

Object3D 常用属性

name

名称

- 类型：string
- 默认值：''

uuid

唯一哈希标示

- 类型：string
- 默认值：随机哈希

visible

可见状态

- 类型：boolean
- 默认值：true

主要警告

不可以手动设置，该属性已被拦截，会随 `level` 计算自动变化，如果想手动控制可见行请使用方法 `.hide()` `.show()`。

position

空间三维坐标

- 类型: object
- 默认值: `{ x: 0, y: 0, z: 0 }`

rotation

旋转弧度

- 类型: object
- 默认值: `{ x: 0, y: 0, z: 0 }`

scale

缩放比

- 类型: object
- 默认值: `{ x: 1, y: 1, z: 1 }`

children

子对象集合

- 类型: array
- 默认值: `[]`

userData

绑定在对象的用户数据

- 类型: object
- 默认值: `{}`

soonspace 扩展属性

sid

对象在 `soonspace` 层的 `ID`

- 类型：string | number
- 默认值：无

stype

对象在 `soonspace` 层的类型

- 类型：string
- 默认值： `BaseObject3D`

level

对象可见层级范围

- 类型： `Level`
- 默认值： `{ max: null, min: null }`

handleHide

对象是否被手动操作隐藏

- 类型：boolean
- 默认值： `false`

主要警告

不可以手动设置，该属性是为了实现 `level` 在场景中的自动显示隐藏扩展，会在调用方法 `.hide()` `.show()` 时自动变化。

方法

show

控制该对象显示。

hide

控制该对象隐藏。

setMove

设置该对象的位置移动。动画过度，使用继承于 [API 补间动画](#)。

定义：

```
function setMove(  
  position: Position,  
  options?: AnimationOptions,  
  onUpdate?: (source: Position, tween: Tween<Position>) => void,  
  onStart?: (tween: Tween<Position>) => void  
): Promise<void> {  
  return Animation<Position>(  
    this.position,  
    position,  
    options,  
    onUpdate,  
    onStart  
  );  
}
```

ts

用法：

```
sbm.setMove(  
  // target  
  { x: 1000, y: 0, z: 1000 }  
  // ... args  
);
```

js

setRotate

设置该对象的角度旋转。动画过度，使用继承于 [API 补间动画](#)。

定义：

```
function setRotate(  
  rotation: Rotation,  
  options?: AnimationOptions,
```

ts

```
onUpdate?: (source: Rotation, tween: Tween<Rotation>) => void,  
onStart?: (tween: Tween<Rotation>) => void  
) : Promise<void> {  
    return Animation<Euler>(this.rotation, rotation, options, onUpdate, onStart);  
}
```

用法:

```
sbm.setRotate(  
    // target  
    // Math.PI 表示半圈  
    { x: 0, y: Math.PI, z: 0 }  
    // ... args  
);
```

js

setScale

设置该对象的尺寸缩放。动画过度，使用继承于 [API 补间动画](#)。

定义:

```
function setScale(  
    scale: Scale,  
    options?: AnimationOptions,  
    onUpdate?: (source: Scale, tween: Tween<Scale>) => void,  
    onStart?: (tween: Tween<Scale>) => void  
) : Promise<void> {  
    return Animation<Scale>(this.scale, scale, options, onUpdate, onStart);  
}
```

ts

用法:

```
sbm.setScale(  
    // target  
    { x: 10, y: 10, z: 10 }  
    // ... args  
);
```

js

getSpaceAttribute

获取该对象的空间属性

定义:

```
function getSpaceAttribute(  
  isFromWorld?: boolean  
): {  
  position: Vector3;  
  rotation: Euler;  
  scale: Vector3;  
};
```

ts

用法:

```
const attr = sbm.getSpaceAttribute(false);
```

js

参数:

isFromWorld

- 描述: 属性值是否相对于世界, 默认为 true
- 类型: boolean
- 必填: **✗**

syncSpaceAttribute

同步空间属性

定义:

```
function syncSpaceAttribute(object: Object3D, isFromWorld?: boolean): void;
```

ts

用法:

```
sbm.getSpaceAttribute(otherSbm, true);
```

js

参数:

object

- 描述: 任何空间对象
- 类型: Object3D
- 必填: ✓

isFromWorld

- 描述: 属性值是否相对于于世界，默认为 true
- 类型: boolean
- 必填: ✗

← [场景事件](#)

[BaseMesh](#) →