

linear_extrude (height = ?, center = true, convexity = 10, twist = ?, slices = ?, scale = ?, \$fn = ?) {...}

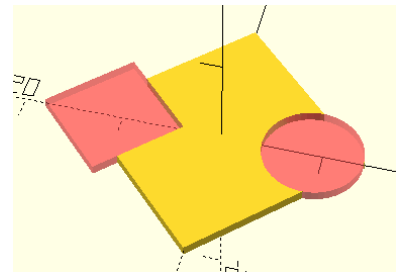
Vertikales Extrudieren von Formen

Extrudieren ist wie ein Vervielfältigen oder Kopieren in die Höhe über die Z-Achse, während die gesamte Form ohne Verluste in der Höhe gestreckt und aufgefüllt wird. Da in 2D die Formen sowieso nur „1“ hoch sind, lässt sich damit hervorragend arbeiten.

Als Beispiel wird ein Quadrat im Nullpunkt gesetzt und um 45° gedreht. Davon wird ein kleineres Quadrat auf der einen Seite, sowie ein Kreis auf der anderen Seite abgezogen.

```
$fn=50;

difference(){
  translate([-14.2,0]) rotate([0,0,-45]) square(20,20);
  #translate([-18,0]) rotate([0,0,-45]) square(10,10);
  #translate([9.5,0]) circle(5);
}
```

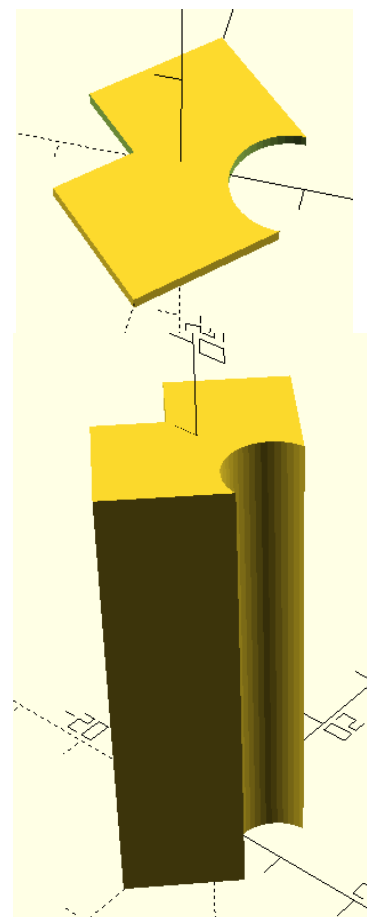


Die abzuziehenden Teile sind noch mit einer vorgestellten Raute (#) versehen, damit diese im Bild noch erkennbar sind. Werden die Rauten entfernt, so kommt das eigentliche gewünschte Objekt nun zum Vorschein:

//007-extrude:

```
$fn=50;
linear_extrude(50)
difference(){
  translate([-14.2,0]) rotate([0,0,-45]) square(20,20);
  translate([-18,0]) rotate([0,0,-45]) square(10,10);
  translate([9.5,0]) circle(5);
}
```

Wird noch ein „linear_extrude (Höhenangabe)“ eingefügt, entsteht das extrudierte Modell.

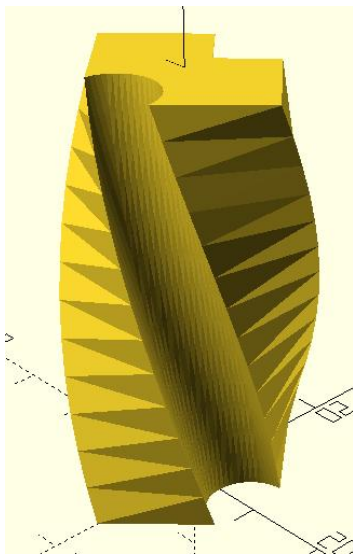


Der Befehl `linear_extrude` nimmt einen einzigen Parameter, die zu erstellende Höhe der 3D-Form, gefolgt von der 2D-Form, die in 3D gedehnt wird.

Für komplexere 3D-Formen können auch für `linear_extrude` Parameter von **twist**, **slices** und **scale**, optional bereitgestellt werden.

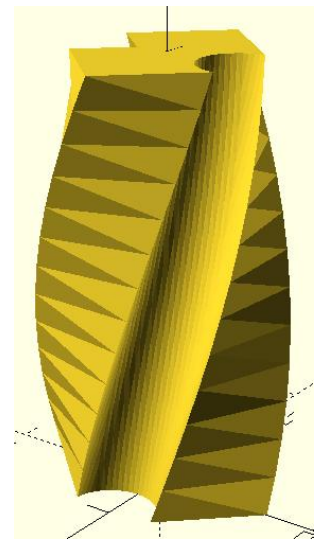
„**twist**“: Der twist-Parameter gibt einen Winkel in Grad an, um den die 2D-Form beim Extrudieren gedreht werden soll. Der Drehpunkt ist hier der Startpunkt des Elements, also die Z-Achse.

twist=90 ergibt eine Drehung im Uhrzeigersinn, -90 dreht es gegen den Uhrzeigersinn



Links:
`linear_extrude(50, twist = 90)`

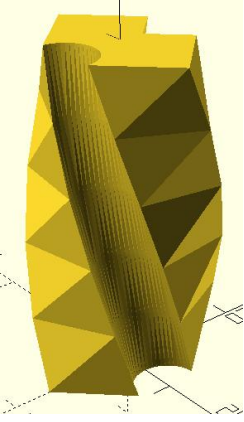
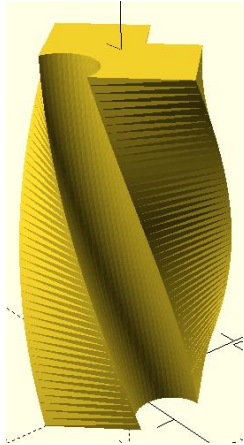
Rechts:
`linear_extrude(50, twist = -90)`



„**slices**“:

Der slices-Parameter steuert wie viele Segmente aufeinander verwendet werden sollen, um die Drehung zu erstellen.

Die vorgegebene Höhe (hier mit 50 angegeben) wird durch die Anzahl der slices (Schichten) geteilt. Je mehr Schichten es sind, desto glatter wird die Außenform.

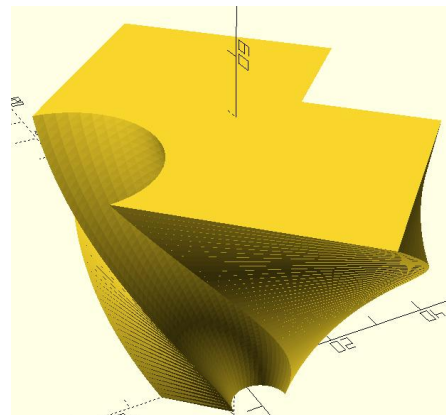
	<pre>//007-extrude linear_extrude(50, twist=90, slices = 5)</pre> <p>Im rechten Bild wurde , slices = 50) eingegeben.</p> <p>Je mehr Zwischenstufen (slices), desto feiner wird die Oberfläche.</p> <p>Fehlt der Wert von „slices“, wird ein relativ grober Wert eingesetzt.</p>	
---	--	---

„scale“:

Mit „scale“ vergrößert oder verkleinert die 2D-Form um diesen Wert über die Höhe der Extrusion. Der Parameter kann eine Zahl oder ein Vektor sein:

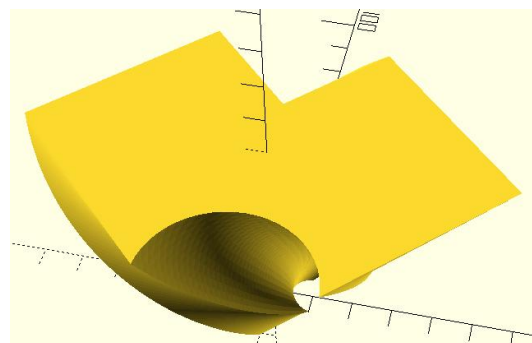
```
linear_extrude(50, twist=90, slices=50, scale=3)
```

Die einzelne Vorgabe von „scale=3“ lässt die untere Ausgangsform um den Faktor 3 vergrößern.



Jedoch mit einer Vektorangabe von z.B. „scale=[4, 1.5]“ kann es sogar unregelmäßig verzerrt werden.

Hierbei ist es wichtig, dass die zwei Zahlen in eckigen Klammern [] stehen!



Hier wird der Wert X mit 4 und der Wert Y mit 1,5 multipliziert.

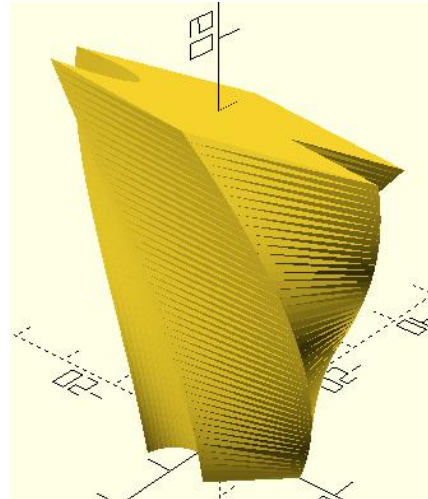
Dies bezieht sich immer auf die letzte Schicht (slice). Zwischendrin wird es gedreht und geändert.

007-04

Verkleinern mit „scale“

Es kann ebenso verkleinert werden: der Schluss der Extrusion wird auf 0,5 vom eigentlichen Maß gesetzt:

mit „**scale=[0.5, 3]**“ als Vektorangabe:
in X-Richtung auf 0,5 verkleinert und
in Y-Richtung 3-fach vergrößert.



Dergleichen kann eine Spirale erstellt werden. Das Muster in 2D wird mit „twist=720“ - dies entsprechen zwei Umdrehungen - dargestellt. Die Höhe beträgt 20.

Mit „slices=300“ werden 300 Schichten auf die gesamte Höhe berechnet.

Zur besseren Darstellung wurde noch beim Kreisbefehl „\$fn=100“ als annehmbare Auflösung eingefügt.

```
//007-extrude
```

```
$fn=100;
```

```
linear_extrude(20, twist=720, slices=300)
```

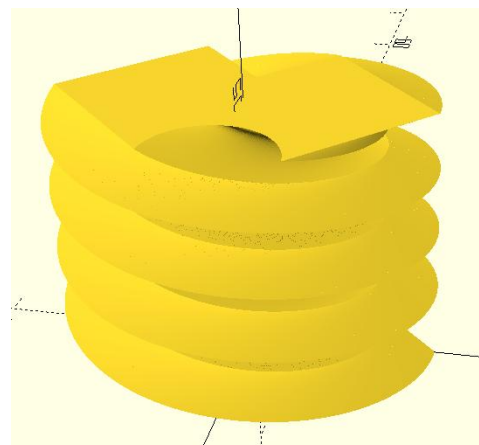
```
difference(){
```

```
translate([-14.2,0])rotate([0,0,-45])square(20,20);
```

```
translate([-18,0])rotate([0,0,-45])square(10,10);
```

```
translate([9.5,0])circle(5);
```

```
}
```

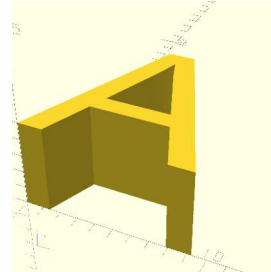


Auch wenn es so aussehen mag, als Herstellung für ein Gewinde ist es völlig ungeeignet.

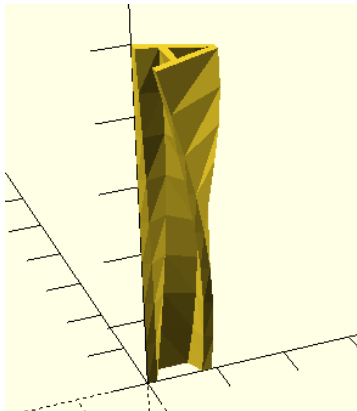
„linear_extrude“ lässt sich hervorragend auf Text übertragen:

```
linear_extrude(5) text("A");
```

Hier wird die Höhe mit „5“ angegeben



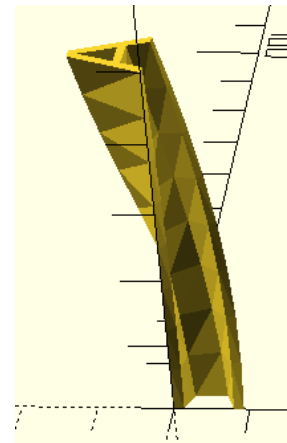
007-extrude_t2.scad



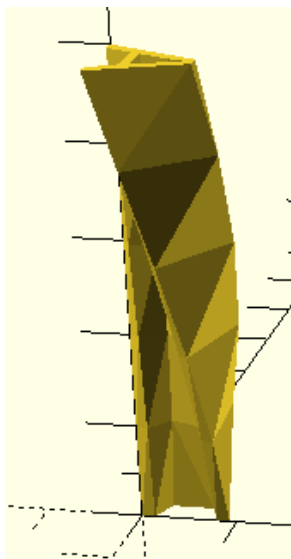
„twist“:

twist=90 ergibt eine Drehung
im Uhrzeigersinn

bei -90 gegen den Uhrzeigersinn.



007-extrude_t3.scad

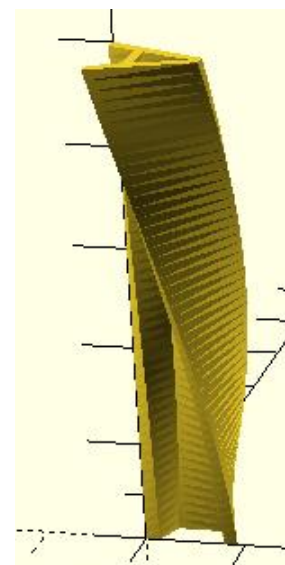


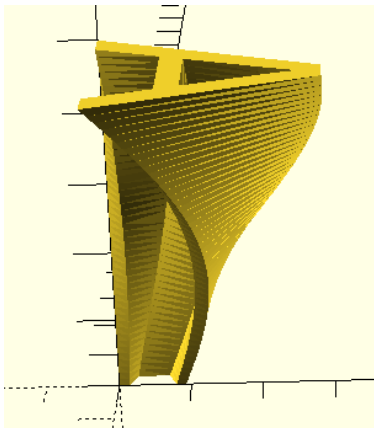
„slices“:

Der slices-Parameter steuert wie
glatt eine Drehung sein soll –
insbesondere, wie viele Segmente
aufeinander verwendet werden,
um die Drehung zu erstellen.

Links: slices=5

Rechts: slices =50



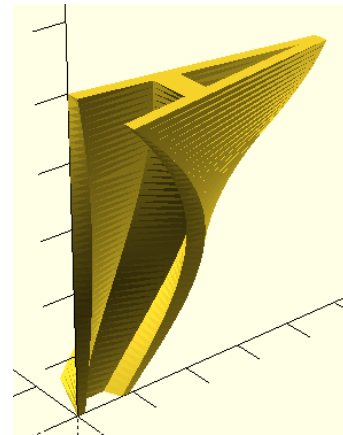


„scale“:

„scale“ vergrößert oder verkleinert die 2D-Form um diesen Wert über die Höhe der Extrusion.

Links: scale=3

Rechts: scale=[4, 1.5]



007-extrude_t4.scad

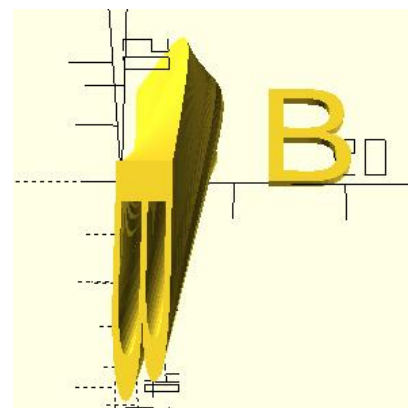
Ebenso kann verkleinert werden: der Schluss der Extrusion wird auf 0,5 vom eigentlichen Wert gesetzt:

extrude

```
translate([12,0,0])text("B");
linear_extrude(10, twist=90, slices=50, scale=[0.5,3]) text("B");
```

Über “ translate([12,0,0])text("B");“ wurde als Muster ein „B“ nach rechts verschoben.

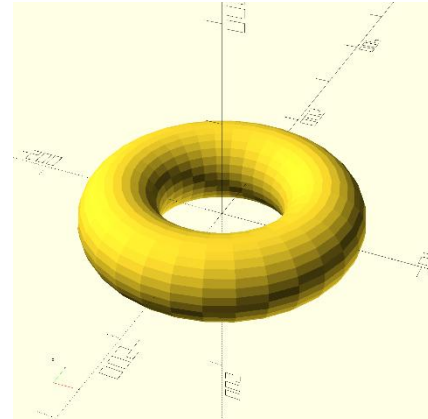
Danach folgt „linear_extrude“ mit scale=[0.5, 3] als Vektorangabe: in X-Richtung auf 0,5 verkleinert und in Y-Richtung 3-fach vergrößert.



rotate_extrude (angle = ?, convexity=10) { }

Extrudieren von Formen entlang eines Kreises

Anstatt eine 2D-Form entlang eines linearen Pfads zu extrudieren, verwendet der Befehl „rotate_extrude“ zum Bewegen der 2D-Form einen kreisförmigen Pfad, wodurch eine Donut-ähnliche Form namens Torus entsteht.



007-rextrude

```
rotate_extrude() {
  translate([100, 0]) circle(40);
}
```

Der Befehl „rotate_extrude“ ist ein zweistufiger Prozess, bei dem zuerst die 2D-Form um 90 Grad um die X-Achse gedreht und damit aufgestellt wird. Daher ist der Startpunkt zum Zeichnen immer die X-Achse.

Als Zweites wird diese Form in einem Kreis um die Z-Achse gedreht.

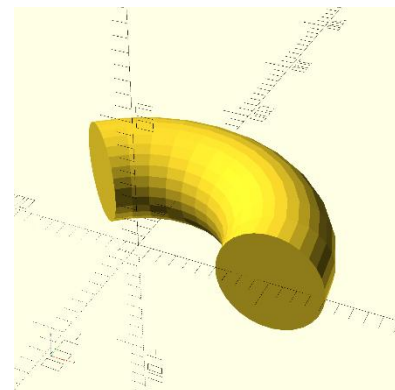
Bei der Verwendung von „rotate_extrude“ ist darauf zu achten, dass die Form sich nicht in sich selbst dreht, da sonst nur eine Fehlermeldung erscheinen wird. Im Beispiel wird zuerst über „translate“ ein Abstand zum Nullpunkt erzeugt, der größer als der Durchmesser des Kreises ist. Außerdem berührt dadurch die 2D-Form nicht die Z-Achse.

Der Befehl „rotate_extrude“ nimmt auch einen optionalen Winkelparameter an, der es ermöglicht den Rotationswinkel festzulegen.

Bild 3.17 zeigt einen Kreis der entlang einer 135-Grad-Drehung um die z-Achse extrudiert wurde.

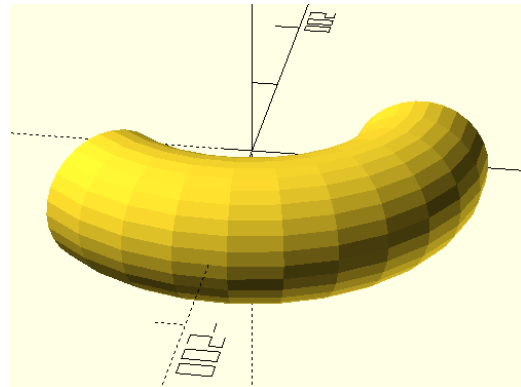
007-rextrude

```
rotate_extrude(angle=135) {
  translate([100, 0]) circle(40);
}
```



007-08

Als erster Wert wird bei „rotate_extrude“ der Rotationswinkel von 135 (in Grad) angegeben. Wird dieser Wert mit einem Minuszeichen versehen, schwenkt die Form in die entgegengesetzte Richtung. Der Start ist immer die X-Achse.



Hier wird mittels translate der erste Parameter (100) eingetragen. Dies ist der Durchmesser des folgenden Kreises: vom Nullpunkt des Koordinatensystems zum Mittelpunkt des Kreises.

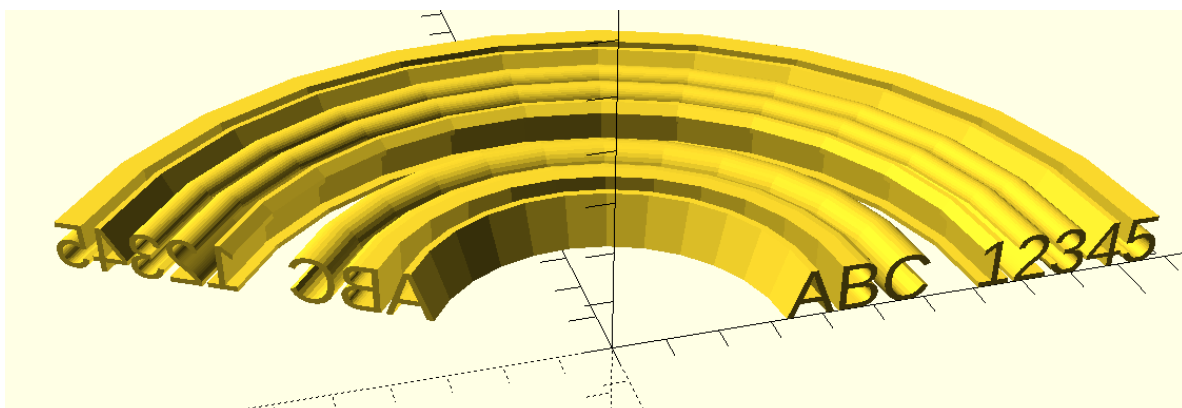
„angle“ mit Minuswert versehen ergibt das obere Bild.

```
rotate_extrude(angle= - 135) {  
  translate([100, 0]) circle(40);  
}
```

Selbstverständlich funktioniert dies auch mit einem Text

007-rextrude

```
rotate_extrude(angle=150){  
  translate([33,0]) text("ABC");  
}  
rotate_extrude(angle=150){  
  translate([70,0]) text("12345");  
}
```



offset

Vergrößern und Verkleinern einer Form

Es soll eine Backform als Teig-Ausstecher in Kreuzform erstellt werden.

Diese Kreuzform kann einfach mit zwei Rechtecken, die über „union“ verbunden sind, erstellt werden.

Ferner kann dieses Kreuz mit „linear_extrude“ in 3D verwandelt werden. Jetzt müsste ein kleineres Kreuz erzeugt und vom größeren subtrahiert werden, um eine gewisse Wandstärke zu erzielen.

Einfacher geht es mit „offset“. Hier wird nur das gleiche Element mit „offset“ und einem Wert angegeben. Ein positiver Wert vergrößert und ein Minuswert verkleinert folglich das Element.

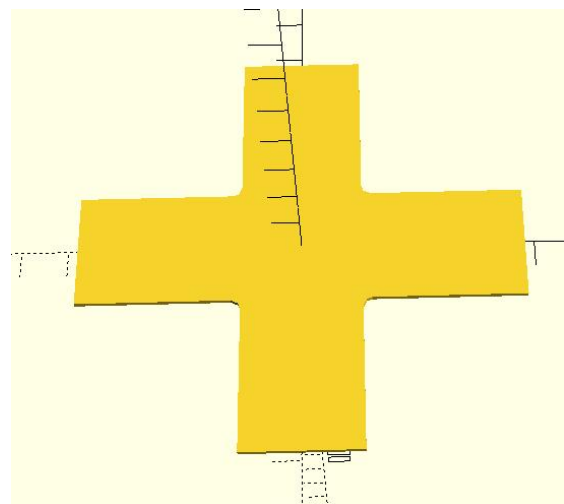
Somit kann mit dem „offset“-Befehl das gleiche Element mit den gleichen Werten voneinander subtrahiert werden – es wird nur die Wandstärke angegeben!

Es wird im folgenden Script ein negativer Wert eingetragen, um die 2D-Form zu verkleinern:

```
offset(-2) {
  union() {
    square([100, 30], center=true);
    square([30, 100], center=true);
  }
}
```

Kaum sichtbar, aber kleiner als die angegebene Form selbst!

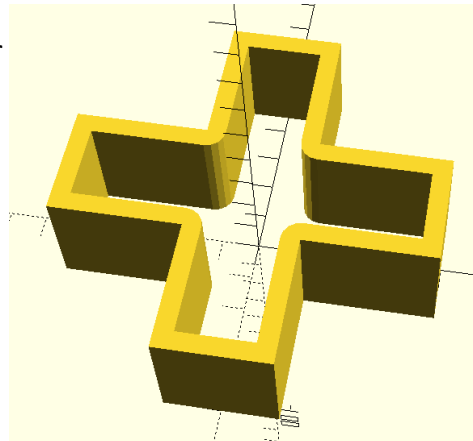
Der Code der Form der versetzt werden soll wird in geschweifte Klammern des Befehles „offset()“ gesetzt. In den runden Klammern wird der zu verändernde Betrag geschrieben. Ein positiver Wert vergrößert eine Form und ein negativer Wert verkleinert diese.



007-10

HINWEIS: Beim Schrumpfen einer Form werden die inneren Ecken abgerundet, aber beim Wachsen der Form werden die äußeren Ecken abgerundet.

Jetzt kann man diesen Code wiederverwenden, um die Wände des Kreuzes der Ausstechform zu erstellen:



```
linear_extrude(30) {  
  difference() {  
    union() {  
      square([100, 30], center=true);  
      square([30, 100], center=true);  
    }  
    offset(-5) {  
      square([100, 30], center=true);  
      square([30, 100], center=true);  
    }  
  }  
}
```

Der Wert bei offset(-5) ergibt die Wandstärke der Form. Die 5 werden vom äußeren Maß abgezogen.

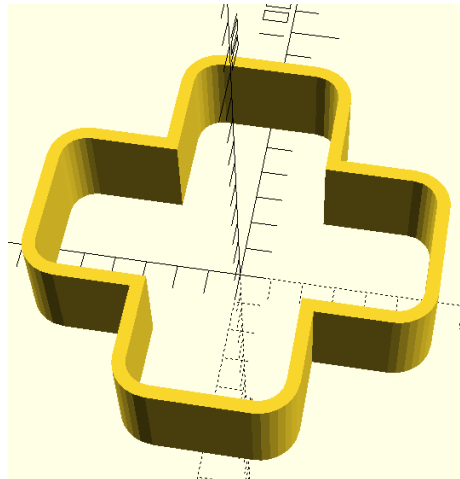
linear_extrude(30) {	// = Höhe der Extrusion
difference() {	// = Subtrahieren
union() {	// = Verbinden von Formen
square([100, 30], center=true);	// = Erstellung vom Kreuz
square([30, 100], center=true);	// = Erstellung vom Kreuz
}	// = Ende von Verbinden
offset(-2) {	// = Aufruf offset mit Wandstärke von 2
square([100, 30], center=true);	// = Erstellung vom Kreuz
square([30, 100], center=true);	// = Erstellung vom Kreuz
}	// = Ende von offset
}	// = Ende von difference
}	// = Ende von linear_extrude

Nutzung des Parameters „r“ von offset:

Ist der Wert „r“ positiv werden die äußeren Ecken mit den angegebenen Werten als Radien versehen.

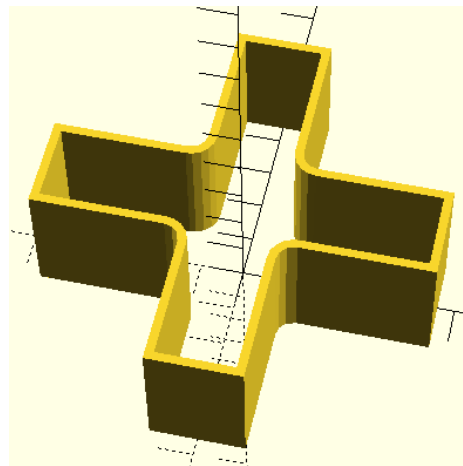
Wenn der Wert negativ ist, wird das Polygon nach innen versetzt. Hier werden dann die inneren Ecken mit den Vorgaben als Radien versehen. 007-offset

```
linear_extrude(30) {
  difference() {
    offset (r=12){ union() {
      square([100, 30], center=true);
      square([30, 100], center=true);
    }
    offset(r=8) {
      square([100, 30], center=true);
      square([30, 100], center=true);
    }
  }
}
```



Ist der äußere Wert (r=12) und der innere (r=8) positiv, so entstehen an den äußeren Ecken die Radien mit den angegebenen Maßen. 007-offset

```
linear_extrude(30) {
  difference() {
    offset (r = -4){ union() {
      square([100, 30], center=true);
      square([30, 100], center=true);
    }
    offset(r = -6 ) {
      square([100, 30], center=true);
      square([30, 100], center=true);
    }
  }
}
```



Werden vom ersten Wert (-4) weitere negative Offset-Angaben (-6) abgezogen, werden in den Inneren Ecken mit den Vorgabe-Werten die Radien dargestellt. Bei den negativen Werten ist zu beachten, dass der erste Minus-Wert (-4) größer ist als der zu subtrahierende (-6).

007-12



Auch für ausgefallene Schriften ist **offset** geeignet:

Anstatt der ebenen square, wie im vorangegangenen Beispiel, werden jetzt Buchstaben verwendet.

007-offset:

```
linear_extrude(15) {  
  difference() {  
    text("ABC",size=20);  
    offset(-.2) {  
      text("ABC",size=20);  
    }  
  }  
}
```

Der Wert für offset ist hier relativ klein zu halten, da sonst die Wandstärken sehr schnell breiter wie der Buchstabe werden können.

Im Beispiel oben liegt der Wert bei -0,2. Muss angegeben werden mit **-0.2** !



offset(-0.2)

offset(-0.5)

offset(-1)