

☐ 基本構文

☐ 変数

`$varname`

名前が *varname* の変数を表します。

☐ ブロック

`[ expr1 [expr2 [...]] ]`

命令群をまとめます。ブロックの値はブロックで最後に評価されたものとなります。

☐ 文の区切り

複数の命令を一行に記述する際、文の区切りが曖昧になる事があります。その場合セミコロン“;”によって明示的に文を区切る事ができます。

`expr1; expr2; ...`

☐ クォート

関数にオブジェクトの名前を渡したい場合など入力の評価して欲しくない事があります。シングルクォーテーション“'”に続けて記述されたものは評価されずにそれそのものを返します。

`' quoting`

☐ 関数の呼び出し

`function [arg1 [arg2 [...]]]`

関数を呼び出すには関数名に続いて引数を並べて記述します。また関数によっては引数を省略できる場合があります。ただし *n* 番目の引数を省略する場合に *n*-1 番目までの引数を省略することはできません。

次のように関数名の後に空白を入れずにカギ括弧“[]”を記述し、その中に引数をコンマ“,”で区切り記述する方法をとることもできます。省略可能な引数があり次の文との区切りが曖昧になるときなどに使います。

`function[ [arg1 [, arg2 [, ...]] ]`

☐ リテラル（データ型）

☐ 数値

☐ 整数

整数を表す型です。10 進法を用いて記述します。

`12345`

☐ 実数

整数を含まない実数を表す型です。10 進法を用いて記述します。数字の中に小数点が含まれていると実数と解釈されます。

`123.45`

☐ 文字列

文字を表す型です。ダブルクォーテーション “” で囲んで記述します。

```
"string"
```

☐ リスト

複数の要素を内部に持つことのできるデータ構造です。リスト自体を要素にすることもできます。要素をコンマ “,” で区切って並べ、中括弧 “{ }” で囲んで記述します。

```
{ [item1 [, item2 [, ...]]] }
```

☐ 真偽値

真と偽の状態を表す型です。真を **T**、偽を **NIL** で記述します。

```
{T | NIL}
```

☐ 定数

☐ on/off（特殊記号）

状態のオン、オフを設定する際に使用します。

```
#on ; オン
```

```
#off ; オフ
```

☐ 色

色を指定する際に使用します。

```
'red ; 赤
```

```
'blue ; 青
```

```
'green ; 緑
```

```
'yellow ; 黄
```

```
'pink ; ピンク
```

```
'black ; 黒
```

```
'white ; 白
```

```
'gray ; 灰色
```

```
'grey ; 灰色
```

```
'darkgray ; 濃い灰
```

```
'darkgrey ; 濃い灰
```

```
'lightgray ; 薄い灰
```

```
'lightgrey ; 薄い灰
```

```
'darkgreen ; 濃い緑
```

```
'lightgreen ; 薄い緑
```

```
'lightblue ; 薄い青
```

```
'purple ; 紫
```

```
'orange ; 橙
```

☐ オブジェクト

初期状態から配置されているオブジェクトです。

```
'turtle' ; 描画タートル  
'eye'      ; 視点タートル  
'light'    ; デフォルト光源
```

☐ 演算子

☐ :=

```
 $\$var := thing$ 
```

変数にオブジェクトを代入します。

- *var* : 変数名
- *thing* : 代入するオブジェクト

☐ 算術演算子

☐ +

```
 $number_1 + number_2$ 
```

$number_1$ と $number_2$ とを加算します。

- $number_1, number_2$: 加算する数 ; 数値

☐ -

```
 $number_1 - number_2$ 
```

$number_1$ から $number_2$ を減算します。

- $number_1, number_2$: 減算する数 ; 数値

☐ *

```
 $number_1 * number_2$ 
```

$number_1$ と $number_2$ を乗算します。

- $number_1, number_2$: 乗算する数 ; 数値

☐ /

```
 $number_1 / number_2$ 
```

$number_1$ から $number_2$ を除算します。

- $number_1, number_2$: 除算する数 ; 数値

☐ ^

```
 $number_1 \wedge number_2$ 
```

べき乗を求めます。

- $number_1$: 底 ; 数値
- $number_2$: 指数 ; 数値

☐ 比較演算子

☐ >

$number_1 > number_2$

$number_1$ が $number_2$ より大きければ真となります。

- $number_1, number_2$: 比較する数値 ; 数値

☐ >=

$number_1 >= number_2$

$number_1$ が $number_2$ より大きいとか等しければ真となります。

- $number_1, number_2$: 比較する数値 ; 数値

☐ <

$number_1 < number_2$

$number_1$ が $number_2$ より小さければ真となります。

- $number_1, number_2$: 比較する数値 ; 数値

☐ <=

$number_1 <= number_2$

$number_1$ が $number_2$ より小さいとか等しければ真となります。

- $number_1, number_2$: 比較する数値 ; 数値

☐ ==

$thing_1 == thing_2$

$thing_1$ と $thing_2$ の内容が等しければ真となります。ただし $thing_1$ と $thing_2$ の型が異なる場合は真となりません。

- $thing_1, thing_2$: 比較するオブジェクト ; 任意のオブジェクト

☐ ===

$thing_1 === thing_2$

$thing_1$ と $thing_2$ が等しければ真となります。

- $thing_1, thing_2$: 比較するオブジェクト ; 任意のオブジェクト

☐ !=

$thing_1 != thing_2$

$thing_1$ と $thing_2$ が等しくなければ真となります。ただし $thing_1$ と $thing_2$ の型が異なる場合も真となります。

- $thing_1, thing_2$: 比較するオブジェクト ; 任意のオブジェクト

☐ 論理演算子

☐ !

$! boolean$

$boolean$ の論理否定を求めます。

- $boolean$: 否定する値 ; 真偽値

☐ &&

$boolean_1 \&\& boolean_2$

$boolean_1$ と $boolean_2$ の論理積を求めます。

- $boolean_1, boolean_2$: 論理積を求める値 ; 真偽値



||

$$boolean_1 \text{ || } boolean_2$$

$boolean_1$ と $boolean_2$ の論理和を求めます

- $boolean_1, boolean_2$: 論理和を求める値 ; 真偽値

☐ 制御構造



IF

$$\text{IF } condition \text{ block}_t [block_f]$$

条件によって実行する処理を選択します。

- $condition$: 条件 ; 真偽値
- $block_t$: 条件が真となった場合実行するブロック ; ブロック
- $block_f$: 条件が偽となった場合実行するブロック (省略可能) ; ブロック



REPEAT

$$\text{REPEAT } number \text{ block}$$

命令を指定回数繰り返し実行します。

- $number$: 繰り返す回数 ; 整数
- $block$: 繰り返すブロック ; ブロック



WHILE

$$\text{WHILE } condition \text{ block}$$

条件が真である間、処理を繰り返し実行します。

- $condition$: 条件 ; 真偽値
- $block$: 繰り返すブロック ; ブロック

☐ 関数定義



DEF

$$\text{DEF } name [\$var_1 [\$var_2 [...]]] \text{ block}$$

関数を定義します。 $name$ と “[” は間を空けずに記述します。引数を使用しない場合でも $name$ の後のカギ括弧 “ $]$ ” は必ず記述します。

- $name$: 関数の名前
- $\$var_1, \$var_2, \dots$: 関数の仮引数
- $block$: 関数とする命令群 ; ブロック



GETD

$$\text{GETD } name$$

定義を問い合わせます。

- $name$: 問い合わせる定義の名前

☐ LOAD

LOAD

ファイル選択ダイアログを表示し、選択したプログラムを読み込みます。

☐ デバッグ関数

☐ DEBUG

DEBUG *name*

関数のデバッグを有効にします。

- *name* : デバッグを有効にする関数の名前

☐ UNDEBUG

UNDEBUG *name*

関数のデバッグを無効にします。

- *name* : デバッグを無効にする関数の名前

☐ TRACE

TRACE *name*

関数のトレースを有効にします。

- *name* : トレースを有効にする関数の名前

☐ UNTRACE

UNTRACE *name*

関数のトレースを無効にします。

- *name* : トレースを無効にする関数の名前

☐ 整数演算関数

☐ MOD

MOD *number*₁ *number*₂

整数の剰余を求めます。

- *number*₁ : 被除数; 整数
- *number*₂ : 法; 整数

☐ ROUND

ROUND *number*

number に最も近い整数を求めます。

- *number* : 丸める数; 実数

☐ FLOOR

FLOOR *number*

number 以下で最も大きい整数を求めます。

- *number* : 丸める数; 実数

☐ 数学関数

☐ SIN

SIN *angle*

正弦を求めます。

- *angle* : 角度 [度]; 数値

☐ COS

COS *angle*

余弦を求めます。

- *angle* : 角度 [度]; 数値

☐ TAN

TAN *angle*

正接を求めます。

- *angle* : 角度 [度]; 数値

☐ ASIN

ASIN *sin*

逆正弦を求めます。

- *sin* : 正弦 ; 数値

☐ ACOS

ACOS *cos*

逆余弦を求めます。

- *cos* : 余弦 ; 数値

☐ ATAN

ATAN *tan*

逆正接を求めます。

- *tan* : 正接 ; 数値

☐ RANDOM

RANDOM *number*

0 以上 *number* 未満の乱数を生成します。*number* が整数の場合は整数、実数の場合は実数で生成します。

- *number* : 乱数の上限値 ; 数値

☐ 検査関数

☐ ODD?

ODD? *number*

number が偶数なら真となります。

- *number* : 検査する値 ; 数値

☐ EVEN?

EVEN? *number*

number が奇数なら真となります。

- *number* : 検査する値 ; 数値

☐ ZERO?

ZERO? *number*

number が 0 なら真となります。

- *number* : 検査する値 ; 数値

☐ 入出力関数

☐ PRINT

PRINT *thing*

オブジェクトをコンソールに印字します。

- *thing* : 印字するオブジェクト ; 任意のオブジェクト

☐ リスト関数

☐ CAR

CAR *list*

list の先頭の要素を返します。

- *list* : リスト ; リスト

☐ CDR

CDR *list*

list の先頭の要素を取り除いたリストを返します。

- *list* : リスト ; リスト

☐ CONS

CONS *item list*

item を *list* の先頭に追加したリストを返します。

- *item* : 追加するオブジェクト ; 任意のオブジェクト
- *list* : リスト ; リスト

☐ APPEND

APPEND *list₁ [list₂ [...]]*

与えられたリストを連結したリストを返します。

- *list₁, list₂, ...*: 連結するリスト ; リスト

☐ 色関数

☐ RGB

RGB $component_r$ $component_g$ $component_b$

RGB 成分から COLOR オブジェクトを生成します。

- $component_r$: 赤色成分 ; 数値 [0.0-1.0]
- $component_g$: 緑色成分 ; 数値 [0.0-1.0]
- $component_b$: 青色成分 ; 数値 [0.0-1.0]

☐ RGBA

RGBA $component_r$ $component_g$ $component_b$ $alpha$

RGBA 成分から COLOR オブジェクトを生成します。

- $component_r$: 赤色成分 ; 数値 [0.0-1.0]
- $component_g$: 緑色成分 ; 数値 [0.0-1.0]
- $component_b$: 青色成分 ; 数値 [0.0-1.0]
- $alpha$: 透明度 ; 数値 [0.0-1.0]

☐ ワールド操作関数

☐ DRAW

DRAW

全てを初期状態にします。

☐ SETBG

SETBG $color$

背景色を設定します。

- $color$: 色 ; COLOR オブジェクト

☐ REFRESH

REFRESH

画面を強制的に更新します。

☐ MANUAL

MANUAL

自動画面更新を無効にします。

☐ AUTOMATIC

AUTOMATIC

自動画面更新を有効にします。更新タイミングはコマンド実行後毎回です。

☐ ADHOC

ADHOC

自動画面更新を有効にします。更新タイミングはシステムに任せます。

☐ REFMODE

REFMODE

現在の画面更新方法を問い合わせます。

☐ FOG

FOG *switch*

フォグ効果の on/off を設定します。初期状態では off になっています。

- *switch* : on/off; [#on, #off]

☐ DOME

DOME *switch*

ドームの表示の on/off を切り替えます。初期状態では off になっています。

- *switch* : on/off; [#on, #off]

☐ MESH

MESH *switch*

グリッドの表示の on/off を切り替えます。初期状態では on になっています。

- *switch* : on/off; [#on, #off]

☐ GROUND

GROUND *switch*

地面の表示の on/off を切り替えます。初期状態では off になっています。

- *switch* : on/off; [#on, #off]

☐ DUMP

DUMP

画面を画像として保存します。

☐ DUMP_DIR

DUMP_DIR *dir*

DUMP 関数で画像を保存するディレクトリを設定します。

- *dir* : 保存先ディレクトリ名; 文字列

☐ DUMP_COUNT

DUMP_COUNT *number*

DUMP 関数で保存される画像ファイル名の番号の初期値を設定します。

- *number* : 初期値; 整数

☐ 光源設定関数

☐ SET_LIGHT

SET_LIGHT *color_{diffuse} color_{specular} color_{ambient} exponent angle*

光源を設定します。光源オブジェクトを選択していなければなりません。

- *color_{diffuse}* : 拡散反射光色 ; COLOR オブジェクト
- *color_{specular}* : 鏡面反射光色 ; COLOR オブジェクト
- *color_{ambient}* : 環境光色 ; COLOR オブジェクト
- *exponent* : スポットライト指数 ; 数値 [0-128]
- *angle* : スポットライトの拡散角 ; 数値 [0-90, 180]

☐ SET_L_DFFS

SET_L_DFFS *color*

光源の拡散反射光色を設定します。光源オブジェクトを選択していなければなりません。

- *color* : 拡散反射光色 ; COLOR オブジェクト

☐ SET_L_SPCL

SET_L_SPCL *color*

光源の鏡面反射光色を設定します。光源オブジェクトを選択していなければなりません。

- *color* : 鏡面反射光色 ; COLOR オブジェクト

☐ SET_L_AMBT

SET_L_AMBT *color*

光源の環境光色を設定します。光源オブジェクトを選択していなければなりません。

- *color* : 環境光色 ; COLOR オブジェクト

☐ SET_L_SPOT

SET_L_SPOT *exponent angle*

スポットライトの属性を設定します。光源オブジェクトを選択していなければなりません。

- *exponent* : スポットライト指数 ; 数値 [0-128]
- *angle* : スポットライトの拡散角 ; 数値 [0-90, 180]

☐ オブジェクト選択関数

☐ TELL

TELL *object*

オブジェクトを選択します。

- *object* : 選択するオブジェクトの名前

☐ TELLA

TELLA *object*

オブジェクトを追加選択します。

- *object* : 追加するオブジェクトの名前

☐ UNTELL

UNTELL *object*

選択中のオブジェクトから取り除きます。

- *object* : 取り除くオブジェクトの名前

☐ ASK

ASK *objects block*

特定のオブジェクトに対して命令を実行します。命令実行中は一時的に *objects* を選択状態にします。

- *objects* : 命令を実行させるオブジェクト名のリスト; リスト
- *block* : 実行する命令のブロック; ブロック

☐ WHO?

WHO?

選択中のオブジェクトを問い合わせます。

☐ ALIAS

ALIAS *name object*

オブジェクトに別名を与えます。

- *name* : オブジェクトに与える別名
- *object* : 別名を与えるオブジェクト

☐ ALL

ALL

全オブジェクトのリストを問い合わせます。

☐ OBJLIST

OBJLIST

ユーザの作成したオブジェクトのリストを問い合わせます。

☐ PUT_TELL

PUT_TELL *switch*

プリミティブ生成関数によってオブジェクトを生成した後、そのオブジェクトに TELL するかを切り替えます。

- *switch* : TELL するかどうかの on/off; [#on, #off]

☐ オブジェクト運動関数

☐ FD

FD *distance*

選択しているオブジェクトを前進させます。

- *distance* : 移動距離; 数値

☐ BK

BK *distance*

選択しているオブジェクトを後退させます。

- *distance* : 移動距離 ; 数値

☐ ML

ML *distance*

選択しているオブジェクトを左に移動させます。

- *distance* : 移動距離 ; 数値

☐ MR

MR *distance*

選択しているオブジェクトを右に移動させます。

- *distance* : 移動距離 ; 数値

☐ MU

MU *distance*

選択しているオブジェクトを上に移動させます。

- *distance* : 移動距離 ; 数値

☐ MD

MD *distance*

選択しているオブジェクトを下に移動させます。

- *distance* : 移動距離 ; 数値

☐ LT

LT *angle*

選択しているオブジェクトを左に向けます。

- *angle* : 回転角度 [度]; 数値

☐ RT

RT *angle*

選択しているオブジェクトを右に向けます。

- *angle* : 回転角度 [度]; 数値

☐ UP

UP *angle*

選択しているオブジェクトを上に向けます。

- *angle* : 回転角度 [度]; 数値

☐ DN

DN *angle*

選択しているオブジェクトを下に向けます。

- *angle* : 回転角度 [度]; 数値

☐ RL

RL *angle*

選択しているオブジェクトを左に傾けます。

- *angle* : 回転角度 [度]; 数値

☐ RR

RR *angle*

選択しているオブジェクトを右に傾けます。

- *angle* : 回転角度 [度]; 数値

☐ SETPOS

SETPOS *coord_x coord_y coord_z*

選択しているオブジェクトを座標に移動します。

- *coord_x* : *x* 座標 ; 数値
- *coord_y* : *y* 座標 ; 数値
- *coord_z* : *z* 座標 ; 数値

☐ SETX

SETX *coord*

x 座標について選択しているオブジェクトを座標に移動します。

- *coord* : *x* 座標 ; 数値

☐ SETY

SETY *coord*

y 座標について選択しているオブジェクトを座標に移動します。

- *coord* : *y* 座標 ; 数値

☐ SETZ

SETZ *coord*

z 座標について選択しているオブジェクトを座標に移動します。

- *coord* : *z* 座標 ; 数値

☐ HOME

HOME

選択しているオブジェクトの位置と向きを初期化します。

☐ オブジェクト制御関数

☐ SHOW

SHOW

選択しているオブジェクトを表示するようにします。

☐ HIDE

HIDE

選択しているオブジェクトを表示しないようにします。

☐ DUP

DUP

選択しているオブジェクトを複製します。

☐ DELETE

DELETE [*name*]

選択しているオブジェクトを削除します。*name* が与えられている場合は *name* で示されるオブジェクトを削除します。

- *name* : オブジェクト名 (省略可能)

☐ ペン操作関数

☐ PD

PD

ペンを下げ、タートルによる軌跡の描画を有効にします。

☐ PU

PU

ペンを上げ、タートルによる軌跡の描画を無効にします。

☐ SETPC

SETPC *color*

ペンの描画色を設定します。

- *color* : 描画色 ; COLOR オブジェクト

☐ SETPS

SETPS *size*

ペンのサイズを設定します。

- *size* : サイズ ; 数値

☐ PENCOLOR

PENCOLOR

現在のペンの描画色を問い合わせます。

☐ 原点タートル操作関数

☐ OFD

OFD *distance*

選択しているオブジェクトの原点タートルを前進させます。

- *distance* : 移動距離; 数値

☐ OBK

OBK *distance*

選択しているオブジェクトの原点タートルを後退させます。

- *distance* : 移動距離; 数値

☐ OML

OML *distance*

選択しているオブジェクトの原点タートルを左に移動させます。

- *distance* : 移動距離; 数値

☐ OMR

OMR *distance*

選択しているオブジェクトの原点タートルを右に移動させます。

- *distance* : 移動距離; 数値

☐ OMU

OMU *distance*

選択しているオブジェクトの原点タートルを上に移動させます。

- *distance* : 移動距離; 数値

☐ OMD

OMD *distance*

選択しているオブジェクトの原点タートルを下に移動させます。

- *distance* : 移動距離; 数値

☐ OLT

OLT *angle*

選択しているオブジェクトの原点タートルを左に向けます。

- *angle* : 回転角度 [度]; 数値

☐ ORT

ORT *angle*

選択しているオブジェクトの原点タートルを右に向けます。

- *angle* : 回転角度 [度]; 数値

☐ OUP

OUP *angle*

選択しているオブジェクトの原点タートルを上に向けます。

- *angle* : 回転角度 [度]; 数値

☐ ODN

ODN *angle*

選択しているオブジェクトの原点タートルを下に向けます。

- *angle* : 回転角度 [度]; 数値

☐ ORL

ORL *angle*

選択しているオブジェクトの原点タートルを左に傾けます。

- *angle* : 回転角度 [度]; 数値

☐ ORR

ORR *angle*

選択しているオブジェクトの原点タートルを右に傾けます。

- *angle* : 回転角度 [度]; 数値

☐ OSHOW

OSHOW

選択しているオブジェクトの原点タートルを表示するようにします。

☐ OHIDE

OHIDE

選択しているオブジェクトの原点タートルを表示しないようにします。

☐ 軌跡オブジェクト関数

☐ HOLD

HOLD [*name*]

それまでに描画した軌跡をオブジェクト化します。*name* が与えられている場合、その名前をつけてオブジェクト化します。

- *name* : 軌跡オブジェクトに与える名前（省略可能）

☐ SFD

SFD *distance*

選択している軌跡オブジェクトを前進させ、その軌跡から面を生成します。

- *distance* : 押し出す距離 ; 数値

☐ SLT

SLT *angle*

選択している軌跡オブジェクトを左に向け、その軌跡から面を生成します。

- *angle* : 回転角度 [度]; 数値

☐ SRT

SRT *angle*

選択している軌跡オブジェクトを右に向け、その軌跡から面を生成します。

- *angle* : 回転角度 [度]; 数値

☐ SUP

SUP *angle*

選択している軌跡オブジェクトを上に向け、その軌跡から面を生成します。

- *angle* : 回転角度 [度]; 数値

☐ SDN

SDN *angle*

選択している軌跡オブジェクトを下に向け、その軌跡から面を生成します。

- *angle* : 回転角度 [度]; 数値

☐ SRL

SRL *angle*

選択している軌跡オブジェクトを左に傾け、その軌跡から面を生成します。

- *angle* : 回転角度 [度]; 数値

☐ SRR

SRR *angle*

選択している軌跡オブジェクトを右に傾け、その軌跡から面を生成します。

- *angle* : 回転角度 [度]; 数値

☐ MAKE_SKIN

MAKE_SKIN

選択している軌跡オブジェクトの軌跡から面を生成します。

☐ SKIN

SKIN *switch*

選択している軌跡オブジェクトについて、軌跡から面を生成するかどうかを設定します。

- *switch* : 面の on/off; [#on, #off]

☐ 材質設定関数

☐ SET_MATERIAL

SET_MATERIAL *color_{diffuse} color_{specular} color_{ambient} exponent color_{emission}*

選択しているオブジェクトの材質を設定します。

- *color_{diffuse}* : 拡散反射係数; COLOR オブジェクト
- *color_{specular}* : 鏡面反射係数; COLOR オブジェクト
- *color_{ambient}* : 環境反射係数; COLOR オブジェクト
- *exponent* : 鏡面指数; 数値 [0-128]
- *color_{emission}* : 発光色; COLOR オブジェクト

☐ SET_DFFS

SET_DFFS *color*

選択しているオブジェクトの拡散反射係数を設定します。

- *color* : 拡散反射係数; COLOR オブジェクト

☐ SET_SPCL

SET_SPCL *color*

選択しているオブジェクトの鏡面反射係数を設定します。

- *color* : 鏡面反射係数; COLOR オブジェクト

☐ SET_EMSS

SET_EMSS *color*

選択しているオブジェクトの発光色を設定します。

- *color* : 発光色; COLOR オブジェクト

☐ SET_SHNN

SET_SHNN *exponent*

選択しているオブジェクトの鏡面指数を設定します。

- *exponent* : 鏡面指数; 数値 [0-128]

☐ SET_AMBT

SET_AMBT *color*

選択しているオブジェクトの環境反射係数を設定します。

- *color* : 環境反射係数; COLOR オブジェクト

☐ SET_ALPHA

SET_ALPHA *alpha*

選択しているオブジェクトの透明度を設定します。全ての反射光成分が同一の値に変更されます。

- *alpha* : 透明度; 数値 [0.0-1.0]

☐ テクスチャ設定関数

☐ SET_TEXTURE

SET_TEXTURE *filename*

選択しているオブジェクトのテクスチャ画像を設定します。

- *filename* : テクスチャ画像のファイル名; 文字列

☐ TEXTURE

TEXTURE *switch*

選択しているオブジェクトのテクスチャの on/off を切り替えます。初期状態では on になっています。

- *switch* : テクスチャの on/off; [#on, #off]

☐ SET_TEXMODE

SET_TEXMODE *mode*

選択しているオブジェクトのテクスチャと材質の色のブレンド方式を設定します。

- *mode* : ブレンド方式; 整数 [0-3]
 - 0: 変調する
 - 1: 混合色
 - 2: テクスチャのみ
 - 3: 貼り付け

☐ プリミティブオブジェクト生成関数

☐ RECT

RECT *width height*

長方形オブジェクトを描画タートルの現在位置に生成します。

- *width* : オブジェクトの幅; 数値
- *height* : オブジェクトの高さ; 数値

☐ CUBOID

CUBOID *width height depth*

直方体オブジェクトを描画タートルの現在位置に生成します。

- *width* : オブジェクトの幅; 数値
- *height* : オブジェクトの高さ; 数値
- *depth* : オブジェクトの奥行; 数値

☐ SPHERE

SPHERE *radius*

球オブジェクトを描画タートルの現在位置に生成します。

- *radius* : オブジェクトの半径; 数値

☐ DISK

DISK *radius*

円盤オブジェクトを描画タートルの現在位置に生成します。

- *radius* : オブジェクトの半径 ; 数値

☐ CYLINDER

CYLINDER *radius length*

円筒オブジェクトを描画タートルの現在位置に生成します。

- *radius* : オブジェクトの半径 ; 数値
- *length* : オブジェクトの長さ ; 数値

☐ グループ関数

☐ MAKE_GROUP

MAKE_GROUP [*list*]

選択しているオブジェクトをグループ化します。*list* が与えられた場合はそのオブジェクトをグループ化します。

- *list* : オブジェクト名のリスト（省略可能） ; リスト

☐ DESTRICT

DESTRICT [*name*]

選択しているグループをグループ解除します。*name* が与えられた場合はそのグループをグループ解除します。

- *name* : グループ名（省略可能）