Capturing Mouse Data

```
void draw() {
  frameRate(12);
  println(mouseX + " : " + mouseY);
}
```

Follow

```
void setup() {
  size(100, 100);
  strokeWeight(8);
}

void draw() {
  background(204);
  line(mouseX, mouseY, pmouseX, pmouseY);
}
```

Snake Follow

```
float[] x = new float[20];
float[] y = new float[20];
float segLength = 18;

void setup() {
  size(640, 360);
  strokeWeight(9);
  stroke(255, 100);
}
```

```
void draw() {
  background(0);
  dragSegment(0, mouseX, mouseY);
  for(int i=0; i<x.length-1; i++) {
    dragSegment(i+1, x[i], y[i]);
  }
}

void dragSegment(int i, float xin, float yin) {
  float dx = xin - x[i];
  float dy = yin - y[i];
  float angle = atan2(dy, dx);
  x[i] = xin - cos(angle) * segLength;
  y[i] = yin - sin(angle) * segLength;
  segment(x[i], y[i], angle);
}

void segment(float x, float y, float a) {
  pushMatrix();
  translate(x, y);
  rotate(a);
  line(0, 0, segLength, 0);
  popMatrix();
}
```

Arcs

```
void setup() {
  size(300, 200);
```

```
  background(255);
  smooth();

  rectMode(CENTER); // show bounding boxes
  stroke(128);
  rect(35, 35, 50, 50);
  rect(105, 35, 50, 50);
  rect(175, 35, 50, 50);
  rect(105, 105, 100, 50);

  stroke(0);
  arc(35, 35, 50, 50, 0, PI / 2.0); // lower quarter circle
  arc(105, 35, 50, 50, -PI, 0);  // upper half of circle
  arc(175, 35, 50, 50, -PI / 6, PI / 6); // 60 degrees
  arc(105, 105, 100, 50, PI / 2, 3 * PI / 2); // 180 degrees
}
```

Along a Path

```
int[] coords = {
  40, 40, 80, 60, 100, 100, 60, 120, 50, 150
};

void setup() {
  size(200, 200);
  background(255);
  smooth();

  noFill();
  stroke(0);
```

```
  beginShape();
  curveVertex(40, 40); // the first control point
  curveVertex(40, 40); // is also the start point of curve
  curveVertex(80, 60);
  curveVertex(100, 100);
  curveVertex(60, 120);
  curveVertex(50, 150); // the last point of curve
  curveVertex(50, 150); // is also the last control point
  endShape();

  // Use the array to keep the code shorter;
  // you already know how to draw ellipses!
  fill(255, 0, 0);
  noStroke();
  for (int i = 0; i < coords.length; i += 2) {
    ellipse(coords[i], coords[i + 1], 3, 3);
  }
}
```

Bezier - Write Your Name

```
size(200, 200);
background(255);
beginShape();
vertex(30, 70); // first point
bezierVertex(25, 25, 100, 50, 50, 100);
bezierVertex(20, 130, 75, 140, 120, 120);
endShape();
```

MousePressed

```
void setup() {
  size(100, 100);
}

void draw() {
  if (mousePressed == true) {
    if (mouseButton == LEFT) {
      fill(0); // Black
    } else if (mouseButton == RIGHT) {
      fill(255); // White
    }
  } else {
    fill(126); // Gray
  }
  rect(25, 25, 50, 50);
}
```

https://processing.org/examples/

Zoog

```
void setup() {
 size(480, 270);
 frameRate(30);
}
void draw() {
  background(255);

  ellipseMode(CENTER);
  rectMode(CENTER);

  stroke(0);
  fill(175);
  rect(mouseX, mouseY, 20, 100);

  stroke(0);
  fill(255);
  ellipse(mouseX, mouseY-30, 60, 60);

  fill(mouseX/2, 0, mouseY/2);
  ellipse(mouseX-19, mouseY-30, 16, 32);
  ellipse(mouseX+19, mouseY-30, 16, 32);

  stroke(0);
  line(mouseX-10, mouseY+50, pmouseX-10, pmouseY+60);
  line(mouseX+10, mouseY+50, pmouseX+10, pmouseY+60);
}
void mousePressed() {
  println("Take me to your leader!");
}
```

Jiggle Zoog

```
float x = 240;
float y = 180;
float w = 60;
float h = 60;
float eyeSize = 16;

void setup() {
  size(480, 270);
}
void draw() {
  background(255); // Draw a white background
  float d = dist(x, y, mouseX, mouseY);
  color c = color(d);
  float factor = constrain(mouseX/10, 0, 5);

  jiggleZoog(factor);
  drawZoog(c);
}

void jiggleZoog(float speed) {
  x = x + random(-1, 1)*speed;
  y = y + random(-1, 1)*speed;

x = constrain(x, 0, width);
  y = constrain(y, 0, height);
}

void drawZoog(color eyeColor) {
  ellipseMode(CENTER);
  rectMode(CENTER);
```

```
  for (float i = y - h/3; i < y + h/2; i += 10) {
    stroke(0);
    line(x - w/4, i, x + w/4, i);
  }
  stroke(0);
  fill(175);
  rect(x, y, w/6, h);
  stroke(0);
  fill(255);
  ellipse(x, y - h, w, h);
  fill(eyeColor);
  ellipse(x - w/3, y - h, eyeSize, eyeSize*2);
  ellipse(x + w/3, y - h, eyeSize, eyeSize*2);
  stroke(0);
  line(x - w/12, y + h/2, x - w/4, y + h/2 + 10);
  line(x + w/12, y + h/2, x + w/4, y + h/2 + 10);
}
```