

# OpenSCAD 3D Editor

OZ6FRS Frederikssund 2022

# En appetit vækker til 'gør det selv' i 3D med OpenSCAD.

Nu har Peter i det foregående foredrag fortalt om Klubbens 3D printer, om hvordan den sættes op, og klargøres til brug.

Peter fortalte også hvordan et emne vi har designet eller f.eks. Hentet fra Websiden '<https://www.thingiverse.com>'. Bliver klargjort og tilpasses i slicer programmet Ultimaker Cura.

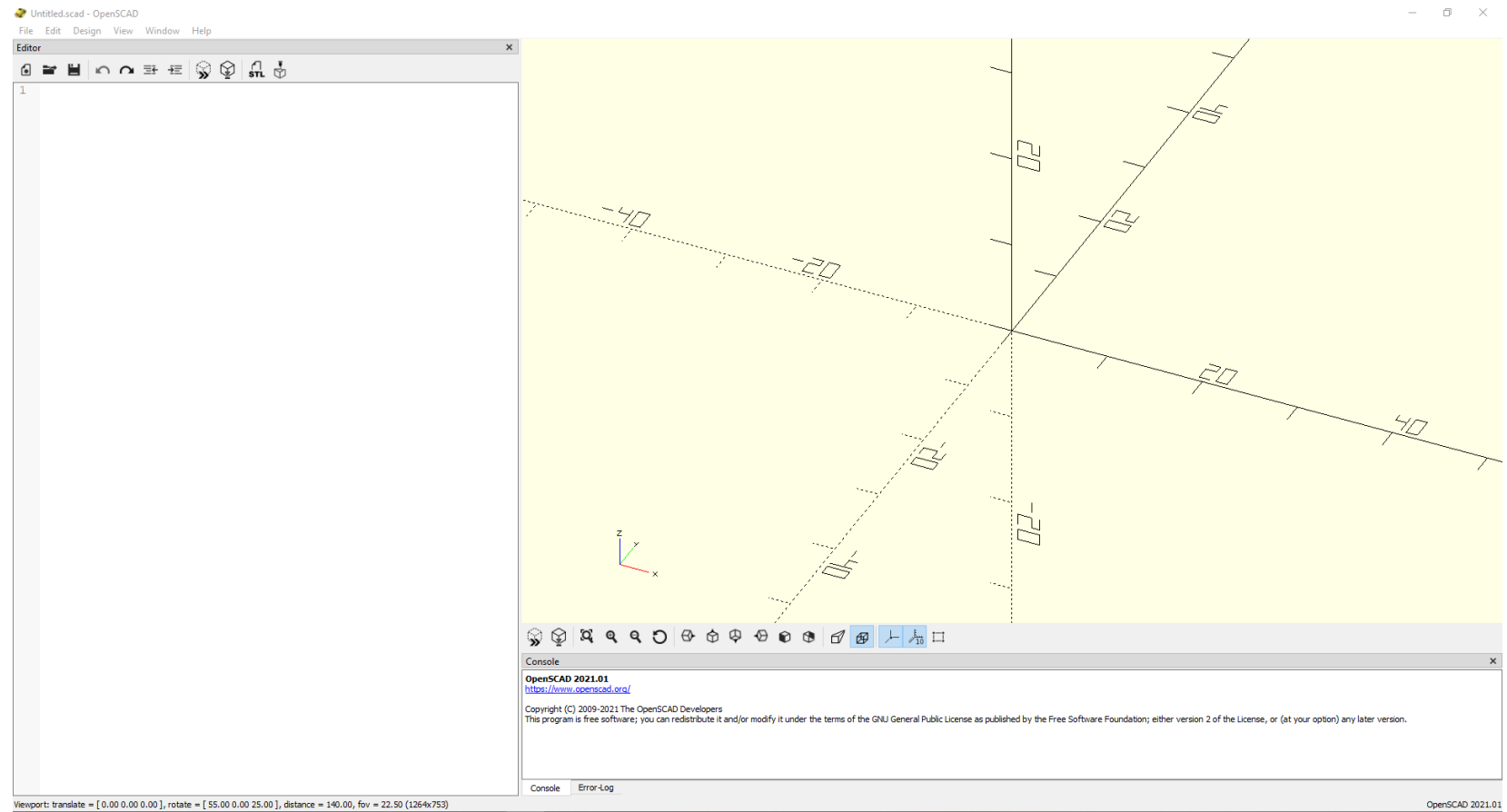
Jeg vil nu prøve at fortælle, om 3D CAD programmet 'OpenSCAD', som vi kan bruge til at designe vores egne 3D ideer/emner.

# OpenSCAD 1

Editoren består grundlæggende af 3 vinduer, til venstre har vi selve Editor vinduet, det er her vi skriver vores kode.

Vinduet til højre viser vores 3D emne. Emnet kan vendes og drejes samt gøres større og mindre.

Nederst til højre er et tekst vindue der viser en rapport over forløbet af Rendering og eventuelle fejl rapporteres her.



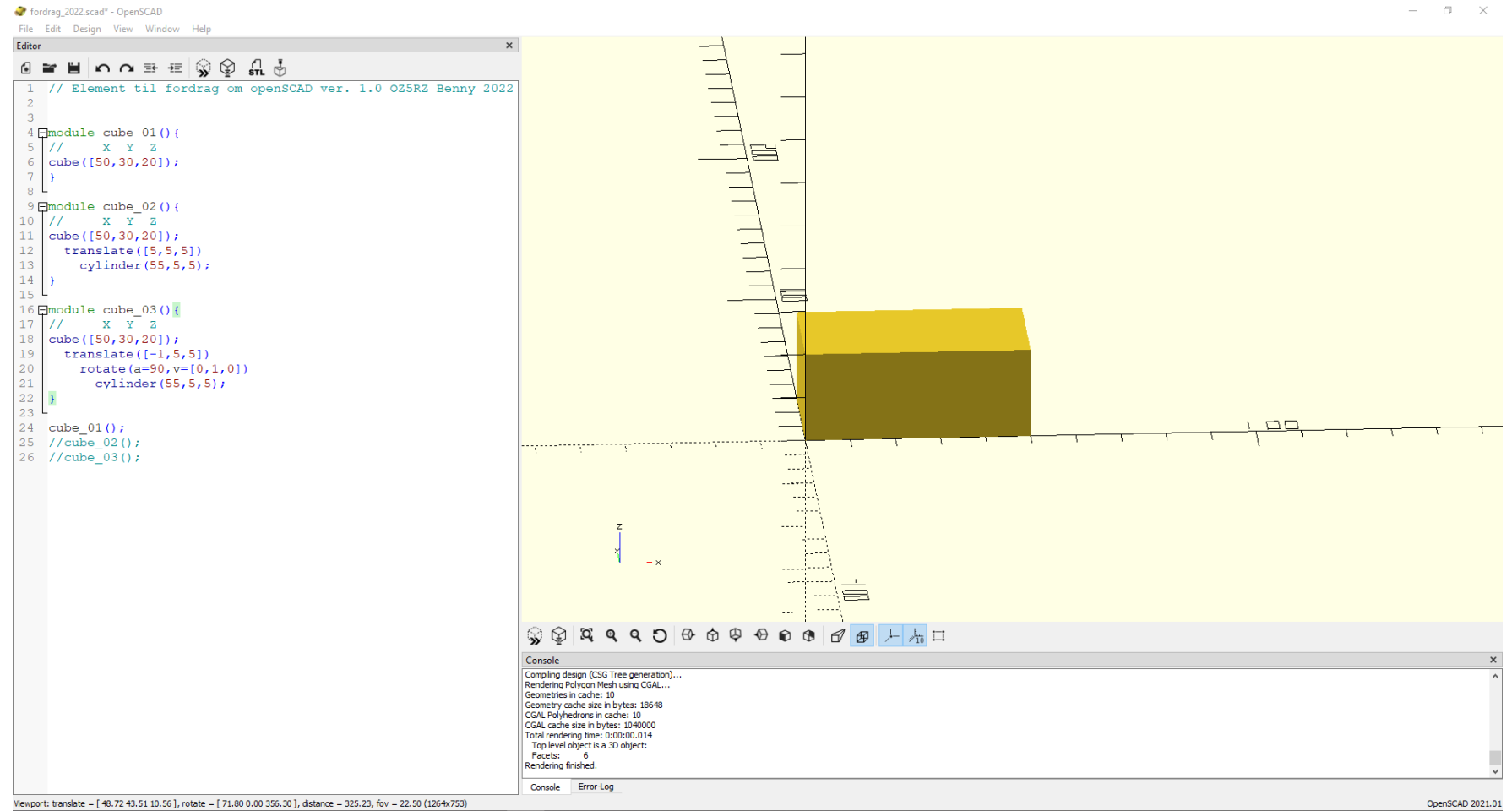
# OpenSCAD 2

Program koden starter gerne med en tekst der fortæller hvad dette er og hvilken version, og hvem der har gjort dette ;o).

Jeg benytter ofte 'moduler', moduler gør det nemt at tænde eller slukke for dele af konstruktionen...

Dette kan være en fordel for nemmere at kunne se andre detaljer eller for at spare tid ved tidskrævende renderingen, og når vi arbejder med en særlig detalje.

For at se mine ændringer kan jeg enten gemme eller renderede emnet (*rendering tager tid men viser detaljer*), emnet placerer sig automatisk på koordinaten 0,0,0 .



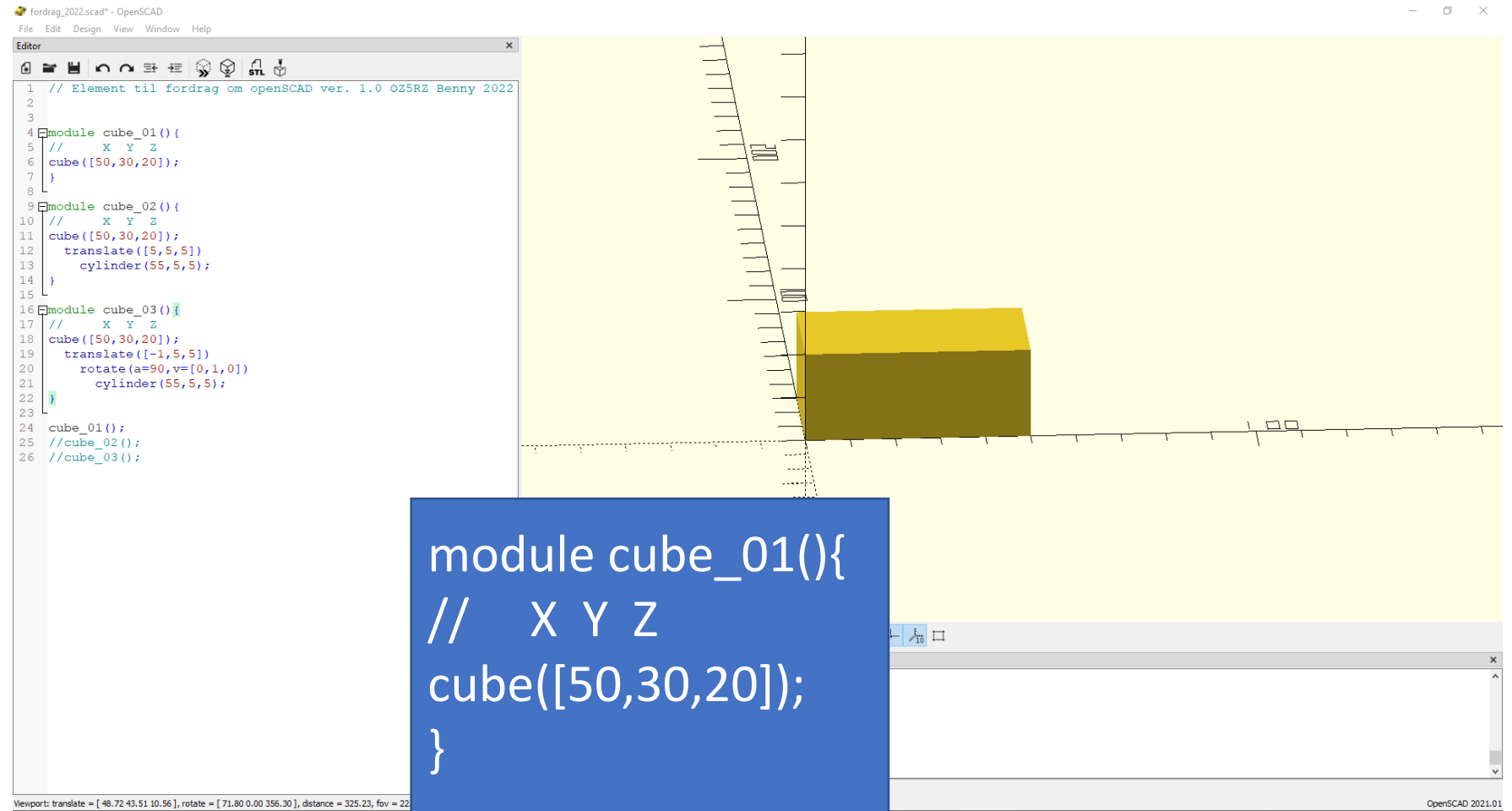
# OpenSCAD 2

Program koden starter gerne med en tekst der fortæller hvad dette er og hvilken version, og hvem der har gjort dette ;o).

Jeg benytter ofte 'moduler', moduler gør det nemt at tænde eller slukke for dele af konstruktionen...

Dette kan være en fordel for nemmere at kunne se andre detaljer eller for at spare tid ved tidskrævende renderingen, og når vi arbejder med en særlig detalje.

For at se mine ændringer kan jeg enten gemme eller renderede emnet (*rendering tager tid men viser detaljer*), emnet placerer sig automatisk på koordinaten 0,0,0 .



```
module cube_01(){
//   X Y Z
cube([50,30,20]);
}

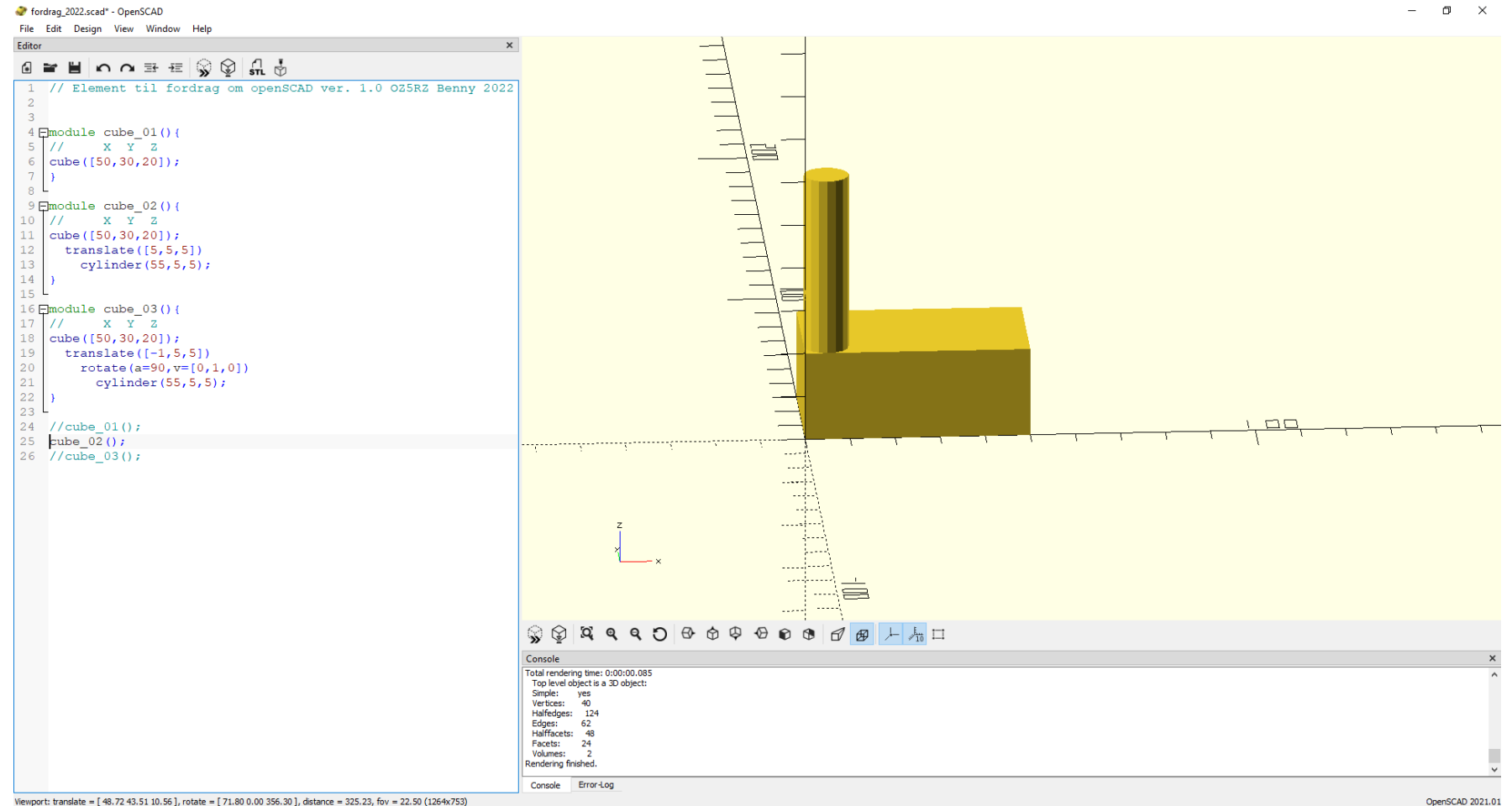
cube_01();
```

# OpenSCAD 3

Nu har jeg deaktiveret module `cube_01()` og aktiveret module `cube_02()` som med kommandoen `'cylinder([55,5,5])` tegner en rund stang.

Denne stang placeres med kommandoen `'translate([5,5,5])` i yderkanten af vores cube, disse koordinat værdien kan være + eller - .

Så nu har vi lavet en lille fabrik med skorsten ;o)

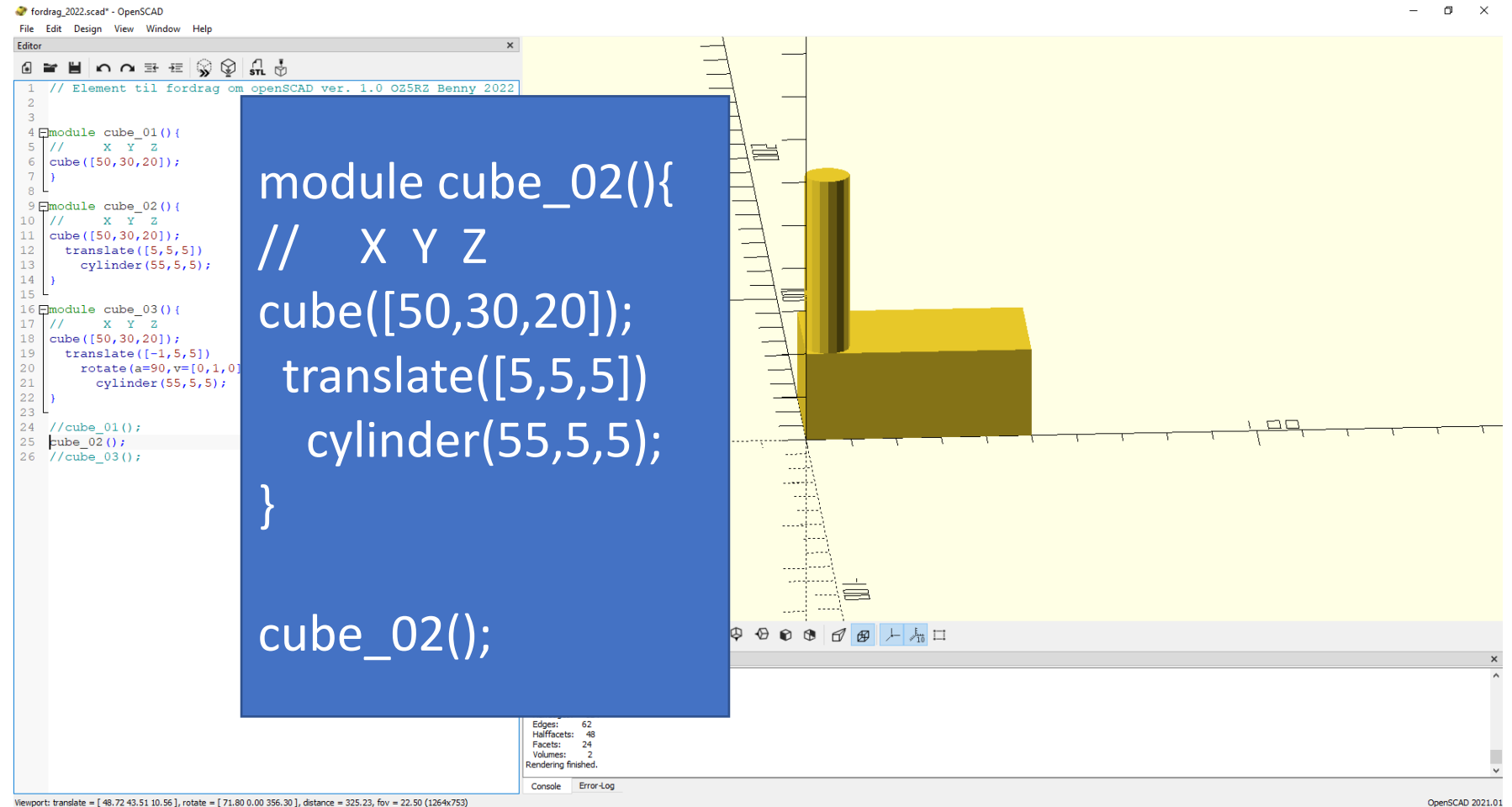


# OpenSCAD 3

Nu har jeg deaktiveret module `cube_01()` og aktiveret module `cube_02()` som med kommandoen `'cylinder([55,5,5])` tegner en rund stang.

Denne stang placeres med kommandoen `'translate([5,5,5])` i yderkanten af vores cube, disse koordinat værdien kan være + eller - .

Så nu har vi lavet en lille fabrik med skorsten ;o)



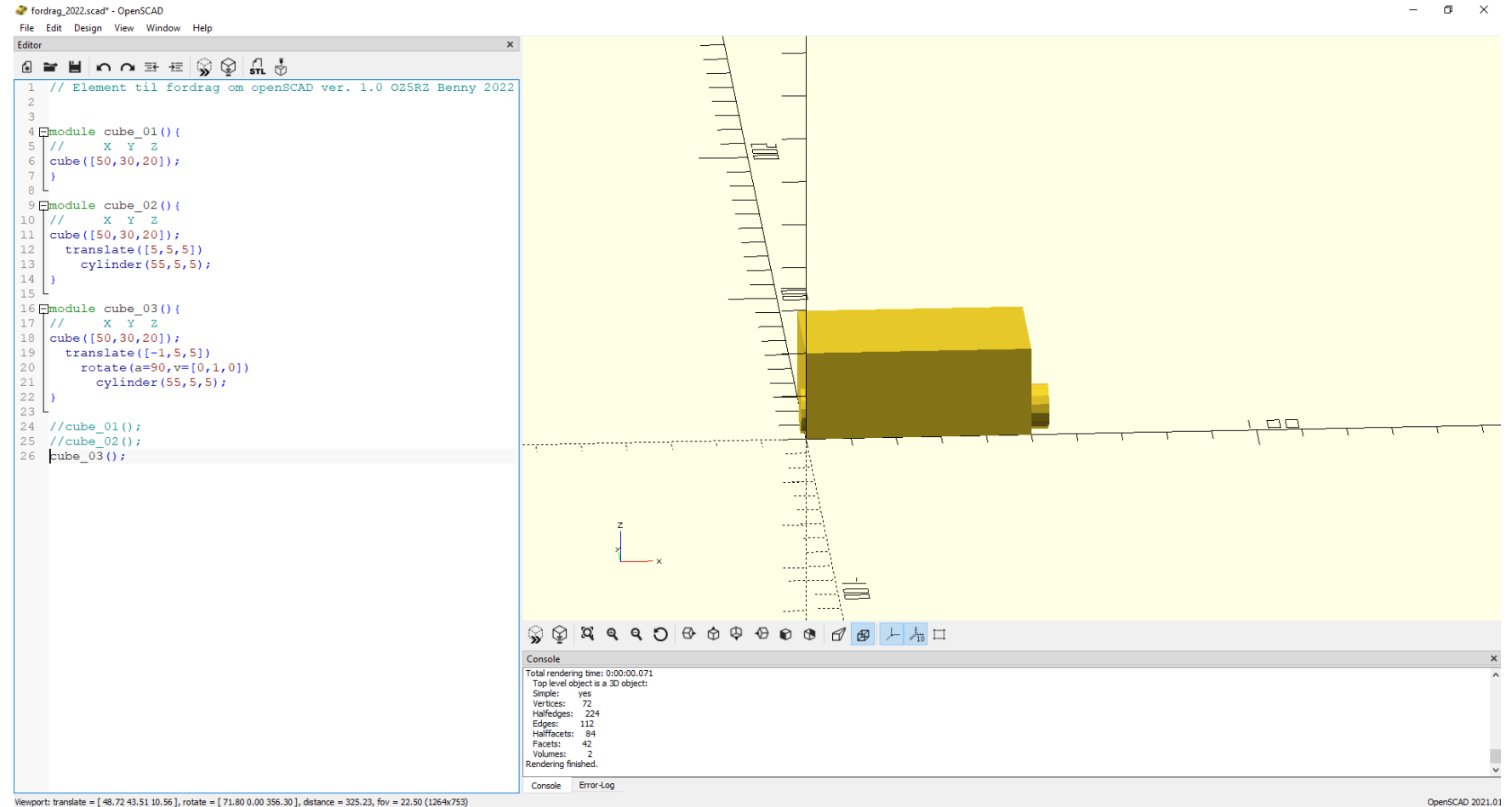
# OpenSCAD 4

Nu er module cube\_02()  
deaktiveret.

Og module cube\_03() er  
aktiveret, med kommandoen  
'rotate(a=90,v=[0,1,0])'  
har lagt vores emne ned.

a=grader v=retning.

Bemærk at jeg med -1 i X  
positionen i kommandoen  
'translate', har gjort cylinderen  
synlig i begge ender af vores  
cube, for at vise at emnet går  
hele vejen igennem.





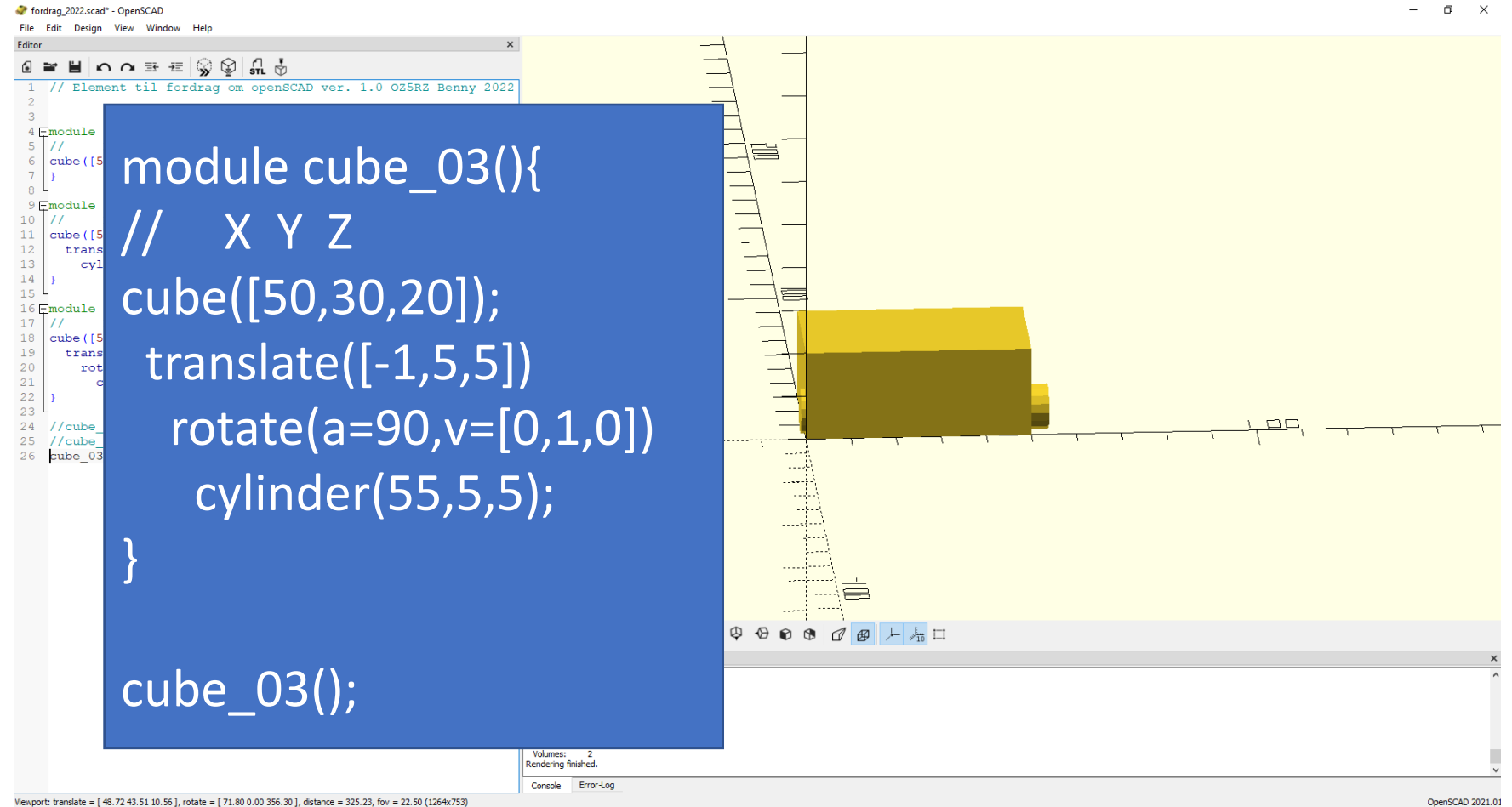
# OpenSCAD 4

Nu er module cube\_02()  
deaktiveret.

Og module cube\_03() er  
aktiveret, med kommandoen  
'rotate(a=90,v=[0,1,0])'  
har lagt vores emne ned.

a=grader v=retning.

Bemærk at jeg med -1 i X  
positionen i kommandoen  
'translate', har gjort cylinderen  
synlig i begge ender af vores  
cube, for at vise at emnet går  
hele vejen igennem.



# OpenSCAD 5

Nu er module cube\_03()  
deaktiveret

Og cube\_04() er aktiveret

Og med kommandoen  
Difference() { kode }  
har gjort emnet cylinder  
**negativt**, så der nu skabes et  
hul i vores cube.

The screenshot shows the OpenSCAD interface. On the left is the code editor with the following SCAD code:

```
1 // Element til fordrag om openSCAD ver. 1.0 Oz5RZ Benny 2022
2
3 module cube_01(){
4 // X Y Z
5 cube([50,30,20]);
6 }
7 module cube_02(){
8 // X Y Z
9 cube([50,30,20]);
10 translate([5,5,5])
11 cylinder(55,5,5);
12 }
13 module cube_03(){
14 // X Y Z
15 cube([50,30,20]);
16 translate([-1,5,5])
17 rotate(a=90,v=[0,1,0])
18 cylinder(55,5,5);
19 }
20 module cube_04(){
21 difference(){
22 // X Y Z
23 cube([50,30,20]);
24 translate([-1,6,6])
25 rotate(a=90,v=[0,1,0])
26 cylinder(55,5,5);
27 }
28 }
29 module cube_05(){
30 $fn=50;
31 difference(){
32 // X Y Z
33 cube([50,30,20]);
34 translate([-1,6,6])
35 rotate(a=90,v=[0,1,0])
36 cylinder(55,5,5);
37 }
38 }
39 //cube_01();
40 //cube_02();
41 //cube_03();
42 cube_04();
43 //cube_05();
44 //cube_06();
45 //cube_07();
```

The 3D view on the right shows a yellow cube with a hole cut through its center. The hole is a cylinder with a radius of 55 and a height of 5, oriented vertically. The hole is positioned at the top of the cube. The hole's surface is rendered with a green and white striped pattern. The 3D view includes a coordinate system with x, y, and z axes. The console at the bottom right shows the following output:

```
Console
Total rendering time: 0:00:00.022
Top level object is a 3D object:
Simple: yes
Vertices: 40
Halfedges: 120
Edges: 60
HalfFaces: 44
Facets: 22
Volumes: 2
Rendering finished.
```

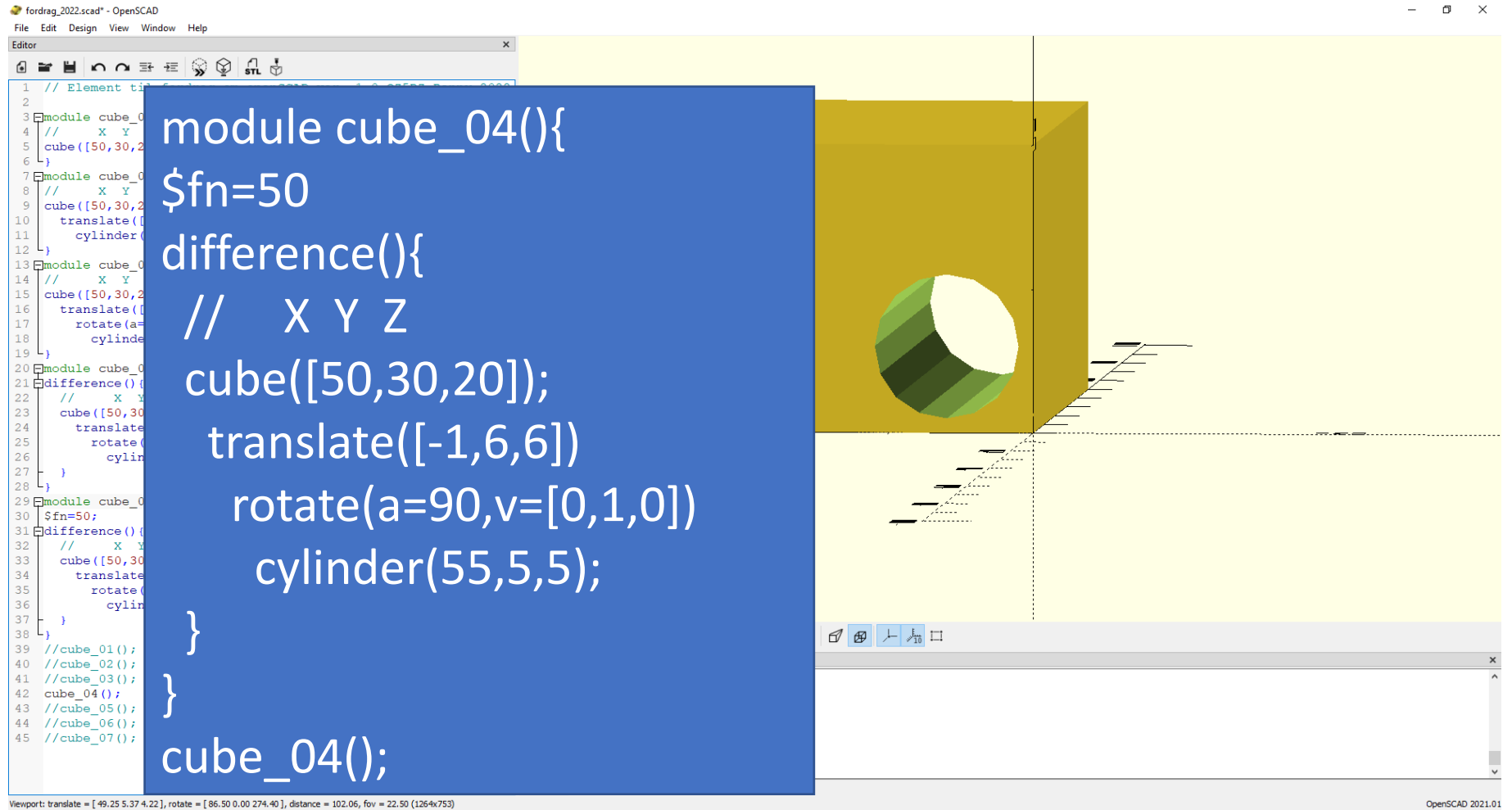
The status bar at the bottom of the window displays the view parameters: Viewport: translate = [ 49.25 5.37 4.22 ], rotate = [ 86.50 0.00 274.40 ], distance = 102.06, fov = 22.50 (1264x753)

# OpenSCAD 5

Nu er module cube\_03()  
deaktiveret

Og cube\_04() er aktiveret

Og med kommandoen  
Difference() { kode }  
har gjort emnet cylinder  
**negativt**, så der nu skabes et  
hul i vores cube.



```
1 // Element til
2
3 module cube_0
4 // X Y
5 cube([50,30,2
6 ])
7 module cube_0
8 // X Y
9 cube([50,30,2
10 translate([
11 cylinder(
12 ])
13 module cube_0
14 // X Y
15 cube([50,30,2
16 translate([
17 rotate(a=
18 cylinde
19 ])
20 module cube_0
21 difference() {
22 // X Y
23 cube([50,30
24 translate
25 rotate(
26 cylin
27 ]
28 }
29 module cube_0
30 $fn=50;
31 difference() {
32 // X Y
33 cube([50,30
34 translate
35 rotate(
36 cylin
37 ]
38 }
39 //cube_01();
40 //cube_02();
41 //cube_03();
42 cube_04();
43 //cube_05();
44 //cube_06();
45 //cube_07();
```

module cube\_04(){  
\$fn=50  
difference(){  
// X Y Z  
cube([50,30,20]);  
translate([-1,6,6])  
rotate(a=90,v=[0,1,0])  
cylinder(55,5,5);  
}  
}  
cube\_04();

Viewport: translate = [ 49.25 5.37 4.22 ], rotate = [ 86.50 0.00 274.40 ], distance = 102.06, fov = 22.50 (1264x753)

OpenSCAD 2021.01

# OpenSCAD 6

Nu er module `cube_04()` deaktiveret

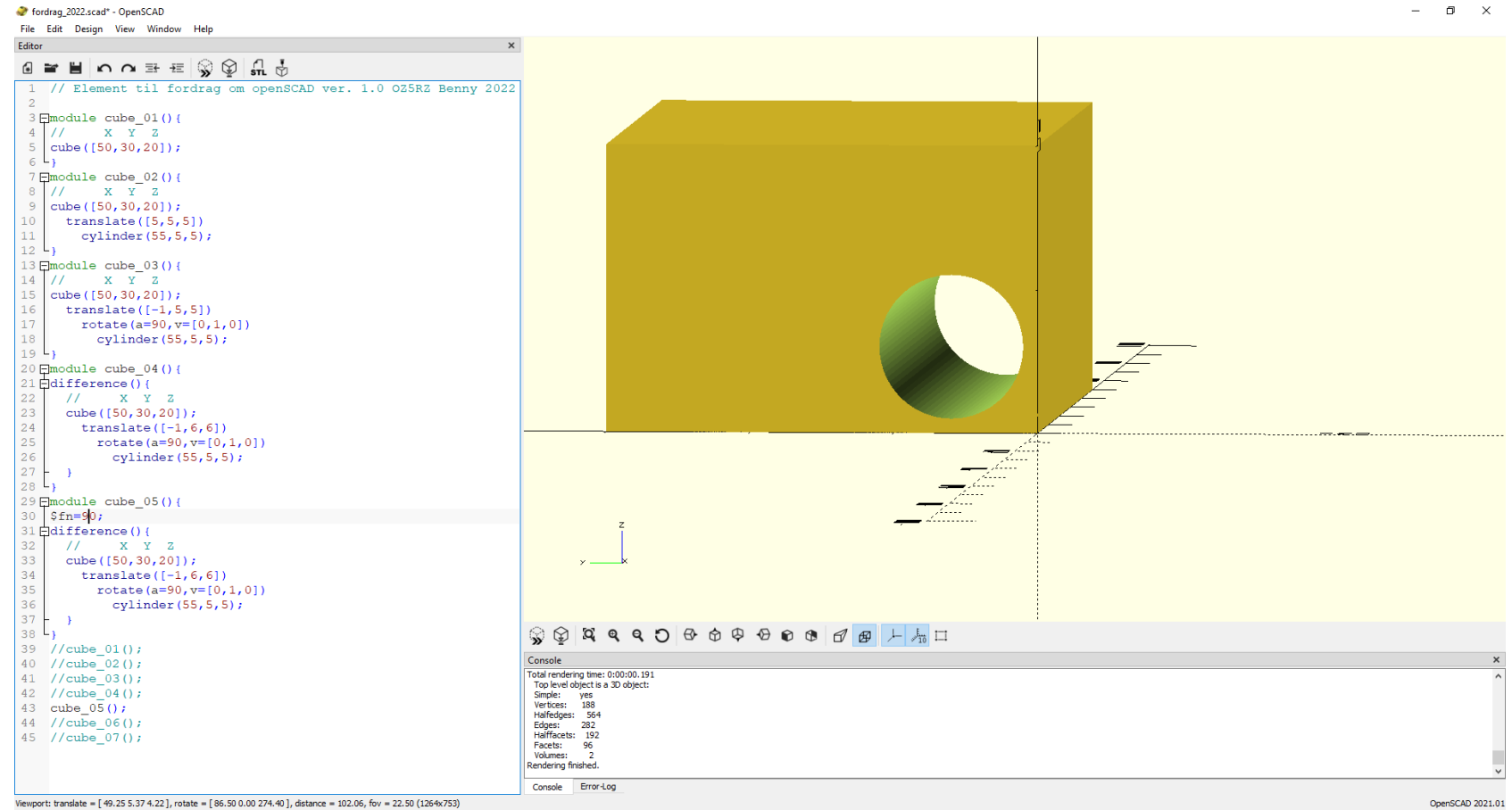
Og `cube_05()` er aktiveret

Her lægger vi mærke til at hullet i vores cube, er blevet jævnt rundt, dette skyldes en 'Special variables' `$fn= 50;` er ændret til 90 fragments.

Tekst med `//` foran, er ikke en del af koden, gælder for **en** linje.

Ved at bruge `/*` i start og `*/` ved slut kan flere linjer tekst indrammes og gøres til ikke aktive program tekst, dette kan bruges til forklarende tekst, eller sovende program stumper.

Ikke aktiv program tekst, er grøn.



# OpenSCAD 6

Nu er module `cube_04()` deaktiveret

Og `cube_05()` er aktiveret

Her lægger vi mærke til at hullet i vores cube, er blevet jævnt rundt, dette skyldes en 'Special variables' `$fn= 50;` er ændret til 90 fragments.

Tekst med `//` foran, er ikke en del af koden, gælder for **en** linje.

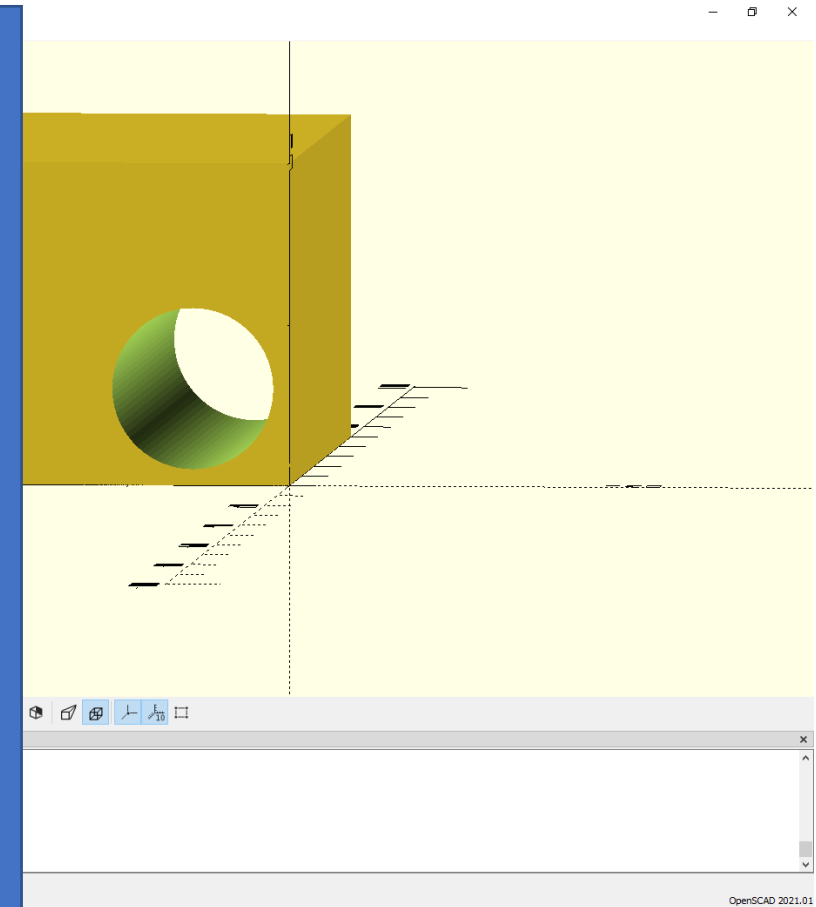
Ved at bruge `/*` i start og `*/` ved slut kan flere linjer tekst indrammes og gøres til ikke aktive program tekst, dette kan bruges til forklarende tekst, eller sovende program stumper.

Ikke aktiv program tekst, er grøn.

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45
```

```
module cube_05(){  
  $fn=90;  
  difference(){  
    // X Y Z  
    cube([50,30,20]);  
    translate([-1,6,6])  
    rotate(a=90,v=[0,1,0])  
    cylinder(55,5,5);  
  }  
}
```

```
cube_05();
```



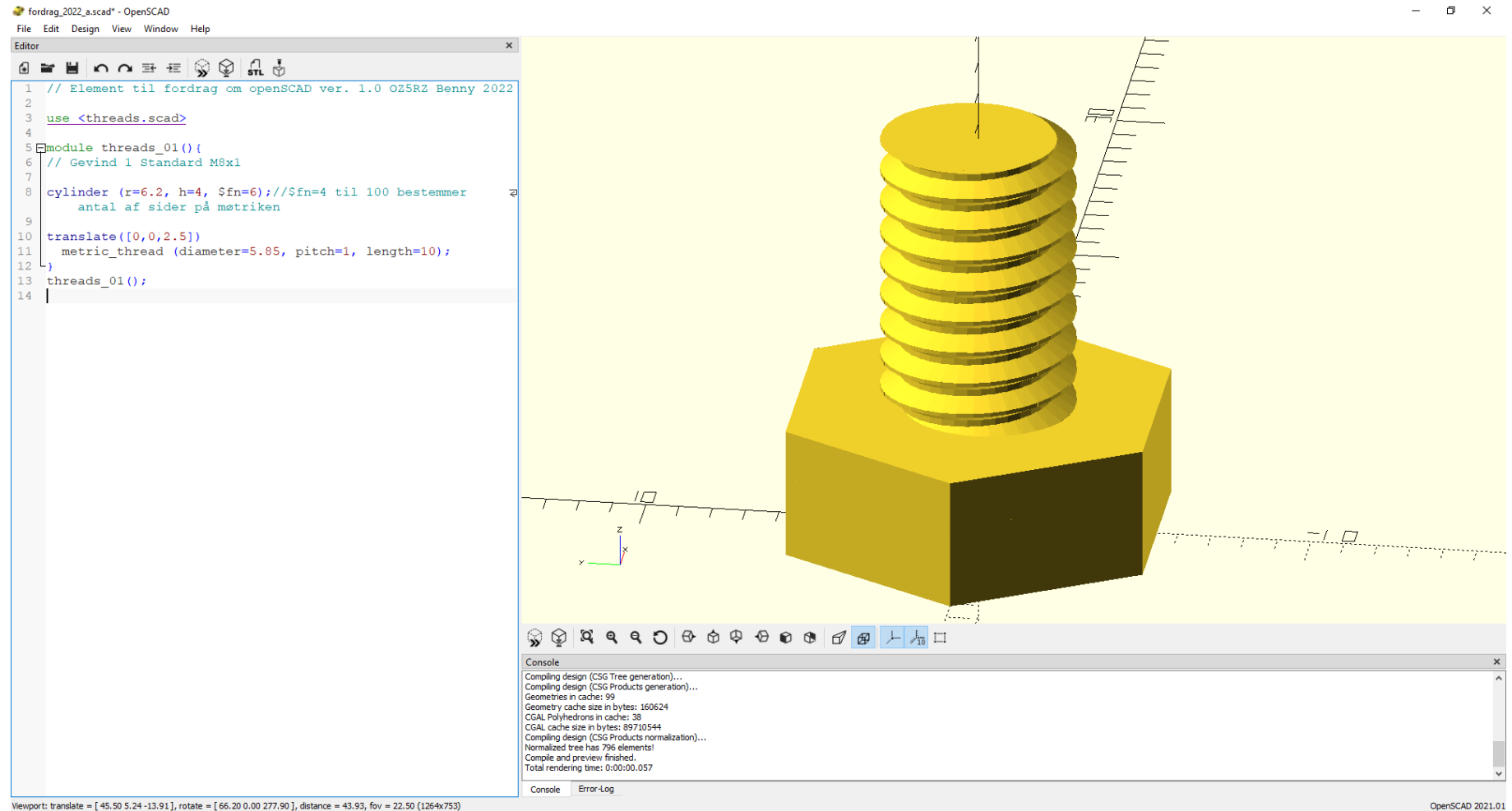
# OpenSCAD 7

I OpenSCAD kan vi ved hjælp af forskellige biblioteker, få hjælp til at udføre ellers meget vanskelige opgaver.

Biblioteket 'threads.scad' giver os mulighed for at lave gevind, både som vist her en skrue eller en møtrik.

Men et positivt eller negativt gevind kan placeres andre steder.

Vigtigt er at bibliotek og kode fil ligger i samme mappe, som det ses her i toppen af koden henvises til biblioteket med 'use <threads.scad>'.

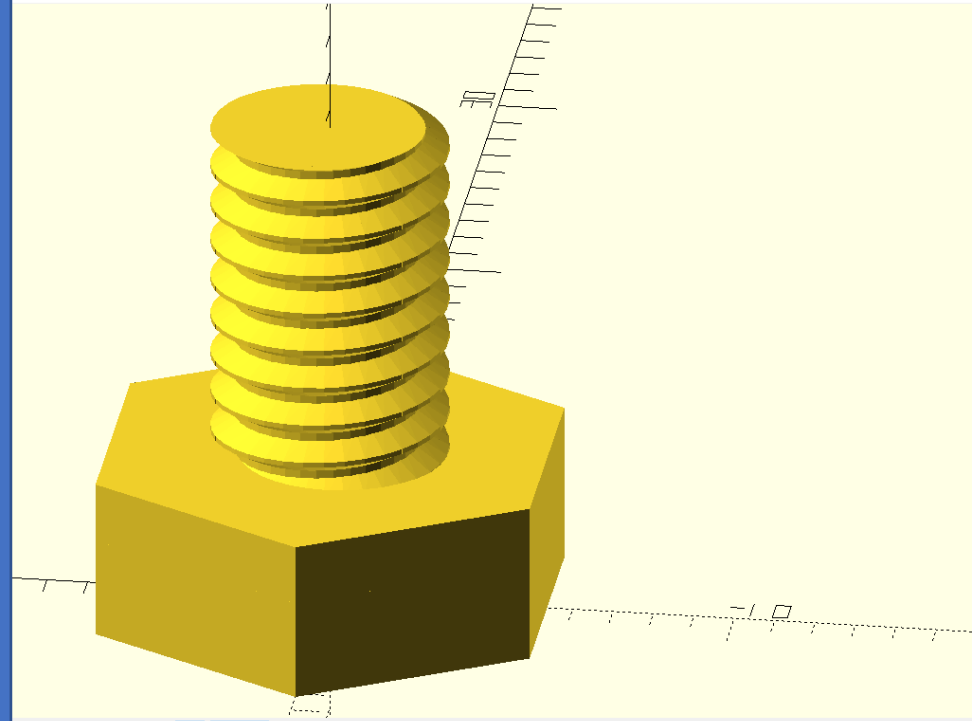


# OpenSCAD 7

```
use <threads.scad>

module threads_01(){
// Gevind 1 Standard M8x1
cylinder (r=6.2, h=4, $fn=6); // $fn=6 sider på skrue
hovedet
translate([0,0,2.5])
  metric_thread (diameter=5.85, pitch=1, length=10);
}

threads_01();
```



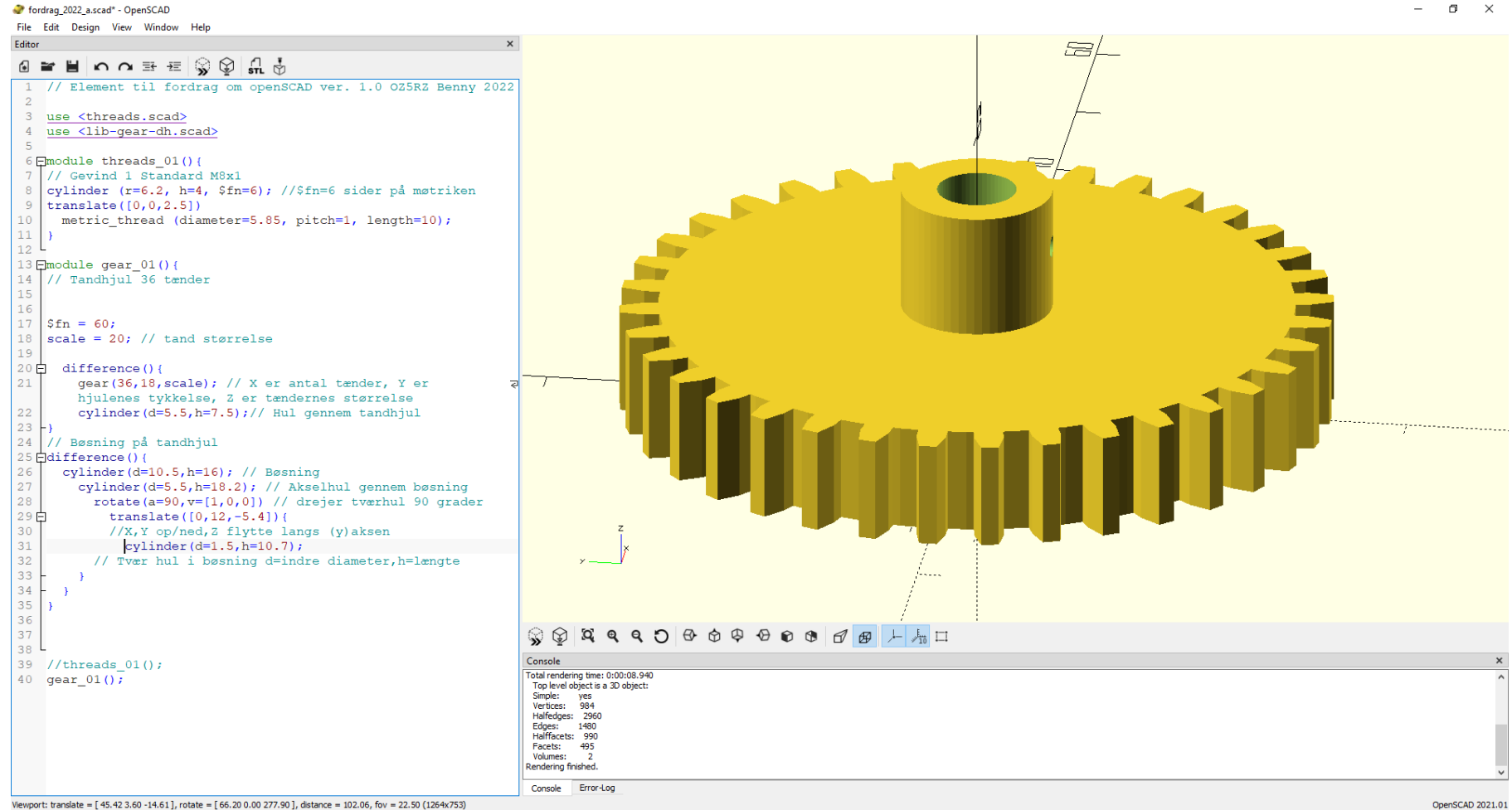
# OpenSCAD 8

Dette er jo en gentagelse af forrige slice, og kan vel mere betragtes som en appetit vækker ;o)

Biblioteket 'lib-gear-dh.scad' giver os mulighed for at lave tandhjul af forskellig størrelse og med forskellig antal tænder.

Og vi kan som f.eks. her hvor jeg har lavet et hul i centrum og påsat en bøsning med et tværhul til fastgørelse.

Vigtigt igen at bibliotek og kode fil ligger i samme mappe.





# OpenSCAD 8

Dette er jo en gentagelse af forrige slice, og kan vel mere betragtes som en appetit vækker ;o)

Biblioteket 'lib-gear-dh.scad' giver os mulighed for at lave tandhjul af forskellig størrelse og med forskellig antal tænder.

Og vi kan som f.eks. her hvor jeg har lavet et hul i centrum og påsat en bøsning med et tværhul til fastgørelse.

Vigtigt igen at bibliotek og kode fil ligger i samme mappe.

```
module gear_01(){ // Tandhjul 36 tænder
  $fn = 60;
  scale = 20; // tand størrelse
  difference(){
    gear(36,18,scale); // X er antal tænder, Y er hjulenes tykkelse,
    // Z er tændernes størrelse
    cylinder(d=5.5,h=7.5); // Hul gennem tandhjul
  }
  difference(){ // Bøsning på tandhjul
    cylinder(d=10.5,h=16); // Bøsning
    cylinder(d=5.5,h=18.2); // Akselhul gennem bøsning
    rotate(a=90,v=[1,0,0]) // drejer tværhul 90 grader
    translate([0,12,-5.4]){ //X,Y op/ned,Z flytte langs (y)aksen
      cylinder(d=1.5,h=10.7); // Tvær hul i bøsning d=indre diameter,
      h=længde
    }
  }
}

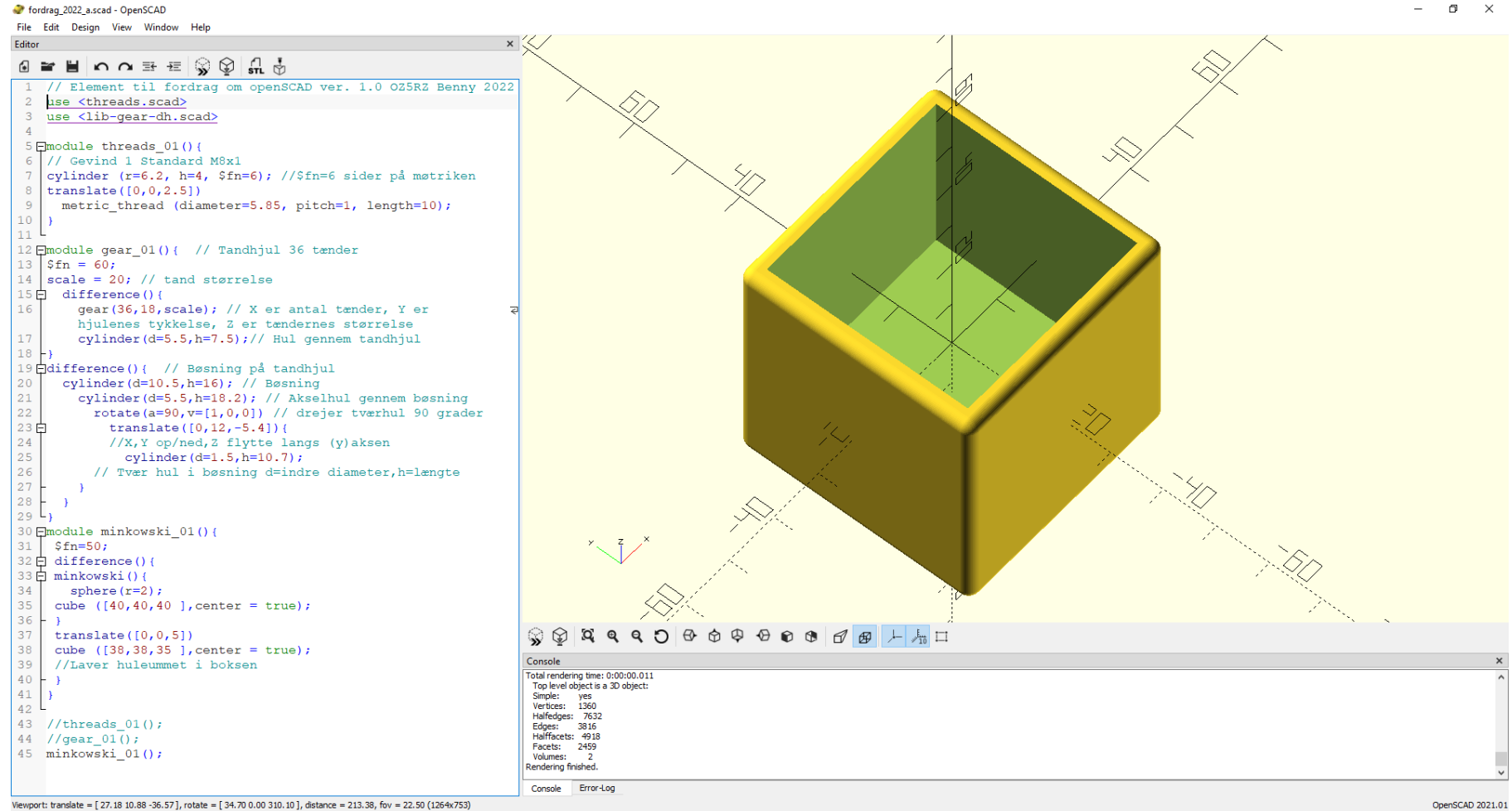
gear_01();
```

for  
File  
Editor  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
Viewpor

# OpenSCAD 9

Kommandoer i OpenSCAD er også interessante, her f.eks. 'minkowski()'.

Bemærk de runde former, og brugen af 'difference' til at gøre den indvendige cube negative, der så bliver til rummet i den yderste cube.



# OpenSCAD 9

Kommandoer i OpenSCAD er også interessante, her f.eks. 'minkowski()'.

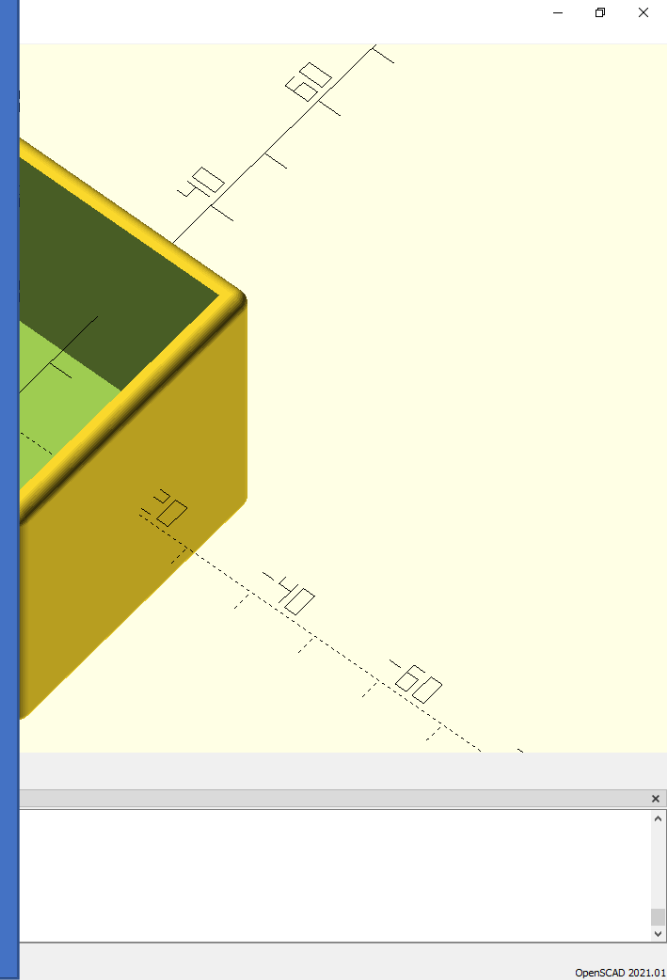
Bemærk de runde former, og brugen af 'difference' til at gøre den indvendige cube negative, der så bliver til rummet i den yderste cube.

Når et projekt er klar til at blive 3D printet, gemmes det som en fil med navn.**stl** som er det filformat Slicer programmet forstår.

```
module minkowski_01(){
  $fn=50;

  difference(){
    minkowski(){
      sphere(r=2); // rundings grad ydre cube
      cube ([40,40,40],center=true);
    }
    translate([0,0,5])
      cube ([38,38,35 ],center=true);
    //Laver hulrummet i boksen
  }
}

minkowski_01();
```



# OpenSCAD 10

Dette er et udsnit af OpenSCADs hjælpe menu, benævnt Cheatsheet.

Det vigtige er jo web adressen:

<https://openscad.org/cheatsheet/>

Når vi dobbelt klikker på en af de blå linjer, får vi en beskrivelse frem.

Yderlig information og undervisning, ligger på denne hjemmeside:

<https://openscad.org/documentation.html>

Her ligger bla. 9 Kapitler om brugen af OpenSCAD, i form af PDF filer, her er megen god viden at hente.

Det var lidt opvarmning til brug af OpenSCAD til at lave spændende ting på klubbens eller andre 3D printere.

Kan I klare at høre mere vil jeg vise et par eksempler her på min Klapsammen PC.

2022 OZ5RZ Benny

The screenshot shows a web browser window with the title "OpenSCAD CheatSheet". The address bar contains the URL "https://openscad.org/cheatsheet/index.html?version=2019.05". The page content is organized into several sections:

- Syntax**:
  - `var = value;`
  - `var = cond ? value_if_true : value_if_false;`
  - `var = function (x) x + x;`
  - `module name(...) { ... }`  
`name();`
  - `function name(...) = ...`  
`name();`
  - `include <...scad>`
  - `use <...scad>`
- Constants**:
  - `undef` undefined value
  - `PI` mathematical constant  $\pi$  (~3.14159)
- Operators**:
  - `n + m` Addition
  - `n - m` Subtraction
  - `n * m` Multiplication
  - `n / m` Division
  - `n % m` Modulo
  - `n ^ m` Exponentiation
  - `n < m` Less Than
  - `n <= m` Less or Equal
- Modifier Characters**:
  - `*` disable
  - `!` show only
  - `#` highlight / debug
  - `%` transparent / background
- 2D**:
  - `circle(radius | d=diameter)`
  - `square(size,center)`
  - `square([width,height],center)`
  - `polygon([points])`
  - `polygon([points],[paths])`
  - `text(t, size, font, halign, valign, spacing, direction, language, script)`
  - `import("...ext")`
  - `projection(cut)`
- 3D**:
  - `sphere(radius | d=diameter)`
  - `cube(size, center)`
  - `cube([width,depth,height], center)`
  - `cylinder(h,r|d,center)`
  - `cylinder(h,r1|d1,r2|d2,center)`