

# 言語 tgl について

## 1 概要

このドキュメントでは、言語 tgl の説明で 8 章では触れられなかった点についてだけ説明しています。8 章の言語 tgl の説明と合わせて読んでください。

言語 tgl は、タートル・グラフィックスのための言語です。タートル・グラフィックスとは、タートル(亀)と呼ばれるグラフィックス・カーソルを使ったコンピュータ・グラフィックスの手法です。タートルはペンを持っていて(亀のしっぽにペンが結わえ付けられている様子を想像してください)、タートルが動くと、その軌跡に沿って線が引かれます。タートルを前進・後退させたり、向きを変えることで、様々な図形を描くことができます。ペンは上げ下げでき、ペンが下がっているときだけ、タートルの軌跡が画面上に描かれます。

tgl では、タートルは図 1 のような 2 次元座標系の表面を移動します。図 1 の四角形で囲まれた領域が実際にウィンドウに表示される範囲です。ウィンドウの中心が原点となり、左上が  $(250, 250)$ 、右下が  $(-250, -250)$  になります。ウィンドウが最初に開いたときには、タートルは図 fig:coord の水色の矢印のように原点にいて、Y 座標の正の方向を向いています(後述のシステム手続き `getDirection()`、`setDirection()` や `towards()` で角度を指定するときには、Y 座標の正の方向を 0 度として、時計回りに角度が増加するものとしします。また、角度の単位はラジアンではなく度です)。

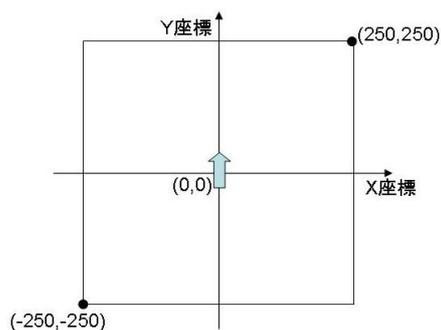


図 1: tgl での座標系とタートルの初期状態

システム定義手続きを使ってペンの色を変更することができます。現在、7色をサポートしています。それぞれの色のペンには 0 から 6 までの番号がついていて、この番号を使ってペンの色を指定します。また、

penBlack(), penOrange() などのシステム定義手続きを使って、ペンの番号を得ることもできます。ペンの番号と色の対応を以下に示します。

番号	0	1	2	3	4	5	6
色	黒	赤	青	緑	黄	オレンジ	紫

## 2 言語 tgl の文法

言語 tgl の文法は、8章に載せた tglc.y の中に埋め込まれていますが、文法だけ取り出したものを以下に示します。なお、ここでは意味がわかりやすいように、非終端記号を日本語で表記しました。

- プログラム → 宣言部 定義部
- 宣言部 →  $\varepsilon$  | 宣言部 宣言
- 宣言 → VAR ID ';' | VAR ID '=' 定数式 ';' |  
| FUNC ID '@' NUM ';' | COMM ID '@' NUM ';' |
- 定数式 → 定数式 ADDOP 定数式 | 定数式 MULOP 定数式  
| SIGNOP 定数式 | '(' 定数式 ')' | NUM
- 定義部 → 定義 | 定義部 定義
- 定義 → 関数定義 | コマンド定義
- 関数定義 → FUNC ID 手続き定義
- コマンド定義 → COMM ID 手続き定義
- 手続き定義 → '(' 仮引数リスト ')' '{' 局所宣言リスト 文リスト '}'
- 仮引数リスト →  $\varepsilon$  | 仮引数リスト'
- 仮引数リスト' → 仮引数 | 仮引数リスト' ',' 仮引数
- 仮引数 → ID
- 局所宣言リスト →  $\varepsilon$  | 局所宣言リスト 局所宣言
- 局所宣言 → VAR ID ';' | VAR ID '=' 式 ';' |
- 文リスト →  $\varepsilon$  | 文リスト 文
- 文 → ID '=' 式 | if 部 else 部  
| WHILE '(' 条件 ')' '{' 文リスト '}'  
| REPEAT '(' 式 ')' '{' 文リスト '}'  
| ID '(' 実引数リスト ')' ';' | RETURN ';' | RETURN 式 ';' |
- if 部 → IF '(' 条件 ')' '{' 文リスト '}'  
| if 部 ELSIF '(' 条件 ')' '{' 文リスト '}'
- else 部 → ELSE '{' 文リスト '}'

式 → 式 ADDOP 式 | 式 MULOP 式  
 | SIGNOP 式 | '( 式 )' | ID | NUM  
 実引数リスト →  $\varepsilon$  | 実引数リスト'  
 実引数リスト' → 式 | 実引数リスト' ', ' 式  
 条件 → 式 RELOP 式

以下の文法に現れる (日本語で書かれた) 非終端記号と 8 章の tglc.y 中の非終端記号の対応表を表??に載せたので、参考にしてください。

上の文法中の非終端記号	tglc.y 中の非終端記号
プログラム	program
宣言部	declPart
宣言	decl
定数式	conExpr
定義部	defPart
定義	definition
関数定義	funcDef
コマンド定義	commDef
手続き定義	procDef
仮引数リスト	paramList
仮引数リスト'	paramList2
仮引数	param
局所宣言リスト	localDeclList
局所宣言	localDecl
文リスト	stmtList
文	stmt
if 部	ifPart
else 部	elsePart
式	expr
実引数リスト	argList
実引数リスト'	argList2
条件	cond

表 1: 非終端記号の対応表

### 3 言語 tgl のシステム定義手続き

言語 tgl の手続きには、ユーザが自分で定義するユーザ定義手続きと、システムがあらかじめ用意しているシステム定義手続きの2種類があります。システム定義手続きを使って、タートルを移動させたりペンの色を変更することができます。

システム定義手続きは宣言も定義もせずに、プログラム中で自由に使うことができます。ただし、システム定義手続きと同じ名前をもつグローバル変数やユーザ定義手続きを宣言したり、システム定義手続きをプログラム中であらためて定義したりするとコンパイルエラーになります。

言語 tgl のシステム手続きの一覧を以下に示します。手続きの種別は、その手続きが関数かコマンドかを表します。関数は値を返す手続き、コマンドは値を返さない手続きです。

手続き名	手続きの種別	説明
move(x)	コマンド	タートルを x ステップ前進させる。負の数を指定すると  x  ステップ後退する。
turn(d)	コマンド	タートルを d 度右へ回転させる。負の数を指定すると  d  度左へ回転する。
setPos(x,y)	コマンド	タートルを位置 (x,y) へ置く。
setDirection(d)	コマンド	タートルを d 度の方向へ向ける。
getX()	関数	タートルの現在の X 方向の位置を返す。
getY()	関数	タートルの現在の Y 方向の位置を返す。
getDirection()	関数	タートルが現在向いている方向を返す。
towards(x,y)	関数	タートルの現在位置から位置 (x,y) への角度を返す。
getDistance(x,y)	関数	タートルの現在位置から位置 (x,y) までの距離を返す。
getPenColorNum()	関数	システムが用意しているペンの色数を返す。
setPenColor(c)	コマンド	ペンの色を c にする。c が 0 以上 getPenColorNum() 未満でないときは、警告してペンの色を変えない。
getPenColor()	関数	現在のペンの色を返す。
penBlack()	関数	黒色のペンの番号を返す。
penRed()	関数	赤色のペンの番号を返す。
penBlue()	関数	青色のペンの番号を返す。
penGreen()	関数	緑色のペンの番号を返す。
penYellow()	関数	黄色のペンの番号を返す。
penOrange()	関数	オレンジ色のペンの番号を返す。
penViolet()	関数	紫色のペンの番号を返す。
getNextColor(c)	関数	現在のペンの番号 c の次の番号を返す。 c が getPenColorNum()-1 なら 0 を返す。
penDown()	コマンド	ペンを下ろす。
penUp()	コマンド	ペンを上げる。
clear()	コマンド	ウィンドウの描画内容をすべて消去する。
home()	コマンド	タートルを原点に移動し、Y 座標の正の方向を向かせる。

手続き名	手続きの種別	説明
isKeyPressed()	関数	キーが押されていたら 1 を、そうでなければ 0 を返す。
readChar()	関数	キーが押された文字の ASCII 文字コードを返す。
sleep(t)	コマンド	t ミリ秒休止する。
sin(x)	関数	x の正弦 (サイン) を返す。
cos(x)	関数	x の余弦 (コサイン) を返す。
sqrt(x)	関数	x の平方根を返す。