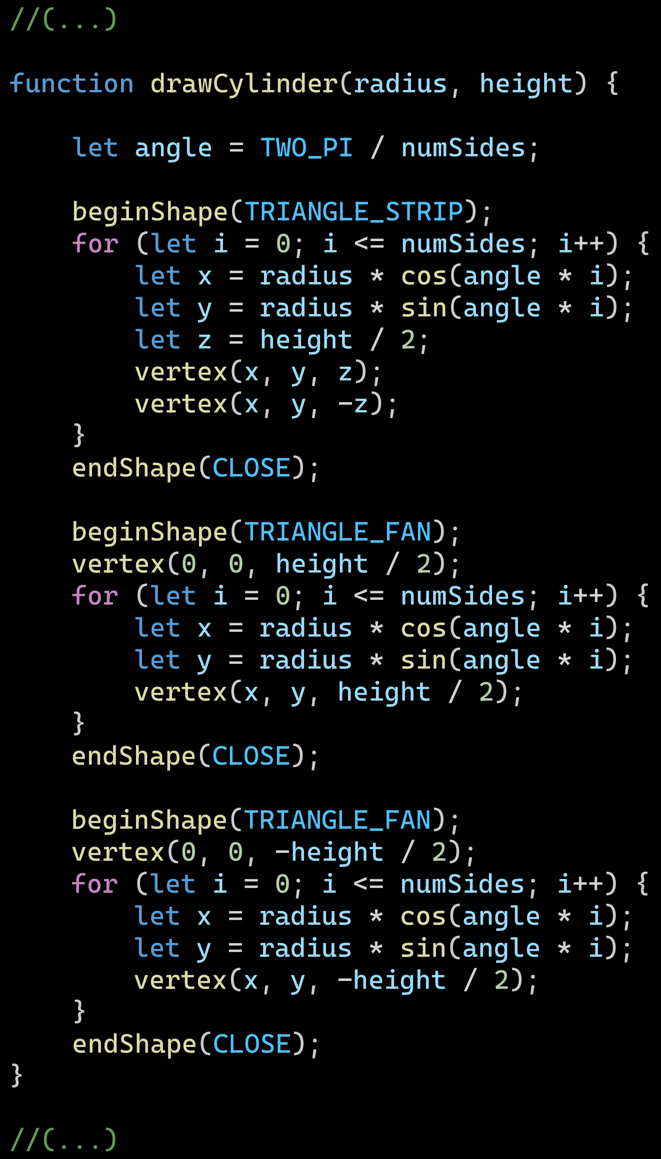
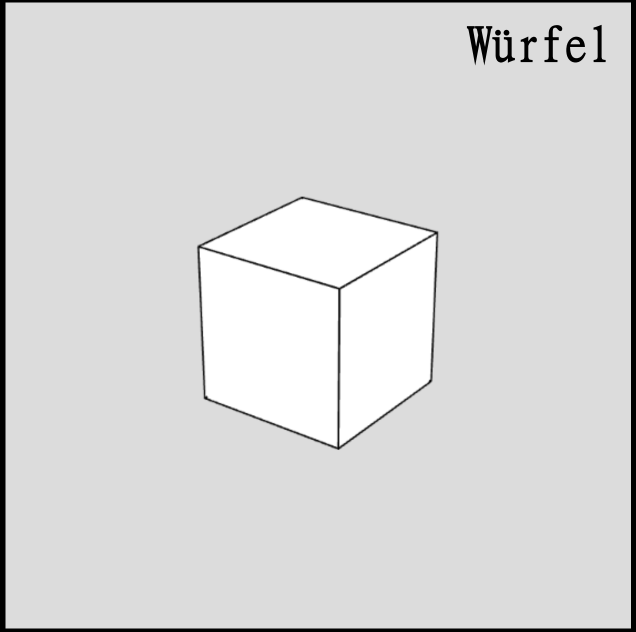
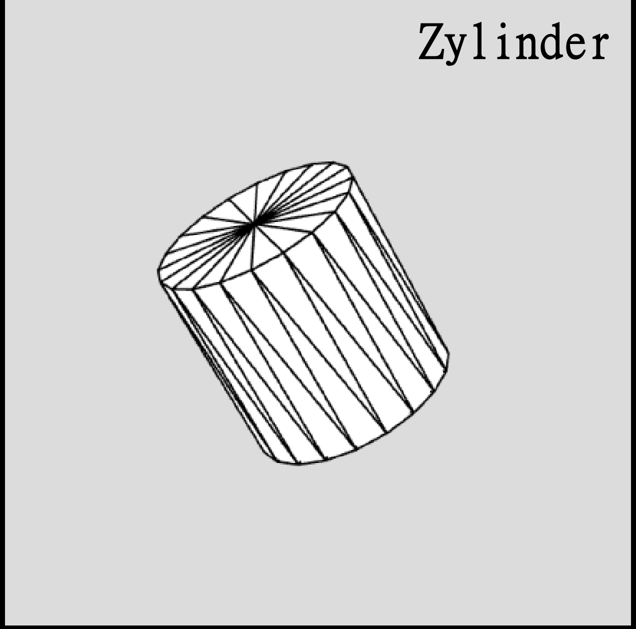
Beautiful Mathematics | Woche 02 | Lukas Bucher



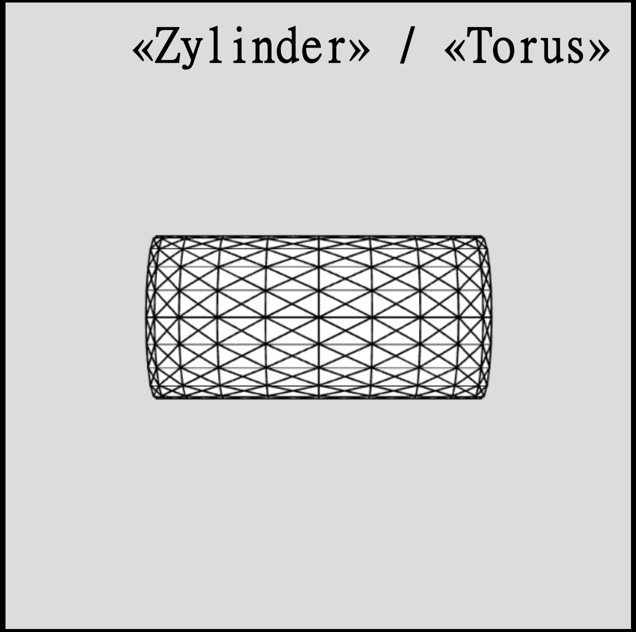
```
//(...)

function drawCylinder(radius, height) {

    let angle = TWO_PI / numSides;

    beginShape(TRIANGLE_STRIP);
    for (let i = 0; i <= numSides; i++) {
        let x = radius * cos(angle * i);
        let y = radius * sin(angle * i);
        let z = height / 2;
        vertex(x, y, z);
        vertex(x, y, -z);
    }
    endShape(CLOSE);

    beginShape(TRIANGLE_FAN);
    vertex(0, 0, height / 2);
    for (let i = 0; i <= numSides; i++) {
        let x = radius * cos(angle * i);
        let y = radius * sin(angle * i);
        vertex(x, y, height / 2);
    }
    endShape(CLOSE);

    beginShape(TRIANGLE_FAN);
    vertex(0, 0, -height / 2);
    for (let i = 0; i <= numSides; i++) {
        let x = radius * cos(angle * i);
        let y = radius * sin(angle * i);
        vertex(x, y, -height / 2);
    }
    endShape(CLOSE);
}

//(...)
```



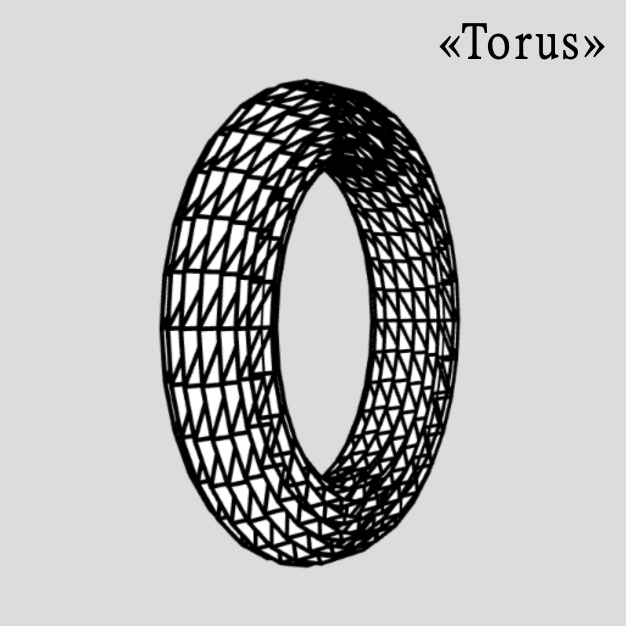
Würfel



Zylinder

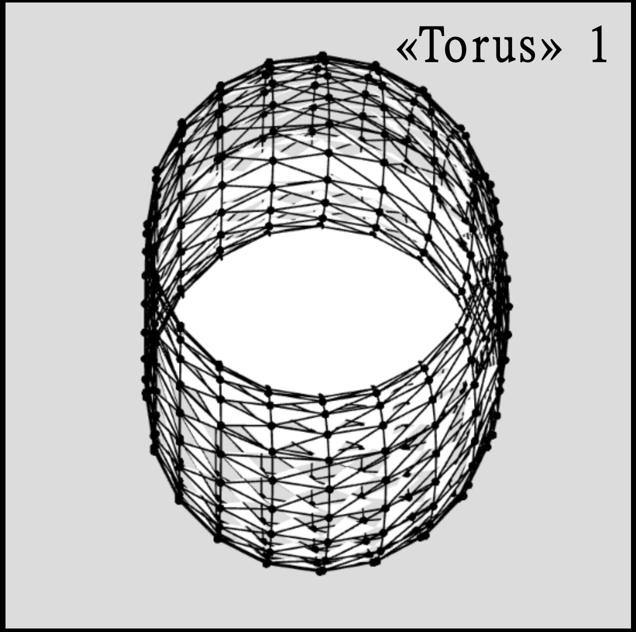


«Zylinder» / «Torus»

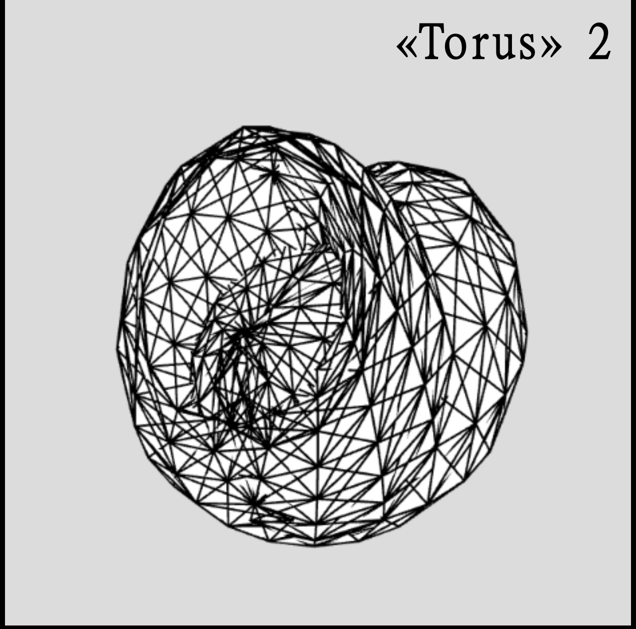


«Torus»

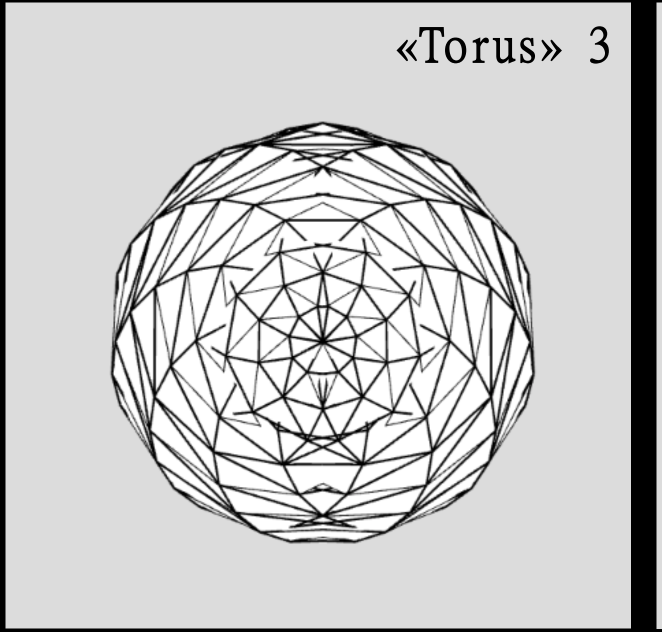
```
//(...)

//Torus 4
function drawTorus(radius, tubeRadius) {
    beginShape(TRIANGLES);
    fill(255);
    stroke(0);
    strokeWeight(1);

    let numSides = 20;
    let vertices = [];
    let angle = TWO_PI / numSides;
    for (let i = 0; i < numSides; i++) {
        let x = radius * cos(angle * i);
        let y = radius * sin(angle * i);
        for (let j = 0; j < numSides; j++) {
            let y = tubeRadius * sin(angle * j);
            let z = tubeRadius * cos(angle * j);
            vertices.push([x, y, z]);
        }
    }

    for (let i = 0; i < numSides; i++) {
        let x = radius * cos(angle * i);
        let y = radius * sin(angle * i);
        for (let j = 0; j < numSides; j++) {
            let z = tubeRadius * cos(angle * j);
            let a = (i + 1) % numSides;
            let b = (j + 1) % numSides;
            let c = (i + 1) % numSides;
            let d = (j + 1) % numSides;
            vertex(vertices[i][0], vertices[i][1], vertices[i][2]);
            vertex(vertices[a][0], vertices[a][1], vertices[a][2]);
            vertex(vertices[b][0], vertices[b][1], vertices[b][2]);
            vertex(vertices[i][0], vertices[i][1], vertices[i][2]);
            vertex(vertices[c][0], vertices[c][1], vertices[c][2]);
            vertex(vertices[d][0], vertices[d][1], vertices[d][2]);
        }
    }

    endShape(CLOSE);
}

//(...)
```



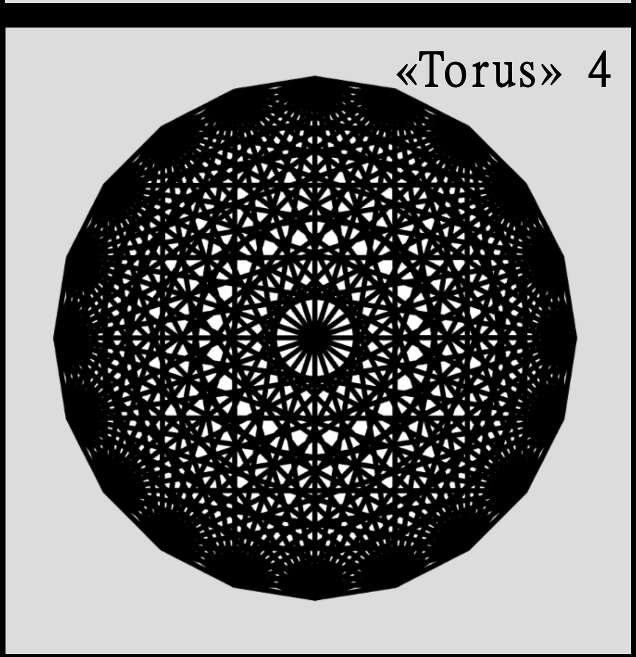
«Torus» 1



«Torus» 2



«Torus» 3

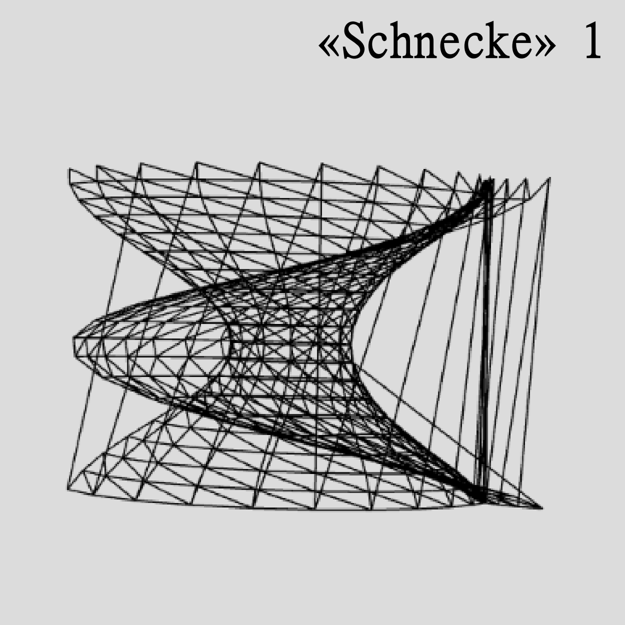


«Torus» 4

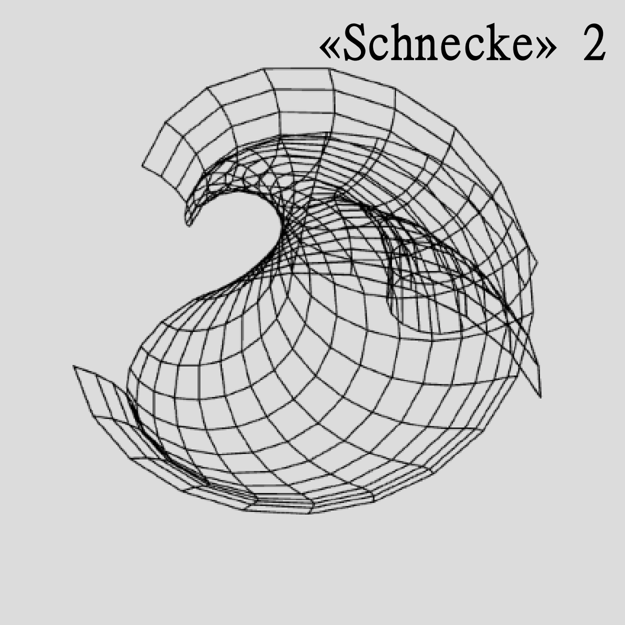
Der Weg zum Torus

Beautiful Mathematics | Woche 02 | Lukas Bucher

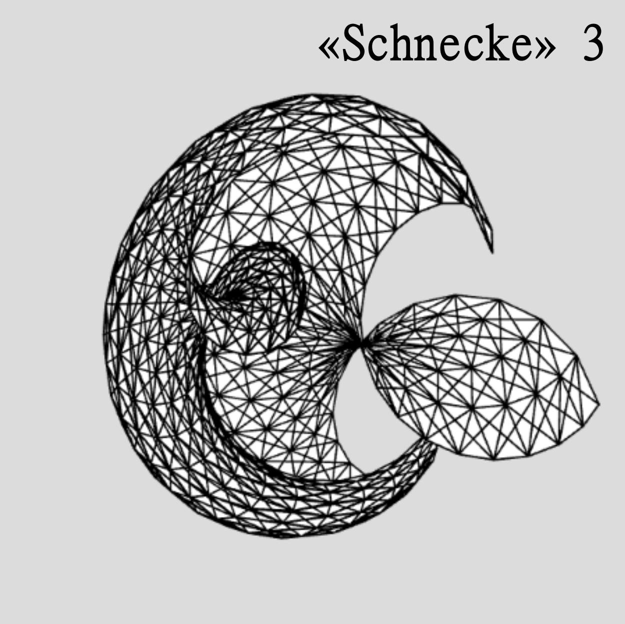
```
//(...)

function drawSnail(radius, tubeRadius) {
    let angle = TWO_PI / numSides;
    for (let i = 0; i < numSides; i++) {
        let vertex = [];
        let xpos = radius * cos(angle * i);
        let ypos = radius * sin(angle * i);
        for (let j = 0; j < numSides; j++) {
            let x = tubeRadius * cos(angle * j);
            let y = tubeRadius * sin(angle * j);
            let z = j * 10;

            vertex.push([x + xpos, y + ypos, z]);
        }
        vertex.push(vertex[vertex.length - 1]);
        vertices.push(vertex);

        tubeRadius -= 10; //Schnecke 4
        radius -= 5; //Schnecke 3
    }
    vertices.push(vertices[vertices.length - 1]);

    for (let i = 0; i < numSides; i++) {
        beginShape(TRIANGLE_STRIP);
        for (let j = 0; j < numSides; j++) {
            let v1 = vertices[i + 1][j];
            let v2 = vertices[i + 1][j + 1];
            let v3 = vertices[i][j + 1];
            let v4 = vertices[i][j];

            vertex(v1[0], v1[1], v1[2]);
            vertex(v2[0], v2[1], v2[2]);
            vertex(v3[0], v3[1], v3[2]);

            vertex(v1[0], v1[1], v1[2]);
            vertex(v3[0], v3[1], v3[2]);
            vertex(v4[0], v4[1], v4[2]);
        }
        endShape();
    }
}

//(...)
```



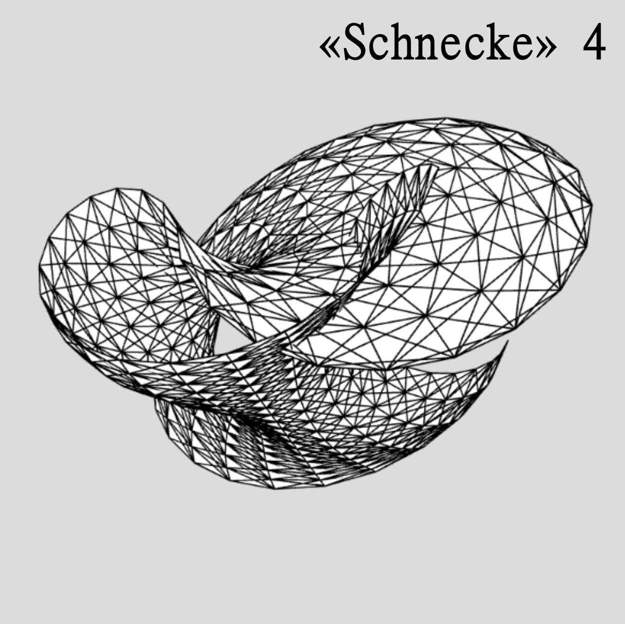
«Schnecke» 1



«Schnecke» 2



«Schnecke» 3



«Schnecke» 4

Der Weg zur «Schnecke»