Graphics Programming

1ST 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

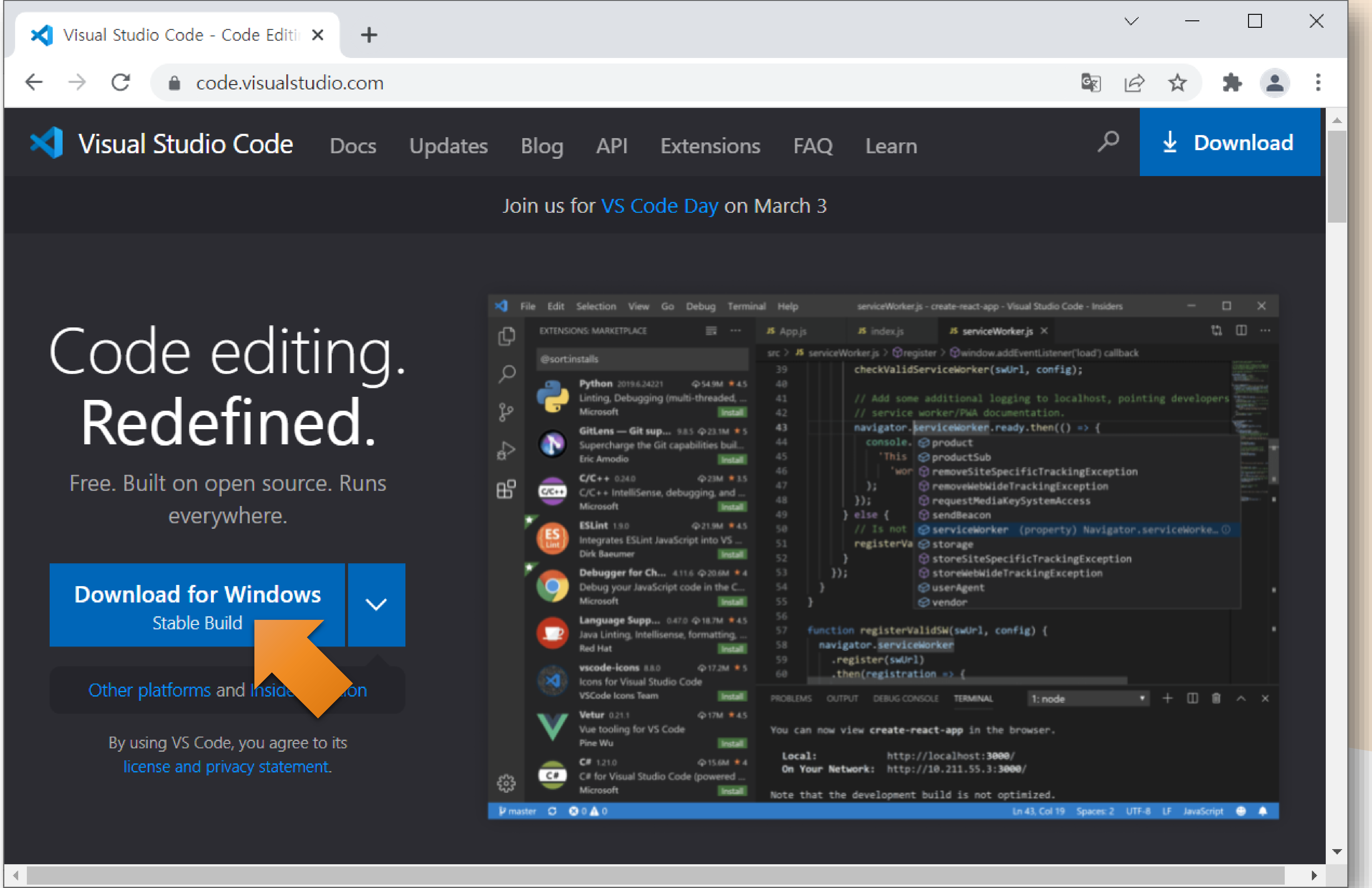
Coding in WebGL

- Can run WebGL on any recent browser
  - Chrome
  - Firefox
  - Safari
  - IE

- Code written in JavaScript

- JS runs within browser
  - Use local resources

Install Visual Studio Code



Join us for VS Code Day on March 3

Code editing. Redefined.

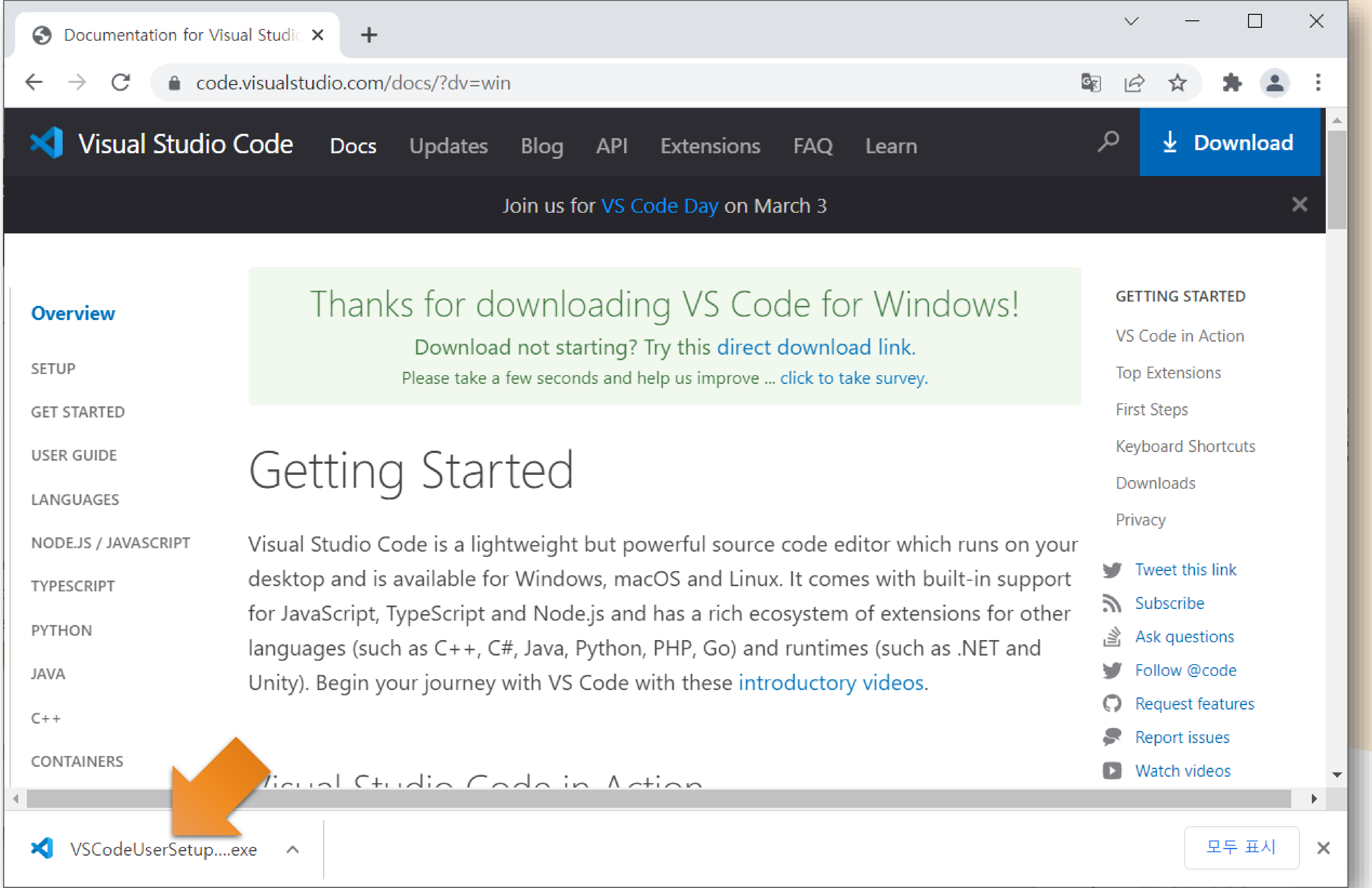
Free. Built on open source. Runs everywhere.

By using VS Code, you agree to its license and privacy statement.

```
checkValidServiceWorker(swUrl, config);

// Add some additional logging to localhost, pointing developers
// service worker/PWA documentation.
navigator.serviceWorker.ready.then(() => {
```

```
function registerValidSW(swUrl, config) {
  navigator.serviceWorker
    .register(swUrl)
    .then(registration => {
```

```
You can now view create-react-app in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://10.211.55.3:3000/

Note that the development build is not optimized.
```



Join us for VS Code Day on March 3

Thanks for downloading VS Code for Windows!

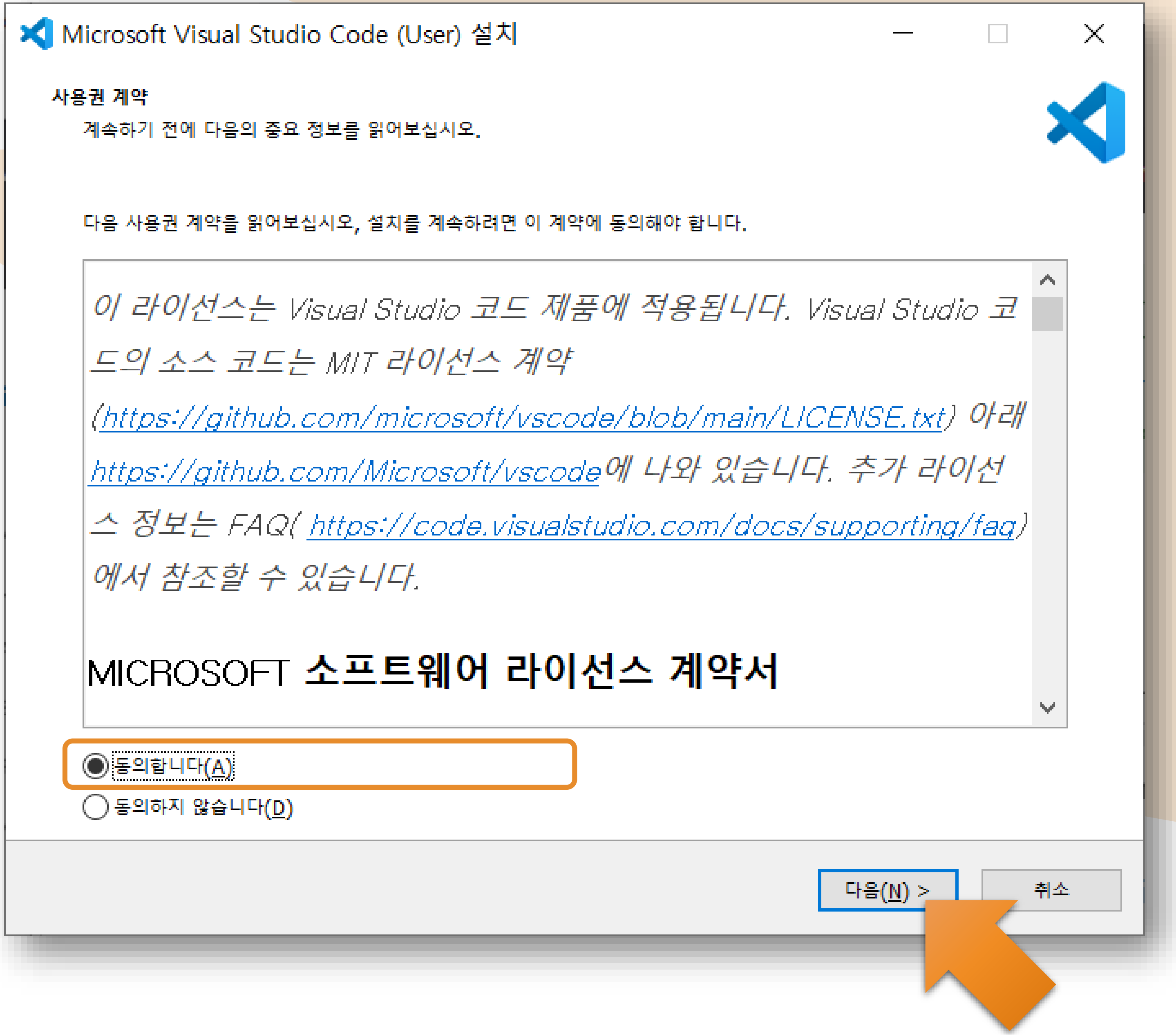
Download not starting? Try this direct download link.

Please take a few seconds and help us improve ... click to take survey.

Getting Started

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity). Begin your journey with VS Code with these introductory videos.

Visual Studio Code in Action



Microsoft Visual Studio Code (User) 설치

사용권 계약

계속하기 전에 다음의 중요 정보를 읽어보십시오.

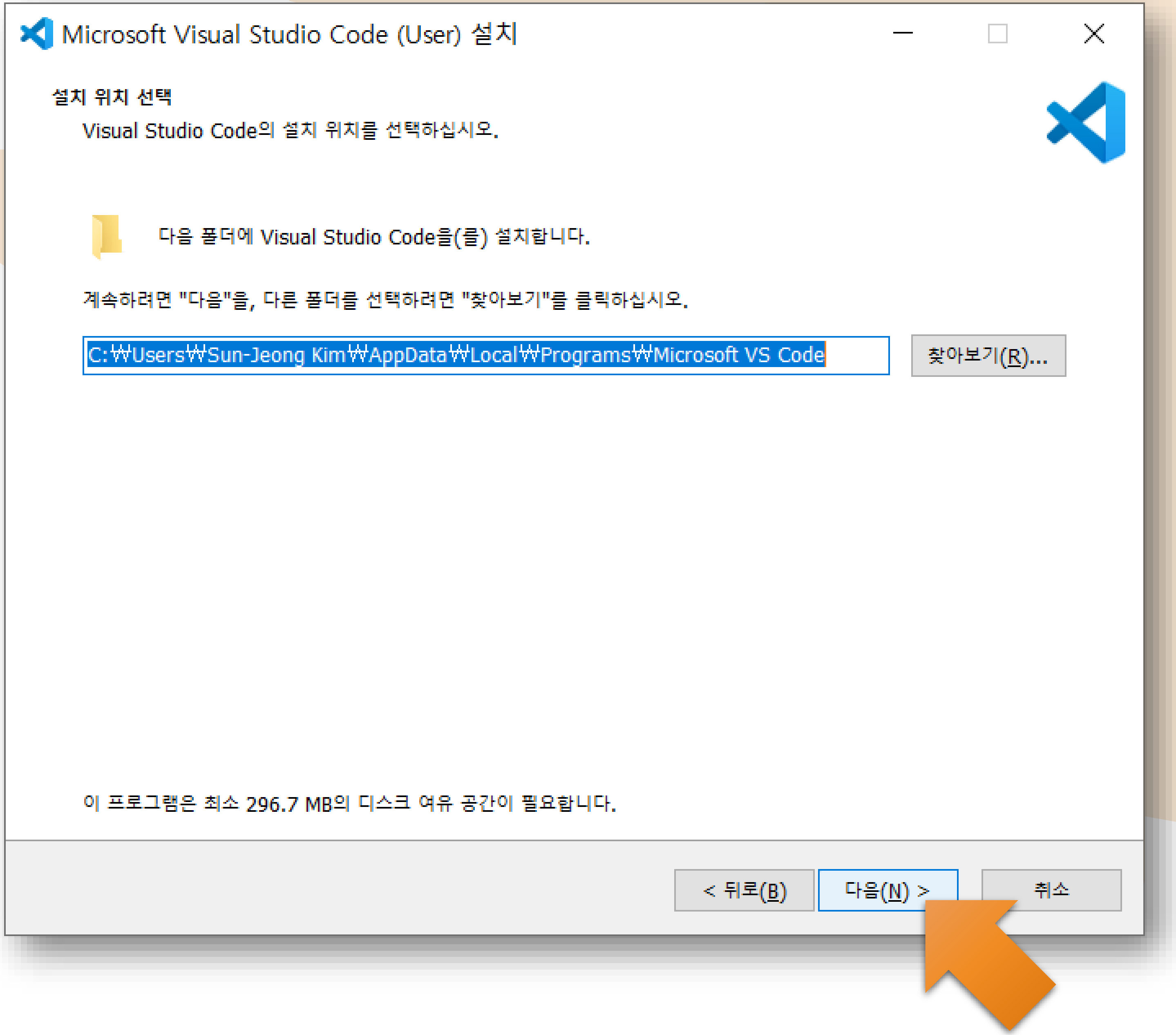
다음 사용권 계약을 읽어보십시오, 설치를 계속하려면 이 계약에 동의해야 합니다.

이 라이선스는 Visual Studio 코드 제품에 적용됩니다. Visual Studio 코드의 소스 코드는 MIT 라이선스 계약 (https://github.com/microsoft/vscode/blob/main/LICENSE.txt) 아래 https://github.com/Microsoft/vscode에 나와 있습니다. 추가 라이선스 정보는 FAQ(https://code.visualstudio.com/docs/supporting/faq) 에서 참조할 수 있습니다.

MICROSOFT 소프트웨어 라이선스 계약서

● 동의합니다(A)

○ 동의하지 않습니다(D)



Microsoft Visual Studio Code (User) 설치

설치 위치 선택

Visual Studio Code의 설치 위치를 선택하십시오.

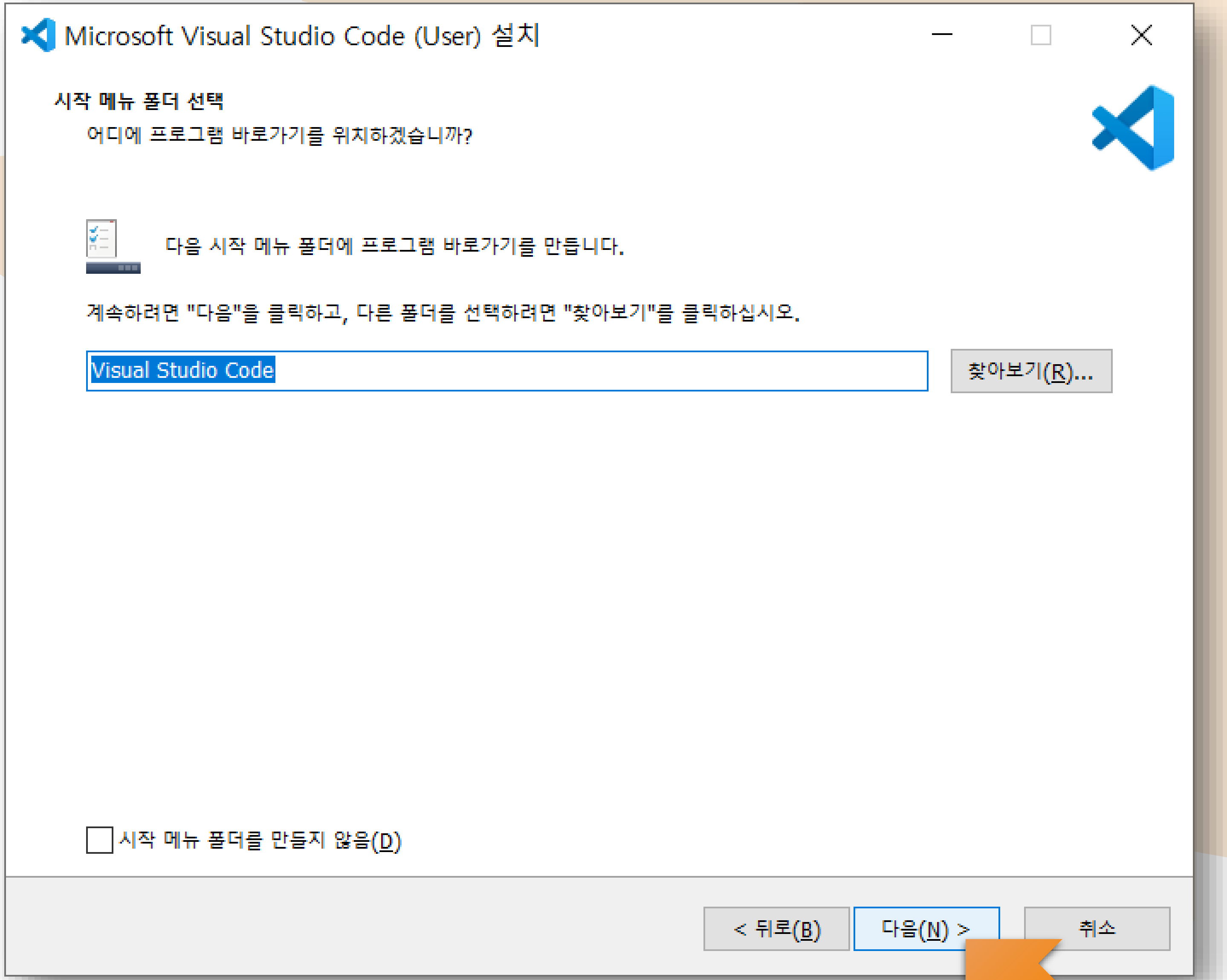
다음 폴더에 Visual Studio Code을(를) 설치합니다.

계속하려면 "다음"을, 다른 폴더를 선택하려면 "찾아보기"를 클릭하십시오.

C:₩Users₩Sun-Jeong Kim₩AppData₩Local₩Programs₩Microsoft VS Code

찾아보기(R)...

이 프로그램은 최소 296.7 MB의 디스크 여유 공간이 필요합니다.



Microsoft Visual Studio Code (User) 설치

시작 메뉴 폴더 선택

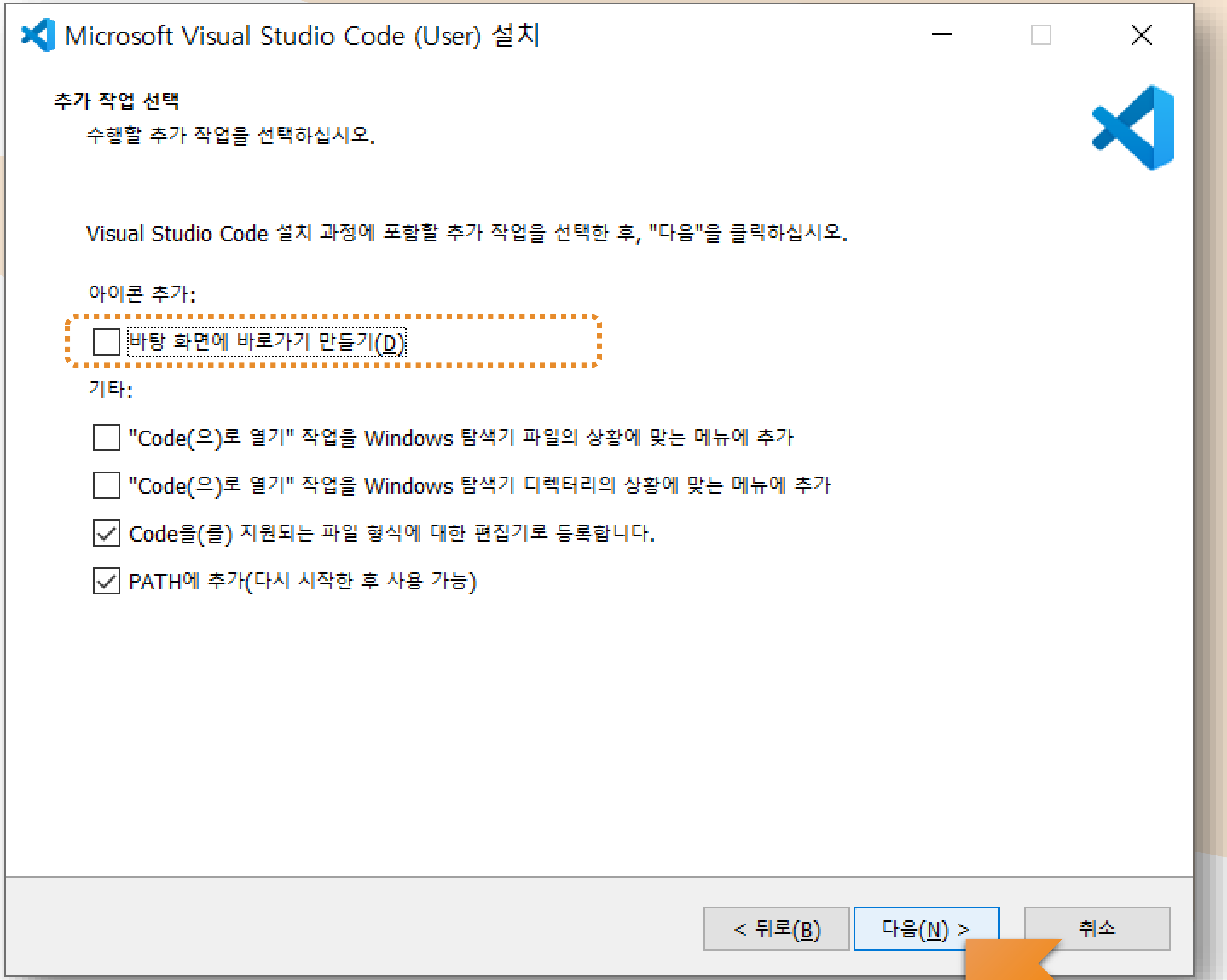
어디에 프로그램 바로가기를 위치하겠습니까?

다음 시작 메뉴 폴더에 프로그램 바로가기를 만듭니다.

계속하려면 "다음"을 클릭하고, 다른 폴더를 선택하려면 "찾아보기"를 클릭하십시오.

Visual Studio Code

□ 시작 메뉴 폴더를 만들지 않음(D)



Microsoft Visual Studio Code (User) 설치

추가 작업 선택

수행할 추가 작업을 선택하십시오.

Visual Studio Code 설치 과정에 포함할 추가 작업을 선택한 후, "다음"을 클릭하십시오.

아이콘 추가:

□ 바탕 화면에 바로가기 만들기(D)

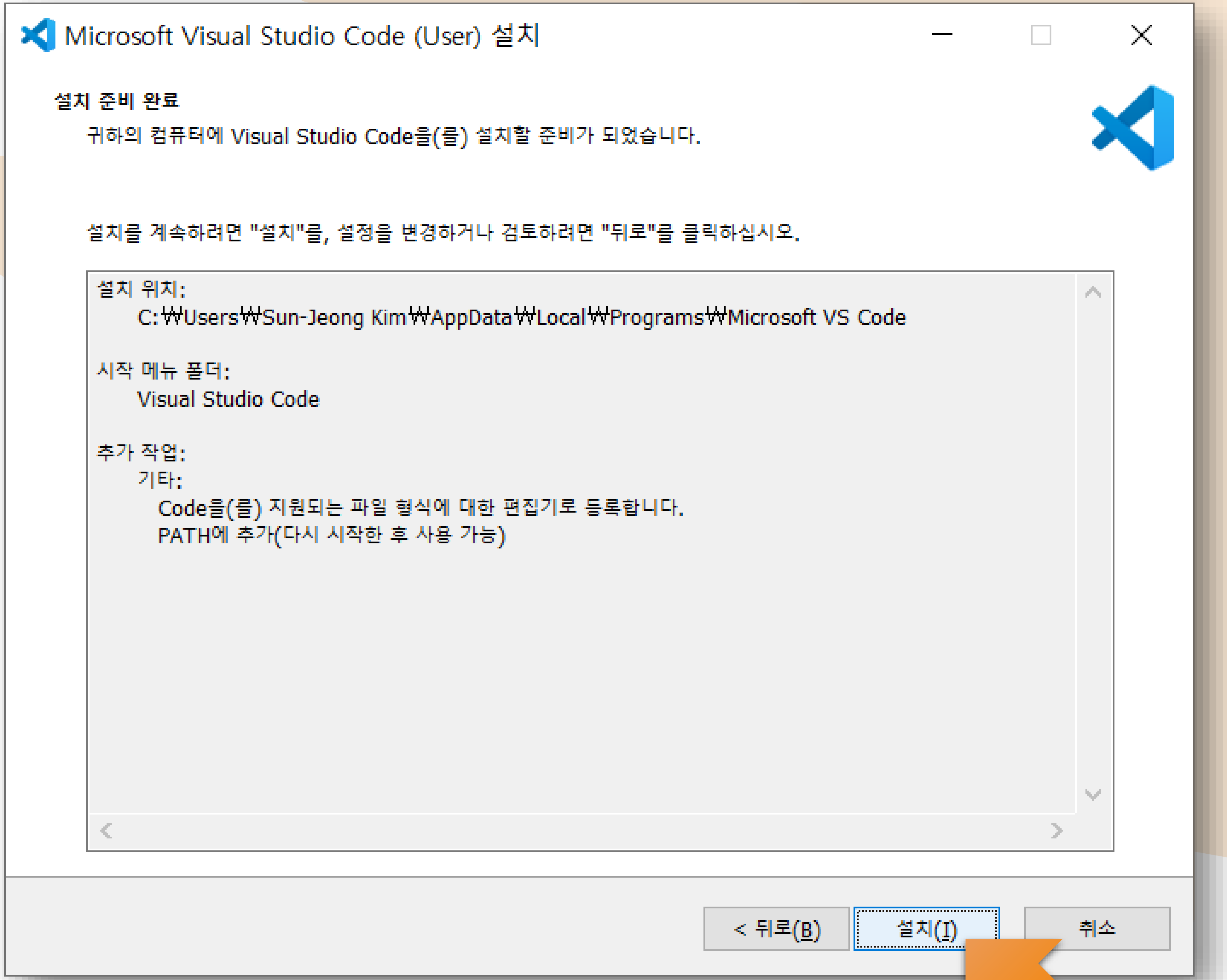
기타:

□ "Code(으)로 열기" 작업을 Windows 탐색기 파일의 상황에 맞는 메뉴에 추가

□ "Code(으)로 열기" 작업을 Windows 탐색기 디렉터리의 상황에 맞는 메뉴에 추가

☑ Code을(를) 지원되는 파일 형식에 대한 편집기로 등록합니다.

☑ PATH에 추가(다시 시작한 후 사용 가능)



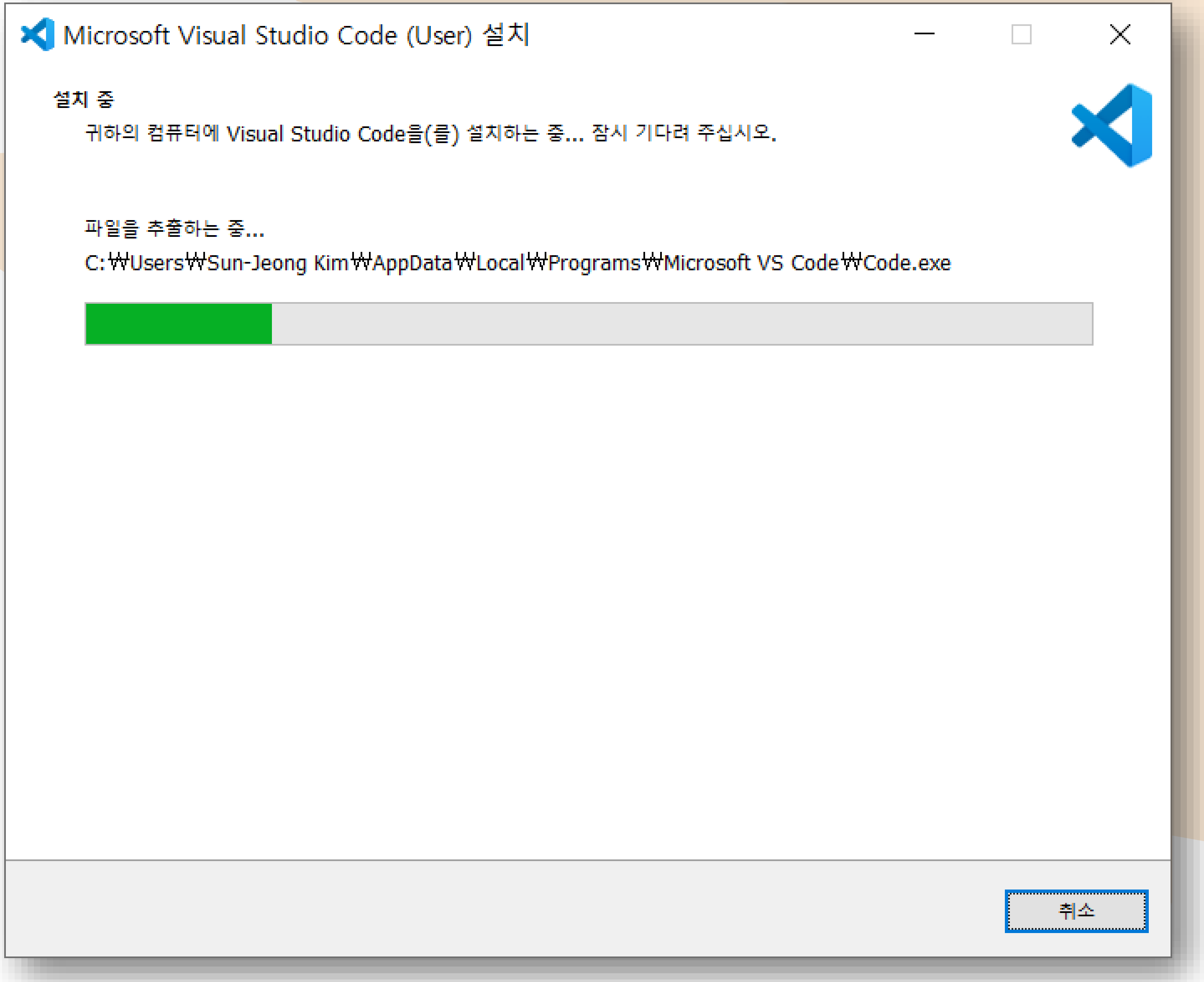
Microsoft Visual Studio Code (User) 설치

설치 준비 완료

귀하의 컴퓨터에 Visual Studio Code을(를) 설치할 준비가 되었습니다.

설치를 계속하려면 "설치"를, 설정을 변경하거나 검토하려면 "뒤로"를 클릭하십시오.

```
설치 위치:
      C:₩Users₩Sun-Jeong Kim₩AppData₩Local₩Programs₩Microsoft VS Code

시작 메뉴 폴더:
      Visual Studio Code

추가 작업:
      기타:
         Code을(를) 지원되는 파일 형식에 대한 편집기로 등록합니다.
         PATH에 추가(다시 시작한 후 사용 가능)
```



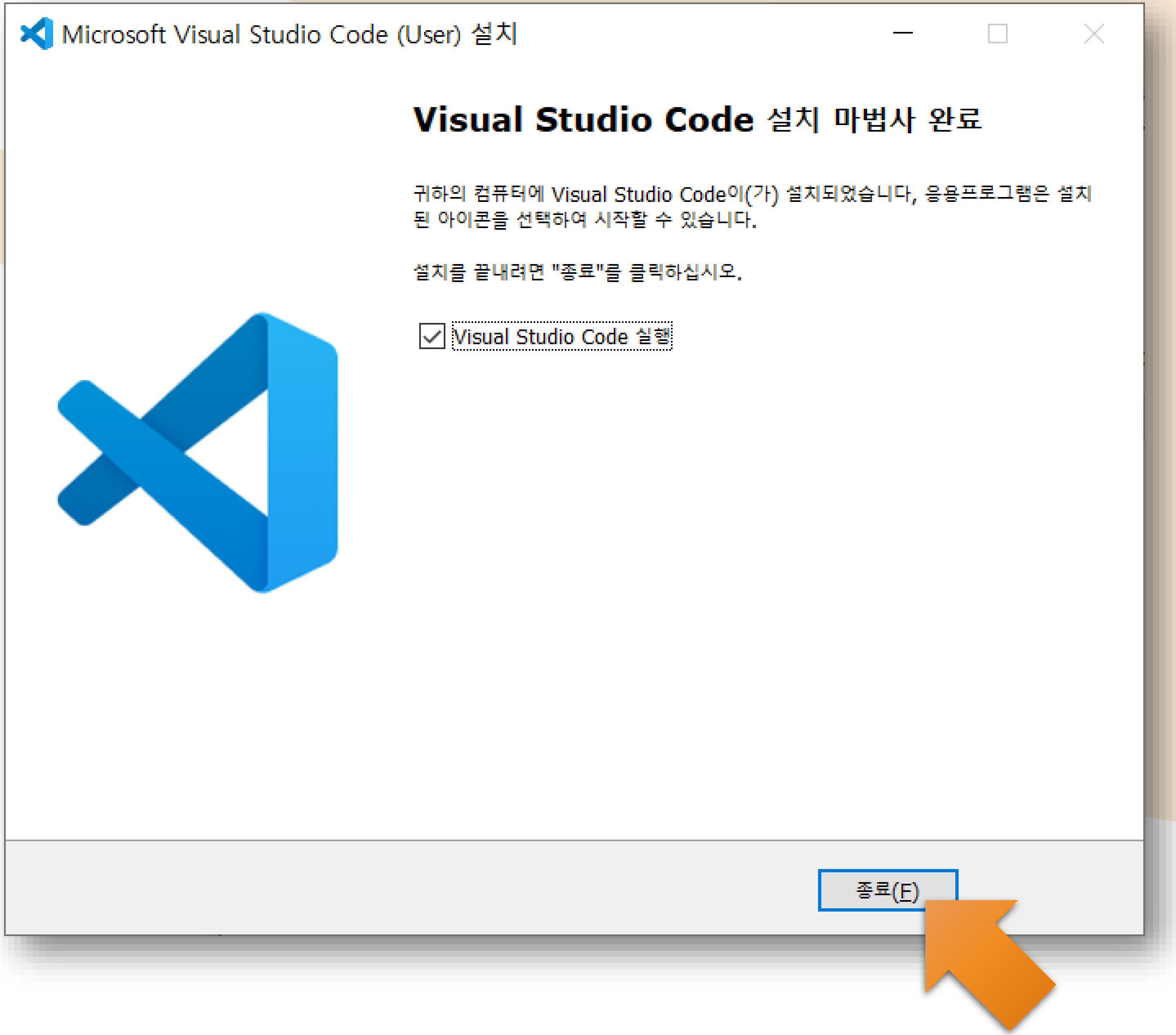
Microsoft Visual Studio Code (User) 설치

설치 중

귀하의 컴퓨터에 Visual Studio Code을(를) 설치하는 중... 잠시 기다려 주십시오.

파일을 추출하는 중...

C:₩Users₩Sun-Jeong Kim₩AppData₩Local₩Programs₩Microsoft VS Code₩Code.exe



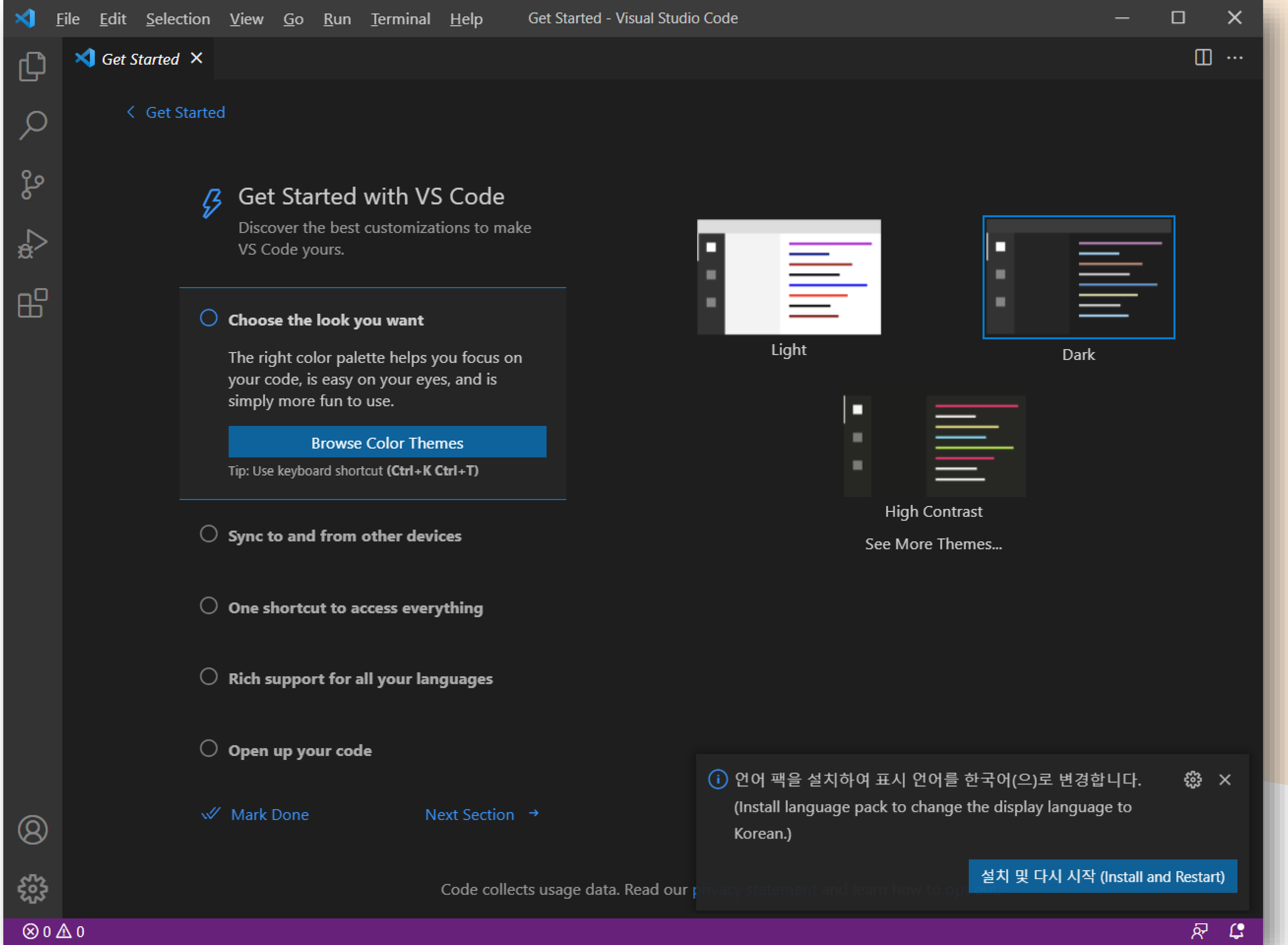
Microsoft Visual Studio Code (User) 설치

Visual Studio Code 설치 마법사 완료

귀하의 컴퓨터에 Visual Studio Code이(가) 설치되었습니다. 응용프로그램은 설치된 아이콘을 선택하여 시작할 수 있습니다.

설치를 끝내려면 "종료"를 클릭하십시오.

☑ Visual Studio Code 실행



Get Started with VS Code

Discover the best customizations to make VS Code yours.

○ **Choose the look you want**

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

Tip: Use keyboard shortcut **(Ctrl+K Ctrl+T)**

○ **Sync to and from other devices**

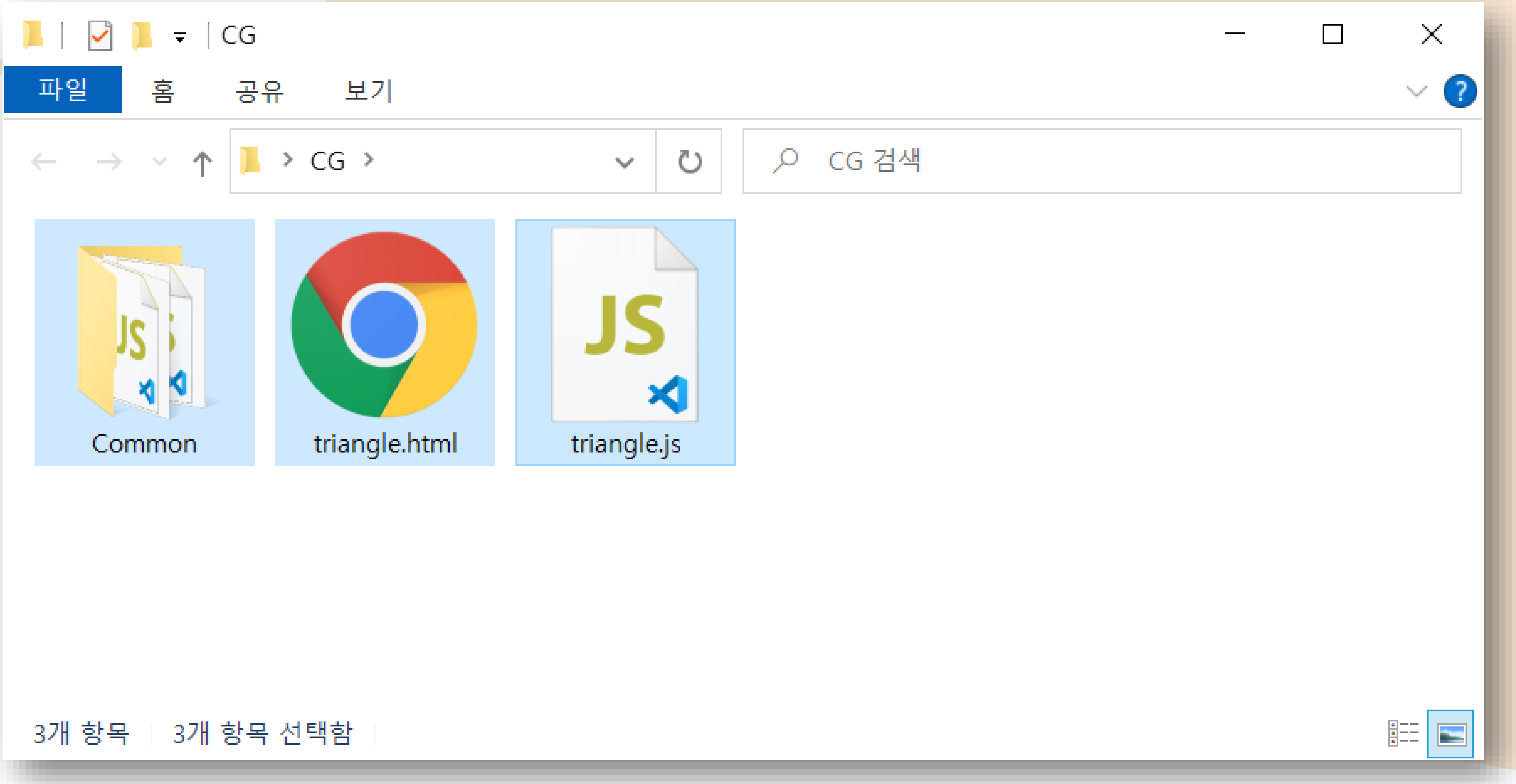
○ **One shortcut to access everything**

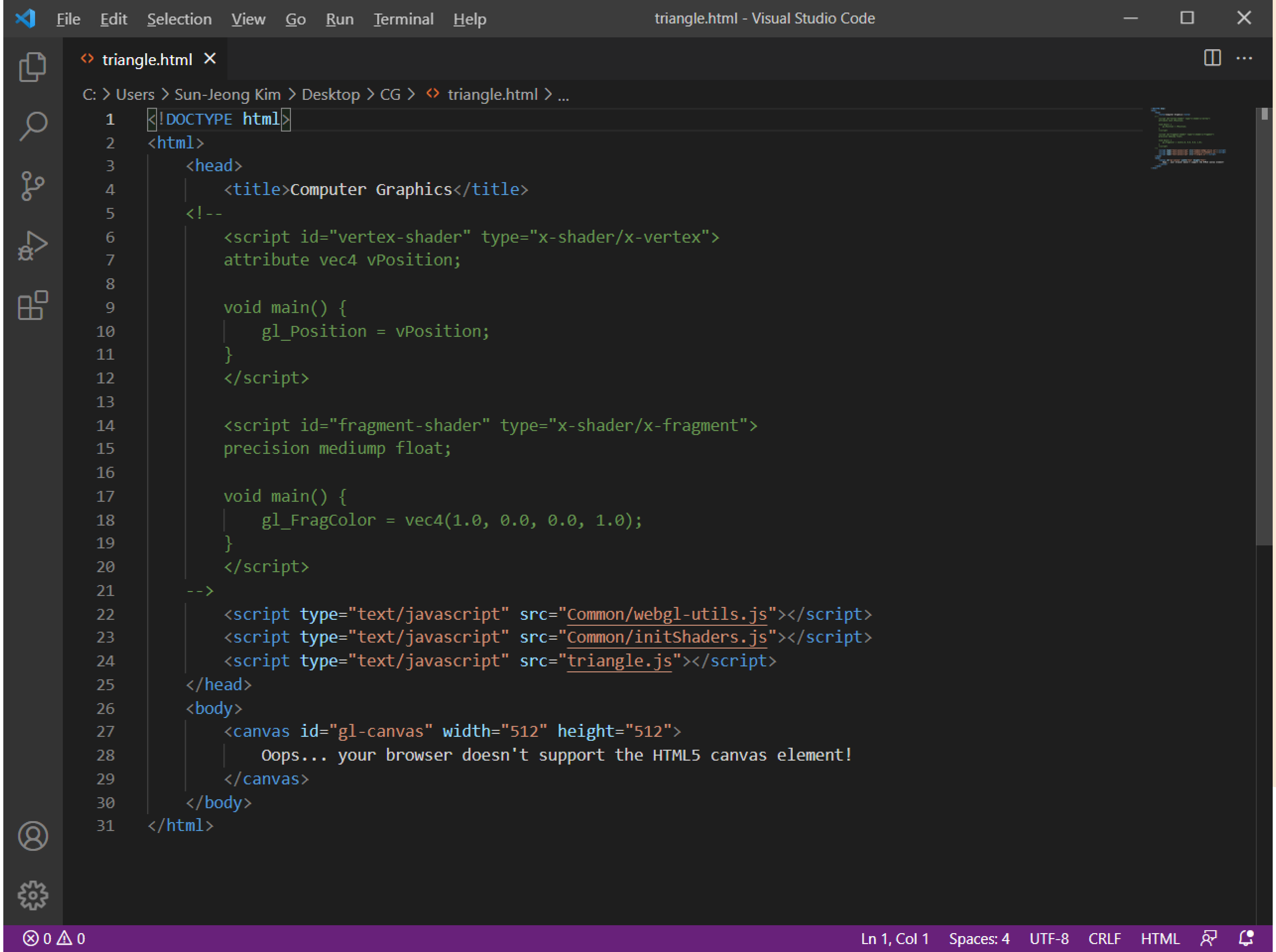
○ **Rich support for all your languages**

○ **Open up your code**

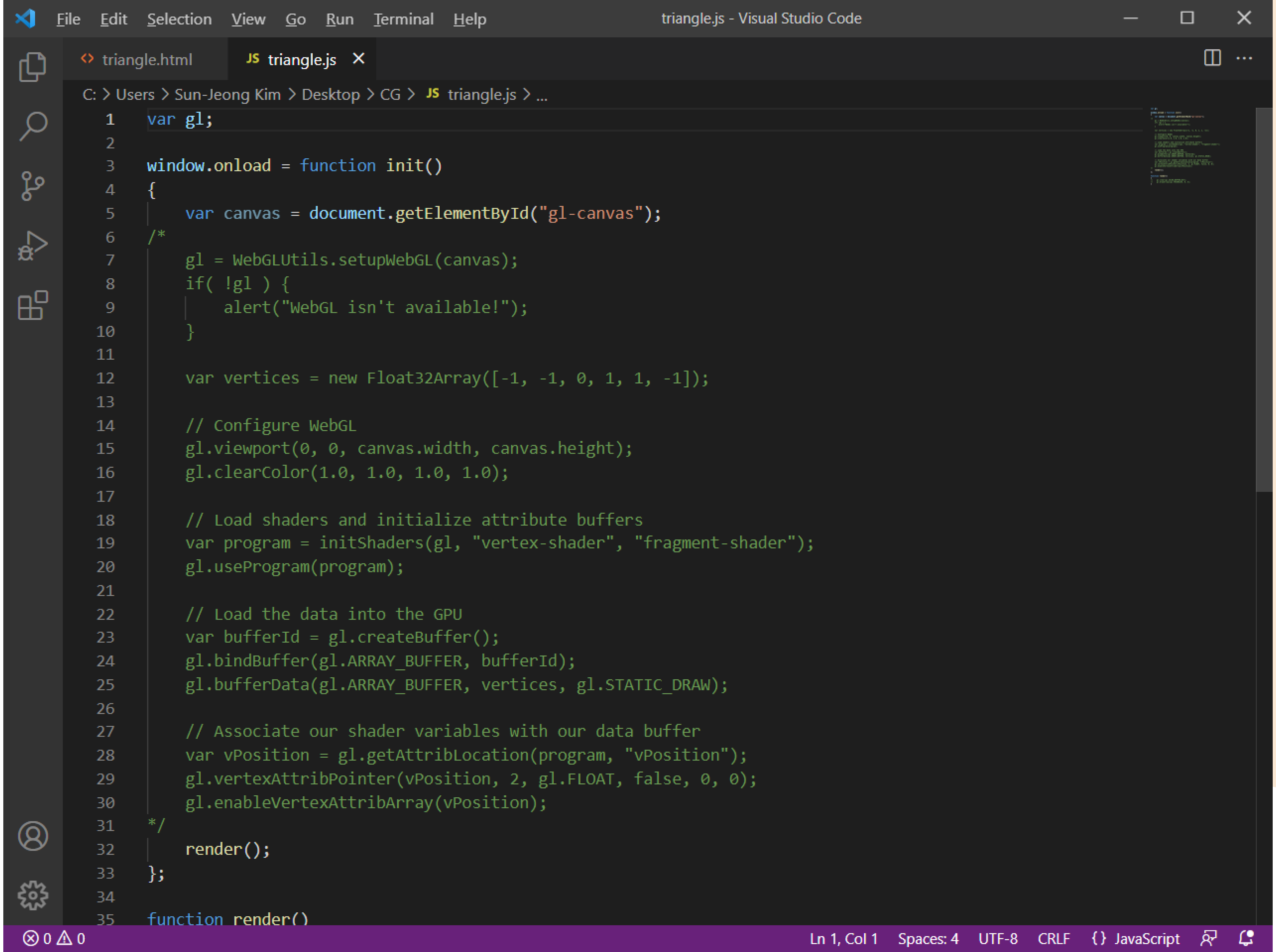
Code collects usage data. Read our

ⓘ 언어 팩을 설치하여 표시 언어를 한국어(으)로 변경합니다. (Install language pack to change the display language to Korean.)

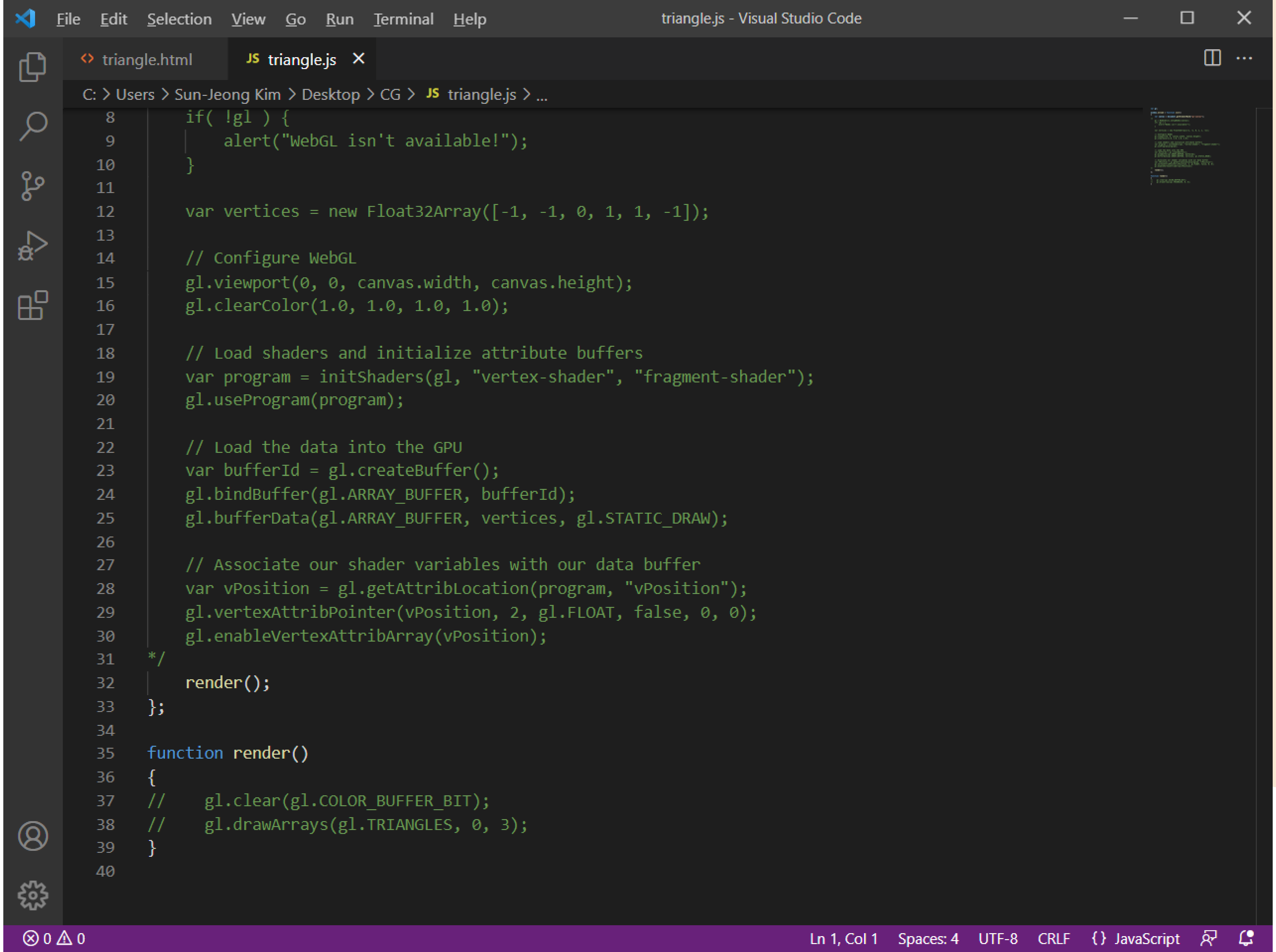

CG
파일
홈
공유
보기
CG
CG 검색
Common
triangle.html
triangle.js
3개 항목
3개 항목 선택함


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

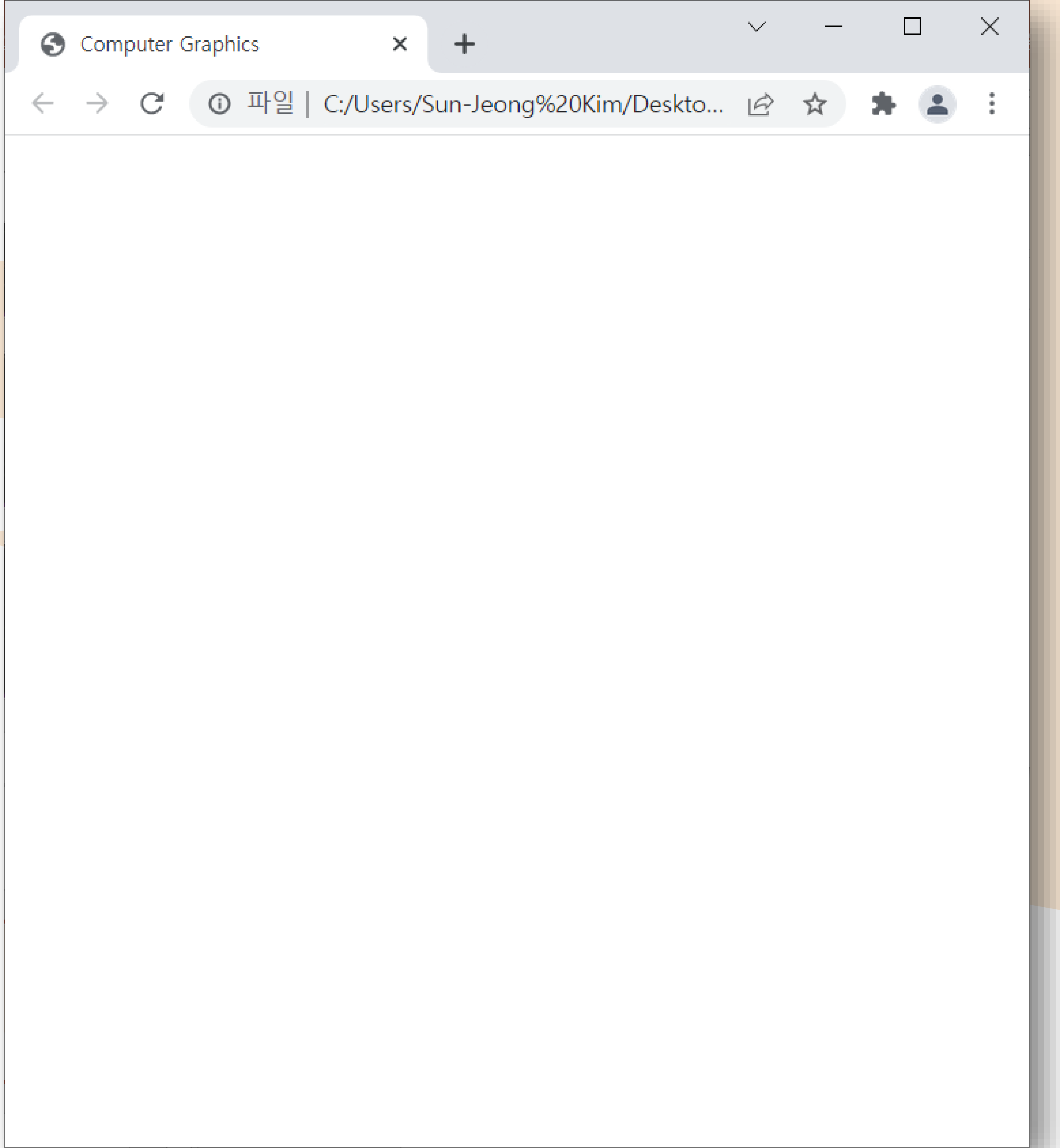
```
<!DOCTYPE html>
<html>
    <head>
        <title>Computer Graphics</title>
    <!--
        <script id="vertex-shader" type="x-shader/x-vertex">
        attribute vec4 vPosition;

        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>
```



```
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()
```

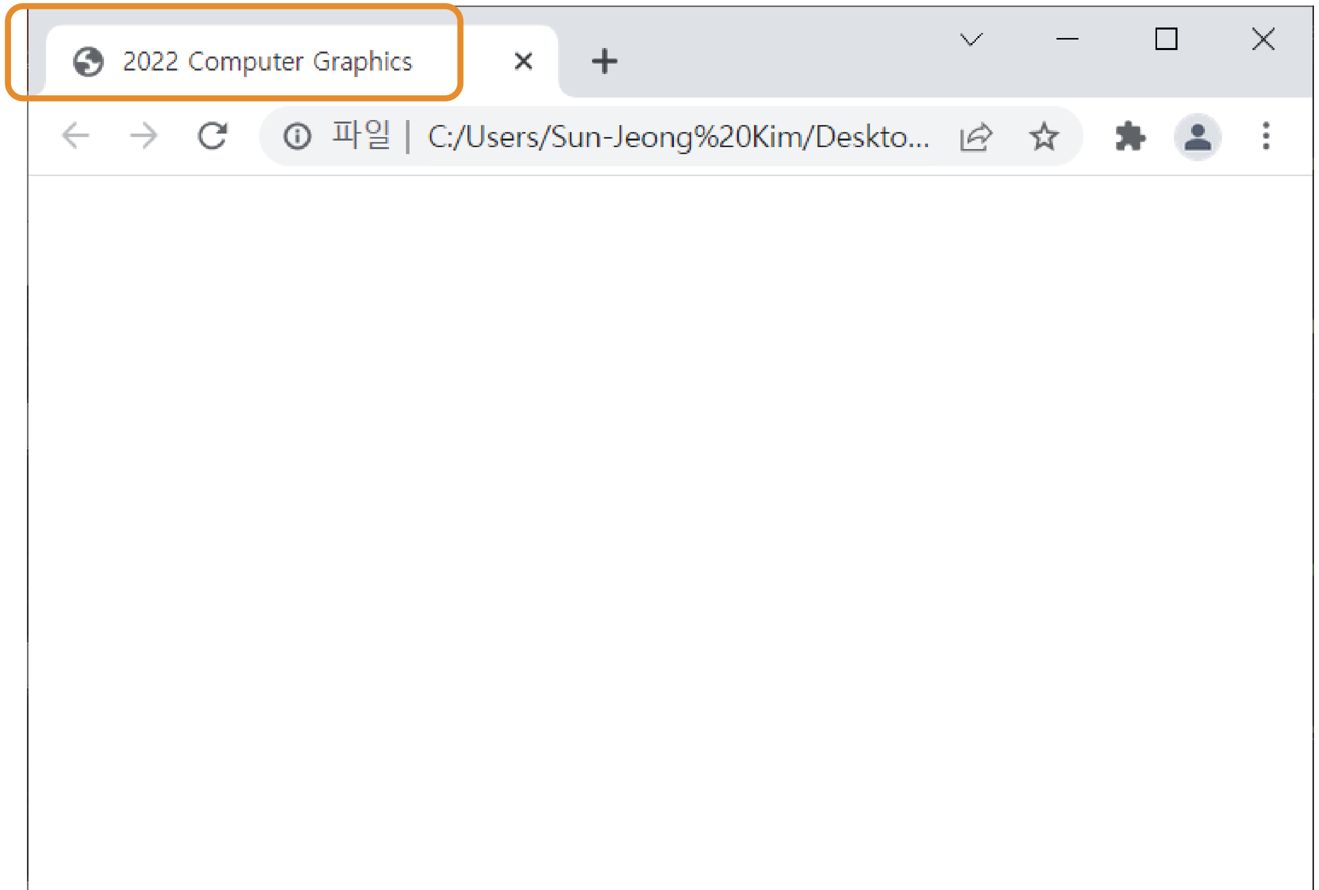


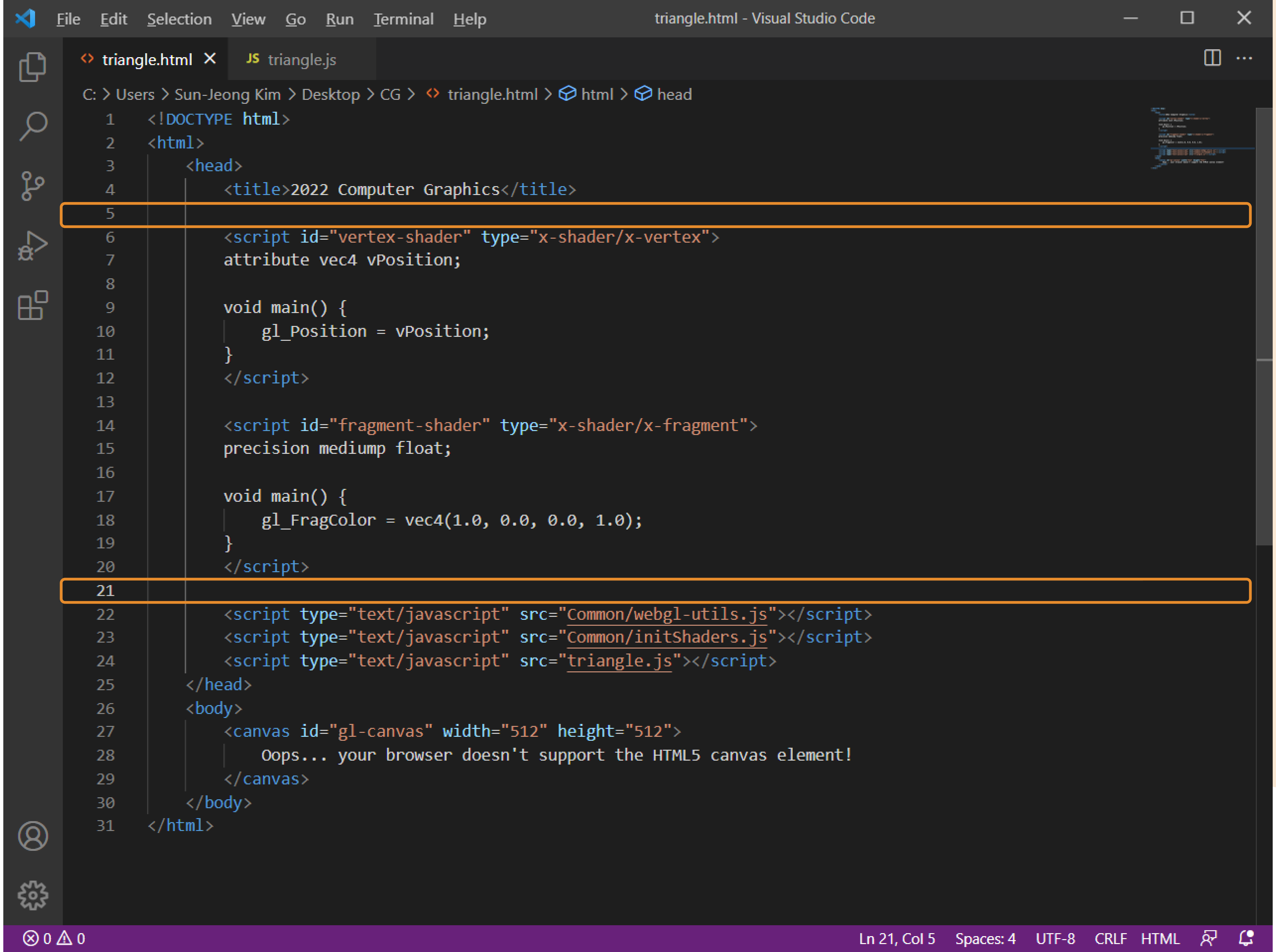
```
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    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);
}
```


Computer Graphics
파일 | C:/Users/Sun-Jeong%20Kim/Deskto...

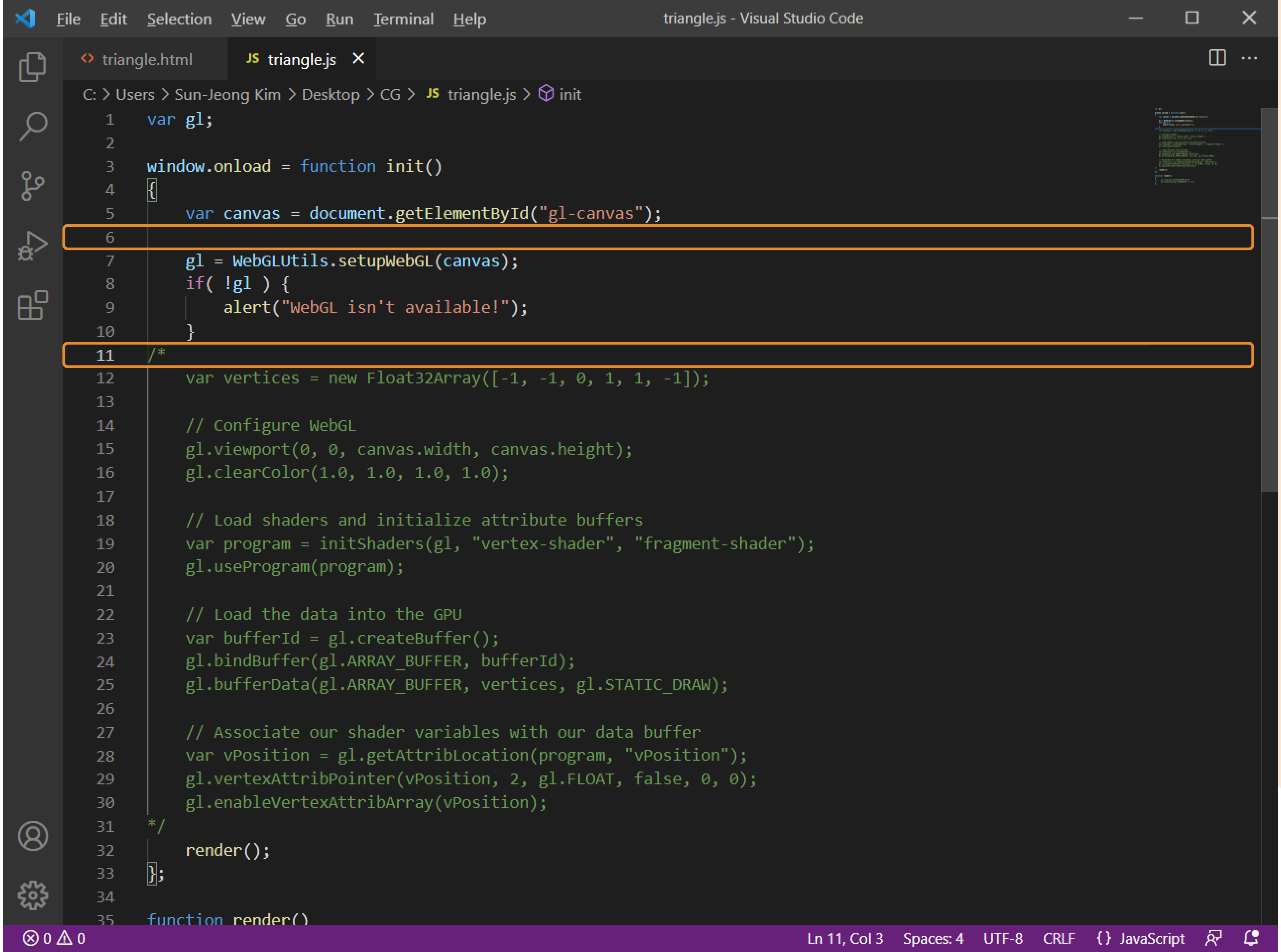
연습문제 (1)

- 웹 문서의 제목(Title)을 자신의 학번과 이름으로 변경하시오.

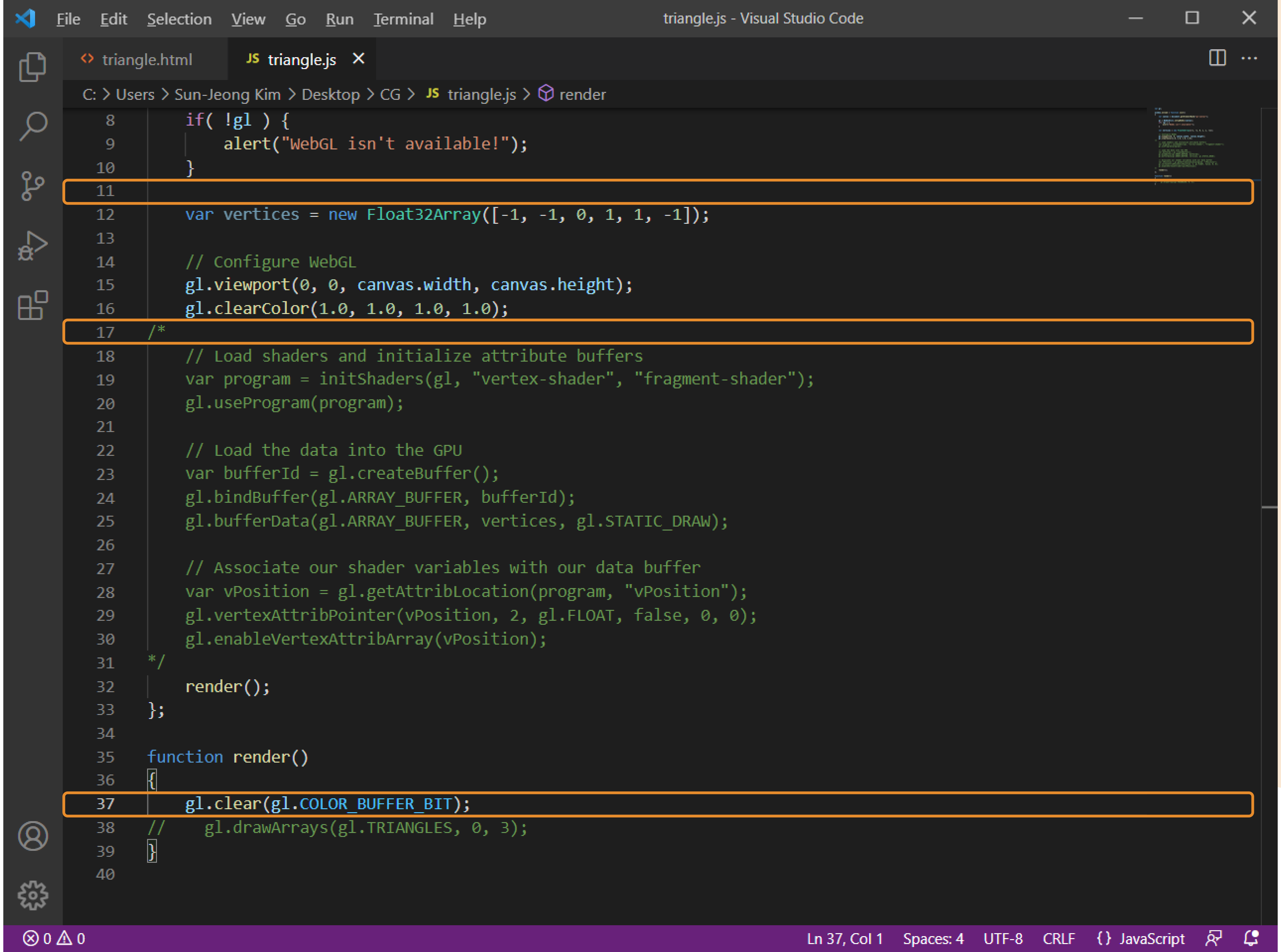



File
Edit
Selection
View
Go
Run
Terminal
Help
triangle.html - Visual Studio Code
triangle.html
triangle.js
C: > Users > Sun-Jeong Kim > Desktop > CG > triangle.html > html > head
⊗ 0 ⚠ 0
Ln 21, Col 5
Spaces: 4
UTF-8
CRLF
HTML

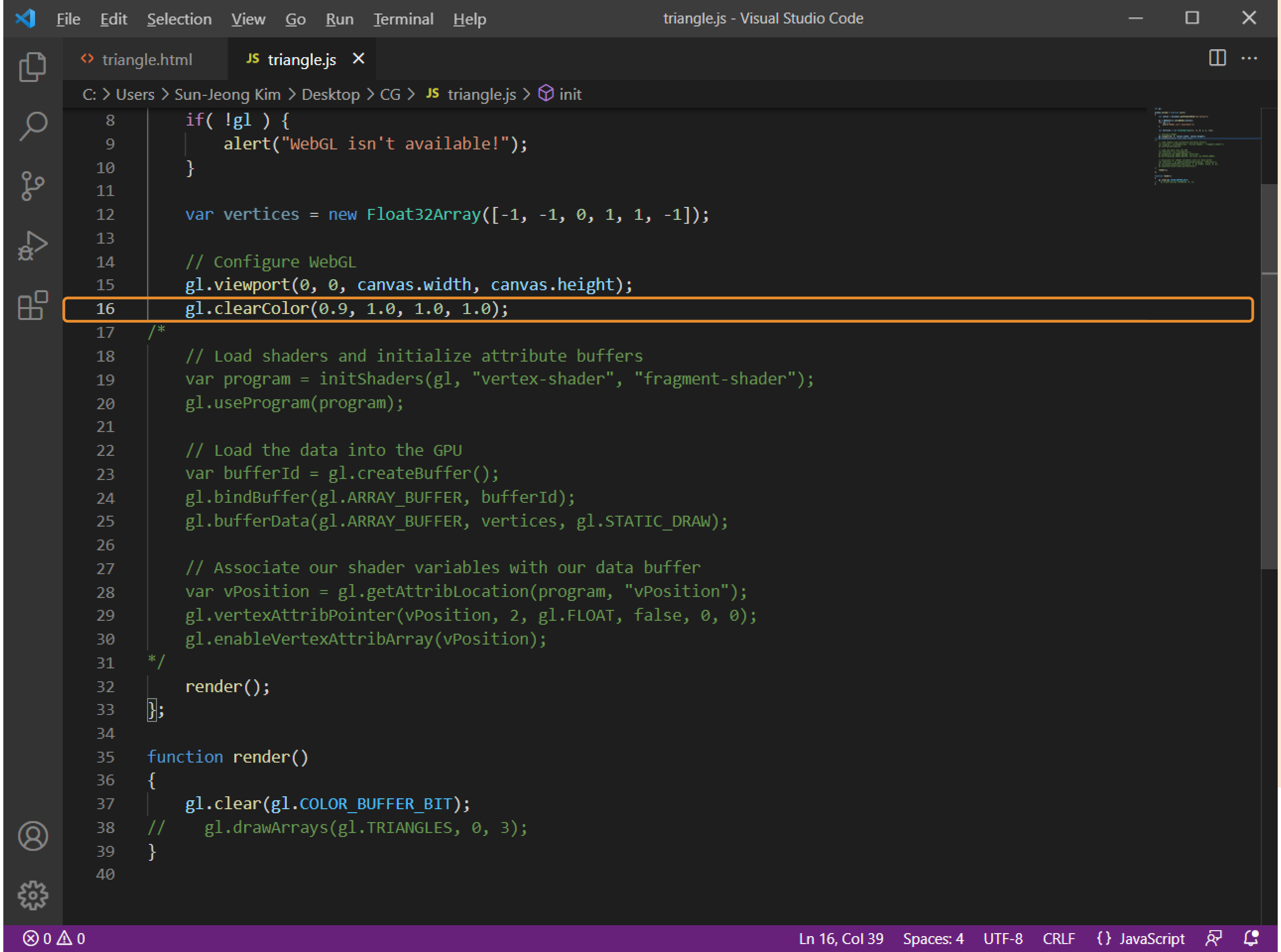
```html
<!DOCTYPE html>
<html>
    <head>
        <title>2022 Computer Graphics</title>

        <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>
```



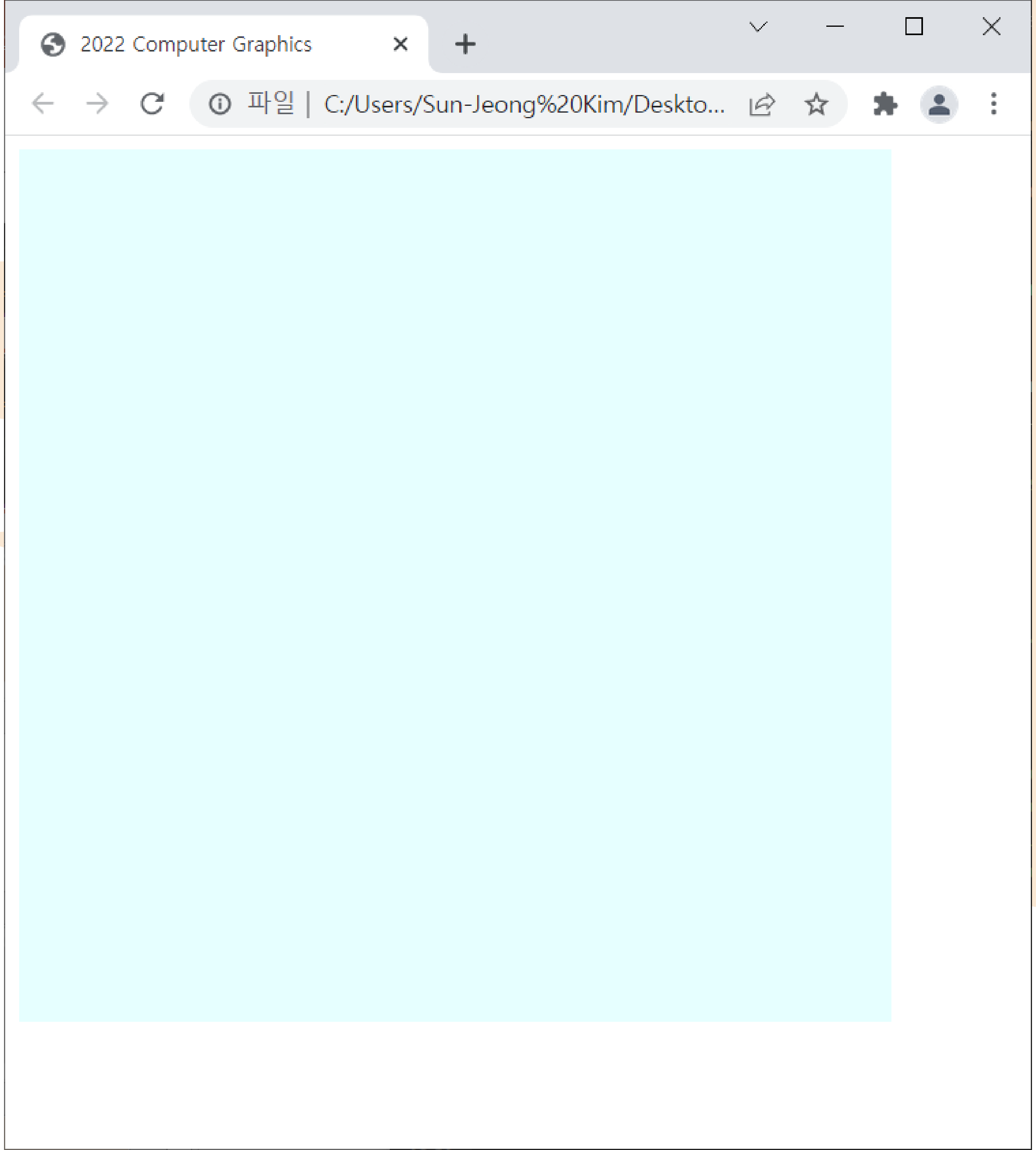
```
var gl;

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()
```



```
    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);
}
```

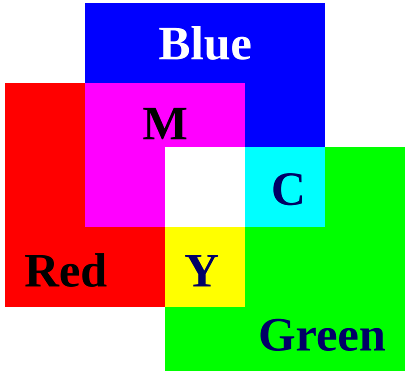


```
    if( !gl ) {
        alert("WebGL isn't available!");
    }

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

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

```

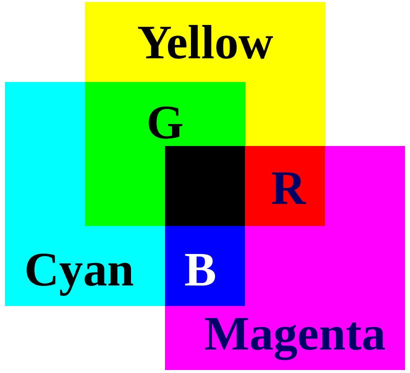

2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%20Kim/Deskto...

Color

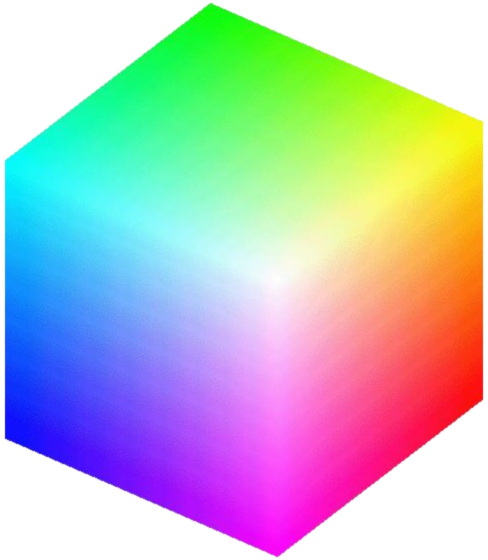
- Three color theory
  - Our brains do not receive the entire color distribution but rather than three values
- Additive color – ex) CRT
- Subtractive color – ex) printing



Additive Color



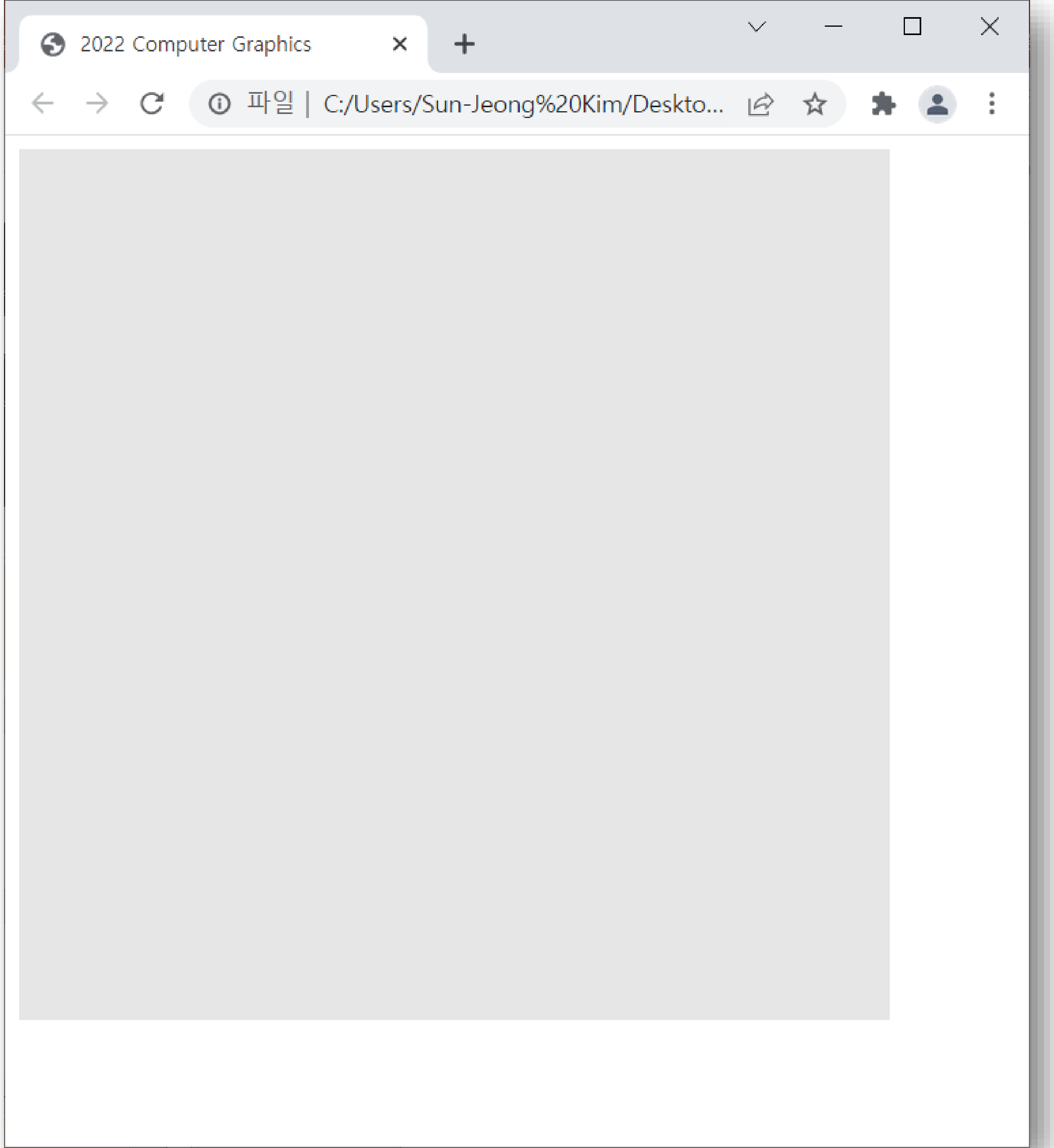
Subtractive Color

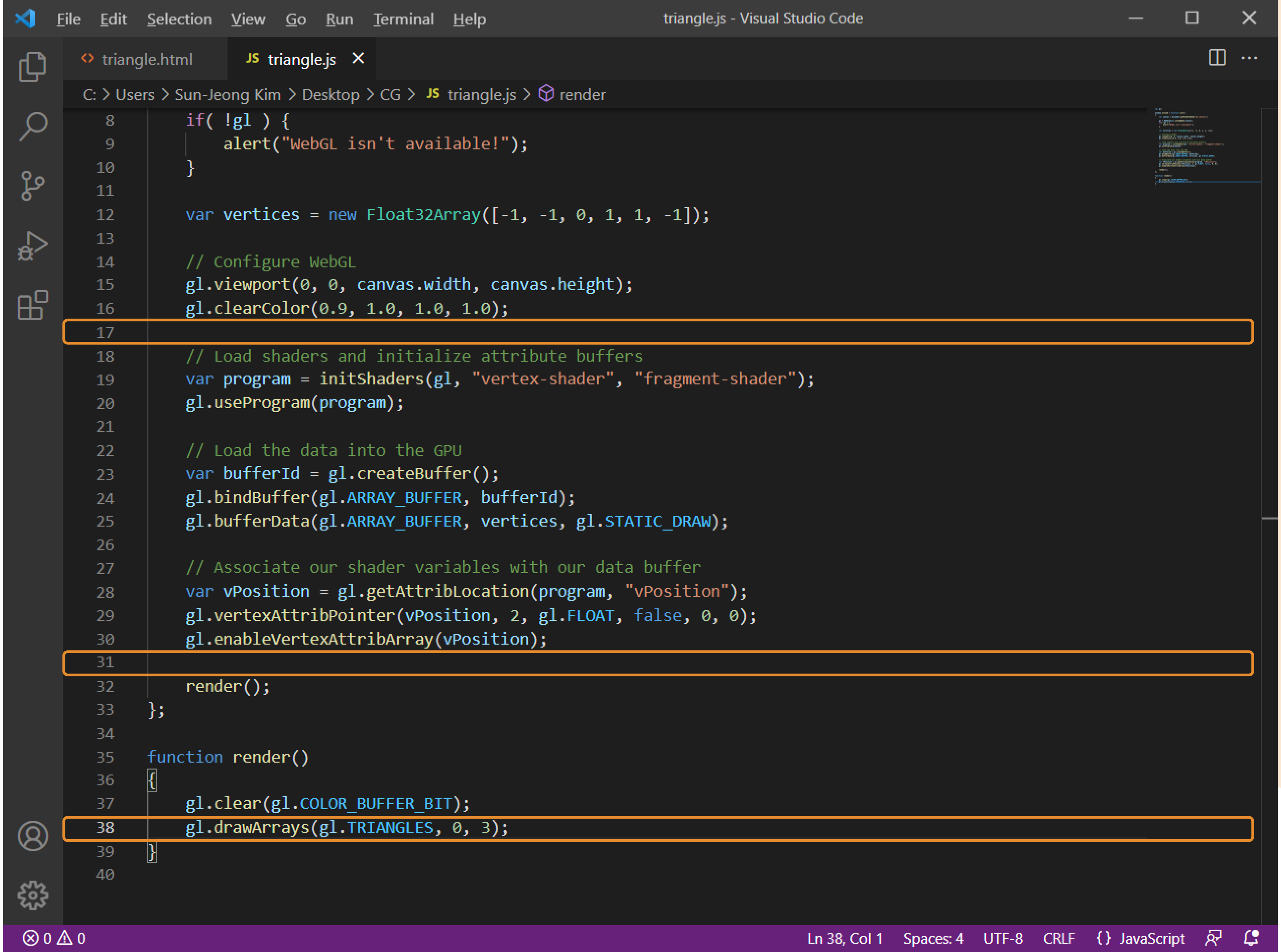


Color Solid

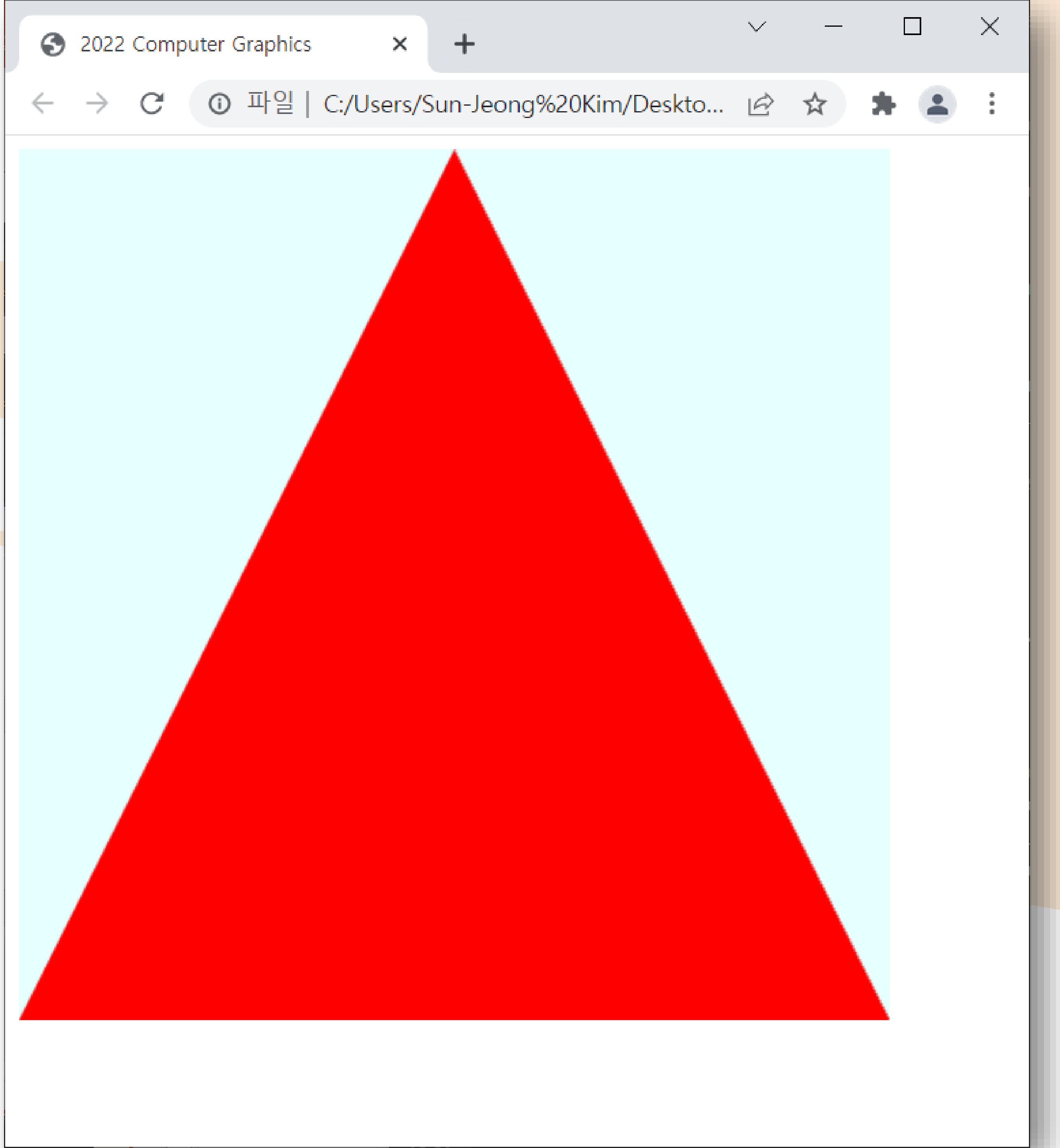
연습문제 (2)

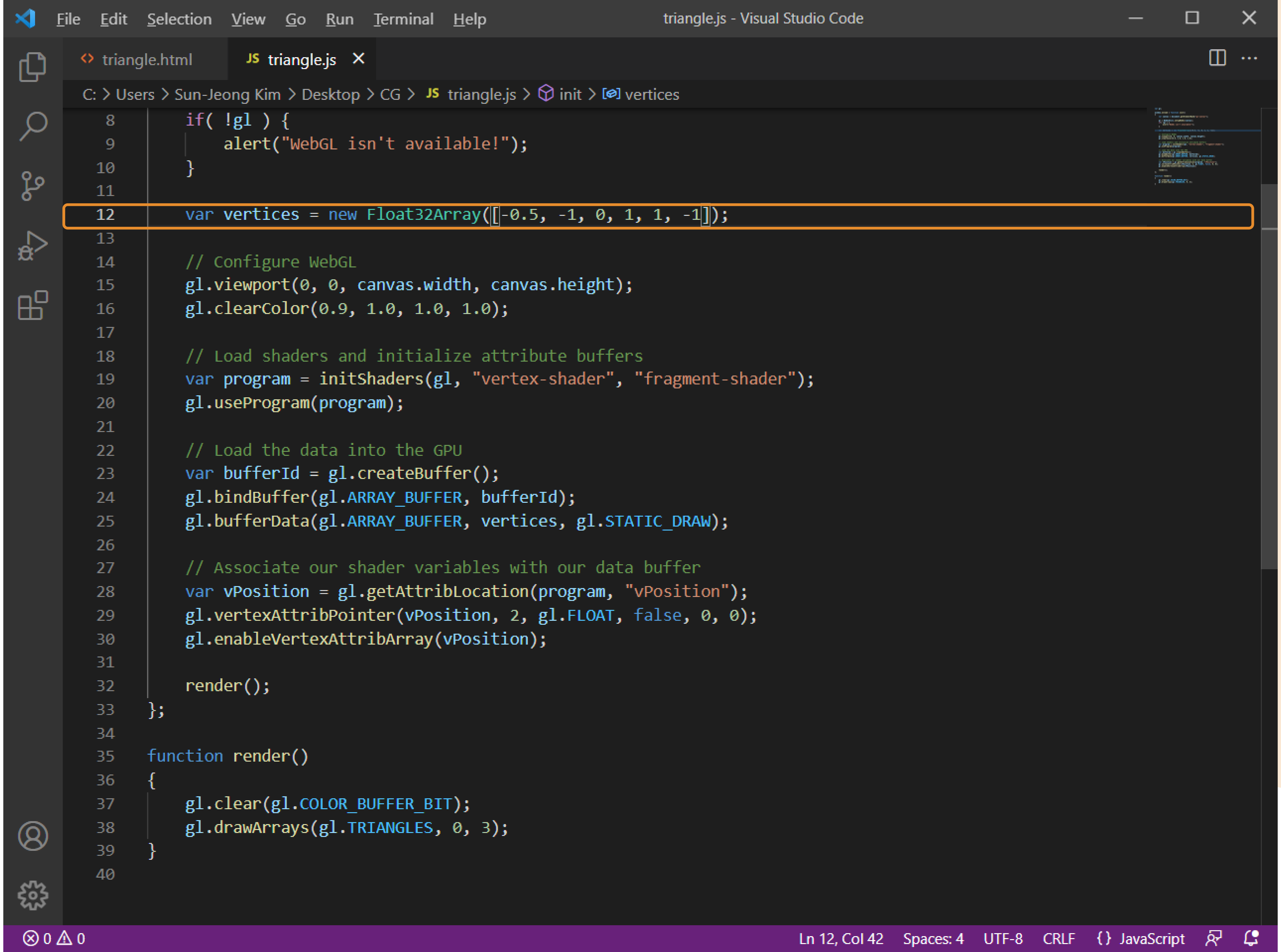
- 배경색을 변경하시오.



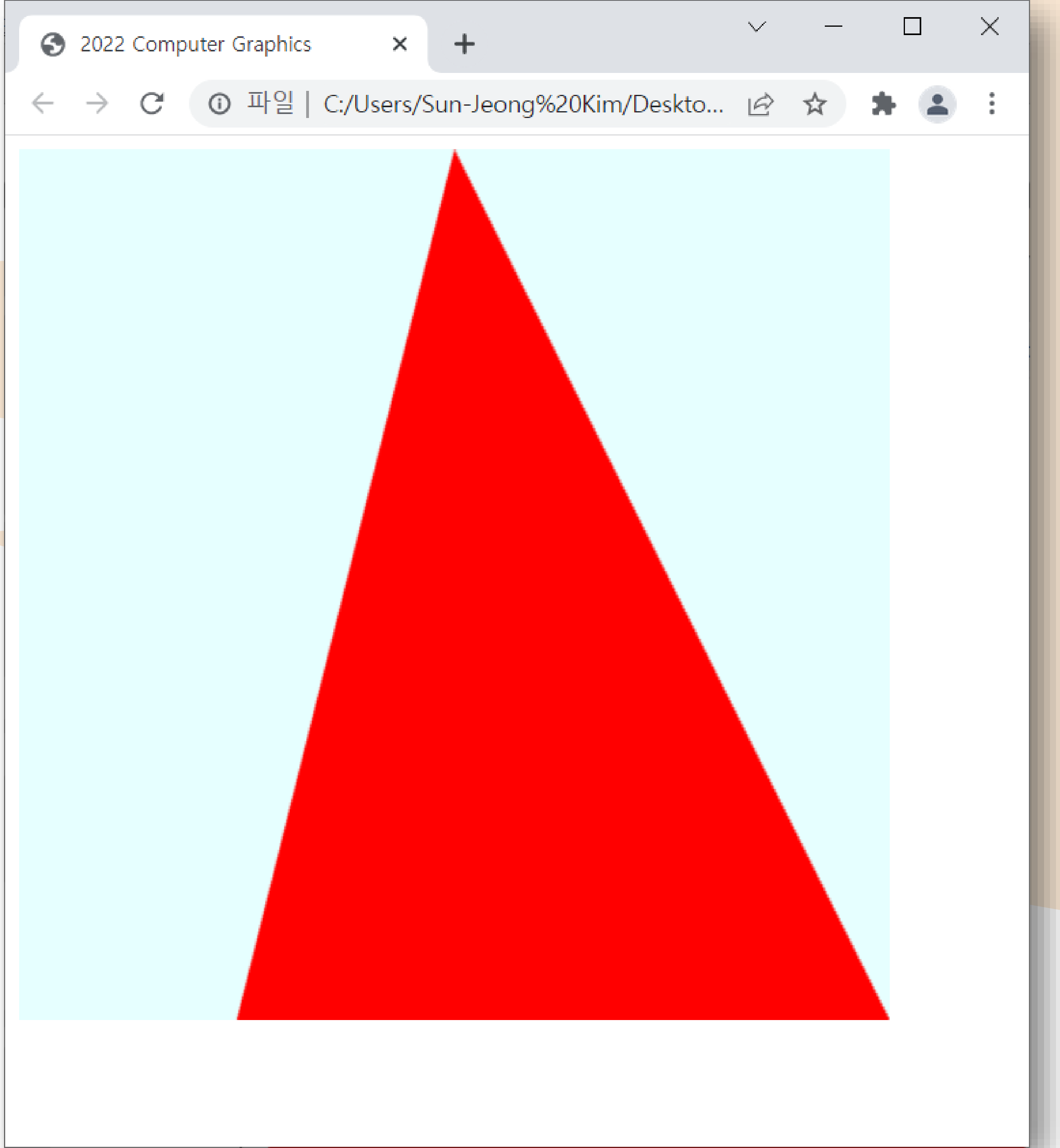


```javascript
    if( !gl ) {
        alert("WebGL isn't available!");
    }

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

    // Configure WebGL
    gl.viewport(0, 0, canvas.width, canvas.height);
    gl.clearColor(0.9, 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%20Kim/Deskto...



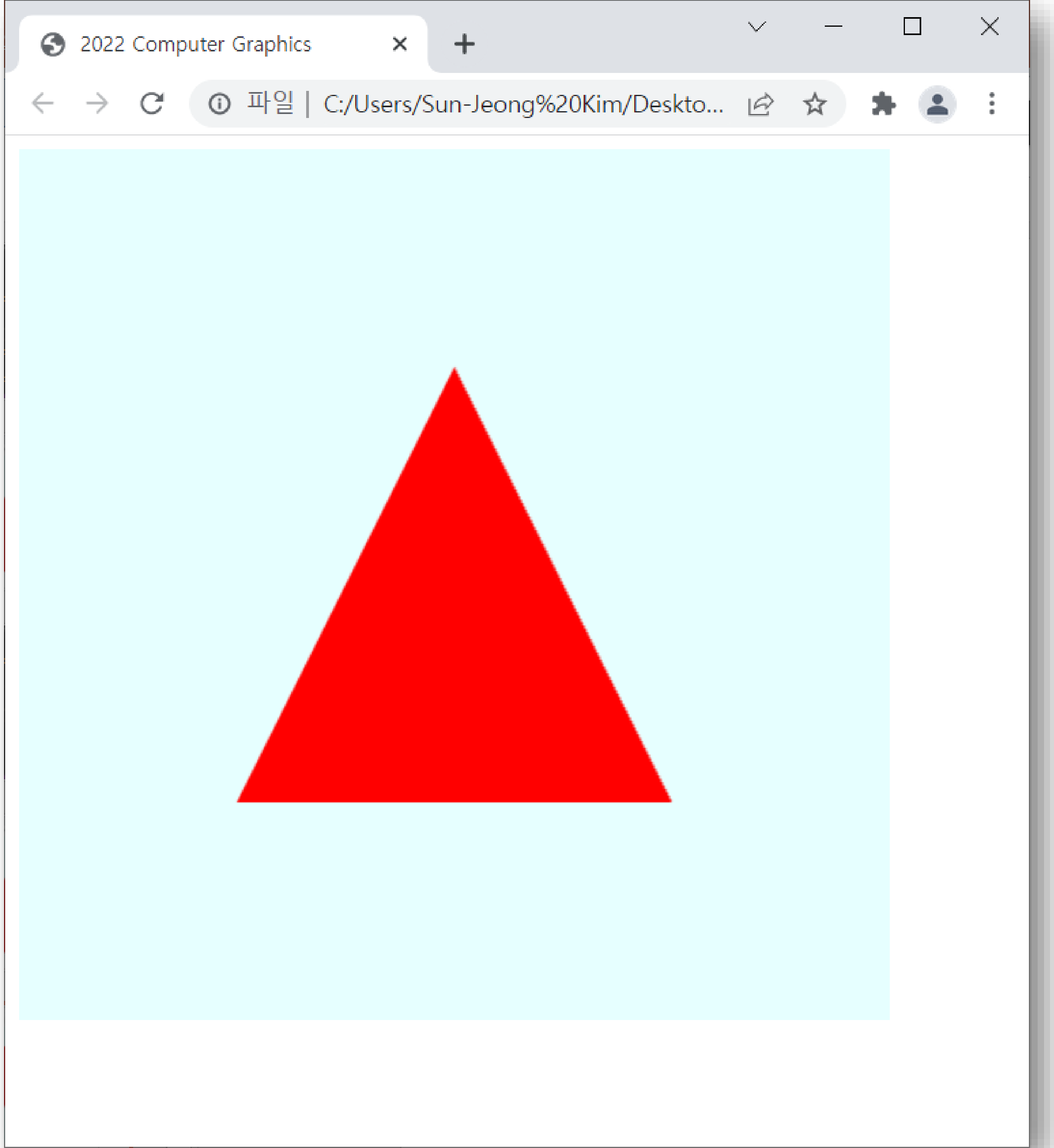
```javascript
    if( !gl ) {
        alert("WebGL isn't available!");
    }

    var vertices = new Float32Array([-0.5, -1, 0, 1, 1, -1]);

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

```

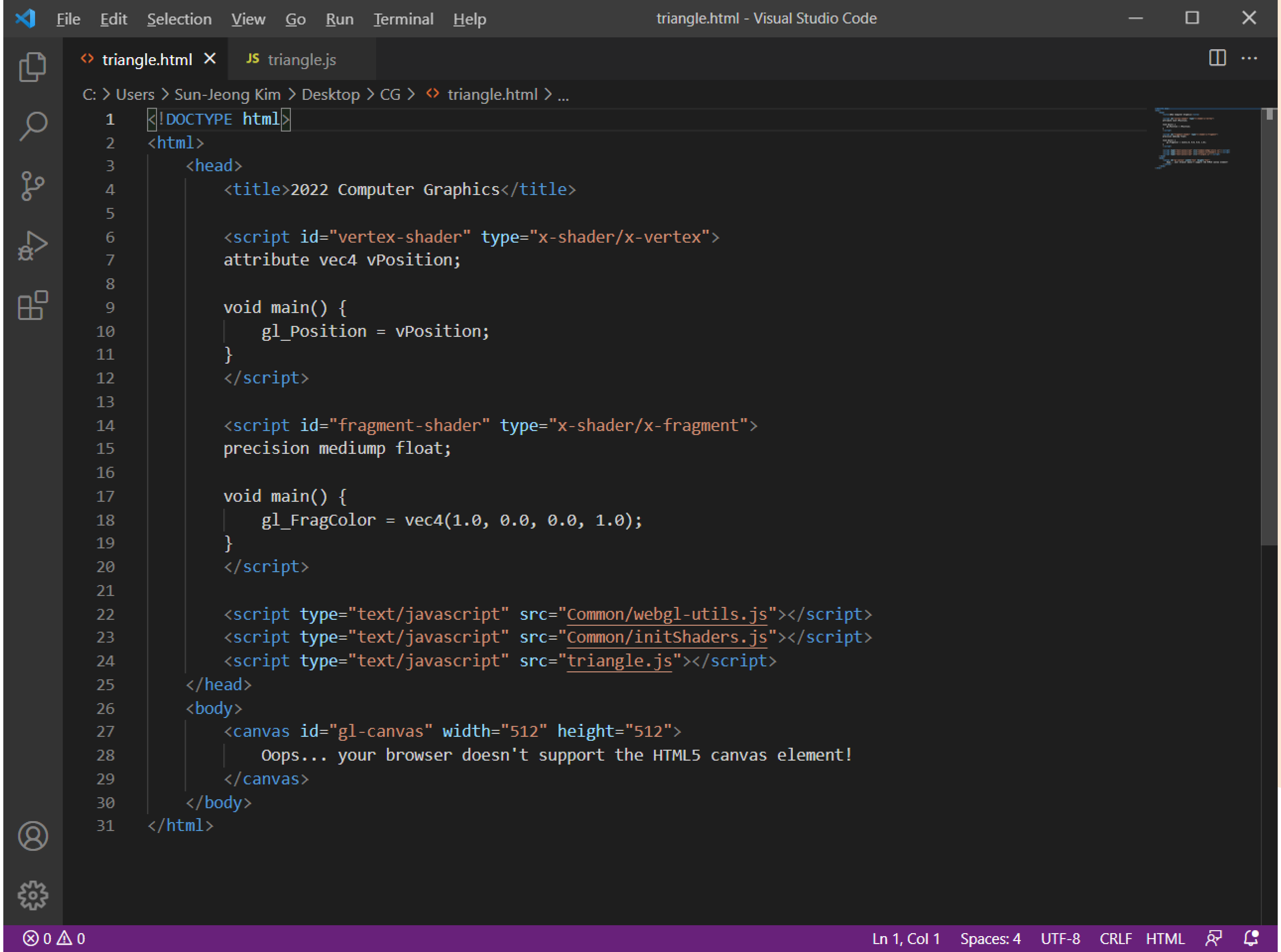


2022 Computer Graphics
파일 | C:/Users/Sun-Jeong%20Kim/Deskto...

연습문제 (3)

- 다음과 같은 삼각형을 그리시오.

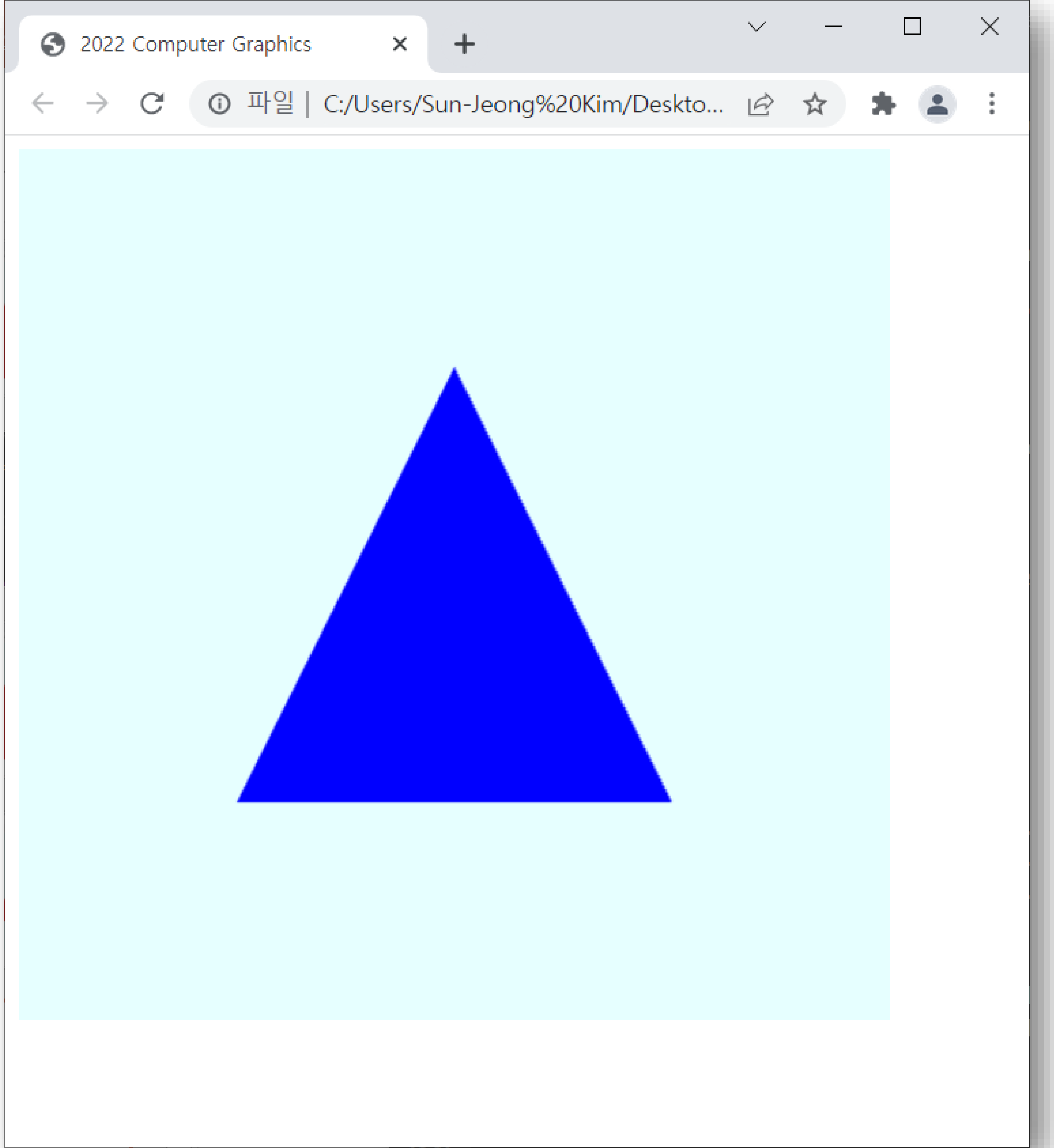



File Edit Selection View Go Run Terminal Help
triangle.html - Visual Studio Code
triangle.html
triangle.js
C: > Users > Sun-Jeong Kim > Desktop > CG > triangle.html > ...
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML
0 0

```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>
```

연습문제 (4)

- 삼각형의 색상을 변경하시오.


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