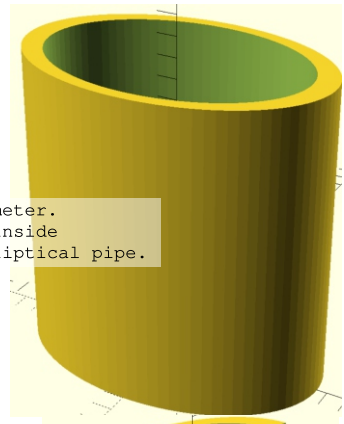
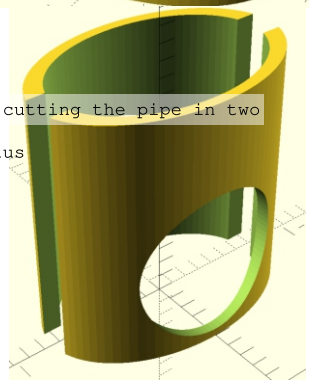
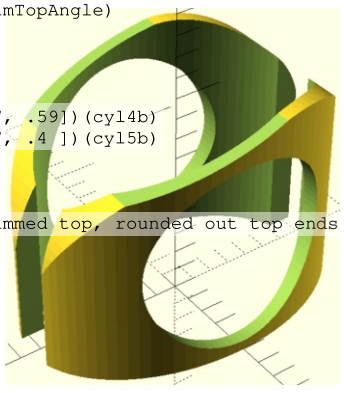
```
# SolidPython example code
# Brygg Ullmer, Clemson University
# Begun 2024-10-24
# Documented on 2024-12-04

from solid import *        # load in SolidPython/SCAD support code

#First, create the main body (an elliptical pipe)

cyl1a = cylinder(r=.5,  h=1.5); cyl1b = scale([1.5, 1,   1])(cyl1a) #cyl1b expresses the outer perimeter.
cyl2a = cylinder(r=.4,  h=2);   cyl2b = scale([1.6, 1.1, 1])(cyl2a) #cyl2b is for cutting away the inside
outGeom = cyl1b - cyl2b #output as exSolid04a.png. this performs the subtraction/cut, leaving an elliptical pipe.
```



```
#Next, cleave it in two halves, and perforate it horizontally

cube1a = cube();  cube1b = translate([ -.5, -.5, 0 ])(cube1a); cube1c = scale([2, .2, 2])(cube1b) #for cutting the pipe in two
cyl3a  = cylinder(r=.45, h=2); cyl3b = rotate(a = [90, 0, 0])(cyl3a)
cyl3c  = translate([0, .7, .5 ])(cyl3b) #cylinder for large hole perforating the two halves of the oculus

outGeom -= cube1c       #cleave the pipe into left and right halves
outGeom -= cyl3c        #output as exSolid04b.png. perforate the two halves of the oculus
```



```
#Then, trim the top at an angle, and round out the front and back (a "bevel/fillet," without the benefit of that function)

trimTopHeight = 1; trimTopAngle = 15 #15 degree angle to trimming of top of oculus
cube2a = cube(2); cube2b = translate([-1.4, -.8, trimTopHeight])(cube2a); cube2c = rotate(a = [0, trimTopAngle, 0])(cube2b)
outGeom -= cube2c       #trims the top of the two oculus halves (at height trimTopHeight, angle trimTopAngle)

cube3a = cube(1.5); cube3b = translate([-1.65,-.8, .7])(cube3a)
cube4a = cube(1.5); cube4b = translate([  .5, -.8, .5])(cube4a)

cyl4a  = cylinder(r=.5,  h=2); cyl4b = rotate(a = [90, 0, 0])(cyl4a); cyl4c  = translate([-.25,  .7, .59])(cyl4b)
cyl5a  = cylinder(r=.65, h=2); cyl5b = rotate(a = [90, 0, 0])(cyl5a); cyl5c  = translate([ .09,  .7, .4 ])(cyl5b)

# SolidPython doesn't presently provide a bevel/fillet, but we can approximate
roundOutTopBack  = cube3b - cyl4c; outGeom -= roundOutTopBack
roundOutTopFront = cube4b - cyl5c; outGeom -= roundOutTopFront #output as exSolid04d.png (angle-trimmed top, rounded out top ends
```



```
#Next, let's make some side perforations, that can be used toward (e.g.) illuminated capacitive touch sensing regions
#We'll first add them together, then  subtract them from the main body

offX, offY, offZ1, offZ2= .575, .7, .15, .225 #x, y, and z offsets, and z offset between
numVerticalPerforations = 3

cyl6a = cylinder(r=.1,  h=2); cyl6b = rotate(a = [90, 0, 0])(cyl6a)

cyl6c1 = translate([ offX, offY, offZ1])(cyl6b)
cyl7c1 = translate([-offX, offY, offZ1])(cyl6b)
sidePerforations = cyl6c1 + cyl7c1 # first perforations on left and right

for i in range(1, numVerticalPerforations):
  cyl6c2 = translate([0, 0, offZ2 * i])(cyl6c1)
  cyl7c2 = translate([0, 0, offZ2 * i])(cyl7c1)
  sidePerforations += cyl6c2 + cyl7c2

outGeom -= sidePerforations

radialSegments = 90; hdr = '$fn = %s;' % radialSegments # create a header, expressing the number of radial segments
scad_render_to_file(outGeom, 'exSolid04e.scad', file_header=hdr) # write the .scad file

### end ###
```