

Es soll eine **Lochscheibe** mit variablen Anschraublöchern erstellt werden.

Alle Maßangaben sollen veränderbar bleiben, damit sie jederzeit geändert werden können.

Es wird mit der Scheibe an sich begonnen.

Die ersten variablen Maße sind:

```
hoch = 10; // Höhe der Scheibe
ad = 100; // Aussendurchmesser
id = 60; // Innendurchmesser
```

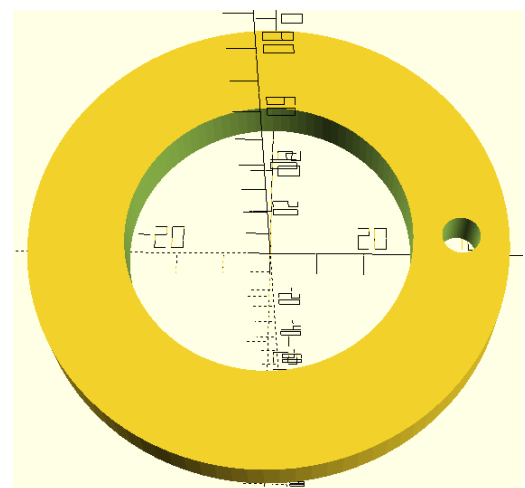
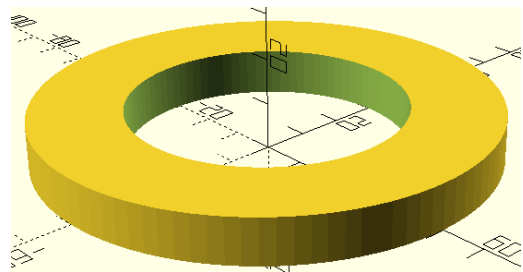
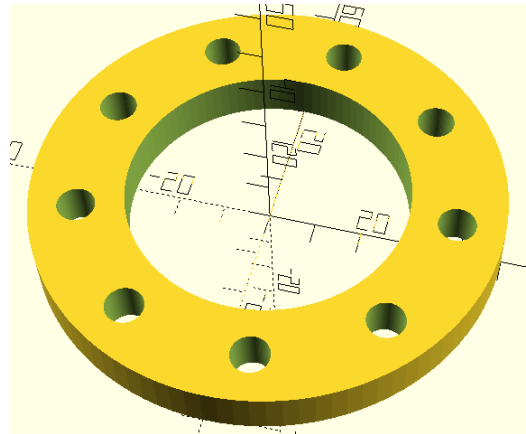
```
difference() {
  cylinder(h=hoch, d= ad, center=false);
  translate([0,0,-0.5])
  cylinder(h=hoch+1, d= id, center=false);
}
```

Der innere Zylinder wird immer um eins erhöht und -0.5 bei der Subtraktion nach unten versetzt. Damit steht der innere Zylinder immer unten wie oben um 0,5 mm über, was einen sauberen Schnitt hinterlässt.

```
hoch = 10; // Höhe der Scheibe
ad = 100; // Aussendurchmesser
id = 60; // Innendurchmesser
bd = 8; // Bohrungsdurchmesser
lkr = ((ad - id)/2); // Lochkreis-Radius
```

```
difference() {
  cylinder(h=hoch, d= ad, center=false);
  translate([0,0,-0.5])
  cylinder(h=hoch+1, d= id, center=false);

  rotate([0,0,0])
  translate([(id+lkr)/2,0,-0.5])
  cylinder(h=hoch+1, d= bd, center=false);
}
```



Das erste Loch sitzt!

V02.02

Neu kamen als Variable hinzu:

bd = 8; // Bohrungsdurchmesser

lkr = ((ad - id)/2); // Lochkreis-Radius

Die erste Hürde liegt in den Angaben „ad“ und „id“. Diese liegen in Durchmessermaßen vor. Um den Radius des Lochkreises zu errechnen, damit der Abstand vom Nullpunkt zum Bohrlochzentrum stimmt, muss dies berücksichtigt werden. Die Variable „lkr“ errechnet den kleinen Wert $((ad-id)/2)$ = im Beispiel: 20.

Bei „translate([(id+lkr)/2,0,-0.5])“ wird dieser Wert (20) zum Innendurchmesser addiert und anschließend halbiert. Denn damit wurde der Abstand vom Nullpunkt zum Bohrzentrum berechnet.

V02_Lochscheibe:

hoch = 10; // Höhe der Scheibe

ad = 100; // Aussendurchmesser

id = 60; // Innendurchmesser

bd = 8; // Bohrungsdurchmesser

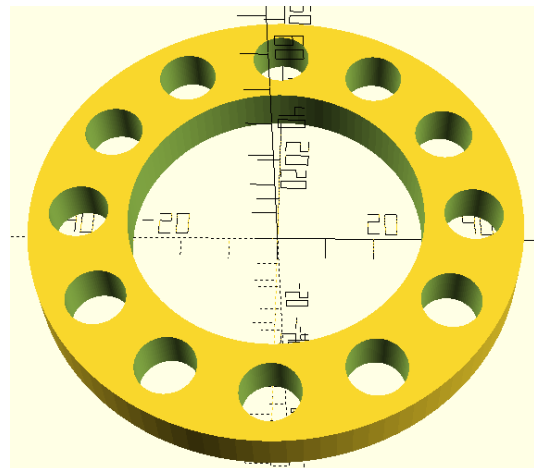
anz = 9; // Anzahl der Löcher

lkr = ((ad - id)/2); // Lochkreis-Radius

```
difference() {  
  cylinder(h=hoch, d= ad, center=false);
```

```
  translate([0,0,-0.5])  
  cylinder(h=hoch+1, d= id, center=false);
```

```
  for(z=[0:(360/anz):360]){  
    rotate([0,0,z])  
    translate([(id+lkr)/2,0,-0.5])  
    cylinder(h=hoch+1, d= bd, center=false);  
  }  
}
```



Über die Zeile „for(z=[0:(360/anz):360])“ wird mittels $(360/anz)$ der Teilungswinkel der Bohrlöcher berechnet. In die Lauf-Variable „z“ wird die Schleifenangabe übergeben. Mit jedem Durchlauf der Schleife erhöht sich der Wert von „z“, der dann in der Zeile „rotate([0,0,z])“ das nächste Bohrloch um den Teilungswinkel weiterdreht.

Nun kann über jede Variable jeder einzelne Wert geändert werden:

```
hoch = 10;    // Höhe der Scheibe
ad = 100;    // Aussendurchmesser
id = 60;     // Innendurchmesser
bd = 12;     // Bohrungsdurchmesser
anz = 12;    // Anzahl der Löcher
```

Das komplette Listing mit Anzeige der Werte in der Konsole:

```
// V02-Lochscheibe, variabel
```

```
$fn=100;
```

```
hoch = 10; // Höhe der Scheibe
ad = 100;  // Aussendurchmesser
id = 60;   // Innendurchmesser
bd = 12;   // Bohrungsdurchmesser
anz = 12;  // Anzahl der Löcher
lkr = ((ad - id)/2); // Lochkreisdurch-Radius
```

```
difference() {
  cylinder(h=hoch, d= ad, center=false);
  translate([0,0,-0.5])
  cylinder(h=hoch+1, d= id, center=false);

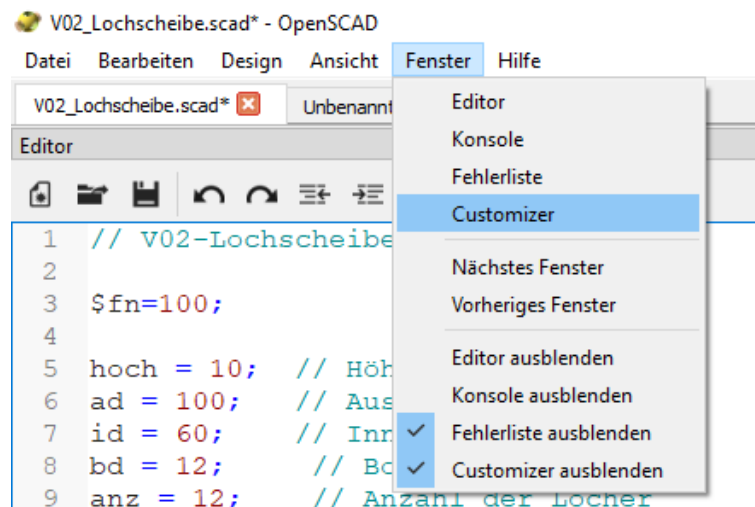
  for(z=[0:(360/anz):360]){

    rotate([0,0,z])

    translate([(id+lkr)/2,0,-0.5])
    cylinder(h=hoch+1, d= bd, center=false);
  }
}
echo ("-----")
echo (" Die Lochscheibe hat ", anz , " Löcher");
echo (" Die Einteilung ist ", 360/anz, " Grad");
echo (" Der Aussendurchmesser: ", ad);
echo (" Der Innendurchmesser : ", id);
echo (" Der Lochkreisdurchmesser : ", lkr+id);
echo (" Die Scheibenhöhe beträgt: ", hoch);
echo ("-----");
```

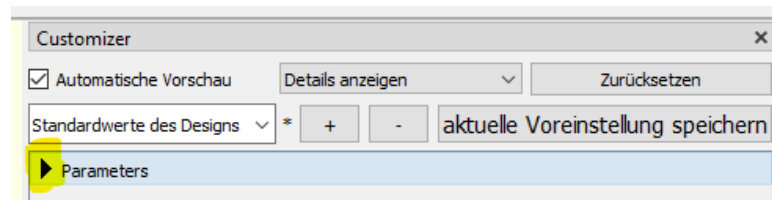
Jetzt kommt der CUSTOMIZER ins Spiel

Aufgerufen wird dieser aus der Haupt-Menüzeile über „Fenster“ und dann auf Customizer klicken.

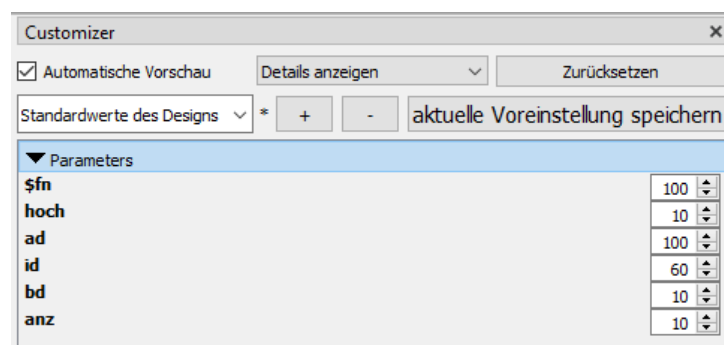


Im rechten Drittel der Monitoranzeige erscheint nun der **Customizer** mit folgendem Bild

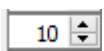
Hier muss man noch auf das kleine schwarze Dreieck vor „Parameters“ klicken!



Nun erscheinen alle Variablen aus dem Script zum Einstellen in der Anzeige.



Rechts davon werden die Werte mit Zahlen angezeigt.



Diese Werte können nun über die kleinen Dreiecke neben den Zahlen durch Klicken verändert werden.

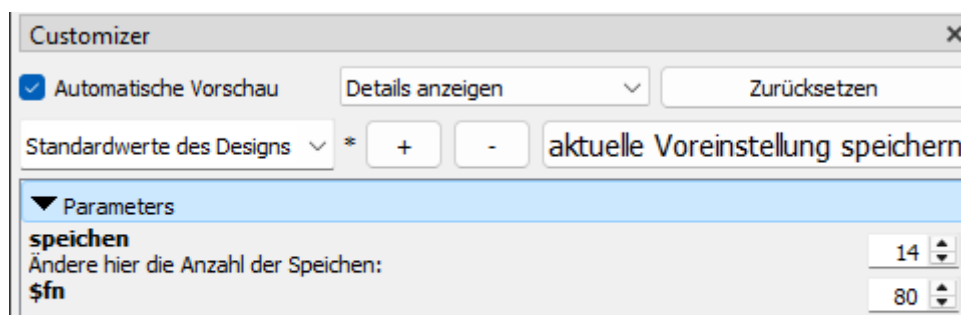
Etwas Zeit geben, und die Parameter aktualisieren die Vorschau! Ich drücke immer noch „F5“ – dann geht es schneller.

Überlegungen zum Customizer

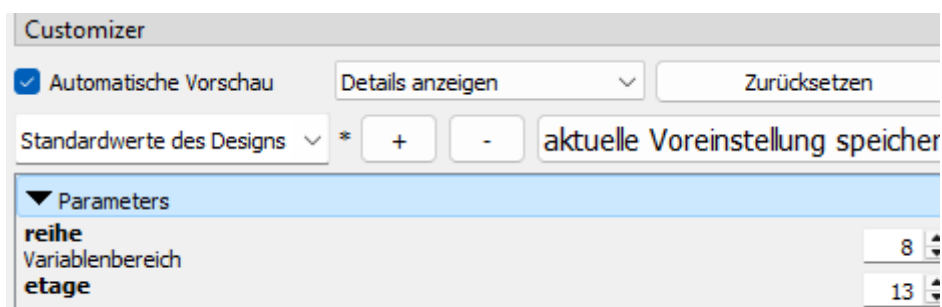
Auf den ersten Blick erscheint es als eine nette Spielerei. Geben Sie die Variablen in Bezug zueinander ein, wie im vorhergehenden Programm, so kann recht schnell eine neue Lochscheibe erstellt werden.

Legen wir diesen Gedanken zugrunde, werden aus einfachen Teilen flugs variable Elemente erzeugt. Es lohnt sich demnach, den Customizer gelegentlich einzusetzen.

Erinnern Sie sich an das Beispiel mit dem Speichenrad. Um die Anzahl der Speichen zu ändern, tippten wir jedes Mal in den Skript neue Zahlen ein. Im Customizer geht das mit Mausklick vonstatten.



Gehen Sie weiter zurück – bis zum Hochhaus:

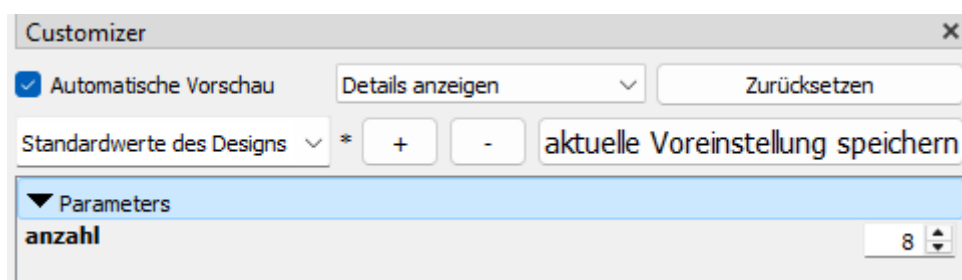


Sogar beim Klemmbaustein – nach einer kleinen Änderung – ist es nun möglich per Maus im Customizer die Vorgaben zu ändern:

Listing von: V02-02-baustein.scad

```
/*  
Klemmbaustein V2  
*/  
  
anzahl=(6);  
// Aufruf mit Zahlenübergabe:  
BAUstein(anzahl);  
  
//Modulbereich  
module BAUstein(noppen_pro_reihe=4) {  
$fn=30;  
laenge = noppen_pro_reihe * 8;  
cube([laenge, 16, 9.6]);  
  for (x_position=[4 : 8 : laenge-4]) {  
    translate([x_position, 4, 1.7]) cylinder(h=9.6, d=4.8);  
    translate([x_position, 12, 1.7]) cylinder(h=9.6, d=4.8);  
  }  
} // module
```

Ich habe lediglich die Variable „anzahl“ eingefügt, deren Wert an den Modul-Aufruf „BAUstein“ übergeben wird.



Nicht vergessen „F5“ zu drücken.