

#1

```
let font

function preload() {
  font = loadFont('assets/FiraCode-Regular.ttf')
}

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

function draw() {
  background(61, 132, 222)
  lights()

  scale(2)

  push()
  rotateY(millis()*0.005)
  rotateX(millis()*0.0001)

  fill(255)
  plane(60, 80)

  push()
  fill(0)
  translate(-18, 0, 0)
  plane(5, 80)
  pop()

  push()
  fill(0)
  textSize(10)
  textFont(font)
  translate(-35, 35, 0)
  rotate(-PI/2)
  text('1 Kind', 10, 50)

  translate(0, -15, 0)
  textSize(7)
  text('Ticket', 10, 50)
  pop()

  pop()
}
```



#2

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

function draw() {
  background(61, 132, 222)
  lights()

  camera(sin(millis()*0.001)*150, -30, 300)

  fill(61, 132, 222)
  push()
  translate(0, 80, 0)
  rotateX(PI/2)
  plane(300, 150)
  pop()

  fill(150)
  for (let i = 0; i < 3; i++) {
    push()
    translate(100*i-100, 0, 0)

    push()
    translate(-22, 24, 0)
    box(40, 100, 100)
    pop()

    for (let a = 0; a < TWO_PI; a += TWO_PI/3) {
      push()
      rotateX(PI/2)
      rotateZ(PI/2)

      rotateX(cos(a)*0.6)
      rotateZ(sin(a)*0.6)

      cylinder(2, 90)
      pop()
    }

    pop()
  }
}
```



#3

```
precision mediump float;

uniform vec2 resolution;
uniform float time;

float circle(vec2 p, float r) {
    float s = 0.2;
    float w = 0.05;
    return smoothstep(r-w-s, r-w, length(p)) -
        smoothstep(r-s, r, length(p));
}

void main() {
    vec2 uv = gl_FragCoord.xy / resolution.xy;

    float d = 0.;
    for (float i = 0.; i < 25.; i++) {
        float t = mod(time/10.+i*5., 85.);
        float r = sin(mod(t, 0.39))*3.;
        d += circle(uv-vec2(sin(i)*0.5+0.5, sin(i*2.)*0.5+0.5), r);
    }
    d /= 2.;

    vec2 p = uv-d;
    float x = mod(float(int(p.x*20.)), 4.);
    vec3 c = x > 0. ? vec3(0.3, 0.6, 0.9) : vec3(0.8, 0.99, 0.99);
    gl_FragColor = vec4(c, 1.);
}
```



#4

```
const radius = 30

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

  noStroke()
  fill(245, 212, 2)
}

function draw() {
  background(61, 132, 222)
  lights()

  camera(0, 10, 75)

  rotateX(millis()*0.001)
  torus(radius, 3)

  for (let a = 0; a < TWO_PI; a += TWO_PI/32) {
    push()
    translate(cos(a)*radius, sin(a)*radius, 0)
    rotateZ(a)
    cylinder(6, 3)
    pop()
  }
}
```



#5

```
precision mediump float;

uniform vec2 resolution;
uniform float time;

#define TWO_PI 6.28

void main() {
    vec2 uv = gl_FragCoord.xy / resolution.xy;

    vec2 p = uv-vec2(0.5);
    float a = atan(p.x, p.y)-time*0.4;
    float s = float(mod(a/TWO_PI, 1./3.)>1./6.);
    float d = max(-(length(p)-0.5), length(p)-0.1);

    float v = float(d<0.3)-s/2.;

    vec3 c = vec3(v)*vec3(0.2, 0.2, 0.8);
    if (v < 0.1) {
        c = vec3(0.24, 0.52, 0.87);
    }

    gl_FragColor = vec4(c, 1.);
}
```

