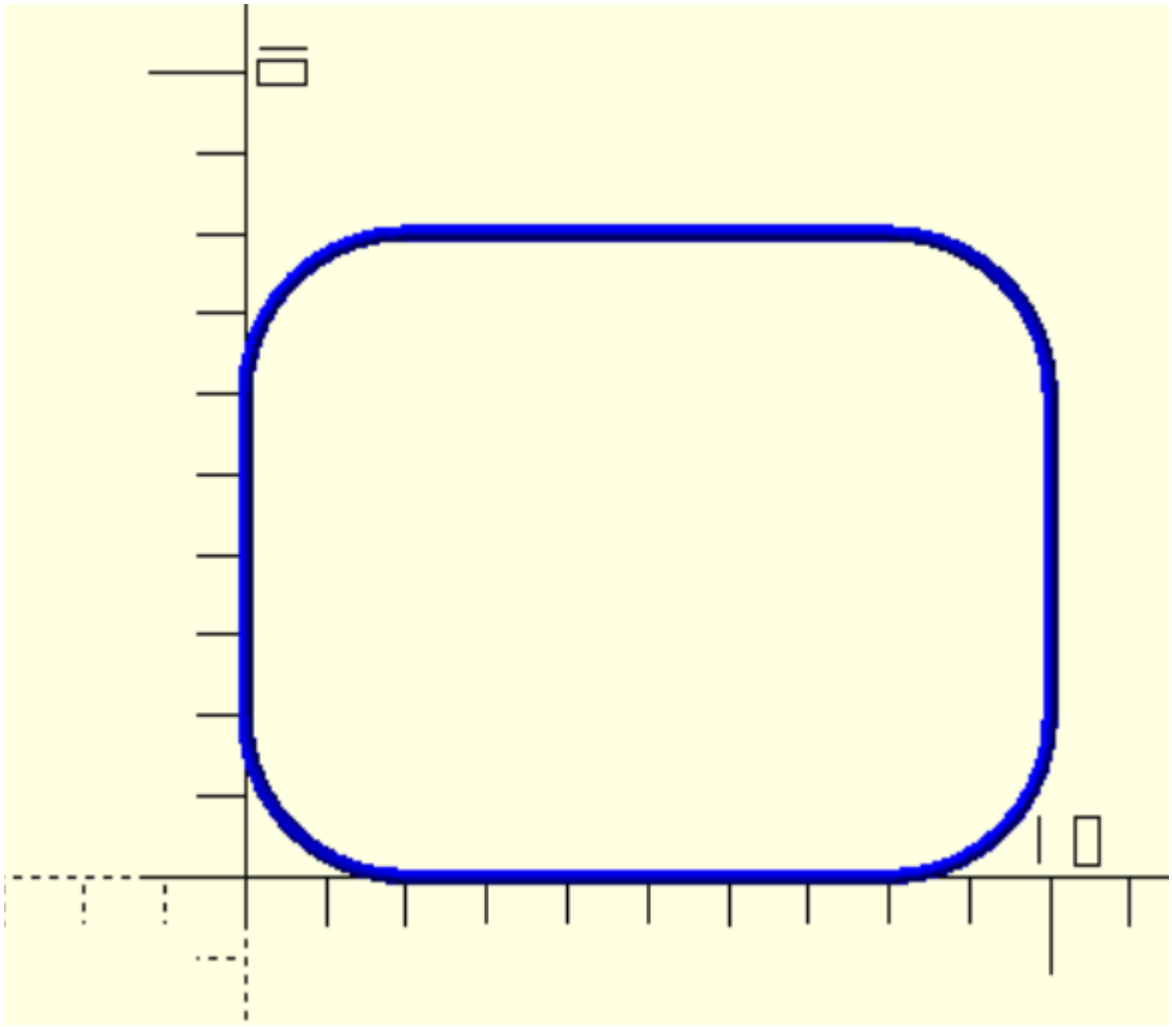
In [1]:
```
from openscad1 import *
```

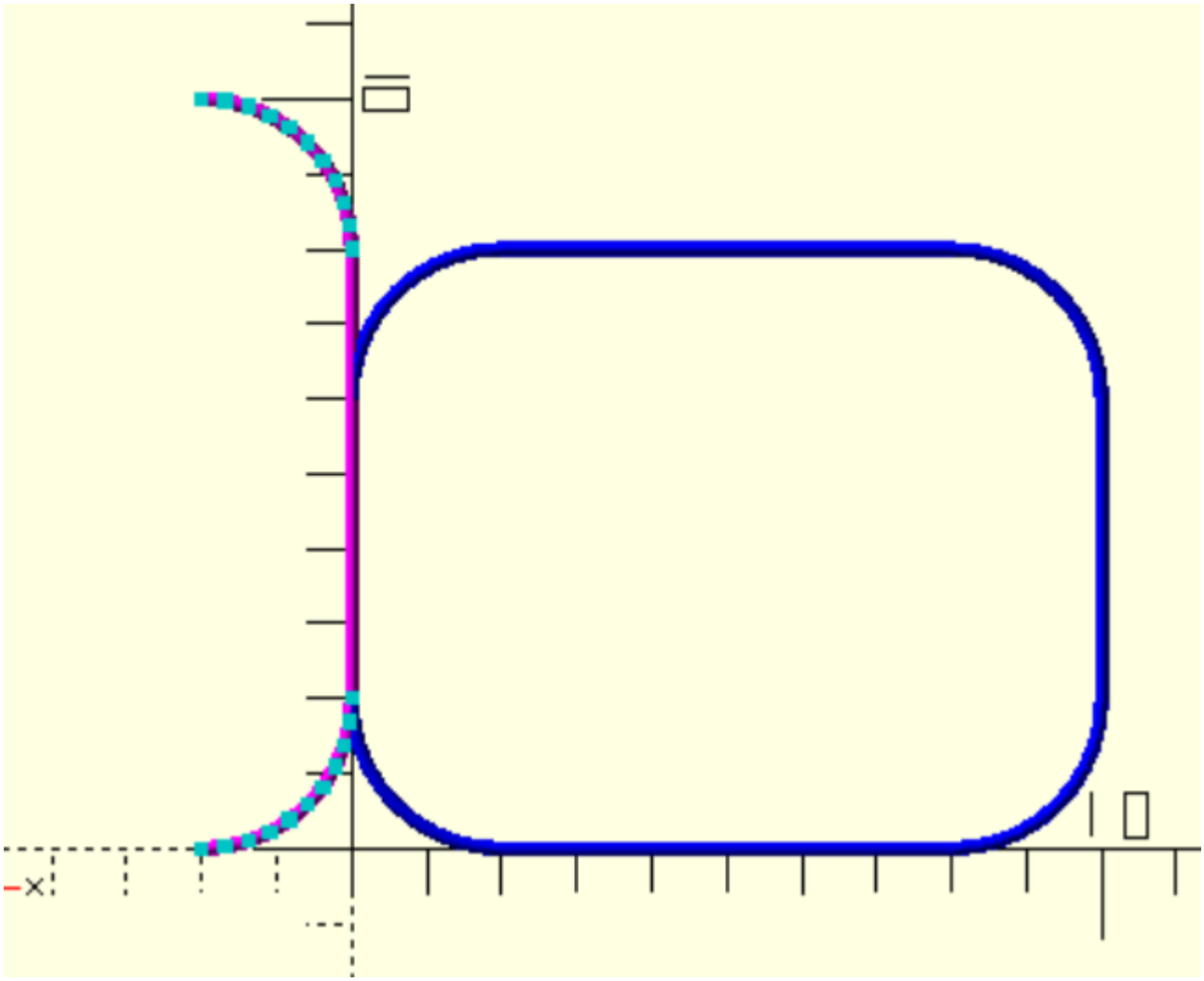
Process of creating a rounded cube or any other shape

In [9]:
```
# Sketch any 2d shape, here in below case it is rounded rectangle
# 10 x 8 with radius of 2 at each corner
s1=corner_radius(pts1([[0,0,2],[10,0,2],[0,8,2],[-10,0,2]]),10)

with open('trial.scad','w+') as f:
    f.write(f'''
    include<dependencies2.scad>
    color("blue")p_line3dc({s1},.2);

    ''')
```



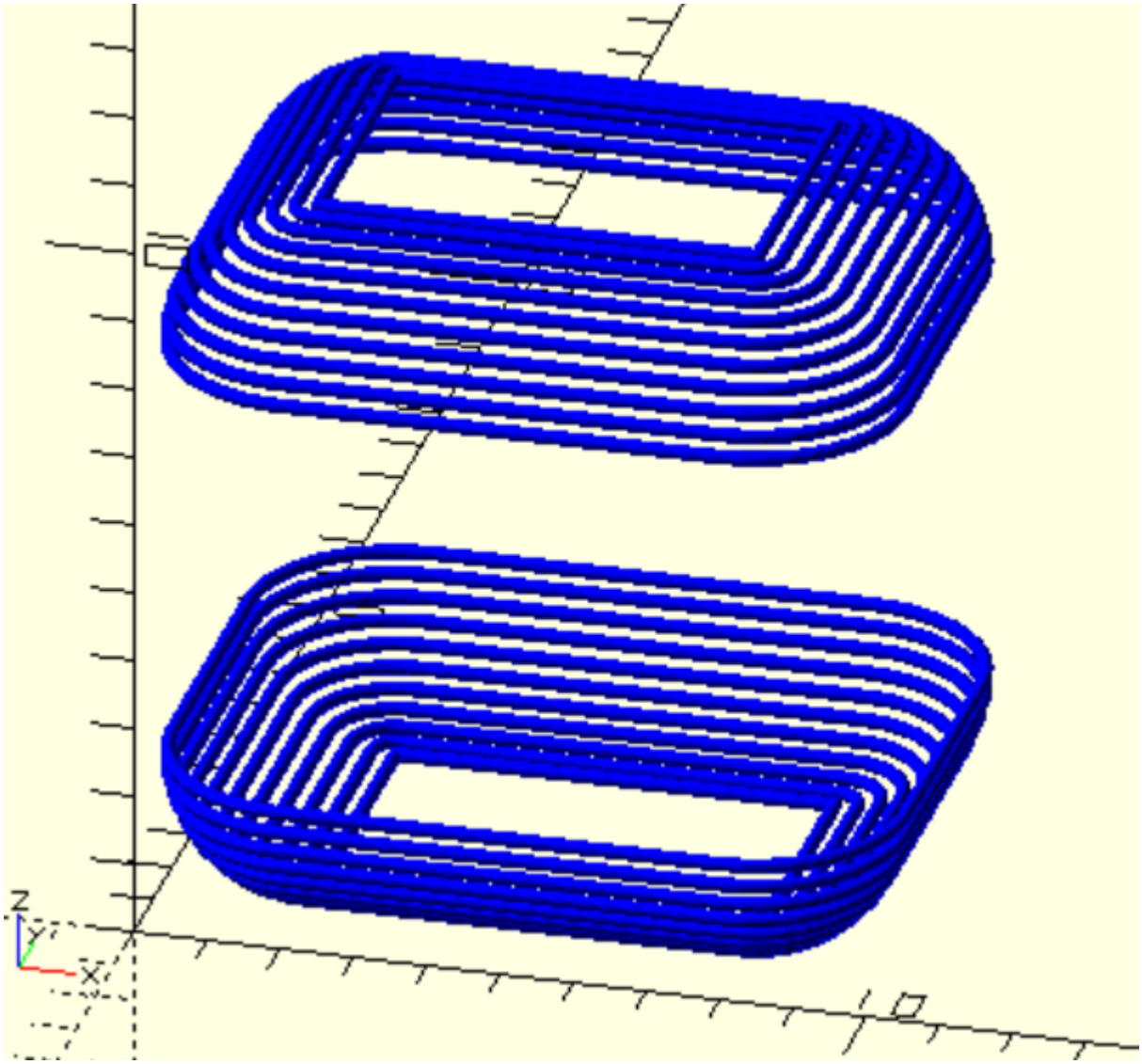
In [10]:
```
# Sketch any 2d shape, here in below case it is rounded rectangle 10 x 8 with radius of 2 at
s1=corner_radius(pts1([[0,0,2],[10,0,2],[0,8,2],[-10,0,2]]),10)

# now draw a path like following
p1=corner_radius(pts1([[-2,0],[2,0,2],[0,10,2],[-2,0]]),10)

# here in above path, x-axis represent amount of offset
# which needs to be given to the 2d shape and y-axis represent
# the height of that shape whic is offset

with open('trial.scad','w+') as f:
    f.write(f'''
    include<dependencies2.scad>
    color("blue")p_line3dc({s1},.2);
```

```
    color("magenta")p_line3d({p1},.2);
    color("cyan")points({p1},.2);


    ''')
```



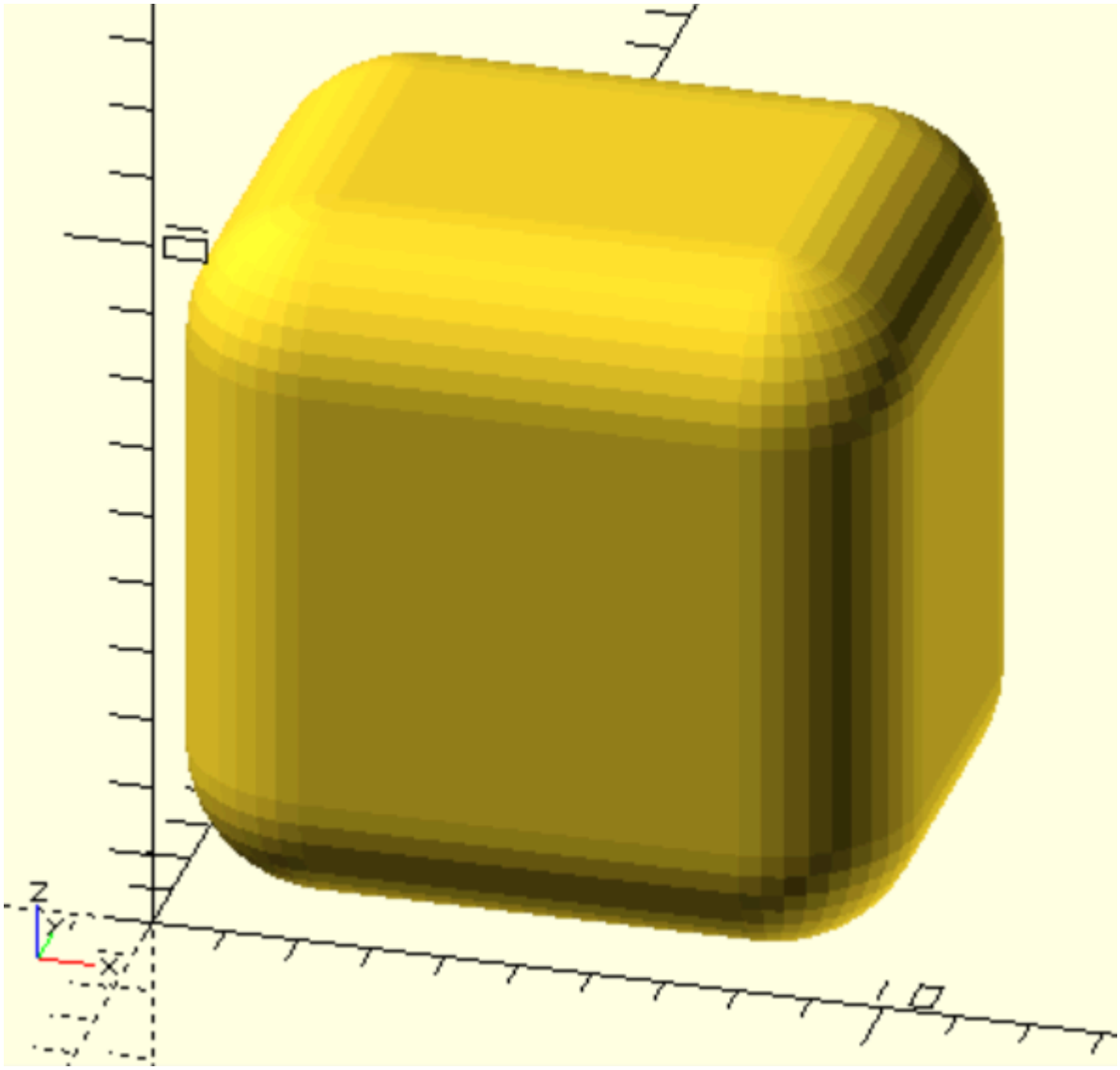
In [11]:

```
# use offset function and translate function to create a rounded cube
# or any other shape likewise
sol_1=[translate([0,0,y],offset(s1,x)) for (x,y) in p1]

with open('trial.scad','w+') as f:
    f.write(f'''
    include<dependencies2.scad>

    color("blue")for(p={sol_1})p_line3dc(p,.2);


    ''')
```



In [12]:

```
# finally render the solid

sol_1=[translate([0,0,y],offset(s1,x)) for (x,y) in p1]

with open('trial.scad','w+') as f:
    f.write(f'''
    include<dependencies2.scad>

    //color("blue")for(p={sol_1})p_line3dc(p,.2);
{swp(sol_1)}

    ''')
```


Z
Y
X