

# camera()

Sets the position and orientation of the current camera in a 3D sketch.

`camera()` allows objects to be viewed from different angles. It has nine parameters that are all optional.

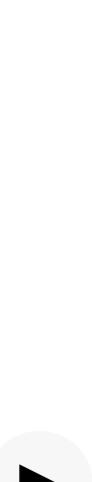
The first three parameters, `x`, `y`, and `z`, are the coordinates of the camera's position. For example, calling `camera(0, 0, 0)` places the camera at the origin `(0, 0, 0)`. By default, the camera is placed at `(0, 0, 800)`.

The next three parameters, `centerX`, `centerY`, and `centerZ` are the coordinates of the point where the camera faces. For example, calling `camera(0, 0, 0, 10, 20, 30)` places the camera at the origin `(0, 0, 0)` and points it at `(10, 20, 30)`. By default, the camera points at the origin `(0, 0, 0)`.

The last three parameters, `upX`, `upY`, and `upZ` are the components of the "up" vector. The "up" vector orients the camera's y-axis. For example, calling `camera(0, 0, 0, 10, 20, 30, 0, -1, 0)` places the camera at the origin `(0, 0, 0)`, points it at `(10, 20, 30)`, and sets the "up" vector to `(0, -1, 0)` which is like holding it upside-down. By default, the "up" vector is `(0, 1, 0)`.

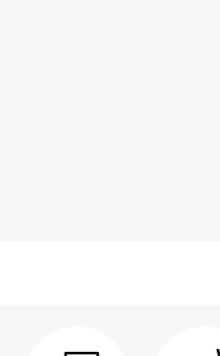
Note: `camera()` can only be used in WebGL mode.

## Examples



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

  describe('A white cube on a gray background.');
}
```



```
function draw() {
  background(200);

  // Move the camera to the top-right.
  camera(200, -400, 800);

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

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

  describe('A white cube appears to sway left and right on a gray background.');
}
```

```
function draw() {
  background(200);

  // Calculate the camera's x-coordinate.
  let x = 400 * cos(frameCount * 0.01);

  // Orbit the camera around the box.
  camera(x, -400, 800);

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

```
// Adjust the range sliders to change the camera's position
```

```
let xSlider;
let ySlider;
let zSlider;
```

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

  // Create slider objects to set the camera's coordinates.
  xSlider = createSlider(-400, 400, 400);
  xSlider.position(0, 100);
  xSlider.size(100);
  ySlider = createSlider(-400, 400, -200);
  ySlider.position(0, 120);
  ySlider.size(100);
  zSlider = createSlider(0, 1600, 800);
  zSlider.position(0, 140);
  zSlider.size(100);
}
```

```
describe(
  'A white cube drawn against a gray background. Three range sliders appear beneath the image. The camera position changes when the user moves the sliders.'
)
```

## Syntax

```
camera([x], [y], [z], [centerX], [centerY], [centerZ], [upX], [upY], [upZ])
```

## Parameters

<code>x</code>	Number: x-coordinate of the camera. Defaults to 0.
<code>y</code>	Number: y-coordinate of the camera. Defaults to 0.
<code>z</code>	Number: z-coordinate of the camera. Defaults to 800.
<code>centerX</code>	Number: x-coordinate of the point the camera faces. Defaults to 0.
<code>centerY</code>	Number: y-coordinate of the point the camera faces. Defaults to 0.
<code>centerZ</code>	Number: z-coordinate of the point the camera faces. Defaults to 0.
<code>upX</code>	Number: x-component of the camera's "up" vector. Defaults to 0.
<code>upY</code>	Number: y-component of the camera's "up" vector. Defaults to 1.
<code>upZ</code>	Number: z-component of the camera's "up" vector. Defaults to 0.

This page is generated from the comments in <src/webgl/p5.Camera.js>. Please feel free to edit it and submit a pull request!

## Related References

`camera` Sets the position and orientation of the camera.

`centerX` The x-coordinate of the place where the camera looks.

`centerY` The y-coordinate of the place where the camera looks.

`centerZ` The z-coordinate of the place where the camera looks.

