

```

import processing.pdf.*;

ArrayList<Integer> lerpColors;
color toAlt;

int cellW = 50;
int cellH = 50;

int totalColors;
int colorIdx = 0;

int x = 0;

color from = color(random(255), 0, random(100));
color to = color(255,21,238);
color altTo = color(106, 244, 205);

void setup() {
  size(1500, 1500);
  beginRecord(PDF, "LerpTriangles.pdf");
  noLoop();
  background(0);
  totalColors = (width/cellW) - 2;
  lerpColors = getLerpedColorList(color(to), color(from), totalColors);
  toAlt = color(altTo);
  println(from, to, altTo);
}

void draw() {

  for (int y = 0; y < height; y+=cellH) {

    ArrayList<Integer> lerpColColors = getLerpedColorList(lerpColors.get(colorIdx), toAlt,
totalColors);

    for (int i = 0; i < lerpColColors.size(); i++) {
      fill(lerpColColors.get(i));
      //stroke(255);
      triangle(x, y, x + cellW, y, x + (cellW/2), y + cellH);
      stroke(0);
      line(x, y + cellH/2, x + cellW/2, y);
      line(x + cellW/2, y, x + cellW, y + cellH/2);
      x+=cellW;
    }

    x = 0;
    colorIdx++;
  }
}

```

```
}  
    endRecord();  
}
```

```
ArrayList<Integer> getLerpedColorList(color myFrom, color myTo, int myTotalColors) {
```

```
    //create the lerp color set that will serve as the row header from  
    ArrayList<Integer> colors = new ArrayList<Integer>();  
    colors.add(myFrom);  
    for (int i = 0; i < myTotalColors; i++) {  
        float fi = float(i);  
        colors.add(lerpColor(myFrom, myTo, fi/myTotalColors));  
    }  
  
    colors.add(myTo);  
    println(colors.size());  
    return colors;  
}
```