ambientLight()

Creates a light that shines from all directions.

Ambient light does not come from one direction. Instead, 3D shapes are lit evenly from all sides. Ambient lights are almost always used in combination with other types of lights.

There are three ways to call `ambientLight()` with optional parameters to set the light's color.

The first way to call `ambientLight()` has two parameters, `gray` and `alpha`. `alpha` is optional. Grayscale and alpha values between 0 and 255 can be passed to set the ambient light's color, as in `ambientLight(50)` or `ambientLight(50, 30)`.

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

The third way to call `ambientLight()` has four parameters, `v1`, `v2`, `v3`, and `alpha`. `alpha` is optional. RGBA, HSBA, or HSLA values can be passed to set the ambient light's colors, as in `ambientLight(255, 0, 0)` or `ambientLight(255, 0, 0, 30)`. Color values will be interpreted using the current colorMode().

Examples

```
// Click and drag the mouse to view the scene from differe
angles.
// Double-click the canvas to turn on the light.

let isLit = false;

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

  describe('A sphere drawn against a gray background. The
sphere appears to change color when the user double-clicks.');
}

function draw() {
  background(200);

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

  // Control the light.
  if (isLit === true) {
    // Use a grayscale value of 80.
    ambientLight(80);
  }

  // Draw the sphere.
  sphere(30);
}

// Turn on the ambient light when the user double-clicks.
function doubleClicked() {
```

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

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

  describe('A faded magenta sphere drawn against a gray
background.');
}

function draw() {
  background(200);

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

  // Turn on the lights.
  // Use a p5.Color object.
  let c = color('orchid');
  ambientLight(c);

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

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

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

  describe('A faded magenta sphere drawn against a gray
background.');
}

function draw() {
  background(200);

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

  // Turn on the lights.
  // Use a CSS color string.
  ambientLight('#DA70D6');

  // Draw the sphere.
  sphere(30);
}
```

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

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

  describe('A faded magenta sphere drawn against a gray
background.');
}

function draw() {
  background(200);

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

  // Turn on the lights.
  // Use RGB values
  ambientLight(218, 112, 214);

  // Draw the sphere.
  sphere(30);
}
```

Syntax

```
ambientLight(v1, v2, v3, [alpha])
```

```
ambientLight(gray, [alpha])
```

```
ambientLight(value)
```

```
ambientLight(values)
```

```
ambientLight(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().
alpha	Number: alpha (transparency) value in the current colorMode().
gray	Number: grayscale value between 0 and 255.
value	String: color as a CSS string.
values	Number[]: color as an array of RGBA, HSBA, or HSLA values.
color	p5.Color: color as a p5.Color object.

This page is generated from the comments in src/webgl/light.js . Please feel free to edit it and submit a pull request!

Related References

ambientLight
Creates a light that shines from all directions.

directionalLight
Creates a light that shines in one direction.

imageLight
Creates an ambient light from an image.

lightFalloff
Sets the falloff rate for pointLight() and spotLight().