

```

w2_05_AdvancedGraphics | Processing 2.1.2
AdvancedGraphics.pde
void setup() {
  size(500, 400);
  smooth(4);
}

void draw() {
  background(255);

  //Head
  stroke(0);
  beginShape();
  noFill();
  float headCenterX = width/2;
  float headCenterY = 100;
  float headSize = 50;
  for (int i = 0; i < 6; i++) {
    float radian = TWO_PI/6;
    float x = headCenterX + headSize*cos(radian/2 + radian*i);
    float y = headCenterY + headSize*sin(radian/2 + radian*i);
    vertex(x, y);
  }

  //the foot center
  strokeWeight(1);
  line(headCenterX, headCenterY, x, y);
  strokeWeight(6);
  endShape(CLOSE);

  //Body
  noFill();
  strokeWeight(1);
  beginShape();
  for (int i = 0; i < 360; i++) {
    //this draws one unit of curve, then repeat 180times.
    float curveHeight = 5;
    float curveWidth = 5;
    float firstPointX = width/2;
    float firstPointY = 150 + curveHeight*i;
    float firstControlPointX = firstPointX + curveWidth*(i/180);
    float firstControlPointY = firstPointY;
    float secondPointX = width/2;
    float secondPointY = 150 + curveHeight*(i/180);
    float secondControlPointX = secondPointX + curveWidth*(i/180);
    float secondControlPointY = secondPointY;
    vertex(firstPointX, firstPointY); // start point of each curve unit
    bezierVertex( firstControlPointX, firstControlPointY,
      secondControlPointX, secondControlPointY ); //leftside
  }
  endShape();

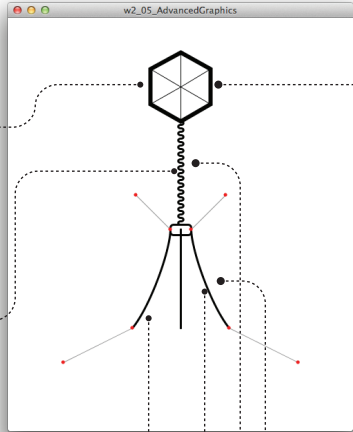
  //Bottom
  float bottomX = width/2;
  float bottomY = 300;
  float bottomWidth = 30;
  float bottomHeight = 15;
  float bottomBound = 5;
  rect(bottomX - bottomWidth/2, bottomY, bottomWidth, bottomHeight, bottomBound);

  //leg-left (L)
  float ll_firstPointX = width/2 + 15;
  float ll_firstPointY = 300;
  float ll_firstControlX = ll_firstPointX - 50;
  float ll_firstControlY = ll_firstPointY - 50;
  float ll_secondPointX = width/2 + 70;
  float ll_secondPointY = 450;
  float ll_secondControlX = ll_secondPointX - 100;
  float ll_secondControlY = ll_secondPointY - 50;
  curve(ll_firstControlX, ll_firstControlY, ll_firstPointX, ll_firstPointY, ll_secondPointX, ll_secondPointY, ll_secondControlX, ll_secondControlY);

  //leg-right (R)
  float lr_firstPointX = width/2 - 15;
  float lr_firstPointY = 300;
  float lr_firstControlX = lr_firstPointX + 50;
  float lr_firstControlY = lr_firstPointY - 50;
  float lr_secondPointX = width/2 - 70;
  float lr_secondPointY = 450;
  float lr_secondControlX = lr_secondPointX + 100;
  float lr_secondControlY = lr_secondPointY - 50;
  curve(lr_firstControlX, lr_firstControlY, lr_firstPointX, lr_firstPointY, lr_secondPointX, lr_secondPointY, lr_secondControlX, lr_secondControlY);

  //leg-middle
  float ml_x = width/2, 300;
  //guide line
  strokeWeight(1);
  stroke(150);
  line(ll_firstPointX, ll_firstPointY, ll_firstControlX, ll_firstControlY);
  line(lr_firstPointX, lr_firstPointY, lr_firstControlX, lr_firstControlY);
  line(ll_secondPointX, ll_secondPointY, ll_secondControlX, ll_secondControlY);
  line(lr_secondPointX, lr_secondPointY, lr_secondControlX, lr_secondControlY);
  noStroke();
  fill(255, 0, 0);
  ellipse(ll_firstPointX, ll_firstPointY, 5, 5);
  ellipse(lr_firstPointX, lr_firstPointY, 5, 5);
  ellipse(ll_secondPointX, ll_secondPointY, 5, 5);
  ellipse(lr_secondPointX, lr_secondPointY, 5, 5);
  ellipse(ll_secondControlX, ll_secondControlY, 5, 5);
  ellipse(lr_secondControlX, lr_secondControlY, 5, 5);
}
    
```

[Abstract drawing of Bacteriophage]
 :Ref <http://www.britannica.com/EB-checked/topic/48324/bacteriophage>



Head : Vertex, and beginShape() and endShape()

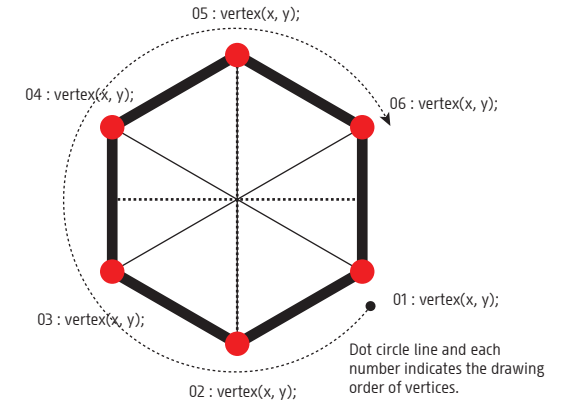
Using the beginShape() and endShape() functions allows the creation of more complex forms. beginShape() begins recording vertices for a shape and endShape() stops recording. The value of the argument tells beginShape() which types of shapes to create from the provided vertices. With no mode specified, the shape can be any irregular polygon. The parameters available for beginShape() are POINTS, LINES, TRIANGLES, TRIANGLE_FAN, TRIANGLE_STRIP, QUADS, and QUAD_STRIP.

Arguments such as POINTS, LINES, TRIANGLES, TRIANGLE_FAN, TRIANGLE_STRIP, QUADS, QUAD_STRIP can be added in here

```

beginShape( );
noFill();
float headCenterX = width/2;
float headCenterY = 100;
float headSize = 50;
for (int i = 0; i < 6; i++) {
  float radian = TWO_PI/6;
  float x = headCenterX + headSize*cos(radian/2 + radian*i);
  float y = headCenterY + headSize*sin(radian/2 + radian*i);
  vertex(x, y);
}
strokeWeight(1);
line(headCenterX, headCenterY, x, y);
strokeWeight(6);
endShape(CLOSE);
    
```

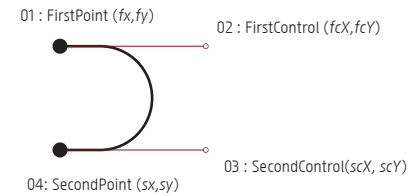
The endShape() function is the companion to beginShape() and may only be called after beginShape(). Use the constant CLOSE as the value for the MODE parameter to connect the beginning and end vertices.



QUAD	QUAD_STRIP	TRIANGLES	TRIANGLE_STRIP	TRIANGLE_FAN

Drawing order: The arrow line refers to the drawing order of vertices. This drawing order is important to draw the shape correctly.
More reference : [beginShape \(http://processing.org/reference/beginShape_.html\)](http://processing.org/reference/beginShape_.html)

Body : bezierVertex (Bezier curve within beginShape() and endShape() functions.



Code Template

```

beginShape();
bezierVertex( fx, fy, fcX, fcY, scX, scY, sx, sy);
endShape();
    
```

Leg : Curve and Bezier

curve(fcX,fcY, fx,fy, sx,sy, scX, scY)



bezier(fx,fy, fcX,fcY, sx,sy, scX, scY)

