

Section A  
Unit #7  
Vertex



## Section A

### Unit #7 Vertex

Sketch A7.1	drawing a line
Sketch A7.2	drawing a square
Sketch A7.3	using CLOSE
Sketch A7.4	irregular shapes



## Unit #7 Vertex

It is one thing to draw regular shapes using the functions circle, square, rectangle and triangle etc but what if you want to draw something that isn't a regular shape. Then you can use the vertex() function to draw any shape. This is great for drawing random patterns or shapes and manipulating them.

We start with beginShape() and add vertex co-ordinates for x and y and then add endShape() at the end.



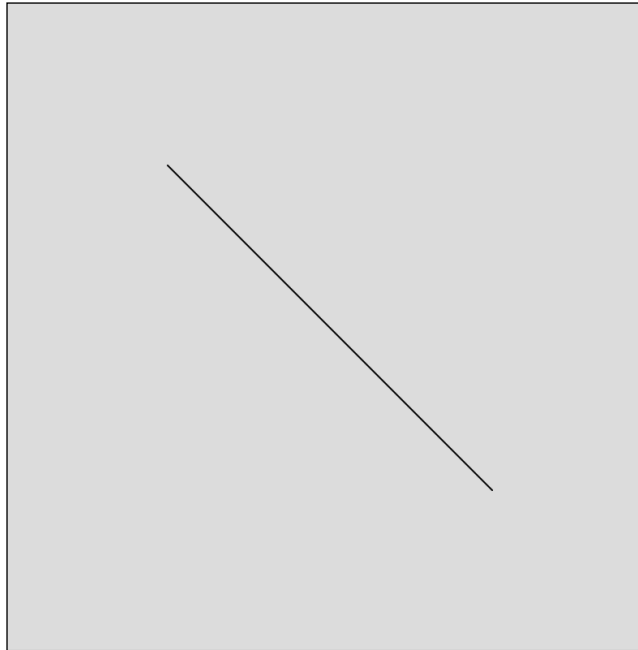
## Sketch A7.1 drawing a line - vertex

We are going to start very simply by drawing a line. We specify the end coordinates of the line and it draws a line between them, there is no line function. Although this seems a little protracted it has great value when you use the coordinates in creative ways drawing complex shapes and patterns.

```
function setup()
{
  createCanvas(400, 400)
}

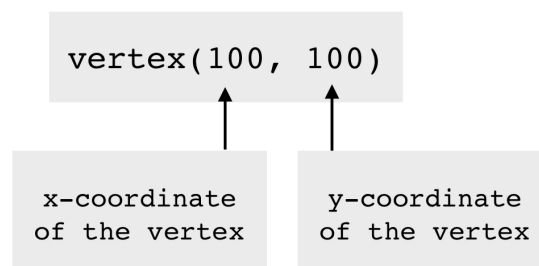
function draw()
{
  background(220)
  beginShape()
  vertex(100, 100)
  vertex(300, 300)
  endShape()
}
```

## Simple line using the vertex



## Notes

Using the `vertex()` function is a useful tool to draw irregular polygons, in other words shapes other than rectangles and triangles. A `Vertex` simply draws a dot (pixel) at the co-ordinate given. The `beginShape()` and `endShape()` join the dots up. In this example to get you started two dots so it gives you one line



## Challenge

Try it without the `beginShape()/endShape()`



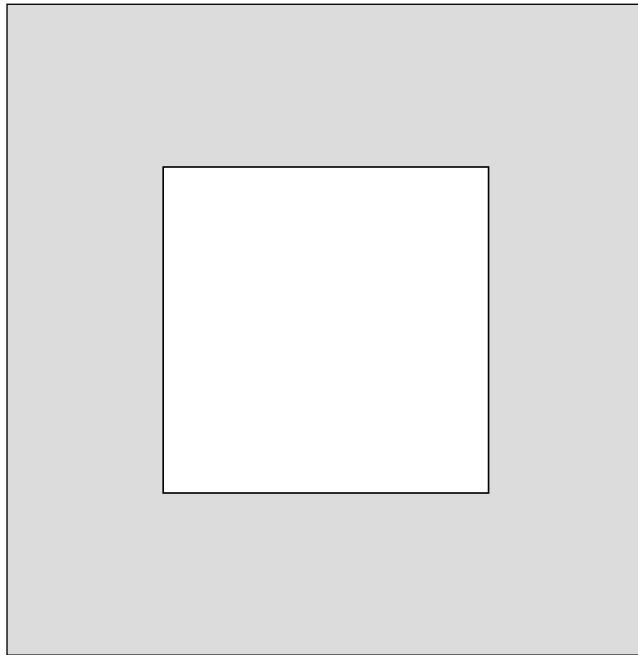
## Sketch A7.2 drawing a square - vertex

Drawing a simple square using the vertices co-ordinates, the order is critical. It draws a line from one vertex to the next one on the list. So don't get them mixed up. Notice that we have five vertices whereas a square should have only four, do you know why?

```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  beginShape()
  vertex(100, 100)
  vertex(300, 100)
  vertex(300, 300)
  vertex(100, 300)
  vertex(100, 100)
  endShape()
}
```

## Drawing the humble square



## Notes

Now to draw a square using more dots and therefore more lines. For this to work you need to close the shape by drawing the last dot where the first one was drawn. In the next example there is a shortcut to this.

## Challenge

1. Reorder the vertices and see what happens
2. Try to draw a couple of rectangles



## Sketch A7.3 using CLOSE - vertex

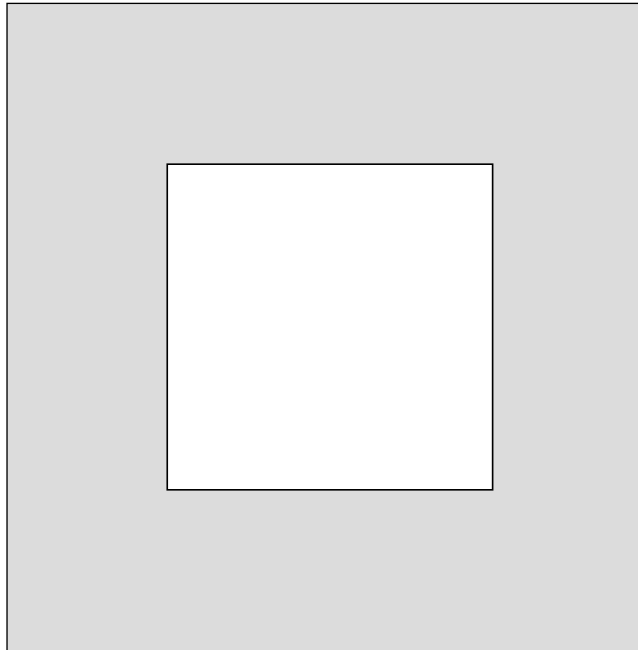
This joins up the last and first vertices automatically. We can remove the last vertex so that we only have four and add the CLOSE command. Although we have four vertices for the square it doesn't draw the final line until you add the word CLOSE to the endShape()

```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  beginShape()
  vertex(100, 100)
  vertex(300, 100)
  vertex(300, 300)
  vertex(100, 300)
  endShape(CLOSE)
}
```

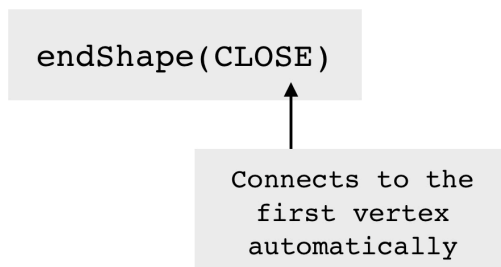


Using CLOSE to draw the final line



## Notes

Using `endShape(CLOSE)` means you don't have to draw the last dot to join it all up. Notice it is in capital letters



## Challenge

Try the above without the CLOSE argument and see the difference



## Sketch A7.4 irregular shapes - vertex

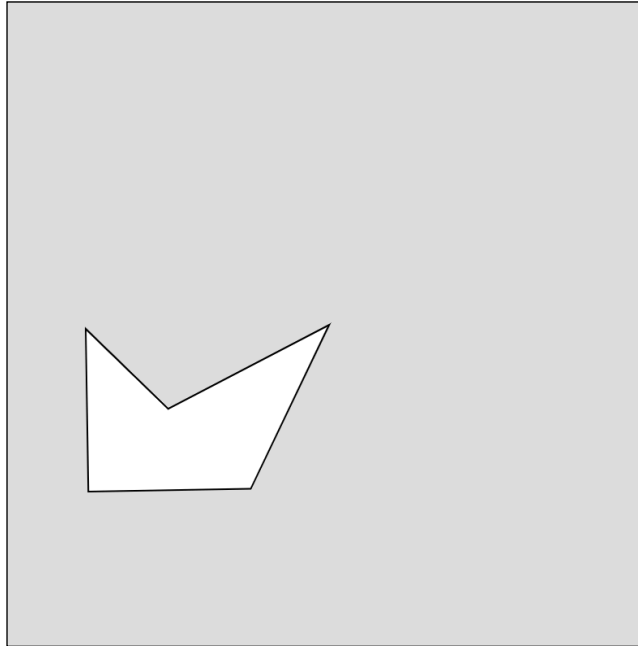
This is where drawing vertices comes in handy

```
let angle = 0

function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  translate(width/2, height/2)
  rotate(angle)
  beginShape()
  vertex(0, 0)
  vertex(100, 50)
  vertex(100, 150)
  vertex(0, 150)
  vertex(50, 100)
  endShape(CLOSE)
  angle += 0.05
}
```

This shape rotates about the centre of the canvas



## Notes

Now you can draw any shape you want. You may find it helpful to have some square paper to work out the co-ordinates if the shape is getting complicated. It is easier than working it out in your head.

## Challenge

Draw a star



## A bit of playing

### Just playing

```
let angle = 0
let amplitudeX = 150
let amplitudeY = 150
let x = 0
let y = 0

function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  translate(width/2, height/2)
  x = amplitudeX*sin(angle)
  y = amplitudeY*cos(angle)
  beginShape()
  vertex(x, y)
  vertex(x, 150)
  vertex(150, y)
  vertex(-x, -y)
  endShape(CLOSE)
  angle += 0.02
}
```

Adding some sine/cosine movement

