

spotLight()

Creates a light that shines from a point in one direction.

Spot lights are like flashlights that shine in one direction creating a cone of light. The shape of the cone can be controlled using the angle and concentration parameters. A maximum of 5 spot lights can be active at once.

There are eight ways to call `spotLight()` with parameters to set the light's color, position, direction. For example, `spotLight(255, 0, 0, 0, 0, 0, 1, 0, 0)` creates a red `(255, 0, 0)` light at the origin `(0, 0, 0)` that points to the right `(1, 0, 0)`.

The angle parameter is optional. It sets the radius of the light cone. For example, `spotLight(255, 0, 0, 0, 0, 0, 1, 0, PI / 16)` creates a red `(255, 0, 0)` light at the origin `(0, 0, 0)` that points to the right `(1, 0, 0)` with an angle of `PI / 16` radians. By default, `angle` is `PI / 3` radians.

The concentration parameter is also optional. It focuses the light towards the center of the light cone. For example, `spotLight(255, 0, 0, 0, 0, 0, 1, 0, 0, PI / 16, 50)` creates a red `(255, 0, 0)` light at the origin `(0, 0, 0)` that points to the right `(1, 0, 0)` with an angle of `PI / 16` radians at concentration of 50. By default, `concentration` is 100.

Examples



```
// Click and drag the mouse to view the scene from different angles.
// Double-click to adjust the spotlight.
```

```
let isLit = false;

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

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

function draw() {
  background(50);

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

  // Turn on the lights.
  lights();

  // Control the spotlight.
  if (isLit === true) {
    // Add a red spot light that shines into the screen.
    // Set its angle to PI / 32 radians.
    spotLight(255, 0, 0, 0, 0, 100, 0, 0, -1, PI / 32);
  }
}
```



```
// Click and drag the mouse to view the scene from different angles.
// Double-click to adjust the spotlight.
```

```
let isLit = false;

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

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

function draw() {
  background(50);

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

  // Turn on the lights.
  lights();

  // Control the spotlight.
  if (isLit === true) {
    // Add a red spot light that shines into the screen.
    // Set its angle to PI / 3 radians (default).
    // Set its concentration to 1000.
    let c = color(255, 0, 0);
    spotLight(c, 0, 0, 0, 0, 0, 1000, 0, 0, -1, PI / 3);
  }
}
```

Syntax

```
spotLight(v1, v2, v3, x, y, z, rx, ry, rz, [angle], [concentration])
```



```
spotLight(color, position, direction, [angle], [concentration])
```



```
spotLight(v1, v2, v3, position, direction, [angle], [concentration])
```



```
spotLight(color, x, y, z, direction, [angle], [concentration])
```



```
spotLight(color, position, rx, ry, rz, [angle], [concentration])
```



```
spotLight(v1, v2, v3, position, rx, ry, rz, [angle], [concentration])
```



```
spotLight(color, x, y, z, rx, ry, rz, [angle], [concentration])
```



Parameters

v1	Number: red or hue value in the current <code>colorMode()</code> .
v2	Number: green or saturation value in the current <code>colorMode()</code> .
v3	Number: blue, brightness, or lightness value in the current <code>colorMode()</code> .
x	Number: x-coordinate of the light.
y	Number: y-coordinate of the light.
z	Number: z-coordinate of the light.
rx	Number: x-component of light direction between -1 and 1.
ry	Number: y-component of light direction between -1 and 1.
rz	Number: z-component of light direction between -1 and 1.
angle	Number: angle of the light cone. Defaults to <code>PI / 3</code> .
concentration	Number: concentration of the light. Defaults to 100.
color	p5.Color Number[] String: color as a <code>p5.Color</code> object, an array of color values, or a CSS string.
position	p5.Vector: position of the light as a <code>p5.Vector</code> object.
direction	p5.Vector: direction of light as a <code>p5.Vector</code> 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()`.



Resources

[Reference](#)

[Tutorials](#)

[Examples](#)

[Contribute](#)

[Community](#)

[About](#)

[Start Coding](#)

[Donate](#)

Information

[Download](#)

[Contact](#)

[Copyright](#)

[Privacy Policy](#)

[Terms of Use](#)

Socials

[GitHub](#) ↗

[Instagram](#) ↗

X ↗

[YouTube](#) ↗

[Discord](#) ↗

[Forum](#) ↗



Donate Today! Support p5.js and the Processing Foundation.

