// initializing a random seed to create a pattern

```
1  var seed = Math.random() * 15283;
2  var t;  // time variable
3  var num, vNum;
4  var radius, mySize, margin;
5  var sizes = []; // empty array for sizes
6
7  let colors = []; // color combos as arrays of HEX values
8  let colors0 = "B7AB98-1ABC9C-B7AB98-1ABC9C".split("-").map((a) => "#" + a);
9  let colors7 = "B7AB98-B7AB98-1ABC9C-1ABC9C-B7AB98".split("-").map((a) => "#" + a);
10 let colors8 = "B7AB98-1ABC9C-B7AB98-1ABC9C-B7AB98-1ABC9C-B7AB98-1ABC9C-B7AB98-1ABC9C-
   B7AB98".split("-").map((a) => "#" + a);
11 let colors11 = "B7AB98-1ABC9C-B7AB98-1ABC9C-B7AB98".split("-").map((a) => "#" + a);
12
13 var color_setup1, color_setup2;
14 let color_bg;
15 let v_planet = []; // array for vector planet positions
16
17 function setup() {
18     randomSeed(seed);
19     // pixelDensity(5);
20     mySize = min(windowWidth, windowHeight);
21     margin = mySize / 100;
22     createCanvas(windowWidth, windowHeight, WEBGL);
23     color_setup1 = colors7;
24     color_setup2 = random([ colors11]);
25     color_bg = "#0D0D0D";
26     background(color_bg);
27     // num = 50;
28     num = int(random(20, 10));
29     radius = mySize * 0.85;
30     for (let a = 0; a < TAU; a += TAU / num) {
31         sizes.push(random(0.1, 0.5))
32     }
33     t = 0;
34 }
35
36 function draw() {
37     randomSeed(seed); // another random seed
38     background(color_bg); // clear the background
39
40     for (let i = 0; i < num; i++) {
41         let a = (TAU / num) * i;
42         let x = radius * sin(a + t) / random(5, 3) / 1.0;
43         let y = radius * cos(a + t) / random(3, 5) / 1.0;
44         v_planet[i] = createVector(x, y);
45     }
46     push(); // 3D PART
47
48     for (let q = 0; q < 1 / 5; q += 2 * random(0.01, 0.02)) {
49         for (let j = 0; j < 1; j++) {
50             let n = noise(q*t, j*t,frameCount*0.01); // noise for rotation
51             rotateX(random(TAU)+sin(-t) / 5 + q );                & position
52             rotateY(random(TAU)+cos(t) / 5 + q );
53             rotateZ(random(TAU)+sin(-t) / 5 + q );
54             noStroke();
55             fill(random(color_setup2));
56                 // torus shapes
57             for (let i = 0; i < num; i += 8) {
58                 let d = random(radius / 2, radius / 4) / 1;
59                 push();
60                 rotateX( random(TAU)+sin(t));
```

setup color combos

generate random sizes for elements & set time

positions for planets

```
                rotateY(random(TAU)+cos(-t)+n/100 );
                rotateZ( random(TAU)+2 * sin(2*t) );

                let x_plus = 1.25 * random(-d, d) / 1;
                let y_plus =  1.25 * random(-d, d) / 1;
                let z_plus =1.25 * random(-d, d) / 1;

                torus(z_plus,random(1),100,100);
                pop();
            }     // spheres
            for (let i = 0; i < num; i += 4) {
                let d = (1.5 + sin(t)) * random(radius / 2, radius / 4);
                let x_plus = 0.5 * random(-d, d) / 1;
                let y_plus = 0.5 * random(-d, d) / 1;
                let z_plus = 0.5 * random(-d, d) / 1;
                stroke(random(color_setup2));
                strokeWeight(random(0.5));
                noFill();
                push();
                translate(v_planet[i].x + x_plus, v_planet[i].y + y_plus, z_plus);
                rotateX(random(TAU)+t);
                rotateY(random(-TAU)+t);
                rotateZ(random(PI)+t);
                sphere(random(2));
                pop();
            }
        }
    }
    pop();  // updating the variables to create an animation

    t += random(2, 1) * random(0.001, 0.005) / 1;
}
    // saves the canvas as an image when "s" is pressed

function keyTyped() {
    if (key === "s" || key === "S") {
        saveCanvas("0712_Emotional lines_12_2022", "png");
    }
}
```