

resetShader()

Restores the default shaders.

`resetShader()` deactivates any shaders previously applied by `shader()`.

Note: Shaders can only be used in WebGL mode.

Examples



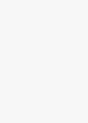
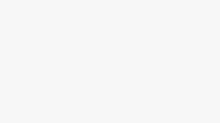
```
// Create a string with the vertex shader program.
// The vertex shader is called for each vertex.
let vertSrc = `
attribute vec3 aPosition;
attribute vec2 aTexCoord;
uniform mat4 uProjectionMatrix;
uniform mat4 uModelViewMatrix;
varying vec2 vTexCoord;

void main() {
    vTexCoord = aTexCoord;
    vec4 position = vec4(aPosition, 1.0);
    gl_Position = uProjectionMatrix * uModelViewMatrix *
position;
}

`;

// Create a string with the fragment shader program.
// The fragment shader is called for each pixel.
let fragSrc = `
precision mediump float;
varying vec2 vTexCoord;

void main() {
    vec2 uv = vTexCoord;
    vec3 color = vec3(uv.x, uv.y, min(uv.x + uv.y, 1.0));
}
```



This page is generated from the comments in [src/webgl/material.js](#). Please feel free to edit it and submit a pull request!

Related References

copyToContext

Copies the shader from one drawing context to another.

inspectHooks

Logs the hooks available in this shader, and their current implementation.

modify

Returns a new shader, based on the original, but with custom snippets of shader code replacing default behaviour.

setUniform

Sets the shader's uniform (global) variables.

