sphere()

Draws a sphere.

A sphere is a 3D shape with triangular faces that connect to form a round surface. Spheres with few faces look like crystals. Spheres with many faces have smooth surfaces and look like balls.

The first parameter, `radius`, is optional. If a `Number` is passed, as in `sphere(20)`, it sets the radius of the sphere. By default, `radius` is 50.

The second parameter, `detailX`, is also optional. If a `Number` is passed, as in `sphere(20, 5)`, it sets the number of triangle subdivisions to use along the x-axis. All 3D shapes are made by connecting triangles to form their surfaces. By default, `detailX` is 24.

The third parameter, `detailY`, is also optional. If a `Number` is passed, as in `sphere(20, 5, 2)`, it sets the number of triangle subdivisions to use along the y-axis. All 3D shapes are made by connecting triangles to form their surfaces. By default, `detailY` is 16.

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

Examples

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

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

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

function draw() {
  background(200);

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

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

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

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

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

function draw() {
  background(200);

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

  // Draw the sphere.
  // Set its radius to 30.
  sphere(30);
}
```

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

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

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

function draw() {
  background(200);

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

  // Draw the sphere.
  // Set its radius to 30.
  // Set its detailX to 6.
  sphere(30, 6);
}
```

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

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

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

function draw() {
  background(200);

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

  // Draw the sphere.
  // Set its radius to 30.
  // Set its detailX to 24.
  // Set its detailY to 4.
  sphere(30, 24, 4);
}
```

Syntax

```
sphere([radius], [detailX], [detailY])
```

Parameters

radius	Number: radius of the sphere. Defaults to 50.
detailX	Integer: number of triangle subdivisions along the x-axis. Defaults to 24.
detailY	Integer: number of triangle subdivisions along the y-axis. Defaults to 16.

This page is generated from the comments in src/webgl/3d_primitives.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.