

1. 十進 BASIC の基本の基本

- (1) プログラムは通常，上から順に実行される。
 プログラム実行の最後には「END」が必要
- (2) 文字列定数以外は英字の大文字小文字は区別されない(どちらを用いてもよい)
- (3) 変数名は英字に，英字，数字，または_を続けたものを用いる。
 また，文字列変数(文字列を代入するための変数)には末尾に「\$」をつける。
- (4) 代入文 LET_変数名=定数または式 (LET は必ず必要)
 INPUT_変数名
 INPUT_ PROMPT_"メッセージ":変数名
- (5) 数値演算 足し算 + 引き算 - 掛け算 * 割り算 /
 巾乗 ^ 2⁵ は 2~5
 演算のカッコは何重になっても丸カッコ()のみを用いる
- (6) 出力文 PRINT_定数または式
 (次に表示されるものは改行して表示される)
 PRINT_定数または式,
 (コンマで終われば，改行せずに間隔をあけて表示)
 PRINT_定数または式;
 (セミコロンで終われば，改行せずに間隔をあけずに表示)
 コンマやセミコロンで区切って，複数個表示することもできる。
- (7) 組み込み関数について
- (i) ABS(x) x の絶対値 $|x|$
- (ii) SQR(x) x の非負の平方根 \sqrt{x}
- (iii) INT(x) x を超えない最大の整数 $[x]$ (Gauss の記号)
- (iv) $\sin x$ SIN(x) $\cos x$ COS(x) $\tan x$ TAN(x)
 $\sin^{-1} x$ ASIN(x) $\cos^{-1} x$ ACOS(x) $\tan^{-1} x$ ATN(x)
- (v) e^x EXP(x) $\log_e x$ LOG(x) $\log_{10} x$ LOG10(x)
- (vi) RND 関数 $0 \leq \text{RND} < 1$ となる一様乱数を与える。
 最初に異なる乱数からはじめるためには，予め RANDOMIZE 文が必要。

2 . ループ (繰り返し)

(1) FOR ~ NEXT 文

```
FOR 変数=数値 1 TO 数値 2 [STEP 数値 3]
  命令 1
  命令 2
  ...
NEXT 変数
```

変数が数値 1 から数値 2 を越えるまで数値 3 の間隔で増加しながら，各命令の実行を繰り返す．

(2) DO WHILE ~ LOOP 文 (前判定反復)

```
DO WHILE 条件
  命令 1
  命令 2
  ...
LOOP
```

条件が満たされている間は各命令を繰り返す．ループに入る前に条件をチェックするので，条件が真でなければ 1 回も実行されない．

条件について

- (i) \neq は 『<>』 または 『><』
- (ii) \geq は 『>=』 または 『=>』
- (iii) \leq は 『<=』 または 『=<』
- (iv) 複数の条件は 『AND』，『OR』 で結ぶ．
- (v) 否定の条件は 『NOT』 ．

(3) DO ~ LOOP WHILE 文 (後判定反復)

```
DO
  命令 1
  ...
LOOP WHILE 条件
```

条件が満たされている間は各命令を繰り返す．ループに入った後で条件をチェックするので，条件が真でなくても 1 回は実行される．

(4) DO ~ LOOP UNTIL 文 (後判定反復)

```

DO_
  _命令 1
  ...
LOOP_UNTIL_条件

```

条件が真になるまで (偽の間は) 各命令を繰り返す。

(5) DO UNTIL ~ LOOP 文 (前判定反復)

```

DO_UNTIL_条件
  _命令 1
  ...
LOOP

```

条件が真になるまで (偽の間は) 各命令を繰り返す。

3 . 条件分岐

(1) 単純 IF 文 (1 行で記述)

```
IF_条件_THEN_命令
```

条件が真であれば命令を実行する。条件が偽であれば命令は実行されず次の行へ行く。

```
IF_条件_THEN_命令 1_ELSE_命令 2
```

条件が真であれば命令 1 を実行し、条件が偽であれば命令 2 を実行する。

(2) IF ~ THEN ~ END IF 文 (複数行で記述)

```

IF_条件_THEN
  _命令 1
  _命令 2
  ...
END_IF

```

条件が真であれば各命令を実行する。条件が偽であれば命令は実行されず END_IF の次の行へ行く

(3) IF ~ THEN ~ ELSE ~ END IF 文 (複数行で記述)

```

IF_条件 THEN
  命令 1
ELSE
  命令 3
END_IF

```

条件が真であれば命令 1 を実行する．条件が偽であれば命令 3 を実行する．

(4) ELSEIF

```

IF_条件 1 THEN
  命令 1
ELSEIF_条件 2 THEN
  命令 2
ELSE
  命令 3
END_IF

```

条件 1 が真であれば命令 1 を実行する．条件 1 が偽で条件 2 が真であれば命令 2 を実行する．条件 1 も条件 2 も偽であれば命令 3 を実行する．

(5) 無限ループとループからの脱出

```

DO
  .....
LOOP

```

のように WHILE や UNTIL がなくても，DO ~ LOOP を用いてよいが，このままでは，繰り返しが無限回になってしまう．あるいは，WHILE や UNTIL のある DO ~ LOOP 文や FOR ~ NEXT 文においても途中でループを抜け出したい場合がある．このような場合には，EXIT_DO や EXIT_FOR を用いて，ループを抜け出すことが出来る．

(6) 多重ループ

FOR ~ NEXT 文や DO ~ LOOP 文は 2 重，3 重に用いることが出来る．使い方については後で具体例で示すことにするが，EXIT_FOR や EXIT_DO を用いてループから抜け出せるのは，すぐ外のループにしか抜け出せない．もっと外のループに抜け出すには GOTO 文を用いる．GOTO 文は余り使わない方がよい (= 分かりやすい) とされている．

GOTO_行番号 : プログラムの実行を指定された行番号の行に移動する．

行番号は付けたい行の先頭に自然数を用い，下の行ほど大きい番号を付けなければならない．行番号の直後には半角のスペースが必要である．

3 . グラフィックについて

(1) 座標系の設定

SET_WINDOW 左端座標, 右端座標, 下端座標, 上端座標

(2) 点を描く

PLOT_POINTS: x 座標, y 座標

予め, 打つ点のスタイルを次の命令で定めておく.

SET_POINT_STYLE 数値式

マークの形 (point style) を設定する .

1 . 2 + 3 * 4 5 x

(3) 線を描く

PLOT_LINES: x1, y1; x2, y2

(x_1, y_1) から (x_2, y_2) までを線分で結ぶ .

線の色やスタイルを指定したい場合は次の命令で行う .

SET_LINE_COLOR 数値式

色は 0 から 255 までが利用でき, あらかじめ次のように割り当てられている .

0 白, 1 黒, 2 青, 3 緑, 4 赤, 5 水色, 6 黄色, 7 赤紫, 8 灰色, 9 濃い青,

10 濃い緑, 11 青緑, 12 えび茶, 13 オリーブ色, 14 濃い紫, 15 銀色, ...

SET_LINE_STYLE 数値式

線の種類は, 1 実線, 2 破線, 3 点線, 4 一点鎖線

(4) 円を描く

DRAW_circle (現在の line color で原点を中心とする半径 1 の円を描く)

点 (x, y) を中心とする半径 r の円を描きたいときは,

DRAW_circle_WITH_SCALE(r)*SHIFT(x, y)

(5) 座標軸や格子を描く

DRAW_AXES (x 軸と y 軸を描く)

DRAW_GRID (x 軸方向 間隔 1, y 軸方向 間隔 1 の格子を描く)

(6) 画面消去

CLEAR

(7) 色を塗る

FLOOD_x, y (点 (x, y) を始点として点 (x, y) と同色でつながる領域を現在の area color で塗りつぶす.)

PAINT_x, y (点 (x, y) を始点として line color の点を境界とする領域を現在の area color で塗りつぶす)

塗りつぶす色の色の設定は

SET_AREA_COLOR 数値式

十進 BASIC の追加説明 1

関数の定義 1

DEF関数名(変数)=変数の数式

PLOT_LINES 文の補足

PLOT_LINES:x1,y1;

(終りがセミコロンで終わるときは紙の上にペンを置いたままの状態)

次の PLOT_LINES の点までを線分で結ぶ

PLOT_LINES:x1,y1

(終りがセミコロンで終わらないときは紙の上からペンを離れた状態)

PLOT_LINES

(コロン以下がないときは、単にペンを紙から離す)

十進 BASIC の追加説明 2

エラー処理

WHEN_EXCEPTION_IN

命令 1

USE

命令 2

END_WHEN

命令 1 の中でエラーが生じれば命令 1 を行わず、命令 2 を行う。

十進 BASIC の追加説明 3

配列について

DIM 配列名と、その次元、添字の上限、下限を宣言する。

例 DIM A(10)

配列 A の次元を 1 とし、添字の上限を 10 とする。

添字の下限は OPTION BASE 文の指定に従う。

OPTION BASE 文がなければ添字の下限は 1。

例 DIM A(4,5)

配列 A の次元を 2 とし、第 1 次元の添字の上限を 4、第 2 次元の添字の上限を 5 とする。

例 DIM A(2 TO 10)

配列 A の次元を 1 とし、添字の下限を 2、上限を 10 とする。

OPTION BASE 0: 配列宣言で添字の下限の指定を省略したときの値を 0 にする。

OPTION BASE 文を書くときは、最初の DIM 文より先に書かなければならない。