我们可以编写多个着色程序，然后使用不同的程序绘制不同的图形

如下，上代码

<body style="margin: 0px;">

    <canvas id="canvas"></canvas>

    <script id="vertexShader" type="x-shader/x-vertex">

        void main() {

            gl\_Position = vec4(0.0, 0.0, 0.0, 1.0);

            gl\_PointSize = 100.0;

        }

    </script>

    <script id="fragmentShader" type="x-shader/x-fragment">

        void main() {

            gl\_FragColor = vec4(1.0, 1.0, 0.0, 1.0);

        }

    </script>

    <script id="vertexShader2" type="x-shader/x-vertex">

        void main() {

            gl\_Position = vec4(0.5, 0.5, 0.0, 1.0);

            gl\_PointSize = 100.0;

        }

    </script>

    <script id="fragmentShader2" type="x-shader/x-fragment">

        void main() {

            gl\_FragColor = vec4(1.0, 1.0, 0.0, 1.0);

        }

    </script>

    <script>

        // 初始化

        function init(context, vertexShaderId, fragmentShaderId) {

            //创建程序对象

            const program = context.createProgram();

            //创建顶点着色器

            const vertexShader = context.createShader(context.VERTEX\_SHADER);

            // 获取顶点着色器代码

            const vsSource = document.getElementById(vertexShaderId).innerText;

            //将着色器源文件传入着色器对象中

            context.shaderSource(vertexShader, vsSource);

            //编译着色器对象

            context.compileShader(vertexShader);

            //把顶点着色对象装进程序对象中

            context.attachShader(program, vertexShader);

            //创建片元着色器

            const fragmentShader = context.createShader(context.FRAGMENT\_SHADER);

            // 获取片元着色器代码

            const fsSource = document.getElementById(fragmentShaderId).innerText;

            //将着色器源文件传入着色器对象中

            context.shaderSource(fragmentShader, fsSource);

            //编译着色器对象

            context.compileShader(fragmentShader)

            //把片元着色对象装进程序对象中

            context.attachShader(program, fragmentShader);

            //连接webgl上下文对象和程序对象

            context.linkProgram(program);

            return program;

        }

        // canvas画布

        const canvas = document.getElementById('canvas');

        canvas.width = document.body.clientWidth;

        canvas.height = document.body.clientHeight;

        // webgl上下文

        const context = canvas.getContext('webgl');

        // 指定将要用来清理绘图区的颜色

        context.clearColor(0., 0.0, 0.0, 1.0);

        // 清理绘图区

        context.clear(context.COLOR\_BUFFER\_BIT);

        // 初始化上下文

        let program1 = init(context, 'vertexShader', 'fragmentShader');

        //启动程序对象

        context.useProgram(program1);

        // 绘制顶点

        context.drawArrays(context.POINTS, 0, 1);

        // 初始化上下文

        let program2 = init(context, 'vertexShader2', 'fragmentShader2');

        //启动程序对象

        context.useProgram(program2);

        // 绘制顶点

        context.drawArrays(context.POINTS, 0, 1);

    </script>

</body>

示例代码有2套着色器程序，分别绘制不同的点，如下

