

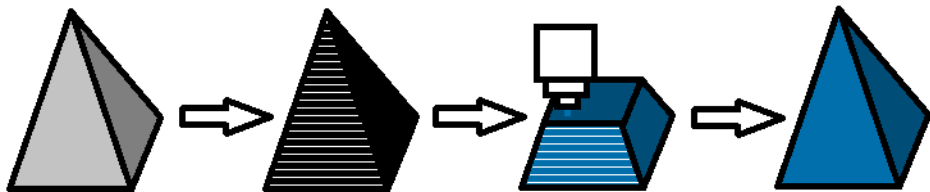
Modelování

KMI/3DT 3D tisk

Mgr. Markéta Trnečková, Ph.D.
www.marketa-trneckova.cz



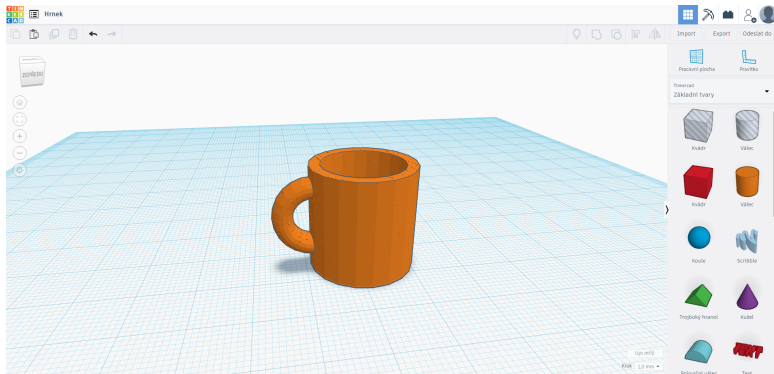
Palacký University, Olomouc

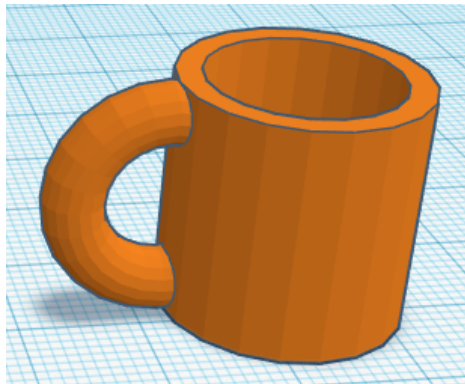
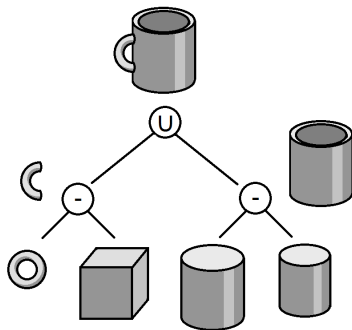


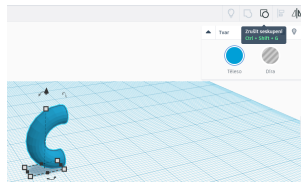
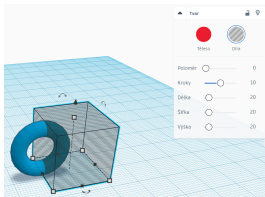
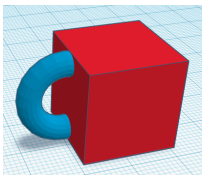
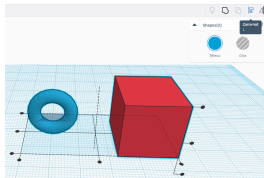
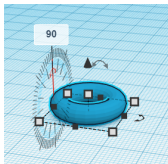


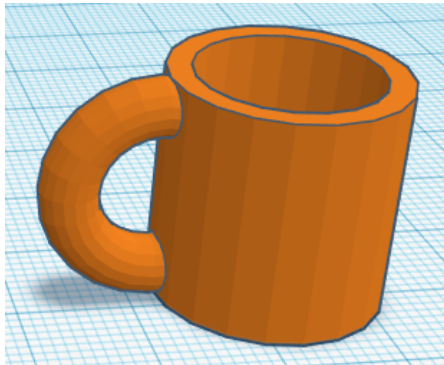
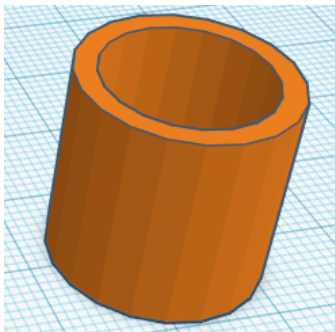
- stáhnout již existující
- použít 3D skenner
- vytvořit vlastní model
 - CAD, 3D modelovací program
 - parametrický CAD

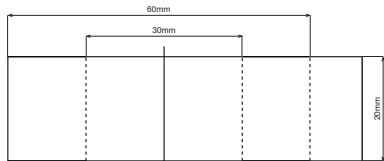
- www.tinkercad.com



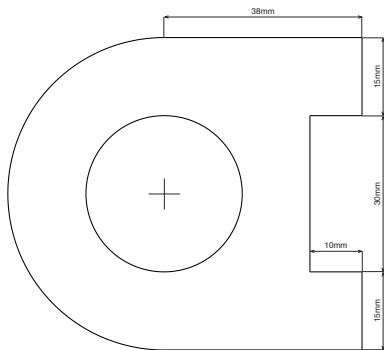




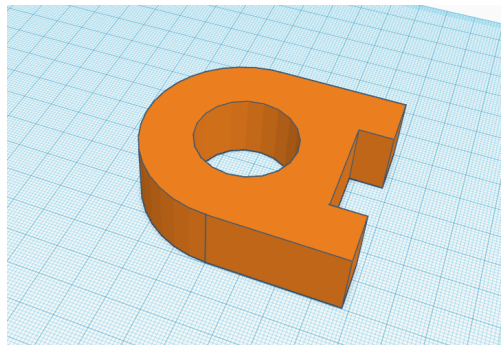


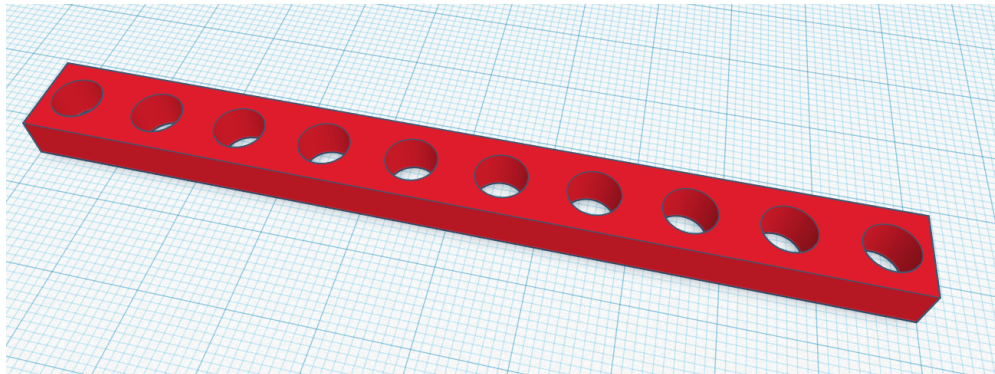


pohled z boku




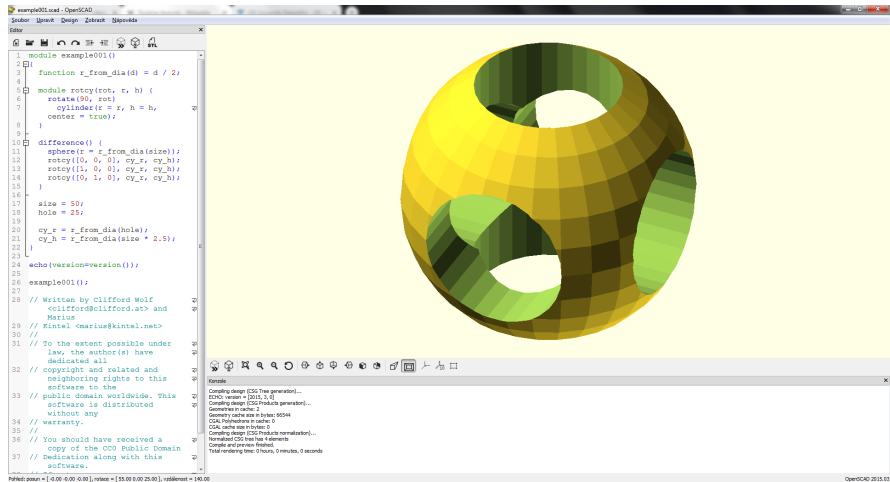
pohled ze shora





```
panelscad - OpenSCAD
Soubor Upravit Design Zobrazit Nápověda
Editor
1 $fn = 100;
2 r = 3;
3 delka = 100;
4 pocet = 10;
5 sirka = 10;
6 vyska = 5;
7
8 rozestup = (delka - 2*r - 2)/(pocet - 1);
9
10 difference()
11 [
12   translate([delka/2,0,0]) cube([delka, sirka, vyska], center=true);
13
14   for( i = [r : rozestup : delka - r])
15   {
16     translate([i,0,0])
17     cylinder(h=vyska, r=r, center=true);
18   }
19 ]
20
Pohled: posun = [48.57 16.00 -15.50], rotace = [41.70 0.00 4.00], vzd@lenost = 213.38 (636x470)
Konzole
OpenSCAD 2019.05
```





<https://www.openscad.org>

- objekty
- akce (vytvoření objektu, přiřazení hodnot proměnným,...)
- operace (transformace, množinové operace, změna vzhledu, ...) { }
- komentář //, /* */



- čísla
- logické hodnoty
- řetězce
- interval
- vektor
- nedefinovaná hodnota (undef), not a number (nan)

- funkcionální programovací jazyk
- `nazev = hodnota;`
- `pole = vektory`

Example (Jaký bude výstup?)

```
a = 3;  
echo(a);  
a = 4;  
echo(a);
```

Je v pořádku konstrukce `x = x + 1;`?

Example (Jaký bude výstup?)

```
a = 0;  
if ( a == 0 )  
{  
    a = 1;  
}
```



Example

```
a = 1;
translate ( [0,0,0] )
{
  a = 2;
  echo(a);
}
echo(a);
```


Example

```
a = 1;
translate ( [0,0,0] ){
  a = 2;
  echo(a);
}
translate ( [0,0,0] ){
  echo(a);
  translate ( [0,0,0] ){
    echo(a);
    a = 3;
  }
}
echo(a);
```



Example

```
a = 1;  
{  
  a = 2;  
}  
echo(a);
```

Vektory

- []
- e[5]

Matice

- [[], []]
- e[5][2]

Example

```
e = [[1], [ ], [3,4,5], "string", "x", [[1,2], [3,4,5], [[6,7], [8,9]]];
```

Jaká je délka e?

e[0] =

e[1] =

e[2] =

e[3] =

e[3][1] =

...



- `concat()`
- `len()`

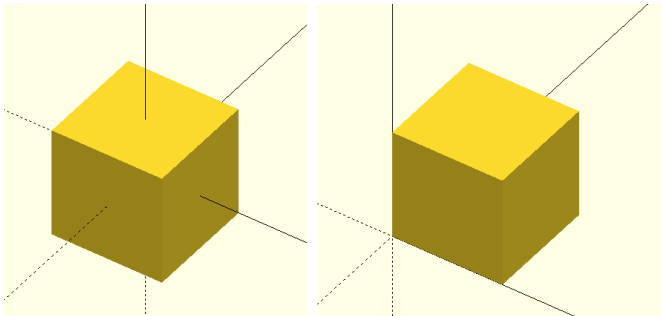
Example

```
a = [1,2];  
b = [3,4];  
c = [a,b];  
d = concat(a,b);
```



- `+`, `-`, `*`, `/`, `%`
- `cos()`, `sin()`, `tan()`, `acos()`, `asin()`, `atan()`
- `abs()`, `ceil()`, `round()`, `floor()`, `exp()`, `ln()`, `log()`, `max()`, `min()`,
`pow()`, `rands()`, `sign()`, `sqrt()`
- `lookup`

- cube
- size
- center, false
- `cube(size = 5, center = true);`



Example

```
cube(size = 5, center = true);  
cube(5, true);  
cube(5);
```

Example

```
cube(true, 5);  
cube(center = true, size = 5);
```

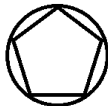
- `cylinder`
- `h`
- `r`
- `r1`
- `r2`
- `d`
- `d1`
- `d2`
- `center, false`
- `$fa`
- `$fs`
- `$fn`

Jak udělat pomocí `cylinder` kužel?

f_a = minimální úhel každého fragmentu

f_s = minimální délka oblouku

f_n = počet fragmentů v 360°



Co se stane, když zvolíme nízké f_n ?

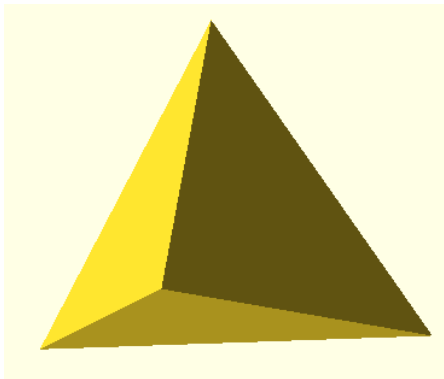
Example

```
cylinder(h =5, d =10,  $f_n=3$ );
```

Pravidelný čtyřstěn



Vytvořte pravidelný čtyřstěn s délkou hrany 10 jednotek.





- `sphere`
- `r, 1`
- `d`
- `$fa, 12`
- `$fs, 2`
- `$fn, 1`



- polyhedron
- points
- triangles, faces
- convexity

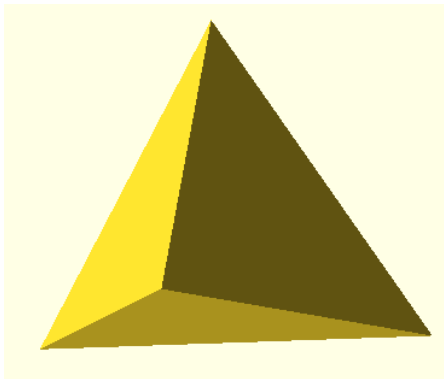
Example

```
polyhedron(  
  points=[[5,5,0],[5,-5,0],[-5,-5,0],[-5,5,0],[0,0,10]],  
  faces=[[0,1,4],[1,2,4],[2,3,4],[3,0,4],[1,0,3],[2,1,3]]  
);
```

Pravidelný čtyřstěn



Vytvořte pravidelný čtyřstěn s délkou hrany 10 jednotek pomocí mnohostěnu.





- `scale([x, y, z])`
- `resize([x, y, z], auto)`
- `rotate(a, [x, y, z]), rotate([x, y, z])`
- Jak převrátit objekt?
- `translate([x, y, z])`
- `mirror([x, y, z])`
- `multimatrix(m = [...])`



V jakém pořadí se vykonají jednotlivé operace?

```
translate([0,20,0]) rotate([90,0,0]) cylinder(h=20,r=10);  
rotate([90,0,0]) translate([0,20,0]) cylinder(h=20,r=10);
```

$$\begin{bmatrix}
 \text{scale X} & \text{scale X sheared along Y} & \text{scale X sheared along Z} & \text{translate X} \\
 \text{scale Y sheared along X} & \text{scale Y} & \text{scale Y sheared along Z} & \text{translate Y} \\
 \text{scale Z sheared along X} & \text{scale Z sheared along Y} & \text{scale Z} & \text{translate Z} \\
 0 & 0 & 0 & 1
 \end{bmatrix}$$

`translate([10,20,30]) rotate([0,0,45])`

$$\begin{bmatrix}
 \cos(45) & -\sin(45) & 0 & 10 \\
 \sin(45) & \cos(45) & 0 & 20 \\
 0 & 0 & 1 & 30 \\
 0 & 0 & 0 & 1
 \end{bmatrix}$$



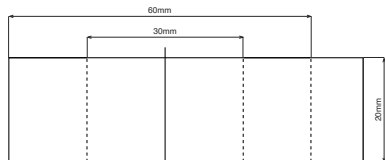
- `union()`
- `difference()`
- `intersection()`



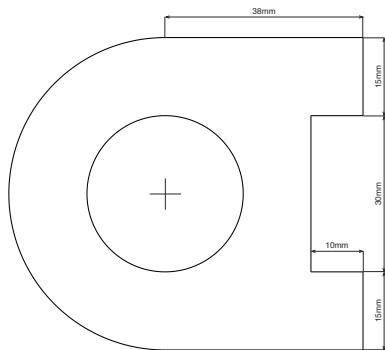
```
union(){  
    translate([-10,0,0]) sphere(r=5);  
    rotate([0,90,0]) cylinder(r=2, h=20, center=true);  
}
```



- `color("red")`, `color([r,g,b])`, `color([r,g,b,a])`
- `#`
- `%`
- `!`



pohled z boku



pohled ze shora

