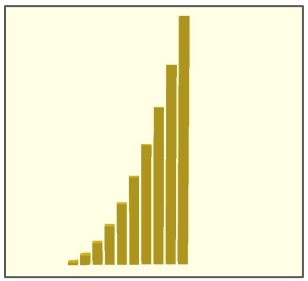
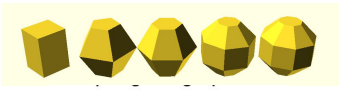


For Loops

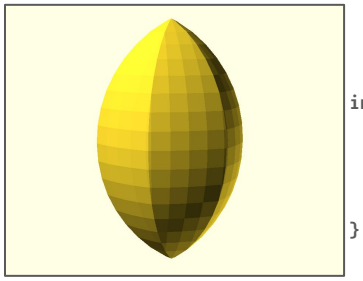
```
for( x = [0: 1: 10]){
    translate([x*5, 0, 0])
        cube([4, 4, x*x]);
}
```



Debugging For Loops

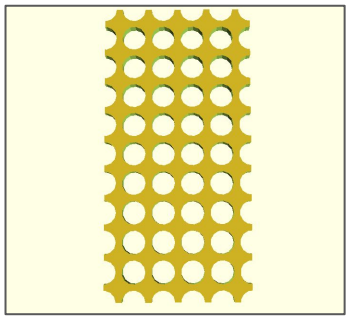
```
for( x = [1: 1: 5]){

    translate([x*8, 0, 0])
        sphere(4, $fn=3+x);

    echo("position:", x*8);
    echo("smoothness:", 3+x);
}
```

```
ECHO: "position:", 8
ECHO: "smoothness:", 4
ECHO: "position:", 16
ECHO: "smoothness:", 5
ECHO: "position:", 24
ECHO: "smoothness:", 6
ECHO: "position:", 32
ECHO: "smoothness:", 7
ECHO: "position:", 40
ECHO: "smoothness:", 8
```



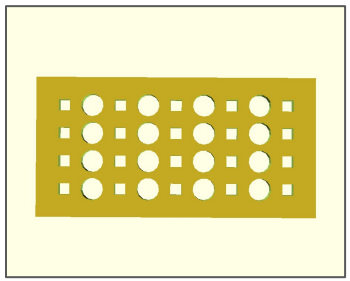
Intersection For Loops

```
intersection_for(n = [1 : 6]){
    rotate([0, 0, n * 60]){
        translate([6,0,0])
        sphere(r=12, $fn=50);
    }
}
```



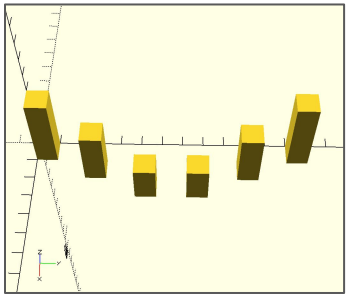
Nested For Loops

```
difference(){
  cube([50, 100, 5]);

  for( x = [0: 1: 5]){
    for( y = [0: 1: 10]){
      translate([x*10, y*10, 0])
        cylinder(h=12, r1=4, r2=4, center=true);
    }
  }
}
```



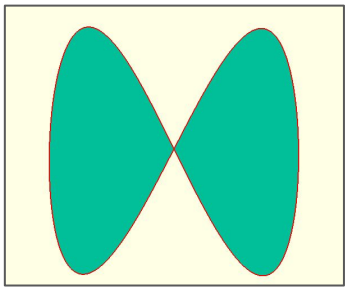
For Loops and if statements

```
difference(){
  cube([50, 100, 5]);

  for( x = [1: 1: 4]){
    for( y = [1: 1: 9]){
      if(y%2==0){
        translate([x*10, y*10, 0])
          cylinder(h=12, r1=4, r2=4, center=true);
      }else{
        translate([x*10, y*10, 0])
          cube([4,4,12], center=true);
      }
    }
  }
}
```



Looping over a Predefined List

```
for(i = [ [[ 0,  0,  0], 40],
          [[10, 24, 10], 30],
          [[20, 48, 20], 20],
          [[20, 72, 30], 20],
          [[10, 96, 40], 30],
          [[0, 120, 50], 40] ])
{
  translate([i[0][0], i[0][1], 0])
   cube([10, 10, i[1]]);
}
```



For Loop Polygons from Dynamically-Generated Lists of Points

```
radius = 10;

points = [for (phi = [0 : 1 : 720])
            [radius * cos(phi/2), radius * sin(phi)]
          ];

polygon(points);
```