**4.4 圓柱**

var Conversions = Core.Conversions;

var Debug = Core.Debug;

var Path2D = Core.Path2D;

var Point2D = Core.Point2D;

var Point3D = Core.Point3D;

var Matrix2D = Core.Matrix2D;

var Matrix3D = Core.Matrix3D;

var Mesh3D = Core.Mesh3D;

var Plugin = Core.Plugin;

var Tess = Core.Tess;

var Sketch2D = Core.Sketch2D;

var Solid = Core.Solid;

var Vector2D = Core.Vector2D;

var Vector3D = Core.Vector3D;

// Template Code:

params = [

{ "id": "dia", "displayName": "Diameter", "type": "length", "rangeMin": 0, "rangeMax": 200, "default": 20.0 },

{ "id": "height", "displayName": "Height", "type": "length", "rangeMin": 0, "rangeMax": 200, "default": 20.0 },

{ "id": "res", "displayName": "Resolution", "type": "int", "rangeMin": 1, "rangeMax": 50, "default": 10 },

{ "id": "x\_pos", "displayName": "x position","type": "float","rangeMin": -200,"rangeMax": 200,"default": 0 },

//預設100是因為將圖形放到桌面中央時，調整x、y position為零時，

//可將圖中心線形移往桌面左下角。

{ "id": "y\_pos", "displayName": "y position", "type": "float","rangeMin": -200,"rangeMax": 200, "default": 0},

{ "id": "z\_pos", "displayName": "z position", "type": "float","rangeMin": -200, "rangeMax": 200,"default": 0 }

];

function process(params) {

var dia = params["dia"];

var height = params["height"];

var res = params["res"]; //取解析度

var r2 = dia/2.0;

var ndivs = Math.round(res \* r2);

//設定解析度\*外圈半徑當作分割值，之後就是三角形數量

var x\_pos = params["x\_pos"];

var y\_pos = params["y\_pos"];

var z\_pos = params["z\_pos"];

var inL = [x\_pos,y\_pos,z\_pos+0]; //設定0度內圈底部點為"目前內圈底部點"

var inH = [x\_pos,y\_pos,z\_pos+height]; //0度內圈頂部點為"目前頂圈底部點"

var outL = [x\_pos+r2,y\_pos,z\_pos+0]; //0度外圈底部點"目前外圈底部點"

var outH = [x\_pos+r2,y\_pos,z\_pos+height]; //0度外圈頂部點"目前外圈頂部點"

var mesh = new Mesh3D();

for (var i = 0; i < ndivs; i++) {

var a = (i+1)/ndivs \* Math.PI\*2.0; //由０開始環繞360度取ndivs個角度

var s = Math.sin(a);

var c = Math.cos(a);

//準備順時鐘繞取點

var nextinL = [x\_pos, y\_pos, z\_pos+0]; //-a度內圈底部點

var nextinH = [x\_pos, y\_pos, z\_pos+height]; //-a度內圈頂部點

var nextoutL = [x\_pos+r2\*c, y\_pos-r2\*s,z\_pos+ 0]; //-a度外圈底部點

var nextoutH = [x\_pos+r2\*c, y\_pos-r2\*s, z\_pos+height]; //-a度外圈頂部點

mesh.triangle(inH, outH, nextoutH); //頂部三角形

mesh.triangle(outH, outL, nextoutL); //外部三角形1

mesh.triangle(outH, nextoutL, nextoutH); //外部三角形2

mesh.triangle(inL, outL, nextoutL); //底部三角形1

//更新所有目前四點

inL = nextinL;

inH = nextinH;

outL = nextoutL;

outH = nextoutH;

}

var solid = Solid.make(mesh);

return solid;

}