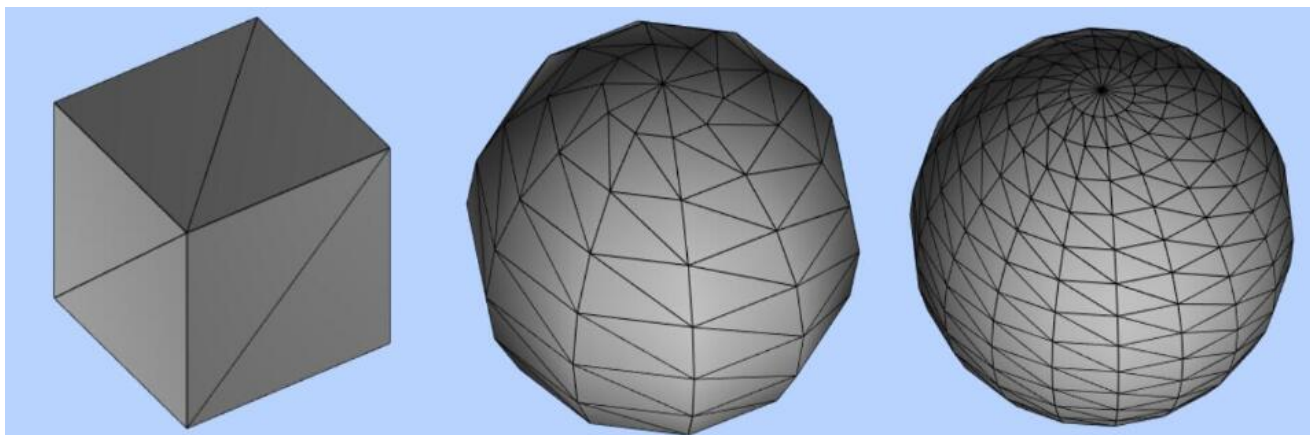
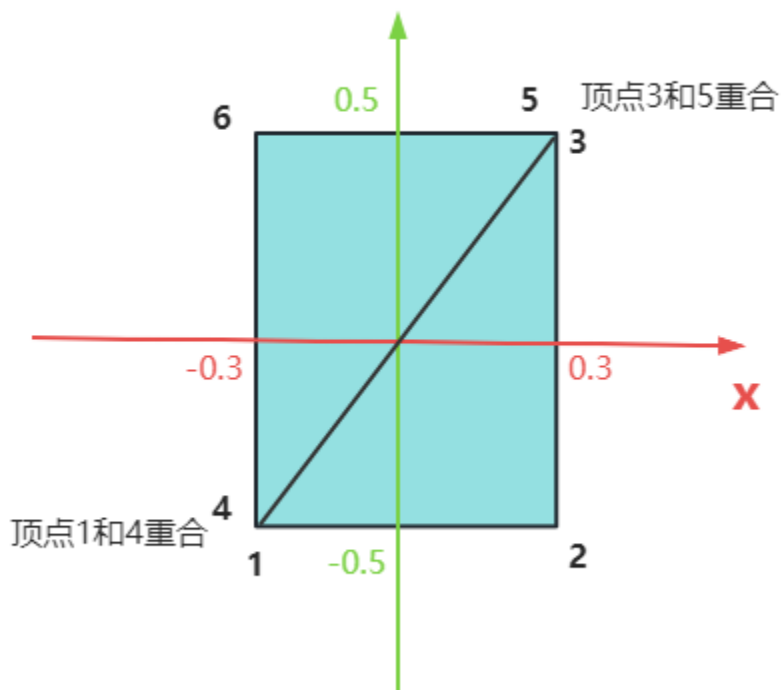


🎯 9. 三角形拼接矩形

实际上在WebGPU中，表达一个复杂的3D几何形状，一般都是通过一个一个三角形拼接出来的。



本节课就给大家讲解一个最简单的案例，用2个三角形拼接一个矩形。



绘制两个三角形

前面说过, `primitive` 属性的值设置为 `triangle-list`, 当执行draw绘制命令的时候, 会以三角形的方式绘制顶点数据, 简单说就是每3个顶点绘制一个三角形, 6个顶点就是绘制两个三角形。

```
const pipeline = device.createRenderPipeline({  
  primitive: {  
    topology: "triangle-list", //绘制三角形  
  }  
});
```

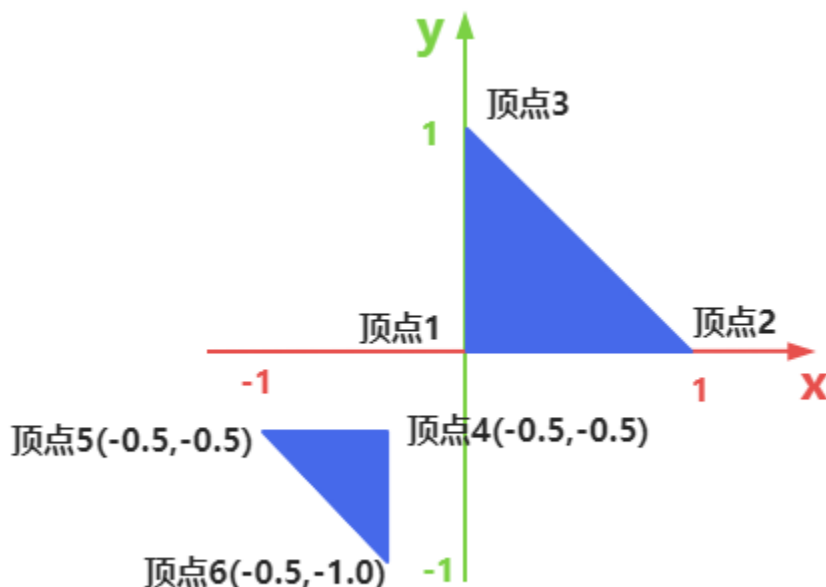
js

前面讲解过的, 绘制一个三角形, 三个顶点, draw绘制命令第一个参数是3。

```
const vertexArray = new Float32Array([  
  // 三角形三个顶点坐标的x、y、z值  
  0.0, 0.0, 0.0, //顶点1坐标  
  1.0, 0.0, 0.0, //顶点2坐标  
  0.0, 1.0, 0.0, //顶点3坐标  
]);  
renderPass.draw(3); // 绘制顶点数据
```

js

下面通过6个顶点绘制两个三角形。



```
//创建顶点数据  
const vertexArray = new Float32Array([  
  // 三角形1三个顶点坐标的x、y、z值
```

js

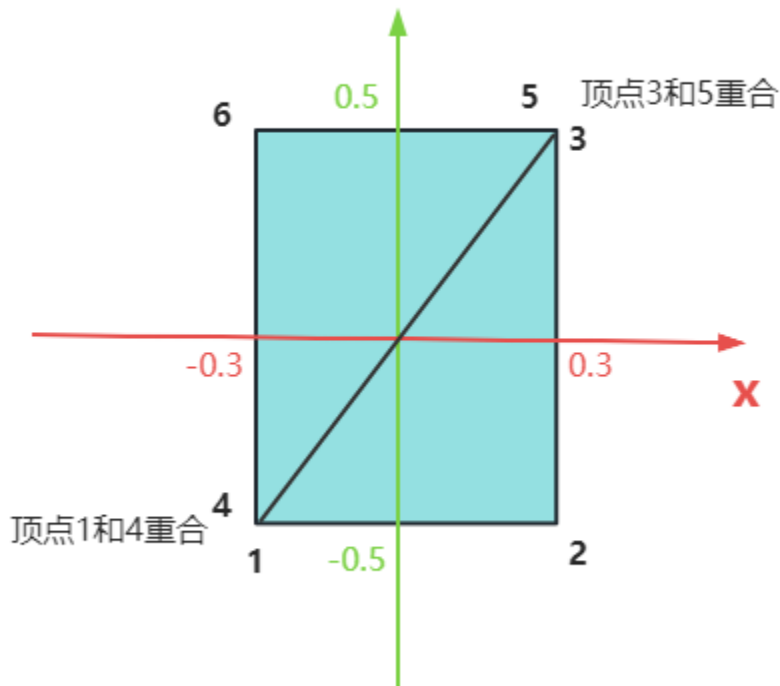
```

0.0, 0.0, 0.0, //顶点1坐标
1.0, 0.0, 0.0, //顶点2坐标
0.0, 1.0, 0.0, //顶点3坐标
// 三角形2三个顶点坐标的x、y、z值
-0.5, -0.5, 0.0, //顶点4坐标
-1.0, -0.5, 0.0, //顶点5坐标
-0.5, -1.0, 0.0, //顶点6坐标
]);
renderPass.draw(6); // 绘制顶点数据

```

三角形拼接一个矩形

一个矩形至少需要两个三角形拼接表示，和上面两个三角形绘制代码逻辑一样，区别不过是，这两个三角形有个两顶点重合罢了。



注意一个面的多个三角形，正反面要保持一致，就是说沿着某个方向看过去，矩形的两个三角形，要么都是正面，要么都是反面，或者说要么都是顺时装，要么都是逆时针。

```

const vertexArray = new Float32Array([
  // 三角形1三个顶点顺时针(沿着z轴负半轴方向观察，如果正半轴方向观察就是逆时针)
  -0.3, -0.5, 0.0, //顶点1坐标
  0.3, -0.5, 0.0, //顶点2坐标
  0.3, 0.5, 0.0, //顶点3坐标
  // 三角形2三个顶点顺时针(沿着z轴负半轴方向观察，如果正半轴方向观察就是逆时针)
  -0.3, -0.5, 0.0, //顶点4坐标 与顶点1重合
  0.3, 0.5, 0.0, //顶点5坐标 与顶点3重合

```

js

```
-0.3, 0.5, 0.0, //顶点6坐标  
]);
```

← 8. WebGPU 3D坐标系(投影)

1. 数学基础(平移、旋转、缩放矩阵)→