ortho()

Sets an orthographic projection for the current camera in a 3D sketch.

In an orthographic projection, shapes with the same size always appear the same size, regardless of whether they are near or far from the camera.

`ortho()` changes the camera's perspective by changing its viewing frustum from a truncated pyramid to a rectangular prism. The camera is placed in front of the frustum and views everything between the frustum's near plane and its far plane. `ortho()` has six optional parameters to define the frustum.

The first four parameters, `left`, `right`, `bottom`, and `top`, set the coordinates of the frustum's sides, bottom, and top. For example, calling `ortho(-100, 100, 200, -200)` creates a frustum that's 200 pixels wide and 400 pixels tall. By default, these coordinates are set based on the sketch's width and height, as in `ortho(-width / 2, width / 2, -height / 2, height / 2)`.

The last two parameters, `near` and `far`, set the distance of the frustum's near and far plane from the camera. For example, calling `ortho(-100, 100, 200, 200, 50, 1000)` creates a frustum that's 200 pixels wide, 400 pixels tall, starts 50 pixels from the camera, and ends 1,000 pixels from the camera. By default, `near` and `far` are set to 0 and `max(width, height) + 800`, respectively.

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

Examples

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

  describe('A row of tiny, white cubes on a gray background.
All the cubes appear the same size.');
}

function draw() {
  background(200);

  // Apply an orthographic projection.
  ortho();

  // Translate the origin toward the camera.
  translate(-10, 10, 600);

  // Rotate the coordinate system.
  rotateY(-0.1);
  rotateX(-0.1);

  // Draw the row of boxes.
  for (let i = 0; i < 6; i += 1) {
    translate(0, 0, -40);
    box(10);
  }
}
```

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

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

function draw() {
  background(200);

  // Apply an orthographic projection.
  // Center the frustum.
  // Set its width and height to 20.
  // Place its near plane 300 pixels from the camera.
  // Place its far plane 350 pixels from the camera.
  ortho(-10, 10, -10, 10, 300, 350);

  // Translate the origin toward the camera.
  translate(-10, 10, 600);

  // Rotate the coordinate system.
  rotateY(-0.1);
  rotateX(-0.1);

  // Draw the row of boxes.
  for (let i = 0; i < 6; i += 1) {
    translate(0, 0, -40);
    box(10);
  }
```

Syntax

```
ortho([left], [right], [bottom], [top], [near], [far])
```

Parameters

left	Number: x-coordinate of the frustum's left plane. Defaults to `-width / 2`.
right	Number: x-coordinate of the frustum's right plane. Defaults to `width / 2`.
bottom	Number: y-coordinate of the frustum's bottom plane. Defaults to `height / 2`.
top	Number: y-coordinate of the frustum's top plane. Defaults to `-height / 2`.
near	Number: z-coordinate of the frustum's near plane. Defaults to 0.
far	Number: z-coordinate of the frustum's far plane. Defaults to `max(width, height) + 800`.

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 y-coordinate of the place where the camera looks.