bezier()

Draws a Bézier curve.

Bézier curves can form shapes and curves that slope gently. They're defined by two anchor points and two control points. Bézier curves provide more control than the spline curves created with the curve() function.

The first two parameters, `x1` and `y1`, set the first anchor point. The first anchor point is where the curve starts.

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

The seventh and eighth 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 `bezier()` has twelve arguments because each point has x-, y-, and z-coordinates.

Examples

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

  background(200);

  // Draw the anchor points in black.
  stroke(0);
  strokeWeight(5);
  point(85, 20);
  point(15, 80);

  // Draw the control points in red.
  stroke(255, 0, 0);
  point(10, 10);
  point(90, 90);

  // Draw a black bezier curve.
  noFill();
  stroke(0);
  strokeWeight(1);
  bezier(85, 20, 10, 10, 90, 90, 15, 80);

  // Draw red lines from the anchor points to the control
points.
  stroke(255, 0, 0);
  line(85, 20, 10, 10);
  line(15, 80, 90, 90);

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

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

let x2 = 10;
let y2 = 10;
let isChanging = false;

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

  describe(
    'A gray square with three curves. A black s-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(85, 20);
  point(15, 80);

  // Draw the control points in red.
  stroke(255, 0, 0);
  point(x2, y2);
  point(90, 90);

  // Draw a black bezier curve.
```

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

  background('skyblue');

  // Draw the red balloon.
  fill('red');
  bezier(50, 60, 5, 15, 95, 15, 50, 60);

  // Draw the balloon string.
  line(50, 60, 50, 80);

  describe('A red balloon in a blue sky.');
}
```

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

  describe('A red balloon in a blue sky. The balloon rotates
slowly, revealing that it is flat.');
}

function draw() {
  background('skyblue');

  // Rotate around the y-axis.
  rotateY(frameCount * 0.01);

  // Draw the red balloon.
  fill('red');
  bezier(0, 0, 0, -45, -45, 0, 45, -45, 0, 0, 0, 0);

  // Draw the balloon string.
  line(0, 0, 0, 0, 20, 0);
}
```

Syntax

```
bezier(x1, y1, x2, y2, x3, y3, x4, y4)
```

```
bezier(x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4)
```

Parameters

x1	Number: x-coordinate of the first anchor point.
y1	Number: y-coordinate of the first anchor point.
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 second anchor point.
y4	Number: y-coordinate of the second anchor point.
z1	Number: z-coordinate of the first anchor point.
z2	Number: z-coordinate of the first control point.
z3	Number: z-coordinate of the second control point.
z4	Number: z-coordinate of the second anchor point.

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

Related References

bezier
Draws a Bézier curve.

bezierDetail
Sets the number of segments used to draw Bézier curves in WebGL mode.

bezierPoint
Calculates coordinates along a Bézier curve using interpolation.

bezierTangent
Calculates coordinates along a line that's tangent to a Bézier curve.