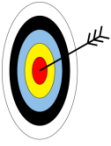
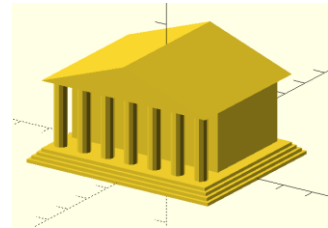


## Der Tempel – Koordinatoren



Hier wird euch erklärt, wie ihr die Grundlage für euer ein Tempelgebäude erstellen könnt. Dafür benötigt ihr:

- × `cylinder()`, `cube()`, `sphere()`
- × for-Schleifen
- × `translate()`, `rotate()`, `resize()`



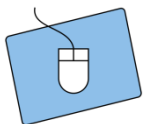
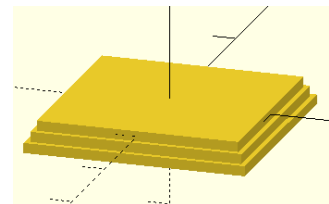
### Der Sockel

Der Sockel soll aus mehreren Stufen bestehen. Dazu benötigt man lediglich einige Quader, die man stapelt. Dabei hilft euch eine sog. for-Schleife:

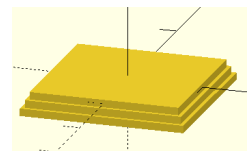
```
for ( variable = [anfang : ende] ) {
  <eure Anweisung>
}
```

Eure Anweisung wird dann für jede Zahl vom `anfang` bis zum `ende` einmal wiederholt. Besser noch, Ihr könnt die `variable` benutzen in den Wiederholungen benutzen. Zum Beispiel, um Treppenstufen zu erstellen:

```
for ( i = [0 : 2] ) {
  translate([0,0,-i]){
    cube( [20+i,20+i,1] , center=true);
  }
}
```



Erstellt ein paar Stufen auf denen euer Tempel stehen soll.



Wenn ihr Variablen benutzt, könnt ihr schneller die Ausmaße eures Tempels verändern:

```
breite = 20;
laenge = 20;
stufenhoehe = 1;
anzahl=3;

for (i = [0:anzahl-1]) {
  translate([0,0,stufenhoehe*i]){
    cube([breite-i,laenge-i,stufenhoehe],center = true);
  }
}
```

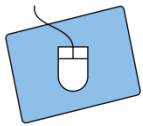
## Der Tempel – Koordinatoren

### Die Säulen

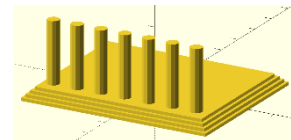
Am Eingang stehen üblicherweise Säulenreihen. Auch dazu könnt ihr eine `for`-Schleife, `translate()` und `cylinder()` benutzen.

Das funktioniert so ähnlich wie die Sockel:

```
Für i gleich 0 bis zu der Anzahl der Säulen {
  Verschiebe um Abstand mal i {
    Zylinder mit Radius r und Höhe h;
  }
}
```

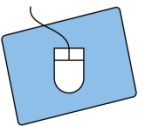
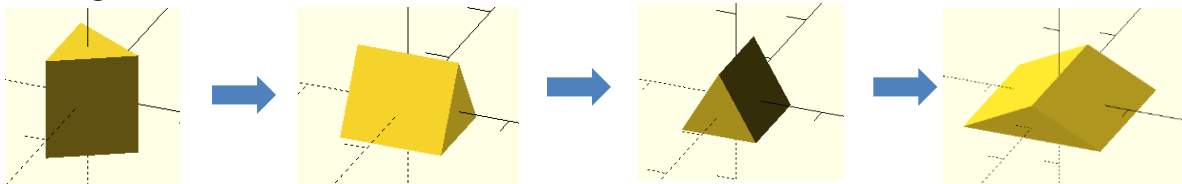


Erstellt eine Säulenreihe an der Front eures Tempels.  
Benutzt dafür eine `for`-Schleife.

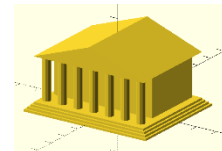


### Die Wände und das Dach

Der Raum selbst ist einfach nur ein großer `cube()`. Um ein Dreiecks- oder Pyramidendach zu erstellen könnt ihr einen `cylinder()` benutzen. Mit einer Feinheit `$fn = 3` hat er genau 3 Seiten. Wenn ihr ihn dann mit `rotate()` dreht und mit `resize()` auf die richtige Größe stutzt, dann ist euer Dach fertig.



- Setzt einen `cube()` auf hinter die Säulen, als Tempelraum.
- Benutzt, `cylinder()`, `rotate()` und `resize()`, um ein Dach zu erstellen.



Sehr gut. Das war es auch schon. Jetzt steht das Grundgerüst eures Tempels. Nun könnt ihr fertige Bauteile der Designer einsammeln und in den Tempel einbauen. Wie das funktioniert, erfahrt ihr auf der nächsten Seite.

## Der Tempel – Koordinatoren

### Module der Designer einbinden

Mit „`module name()`“ können die Designer einen eigenen Befehl erstellen. Immer wenn ihr irgendwo in eurem Programm den Namen des Befehls verwendet, wird alles in den geschweiften Klammern eures Befehls ausgeführt. So könnt ihr leicht eigene Objekte vervielfältigen.

Ihr könnt Module aus anderen Programmen oder Dateien benutzen, indem ihr `use<datei.scad>` benutzt. Dafür müssen alle Dateien im gleichen Ordner sein und ihr müsst ihre Namen kennen.

Mit den Befehlen `resize()` ihr das Modul auf die richtige Größe bringen und mit `rotate()` und `translate()` neu ausrichten.

```
for (x = [0:6]) {
  translate([-24 + 8*x,-18,2]){
    cylinder(r=2,h=20,$fn=feinheit);
  }
}
```

Vorher:

```
use<Saeule.scad>

// Säulen

//alt
for (x = [0:6]) {
  translate([-24 + 8*x,-18,2]){
    //neu
    rotate([0,0,90]) {
      resize([4,4,22]){
        saeule();
      }
    }
  }
}
```

Nachher:



Die Dateien müssen alle im gleichen Ordner sein. Der Name muss mit Endung und Groß- und Kleinschreibung richtig geschrieben werden.



- Tauscht euch mit euren Designern aus und bindet ihre Objekte ein.
- Besprecht mit Designern, die keine Aufgabe mehr haben, wie man den Tempel erweitern könnte.

