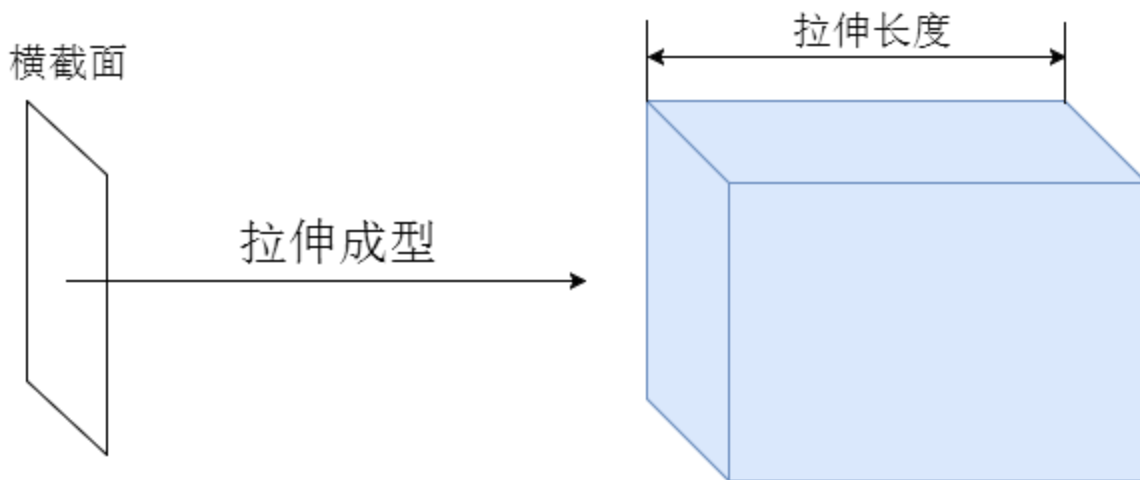


🔗 12. 拉伸ExtrudeGeometry

拉伸几何体 `ExtrudeGeometry` 和上节课讲到的轮廓填充几何体 `ShapeGeometry` 一样，都是基于一个基础的平面轮廓 `Shape` 进行变换，生成一个几何体。

查看案例源码，你可以看到 `ExtrudeGeometry` 的一个拉伸变换案例。



定义一个 `Shape` 轮廓

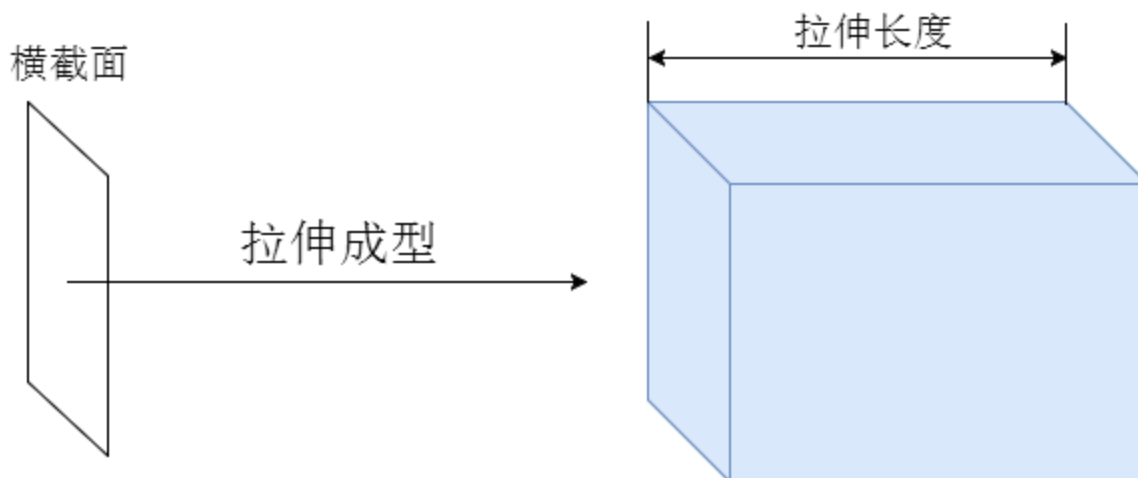
使用拉伸扫描 `ExtrudeGeometry` 和轮廓填充 `ShapeGeometry` 一样，需要首先定义一个用于拉伸或扫描的平面轮廓 `Shape`。

多边形上随意选择一个点，作为起点，也就是 `Shape` 的第一个坐标，然后按照逆时针或顺时针方向依次书写坐标。

```
// Shape表示一个平面多边形轮廓
const shape = new THREE.Shape([
  // 按照特定顺序，依次书写多边形顶点坐标
  new THREE.Vector2(-50, -50), // 多边形起点
  new THREE.Vector2(-50, 50),
  new THREE.Vector2(50, 50),
  new THREE.Vector2(50, -50),
]);
```

js

拉伸成型



```
//拉伸造型
const geometry = new THREE.ExtrudeGeometry(
  shape, //二维轮廓
  {
    depth: 20, //拉伸长度
  }
);
```

js

拉伸倒角

倒圆角

```
const geometry = new THREE.ExtrudeGeometry(
  shape, {
    depth: 20,
    bevelThickness: 5, //倒角尺寸:拉伸方向
    bevelSize: 5, //倒角尺寸:垂直拉伸方向
    bevelSegments: 20, //倒圆角: 倒角细分精度, 默认3
  }
);
```

js

倒直角

```
const geometry = new THREE.ExtrudeGeometry(
  shape, {
```

js

```
        bevelSegments: 1, //倒直角
    }
};
```

拉伸取消默认倒角

```
//拉伸造型
const geometry = new THREE.ExtrudeGeometry(
    shape, //二维轮廓
    {
        depth: 20, //拉伸长度
        bevelEnabled: false, //禁止倒角,默认true
    }
);
```

js

[← 11. 轮廓填充ShapeGeometry](#)

[13. 扫描ExtrudeGeometry→](#)