ambientMaterial()

Sets the ambient color of shapes' surface material.

The `ambientMaterial()` color sets the components of the ambientLight() color that shapes will reflect. For example, calling `ambientMaterial(255, 255, 0)` would cause a shape to reflect red and green light, but not blue light.

`ambientMaterial()` can be called three ways with different parameters to set the material's color.

The first way to call `ambientMaterial()` has one parameter, `gray`. Grayscale values between 0 and 255, as in `ambientMaterial(50)`, can be passed to set the material's color. Higher grayscale values make shapes appear brighter.

The second way to call `ambientMaterial()` has one parameter, `color`. A p5.Color object, an array of color values, or a CSS color string, as in `ambientMaterial('magenta')`, can be passed to set the material's color.

The third way to call `ambientMaterial()` has three parameters, `v1`, `v2`, and `v3`. RGB, HSB, or HSL values, as in `ambientMaterial(255, 0, 0)`, can be passed to set the material's colors. Color values will be interpreted using the current colorMode().

Note: `ambientMaterial()` can only be used in WebGL mode.

Examples

```
// Click and drag the mouse to view the scene from differe
angles.

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

  describe('A magenta cube drawn on a gray background.');
}

function draw() {
  background(200);

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

  // Turn on a magenta ambient light.
  ambientLight(255, 0, 255);

  // Draw the box.
  box();
}
```

```
// Click and drag the mouse to view the scene from differe
angles.

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

  describe('A purple cube drawn on a gray background.');
}

function draw() {
  background(200);

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

  // Turn on a magenta ambient light.
  ambientLight(255, 0, 255);

  // Add a dark gray ambient material.
  ambientMaterial(150);

  // Draw the box.
  box();
}
```

```
// Click and drag the mouse to view the scene from differe
angles.

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

  describe('A red cube drawn on a gray background.');
}

function draw() {
  background(200);

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

  // Turn on a magenta ambient light.
  ambientLight(255, 0, 255);

  // Add a yellow ambient material using RGB values.
  ambientMaterial(255, 255, 0);

  // Draw the box.
  box();
}
```

```
// Click and drag the mouse to view the scene from differe
angles.

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

  describe('A red cube drawn on a gray background.');
}

function draw() {
  background(200);

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

  // Turn on a magenta ambient light.
  ambientLight(255, 0, 255);

  // Add a yellow ambient material using a p5.Color object.
  let c = color(255, 255, 0);
  ambientMaterial(c);

  // Draw the box.
  box();
}
```

```
// Click and drag the mouse to view the scene from differe
angles.

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

  describe('A red cube drawn on a gray background.');
}

function draw() {
  background(200);

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

  // Turn on a magenta ambient light.
  ambientLight(255, 0, 255);

  // Add a yellow ambient material using a color string.
  ambientMaterial('yellow');

  // Draw the box.
  box();
}
```

```
// Click and drag the mouse to view the scene from differe
angles.

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

  describe('A yellow cube drawn on a gray background.');
}

function draw() {
  background(200);

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

  // Turn on a white ambient light.
  ambientLight(255, 255, 255);

  // Add a yellow ambient material using a color string.
  ambientMaterial('yellow');

  // Draw the box.
  box();
}
```

Syntax

```
ambientMaterial(v1, v2, v3)
```

```
ambientMaterial(gray)
```

```
ambientMaterial(color)
```

Parameters

v1	Number: red or hue value in the current colorMode().
v2	Number: green or saturation value in the current colorMode().
v3	Number: blue, brightness, or lightness value in the current colorMode().
gray	Number: grayscale value between 0 (black) and 255 (white).
color	p5.Color|Number[]|String: color as a p5.Color object, an array of color values, or a CSS string.

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.