beginShape()

Begins adding vertices to a custom shape.

The `beginShape()` and `endShape()` functions allow for creating custom shapes in 2D or 3D. `beginShape()` begins adding vertices to a custom shape and `endShape()` stops adding them.

The parameter, `kind`, sets the kind of shape to make. By default, any irregular polygon can be drawn. The available modes for kind are:

- `POINTS` to draw a series of points.
- `LINES` to draw a series of unconnected line segments.
- `TRIANGLES` to draw a series of separate triangles.
- `TRIANGLE_FAN` to draw a series of connected triangles sharing the first vertex in a fan-like fashion.
- `TRIANGLE_STRIP` to draw a series of connected triangles in strip fashion.
- `QUADS` to draw a series of separate quadrilaterals (quads).
- `QUAD_STRIP` to draw a strip using adjacent edges to form the next quad.
- `TESS` to create a filling curve by explicit tessellation (WebGL only).

After calling `beginShape()`, shapes can be built by calling `vertex()`, `bezierVertex()`, `quadraticVertex()`, and/or `curveVertex()`. Calling `endShape()` will stop adding vertices to the shape. Each shape will be outlined with the current stroke color and filled with the current fill color.

Transformations such as `translate()`, `rotate()`, and `scale()` don't work between `beginShape()` and `endShape()`. It's also not possible to use other shapes, such as `ellipse()` or `rect()`, between `beginShape()` and `endShape()`.

Examples

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

  background(200);

  // Start drawing the shape.
  beginShape();

  // Add vertices.
  vertex(30, 20);
  vertex(85, 20);
  vertex(85, 75);
  vertex(30, 75);

  // Stop drawing the shape.
  endShape(CLOSE);

  describe('A white square on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Only draw the vertices (points).
  beginShape(POINTS);

  // Add vertices.
  vertex(30, 20);
  vertex(85, 20);
  vertex(85, 75);
  vertex(30, 75);

  // Stop drawing the shape.
  endShape();

  describe('Four black dots that form a square are drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Only draw lines between alternating pairs of vertices.
  beginShape(LINES);

  // Add vertices.
  vertex(30, 20);
  vertex(85, 20);
  vertex(85, 75);
  vertex(30, 75);

  // Stop drawing the shape.
  endShape();

  describe('Two horizontal black lines on a gray background.');
}
```

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

  background(200);

  // Style the shape.
  noFill();

  // Start drawing the shape.
  beginShape();

  // Add vertices.
  vertex(30, 20);
  vertex(85, 20);
  vertex(85, 75);
  vertex(30, 75);

  // Stop drawing the shape.
  endShape();

  describe('Three black lines form a sideways U shape on a gray background.');
}
```

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

  background(200);

  // Style the shape.
  noFill();

  // Start drawing the shape.
  beginShape();

  // Add vertices.
  vertex(30, 20);
  vertex(85, 20);
  vertex(85, 75);
  vertex(30, 75);

  // Stop drawing the shape.
  // Connect the first and last vertices.
  endShape(CLOSE);

  describe('A black outline of a square drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Draw a series of triangles.
  beginShape(TRIANGLES);

  // Left triangle.
  vertex(30, 75);
  vertex(40, 20);
  vertex(50, 75);

  // Right triangle.
  vertex(60, 20);
  vertex(75, 75);
  vertex(80, 20);

  // Stop drawing the shape.
  endShape();

  describe('Two white triangles drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Draw a series of triangles.
  beginShape(TRIANGLE_STRIP);

  // Add vertices.
  vertex(30, 75);
  vertex(40, 20);
  vertex(50, 75);
  vertex(60, 20);
  vertex(70, 75);
  vertex(80, 20);
  vertex(90, 75);

  // Stop drawing the shape.
  endShape();

  describe('Five white triangles that are interleaved drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Draw a series of triangles that share their first vertex.
  beginShape(TRIANGLE_FAN);

  // Add vertices.
  vertex(57.5, 50);
  vertex(57.5, 15);
  vertex(92, 50);
  vertex(57.5, 85);
  vertex(22, 50);
  vertex(57.5, 15);

  // Stop drawing the shape.
  endShape();

  describe('Four white triangles form a square are drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Draw a series of quadrilaterals.
  beginShape(QUADS);

  // Left rectangle.
  vertex(30, 20);
  vertex(30, 75);
  vertex(50, 75);
  vertex(50, 20);

  // Right rectangle.
  vertex(65, 20);
  vertex(65, 75);
  vertex(85, 75);
  vertex(85, 20);

  // Stop drawing the shape.
  endShape();

  describe('Two white rectangles drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Draw a series of quadrilaterals.
  beginShape(QUAD_STRIP);

  // Add vertices.
  vertex(30, 20);
  vertex(30, 75);
  vertex(50, 20);
  vertex(50, 75);
  vertex(65, 75);
  vertex(65, 20);
  vertex(85, 75);

  // Stop drawing the shape.
  endShape();

  describe('Three white rectangles that share edges are drawn on a gray background.');
}
```

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

  background(200);

  // Start drawing the shape.
  // Draw a series of quadrilaterals.
  beginShape(TESS);

  // Add the vertices.
  vertex(-10, -30, 0);
  vertex(30, -30, 0);
  vertex(30, -10, 0);
  vertex(-10, -10, 0);
  vertex(-10, 10, 0);
  vertex(30, 10, 0);
  vertex(30, 30, 0);
  vertex(-30, 30, 0);

  // Stop drawing the shape.
  // Connect the first and last vertices.
  endShape(CLOSE);

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

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

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

  describe('A blocky C shape drawn in red, blue, and green on
a gray background.');
}

function draw() {
  background(200);

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

  // Start drawing the shape.
  // Draw a series of quadrilaterals.
  beginShape(TESS);

  // Add the vertices.
  fill('red');
  stroke('red');
  vertex(-10, -30, 0);
  vertex(30, -30, 0);
  vertex(30, -10, 0);
  fill('green');
  stroke('green');
  vertex(-10, -10, 0);
  vertex(-10, 10, 0);
  vertex(30, 10, 0);
```

Syntax

```
beginShape([kind])
```

Parameters

kind	Constant; either POINTS, LINES, TRIANGLES, TRIANGLE_FAN TRIANGLE_STRIP, QUADS, QUAD_STRIP or TESS.

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.