

2D

`circle(r=1 | d)`

`square(size | [w,h], center)`

`polygon(points, paths, convexity)`

`text(str, size, font, halign, valign, spacing, dir, lang, script)`

`import("...dxf|.svg")`

Polygones réguliers: `circle(r|d, $fn=3|4|...)`

3D

`sphere(r | d)`

`cylinder(h, r | d, center)`

`cylinder(h, r1 | d1, r2 | d2, center)`

`cube(size | [w,d,h], center)`

`polyhedron(points, faces, convexity)`

`surface("...dat|.png", center, invert, convexity)`

`import("...stl")`

2D / 3D

`linear_extrude(h, center, convexity, twist, slices, scale)`

`projection(cut)`

`rotate_extrude(convexity)`

Transformations

`translate([x,y,z])`

`rotate([x,y,z])`

`scale([x,y,z])`

`resize([x,y,z], auto | [autox,autoy,autoz])`

`offset(r | delta, chamfer)`

`multmatrix(m)`

`mirror([x,y,z])`

`color("name" | [r,g,b,a] | [r,g,b], a)`

resize: auto calcule avec paramètres à 0

Opérations

`union()`

`difference()`

`intersection()`

`hull()`

`minkowski()`

Rendu

`$fa` angle minium

`$fs` taille face minimum

`$fn` nombre de faces

`$t` timeur animation

Affichage

`*` ignore

`#` debug

`%` transparent

`!` ignore avant

Point de vue

`$vpr` rotation (°)

`$vpt` translation

`$vpd` distance caméra



By **Vicnet**
cheatography.com/vicnet/

Not published yet.
Last updated 23rd November, 2015.
Page 1 of 2.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>

Module	
module <i>name</i> (...) { ... }	déclaration
children(<i>index</i>)	
\$children	nombre d'enfants
parent_module(<i>n</i>)	nom module niveau n

Fonctions	
function <i>name</i> (...) = exp ;	déclaration

Librairies	
include <...scad>	importe et exécute
use <...scad>	importe seulement
File->Show Library Folder...	
<ul style="list-style-type: none"> installation/libraries <ul style="list-style-type: none"> \$HOME/.local/share/OpenSCAD/libraries OPENS CADPATH répertoires multiples : 	

Contrôles	
if (...) { ... }	
for (i = [start:incr:end]) { ... }	intervalle
for (i = [x, x, ...]) { ... }	vecteur
intersection_for(i = ...) { ... }	évite union

Liste	
[for (i = range(list) i]	génération
[for (i = ...) if (condition(i)) i]	condition
[for (i = ...) let(assign) a]	assignement

Chaines	
chr(<i>n</i>)	
len(str)	
search(value, str)	index
str(str, str, ...)	concatène

Basique	
variable = expression;	
let(name = expression)	
// / .../	commentaire
9.99 1.2e20 PI	nombre
true false	booléen
"..."	chaîne
[exp, ...]	vecteur
[start:incr:end]	intervalle
undef	indéfini
+ - * / %	scalaire
+ - * /	vecteur ou matrice
== != < <= > >=	relation
&&	logique
boolexp ? trueexp :	conditionnel
falseexp	

Système	
echo(list)	
version()	vecteur
vector_num()	nombre

Vecteurs	
concat(v1, v2, ...)	concatène vecteurs
concat(x, x, ...)	crée un vecteur
cross(v, w)	produit vectoriel
len(v)	
lookup(key, v)	valeur ou interpolation
max(v)	
min(v)	
norm(v)	longueur
search(value, v)	index

Mathématique	
abs(x)	
sign(x)	-1, 0, 1
round(x)	plus proche
ceil(x)	supérieur
floor(x)	inférieur
exp(x)	
log(x)	
ln(x)	
max(x, x, ...)	
min(x, x, ...)	
%	modulo
pow(x,y)	
sqrt(x)	
rands(min, max, n, seed)	vecteur taille n

Trigonométrie	
cos(deg)	
sin(deg)	
tan(deg)	
acos(x)	en degrés
asin(x)	en degrés
atan(x)	en degrés
atan2(x,y)	en degrés

