

## Inspired by Ann H Murray's Archimedean Spiral

After doing the [reading](#) and connecting to both the visual aspects of Murray's work and the quote "The final abstract expression of every art is number." (Kandinsky's italics)

I wanted to look into using math and processing to create work, beginning by searching the archimedean spiral and looking into the mathematics behind it.  $R = a + b0^{1/c}$  is the most common representation for a spiral, having multiple possibilities where  $c=1$  and other possibilities including  $c=-1$  (hyperbolic's spiral),  $c=2$  (fermat's) and  $c=-2$  (lituus). Also learning that most spirals which appear in nature are logarithmic spirals and not archimedean.

Historically it seems that Archimedes invented or developed the spiral in Syracuse, Sicily after his studies in Egypt<sup>1</sup>.

I looked further into how to create a spiral and found [this](#):

```
void setup() {
  size(400, 400);
}

void draw() {
  translate(width/2, height/2);
  for (float t = 0; t < 2*TWO_PI; t+=0.1 ) {
    float x = t * cos(t);
    float y = t * sin(t);
    point(x, y);
  }
}
```

I liked that it was a forum which was asking the same question I had, and that this was a very simple piece of code which I could understand and edit. Another inspiration was [Liriel Saarinen](#) who I found by accident and used her code to help me animate. After playing with her code a bit I realised that `rotate( -float(frameCount)/ 20);` could be used to animate almost anything as long as you edit the values accordingly.

I ended up editing the code to create animation of a spiral which then transform into other shapes using animation.

### Example 1

```
void setup() {
  size(800, 800);
  frameRate(60);
}
```

1. <https://mathshistory.st-andrews.ac.uk/Biographies/Archimedes/>

```

void draw() {
  translate(width/2, height/2);
  rotate( -float(frameCount)/ .012);
  for (float t = 0; t < 460*TWO_PI; t+=0.001 ) {
    float x = t * cos(t+1);
    float y = t * sin(t+1);
    point(x, y);
  }
}

```

## Example 2

### Radiant lines

```

void setup() {
  size(800, 800);
  frameRate(60);
}

void draw() {
  translate(width/2, height/2);
  rotate( +float(frameCount)/ 1
);
  for (float t = 1; t < 720*TWO_PI; t+=PI
)
  {
    float x = t * cos(t);
    float y = t * sin(t);
    point(x, y);
  }
}

```

### The eye

```

size(800, 800);
frameRate(60);
}

void draw() {
  translate(width/2, height/2);
  rotate( +float(frameCount)/ 200

```

```

);
for (float t = 1; t < 720*TWO_PI; t+=PI+cos(-2)/24
)
{
    float x = t * cos(t-1
);
    float y = t * sin(t
);
    point(x, y);
}
}

```

Spiral eye 2

```

void setup() {
    size(800, 800);
    frameRate(60);
}

void draw() {
    translate(width/2, height/2);
    rotate( +float(frameCount)/ 300
);
    for (float t = 1; t < 720*TWO_PI; t+=PI+cos(-1)/23
    )
    {
        float x = t * cos(t-7
);
        float y = t * sin(t
);
        point(x, y);
    }
}

```

Flower

```

void setup() {
    size(800, 800);
    frameRate(90);
}

void draw() {
    translate(width/2, height/2);

```

```

    rotate( +float(frameCount)/ .44444
    );
    for (float t = 3; t < 720*TWO_PI; t+=PI+cos(-9
    )/61
    )
    {
        float x = t * cos(t
        );
        float y = t * sin(t/222.22
        );
        point(x, y);
    }
}

```

## Flower2

```

void setup() {
    size(800, 800);
    frameRate(160);
}

void draw() {
    translate(width/2, height/2);
    rotate( +float(frameCount)/ 1/3
    );
    for (float t = 1/8; t < 720*TWO_PI; t+=PI+cos(-5
    )/143
    )
    {
        float x = t * cos(t+.899898998989
        );
        float y = t * sin(t/1
        );
        point(x, y);
    }
}

```

Made an [open processing account](#)