

🎨 4. 椭圆、圆

接着上节课内容给大家介绍椭圆和圆曲线。

椭圆弧线 `EllipseCurve`

```
EllipseCurve( aX, aY, xRadius,yRadius, aStartAngle, aEndAngle, aClockwise ) js
```

参数	含义
aX, aY	椭圆中心坐标
xRadius	椭圆x轴半径
yRadius	椭圆y轴半径
aStartAngle	弧线开始角度，从x轴正半轴开始，默认0，弧度单位
aEndAngle	弧线结束角度，从x轴正半轴算起，默认2 x Math.PI，弧度单位
aClockwise	是否顺时针绘制，默认值为false

```
// 参数1和2表示椭圆中心坐标 参数3和4表示x和y方向半径
const arc = new THREE.EllipseCurve(0, 0, 100, 50);
```

js

椭圆曲线x和y方向半径相同，就是一个圆的效果。

```
// 参数1和2表示椭圆中心坐标 参数3和4表示x和y方向半径
const arc = new THREE.EllipseCurve(0, 0, 50, 50);
```

js

圆弧线 `ArcCurve`

圆弧线 `ArcCurve` 的父类是椭圆弧线 `EllipseCurve` ,语法和椭圆弧线 `EllipseCurve` 相似，区别是参数3和参数4不同，椭圆需要定义xRadius和yRadius两个半径，圆只需要通过参数3定义半径aRadius即可。

```
//参数: 0, 0圆弧坐标原点x, y 100: 圆弧半径 0, 2 * Math.PI: 圆弧起始角度
const arc = new THREE.ArcCurve(0, 0, 100, 0, 2 * Math.PI);
```

js

```
ArcCurve( aX, aY, aRadius, aStartAngle, aEndAngle, aClockwise )
```

js

参数	含义
aX, aY	圆心坐标
aRadius	圆弧半径
aStartAngle	弧线开始角度, 从x轴正半轴开始, 默认0, 弧度单位
aEndAngle	弧线结束角度, 从x轴正半轴算起, 默认2 x Math.PI, 弧度单位
aClockwise	是否顺时针绘制, 默认值为false

曲线精度

```
//曲线上取点, 参数表示取点细分精度
const pointsArr = arc.getPoints(50); //分段数50, 返回51个顶点
// const pointsArr = arc.getPoints(10); //取点数比较少, 圆弧线不那么光滑
```

js

弧线起始角度

参数4和5表示圆弧线的起始角度, three.js默认是一个完整的圆弧, 其实你也可以绘制一个半圆弧。

```
// 完整圆弧
const arc = new THREE.ArcCurve(0, 0, 100, 0, 2 * Math.PI);
```

js

```
// 半圆弧
const arc = new THREE.ArcCurve(0, 0, 100, 0, Math.PI);
```

js

```
// 四分之一圆弧
const arc = new THREE.ArcCurve(0, 0, 100, 0, Math.PI/2);
```

js

顺逆时针

参数6默认false，逆时针绘制圆弧

```
const arc = new THREE.ArcCurve(0, 0, 100, 0, Math.PI/2, false);
```

js

参数6设置为true，顺时针绘制圆弧

```
const arc = new THREE.ArcCurve(0, 0, 100, 0, Math.PI/2, true);
```

js

← [3. 曲线Curve简介](#)

[5. 样条曲线](#) →