

# frustum()

Sets the frustum of the current camera in a 3D sketch.

In a frustum projection, shapes that are further from the camera appear smaller than shapes that are near the camera. This technique, called foreshortening, creates realistic 3D scenes.

`frustum()` changes the default camera's perspective by changing its viewing frustum. The frustum is the volume of space that's visible to the camera. The frustum's shape is a pyramid with its top cut off. The camera is placed where the top of the pyramid should be and points towards the base of the pyramid. It views everything within 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 `frustum(-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 / 20, width / 20, height / 20, -height / 20)`.

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` is set to `0.1 * 800`, which is 1/10th the default distance between the camera and the origin. `far` is set to `10 * 800`, which is 10 times the default distance between the camera and the origin.

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

## Examples

```

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

      describe('A row of white cubes on a gray background.');
    }

    function draw() {
      background(200);

      // Apply the default frustum projection.
      frustum();

      // 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);

      // Adjust the frustum.
      // Center it.
      // Set its width and height to 20 pixels.
      // Place its near plane 300 pixels from the camera.
      // Place its far plane 350 pixels from the camera.
      frustum(-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

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

## Parameters

<code>left</code>	Number: x-coordinate of the frustum's left plane. Defaults to <code>-width / 20</code> .
<code>right</code>	Number: x-coordinate of the frustum's right plane. Defaults to <code>width / 20</code> .
<code>bottom</code>	Number: y-coordinate of the frustum's bottom plane. Defaults to <code>height / 20</code> .
<code>top</code>	Number: y-coordinate of the frustum's top plane. Defaults to <code>-height / 20</code> .
<code>near</code>	Number: z-coordinate of the frustum's near plane. Defaults to <code>0.1 * 800</code> .
<code>far</code>	Number: z-coordinate of the frustum's far plane. Defaults to <code>10 * 800</code> .

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

<code>camera</code>	Sets the position and orientation of the camera.	<code>centerX</code>	The x-coordinate of the place where the camera looks.	<code>centerY</code>	The y-coordinate of the place where the camera looks.	<code>centerZ</code>	The z-coordinate of the place where the camera looks.
---------------------	--	----------------------	---	----------------------	---	----------------------	---

