

curveVertex()

Adds a spline curve segment to a custom shape.

`curveVertex()` adds a curved segment to custom shapes. The spline curves it creates are defined like those made by the `curve()` function. `curveVertex()` must be called between the `beginShape()` and `endShape()` functions.

Spline curves can form shapes and curves that slope gently. They're like cables that are attached to a set of points. Splines are defined by two anchor points and two control points. `curveVertex()` must be called at least four times between `beginShape()` and `endShape()` in order to draw a curve:

```
beginShape();

// Add the first control point.
curveVertex(84, 91);

// Add the anchor points to draw between.
curveVertex(68, 19);
curveVertex(21, 17);

// Add the second control point.
curveVertex(32, 91);

endShape();
```

The code snippet above would only draw the curve between the anchor points, similar to the `curve()` function. The segments between the control and anchor points can be drawn by calling `curveVertex()` with the coordinates of the control points:

```
beginShape();

// Add the first control point and draw a segment to it.
curveVertex(84, 91);
curveVertex(84, 91);

// Add the anchor points to draw between.
curveVertex(68, 19);
curveVertex(21, 17);

// Add the second control point.
curveVertex(32, 91);

// Uncomment the next line to draw the segment to the second control point.
// curveVertex(32, 91);

endShape();
```

The first two parameters, `x` and `y`, set the vertex's location. For example, calling `curveVertex(10, 10)` adds a point to the curve at (10, 10).

Spline curves can also be drawn in 3D using WebGL mode. The 3D version of `curveVertex()` has three arguments because each point has x-, y-, and z-coordinates. By default, the vertex's z-coordinate is set to 0.

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

Examples

▶

■

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

  background(200);

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

  // Start drawing the shape.
  beginShape();

  // Add the first control point.
  curveVertex(32, 91);

  // Add the anchor points.
  curveVertex(21, 17);
  curveVertex(68, 19);

  // Add the second control point.
  curveVertex(84, 91);

  // Stop drawing the shape.
  endShape();

  // Style the anchor and control points.
  strokeWeight(5);

  // Draw the anchor points in black.
  stroke(0);
  point(21, 17);
  point(68, 19);

  // Draw the control points in red.
  stroke(255, 0, 0);
}
```

▶

■

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

  background(200);

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

  // Start drawing the shape.
  beginShape();

  // Add the first control point and draw a segment to it.
  curveVertex(32, 91);
  curveVertex(32, 91);

  // Add the anchor points.
  curveVertex(21, 17);
  curveVertex(68, 19);

  // Add the second control point.
  curveVertex(84, 91);

  // Stop drawing the shape.
  endShape();

  // Style the anchor and control points.
  strokeWeight(5);

  // Draw the anchor points in black.
  stroke(0);
  point(21, 17);
  point(68, 19);

  // Draw the control points in red.
}
```

▶

■

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

  background(200);

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

  // Start drawing the shape.
  beginShape();

  // Add the first control point and draw a segment to it.
  curveVertex(32, 91);
  curveVertex(32, 91);

  // Add the anchor points.
  curveVertex(21, 17);
  curveVertex(68, 19);

  // Add the second control point and draw a segment to it.
  curveVertex(84, 91);
  curveVertex(84, 91);

  // Stop drawing the shape.
  endShape();

  // Style the anchor and control points.
  strokeWeight(5);

  // Draw the anchor points in black.
  stroke(0);
}
```

▶

■

```
// Click the mouse near the red dot in the bottom-left corner
// and drag to change the curve's shape.

let x1 = 32;
let y1 = 91;
let isChanging = false;

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

  describe(
    'A black U curve drawn upside down on a gray background. The curve passes from one red dot through two black dots and ends at another red dot.'
  );
}

function draw() {
  background(200);

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

  // Start drawing the shape.
  beginShape();

  // Add the first control point and draw a segment to it.
  curveVertex(x1, y1);
}
```

▶

■

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

  background(200);

  // Start drawing the shape.
  beginShape();

  // Add the first control point and draw a segment to it.
  curveVertex(32, 91);
  curveVertex(32, 91);

  // Add the anchor points.
  curveVertex(21, 17);
  curveVertex(68, 19);

  // Add the second control point.
  curveVertex(84, 91);
  curveVertex(84, 91);

  // Stop drawing the shape.
  endShape();

  describe('A ghost shape drawn in white on a gray background.');
```

Syntax

```
curveVertex(x, y)
```

```
curveVertex(x, y, [z])
```

Parameters

- x

Number: x-coordinate of the vertex
- y

Number: y-coordinate of the vertex
- z

Number: z-coordinate of the vertex.

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.	bezierVertex Adds a Bézier curve segment to a custom shape.	curveVertex Adds a spline curve segment to a custom shape.
--	--	---	--