

Syntax

```
var = value;  
module name(...) { ... }  
name();  
function name(...) = ...  
name();  
include <...scad>  
use <...scad>
```

2D

```
circle(radius | d=diameter)  
square(size,center)  
square([width,height],center)  
polygon([points])  
polygon([points],[paths])  
text(t, size, font,  
      halign, valign, spacing,  
      direction, language, script)
```

3D

```
sphere(radius | d=diameter)  
cube(size, center)  
cube([width,depth,height], center)  
cylinder(h,r|d,center)  
cylinder(h,r1|d1,r2|d2,center)  
polyhedron(points, triangles, convexity)
```

Transformations

```
translate([x,y,z])  
rotate([x,y,z])  
scale([x,y,z])  
resize([x,y,z],auto)  
mirror([x,y,z])  
multmatrix(m)  
color("colorname")  
color([r,g,b,a])  
offset(r|delta,chamfer)  
hull()  
minkowski()
```

Boolean operations

```
union()  
difference()  
intersection()
```

Modifier Characters

*	disable
!	show only
#	highlight
%	transparent

Mathematical

```
abs  
sign  
sin  
cos  
tan  
acos  
asin  
atan  
atan2  
floor  
round  
ceil  
ln  
len  
let  
log  
pow  
sqrt  
exp  
rands  
min  
max
```

Functions

```
concat  
lookup  
str  
chr  
search  
version  
version_num  
norm  
cross  
parent_module(idx)
```

Other

```
echo(...)  
for (i = [start:end]) { ... }  
for (i = [start:step:end]) { ... }  
for (i = [...,,,...]) { ... }  
intersection for(i = [start:end]) { ... }  
intersection for(i = [start:step:end]) { ... }  
intersection for(i = [...,,,...]) { ... }  
if (...) { ... }  
assign (...) { ... }  
import("...stl")  
linear_extrude(height,center,convexity,twist,slices)  
rotate_extrude(convexity)  
surface(file = "...dat",center,convexity)  
projection(cut)  
render(convexity)  
children([idx])
```

List Comprehensions

```
Generate [ for (i = range|list) i ]  
Conditions [ for (i = ...) if (condition(i)) i ]  
Assignments [ for (i = ...) let (assignments) a ]
```

Special variables

\$fa	minimum angle
\$fs	minimum size
\$fn	number of fragments
\$t	animation step
\$vpr	viewport rotation angles in degrees
\$vpt	viewport translation
\$vpd	viewport camera distance
\$children	number of module children