

## Understanding The Problem

First of all, redbox.js and assignment2.html codes were analyzed. It is the fragment shader that determines the main color element in the WebGL program. `gl.shaderSource(fragmentShader, fragmentShaderSource)`; With the code, we can understand that the color element is actually called from redbox.js as the source shader in Assignment2.html.

## Focusing On Redbox.js

At this point we can see that `fragmentShaderSource` is actually defined. We can see that `FragColor` takes color exactly with the RGB value defined in the vector. Therefore, we can understand that in order to change the color in the box from red to blue, we actually need to change the RGB value here.

## Changing RGB Values

The first value in `vec4` indicates red, the second value indicates green, the third value indicates blue, and the fourth value indicates opacity. In the unchanged version of the code, the box is red because only the first and fourth values are 1, that is, the red value is actually 1, and the values of the other colors are 0. If we set the third value, the blue value, to 1 and the first value, the red value, to 0, this shader source will now be blue. After these changes, when we run `assignment2.html` through the browser, we see the blue box.

```
const fragmentShaderSource = `
precision mediump float;

void main() {
    gl_FragColor = vec4(0.0, 0.0, 1.0, 1.0);
}
`;
```

Change:

**Result:**

