copyToContext()

Copies the shader from one drawing context to another.

Each `p5.Shader` object must be compiled by calling shader() before it can run. Compilation happens in a drawing context which is usually the main canvas or an instance of p5.Graphics. A shader can only be used in the context where it was compiled. The `copyToContext()` method compiles the shader again and copies it to another drawing context where it can be reused.

The parameter, `context`, is the drawing context where the shader will be used. The shader can be copied to an instance of p5.Graphics, as in `myShader.copyToContext(pg)`. The shader can also be copied from a p5.Graphics object to the main canvas using the `window` variable, as in `myShader.copyToContext(window)`.

Note: A p5.Shader object created with createShader(), createFilterShader(), or loadShader() can be used directly with a p5.Framebuffer object created with createFramebuffer(). Both objects have the same context as the main canvas.

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;
```


Syntax

```
copyToContext(context)
```

Parameters

context	p5|p5.Graphics: WebGL context for the copied shader.

Returns

p5.Shader: new shader compiled for the target context.

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.