Didatticaduepuntozero:
Generazione Web 2017/18

Tecnologia creativa: Progettare e creare in 3D

Modellazione 3D

Usare OpenScad

Il sito ufficiale: http://www.openscad.org/

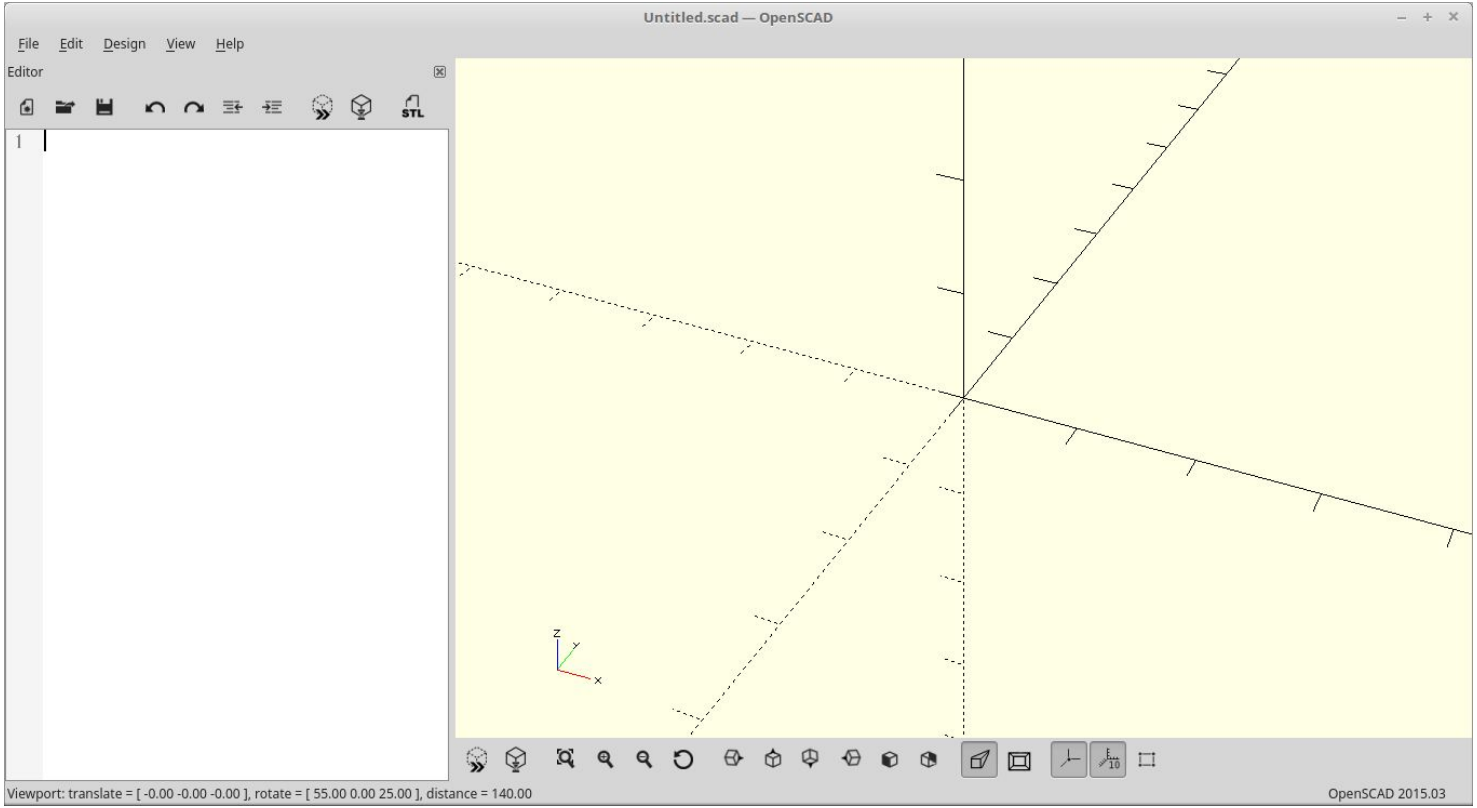
Il Manuale utente Wikibook:

https://en.wikibooks.org/wiki/OpenSCAD_User_Manual

Un tutorial:

http://edutechwiki.unige.ch/en/OpenScad_beginners_tutorial#Some_3D_principles

Interfaccia utente



Oggetti primitivi 3D /cube

```
cube(size = [x,y,z], center = true/false);
cube(size =  x ,     center = true/false);

default values:  cube();   yields:  cube(size = [1, 1, 1],
center = false);
```

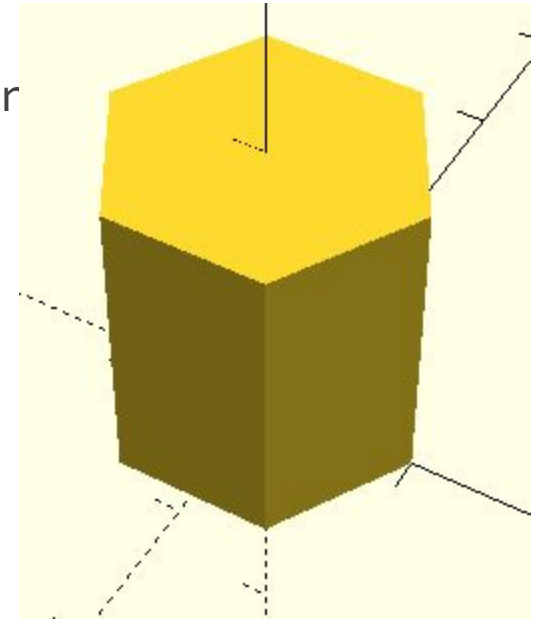
Oggetti primitivi 3D /sphere e cylinder

~~sphere~~(r/d = 10,$fn=32);
r = raggio; d= diametro; $fn = risoluzione

cylinder(h = height, r1 = BottomRadius, r2 = TopRadius, center = true/false);
parametri:
h : height of the cylinder or cone
r : radius of cylinder. r1 = r2 = r.
r1 : radius, bottom of cone
r2 : radius, top of cone.
d : diameter of cylinder.
d1 : diameter, bottom of cone
d2 : diameter, top of cone. r2 = d2 /2

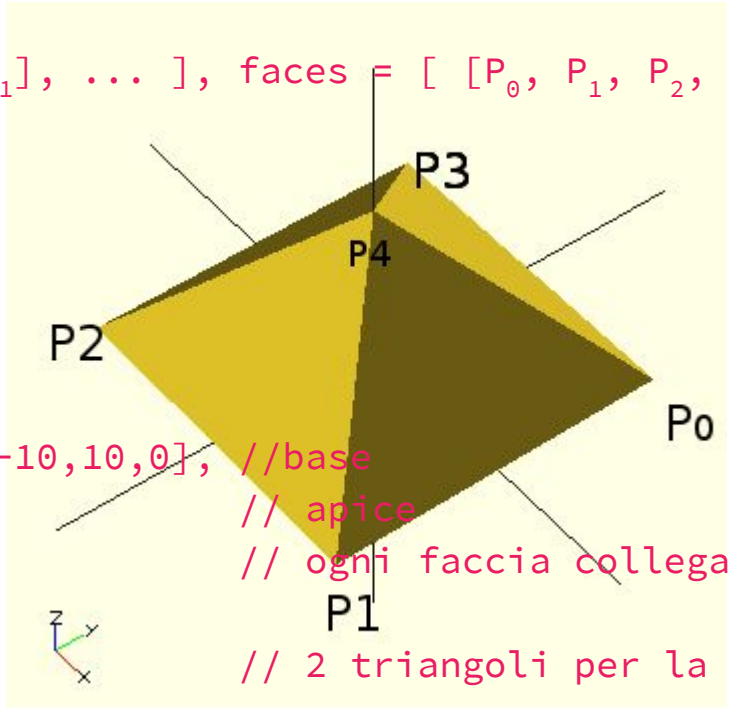
Creare prismi con base regolare

```
cylinder(r=10,h=20,,$fr
```



Oggetti primitivi 3D /polyhedron

polyhedron(points = [[X_0, Y_0, Z_0], [X_1, Y_1, Z_1], ...], faces = [[P_0, P_1, P_2, P_3, ...], ...], convexity = N);




```
polyhedron(
  points=[ [10,10,0],[10,-10,0],[-10,-10,0],[-10,10,0], //base
           [0,0,10]  ],                                  // apice
  faces=[ [0,1,4],[1,2,4],[2,3,4],[3,0,4],               // ogni faccia collega
i punti
             [1,0,3],[2,1,3] ]                          // 2 triangoli per la
base
 );
```

Oggetti primitivi 2D

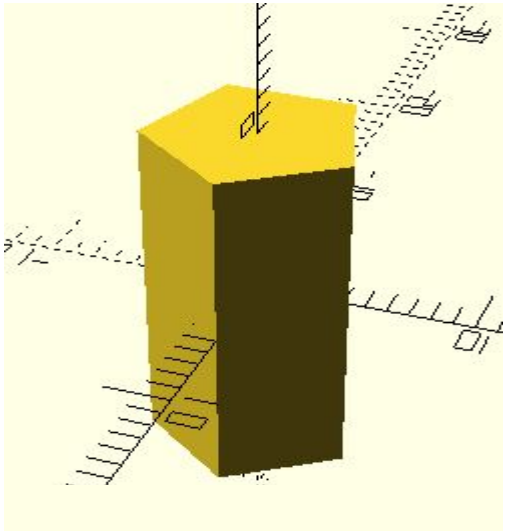
```
square(size = [x, y], center = true/false);//genera
rettangoli

circle(r=radius | d=diameter);//genera cerchi

circle(20,$fn=5);// genera un pentagono

polygon(points = [ [x, y], ... ], paths = [ [p1, p2, p3..],
...]);//genera un poligono (con eventuali “buchi”)

polygon(points=[[0,0],[100,0],[0,100],[10,10],[80,10],[10,80
]], paths=[[0,1,2],[3,4,5]]);//una squadra da disegno
```

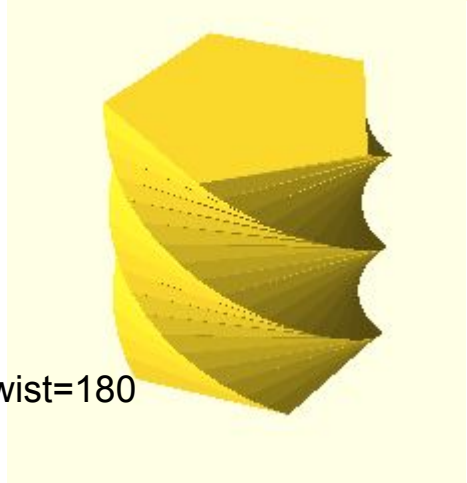
Da un modello 2D a uno 3D

```
linear_extrude(height = fanwidth, center =
true, convexity = 10, twist = -fanrot,
slices = 20, scale = 1.0)
```

esempi:

```
linear_extrude(height = 20, center = true,
convexity = 10, twist = 0, slices = 20,
scale = 1.0)circle(r=5,$fn=5);
```





twist=180

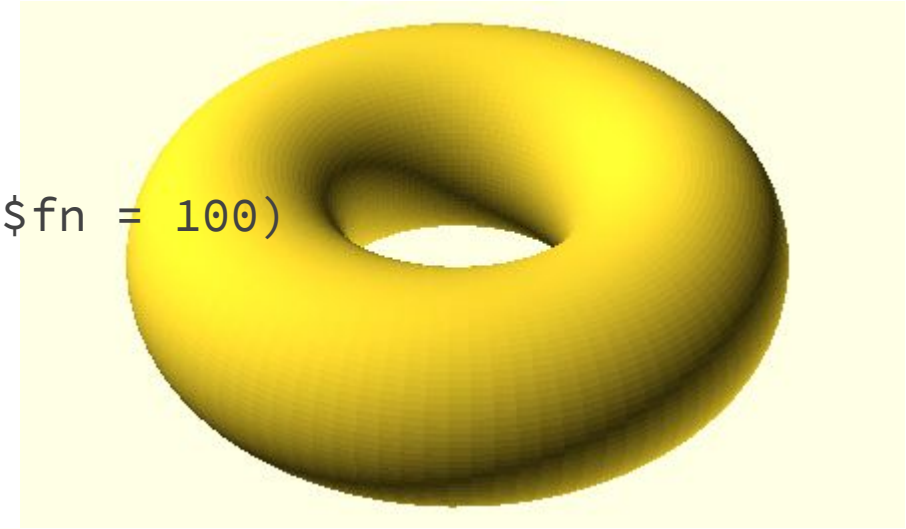
Da un modello 2D a uno 3D

```
rotate_extrude(angle = 360, convexity = 2)
```

genera un solido di rotazione attorno all'asse z
a partire da un oggetto 2D

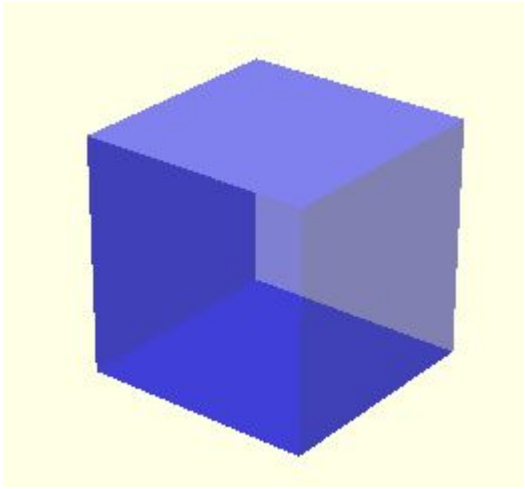
esempio:

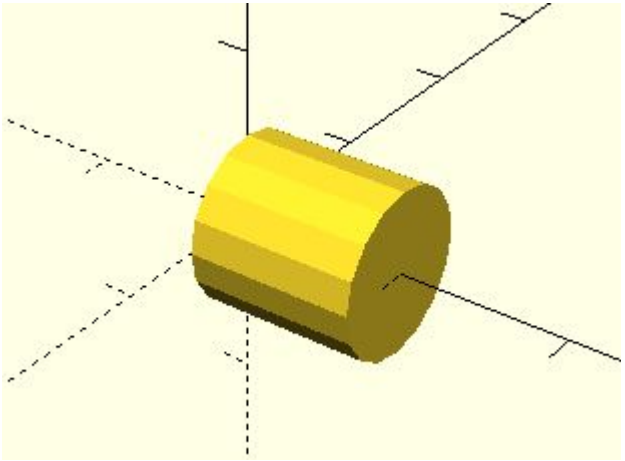
```
rotate_extrude(convexity = 10, $fn = 100)
translate([2, 0, 0])
circle(r = 1, $fn = 100);
```



Trasformazioni

```
color("Blue",0.5) cube(5);
//colora il cubo di blu,
con trasparenza del 50%

rotate([0, 90, 0])
cylinder(r=5,h=10);

//ruota il cilindro di 90°
attorno all’asse y
```

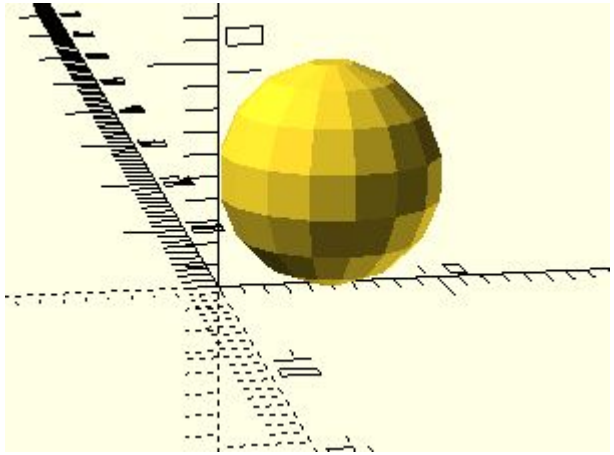


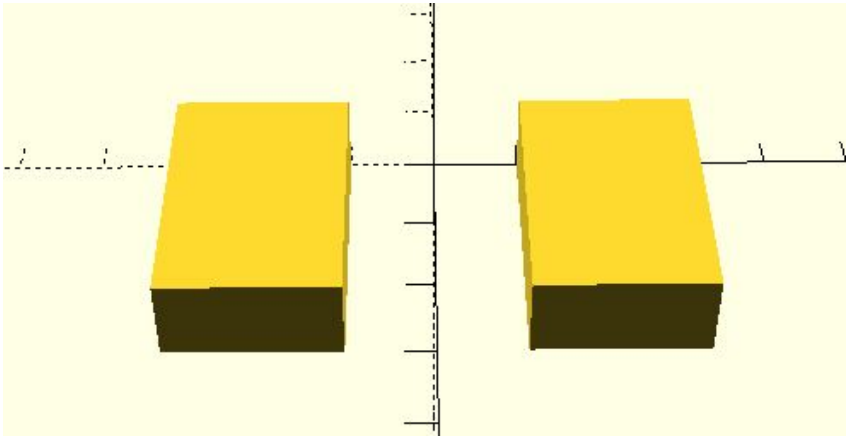


Trasformazioni

___

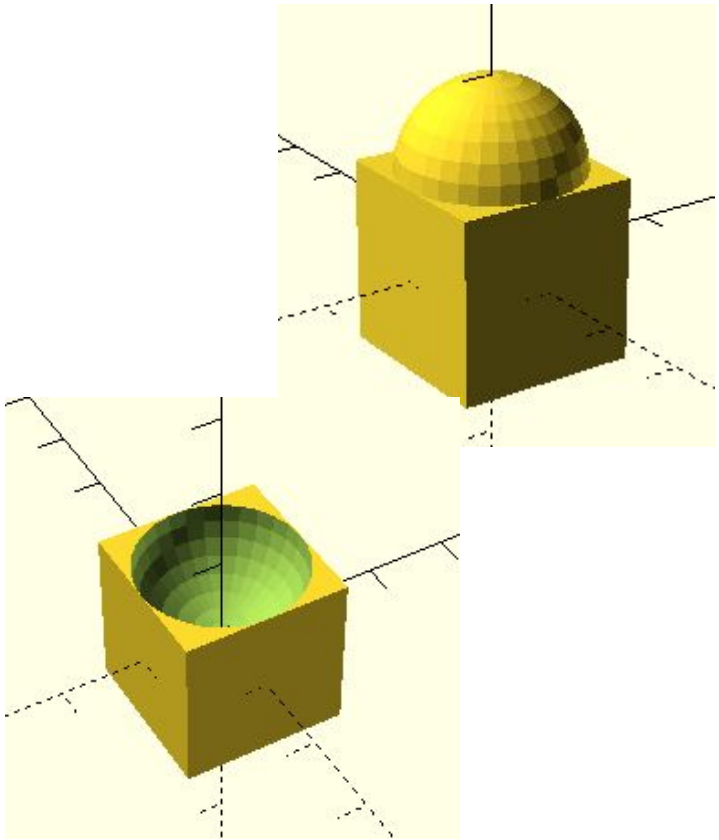
```
translate([5,0,5])
sphere(5,center = true);
//trasla la sfera di 5 unità
lungo gli assi x e z

translate
([0,10,0])cube([30,20,10]);

mirror([0,1,0])translate
([0,10,0])cube([30,20,10]);
//riflette il parallepipedo
rispetto al piano xz
```



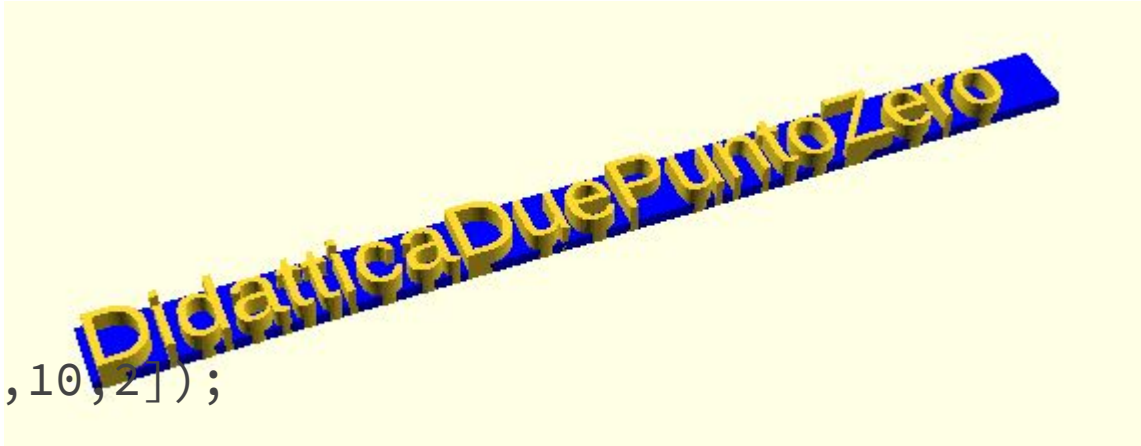


Operazioni booleane

```
---
union () {
cube(20,center=true);
translate([0,0,10])sphere(10
);
}

difference () {
cube(20,center=true);
translate([0,0,10])sphere(10
);
}
```



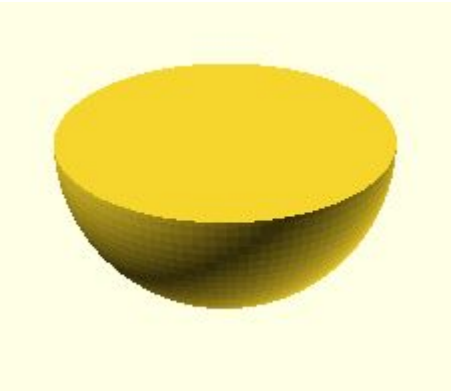
Testo




```
color("Blue",1)cube([150,10,2]);


linear_extrude(5) { text("DidatticaDuePuntoZero", font
= "Liberation Sans");
}

//parametri: size (default:10), font
```

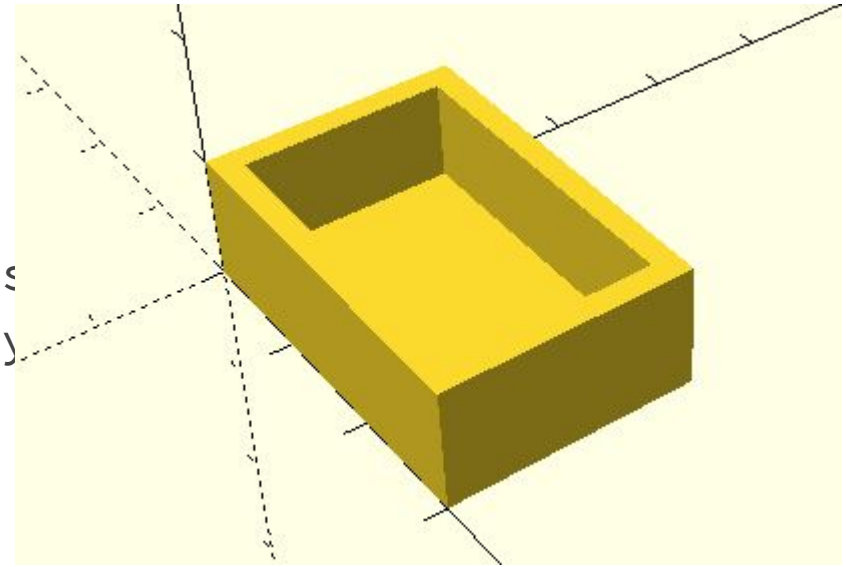
Operazioni booleane

```
intersection () {
cube(20,center=true);
translate([0,0,10])sphere(10
);
}
```



Moduli

```
module scatola(x,y,z,s){
    cube([x,y,s]);
    cube([s,y,z]);
    cube([x,s,z]);
    translate([0,y-s,0]) cube([x,s
    translate([x-s,0,0]) cube([s,y
}

scatola(30,20,10,2);
```



Compilazione ed esportazione

---

1. Design > Render (F6)
2. File > Export > Export as STL