

BaseMesh

该对象继承自 [threejs](#) `Mesh` 对象。

Mesh 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

geometry

网格几何结构

- 类型: [BufferGeometry](#)
- 默认值: `BufferGeometry`

material

网格材质

- 类型: [Material](#)
- 默认值: `MeshBasicMaterial`

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

☰ SoonSpace.js 2.x



• 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: `BaseMesh`

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

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)  
}
```

用法：

```
// ... args
{ x: 1000, y: 0, z: 1000 },
// ... args
)
```

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,
  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
)
```

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：

☰ SoonSpace.js 2.x



```
source: 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)  
}
```

用法:

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

js

[← BaseObject3D](#)[Group →](#)