torus()

Draws a torus.

A torus is a 3D shape with triangular faces that connect to form a ring. Toruses with few faces look flattened. Toruses with many faces have smooth surfaces.

The first parameter, `radius`, is optional. If a `Number` is passed, as in `torus(30)`, it sets the radius of the ring. By default, `radius` is 50.

The second parameter, `tubeRadius`, is also optional. If a `Number` is passed, as in `torus(30, 15)`, it sets the radius of the tube. By default, `tubeRadius` is 10.

The third parameter, `detailX`, is also optional. If a `Number` is passed, as in `torus(30, 15, 5)`, it sets the number of edges used to draw the hole of the torus. Using more edges makes the hole look more like a circle. By default, `detailX` is 24.

The fourth parameter, `detailY`, is also optional. If a `Number` is passed, as in `torus(30, 15, 5, 7)`, it sets the number of triangle subdivisions to use while filling in the torus' height. By default, `detailY` is 16.

Note: `torus()` 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 torus on a gray background.');
}

function draw() {
  background(200);

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

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

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

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

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

function draw() {
  background(200);

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

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

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

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

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

function draw() {
  background(200);

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

  // Draw the torus.
  // Set its radius to 30 and tubeRadius to 15.
  torus(30, 15);
}
```

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

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

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

function draw() {
  background(200);

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

  // Draw the torus.
  // Set its radius to 30 and tubeRadius to 15.
  // Set its detailX to 5.
  torus(30, 15, 5);
}
```

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

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

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

function draw() {
  background(200);

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

  // Draw the torus.
  // Set its radius to 30 and tubeRadius to 15.
  // Set its detailX to 5.
  // Set its detailY to 3.
  torus(30, 15, 5, 3);
}
```

Syntax

```
torus([radius], [tubeRadius], [detailX], [detailY])
```

Parameters

radius	Number: radius of the torus. Defaults to 50.
tubeRadius	Number: radius of the tube. Defaults to 10.
detailX	Integer: number of edges that form the hole. 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.