# curvePoint()

Calculates coordinates along a spline curve using interpolation.

`curvePoint()` calculates coordinates along a spline curve using the anchor and control points. It expects points in the same order as the curve() function. `curvePoint()` works one axis at a time. Passing the anchor and control points' x-coordinates will calculate the x-coordinate of a point on the curve. Passing the anchor and control points' y-coordinates will calculate the y-coordinate of a point on the curve.

The first parameter, `a`, is the coordinate of the first control point.

The second and third parameters, `b` and `c`, are the coordinates of the anchor points.

The fourth parameter, `d`, is the coordinate of the last control point.

The fifth parameter, `t`, is the amount to interpolate along the curve. 0 is the first anchor point, 1 is the second anchor point, and 0.5 is halfway between them.

## Examples

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

  background(200);

  // Set the coordinates for the curve's anchor and control
points.
  let x1 = 5;
  let y1 = 26;
  let x2 = 73;
  let y2 = 24;
  let x3 = 73;
  let y3 = 61;
  let x4 = 15;
  let y4 = 65;

  // Draw the curve.
  noFill();
  curve(x1, y1, x2, y2, x3, y3, x4, y4);

  // Draw circles along the curve's path.
  fill(255);

  // Top.
  let x = curvePoint(x1, x2, x3, x4, 0);
  let y = curvePoint(y1, y2, y3, y4, 0);
  circle(x, y, 5);

  // Center.
  x = curvePoint(x1, x2, x3, x4, 0.5);
  y = curvePoint(y1, y2, y3, y4, 0.5);
  circle(x, y, 5);

  // Bottom.
```

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

  describe('A black curve on a gray square. A white circle
moves back and forth along the curve.');
}

function draw() {
  background(200);

  // Set the coordinates for the curve's anchor and control
points.
  let x1 = 5;
  let y1 = 26;
  let x2 = 73;
  let y2 = 24;
  let x3 = 73;
  let y3 = 61;
  let x4 = 15;
  let y4 = 65;

  // Draw the curve.
  noFill();
  curve(x1, y1, x2, y2, x3, y3, x4, y4);

  // Calculate the circle's coordinates.
  let t = 0.5 * sin(frameCount * 0.01) + 0.5;
  let x = curvePoint(x1, x2, x3, x4, t);
  let y = curvePoint(y1, y2, y3, y4, t);
```

## Syntax

```
curvePoint(a, b, c, d, t)
```

## Parameters

| | |
|---|---|
| a | Number: coordinate of first control point. |
| b | Number: coordinate of first anchor point. |
| c | Number: coordinate of second anchor point. |
| d | Number: coordinate of second control point. |
| t | Number: amount to interpolate between 0 and 1. |

## Returns

Number: coordinate of a point on the curve.

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.