

# Graphics Programming

2<sup>ND</sup> WEEK, 2021

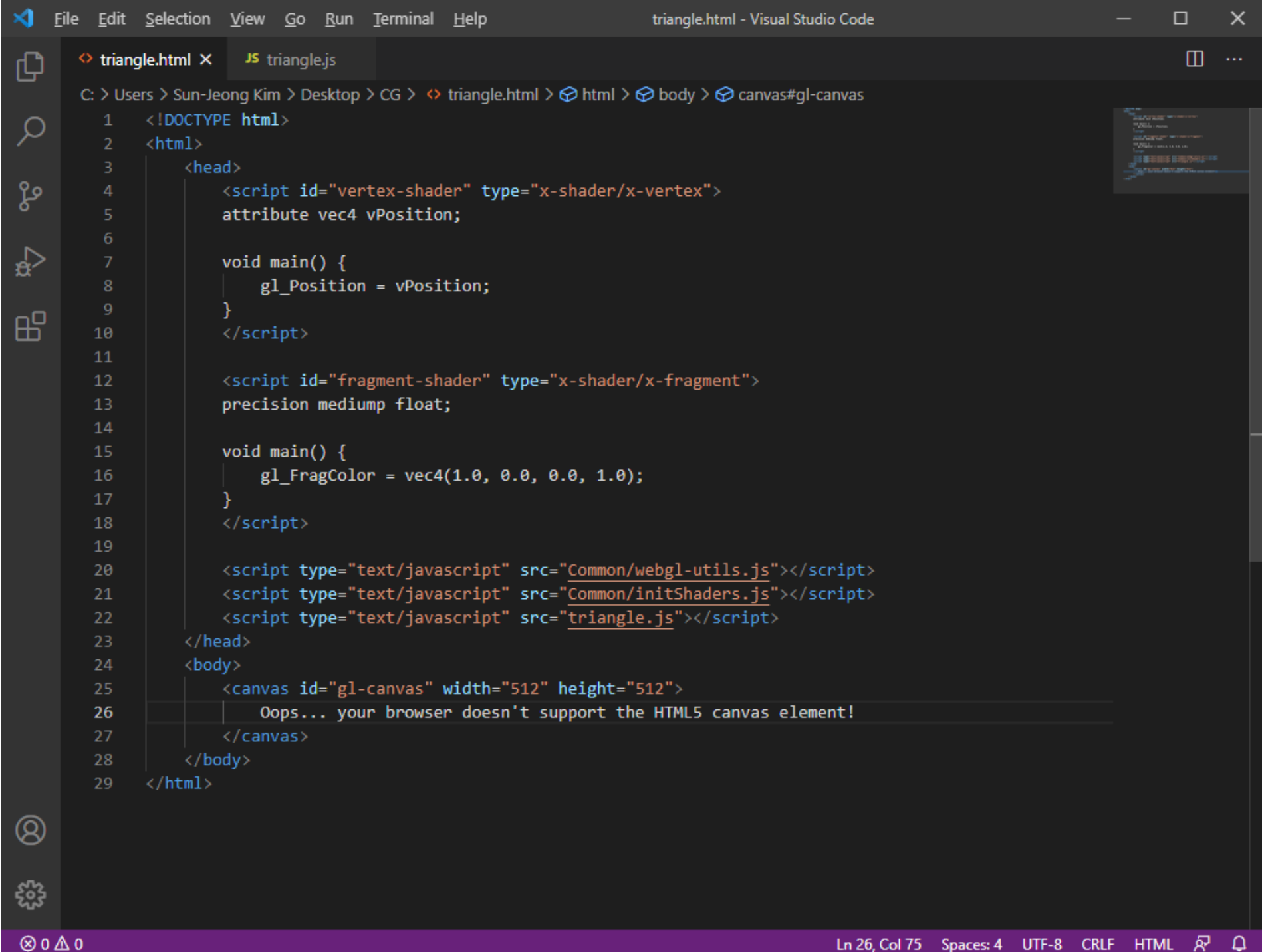


# Example: Draw a Triangle

- 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.js - Visual Studio Code

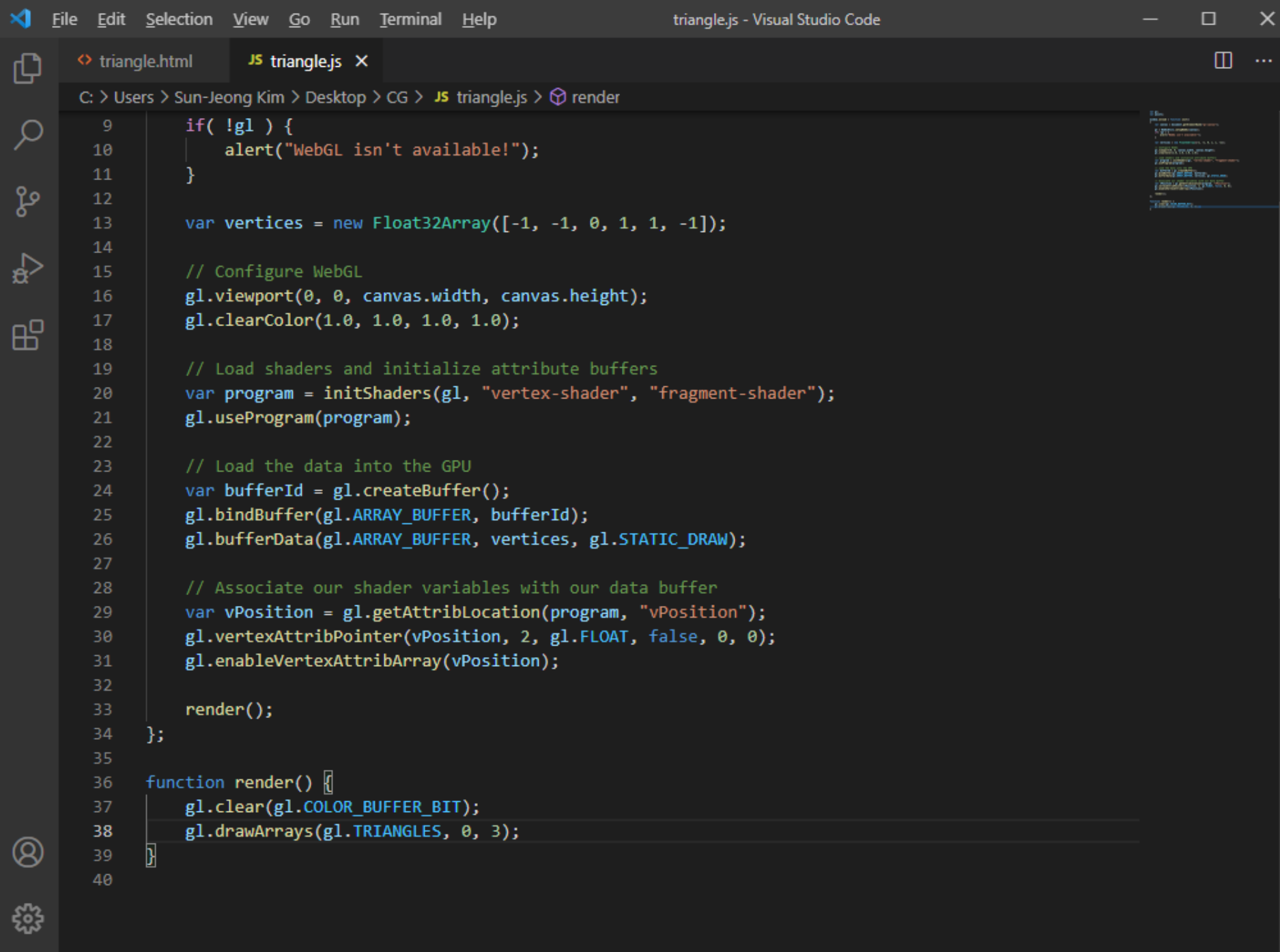
File Edit Selection View Go Run Terminal Help

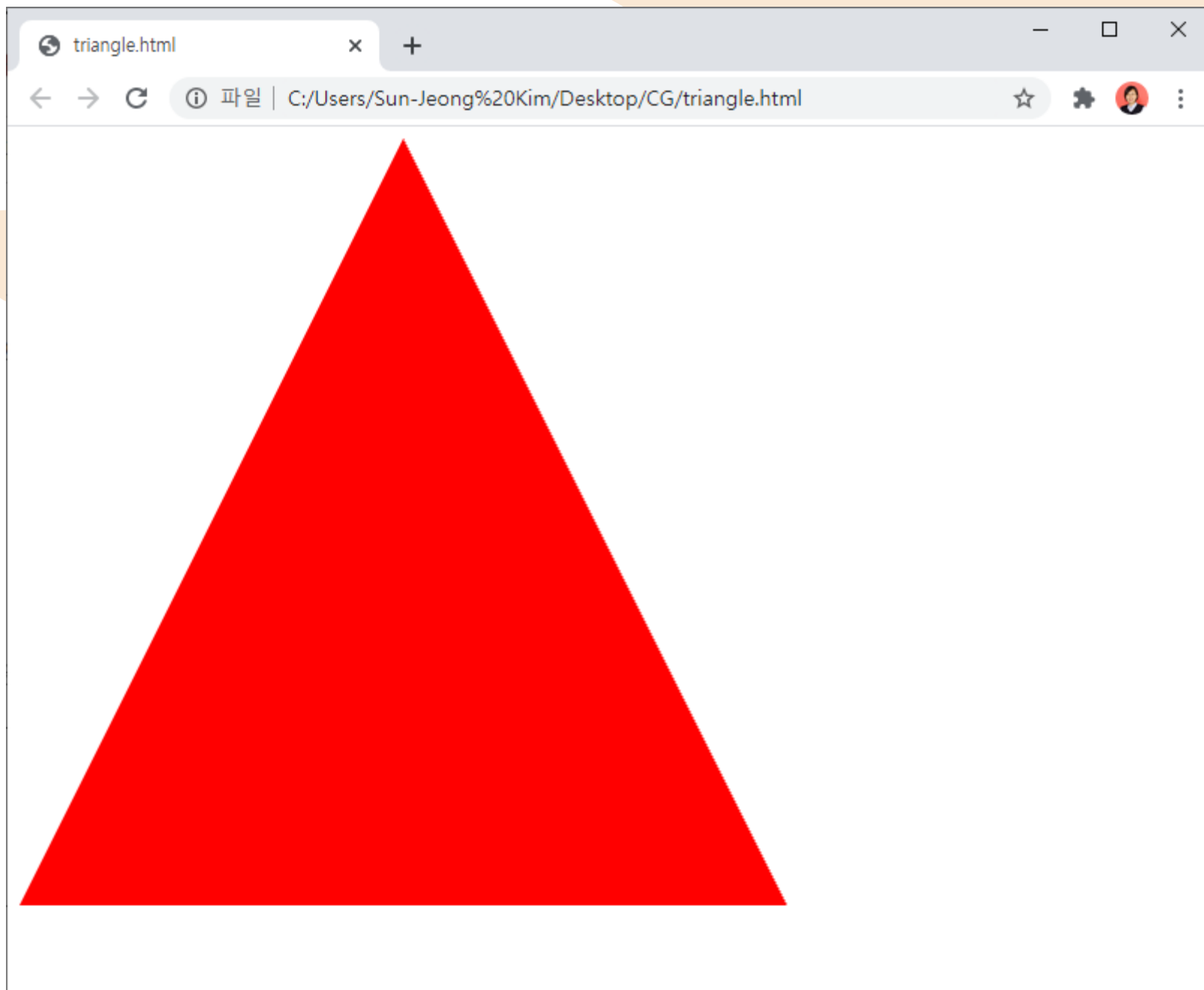
triangle.html JS triangle.js X

C: > Users > Sun-Jeong Kim > Desktop > CG > JS triangle.js > render

```
1  var gl;
2  var points;
3
4  window.onload = function init()
5  {
6      var canvas = document.getElementById("gl-canvas");
7
8      gl = WebGLUtils.setupWebGL(canvas);
9      if( !gl ) {
10         alert("WebGL isn't available!");
11     }
12
13     var vertices = new Float32Array([-1, -1, 0, 1, 1, -1]);
14
15     // Configure WebGL
16     gl.viewport(0, 0, canvas.width, canvas.height);
17     gl.clearColor(1.0, 1.0, 1.0, 1.0);
18
19     // Load shaders and initialize attribute buffers
20     var program = initShaders(gl, "vertex-shader", "fragment-shader");
21     gl.useProgram(program);
22
23     // Load the data into the GPU
24     var bufferId = gl.createBuffer();
25     gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
26     gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
27
28     // Associate our shader variables with our data buffer
29     var vPosition = gl.getAttribLocation(program, "vPosition");
30     gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
31     gl.enableVertexAttribArray(vPosition);
32
33     render();
34 };
35
```

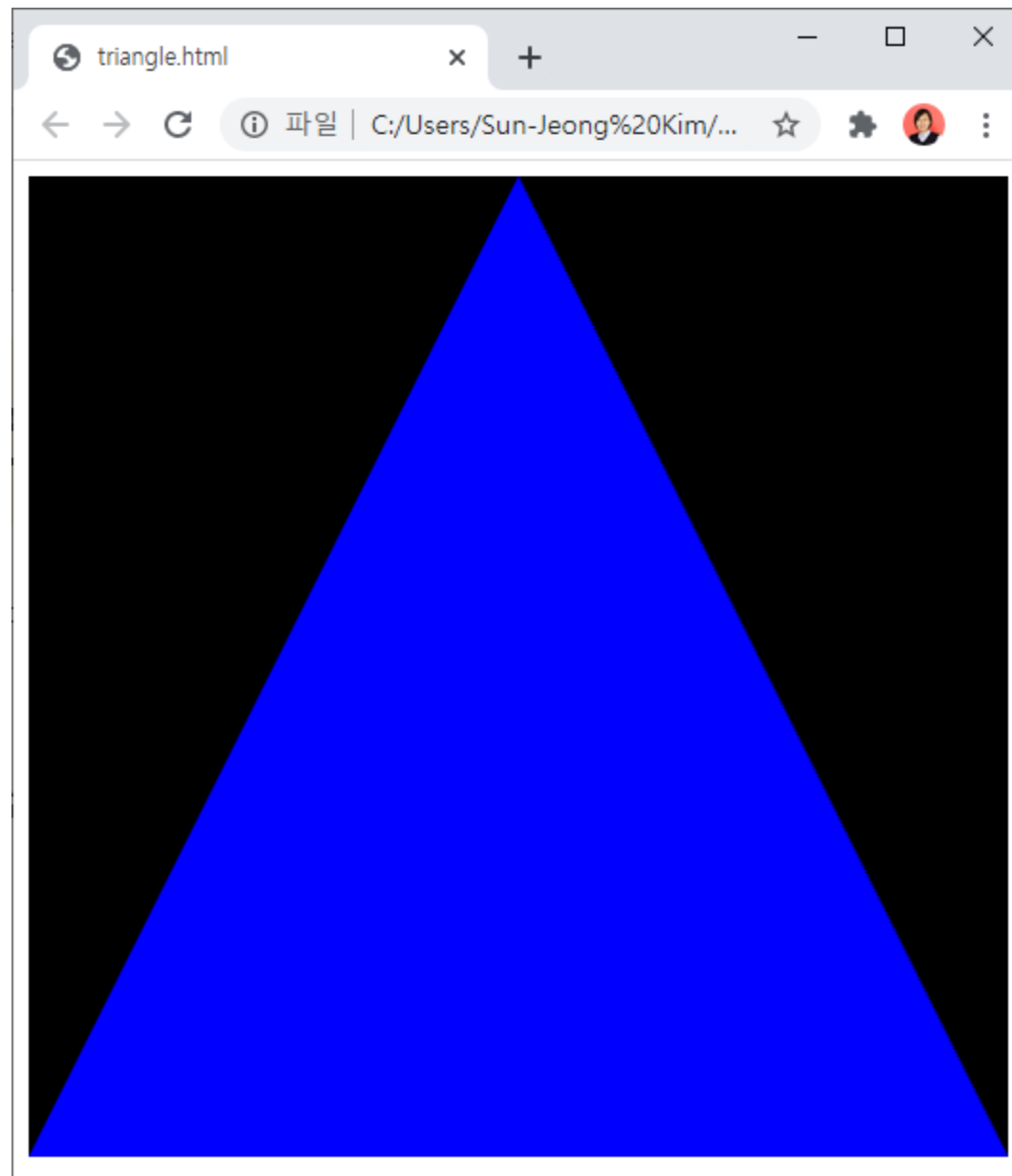
Ln 38, Col 39 Spaces: 4 UTF-8 CRLF JavaScript





# 연습 문제 (1)

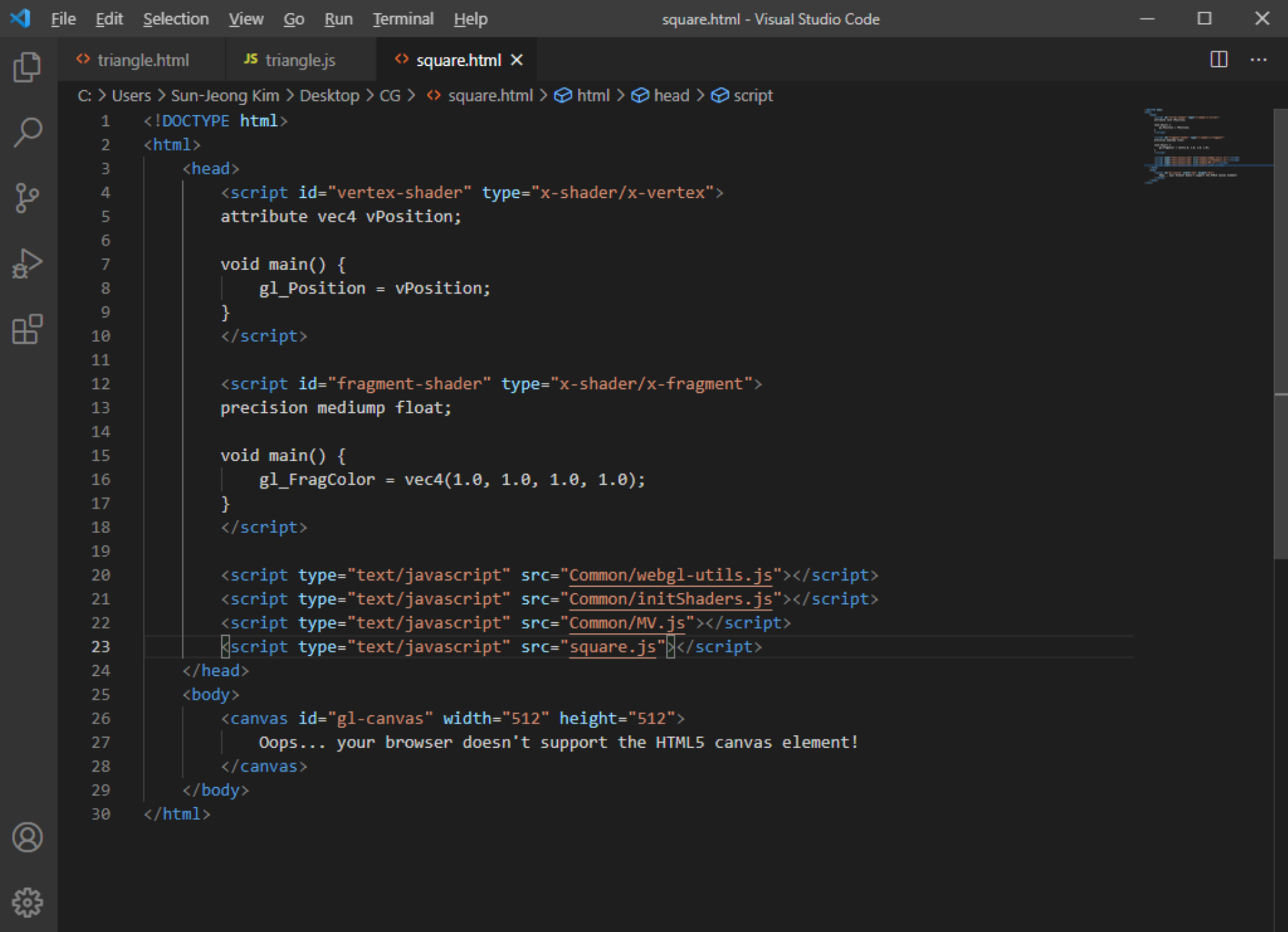
- 배경색을 변경해보시오.
  - 예) 흰색 → 검정색
- 삼각형의 색상을 변경해보시오.
  - 예) 빨강 → 파랑색





# Example: Draw a Square

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



# 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

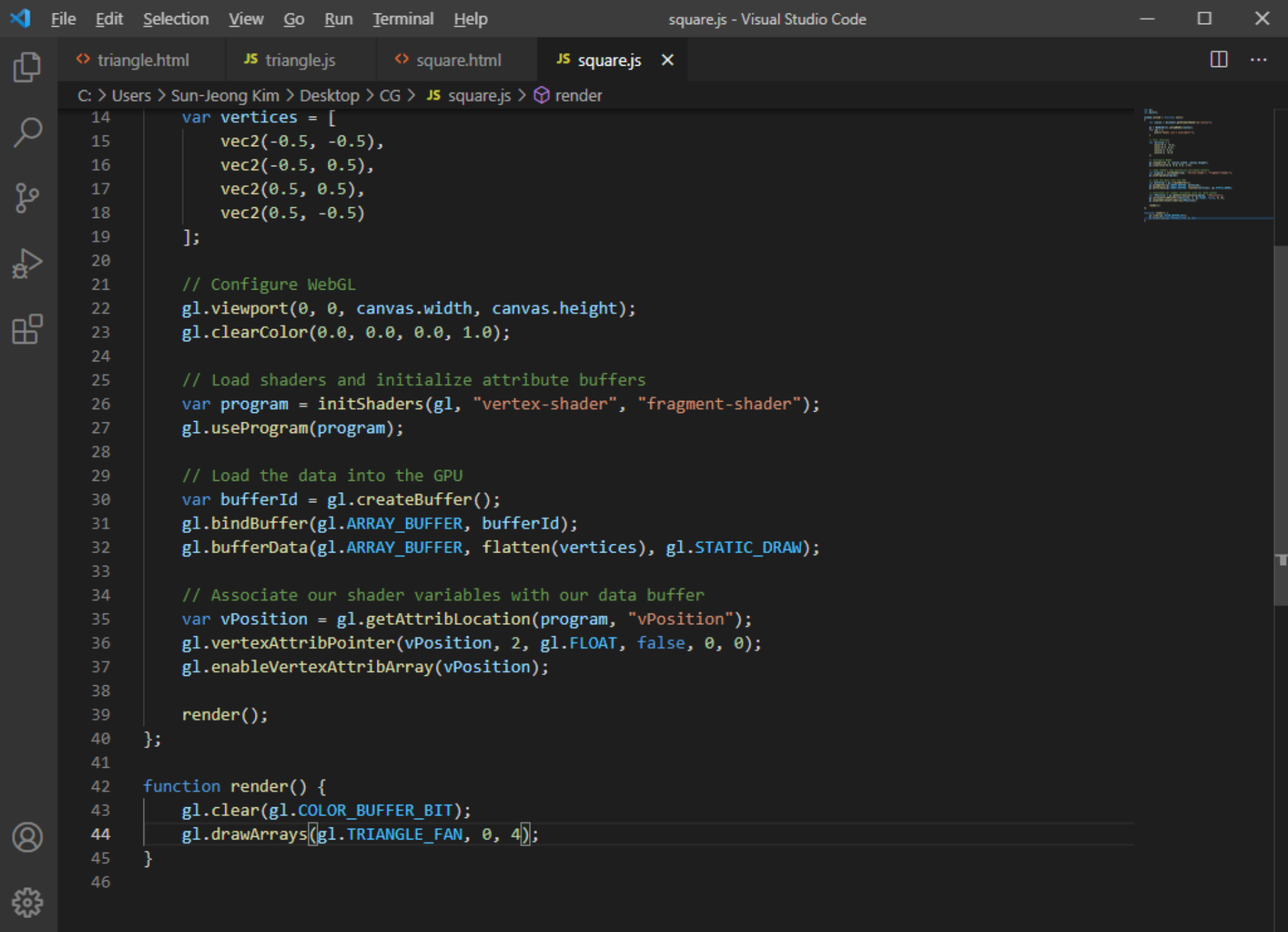
File Edit Selection View Go Run Terminal Helpsquare.js - Visual Studio Code

triangle.htmlJS triangle.js<> square.htmlJS square.js X

C: > Users > Sun-Jeong Kim > Desktop > CG > JS square.js > render

```
1  var gl;
2  var points;
3
4  window.onload = function init()
5  {
6      var canvas = document.getElementById("gl-canvas");
7
8      gl = WebGLUtils.setupWebGL(canvas);
9      if( !gl ) {
10         alert("WebGL isn't available!");
11     }
12
13     // Four vertices
14     var vertices = [
15         vec2(-0.5, -0.5),
16         vec2(-0.5, 0.5),
17         vec2(0.5, 0.5),
18         vec2(0.5, -0.5)
19     ];
20
21     // Configure WebGL
22     gl.viewport(0, 0, canvas.width, canvas.height);
23     gl.clearColor(0.0, 0.0, 0.0, 1.0);
24
25     // Load shaders and initialize attribute buffers
26     var program = initShaders(gl, "vertex-shader", "fragment-shader");
27     gl.useProgram(program);
28
29     // Load the data into the GPU
30     var bufferId = gl.createBuffer();
31     gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
32     gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
33
34     // Associate our shader variables with our data buffer
35     var vPosition = gl.getAttribLocation(program, "vPosition");
```

0 0Ln 44, Col 40Spaces: 4UTF-8CRLFJavaScript



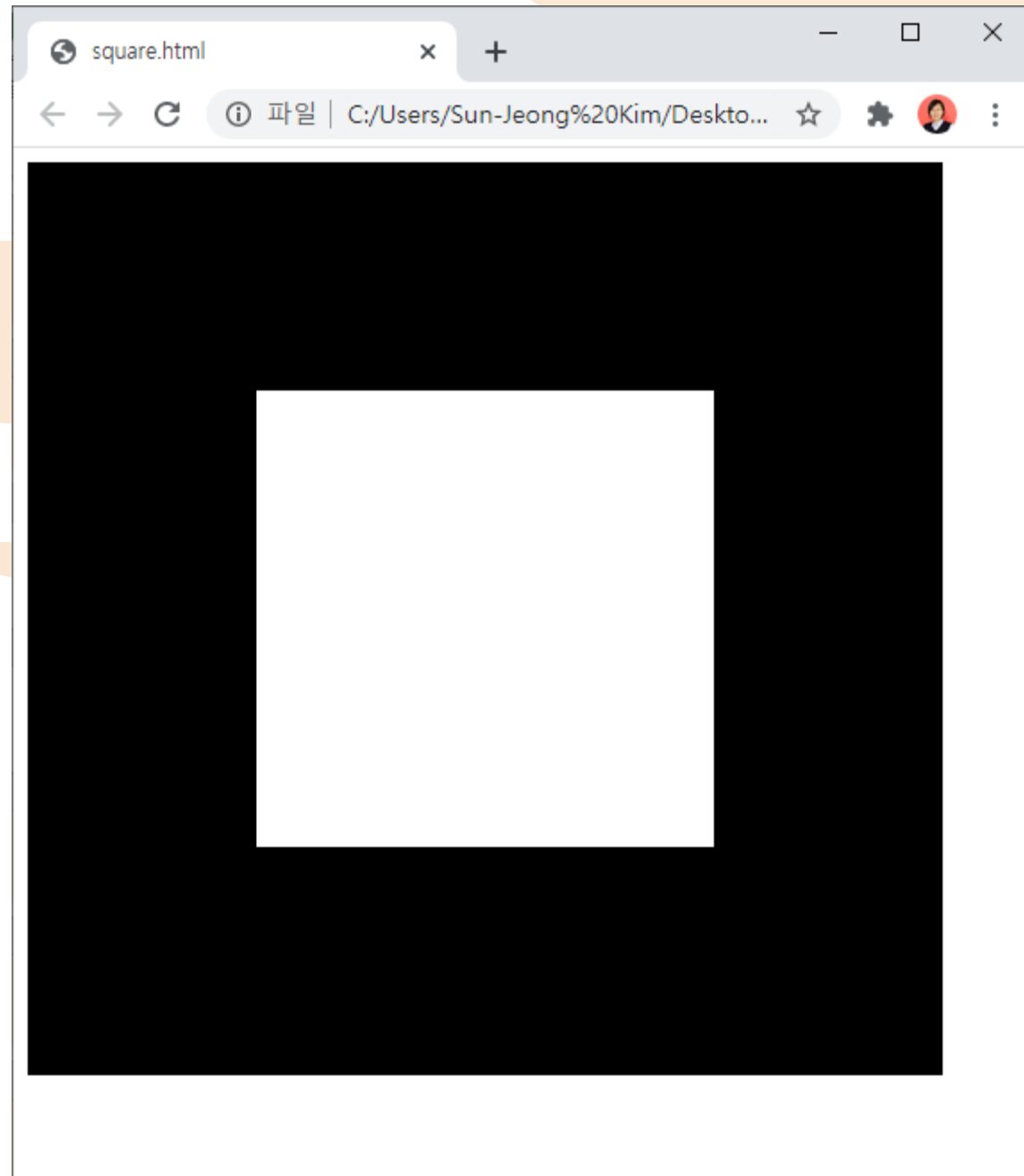
# Notes

- `onload`
  - Determines where to start execution when all code is loaded
- `canvas` gets WebGL context from 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

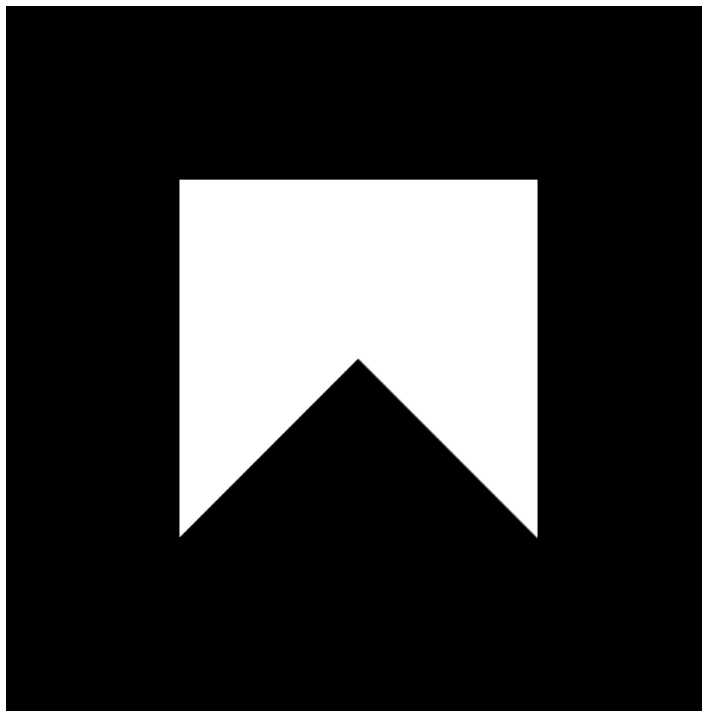
- `initShaders` used to load, compile and link shaders to form a program object
- Load data onto GPU by creating vertex buffer object on the GPU
  - Note use of `flatten()` to convert JS array to an array of float32's
- Finally we must connect variable in program with variable in shader
  - Need name, type, location in buffer



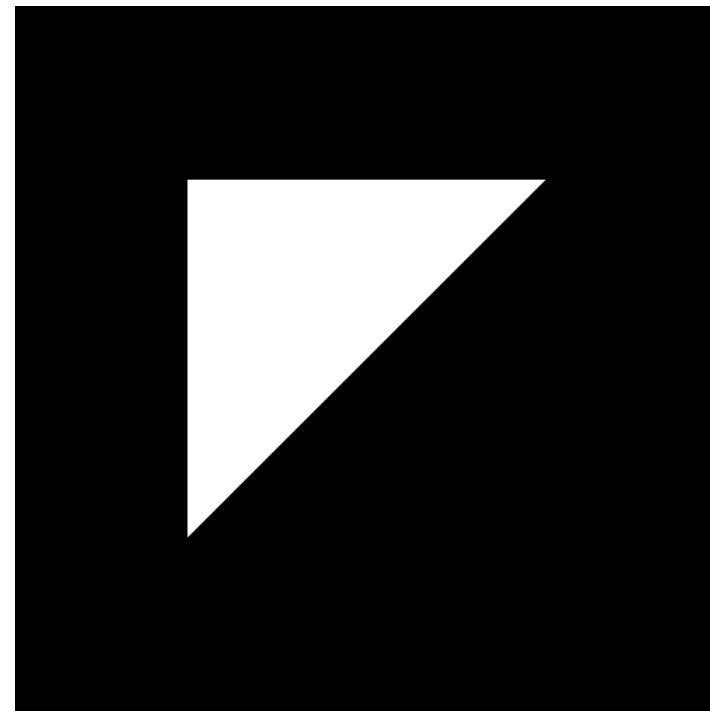


## 연습 문제 (2)

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



(a)



(b)

# Program Execution

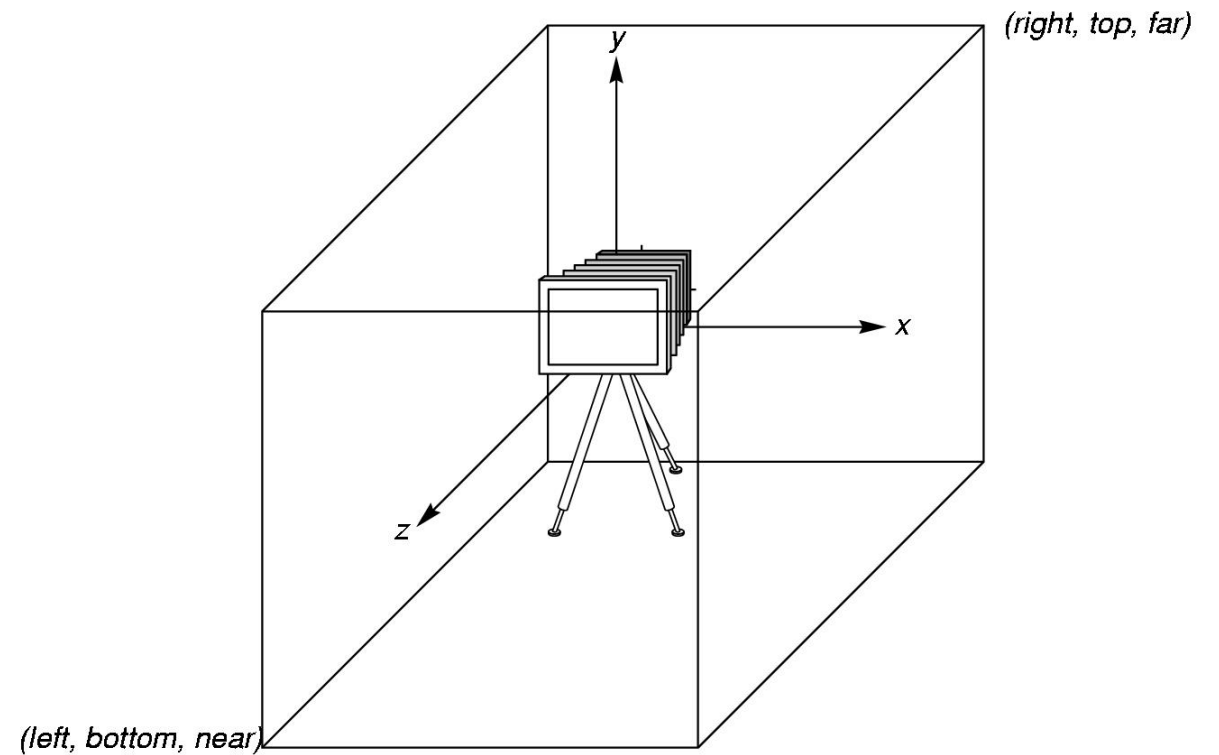
- WebGL runs within the browser
  - Complex interaction among operating system, the window system, the browser and your code (HTML and JS)
- Simple model
  - Start with HTML file
  - Files read in asynchronously
  - Start with **onload** function
    - Event driven input

# 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

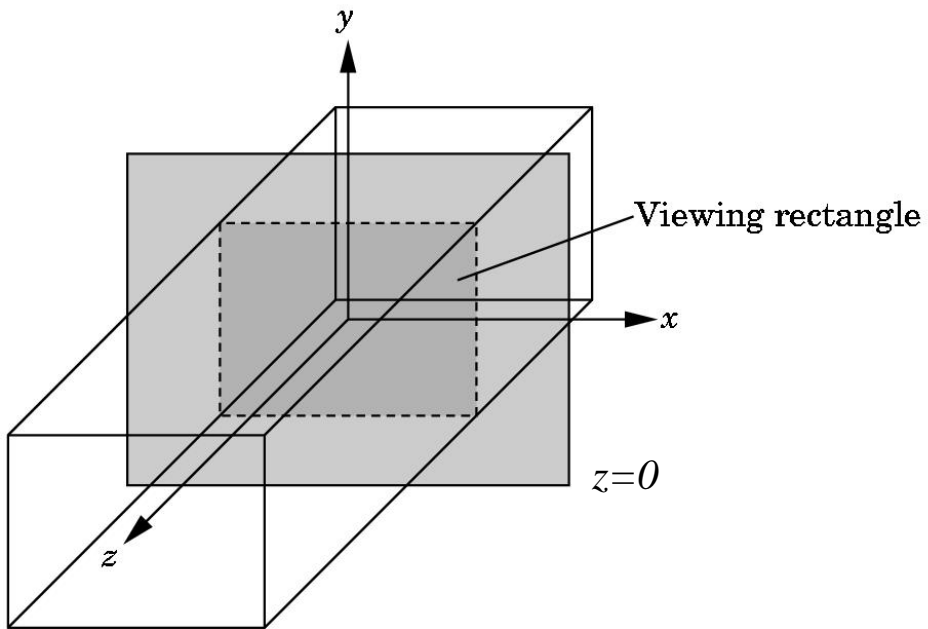
- WebGL places a camera at the origin in world space pointing in the negative z direction
  - Default **view volume** – a box centered at the origin with a side of length 2



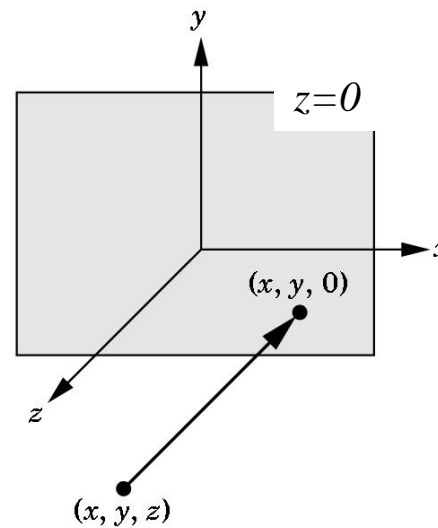
The default camera and an orthographic view volume

# Orthographic Viewing

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



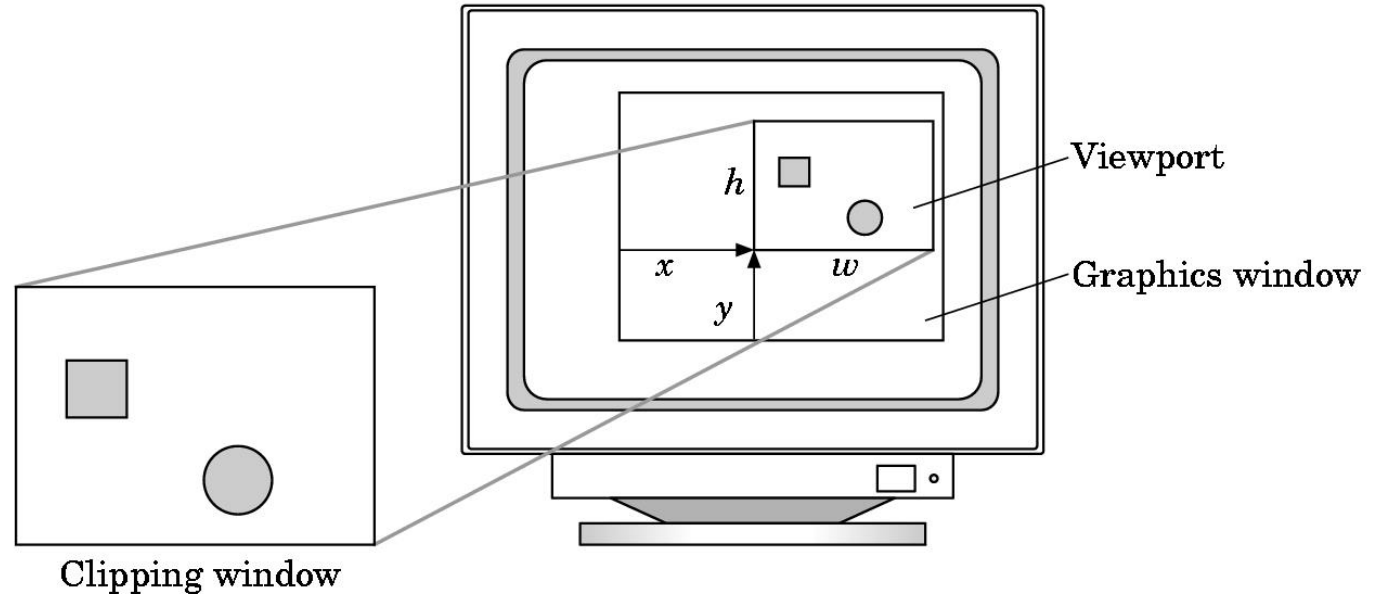
View Volume



Orthographic Projection

# Viewport

- **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

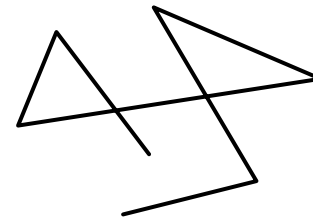
- Points
- Lines
- Triangles



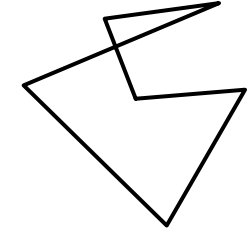
GL\_POINTS



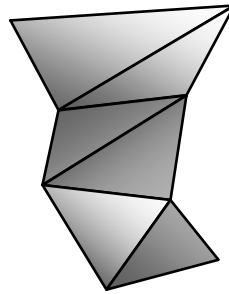
GL\_LINES



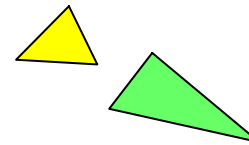
GL\_LINE\_STRIP



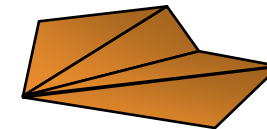
GL\_LINE\_LOOP



GL\_TRIANGLE\_STRIP



GL\_TRIANGLES



GL\_TRIANGLE\_FAN

File Edit Selection View Go Run Terminal Help

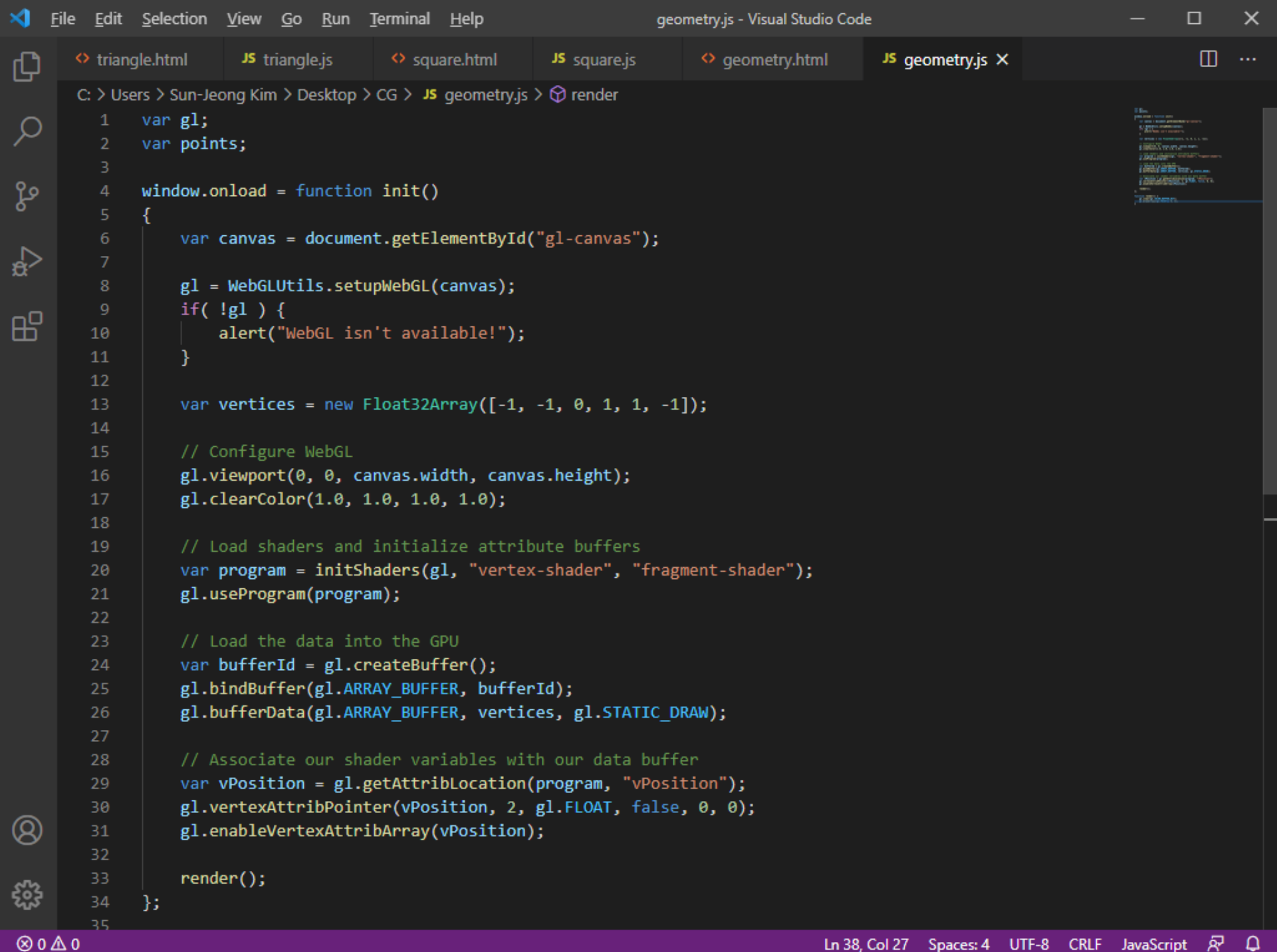
triangle.html JS triangle.js square.html JS square.js geometry.html X

C: > Users > Sun-Jeong Kim > Desktop > CG > geometry.html > html > head > script

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script id="vertex-shader" type="x-shader/x-vertex">
5       attribute vec4 vPosition;
6
7       void main() {
8         gl_PointSize = 5.0;
9         gl_Position = vPosition;
10      }
11    </script>
12
13    <script id="fragment-shader" type="x-shader/x-fragment">
14      precision mediump float;
15
16      void main() {
17        gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
18      }
19    </script>
20
21    <script type="text/javascript" src="Common/webgl-utils.js"></script>
22    <script type="text/javascript" src="Common/initShaders.js"></script>
23    <script type="text/javascript" src="Common/MV.js"></script>
24    <script type="text/javascript" src="geometry.js"></script>
25  </head>
26  <body>
27    <canvas id="gl-canvas" width="512" height="512">
28      Oops... your browser doesn't support the HTML5 canvas element!
29    </canvas>
30  </body>
31 </html>
```

0 0

Ln 24, Col 57 Spaces: 4 UTF-8 CRLF HTML



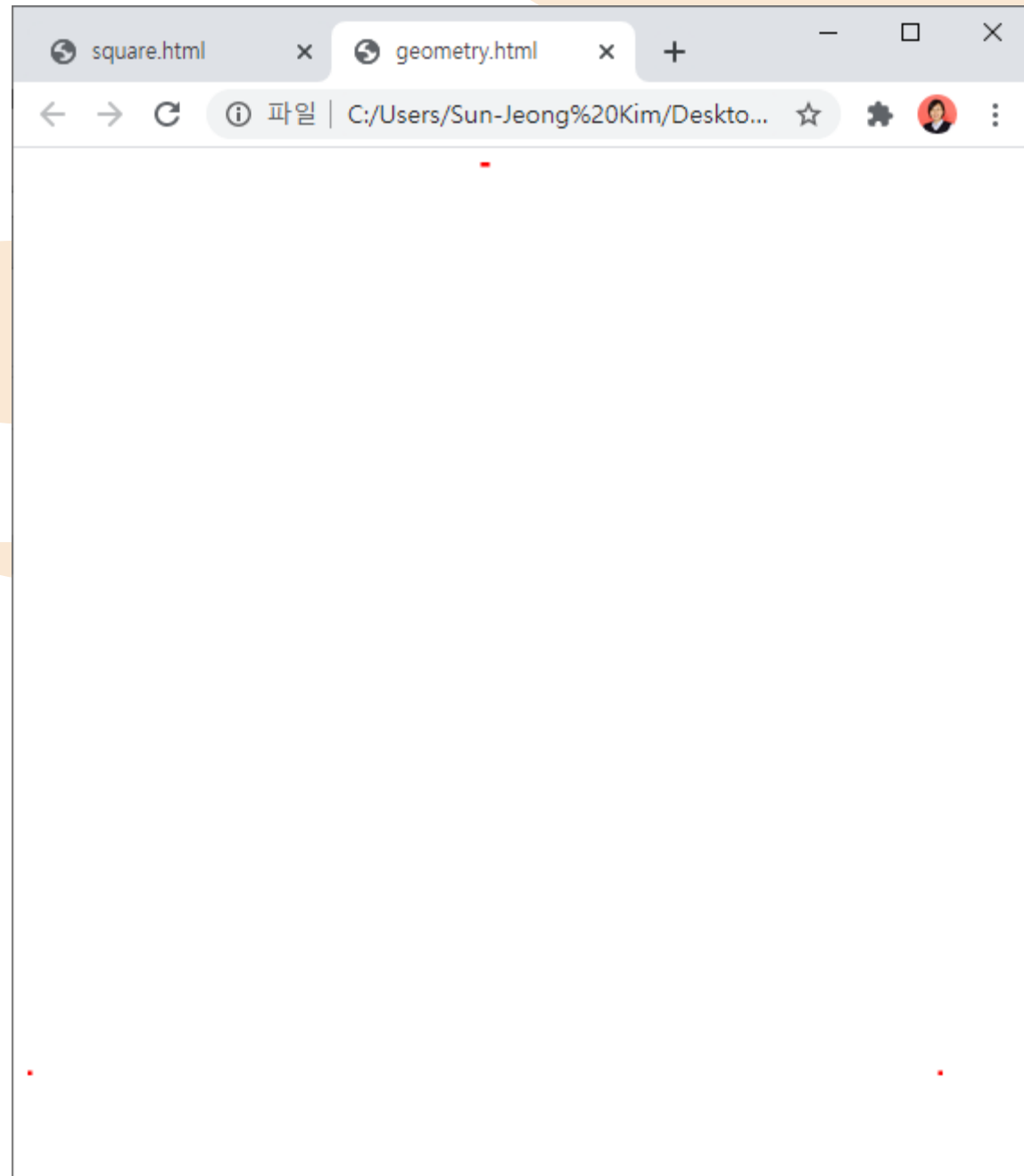
File Edit Selection View Go Run Terminal Help

geometry.js - Visual Studio Code

triangle.html JS triangle.js square.html JS square.js geometry.html JS geometry.js X

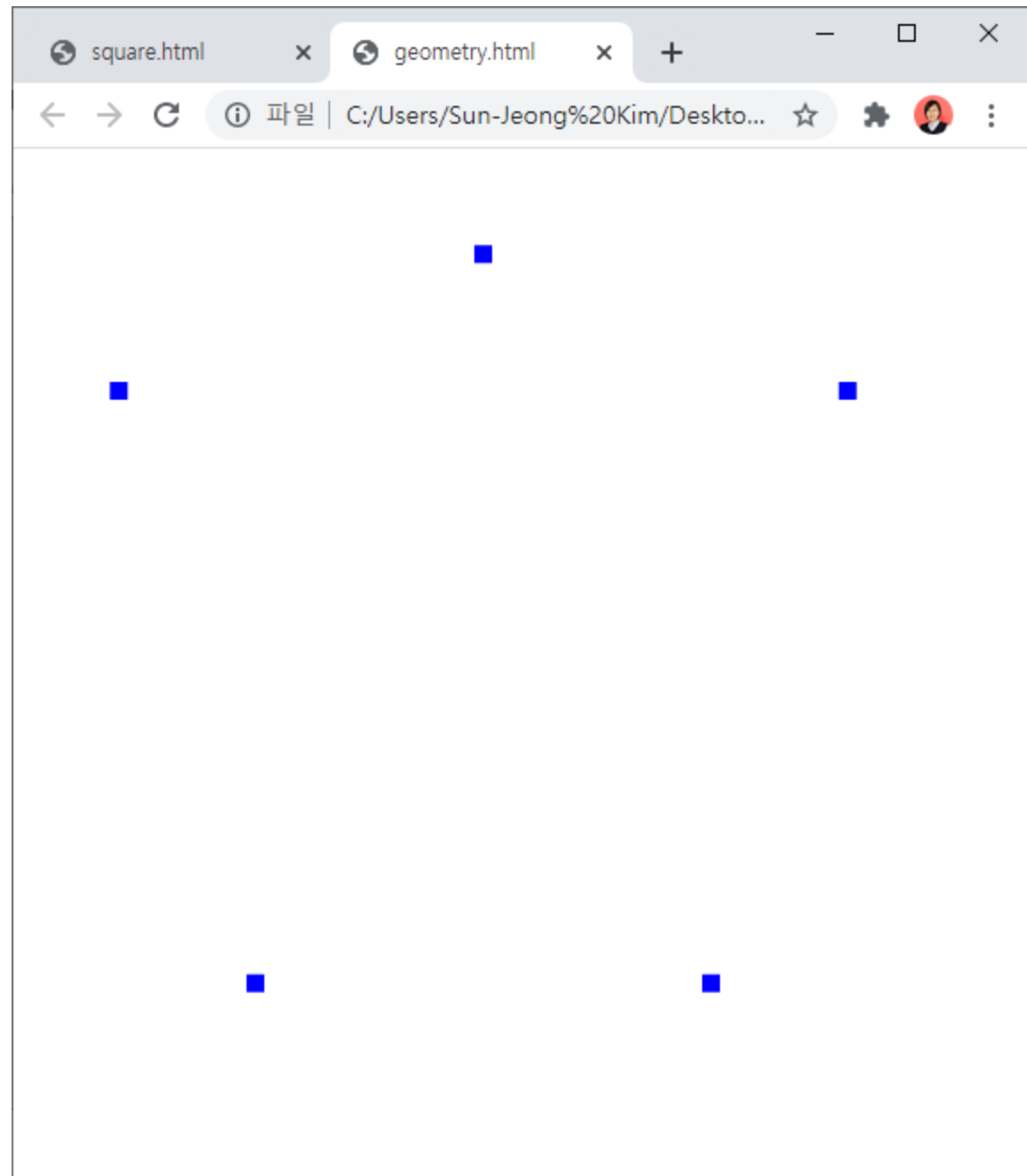
C: > Users > Sun-Jeong Kim > Desktop > CG > JS geometry.js > render

```
6   var canvas = document.getElementById("gl-canvas");
7
8   gl = WebGLUtils.setupWebGL(canvas);
9   if( !gl ) {
10     alert("WebGL isn't available!");
11   }
12
13   var vertices = new Float32Array([-1, -1, 0, 1, 1, -1]);
14
15   // Configure WebGL
16   gl.viewport(0, 0, canvas.width, canvas.height);
17   gl.clearColor(1.0, 1.0, 1.0, 1.0);
18
19   // Load shaders and initialize attribute buffers
20   var program = initShaders(gl, "vertex-shader", "fragment-shader");
21   gl.useProgram(program);
22
23   // Load the data into the GPU
24   var bufferId = gl.createBuffer();
25   gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
26   gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
27
28   // Associate our shader variables with our data buffer
29   var vPosition = gl.getAttribLocation(program, "vPosition");
30   gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
31   gl.enableVertexAttribArray(vPosition);
32
33   render();
34 };
35
36 function render() {
37   gl.clear(gl.COLOR_BUFFER_BIT);
38   gl.drawArrays(gl.POINTS, 0, 3);
39 }
40
```



## 연습 문제 (3)

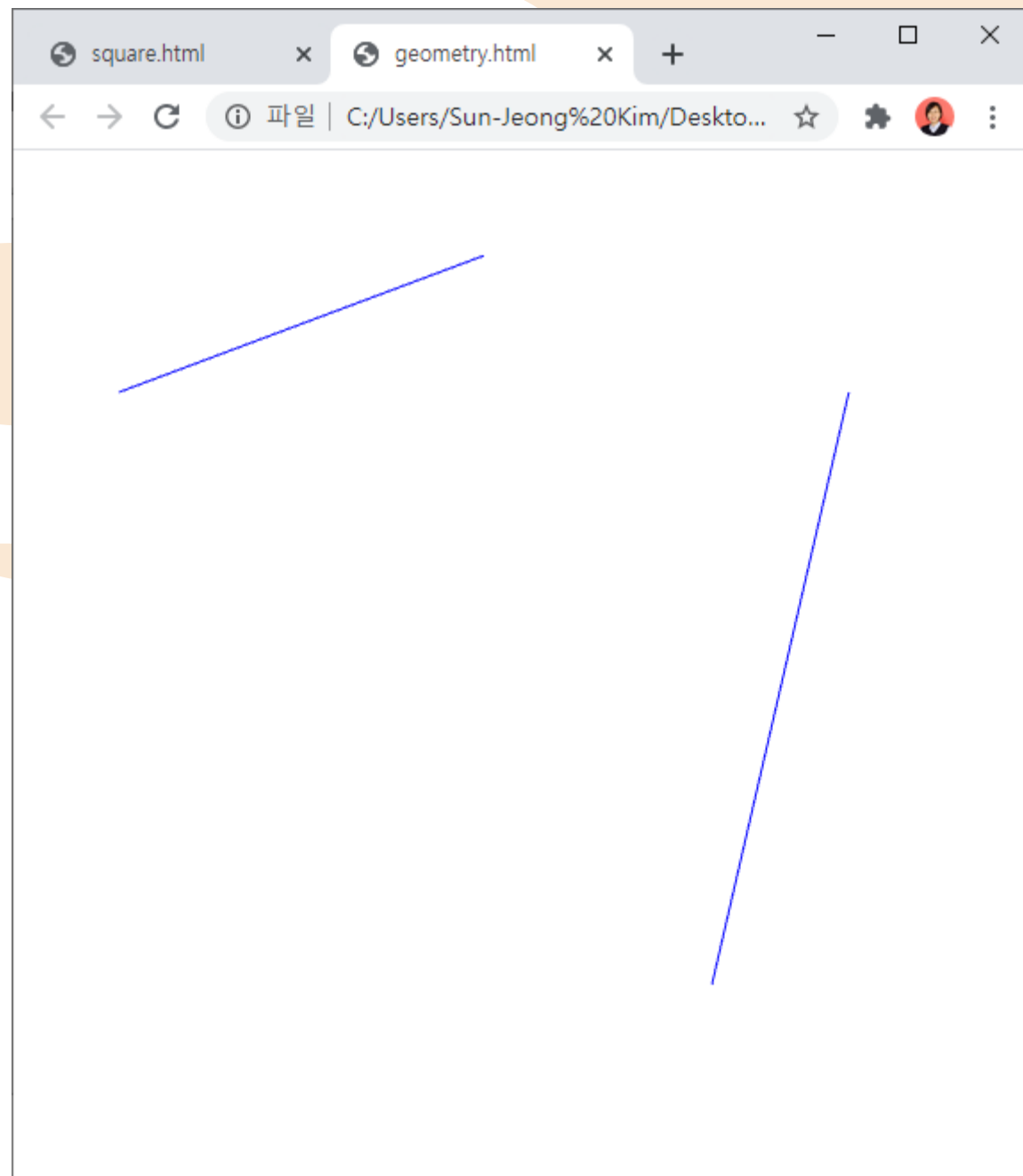
- Point의 크기와 색상을 변경하여 5개를 찍어 보시오.



```
File Edit Selection View Go Run Terminal Help
geometry.js - Visual Studio Code

triangle.html JS triangle.js square.html JS square.js geometry.html JS geometry.js X

C: > Users > Sun-Jeong Kim > Desktop > CG > JS geometry.js > render
6   var canvas = document.getElementById("gl-canvas");
7
8   gl = WebGLUtils.setupWebGL(canvas);
9   if( !gl ) {
10      alert("WebGL isn't available!");
11   }
12
13   var vertices = new Float32Array([-0.8, 0.5, 0, 0.8, 0.8, 0.5, 0.5, -0.8, -0.5, -0.8]);
14
15   // Configure WebGL
16   gl.viewport(0, 0, canvas.width, canvas.height);
17   gl.clearColor(1.0, 1.0, 1.0, 1.0);
18
19   // Load shaders and initialize attribute buffers
20   var program = initShaders(gl, "vertex-shader", "fragment-shader");
21   gl.useProgram(program);
22
23   // Load the data into the GPU
24   var bufferId = gl.createBuffer();
25   gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
26   gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
27
28   // Associate our shader variables with our data buffer
29   var vPosition = gl.getAttribLocation(program, "vPosition");
30   gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
31   gl.enableVertexAttribArray(vPosition);
32
33   render();
34 };
35
36 function render() {
37     gl.clear(gl.COLOR_BUFFER_BIT);
38     gl.drawArrays(gl.LINES, 0, 5);
39 }
40
```





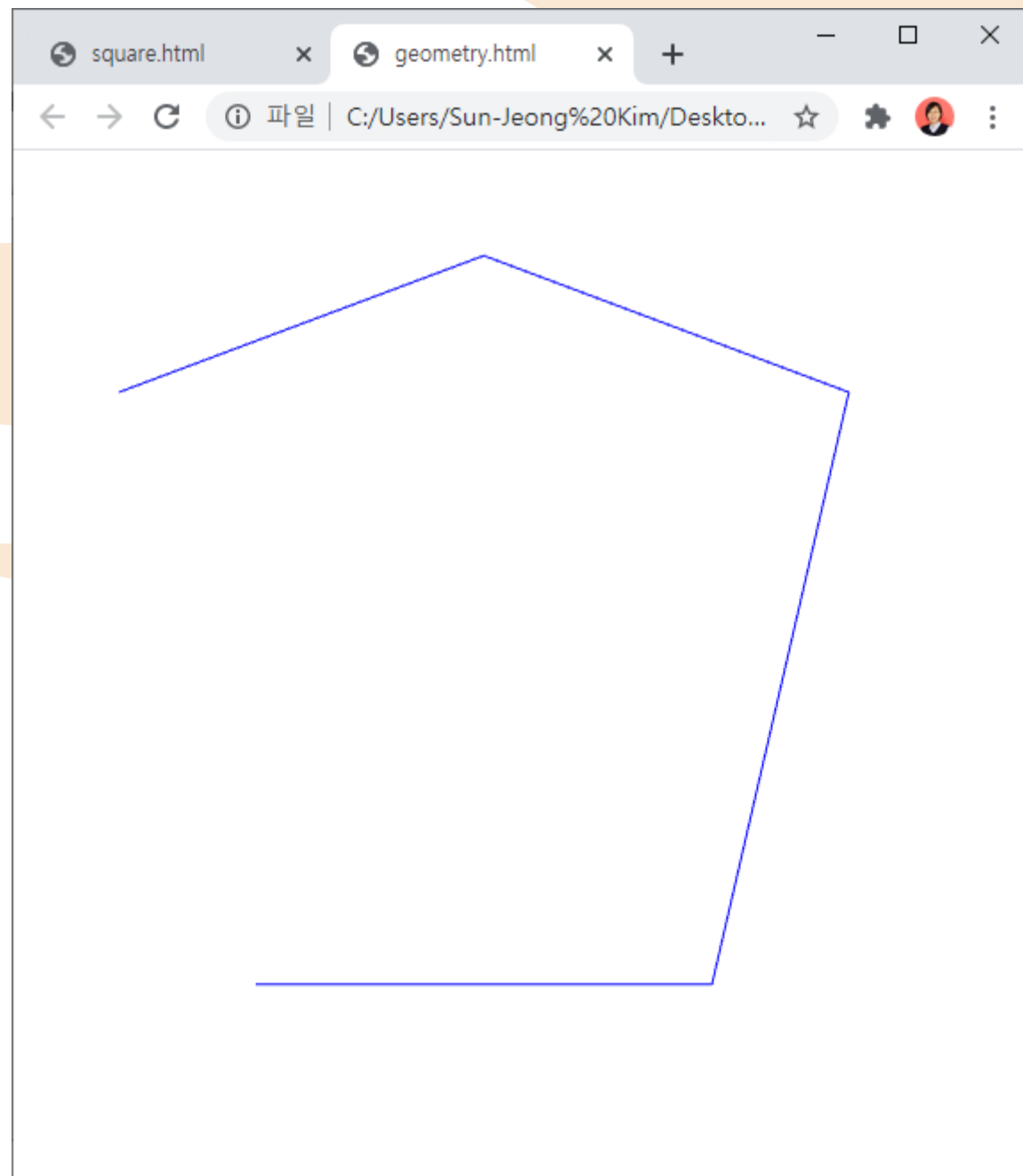
File Edit Selection View Go Run Terminal Help

geometry.js - Visual Studio Code

triangle.html JS triangle.js square.html JS square.js geometry.html JS geometry.js X

C: > Users > Sun-Jeong Kim > Desktop > CG > JS geometry.js > render

```
6   var canvas = document.getElementById("gl-canvas");
7
8   gl = WebGLUtils.setupWebGL(canvas);
9   if( !gl ) {
10     alert("WebGL isn't available!");
11   }
12
13   var vertices = new Float32Array([-0.8, 0.5, 0, 0.8, 0.8, 0.5, 0.5, -0.8, -0.5, -0.8]);
14
15   // Configure WebGL
16   gl.viewport(0, 0, canvas.width, canvas.height);
17   gl.clearColor(1.0, 1.0, 1.0, 1.0);
18
19   // Load shaders and initialize attribute buffers
20   var program = initShaders(gl, "vertex-shader", "fragment-shader");
21   gl.useProgram(program);
22
23   // Load the data into the GPU
24   var bufferId = gl.createBuffer();
25   gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
26   gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
27
28   // Associate our shader variables with our data buffer
29   var vPosition = gl.getAttribLocation(program, "vPosition");
30   gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
31   gl.enableVertexAttribArray(vPosition);
32
33   render();
34 };
35
36 function render() {
37   gl.clear(gl.COLOR_BUFFER_BIT);
38   gl.drawArrays(gl.LINE_STRIP, 0, 5);
39 }
40
```



File Edit Selection View Go Run Terminal Help

geometry.js - Visual Studio Code

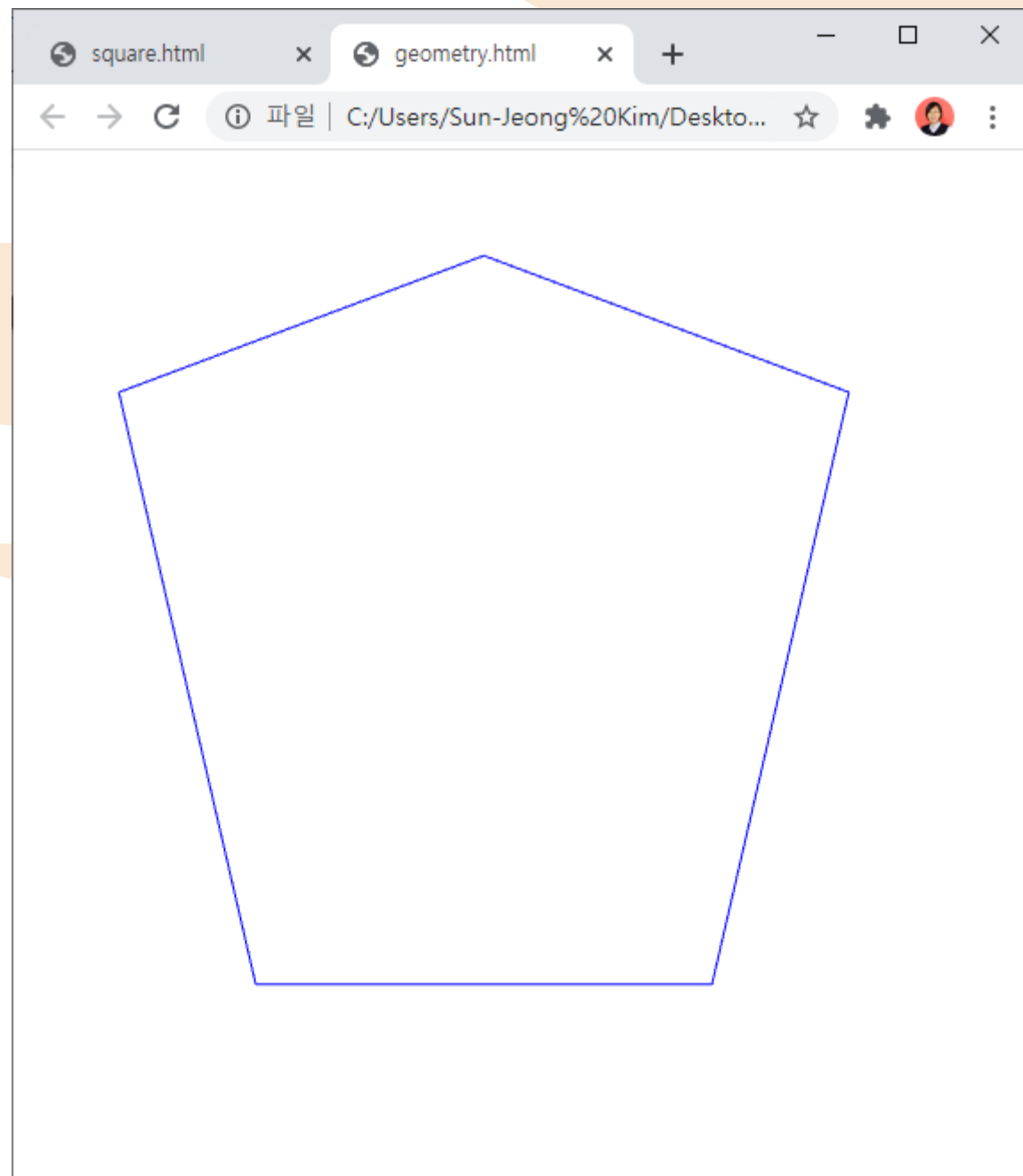
triangle.html JS triangle.js square.html JS square.js geometry.html JS geometry.js X

C: > Users > Sun-Jeong Kim > Desktop > CG > JS geometry.js > render

```
6   var canvas = document.getElementById("gl-canvas");
7
8   gl = WebGLUtils.setupWebGL(canvas);
9   if( !gl ) {
10     alert("WebGL isn't available!");
11   }
12
13   var vertices = new Float32Array([-0.8, 0.5, 0, 0.8, 0.8, 0.5, 0.5, -0.8, -0.5, -0.8]);
14
15   // Configure WebGL
16   gl.viewport(0, 0, canvas.width, canvas.height);
17   gl.clearColor(1.0, 1.0, 1.0, 1.0);
18
19   // Load shaders and initialize attribute buffers
20   var program = initShaders(gl, "vertex-shader", "fragment-shader");
21   gl.useProgram(program);
22
23   // Load the data into the GPU
24   var bufferId = gl.createBuffer();
25   gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
26   gl.bufferData(gl.ARRAY_BUFFER, vertices, gl.STATIC_DRAW);
27
28   // Associate our shader variables with our data buffer
29   var vPosition = gl.getAttribLocation(program, "vPosition");
30   gl.vertexAttribPointer(vPosition, 2, gl.FLOAT, false, 0, 0);
31   gl.enableVertexAttribArray(vPosition);
32
33   render();
34 };
35
36 function render() {
37   gl.clear(gl.COLOR_BUFFER_BIT);
38   gl.drawArrays(gl.LINE_LOOP, 0, 5);
39 }
40
```

0 0 0

Ln 38, Col 31 Spaces: 4 UTF-8 CRLF JavaScript



## 연습 문제 (4)

- 별을 그리시오.

