pointLight()

Creates a light that shines from a point in all directions.

Point lights are like light bulbs that shine in all directions. They can be placed at different positions to achieve different lighting effects. A maximum of 5 point lights can be active at once.

There are four ways to call `pointLight()` with parameters to set the light's color and position.

The first way to call `pointLight()` has six parameters. The first three parameters, `v1`, `v2`, and `v3`, set the light's color using the current colorMode(). The last three parameters, `x`, `y`, and `z`, set the light's position. For example, `pointLight(255, 0, 0, 50, 0, 0)` creates a red `(255, 0, 0)` light that shines from the coordinates `(50, 0, 0)`.

The second way to call `pointLight()` has four parameters. The first three parameters, `v1`, `v2`, and `v3`, set the light's color using the current colorMode(). The last parameter, position sets the light's position using a p5.Vector object. For example, `pointLight(255, 0, 0, lightPos)` creates a red `(255, 0, 0)` light that shines from the position set by the `lightPos` vector.

The third way to call `pointLight()` has four parameters. The first parameter, `color`, sets the light's color using a p5.Color object or an array of color values. The last three parameters, `x`, `y`, and `z`, set the light's position. For example, `directionalLight(myColor, 50, 0, 0)` creates a light that shines from the coordinates `(50, 0, 0)` with the color value of `myColor`.

The fourth way to call `pointLight()` has two parameters. The first parameter, `color`, sets the light's color using a p5.Color object or an array of color values. The second parameter, `position`, sets the light's position using a p5.Vector object. For example, `directionalLight(myColor, lightPos)` creates a light that shines from the position set by the `lightPos` vector with the color value of `myColor`.

Examples

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

let isLit = false;

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

  describe('A sphere drawn on a gray background. A red light
starts shining from above when the user double-clicks.');
}

function draw() {
  background(200);

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

  // Control the light.
  if (isLit === true) {
    // Add a red point light from above.
    // Use RGB values and XYZ coordinates.
    pointLight(255, 0, 0, 0, -150, 0);
  }

  // Style the sphere.
  noStroke();

  // Draw the sphere.
```

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

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

  describe('A sphere drawn on a gray background. The top of
the sphere appears bright red. The color gets darker toward
the bottom.');
}

function draw() {
  background(200);

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

  // Add a red point light from above.
  // Use a p5.Color object and XYZ directions.
  let c = color(255, 0, 0);
  pointLight(c, 0, -150, 0);

  // Style the sphere.
  noStroke();

  // 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 sphere drawn on a gray background. The top of
the sphere appears bright red. The color gets darker toward
the bottom.');
}

function draw() {
  background(200);

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

  // Add a red point light from above.
  // Use a p5.Color object and a p5.Vector object.
  let c = color(255, 0, 0);
  let lightPos = createVector(0, -150, 0);
  pointLight(c, lightPos);

  // Style the sphere.
  noStroke();

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

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

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

  describe('Four spheres arranged in a square and drawn on a
gray background. The spheres appear bright red toward the
center of the square. The color gets darker toward the corners
of the square.');
}

function draw() {
  background(200);

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

  // Add a red point light that points to the center of the
scene.
  // Use a p5.Color object and a p5.Vector object.
  let c = color(255, 0, 0);
  let lightPos = createVector(0, 0, 65);
  pointLight(c, lightPos);

  // Style the spheres.
```

Syntax

```
pointLight(v1, v2, v3, x, y, z)
```

```
pointLight(v1, v2, v3, position)
```

```
pointLight(color, x, y, z)
```

```
pointLight(color, position)
```

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().
x	Number: x-coordinate of the light.
y	Number: y-coordinate of the light.
z	Number: z-coordinate of the light.
position	p5.Vector: position of the light as a p5.Vector object.
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/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().