
Tiny Basic for Windows の仕様概略

tbasic.org ^{*1}

[2013 年 10 月版]

Tiny Basic for Windows の現在の仕様について簡単に纏めます。以下は、Ver.1.2 の仕様です。将来変更する可能性があります。演算子・文・関数等については、Help 等の別項を参照してください。

1 概略

[動作環境]

Windows 8, Windows 7 の環境で動作確認を行っています。また、Tiny Basic for Windows は 32 ビットアプリケーションですが、64 ビット環境でも動作します。

[使用メモリー]

TBasic.exe 本体は 1.6 M 程の大きさです。動作には 10M バイト以上のメモリーが必要です。もはや Tiny とは言えないファイルサイズ、メモリーを必要としますが、現在使用されている標準的 PC であれば、問題なく動作します。また、いくつかの Tiny Basic を並行的に起動することもできます。

大きなプログラムや大きな配列変数を使用する場合、その量に応じたメモリーが必要です。逆に言えば大きなメモリーを搭載した PC では大きな配列変数を利用可能です。

[変数名の長さ]

変数名は英字から始まり英数字からなる半角 30 文字以下の文字列から構成されます。大文字小文字の区別はありません。例えば、ABC と aBc は同じ変数を表します。アンダースコア _ は先頭には使えませんが、変数名の中で使えます。

[使用可能配列変数の個数]

変数、配列変数はメモリーの許す範囲で使うことができます。Dim で宣言してエラーが出ない範囲で可能です。

[配列変数の最大次元]

5 次元。例えば、Dim A(2,2,2,2,2) といった宣言が可能です。

[プログラム]

プログラムは行の並びによって構成されます。行は、いくつかの文によって構成されます。文は行頭から始まるのが基本ですが、区切り文字コロン : によって区切られた後に文をつづけることも出来ます。

[プログラムの拡張子]

プログラムファイルの拡張子は tbt (tiny basic text) です。この拡張子は関連付けのために使われるもので、動作には関係しません。例えば、拡張子が txt でもプログラムファイルとして使用可能です。

*1 <http://www.tbasic.org>

[プログラムで使用可能な文字]

プログラムはテキストファイルです。内部的には Unicode で処理されます。キーワードはすべて、アスキーコード 32 ~ 126 の範囲です。この範囲の文字でも使用できないものもあります。コメント文や文字列としては Unicode 文字が使えます。但し日本漢字や欧米系文字以外の文字は編集画面・実行画面での表示が乱れることがあります。これは、利用しているライブラリの対応に依存します。

[エンコーディング]

プログラムは内部では UTF16 で扱われています。ファイルの読み込み、書き出しは標準的なもので可能です。S-JIS が基本ですが、EUC や JIS でも可能です。Unicode 特有の文字を含む場合、Unicode 系のエンコーディングで保存する必要があります。

[使用可能プログラムのサイズ]

最大 1M バイト、プログラムは内部的には UTF16 で処理されていますので、殆んど 1 文字 2 バイトを使用しています。

[プログラムの 1 行の長さ]

最大 1024 文字。Unicode で処理されますので全角半角の区別なく 1024 文字です。

[実行画面の最大サイズ]

実行画面用のメモリーとして起動時に 5M バイト確保されます。このメモリーの範囲で、最大 1000000 行表示されます。メモリーを超えるとスクロールします。スクロールすると動作が遅くなりますので、適宜、CLS 命令で画面を消去して使うことを推奨します。

[扱う型]

Tiny Basic for Windows で扱う型は次の 3 種類です。それぞれ変数と定数があります。

数値型 : 10 バイト浮動小数点で、約 19 桁の有効桁数。

文字列型 : メモリーに余裕があれば最大 2^{30} 文字まで可能です。

ただし、print # で一度に書き込める最大は 2048 文字です。

論理型 : 真理値 (真, 偽) を保持します。

[論理定数]

論理定数は論理型の定数です。True と False があります。True は真, False は偽を表します。

[使用可能グラフ画面]

1 面, 最大 3000 × 2500 ピクセル (使用法については, GSreen の項目を参照して下さい。)

[コントロール画面]

1 面, ボタンやラベル, テキストボックスが使えます。各々 40 個使えます。

[同時使用可能ファイル数]

ファイル番号 : ファイル番号は #1 ~ #8 まで, 合計で 8 個使えます。

[ファイルの読み書き]

ファイル番号を使ってファイルを読み書きする方法は旧式な方法です。標準的には

`ReadAllLines`, `ReadAllText`, `WriteAllLines`, `WriteAllText`

の使用を薦めます。

2 プログラムの構成

プログラムは主プログラム、副プログラムによって構成されます。主プログラムは実行部分のプログラムで、必ず必要です。副プログラムは **Sub** 及び **Function** で構成される部分です。

主プログラム及び副プログラムをそれぞれ、プログラムブロックということにします。Tiny Basic for Windows では外部・内部副プログラムの区別はありません。主プログラム、副プログラムの位置は特に決りはありませんが、

```
主プログラム
副プログラム
. . .
副プログラム
```

の順に置くことを推奨します。また、主プログラムの最後は **End** を記述することを推奨します。プログラムは実行されると、主プログラムが実行されます。副プログラムは主プログラムから呼び出されて使うもので、副プログラムのみでは、何の実行もしません。

3 変数

3.1 変数宣言

Tiny Basic for Windows では、変数は特に宣言をしなくても使うことができます。その場合、普通の場合は、数値変数、そして変数名の最後の文字が\$の場合は文字列変数となります。

配列変数を使う場合は、添え字の最大数を指定して宣言をする必要があります。宣言は **Dim** と **Public** を使って宣言します。Dim は主プログラム、副プログラムいずれでも使うことが出来ませんが、Public は主プログラムのみで使えます。

また、**Function** と **Sub** の名前や **Function** と **Sub** での引数は宣言と考えます。宣言によって使える型は
実数型 : **as Real** の形

文字列型 : **as String** の形

論理型 : **as Boolean** の形

です。このうち、実数型、文字列型は宣言をしなくても使えますが、論理型を使う場合は宣言が必要です。

3.2 変数の適用範囲

Tiny Basic for Windows の変数は適用範囲から分類すると次の5つの型があります。

- 特に宣言しないで使う変数
- Public 宣言して使う変数
- Dim 宣言して使う変数
- Sub や Function の名前
- Sub や Function で引数として使われる変数

(1) 特に宣言しないで使う変数

この変数は、各プログラムブロックの中で有効です。例えば、主プログラムで使われた、変数 A と副プログラム 1 で使われた変数 A は別の変数として扱われます。

(2) Public 宣言して使う変数

Public 宣言は主プログラムでのみ使えます。この変数は主プログラム、副プログラム全てで共通な変数になります。

(3) Dim 宣言して使う変数

Dim 宣言は主・副プログラムいずれでも使えますが、Dim 宣言された変数は、主プログラム・副プログラム中だけで有効なものです。Public 宣言された同名の変数があっても、この変数は宣言された、副プログラム内だけで有効なものとしします。

副プログラムで Dim 宣言された変数は、主プログラムで Public 宣言された変数と同名変数があっても別の変数として扱われます。即ち、Dim 宣言が優先されます。

(4) Sub や Function の名前

Sub や Function の名前は変数とは少し違いますが、Public 宣言された変数と同様の扱いになります。ですから、Sub や Function の名前として使われたものは、プログラムどこからでも引用することが出来ます。つまり、Sub や Function として定義されたものはどこでも使用することが出来ます。このことは、自己引用（再帰）的使用も可能なことを意味します。

(5) Sub や Function で引数として使われる変数

引数として使われる変数は、特別な宣言はしませんが、常に副プログラムの中で Dim 宣言した変数と同じ動作をします。