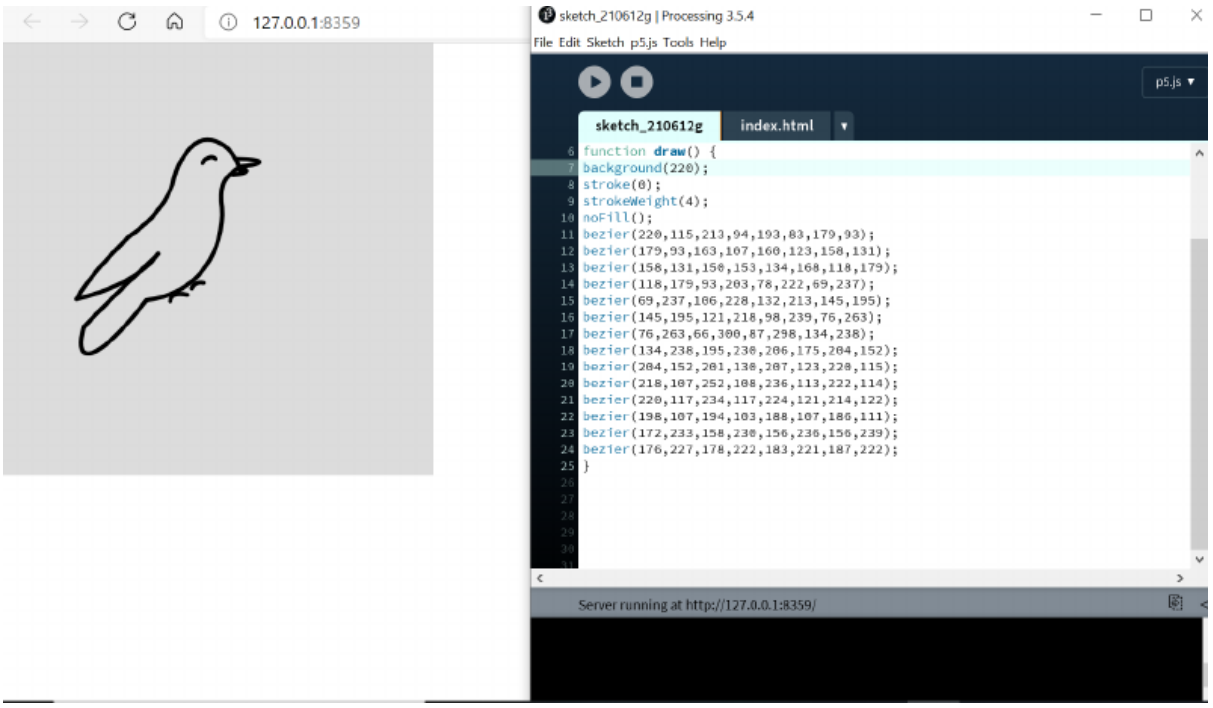
CSE2006 LAB 13

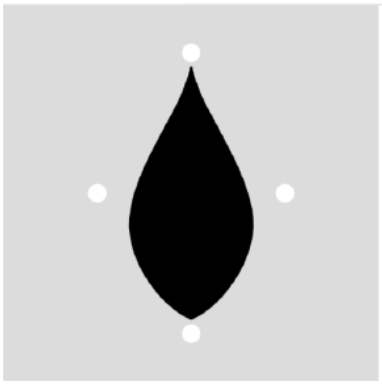
- **Job Fernandez 19BCD7154**

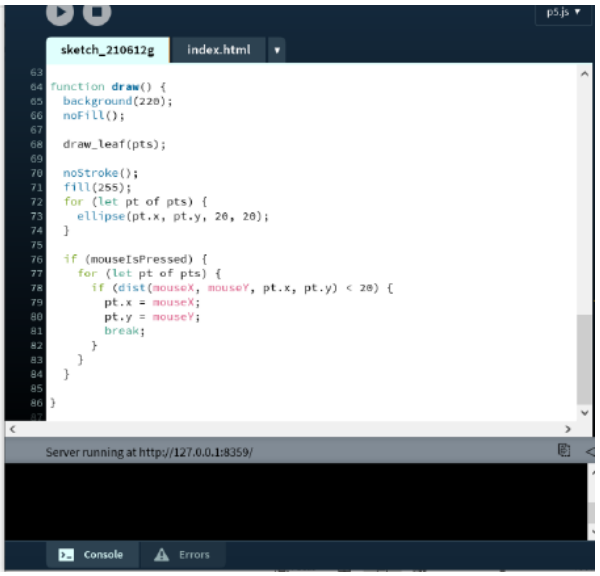
```
function setup() {
createCanvas(400,400);
}
function draw() {
background(220);
stroke(0);
strokeWeight(4);
noFill();
bezier(220,115,213,94,193,83,179,93);
bezier(179,93,163,107,160,123,158,131);
bezier(158,131,150,153,134,168,118,179);
bezier(118,179,93,203,78,222,69,237);
bezier(69,237,106,228,132,213,145,195);
bezier(145,195,121,218,98,239,76,263);
bezier(76,263,66,300,87,298,134,238);
bezier(134,238,195,230,206,175,204,152);
bezier(204,152,201,130,207,123,220,115);
bezier(218,107,252,108,236,113,222,114);
bezier(220,117,234,117,224,121,214,122);
bezier(198,107,194,103,188,107,186,111);
bezier(172,233,158,230,156,236,156,239);
bezier(176,227,178,222,183,221,187,222);
}
```



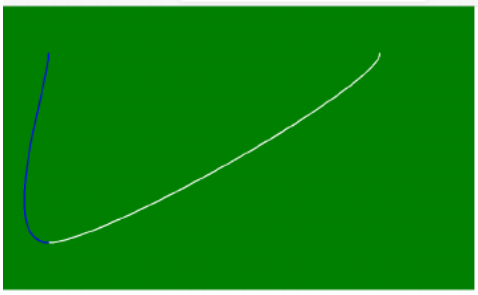
```
let pts = [];
function setup() {
createCanvas(400, 400);
pts = [
createVector(200, 50), //top
createVector(100, 200),//left
createVector(200, 350),//bottom
createVector(300, 200) //right
];
}
function draw_leaf(points) {
curveTightness(-0.2);
noStroke();
fill(0);
let p1 = p5.Vector.lerp(points[1], points[2], 0.5);
let p1_reverse = p5.Vector.lerp(points[3], points[2], 0.5);
let p2 = p5.Vector.lerp(points[0], points[2], 0.95);
let p23 = p5.Vector.lerp(points[2], points[3], 0.75);
let p23_reverse = p5.Vector.lerp(points[2], points[1], 0.75);
let p3 = p5.Vector.lerp(points[1], p23, 0.95)
let p3_reverse = p5.Vector.lerp(points[3], p23_reverse, 0.95);
let p4 = p5.Vector.lerp(points[3], points[0], 0.8);
let p4_reverse = p5.Vector.lerp(points[1], points[0], 0.8);
let p_5 = p5.Vector.lerp(p4, points[1], 0.09);
let p_5_reverse = p5.Vector.lerp(p4_reverse, points[3], 0.09);
let p6 = p5.Vector.lerp(points[0], points[2], 0.05);
beginShape();
curveVertex(p1.x, p1.y);
curveVertex(p2.x, p2.y);
curveVertex(p3.x, p3.y);
curveVertex(p_5.x, p_5.y);
vertex(p6.x, p6.y);
curveVertex(p_5_reverse.x, p_5_reverse.y);
curveVertex(p3_reverse.x, p3_reverse.y);
curveVertex(p2.x, p2.y);
curveVertex(p1_reverse.x, p1_reverse.y);
endShape();
}
function draw() {
background(220);
noFill();
draw_leaf(pts);
noStroke();
fill(255);
for (let pt of pts) {
ellipse(pt.x, pt.y, 20, 20);
}
```

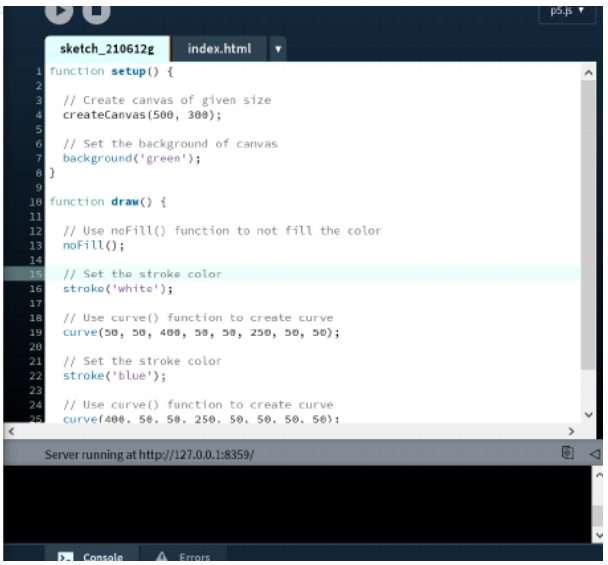
```
if (mouseIsPressed) {
for (let pt of pts) {
if (dist(mouseX, mouseY, pt.x, pt.y) < 20) {
pt.x = mouseX;
pt.y = mouseY;
break;
}
}
}
}
```



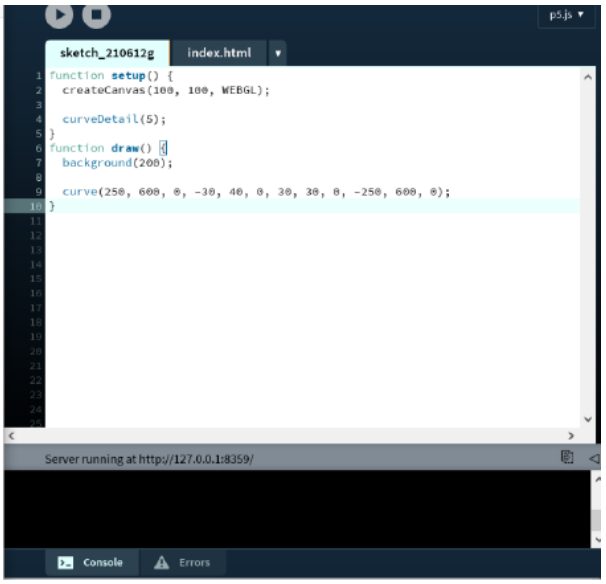



```
function setup() {
// Create canvas of given size
createCanvas(500, 300);
// Set the background of canvas
background('green');
}
function draw() {
// Use noFill() function to not fill the color
noFill();
// Set the stroke color
stroke('white');
// Use curve() function to create curve
curve(50, 50, 400, 50, 50, 250, 50, 50);
// Set the stroke color
stroke('blue');
// Use curve() function to create curve
curve(400, 50, 50, 250, 50, 50, 50, 50);
}
```






```
function setup() {
createCanvas(100, 100, WEBGL);
curveDetail(5);
}
function draw() {
background(200);
curve(250, 600, 0, -30, 40, 0, 30, 30, 0, -250, 600, 0);
}
```




```
function setup() {
createCanvas(600, 600);
noFill();
stroke(255, 0, 0); //red
ellipse(100, 50, 10, 10);
ellipse(400, 300, 10, 10);
```

```
line(100, 50, 400, 300);
strokeWeight(2);
curveBetween(100, 50, 400, 300, 0.3, 0.2, 1);
}
function draw() {
}
function curveBetween(x1, y1, x2, y2, d, h, flip) {
var original = p5.Vector.sub(createVector(x2, y2), createVector(x1, y1));
var inline = original.copy().normalize().mult(original.mag() * d);
var rotated =
inline.copy().rotate(radians(90)+flip*radians(180)).normalize().mult(original.mag() * h);
var p1 = p5.Vector.add(p5.Vector.add(inline, rotated), createVector(x1, y1));
rotated.mult(-1);
var p2 = p5.Vector.add(p5.Vector.add(inline, rotated).mult(-1), createVector(x2, y2));
bezier(x1, y1, p1.x, p1.y, p2.x, p2.y, x2, y2);
}
```