

🔹 12. 点到直线的距离

通过threejs叉乘方法 `.cross()` 和点乘方法 `.dot()` 计算点到直线的距离。

已知条件

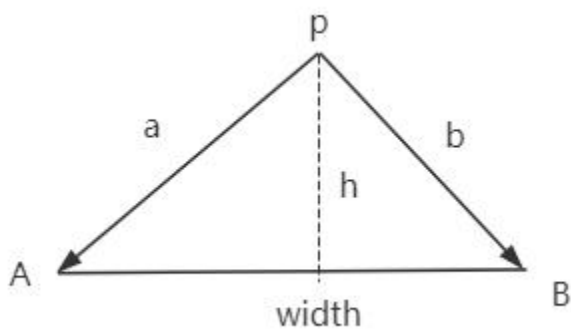
直线上的两个点的坐标，和直线外的的一点坐标。

```
// 已知条件
// 直线经过两点坐标A、B
const A = new THREE.Vector3(0, 0, 0);
const B = new THREE.Vector3(100, 0, 0);
// 直线外一点p
const p = new THREE.Vector3(50, 0, 30);
```

js

ApB构成一个三角形，计算三角形面积

向量ApB构建一个三角形，计算三角形面积。



```
// ApB构建一个三角形，其中两条边构建向量a、向量b
const a = A.clone().sub(p);
const b = B.clone().sub(p);
const c = a.clone().cross(b);
const S = 0.5*c.length(); // 叉乘结果长度一半是三角形ApB的面积
```

js

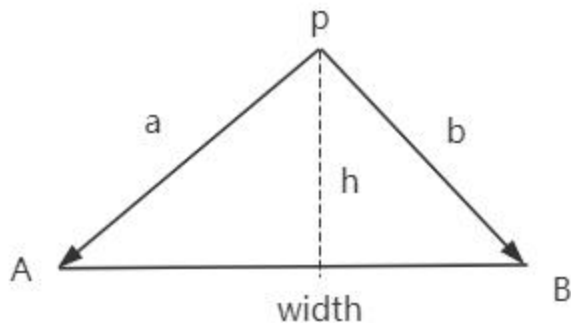
计算三角形ApB底边AB的长度

```
const AB = B.clone().sub(A);  
const width = AB.length();//AB两点距离
```

js

计算三角形高度(点到直线的距离)

点p到AB直线的距离，就是三角形ApB在AB上的高度。



```
//叉乘结果长度一半是三角形ApB的面积  
const S = 0.5*c.length();  
//AB两点距离  
const width = AB.length();  
const H = S / width * 2;//三角形高度，也就是点到直线的距离  
console.log('点到直线的距离',H);
```

js

点p到直线AB距离最终代码

```
const AB = B.clone().sub(A);  
const width = AB.length();//AB两点距离  
// ApB构建一个三角形，其中两条边构建向量a、向量b  
const a = A.clone().sub(p);  
const b = B.clone().sub(p);  
const c = a.clone().cross(b);  
const H = c.length()/width;//点到直线的距离
```

js

