

setUniform()

Sets the shader's uniform (global) variables.

Shader programs run on the computer's graphics processing unit (GPU). They live in part of the computer's memory that's completely separate from the sketch that runs them. Uniforms are global variables within a shader program. They provide a way to pass values from a sketch running on the CPU to a shader program running on the GPU.

The first parameter, `uniformName`, is a string with the uniform's name. For the shader above, `uniformName` would be `'r'`.

The second parameter, `data`, is the value that should be used to set the uniform. For example, calling `myShader.setUniform('r', 0.5)` would set the `r` uniform in the shader above to `0.5`. `data` should match the uniform's type. Numbers, strings, booleans, arrays, and many types of images can all be passed to a shader with `setUniform()`.

Examples



▶

■

// Note: A "uniform" is a global variable within a shader program.

// Create a string with the vertex shader program.
// The vertex shader is called for each vertex.
let vertSrc = `
precision highp float;
uniform mat4 uModelViewMatrix;
uniform mat4 uProjectionMatrix;

attribute vec3 aPosition;
attribute vec2 aTexCoord;
varying vec2 vTexCoord;

void main() {
 vTexCoord = aTexCoord;
 vec4 positionVec4 = vec4(aPosition, 1.0);
 gl_Position = uProjectionMatrix * uModelViewMatrix *
 positionVec4;
}
`;

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

▶

■

// Note: A "uniform" is a global variable within a shader program.

// Create a string with the vertex shader program.
// The vertex shader is called for each vertex.
let vertSrc = `
precision highp float;
uniform mat4 uModelViewMatrix;
uniform mat4 uProjectionMatrix;

attribute vec3 aPosition;
attribute vec2 aTexCoord;
varying vec2 vTexCoord;

void main() {
 vTexCoord = aTexCoord;
 vec4 positionVec4 = vec4(aPosition, 1.0);
 gl_Position = uProjectionMatrix * uModelViewMatrix *
 positionVec4;
}
`;

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

▶

■

// Note: A "uniform" is a global variable within a shader program.

// Create a string with the vertex shader program.
// The vertex shader is called for each vertex.
let vertSrc = `
precision highp float;
uniform mat4 uModelViewMatrix;
uniform mat4 uProjectionMatrix;

attribute vec3 aPosition;
attribute vec2 aTexCoord;
varying vec2 vTexCoord;

void main() {
 vTexCoord = aTexCoord;
 vec4 positionVec4 = vec4(aPosition, 1.0);
 gl_Position = uProjectionMatrix * uModelViewMatrix *
 positionVec4;
}
`;

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

Syntax

setUniform(uniformName, data)

Parameters

<code>uniformName</code>	String: name of the uniform. Must match the name used in the vertex and fragment shaders.
<code>data</code>	<code>Boolean</code> <code>Number</code> <code>Number[]</code> <code>p5.Image</code> <code>p5.Graphics</code> <code>p5.MediaElement</code> <code>p5.Texture</code> : value to assign to the uniform. Must match the uniform's data type.

This page is generated from the comments in [src/webgl/p5.Shader.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.
---	--	---	---