vertexNormals

An array with the vectors that are normal to the geometry's vertices.

A face's orientation is defined by its *normal vector* which points out of the face and is normal (perpendicular) to the surface. Calling `myGeometry.computeNormals()` first calculates each face's normal vector. Then it calculates the normal vector for each vertex by averaging the normal vectors of the faces surrounding the vertex. The vertex normals are stored as p5.Vector objects in the `myGeometry.vertexNormals` array.

Examples

```
// Click and drag the mouse to view the scene from differe
angles.

let myGeometry;

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

  // Create a p5.Geometry object.
  beginGeometry();
  torus(30, 15, 10, 8);
  myGeometry = endGeometry();

  // Compute the vertex normals.
  myGeometry.computeNormals();

  describe(
    'A white torus rotates against a dark gray background. Red
lines extend outward from its vertices.'
  );
}

function draw() {
  background(50);

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

  // Turn on the lights.
  lights();

  // Rotate the coordinate system.
  rotateY(frameCount * 0.01);
```

```
// Click and drag the mouse to view the scene from differe
angles.

let myGeometry;

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

  // Create a p5.Geometry object.
  myGeometry = new p5.Geometry();

  // Create p5.Vector objects to position the vertices.
  let v0 = createVector(-40, 0, 0);
  let v1 = createVector(0, -40, 0);
  let v2 = createVector(0, 40, 0);
  let v3 = createVector(40, 0, 0);

  // Add the vertices to the p5.Geometry object's vertices
array.
  myGeometry.vertices.push(v0, v1, v2, v3);

  // Compute the faces array.
  myGeometry.computeFaces();

  // Compute the surface normals.
  myGeometry.computeNormals();
```

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.