

bezierVertex()

Adds a Bézier curve segment to a custom shape.

`bezierVertex()` adds a curved segment to custom shapes. The Bézier curves it creates are defined like those made by the `bezier()` function. `bezierVertex()` must be called between the `beginShape()` and `endShape()` functions. The curved segment uses the previous vertex as the first anchor point, so there must be at least one call to `vertex()` before `bezierVertex()` can be used.

The first four parameters, `x2`, `y2`, `x3`, and `y3`, set the curve's two control points. The control points "pull" the curve towards them.

The fifth and sixth parameters, `x4`, and `y4`, set the last anchor point. The last anchor point is where the curve ends.

Bézier curves can also be drawn in 3D using WebGL mode. The 3D version of `bezierVertex()` has eight arguments because each point has x-, y-, and z-coordinates.

Note: `bezierVertex()` won't work when an argument is passed to `beginShape()`.

Examples



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

  background(200);

  // Style the shape.
  noFill();

  // Start drawing the shape.
  beginShape();

  // Add the first anchor point.
  vertex(30, 20);

  // Add the Bézier vertex.
  bezierVertex(80, 0, 80, 75, 30, 75);

  // Stop drawing the shape.
  endShape();

  describe('A black C curve on a gray background.');
}
```

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

  background(200);

  // Draw the anchor points in black.
  stroke(0);
  strokeWeight(5);
  point(30, 20);
  point(30, 75);

  // Draw the control points in red.
  stroke(255, 0, 0);
  point(80, 0);
  point(80, 75);

  // Style the shape.
  noFill();
  stroke(0);
  strokeWeight(1);

  // Start drawing the shape.
  beginShape();

  // Add the first anchor point.
  vertex(30, 20);

  // Add the Bézier vertex.
  bezierVertex(80, 0, 80, 75, 30, 75);

  // Stop drawing the shape.
  endShape();

  // Draw red lines from the anchor points to the control
  // points.
  stroke(255, 0, 0);
}
```

// Click the mouse near the red dot in the top-right corner and drag to change the curve's shape.

```
let x2 = 80;
let y2 = 0;
let isChanging = false;
```

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

  describe(
    'A gray square with three curves. A black curve has two
    straight, red lines that extend from its ends. The endpoints
    of all the curves are marked with dots.'
  );
}
```

```
function draw() {
  background(200);

  // Draw the anchor points in black.
  stroke(0);
  strokeWeight(5);
  point(30, 20);
  point(30, 75);

  // Draw the control points in red.
  stroke(255, 0, 0);
  point(x2, y2);
  point(80, 75);

  // Style the shape.
}
```

/* Stroke the moons.

stroke(0);
 fill('lemonchiffon');

// Draw the first moon.
 beginShape();
 vertex(-20, -30, 0);
 bezierVertex(30, -50, 0, 30, -25, 0, -20, -30, 0);
 bezierVertex(0, 30, 0, 10, -25, 0, -20, -30, 0);
 endShape();

// Draw the second moon.
 beginShape();
 vertex(-20, 30, 0);
 bezierVertex(30, 50, 0, 30, 25, 0, 20, 30, 0);
 bezierVertex(0, 30, 0, 10, 25, 0, 20, 30, 0);
 endShape();

describe('A crescent moon shape drawn in white on a gray
 background.');
}

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

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

  describe('A crescent moon shape drawn in white on a blue
  background. When the mouse is dragged, the scene rotates
  and a second moon is revealed. The second moon is also
  rotated when the mouse is dragged.');
}
```

```
function draw() {
  background('midnightblue');
  orbitControl();

  // Style the moons.
  noStroke();
  fill('lemonchiffon');

  // Draw the first moon.
  beginShape();
  vertex(-20, -30, 0);
  bezierVertex(30, -50, 0, 30, -25, 0, -20, -30, 0);
  bezierVertex(0, 30, 0, 10, -25, 0, -20, -30, 0);
  endShape();
```

// Draw the second moon.
 beginShape();
 vertex(-20, 30, 0);
 bezierVertex(30, 50, 0, 30, 25, 0, 20, 30, 0);
 bezierVertex(0, 30, 0, 10, 25, 0, 20, 30, 0);
 endShape();

Syntax

```
bezierVertex(x2, y2, x3, y3, x4, y4)
```

```
bezierVertex(x2, y2, x3, y3, x4, y4, z4)
```

Parameters

`x2` Number: x-coordinate of the first control point.
`y2` Number: y-coordinate of the first control point.
`x3` Number: x-coordinate of the second control point.
`y3` Number: y-coordinate of the second control point.
`x4` Number: x-coordinate of the anchor point.
`y4` Number: y-coordinate of the first control point.
`z4` Number: z-coordinate of the second control point.

This page is generated from the comments in `src/core/shape/vertex.js`. Please feel free to edit it and submit a pull request.

Related References

`beginContour()` Begins creating a hole within a flat shape.
`beginShape()` Begins adding vertices to a custom shape.
`bezier()` Adds a Bézier curve segment to a custom shape.
`curveVertex()` Adds a spline curve segment to a custom shape.