

metalness()

Sets the amount of "metalness" of a [specularMaterial\(\)](#).

`metalness()` can make materials appear more metallic. It affects the way materials reflect light sources including affects the way materials reflect light sources including `directionalLight()`, `pointLight()`, `spotLight()`, and `imageLight()`.

The parameter, `metallic`, is a number that sets the amount of metalness. `metallic` must be greater than 1, which is its default value. Higher values, such as `metalness(100)`, make specular materials appear more metallic.

Examples



```
function setup() {
  createCanvas(100, 100, WEBGL);

  describe(
    'Two blue spheres drawn on a gray background. White light reflects from their surfaces as the mouse moves. The right sphere is more metallic than the left sphere.'
  );
}

function draw() {
  background(200);

  // Turn on an ambient light.
  ambientLight(200);

  // Get the mouse's coordinates.
  let mx = mouseX - 50;
  let my = mouseY - 50;

  // Turn on a white point light that follows the mouse.
  pointLight(255, 255, 255, mx, my, 50);

  // Style the spheres.
  noStroke();
  fill(30, 30, 255);
  specularMaterial(255);
  shininess(20);

  // Draw the left sphere with low metalness.
```

// Click and drag the mouse to view the spheres from different angles.

```
let img;

function preload() {
  img =
  loadImage('/assets/outdoor_spheremap.jpg');
}

function setup() {
  createCanvas(100, 100, WEBGL);

  describe(
    'Two spheres floating above a landscape. The surface of the spheres reflect the landscape. The right sphere is more reflective than the left sphere.'
  );
}

function draw() {
  // Add the panorama.
  panorama(img);

  // Enable orbiting with the mouse.
  orbitControl();

  // Use the image as a light source.
  imageLight(img);

  // Style the spheres.
  noStroke();
  specularMaterial(50);
```

Syntax

```
metalness(metallic)
```

Parameters

`metallic` Number: amount of metalness.

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.

