

quadraticVertex()

Adds a quadratic Bézier curve segment to a custom shape.

`quadraticVertex()` adds a curved segment to custom shapes. The Bézier curve segments it creates are similar to those made by the `bezierVertex()` function.

`quadraticVertex()` 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 `quadraticVertex()` can be used.

The first two parameters, `cx` and `cy`, set the curve's control point. The control point "pulls" the curve towards its.

The last two parameters, `x3`, and `y3`, 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: `quadraticVertex()` won't work when an argument is passed to `beginShape()`.

Examples



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

  background(200);

  // Style the curve.
  noFill();

  // Draw the curve.
  beginShape();
  vertex(20, 20);
  quadraticVertex(80, 20, 50, 50);
  endShape();

  describe('A black curve drawn on a gray square. The curve starts at the top-left corner and ends at the center.');
}
```



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

  background(200);

  // Draw the curve.
  noFill();
  beginShape();
  vertex(20, 20);
  quadraticVertex(80, 20, 50, 50);
  endShape();

  // Draw red lines from the anchor points to the control point.
  stroke(255, 0, 0);
  line(20, 20, 80, 20);
  line(50, 50, 80, 20);

  // Draw the anchor points in black.
  strokeWeight(5);
  stroke(0);
  point(20, 20);
  point(50, 50);

  // Draw the control point in red.
  stroke(255, 0, 0);
  point(80, 20);

  describe('A black curve that starts at the top-left corner and ends at the center. Its anchor and control points are marked with dots. Red lines connect both anchor points to the control point.');
}

// 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 = 20;
let isChanging = false;

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

  describe('A black curve that starts at the top-left corner and ends at the center. Its anchor and control points are marked with dots. Red lines connect both anchor points to the control point.');
}

function draw() {
  background(200);

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

  // Draw the curve.
  beginShape();
  vertex(20, 20);
  quadraticVertex(x2, y2, 50, 50);
  endShape();

  // Draw red lines from the anchor points to the control point.
```



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

  background(200);

  // Start drawing the shape.
  beginShape();

  // Add the curved segments.
  vertex(20, 20);
  quadraticVertex(80, 20, 50, 50);
  quadraticVertex(20, 80, 80, 80);

  // Add the straight segments.
  vertex(80, 10);
  vertex(20, 10);
  vertex(20, 20);

  // Stop drawing the shape.
  endShape();

  describe('A white puzzle piece drawn on a gray background.');
}
```



```
// Click the and drag the mouse to view the scene from a different angle.

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

  describe('A white puzzle piece on a dark gray background. When the user clicks and drags the scene, the outline of a second puzzle piece is revealed.');
}

function draw() {
  background(50);

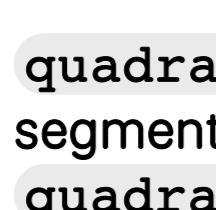
  // Enable orbiting with the mouse.
  orbitControl();

  // Style the first puzzle piece.
  noStroke();
  fill(255);

  // Draw the first puzzle piece.
  beginShape();
  vertex(-30, -30, 0);
  quadraticVertex(30, -30, 0, 0, 0, 0);
  quadraticVertex(-30, 30, 0, 30, 30, 0);
  vertex(30, -40, 0);
  vertex(-30, -40, 0);
  vertex(-30, -30, 0);
```

Syntax

```
quadraticVertex(cx, cy, x3, y3)
```



```
quadraticVertex(cx, cy, cz, x3, y3, z3)
```



Parameters

`cx` Number: x-coordinate of the control point.

`cy` Number: y-coordinate of the control point.

`x3` Number: x-coordinate of the anchor point.

`y3` Number: y-coordinate of the anchor point.

`cz` Number: z-coordinate of the control point.

`z3` Number: z-coordinate of the anchor 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.

`bezierVertex` Adds a Bézier curve segment to a custom shape.

`curveVertex` Adds a spline curve segment to a custom shape.

`curveVertex` Adds a spline curve segment to a custom shape.

`endContour`

`endShape`

`bezierVertex`

`curveVertex`

`normal`

`normal`

`normal`

`normal`

`quadraticVertex`

`quadraticVertex`

`quadraticVertex`

`quadraticVertex`

`vertex`

`vertex`