normalize()

Transforms the geometry's vertices to fit snugly within a 100×100×100 box centered at the origin.

Calling `myGeometry.normalize()` translates the geometry's vertices so that they're centered at the origin `(0, 0, 0)`. Then it scales the vertices so that they fill a 100×100×100 box. As a result, small geometries will grow and large geometries will shrink.

Note: `myGeometry.normalize()` only works when called in the setup() function.

Examples

```
let myGeometry;

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

  // Create a very small torus.
  beginGeometry();
  torus(1, 0.25);
  myGeometry = endGeometry();

  // Normalize the torus so its vertices fill
  // the range [-100, 100].
  myGeometry.normalize();

  describe('A white torus rotates slowly against a dark gray background.');
}

function draw() {
  background(50);

  // Turn on the lights.
  lights();

  // Rotate around the y-axis.
  rotateY(frameCount * 0.01);

  // Style the torus.
  noStroke();

  // Draw the torus.
  model(myGeometry);
}
```

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

Related References

calculateBoundingBox
Calculates the position and size of the smallest box that contains the geometry.

clearColors
Removes the geometry’s internal colors.

computeFaces
Computes the geometry's faces using its vertices.

computeNormals
Calculates the normal vector for each vertex on the geometry.