Graphics Programming

2ND WEEK, 2022

Example: Drawing a Triangle

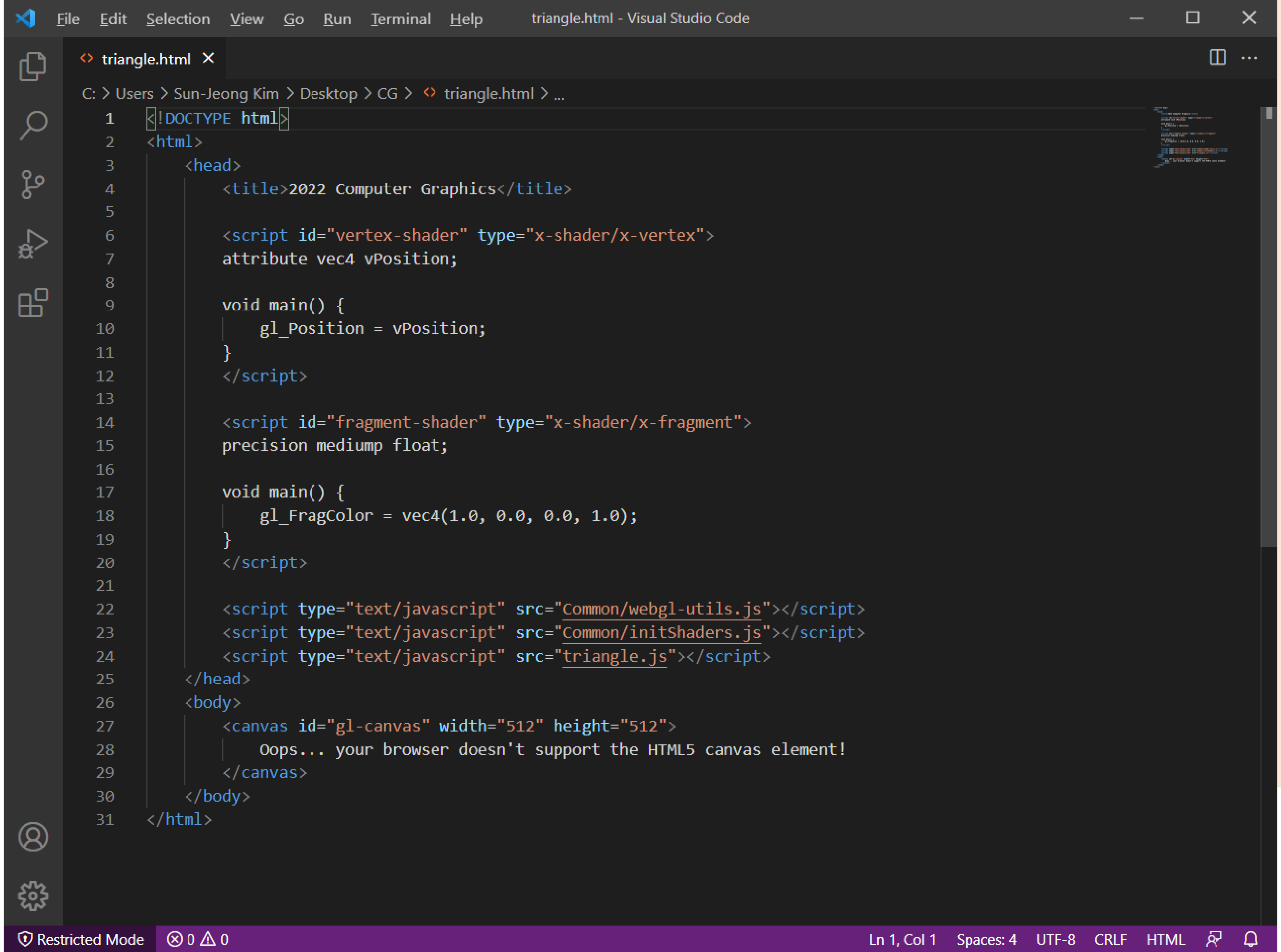
- Each application consists of (at least) two files
  - HTML file + a JavaScript file

- HTML
  - Describes page
  - Includes utilities
  - Includes shaders

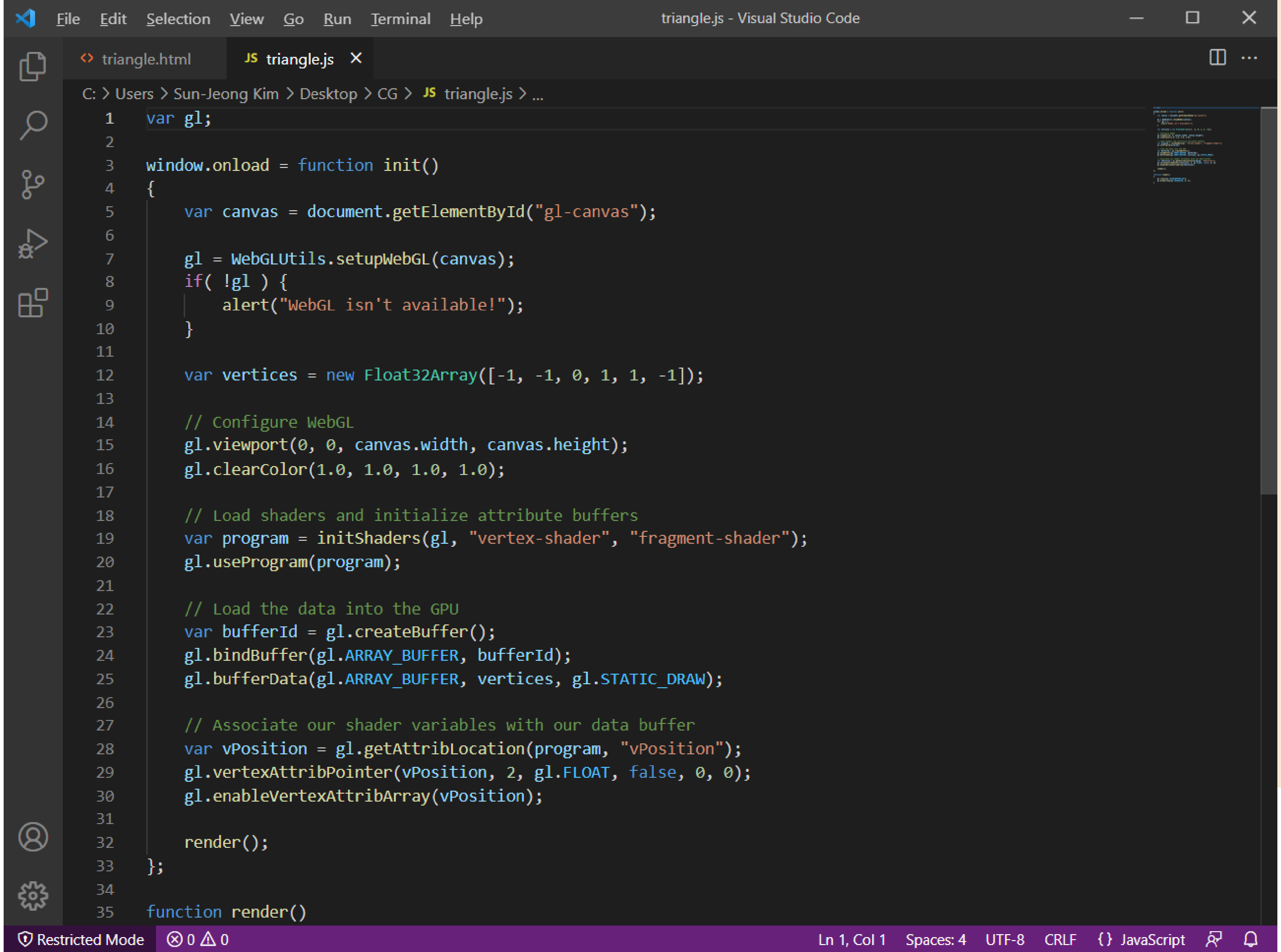
- Java Script
  - Contains the graphics

Coding in WebGL

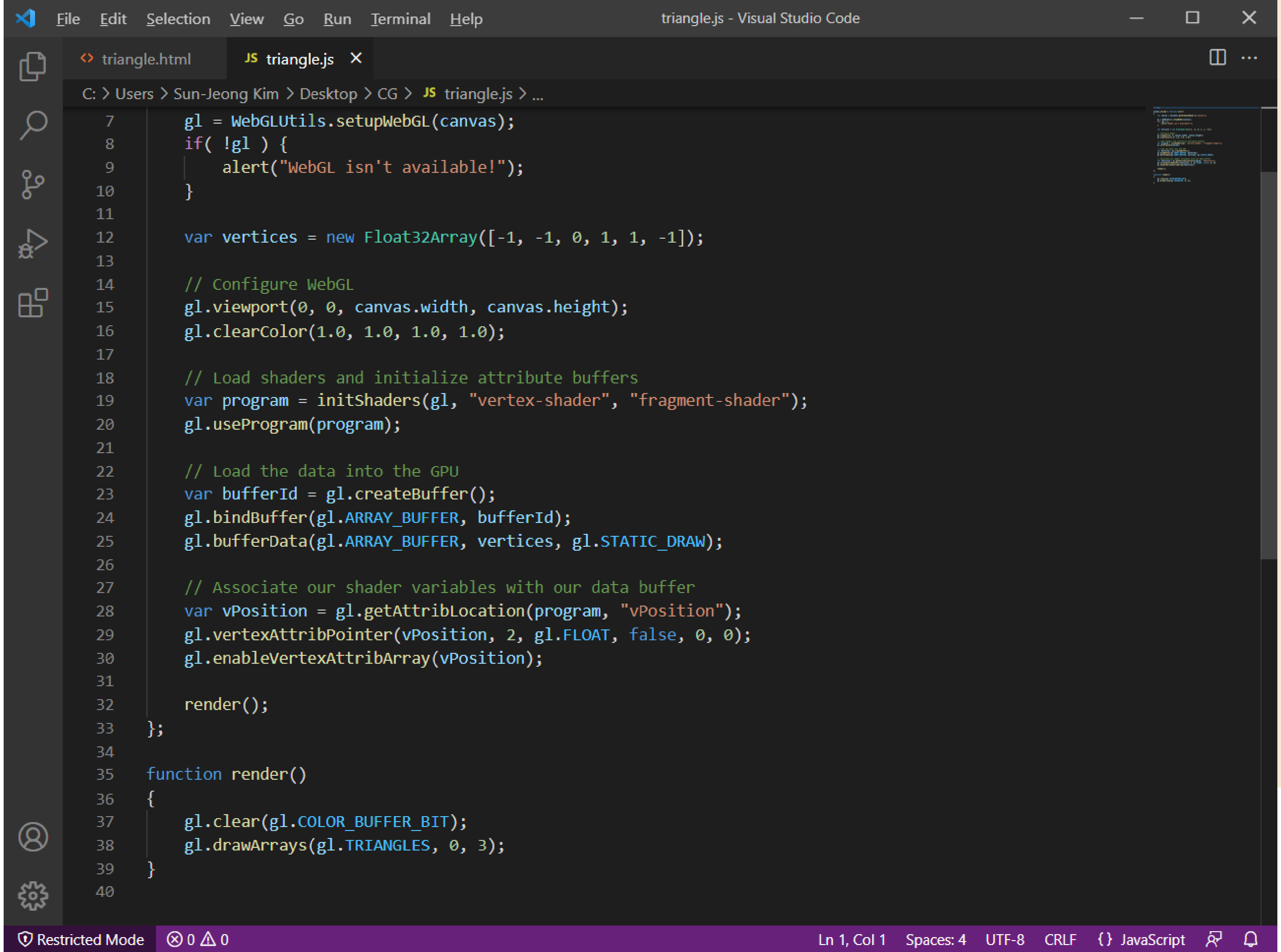
- Can run WebGL on any recent browser
  - Chrome
  - Firefox
  - Safari
  - IE
- Code written in JavaScript
- JS runs within browser
  - Use local resources


triangle.html - Visual Studio Code
File Edit Selection View Go Run Terminal Help
triangle.html
C: > Users > Sun-Jeong Kim > Desktop > CG > triangle.html > ...
Restricted Mode
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML

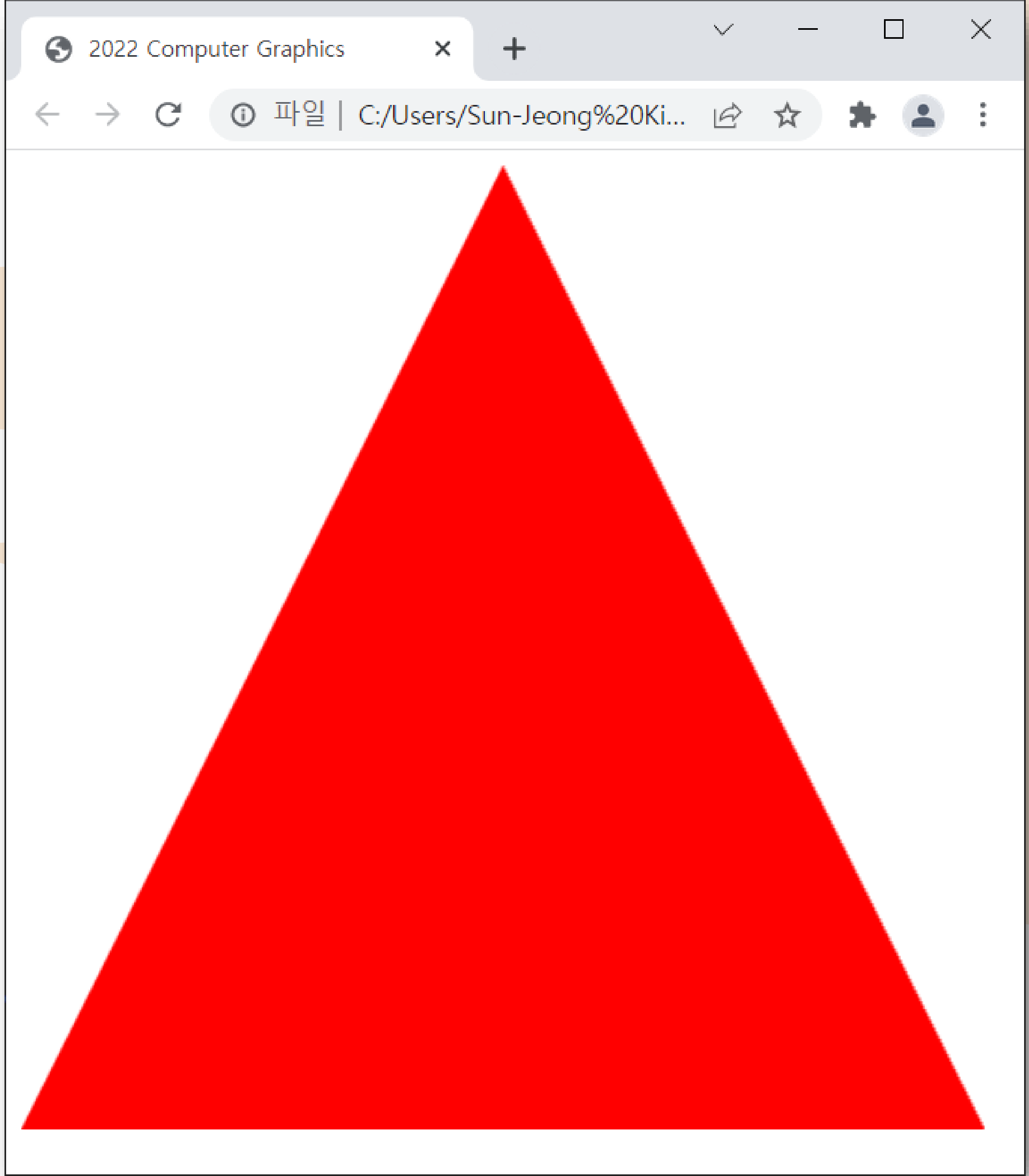
```html
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics</title>

        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;

        void main() {
            gl_Position = vPosition;
        }
        </script>

        <script id="fragment-shader" type="x-shader/x-fragment">
        precision mediump float;

        void main() {
            gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
        }
        </script>

        <script type="text/javascript" src="Common/webgl-utils.js"></script>
        <script type="text/javascript" src="Common/initShaders.js"></script>
        <script type="text/javascript" src="triangle.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
    </body>
</html>
```


File Edit Selection View Go Run Terminal Help
triangle.js - Visual Studio Code
triangle.html
triangle.js
C: > Users > Sun-Jeong Kim > Desktop > CG > triangle.js > ...
Restricted Mode
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript

```javascript
var gl;

window.onload = function init()
{
    var canvas = document.getElementById("gl-canvas");

    gl = WebGLUtils.setupWebGL(canvas);
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    var vertices = new Float32Array([-1, -1, 0, 1, 1, -1]);

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(1.0, 1.0, 1.0, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    render();
};

function render()
```


File Edit Selection View Go Run Terminal Help
triangle.js - Visual Studio Code
triangle.html
triangle.js
C: > Users > Sun-Jeong Kim > Desktop > CG > triangle.js > ...
Restricted Mode
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript

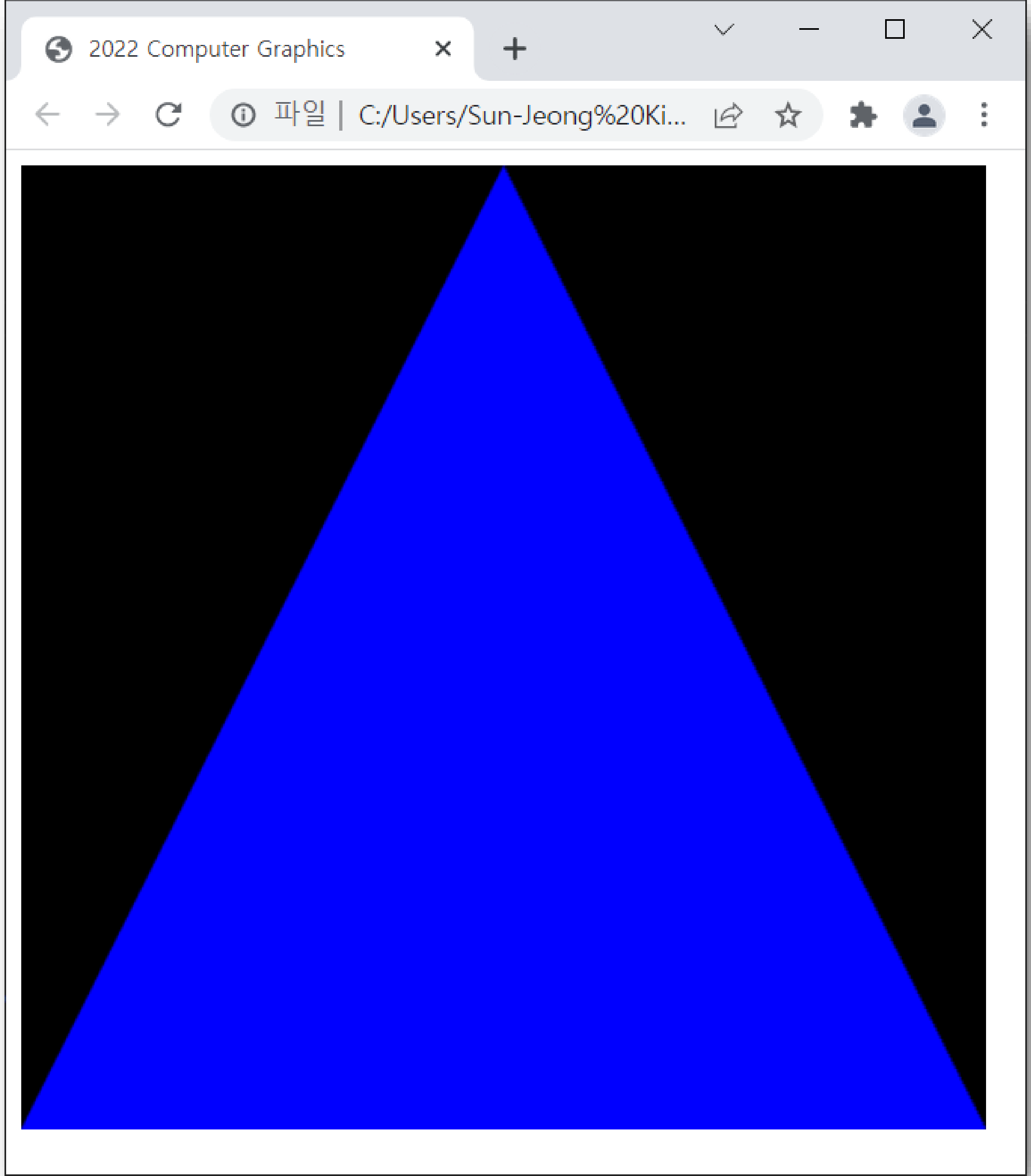
```javascript
    gl = WebGLUtils.setupWebGL(canvas);
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    var vertices = new Float32Array([-1, -1, 0, 1, 1, -1]);

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(1.0, 1.0, 1.0, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    render();
};

function render()
{
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.drawArrays(gl.TRIANGLES, 0, 3);
}

```


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%20Ki...

연습문제 (1)

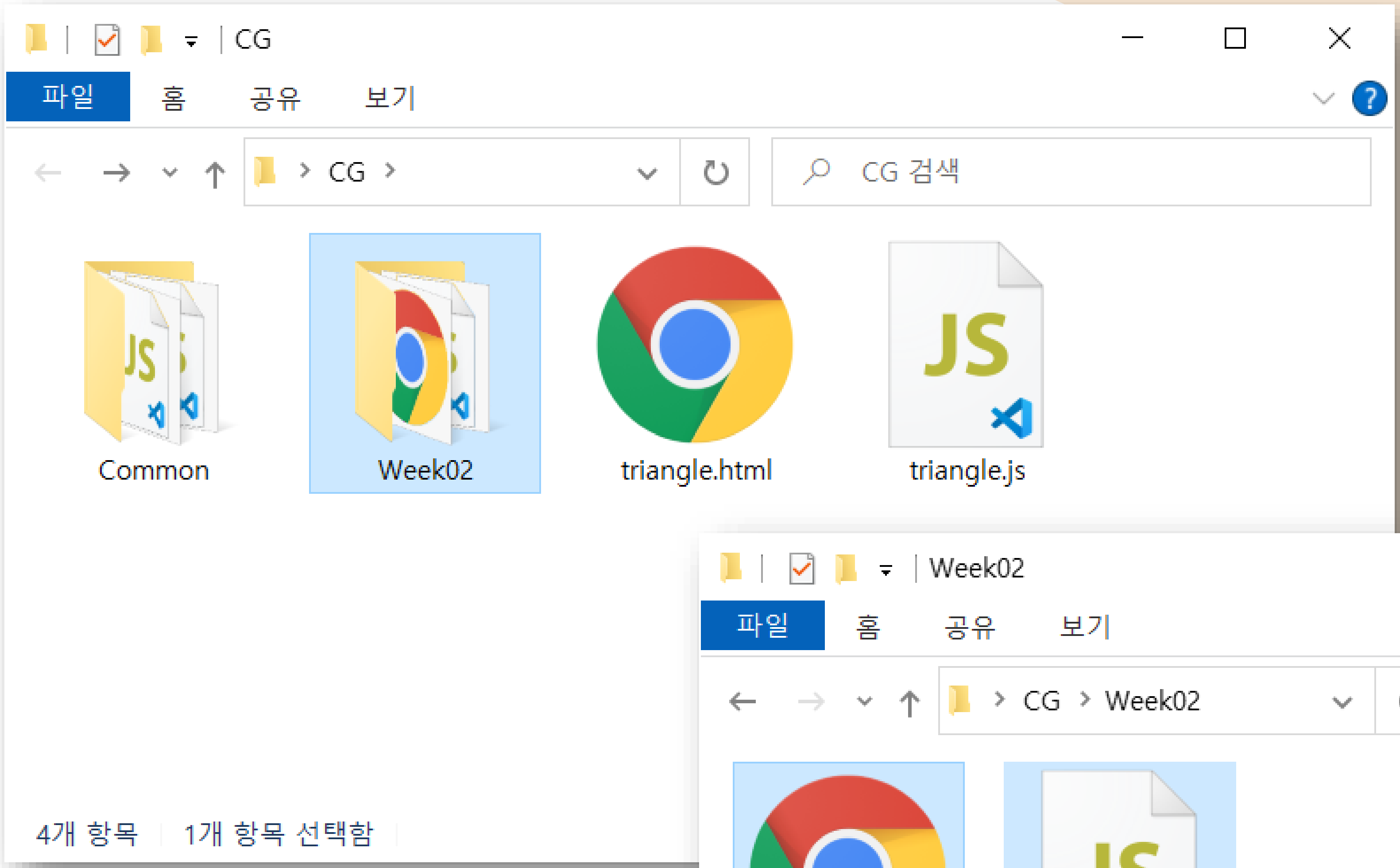
- 배경색을 변경하시오.
  - 흰색 → 검정색

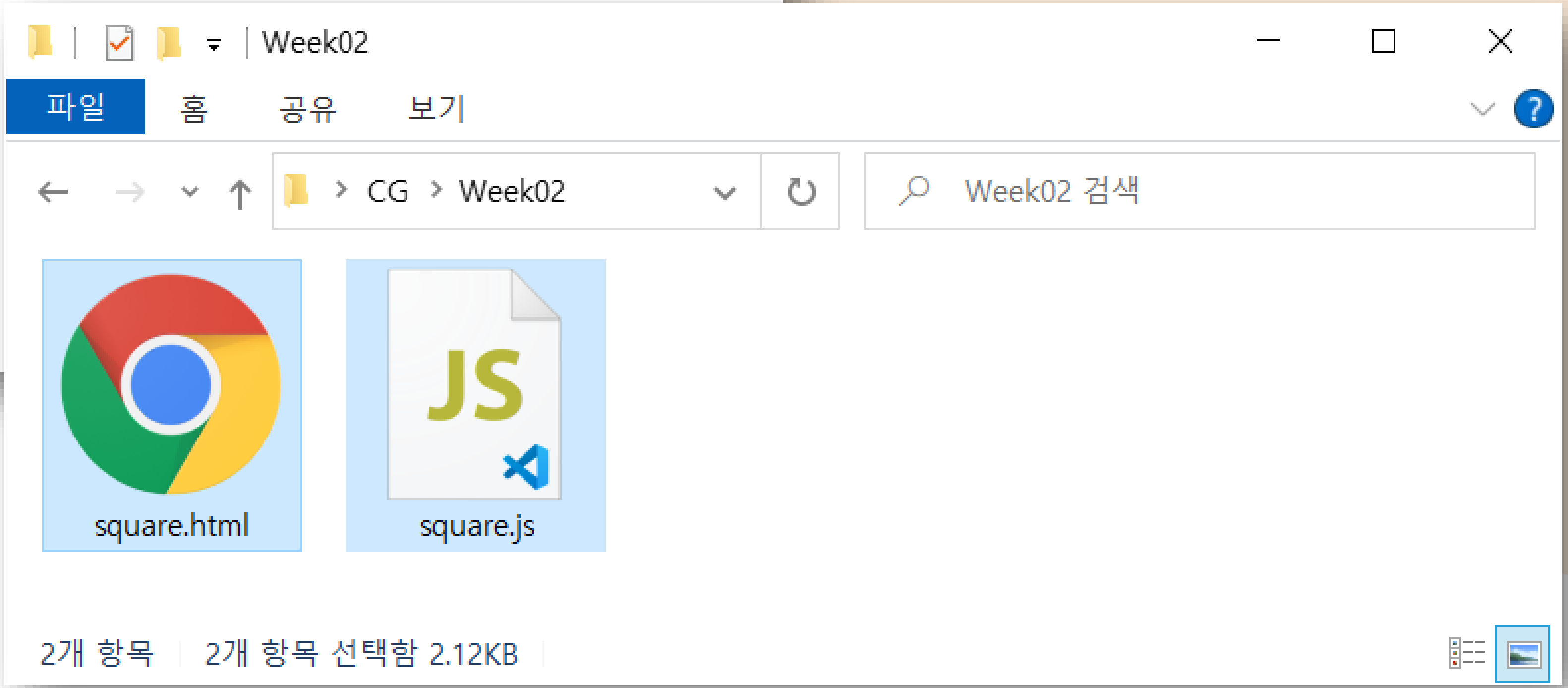
- 삼각형의 색상을 변경하시오
  - 빨강색 → 파랑색

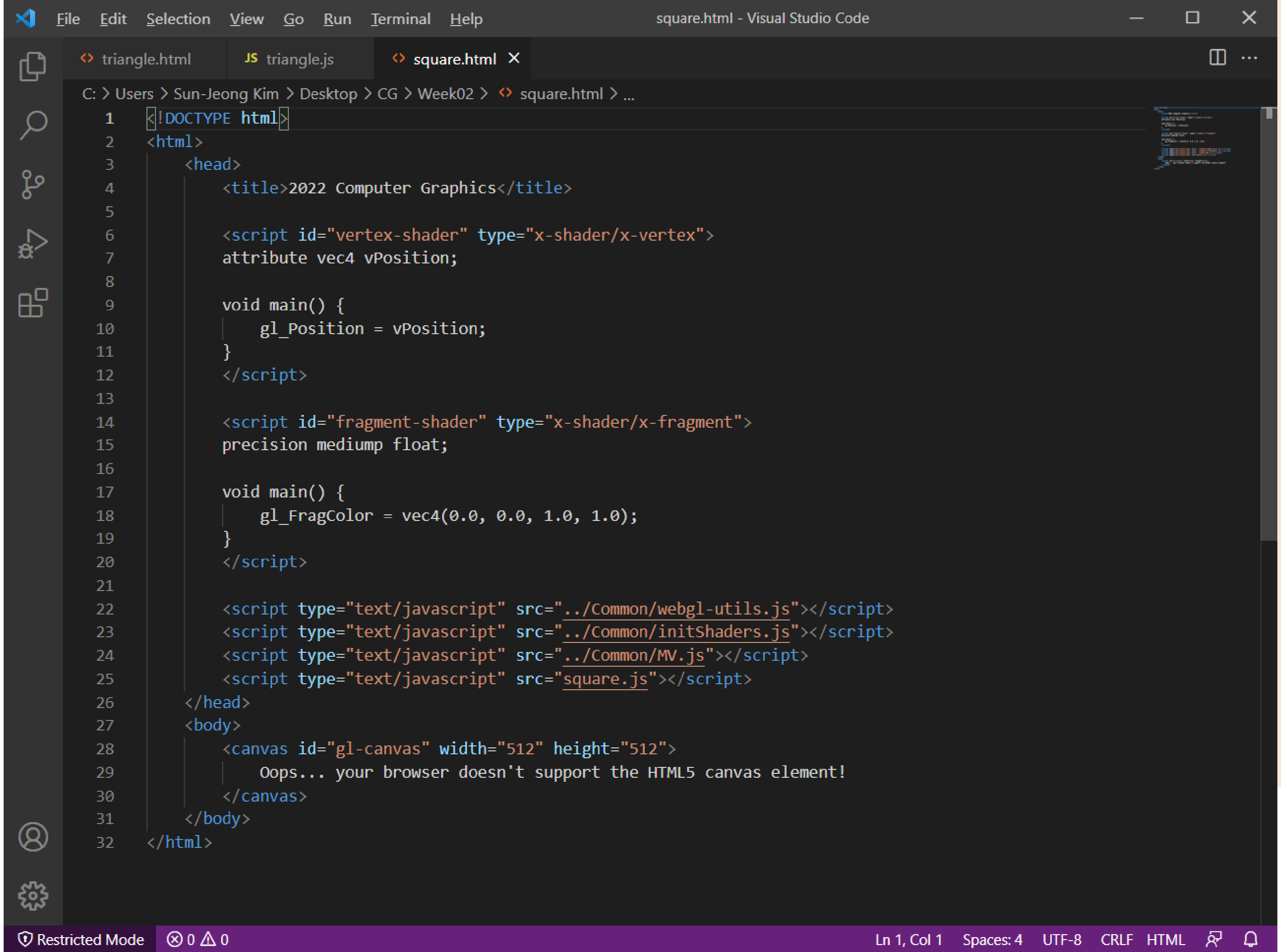


Example: Draw a Square

- WebGL – five steps
  - 1) <u>Describe page</u> (HTML file)
    - Request WebGL canvas
    - Read in necessary files
  - 2) Define <u>shaders</u> (HTML file)
    - Could be done with a separate file (browser dependent)
  - 3) <u>Compute</u> or specify data (JS file)
  - 4) <u>Send data to GPU</u> (JS file)
  - 5) <u>Render</u> data (JS file)


CG
파일
홈
공유
보기
CG
CG 검색
Common
Week02
triangle.html
triangle.js
4개 항목
1개 항목 선택함


Week02
파일
홈
공유
보기
CG
Week02
Week02 검색
square.html
square.js
2개 항목
2개 항목 선택함 2.12KB


File Edit Selection View Go Run Terminal Help
square.html - Visual Studio Code
triangle.html
triangle.js
square.html
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > square.html > ...
Restricted Mode
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML

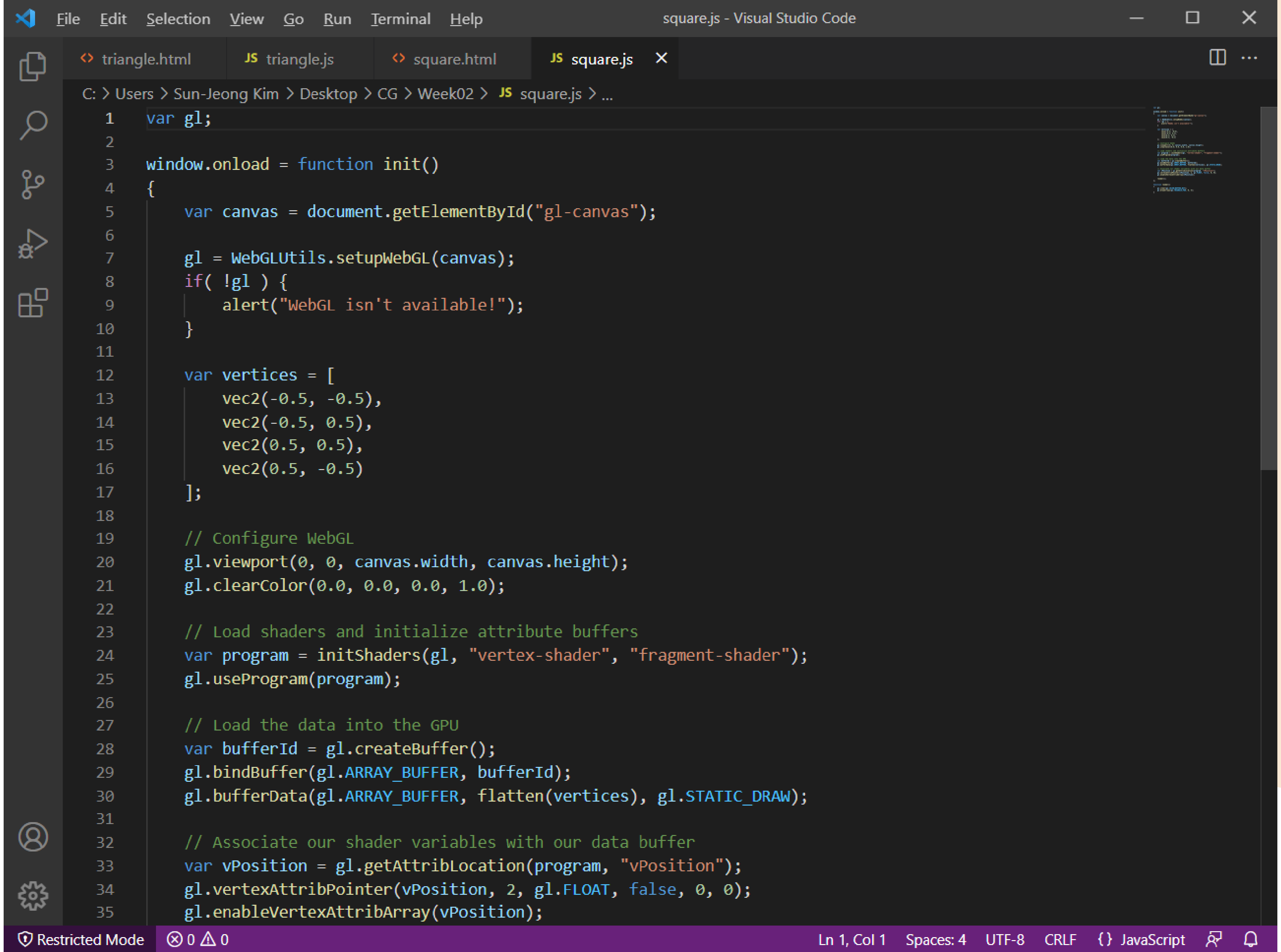
```html
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics</title>

        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;

        void main() {
            gl_Position = vPosition;
        }
        </script>

        <script id="fragment-shader" type="x-shader/x-fragment">
        precision mediump float;

        void main() {
            gl_FragColor = vec4(0.0, 0.0, 1.0, 1.0);
        }
        </script>

        <script type="text/javascript" src="../Common/webgl-utils.js"></script>
        <script type="text/javascript" src="../Common/initShaders.js"></script>
        <script type="text/javascript" src="../Common/MV.js"></script>
        <script type="text/javascript" src="square.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
    </body>
</html>
```

Shaders

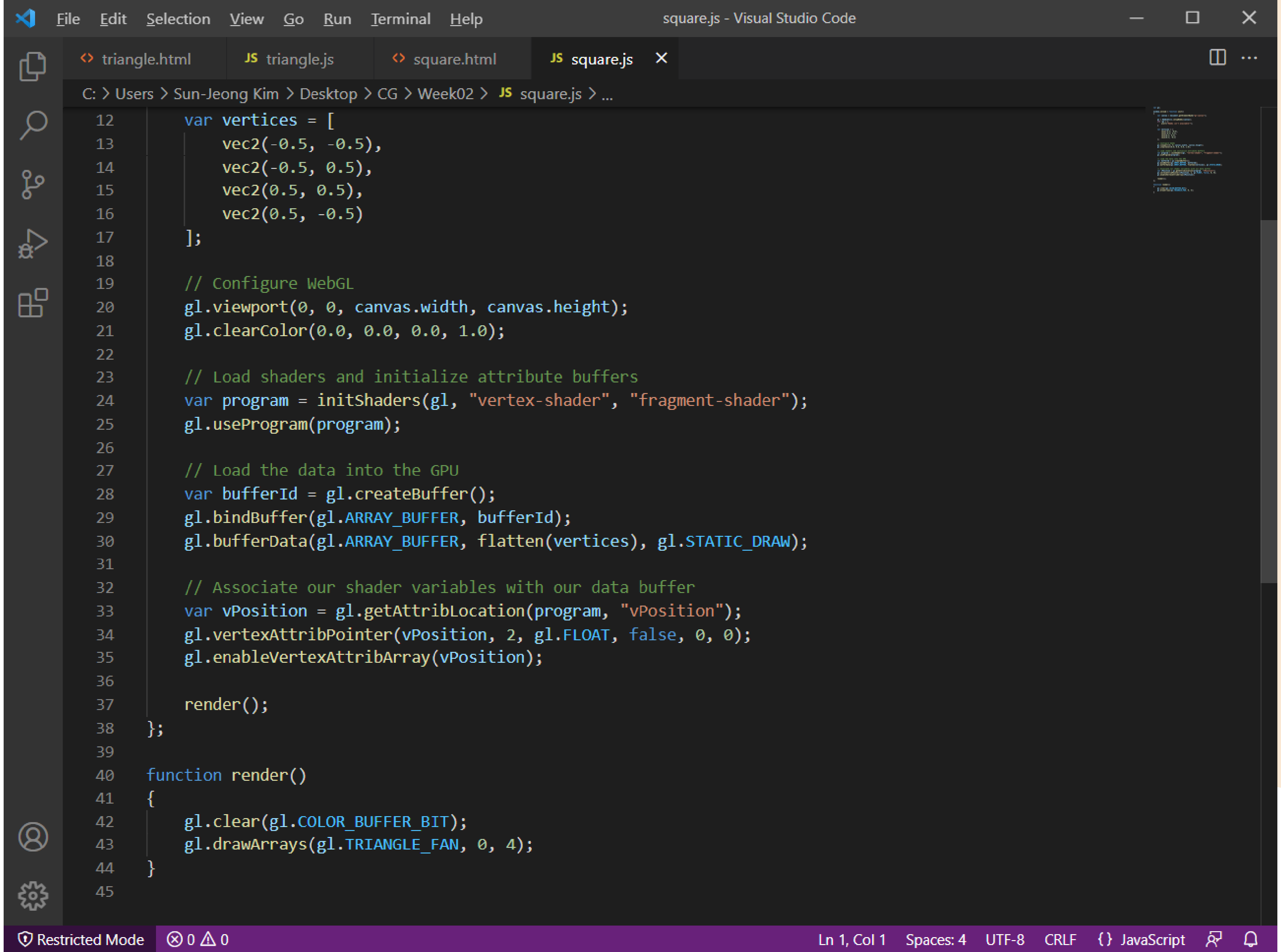
- We assign names to the shaders that we can use in the JS file
- These are trivial pass-through (do nothing) shaders that which set the two required built-in variables
    - gl_Position
    - gl_FragColor
- Note both shaders are full programs
- Note vector type vec2
- Must set precision in fragment shader

Files

- “Common/webgl-utils.js”
  - Standard utilities for setting up WebGL context in Common directory on website
- “Common/initShaders.js”
  - Contains JS and WebGL code for reading, compiling and linking the shaders
- “Common/MV.js”
  - Our matrix-vector package
- “square.js”
  - The application file


square.js - Visual Studio Code
File Edit Selection View Go Run Terminal Help
triangle.html
triangle.js
square.html
square.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > square.js > ...
Restricted Mode
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript

```javascript
var gl;

window.onload = function init()
{
    var canvas = document.getElementById("gl-canvas");

    gl = WebGLUtils.setupWebGL(canvas);
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    var vertices = [
        vec2(-0.5, -0.5),
        vec2(-0.5, 0.5),
        vec2(0.5, 0.5),
        vec2(0.5, -0.5)
    ];

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.0, 0.0, 0.0, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);
```


File Edit Selection View Go Run Terminal Help
square.js - Visual Studio Code
triangle.html
triangle.js
square.html
square.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > square.js > ...
Restricted Mode
0 0
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript

```javascript
    var vertices = [
        vec2(-0.5, -0.5),
        vec2(-0.5, 0.5),
        vec2(0.5, 0.5),
        vec2(0.5, -0.5)
    ];

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.0, 0.0, 0.0, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    render();
};

function render()
{
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.drawArrays(gl.TRIANGLE_FAN, 0, 4);
}
```

Notes

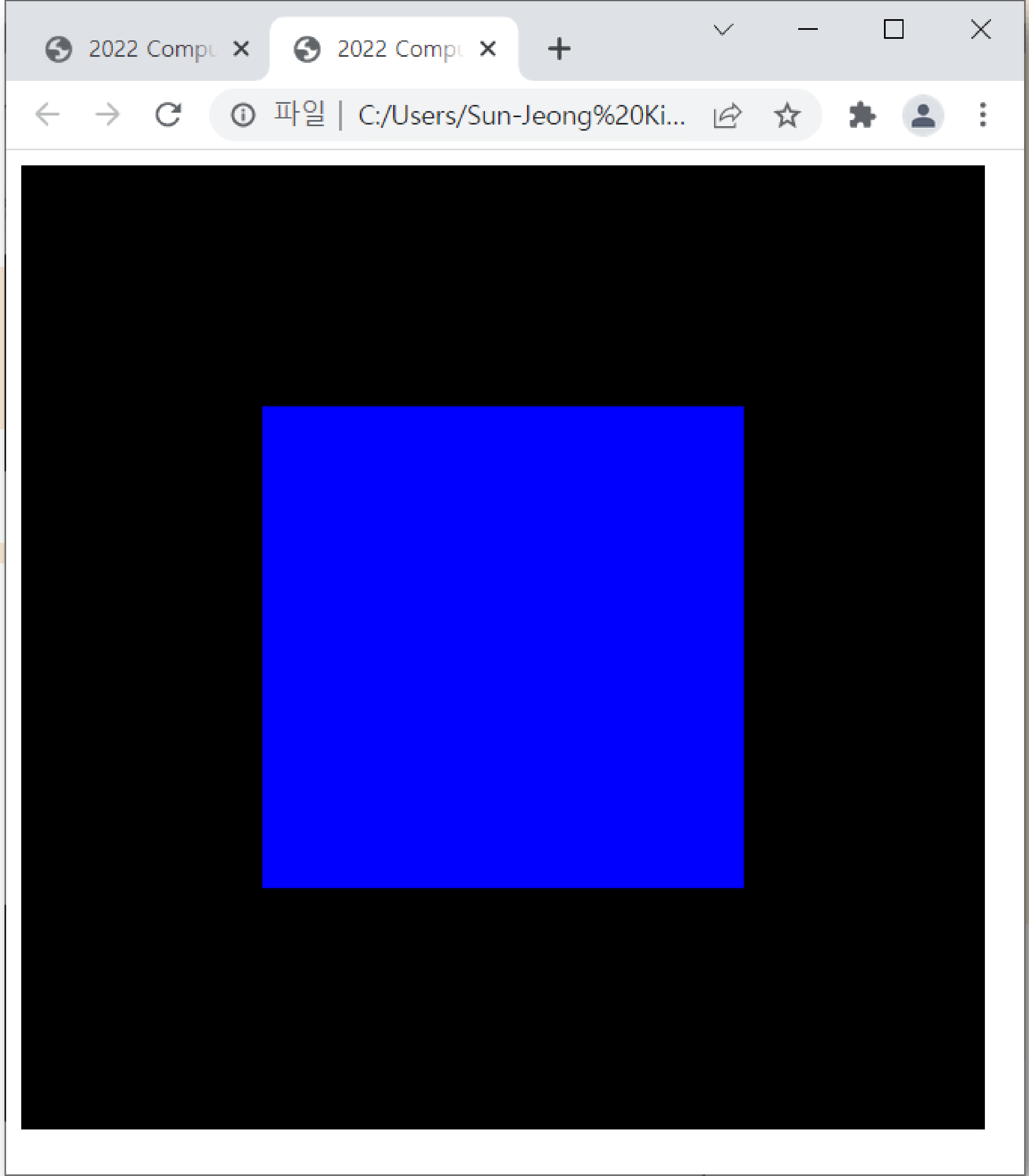
- onload
    - Determines where to start execution when all code is loaded
- canvas gets WebGL context form HTML file
- vertices use vec2 type in MV.js
- JS array is not same as a C or Java array
    - Object with methods
    - vertices.length // 4
- Values in clip coordinates

Notes

- initShaders used to load, compile and link shaders to form a program object

- Load data onto GPU by creating vertex buffer object on the GPU
  - Note use of flatten( ) to convert JS array to an array of float32's

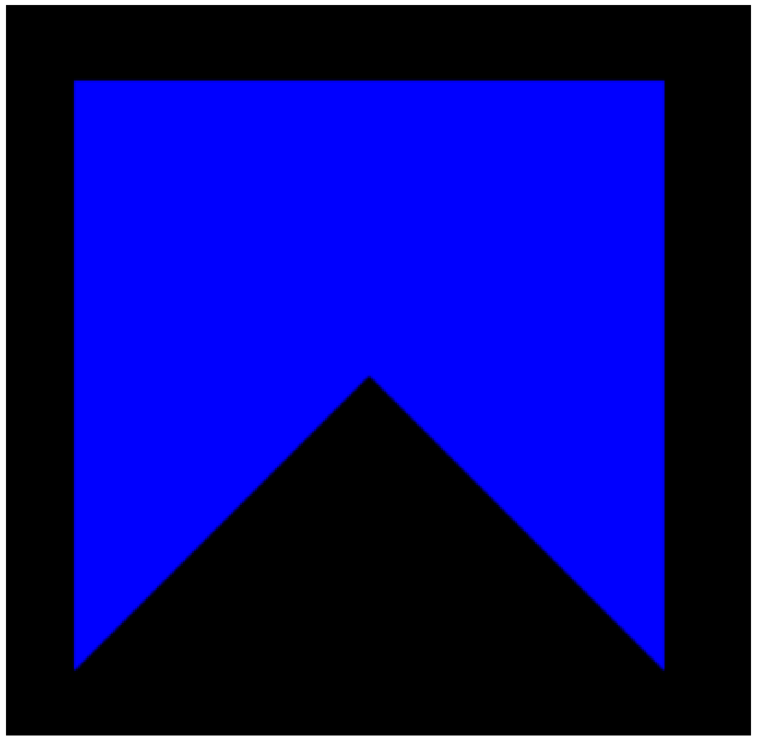
- Finally we must connect variable in program with variable in shader
  - Need name, type, location in buffer

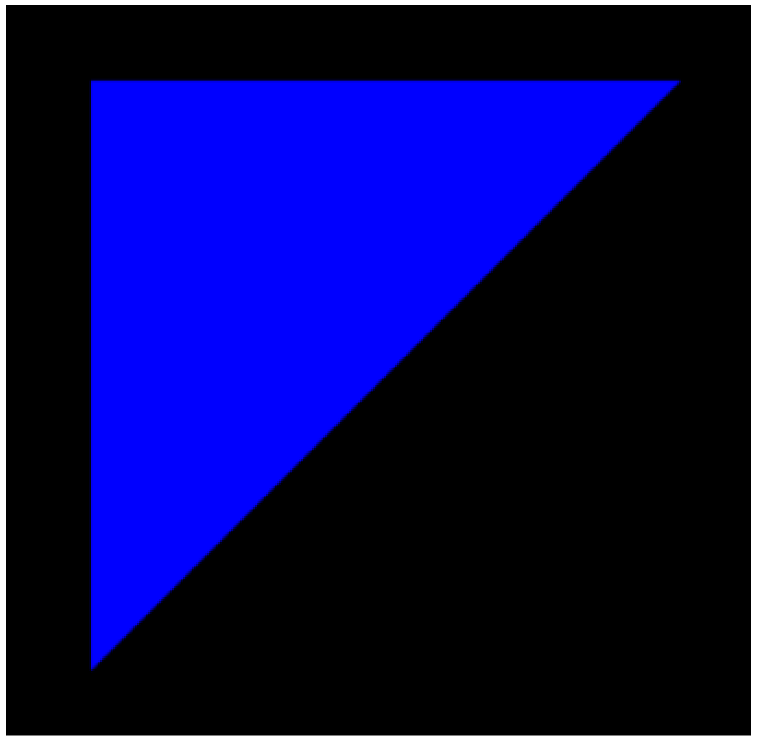

2022 Compu
2022 Compu
파일 | C:/Users/Sun-Jeong%20Ki...

연습문제 (2)

시험. 그려보기!

- gl.TRIANGLE_FAN 대신 아래 파라미터들을 이용했을 때, 각각에 대해 알맞은 출력 결과는 어느 쪽인가?
  - gl.TRIANGLES
  - gl.TRIANGLE_STRIP


(a)


(b)

Program Execution

- WebGL runs within the browser
  - Complex interaction among operating system, the window system, the browser and your code (HTML and JS)
- Simple model
  - Start with HTML file
  - Files read in asynchronously
  - Start with onload function
    - Event driven input

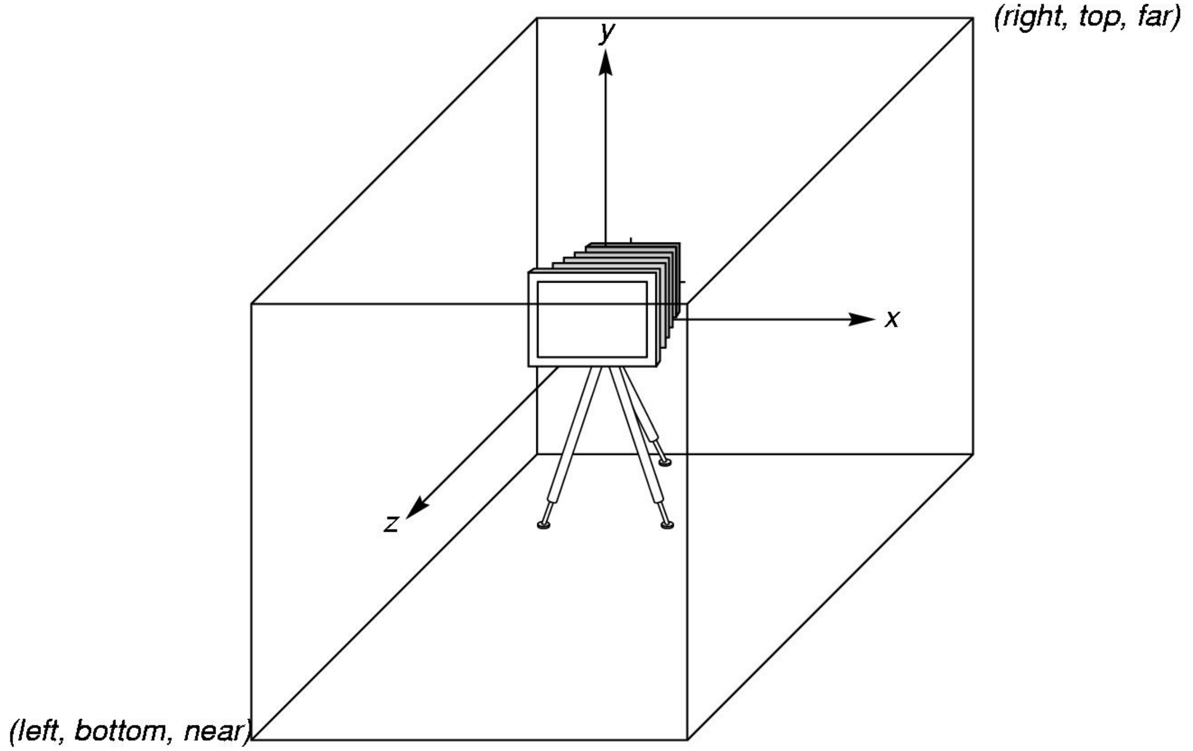
Coordinate System

- To specify vertex locations
  - Object or modeling coordinates
    - Vertices
  - World coordinates
    - Transformations
  - Camera or viewing coordinates
    - Viewing specification
  - Window or screen coordinates
    - Projection
    - Viewport transformations
  - Physical-device or device coordinates
    - Rasterization

WebGL Camera

정중앙에 0, 0 인
이유

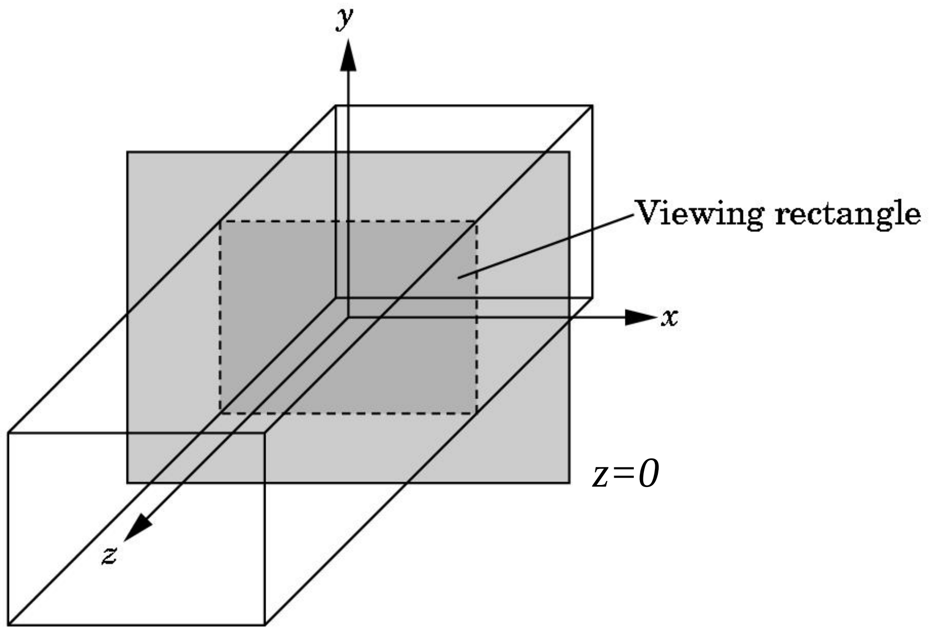
- WebGL places a camera at the origin in world space pointing in the negative z direction
  - Default view volume – a box centered at the origin with a side of length 2



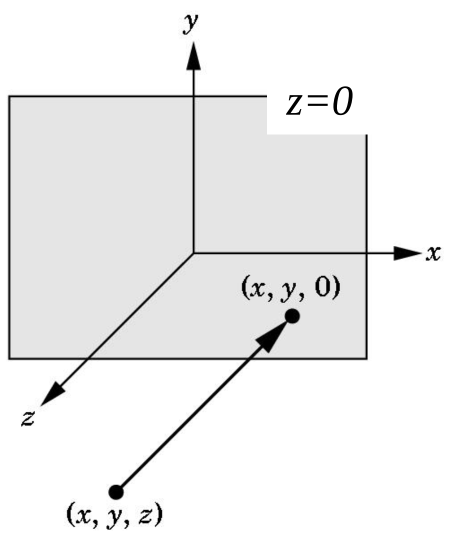
The default camera and an orthographic view volume

Orthographic Viewing

- Default orthographic view
  - Projecting points forward along the z axis onto the plane z=0



View Volume



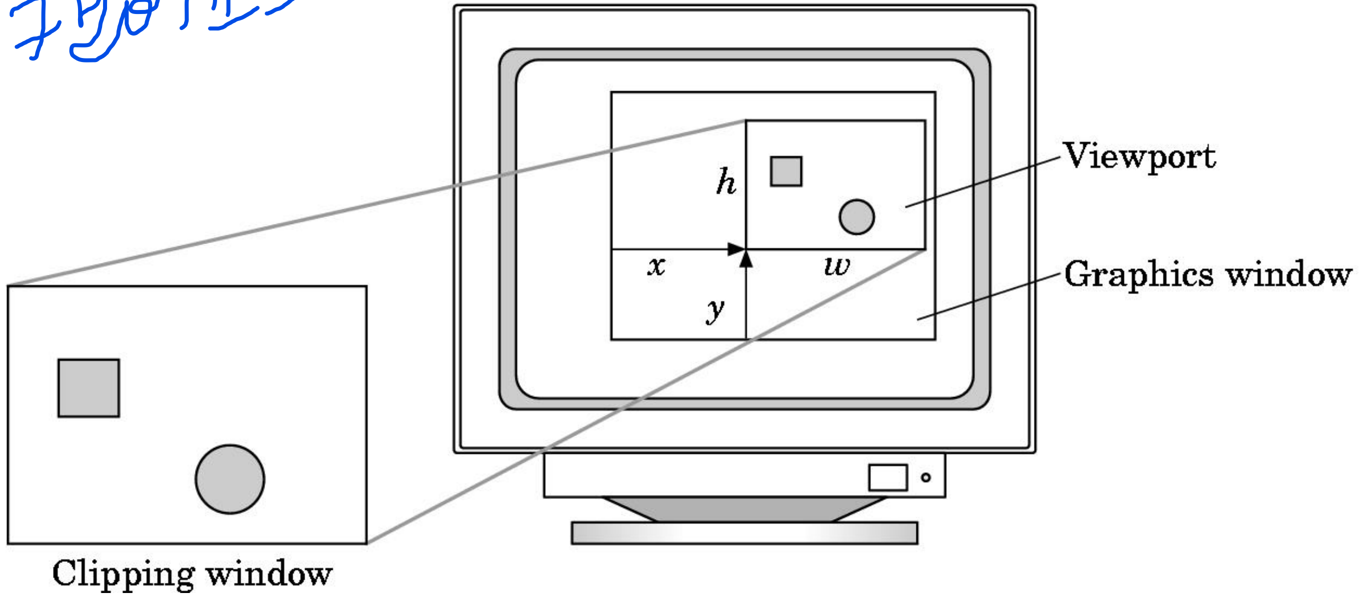
Orthographic Projection

Viewport

뷰포트에 따라 보이는 모양이 달라질 수 있음

[illegible]크기 = 뷰포트크기 여야 함

- Viewport
  - A rectangular area of the display window
    - Values in pixels: gl.viewport(x, y, w, h);
- Aspect ratio of a rectangle
  - The ratio of the rectangle's width to its height

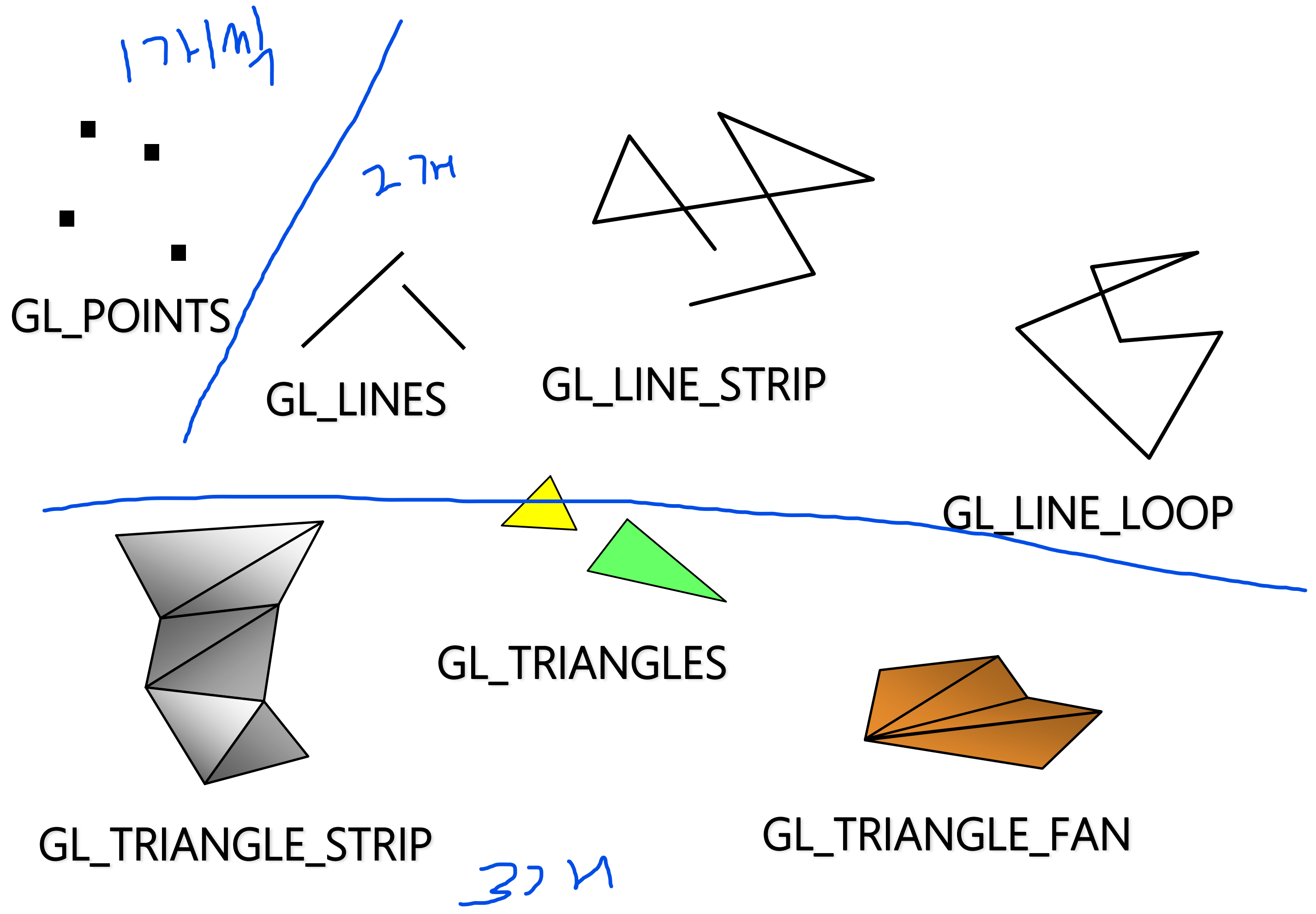


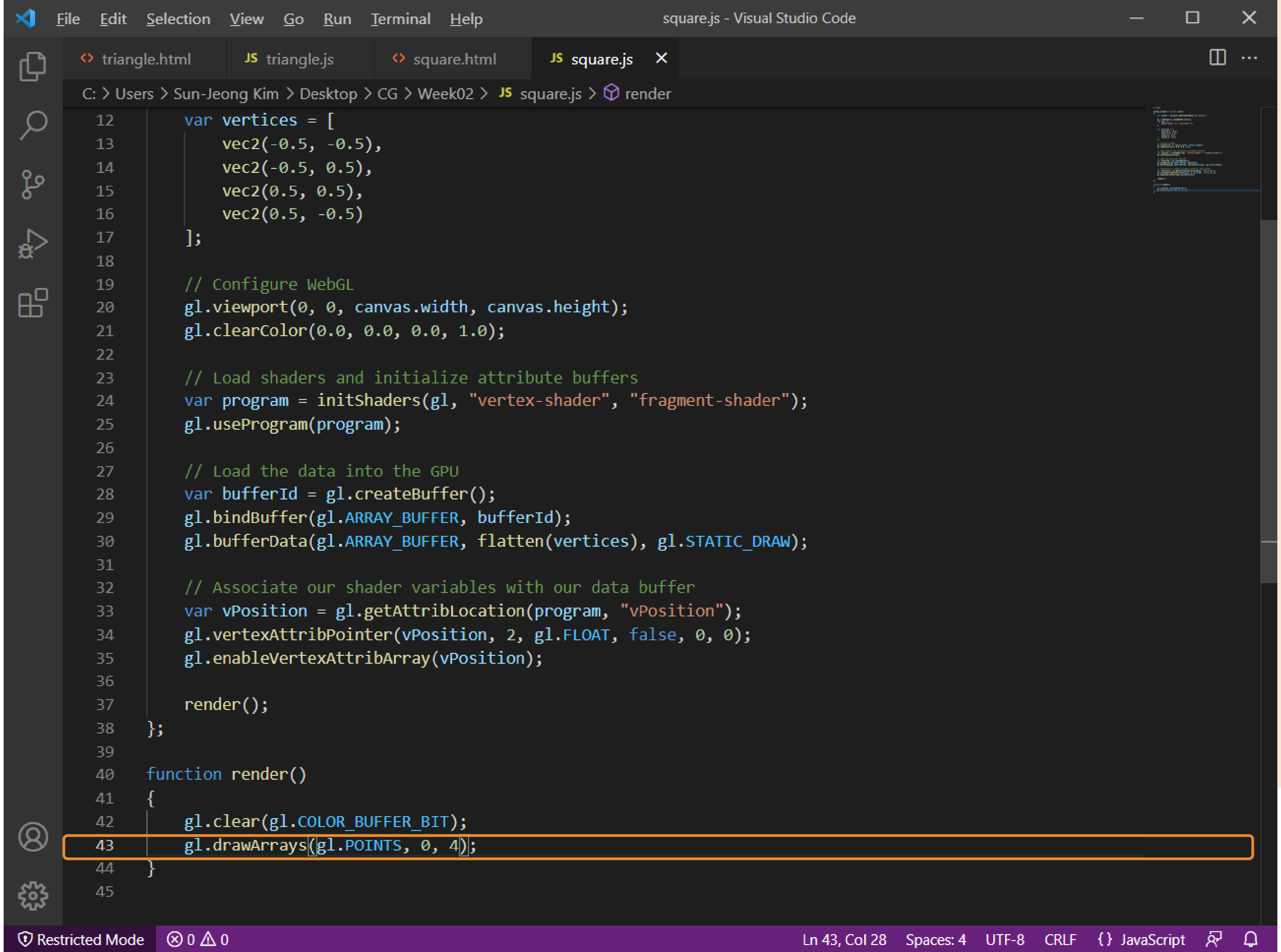
Transformations and Viewing

- In WebGL, projection is usually carried out using projection matrix (transformation) before rasterization
- Transformation functions are also used for changes in coordinates system
- Pre 3.1 OpenGL had a set of transformation functions which has been deprecated
- Three choices in WebGL
  - Application code
  - GLSL functions
  - MV.js

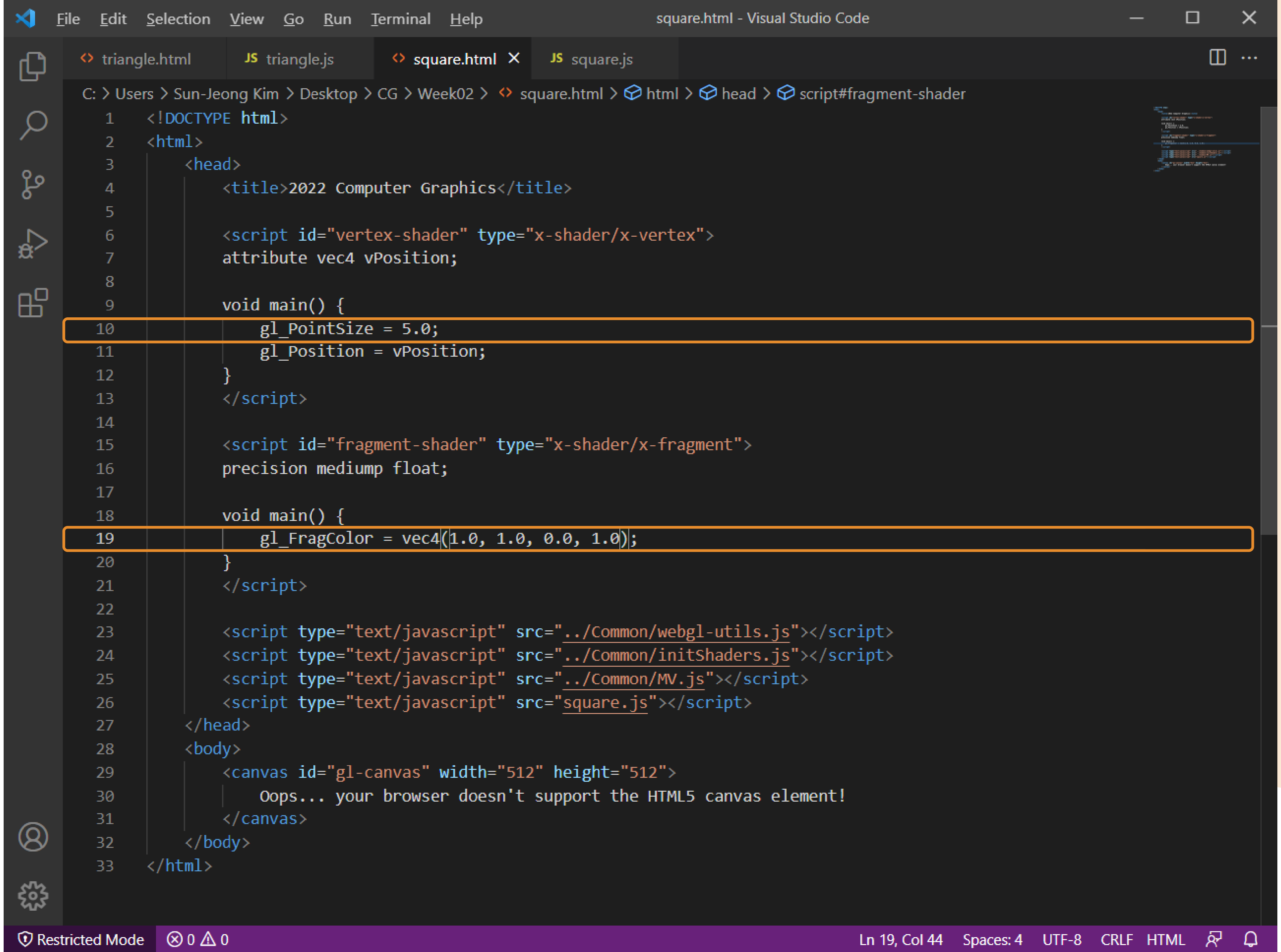
Geometric Primitives

- Points
- Lines
- Triangles

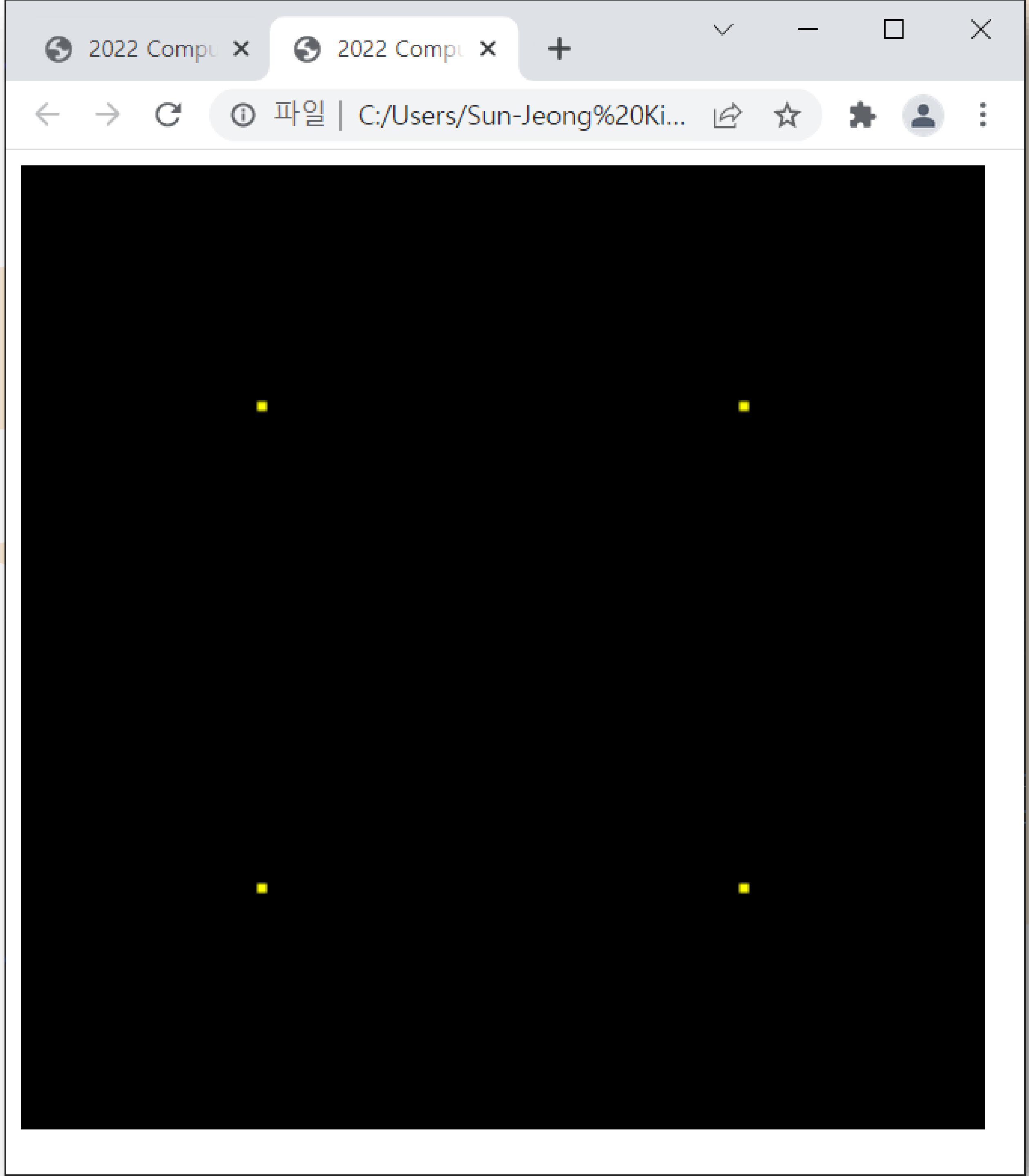


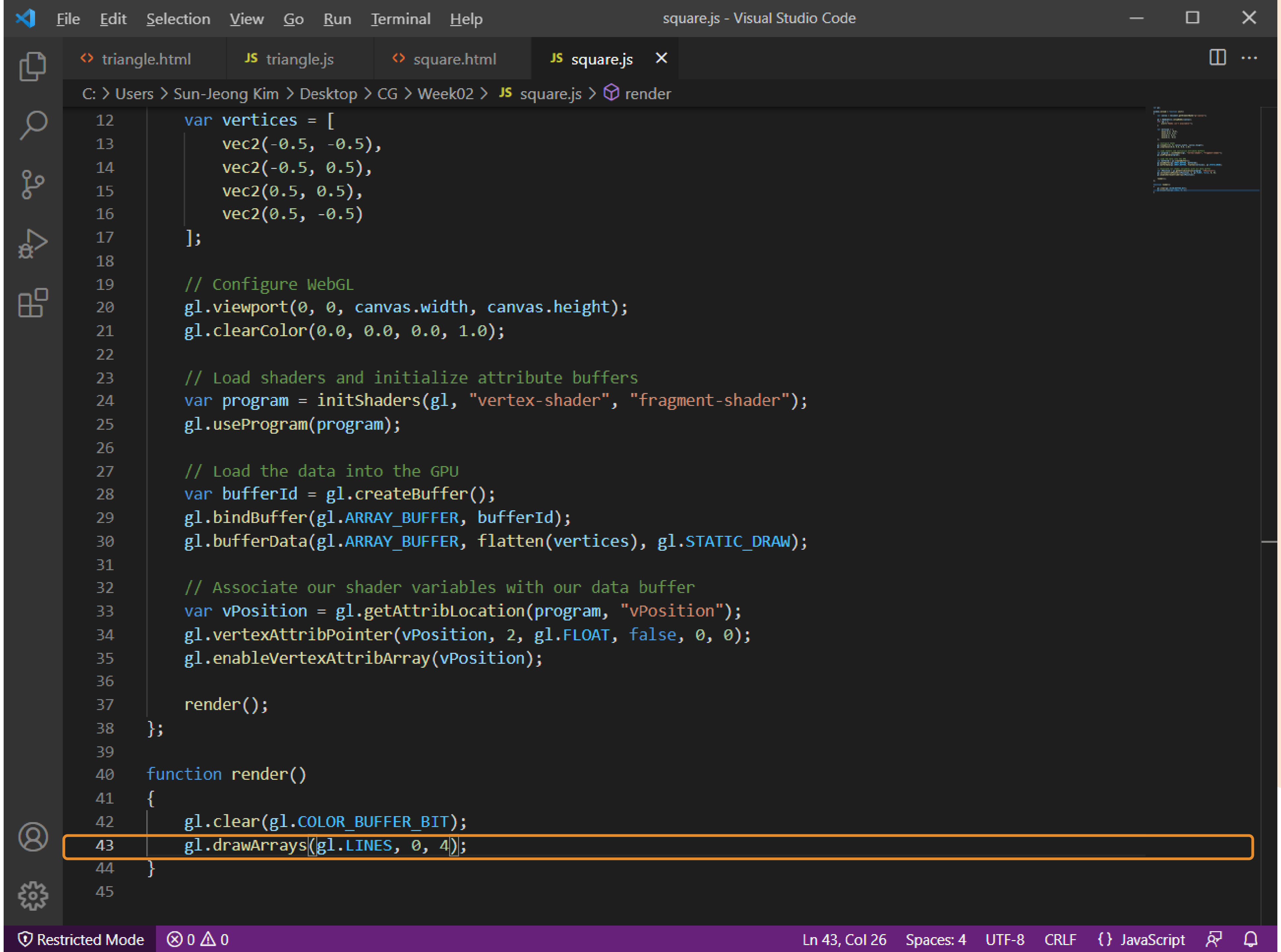

File Edit Selection View Go Run Terminal Help
square.js - Visual Studio Code
triangle.html
triangle.js
square.html
square.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > square.js > render
Restricted Mode
0 0
Ln 43, Col 28 Spaces: 4 UTF-8 CRLF {} JavaScript

```
    var vertices = [
        vec2(-0.5, -0.5),
        vec2(-0.5, 0.5),
        vec2(0.5, 0.5),
        vec2(0.5, -0.5)
    ];

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.0, 0.0, 0.0, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    render();
};

function render()
{
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.drawArrays(gl.POINTS, 0, 4);
}
```

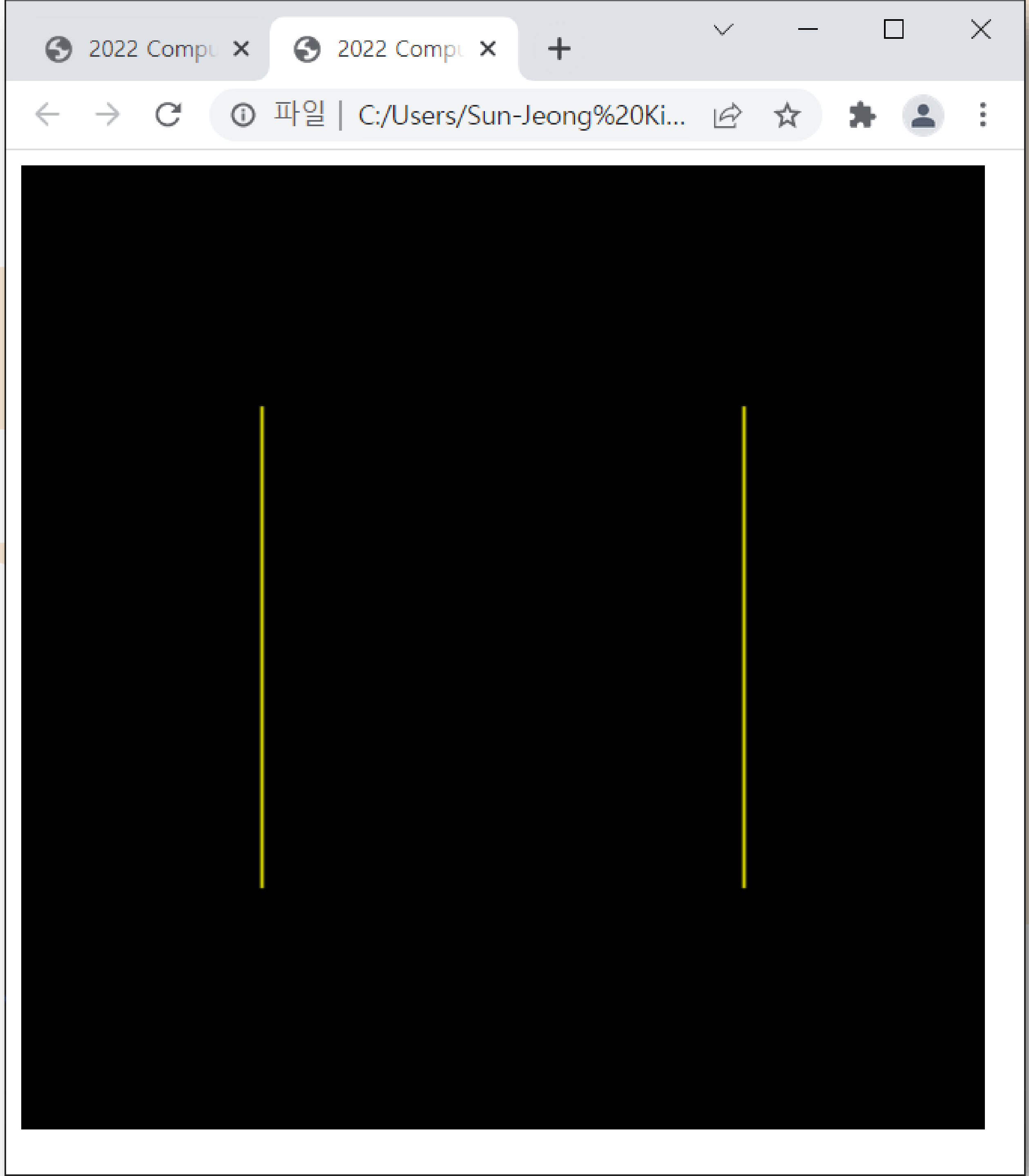

File Edit Selection View Go Run Terminal Help
square.html - Visual Studio Code
triangle.html
triangle.js
square.html
square.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > square.html > html > head > script#fragment-shader
Restricted Mode
Ln 19, Col 44 Spaces: 4 UTF-8 CRLF HTML

```html
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics</title>

        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;

        void main() {
            gl_PointSize = 5.0;
            gl_Position = vPosition;
        }
        </script>

        <script id="fragment-shader" type="x-shader/x-fragment">
        precision mediump float;

        void main() {
            gl_FragColor = vec4(1.0, 1.0, 0.0, 1.0);
        }
        </script>

        <script type="text/javascript" src="../Common/webgl-utils.js"></script>
        <script type="text/javascript" src="../Common/initShaders.js"></script>
        <script type="text/javascript" src="../Common/MV.js"></script>
        <script type="text/javascript" src="square.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
    </body>
</html>
```


2022 Compu
2022 Compu
파일 | C:/Users/Sun-Jeong%20Ki...

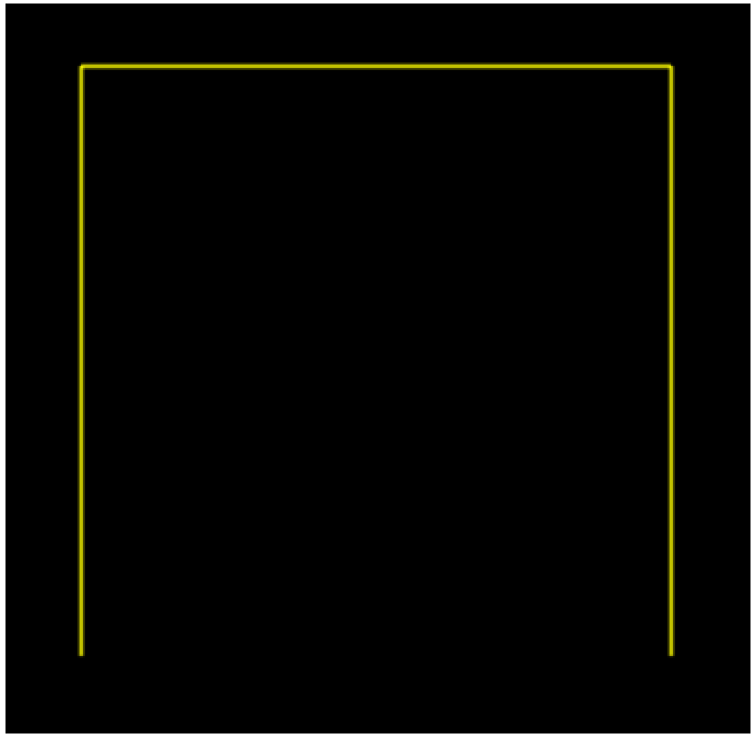

square.js - Visual Studio Code
File Edit Selection View Go Run Terminal Help
triangle.html
triangle.js
square.html
square.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > square.js > render
Restricted Mode
Ln 43, Col 26
Spaces: 4
UTF-8
CRLF
JavaScript

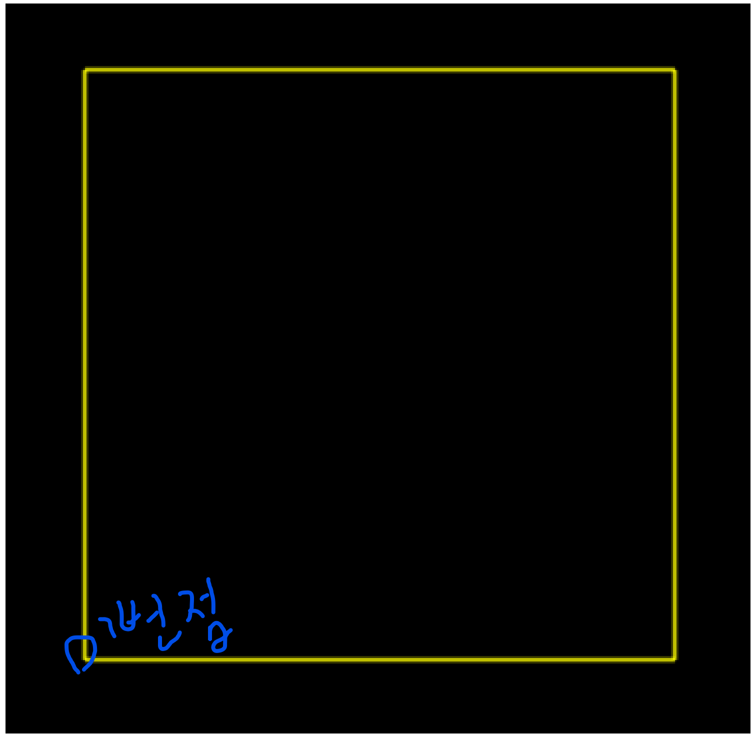
```javascript
    var vertices = [
        vec2(-0.5, -0.5),
        vec2(-0.5, 0.5),
        vec2(0.5, 0.5),
        vec2(0.5, -0.5)
    ];

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.0, 0.0, 0.0, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    render();
};

function render()
{
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.drawArrays(gl.LINES, 0, 4);
}
```


2022 Compu
2022 Compu
파일 | C:/Users/Sun-Jeong%20Ki...

연습문제 (3)

- gl.LINES 대신 아래 파라미터들을 이용했을 때, 각각에 대해 알맞은 출력 결과는 어느 쪽인가?
  - gl.LINE_STRIP a
  - gl.LINE_LOOP b

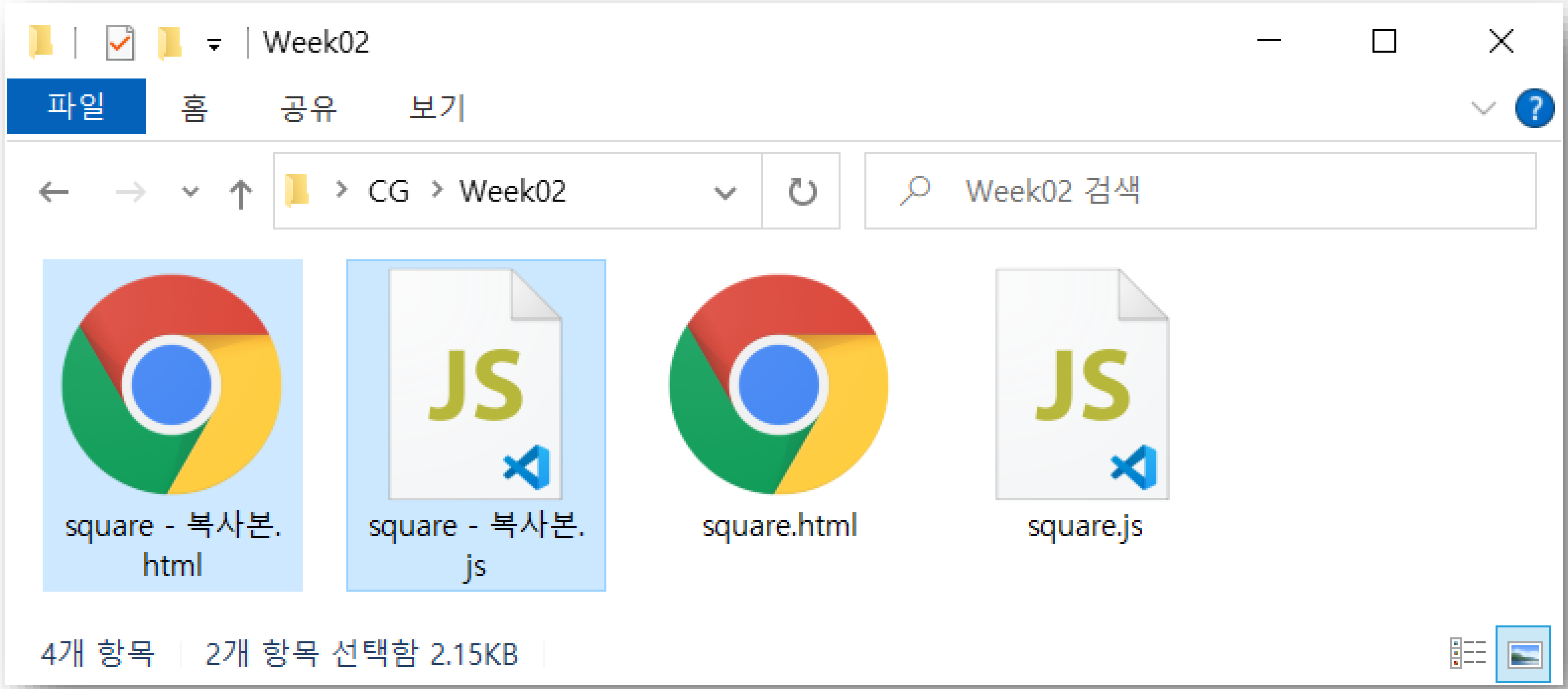

(a)

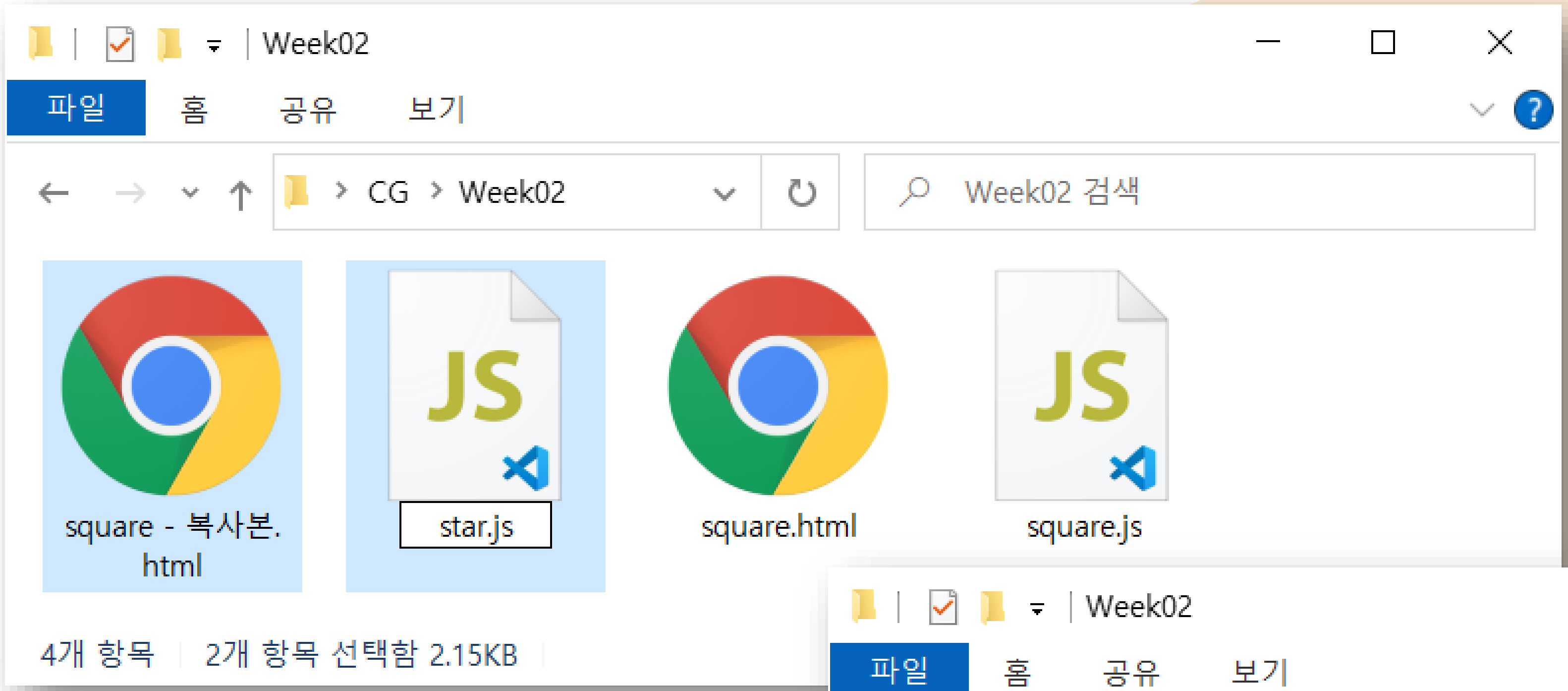


(b)

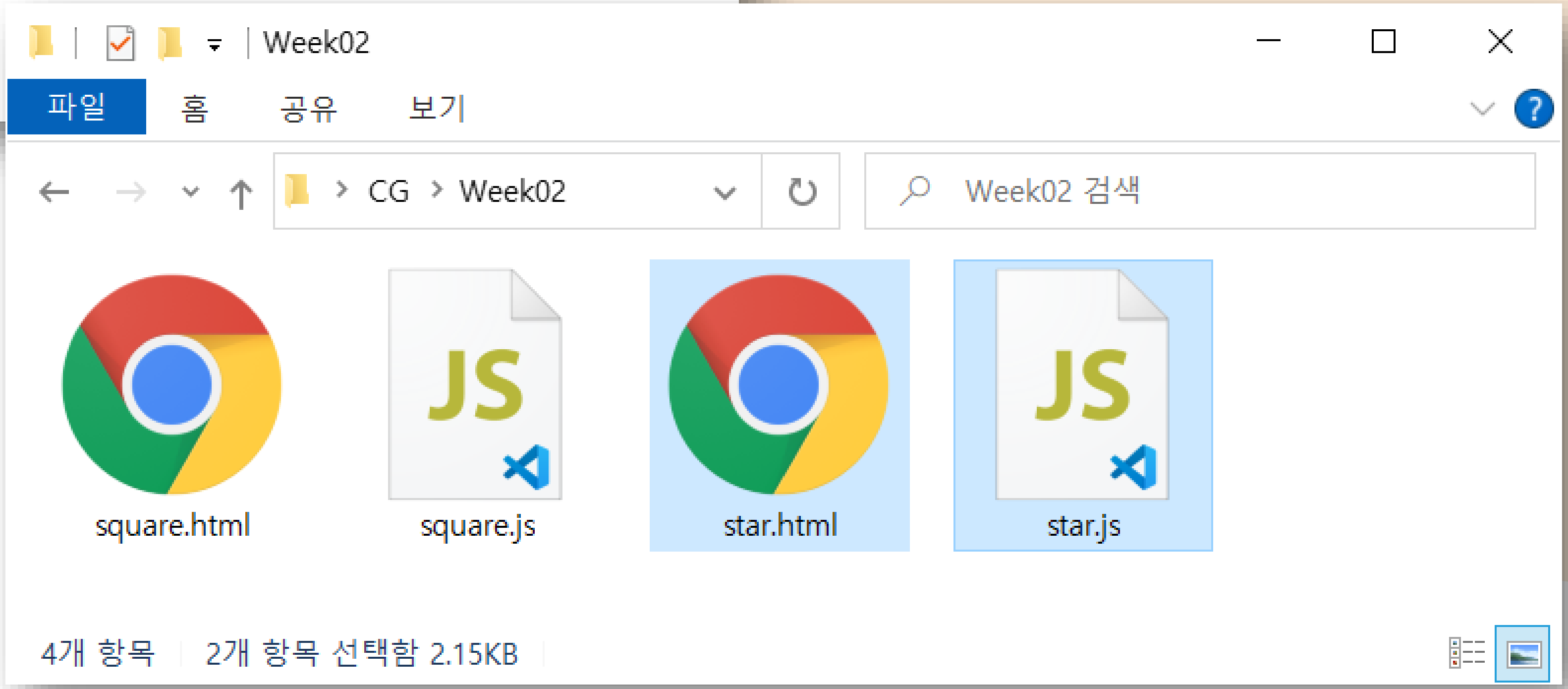
연습문제 (4)

- star.html과 star.js를 생성하시오.

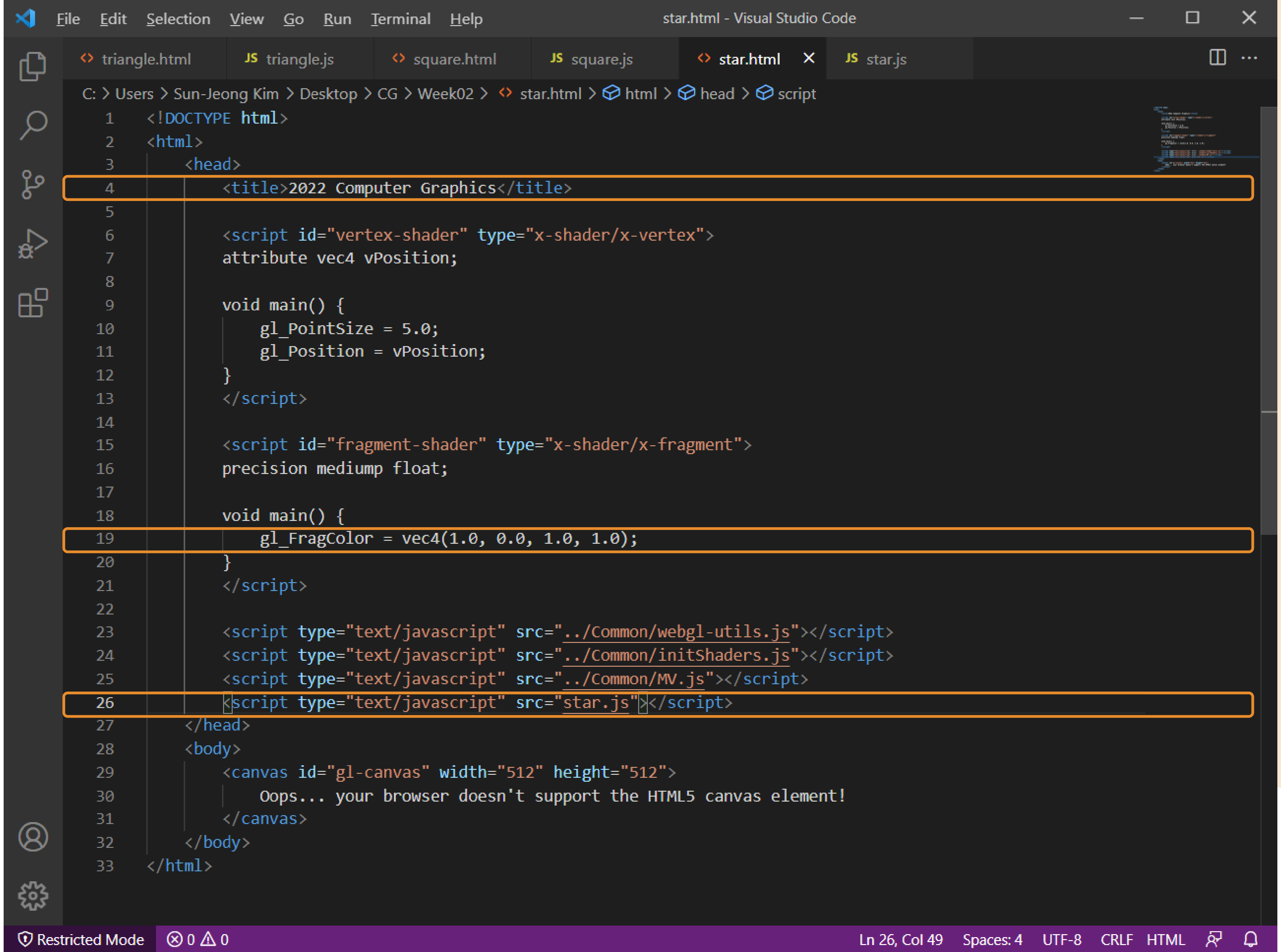




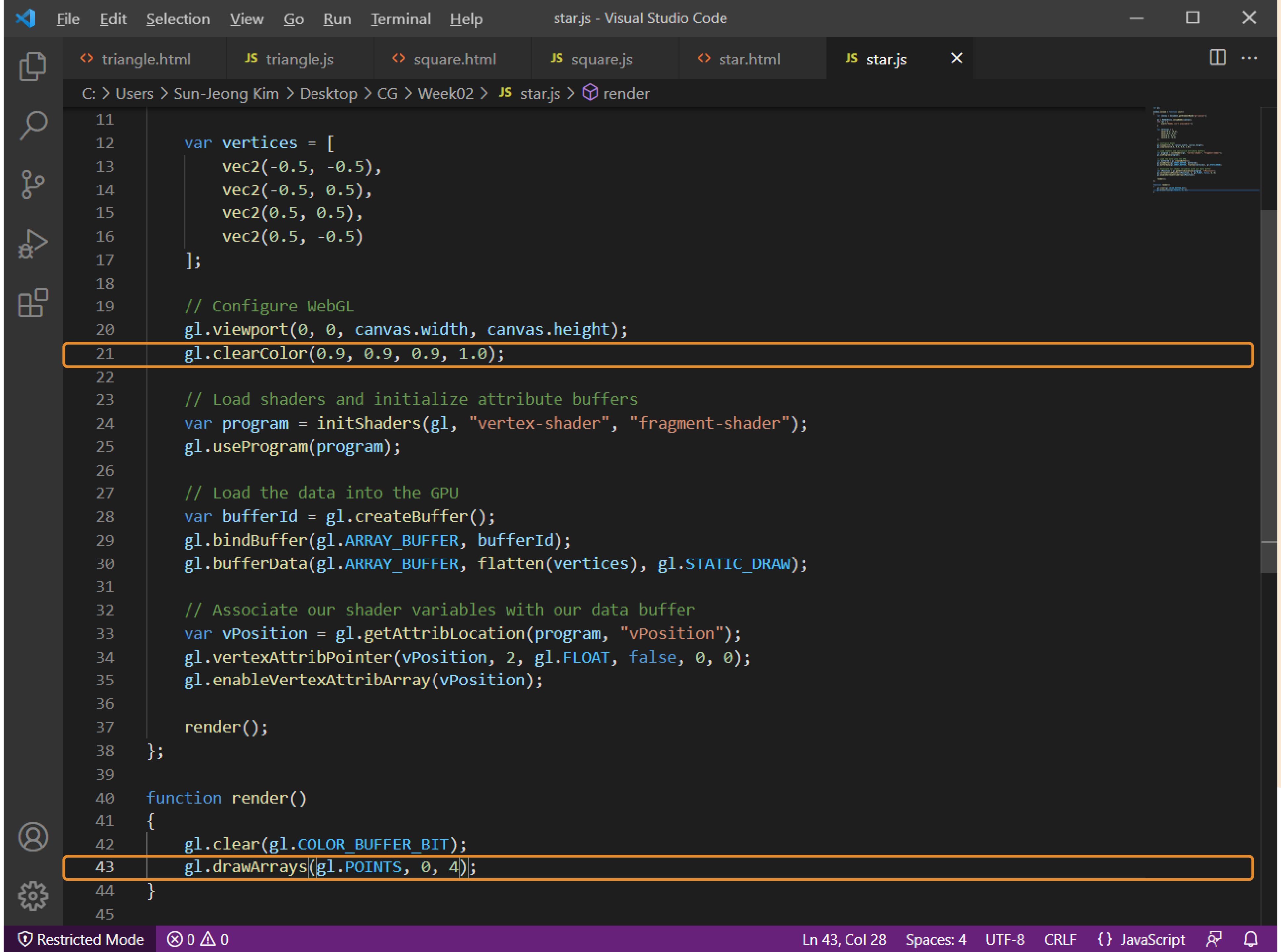
Week02
파일
홈
공유
보기
CG > Week02
Week02 검색
square - 복사본.html
star.js
square.html
square.js
4개 항목
2개 항목 선택함 2.15KB



Week02
파일
홈
공유
보기
CG > Week02
Week02 검색
square.html
square.js
star.html
star.js
4개 항목
2개 항목 선택함 2.15KB

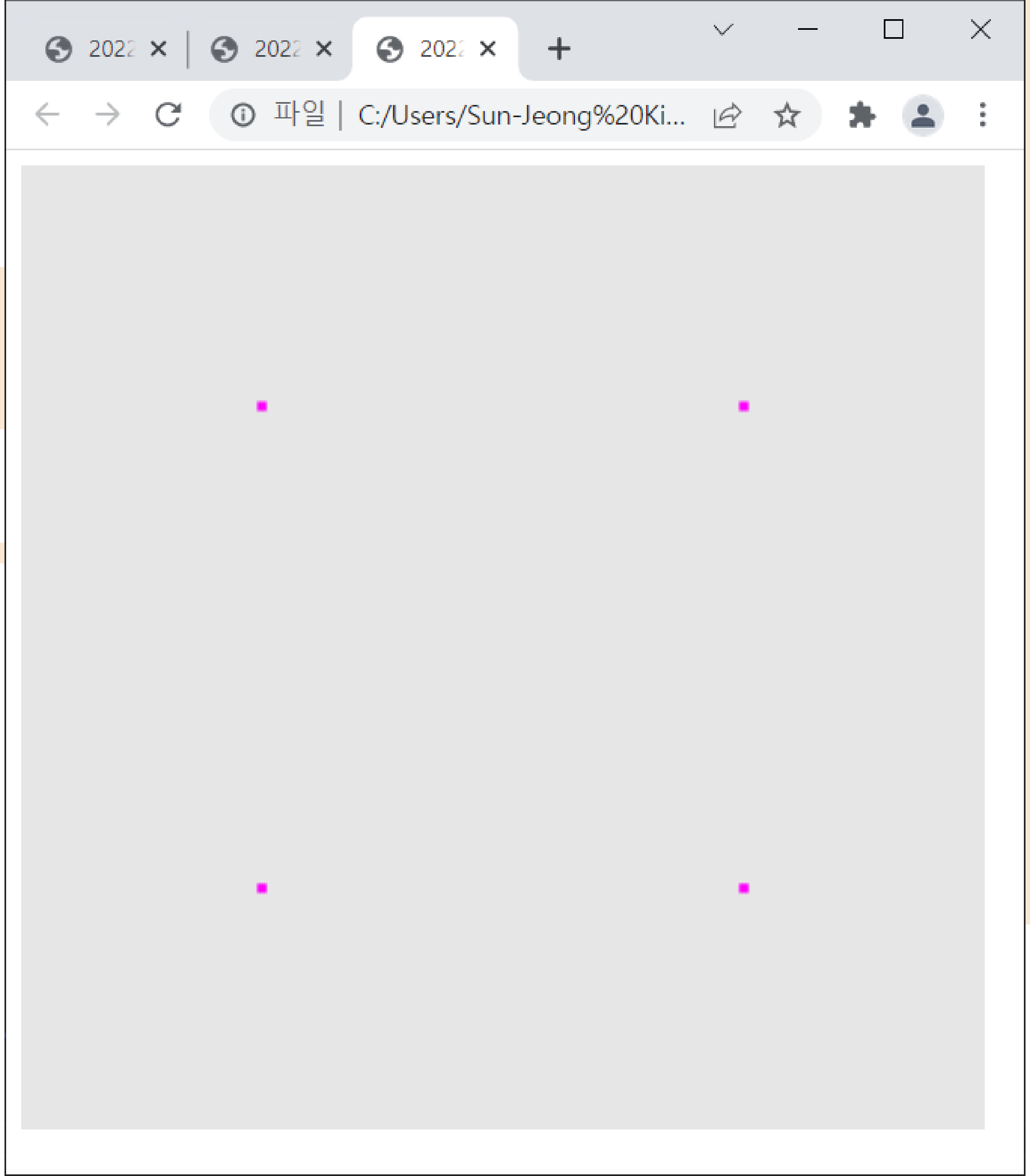

File Edit Selection View Go Run Terminal Help
star.html - Visual Studio Code
triangle.html triangle.js square.html square.js star.html star.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > star.html > html > head > script
Restricted Mode
Ln 26, Col 49 Spaces: 4 UTF-8 CRLF HTML

```html
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics</title>

        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;

        void main() {
            gl_PointSize = 5.0;
            gl_Position = vPosition;
        }
        </script>

        <script id="fragment-shader" type="x-shader/x-fragment">
        precision mediump float;

        void main() {
            gl_FragColor = vec4(1.0, 0.0, 1.0, 1.0);
        }
        </script>

        <script type="text/javascript" src="../Common/webgl-utils.js"></script>
        <script type="text/javascript" src="../Common/initShaders.js"></script>
        <script type="text/javascript" src="../Common/MV.js"></script>
        <script type="text/javascript" src="star.js"></script>
    </head>
    <body>
        <canvas id="gl-canvas" width="512" height="512">
            Oops... your browser doesn't support the HTML5 canvas element!
        </canvas>
    </body>
</html>
```



File Edit Selection View Go Run Terminal Help
star.js - Visual Studio Code
triangle.html
triangle.js
square.html
square.js
star.html
star.js
C: > Users > Sun-Jeong Kim > Desktop > CG > Week02 > star.js > render
Restricted Mode
Ln 43, Col 28
Spaces: 4
UTF-8
CRLF
JavaScript

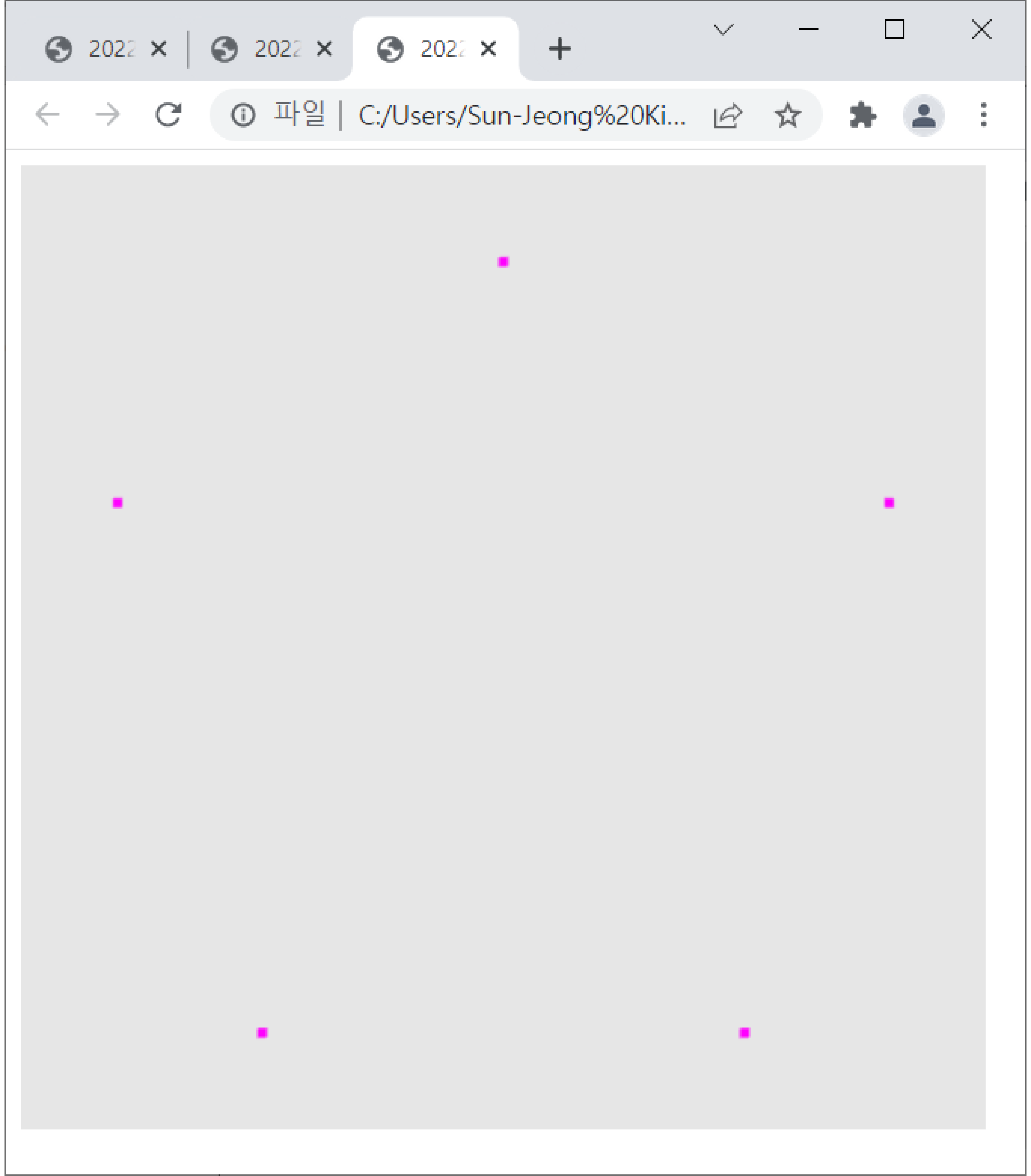
```
    var vertices = [
        vec2(-0.5, -0.5),
        vec2(-0.5, 0.5),
        vec2(0.5, 0.5),
        vec2(0.5, -0.5)
    ];

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.9, 0.9, 0.9, 1.0);

    // Load shaders and initialize attribute buffers
    var program = initShaders(gl, "vertex-shader", "fragment-shader");
    gl.useProgram(program);

    // Load the data into the GPU
    var bufferId = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
    gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);

    // Associate our shader variables with our data buffer
    var vPosition = gl.getAttribLocation(program, "vPosition");
    gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
    gl.enableVertexAttribArray(vPosition);

    render();
};

function render()
{
    gl.clear(gl.COLOR_BUFFER_BIT);
    gl.drawArrays(gl.POINTS, 0, 4);
}
```



2022
2022
2022
파일 | C:/Users/Sun-Jeong%20Ki...

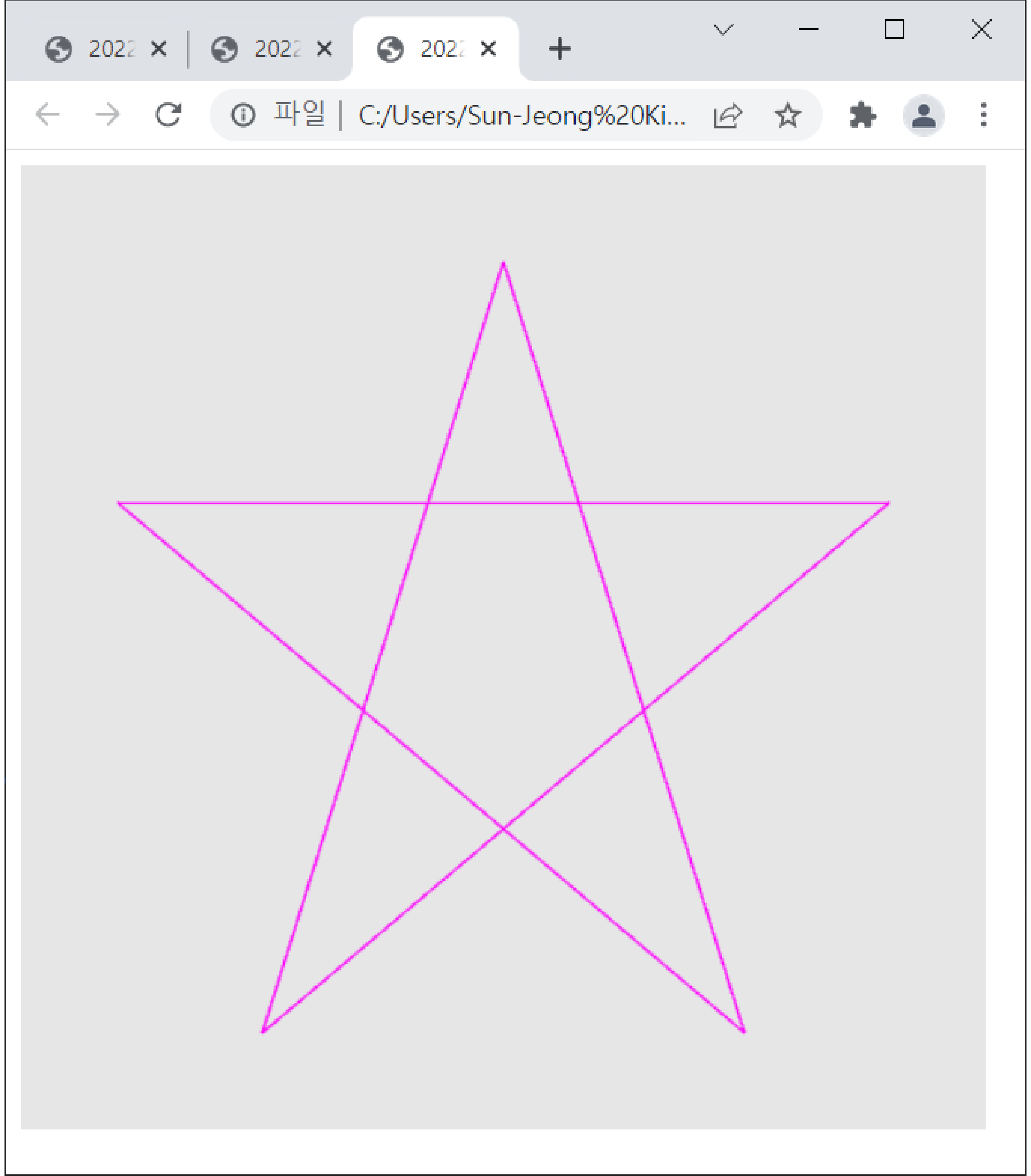
연습문제 (5)

- 점 5개를 찍으시오.



연습문제 (6)

- 선으로 연결하여 별을 그리시오.



2022
2022
2022
파일 | C:/Users/Sun-Jeong%20Ki...