

🔹 2. 三维向量Vector3简介

如果你已经掌握了数学中关于**向量**的运算规则，直接去看看threejs `Vector3` 的文档，很多方法和属性都可以快速理解，比如点乘 `.dot()`、叉乘 `.cross()`，这时候你可以回忆你学过的数学知识来理解threejs的API `Vector3` 中的部分方法和属性。

不过有很多同学可能学过，但是都还给老师了，还有些同学干脆是没有**向量**的概念，这时候直接看threejs文档会不太理解，不过这没关系，下面几节课会通过threejs具体的代码案例讲解，你也不用刻意记忆这些数学概念，只要跟着课程写代码就行。

向量和标量

数学中有两个概念**向量**和**标量**，简单地说**向量**就是一个有方向的量，比如在3D空间中描述一个人的速度，你需要表达速度的**大小**，同时也需要表达速度运动的**方向**，这样才能计算出3D空间中人经过一段时间，在x、y、z三个方向分别走了多远距离。至于标量，就是一个没有方向的量，比如模型的位置具体坐标 `mesh.position`。

threejs的类 `Vector3`，虽然英文字面意思是向量，除了向量，`Vector3` 也可以用来表示标量，具体看你代码赋予 `Vector3` 什么样的含义。比如 `Vector3` 作为一个标量，表示网格模型的Mesh的位置 `mesh.position`，`Vector3` 也可以用来表示一个向量，比如一个模型沿着x轴平移，那么平移的方向可以用 `new THREE.Vector3(1,0,0)` 表示。

`Vector3` 表示3D空间中位置坐标(非向量)

在基础课程中[3.1三维向量Vector3与模型位置](#)🔗 就给大家介绍过向量相关知识点，下面简单回顾下。

`Vector3` 对象具有属性 `.x`、`.y`、`.z` 三个属性，这意味着你可以用 `Vector3` 对象表示3D空间中的位置坐标x、y、z。

```
const v3 = new THREE.Vector3(30,30,0);
console.log('v3',v3);
```

js

threejs本身就会给 `mesh.position` 一个默认值 `THREE.Vector3(0,0,0)`，这就是说你可以不用给 `mesh.position` 赋值Vector3对象，你可以直接访问 `mesh.position`，获取或设置Vector3 的 `.x`、`.y`、`.z` 属性。

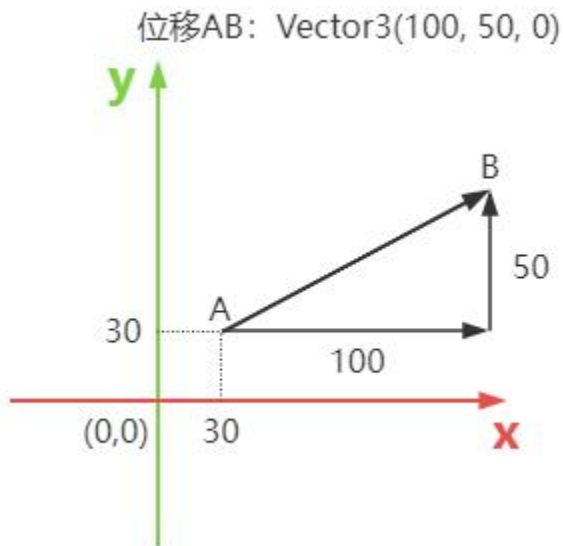
```
console.log('mesh.position',mesh.position);
```

js

```
mesh.position.y = 80;// 设置网格模型y坐标  
mesh.position.set(80,2,10);// 设置模型xyz坐标
```

js

Vector3 表示位移量(向量)



比如已知人在3D空间中的坐标A点是 `(30,30,0)`，此人运动到B点，从A到B的位移变化量可以用一个向量 `Vector3` 表示，已知AB在x轴上投影长度是100，y方向投影长度是50，这个变化可以用三维向量 `THREE.Vector3(100,50,0)` 表示,换句话u说，你也可以理解为人沿着x轴走了100，沿着y方向走了50，到达B点。

```
const A = new THREE.Vector3(30, 30, 0);// 人起点A  
// walk表示运动的位移量用向量  
const walk = new THREE.Vector3(100, 50, 0);  
const B = new THREE.Vector3();// 人运动结束点B  
// 计算结束点xyz坐标  
B.x = A.x + walk.x;  
B.y = A.y + walk.y;  
B.z = A.z + walk.z;  
console.log('B',B);
```

js

向量加法运算 `.addVectors()`

`B.addVectors(A,walk)` 的含义就是向量A和向量walk的x、y、z三个分量分别相加(`B.x = A.x + walk.x` ; 、 `B.y = A.y + walk.y` ; 、 `B.z = A.z + walk.z` ;)。

```
const A = new THREE.Vector3(30, 30, 0); // 人起点A
// walk表示运动的位移量用向量
const walk = new THREE.Vector3(100, 50, 0);
const B = new THREE.Vector3(); // 人运动结束点B
// addVectors的含义就是参数中两个向量xyz三个分量分别相加
B.addVectors(A,walk);
console.log('B',B);
```

js

向量复制方法 `.copy()`

通过向量复制方法 `.copy()` , 把A和B点的坐标赋值给两个表示网格模型对象,可视化A点和B点。

```
// 两个小球网格模型可视化A点和B点
mesh1.position.copy(A);
mesh2.position.copy(B);
```

js

向量加法运算 `.add()`

`.add()` 和 `.addVectors()` 功能一样, 只是语法细节不同。

A和walk的x、y、z属性分别相加, 相加的结果替换A原来的x、y、z。

```
A.add(walk);
```

js

`const B = A.add(walk);` 这种写法, B本质上就是A, 两个变量指向同一个对象。

```
const B = A.add(walk);
console.log('A',A);
```

js

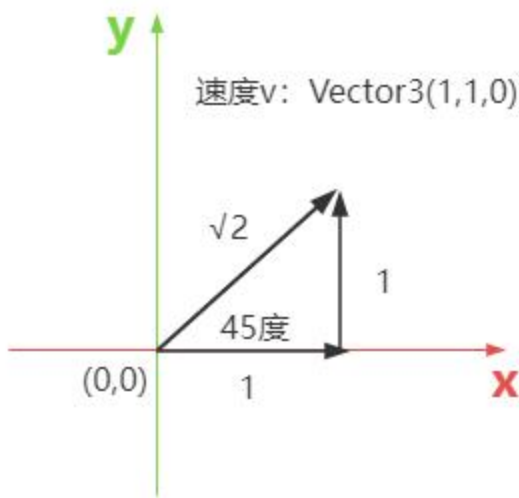
```
console.log('B',B);
```

如果不希望A被改变，且创建一个新的对象表示B点坐标，通过克隆方法 `.clone()`。

```
// A.clone()克隆一个和A一样对象，然后再加上walk，作为B
// A不执行.clone()，A和B本质上都指向同一个对象
const B = A.clone().add(walk);
```

js

Vector3 表示速度(向量)



假设一个人的运动速度大小是 $\sqrt{2}$,方向是x和y正半轴的角平分线，那么人的速度可以用向量 `THREE.Vector3(1, 1, 0)` 表示。

```
// 向量v表示人速度，大小√2米每秒，方向是x、y正半轴的角平分线
const v = new THREE.Vector3(1, 1, 0);
```

js

以速度v运动50秒，计算运动位移变化量。

```
// xyz三个方向上速度分别和时间相乘，得到三个方向上位移
const walk = new THREE.Vector3(v.x * 50, v.y * 50, v.z * 50);
```

js

假设人起点坐标A `(30, 30, 0)`，以速度v运动50秒，计算运动结束位置。

```
const v = new THREE.Vector3(1, 1, 0);
const walk = new THREE.Vector3(v.x * 50, v.y * 50, v.z * 50);
// 运动50秒结束位置B
```

js

```
const B = A.clone().add(walk);
```

向量方法 `.multiplyScalar()`

向量方法 `.multiplyScalar(50)` 表示向量x、y、z三个分量和参数分别相乘。

`v.clone().multiplyScalar(50)` 的含义和 `Vector3(v.x * 50, v.y * 50, v.z * 50)` 是一样的。

```
// `.multiplyScalar(50)`表示向量x、y、z三个分量和参数分别相乘
const walk = v.clone().multiplyScalar(50);
// 运动50秒结束位置B
const B = A.clone().add(walk);
```

js

← [1. threejs坐标系与三角函数](#)

[3. 向量大小\(Vector3长度.length\(\)\)](#) →