
Tiny Basic for Windows 操作入門

tbasic.org *1

[2014 年 4 月版]

ここでは、Tiny Basic for Windows をインストールした後、実際にどのように操作をするかについて説明します。

目次

1	例プログラムの実行	2
1.1	tbasic の起動	2
1.2	プログラムを開く	2
1.3	プログラムの実行	5
1.4	プログラムを閉じる	6
1.5	色々な例プログラムを実行してみよう	9
2	ダイレクトモードの実行	9
2.1	ダイレクトモード	9
2.2	tbasic の実行画面	10
2.3	Direct.tbt の実行	12
2.4	プログラムはダイレクトモードの進化形	13
3	エディター操作	14
3.1	エディターとワープロ	14
3.2	エディターでの一般的操作	15
3.3	tbasic 内蔵エディターの特徴	16
3.4	実行画面と編集画面の違い	18
4	tbasic でのプログラムの実行	18
4.1	Hello Basic!	19
4.2	本格的な入力に挑戦	24
4.3	最後に	25

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

1 例プログラムの実行

Tiny Basic Set には例としていくつかのプログラムが同梱されています。この例プログラムは 主に Tiny Basic for Windows（以下では tbasic と表します。）の機能を紹介するもので、必ずしも入門的でないものも含まれています。しかし、それらを実行することで、 tbasic の雰囲気を知ることが出来ます。それらを実行して tbasic の世界を体験して下さい。

プログラムの実行から終了までの手順は次の通りです。

■ 例プログラムの実行の手順

- (1) tbasic の起動
- (2) プログラムを開く
- (3) プログラムの実行
- (4) プログラムを閉じる

1.1 tbasic の起動

tbasic はそのアイコンをダブルクリックすることで起動します。tbasic の起動と終了については、Tiny Basic for Windows 導入編の「Tiny Basic for Windows のインストール」の中で説明されています。必要ならそちらを参照して下さい。

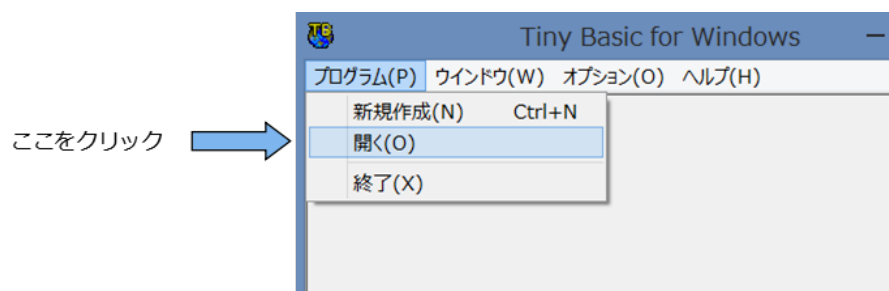
1.2 プログラムを開く

tbasic を起動したら、目的のプログラムを開く必要があります。

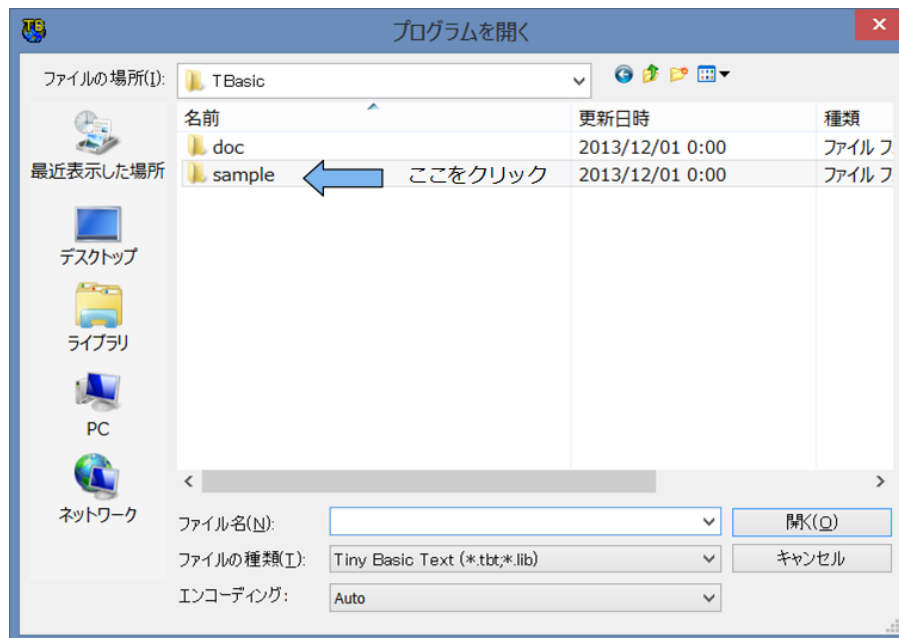
プログラムを開く（Open する）とは、
プログラムを実行可能なように、コンピュータのメモリの中に読み込むこと

ワープロで文書ファイルを開く（Open する）と言うと、文書をワープロの中に読み込み、利用可能にすることでした。tbasic の場合、プログラムを開く（Open する）とは、プログラムを tbasic エディターに読み込み、編集したり、実行したりできる状況にすると言います。

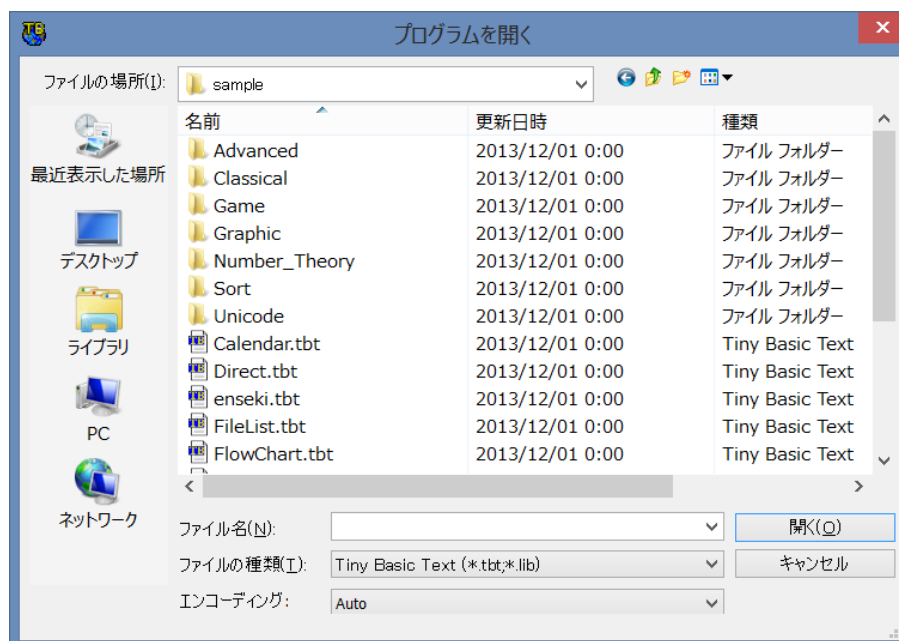
では実際に、プログラムを開いてみましょう。メニューバーでプログラム (P)-開く (O) をクリックします。



そうすると、標準的にインストールした場合、「プログラムを開く」あるいは「Open a Program」というウインドウが現われ、次のようになります。環境設定をした場合は違う場所になることもありますので、その場合は Sample フォルダのある場所に移動してください。



例プログラムは Sample フォルダに入っていますから、Sample のフォルダアイコンをクリックして、Sample フォルダの中に入ってください。Sample フォルダに入ると次のような画面になります。



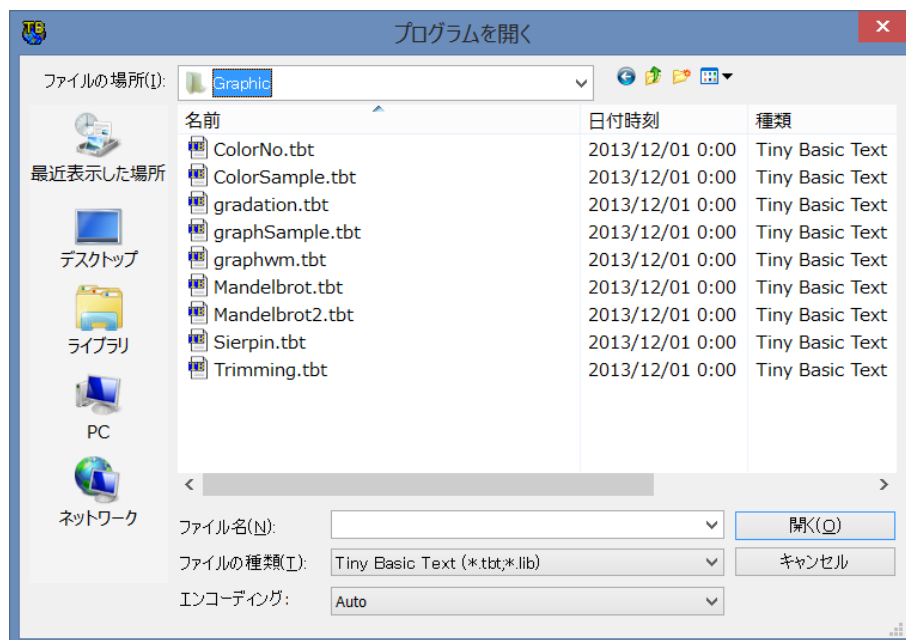
コンピュータの設定によっては、拡張子 tbt が表示されない場合もあります。

Sample フォルダには更に、7個*2のサブフォルダがあります。サブフォルダにはそれぞれ

- Advanced : 少し高度なプログラム例
- Classical: Basic 初期のプログラムの例
- Game: ゲームの例
- Graphic: グラフ関係のプログラム例
- Number-Theory: 整数論のプログラム
- Sort: 整列法のプログラム
- Unicode: ユニコード関係のプログラム

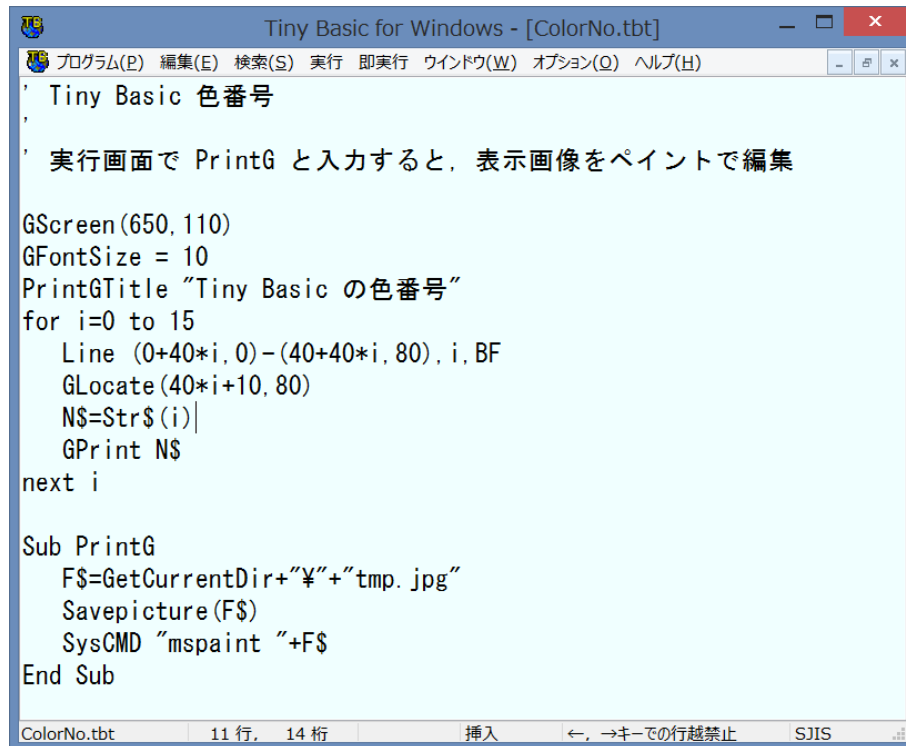
が収められています。その他のプログラムが Sample フォルダ直下に収められています。

例えば、Graphic サブフォルダをダブルクリックすると、



となります。このうち例えば、一番上の ColorNo.tbt をダブルクリックすると、ColorNo.tbt が Open され、次のようになります。

*2 バージョンによっては違うこともあります。



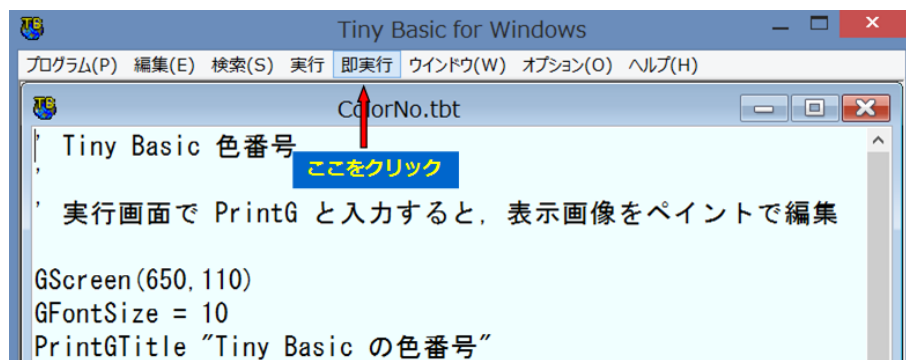
これで、ColorNo.tbt が開かれ、実行可能な状態になりました。この画面ではこのプログラムを編集・変更したりすることが出来ます。

1.3 プログラムの実行

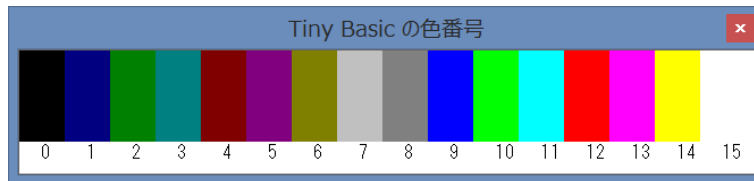
プログラムを実行するとは、
エディター画面に開かれプログラムを実行すること

開いたプログラムを実行するのは、簡単です。メニュー画面にある「即実行」アイコンをクリックして下さい。それだけです。

例えば、上のように ColorNo.tbt が開かれていたときに、「即実行」アイコンをクリックしてみましょう。



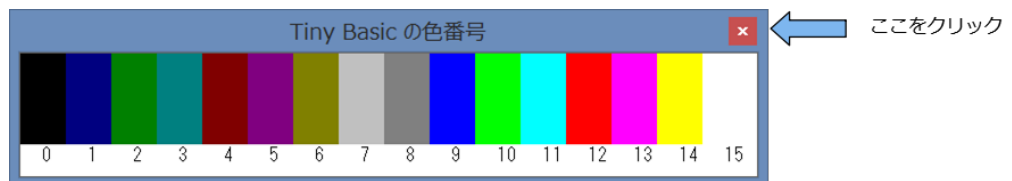
そうすると、プログラムが直ちに実行され、結果が表示されます。このプログラムは tbasic での色番号の例を表示するもので、グラフ画面が現われ、次のような結果が表示されます。



ここで tbasic でグラフ画面の扱いについて少し説明します。グラフ画面はプログラムの中で表示・非表示を制御することが出来ます。このプログラムの場合、上の大きさの画面を表示させ、それに上のような結果を描きました。プログラムの中で、この画面を非表示（画面全体を消してしまう）ことも可能ですが、結果を見るために、消さないのが普通でしょう。

さらにまた別のグラフを描いたとき、前の画面と比較したい場合などもあるでしょう。tbasic で扱うことの出来るグラフ画面は一度に一つだけですが、表示したグラフ画面は特に指定しない限り、次々と残っていき、適宜利用できます。これは便利な機能ですが、他方、不必要になったグラフ画面は適宜消去（閉じる）していく必要があります。

グラフ画面の右上端にある「×」をクリックすると、グラフ画面が閉じます。ですから、今の場合



とすればグラフ画面が消去されます。

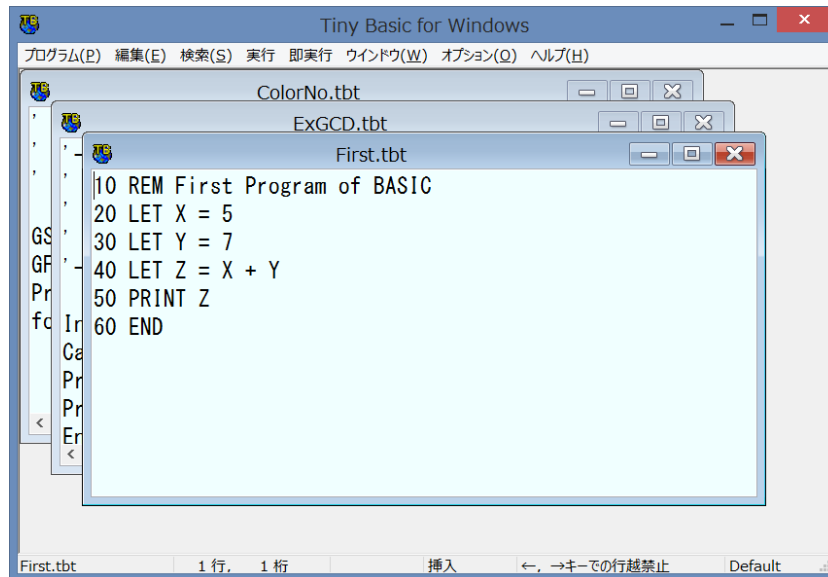
多くのグラフ画面を同時に表示するとメモリの関係からコンピュータの動作が不安定になることもあるかもしれません。必要なくなったグラフ画面は適宜消去して下さい。勿論 tbasic を終了させればすべてのグラフ画面は自動的に消去されます。

1.4 プログラムを閉じる

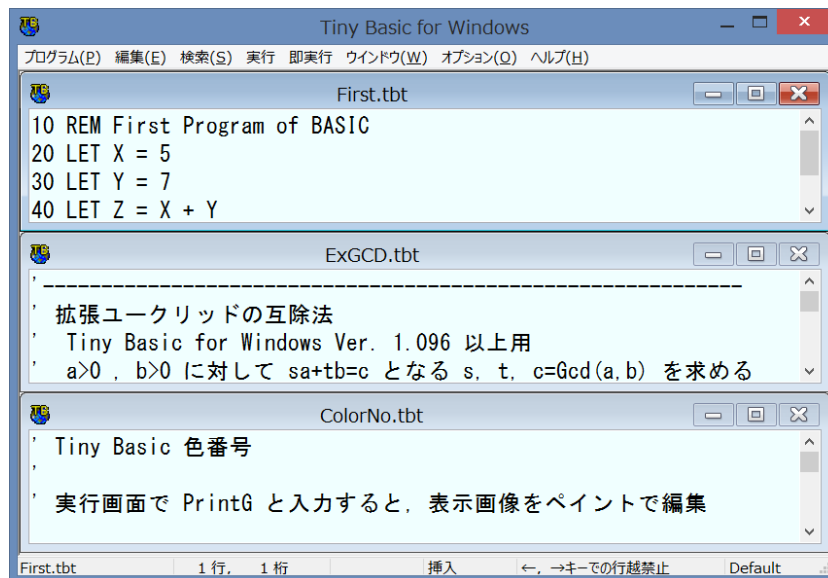
プログラムを閉じるとは、
エディター画面に開かれプログラムの使用を終了して、消すこと

別のプログラムを開いて、実行することを考えましょう。上に述べた方法に従って、別のファイルを開いて実行することも出来ます。tbasic のエディターはマルチウインドウですから、いくつでも同時にファイルを開き、使用することも出来ます。

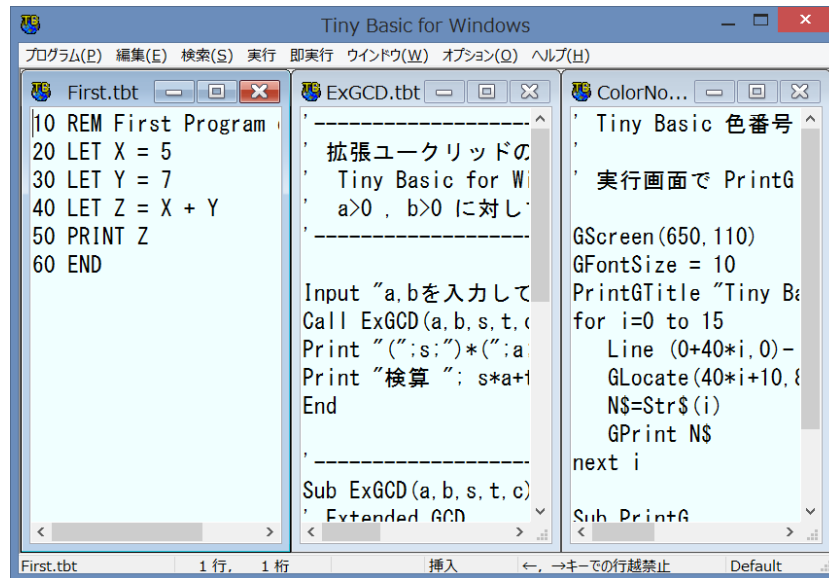
例えば、次の画面は3個のファイルを同時に開いた時の画面です。



これはカスケードと言われる表示の仕方です。タイルと言われる次のような表示も出来ます。



横並びタイル



縦並びタイル

勿論、これらのうち一つのファイルだけ全画面にしたり、色々な大きさにしたり、また tbasic の編集画面自身の大きさを色々変えることも出来ます。

しかし、現実問題として、多くのプログラムを同時に開いて置くのは好ましくありません。不必要になったものは適宜閉じる（Close する）必要があります。

プログラムを閉じる（Close する）とは、
コンピュータのメモリからそのプログラムを消去すること

新たに作ったプログラムや、変更を行ったプログラムは、「閉じる」前に保存するのが普通です。しかし、今の場合は既にあるプログラムを実行しただけですから、保存の必要はありません。この保存については「エディターの使用法」の項で説明します。

プログラムを閉じるのは簡単です、プログラムが表示されているウインドウの右上端にある×アイコンをクリックするだけです。（tbasic の画面全体の右上端にある×アイコンではありません。この×アイコンをクリックすると tbasic が終了してしまいます。）

プログラムを閉じるには
右上端にある×アイコンをクリックする

1.5 色々な例プログラムを実行してみよう

Sample フォルダには他にも色々なプログラムがありますので、実行してみてください。操作が判らなくなったら、キャンセルや終了をしてやり直しても構いません。

一般に、コンピュータのソフトウェアでは余程の特別な操作をしない限り（ファイルの削除コマンドを実行することを別にして）、基本的にコンピュータを壊すことはありません。色々と試してみてください。

色々な例プログラムを実行して、
試してみよう

2 ダイレクトモードの実行

2.1 ダイレクトモード

ダイレクトモードとは
プログラムを書かないで、入力した命令をその都度、
直ちに実行するモード（状態）です。

N88-BASIC ではダイレクトモードは最も基本的なモードでしたが、Windows 上のもの、あるいは移植された Basic には何故かダイレクトモードが無いものが多いようです。ダイレクトモードはちょっとした使い方には中々便利なので、Tiny Basic for Windows でもサポートしています。tbasic の一つの特徴かもしれません。

しかし、コンピュータの世界でこのような使い方が特殊というわけではありません。Windows には Dos 窓というのがありますが、これはまさにダイレクトモードです。このような実行形態は、Linux や Mac 上のソフトウェアでもよくある形態です。tbasic のダイレクトモードはそれの BASIC 版ともいえます。

tbasic のダイレクトモードは Dos 窓の Basic 版

使い方は非常に簡単で、いわゆる電卓的な使い方です。簡単な計算であれば、プログラムを書かないで実行したほうが早くて、便利なこともあります。ダイレクトモードはそういう使い方を主としたものです。Dos 窓では計算は出来ませんが、tbasic のダイレクトモードではかなりの計算が出来ます。

N88-Basic のダイレクトモードには無い機能が tbasic のダイレクトモードにはあります。それは、ヒストリ機能です。これは現在の Dos 窓や Unix 系のシェルでは標準的なものですが、tbasic ダイレクトモードもこれを少しだけ真似してサポートしています。このダイレクトモードのシェルの機能の向上は魅力的ですが、実際は余り実装されていません。

■ ダイレクトモード実行の手順

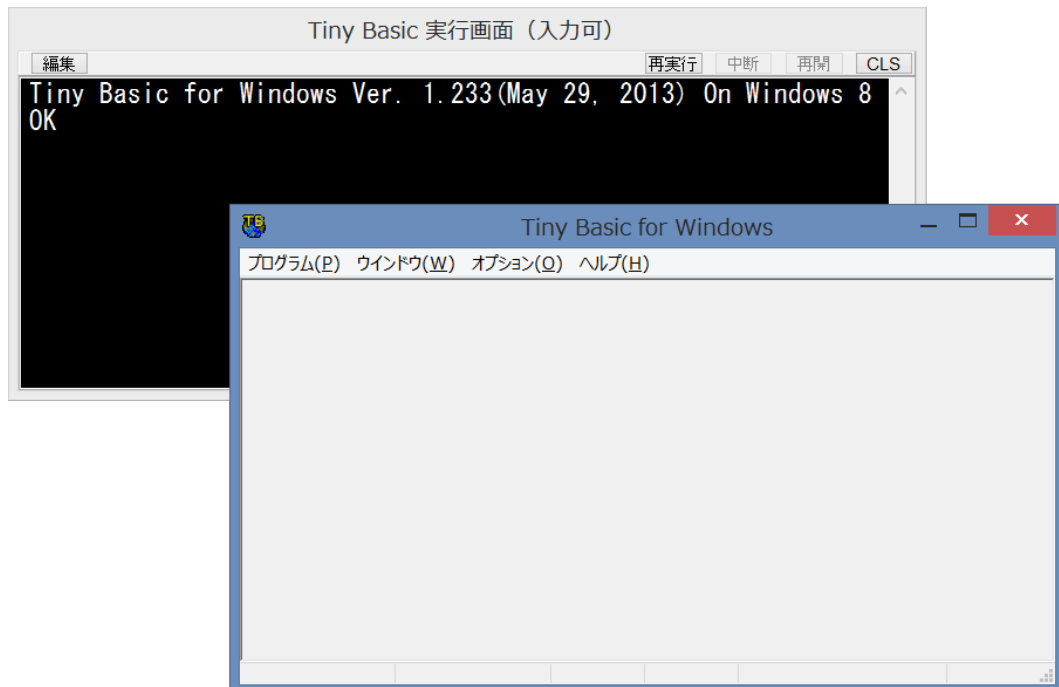
ダイレクトモード実行の手順は以下の通りです。

- (1) tbasic の起動
- (2) 実行画面のどこかをクリック
- (3) コマンドの入力

これだけです。

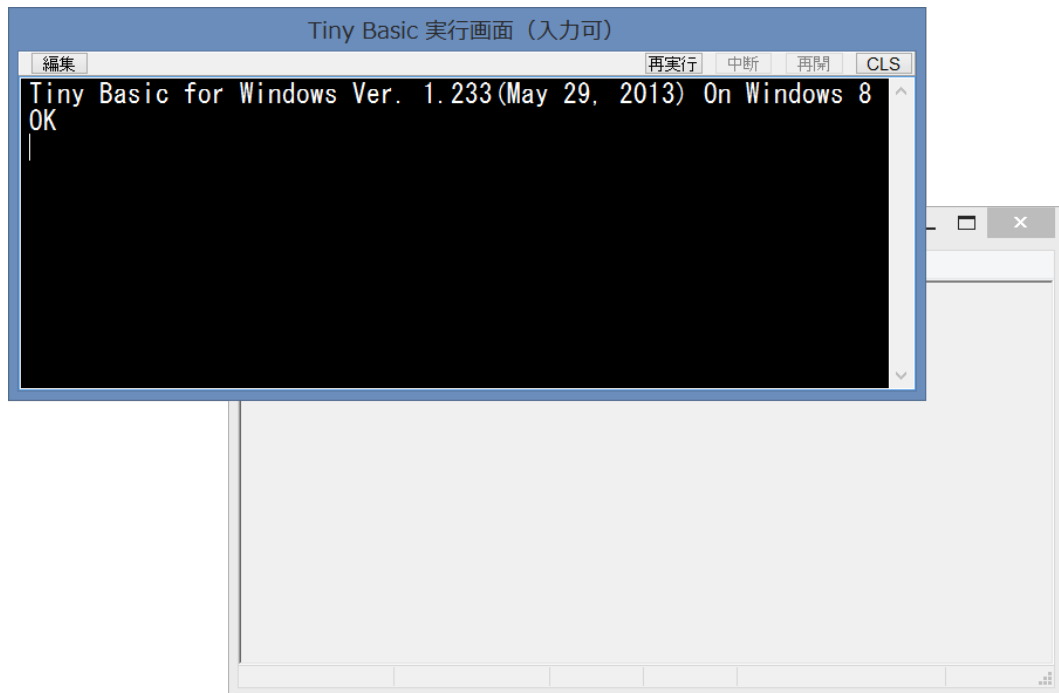
2.2 tbasic の実行画面

tbasic の起動直後の画面は下のような感じです。後左上にあるタイトルバーに「Tiny Basic 実行画面（入力可）」と表示されている黒い画面が、実行画面です。

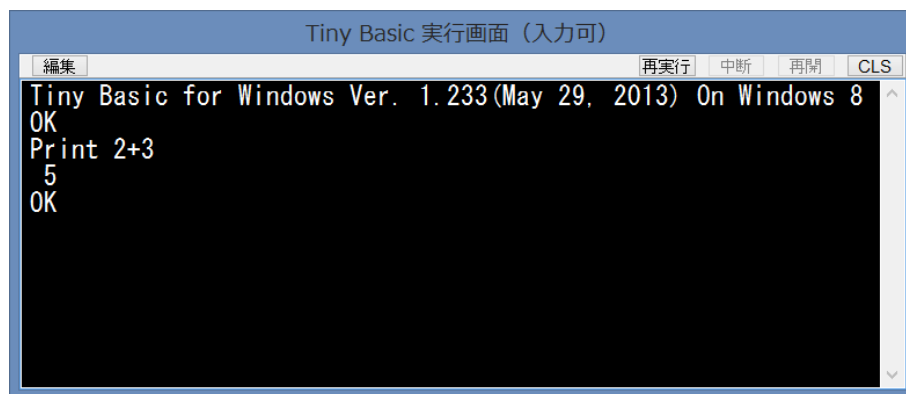


この実行画面は、プログラムなどで計算を行ったとき結果を表示する画面ですが、この画面は結果を表示するだけでなく、コマンドを直接入力して実行することも出来ます。この画面で直接コマンドを入力すると直ちに実行され、その結果が表示されます。このモードを**ダイレクト（直接）モード**と言います。

この実行画面のどこかをマウスでクリックして下さい。すると、タイトルバーの色が変わると同時に画面が前面になります。



これで、ダイレクトモードが実行可能になりました。ここで例えば、キーボードから
`print 2+3`
と入力して、Enter キーを押して下さい。すると



とその結果が表示されます。

このように、`print` を入力することを除けば電卓を使っているようです。ダイレクトモードの使い方については、自習用のプログラム `Deirect.tbt` が `Sample` の中にあります。それを実行することで、ダイレクトモードの感覚をつかむことができます。

■ 重要な注意：Basic では大文字小文字は関係ない

コマンドの入力で一つ重要な注意があります。Basic では原則（実は殆どすべて）として、大文字小文字の区別は重要ではありません。ですから、

Print

PRINT

pRinT

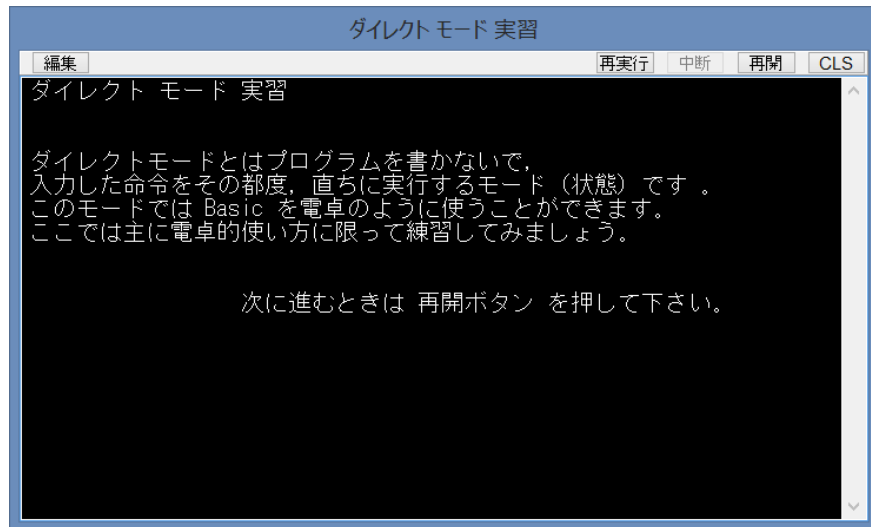
は全部全く同じ意味です。勿論普通は分かりやすいために、Print または print とします。

2.3 Direct.tbt の実行

Sample フォルダの中に Direct.tbt あるいはコンピュータの設定では Direct というプログラムがあります。それを開いて実行してみてください。下は Direct.tbt を開いた直後の編集画面です。



ここで「即実行」ボタンをクリックすると、プログラムが実行されます。すると実行画面は



となります。後はプログラムの指示に従って進んで下さい。このプログラムを実行すれば、ダイレクトモードの大体の感じはつかめます。これで殆どダイレクトモード修了です。

■ ヒント

ヒストリ機能について説明します。

ヒストリ機能は
一度入力したものを極力利用し、入力の手間を減らそうと言うもの

実行画面で、いくつかのコマンドを実行した後、↑キーを押してみてください。すると直前に入力したコマンドが表示されます。このコマンドを←キーや BackSpace キー Del キーなどを使って適当に変更して、Enter キーを押すことそのコマンドが実行されます。似たようなコマンドを何度も入力する場合随分と楽です。

■ もう一つヒント

もう一つ、Print コマンドの入力が手間と考える人もいるかもしれません。実は、

Print の代わりに ? が使えます。

例えば、

? 1+2

は

Print 1+2

と同じです。これを利用すると随分と楽です。

2.4 プログラムはダイレクトモードの進化形

ダイレクトモードとプログラムモードの関係を少しだけ説明します。

プログラムを書いて実行するモードをプログラムモードと言います。プログラムモードは基本的にダイレクトモードで行うことを予め指示してまとめて行うと理解すると良いでしょう。勿論、プログラムでは色々と複雑なことが出来るような仕組みになっていて、かなり進化してます。しかし、ダイレクトモードでのコマンドを順次並べても、一応はプログラムになります。ですから、

プログラムはダイレクトモードの進化した自動実行指示書である

といえるでしょう。

3 エディター操作

tbasic に限らずプログラミングを行うときは、普通エディターでプログラムを書きます。どのような言語システムでもほとんどの場合、どのようなエディターを使っても構いません。しかし、最近のプログラミングはその言語のためのエディターと実行環境が一体化していることが多くなっています。この一体化されている状況を統合環境と言います。

tbasic にもエディターが内蔵されていて、それを使ってプログラムを作り、実行するのが普通です。勿論 tbasic のプログラムは単なるテキストファイルですから、理屈の上では、別の使い慣れたエディターを使ってプログラムを作成しても構いません。しかし、それでは実際問題としてはかえって使いづらいので、tbasic 内蔵のエディターを使うことを勧めます。

tbasic のエディターはいくつかの特殊機能を持っていますが、全体としては Windows 上の普通のエディターです。特別な操作法はそれほどありませんので、既にエディターを使った経験のある人はすぐに使えます。

以下では、ワープロは使ったことがあるが、エディターは余り使ったことが無い人を想定して説明をします。

3.1 エディターとワープロ

エディターは文書を作成するソフトウェアです。文書を作成するソフトウェアとしてはワープロが良く使われます。コンピュータを使う人で、ワープロを使ったことの無い人は恐らくいないでしょう。これに対して、エディターを使う人は割合からすればかなり少ないでしょう。

では、ワープロとエディターの違いは何でしょうか。それは目的の違いです。

- ワープロは見栄えの良い、主に他人に見せるための文書を作成します。
- エディターは文書の作成を主目的とした文書を作成します。

もう少し、具体的に説明しましょう。

- ワープロでは文字の大きさ、種類、配置、画像、色など色々な装飾技法を使って、見栄えの良い文書を作成します。
- これに対して、エディターでは、使えるのは一種類の文字体系で大きさもすべて同じです。

勿論、ワープロでエディターで書いたような文書を作成することは出来ます。ですから

ワープロ = エディター + 色々な文書装飾技法

とも言えます。

それなら、何も2種類のソフトウェアを使わずにワープロですべて済ませば良いと思うかもしれませんが。しかし、実際はそうではありません。ワープロはその高機能化を追求した結果、現在普通に使われているワープロは大変大きなシステムになってしまいました。そのためワープロをインストールするためのディスク容量はかなり必要ですが、エディターはそうでもありません。

また、起動に要する時間や、文書内でのカーソル移動の速さなど、ワープロとエディターでは大変違います。つまり、

ワープロは重く、遅い。これに対してエディターは軽く、速い

ということです。

プログラムはコンピュータに対する指示文書ですから、色々な装飾は必要ありません。ですから、軽快に動くエディターの方が適しているわけです。また言語の統合環境に内蔵させる為にも適しています。勿論エディター文書も我々人間が読むことも多いわけですから、見栄えとは別に読みやすく書くという工夫は必要です。そしてそのような工夫の方法もあります。

3.2 エディターでの一般的操作

tbasic 内蔵のエディターはいくつかの機能を除けば普通のエディターですので、その操作法は特別なものではありません。その使用方法について詳しく説明するのは、このページの趣旨に少し外れますので、簡単な注意的なものをまとめます。

エディターに限りませんが、コンピュータでの文字入力に慣れた人でも中々大変です。それを克服するには2つの方向が考えられます。

■ 文字入力の上達法

- (1) キー入力の練習をしっかりとる。
- (2) エディターの機能を最大限利用する。

(1) については、恐らく誰でも一度はタイピングソフトで練習をしたことがあるでしょう。素早く正確にキー入力出来ることは素晴らしいことです。

でもこれだけでは、限度があります。実は (2) の方法の活用が場合によっては強力で、そしてその場合が割合あるのです。実は私がコンピュータ入力でモットーとしていることがあります。それは、「なるべくキー入力をしないで済ませる。」ということ、つまり、

一度入力したものは極力利用する

ということです。

このためのエディターの機能としては

- 複写 (コピー)
- 切り取り (カット)
- 貼り付け (ペースト)

があります。この他に、エディターの重要な機能として、

- 検索
- 置換

などが、あります。これらを利用するための操作として、「**範囲指定**」と言ったこともあります。

これらのことを一通り練習して、身に付けて下さい。以上のことはワープロを良く使う人は既に知っていることでしょう。実際、ワープロはエディターの進化形ですから、エディターの機能は殆どワープロにあります。ですから、エディターの使い方を学ぶことは、ワープロの使い方の勉強にもなります。

ですから、もし上の機能に慣れていなかったら、是非、身に付けて下さい。

これらのことの説明はここではしませんが、この操作法の説明は恐らくどのワープロの説明書にあります。また、Windows に標準で附属しているメモ帳 (notepad.exe) を起動して、そのヘルプを見るのも良いでしょう。メモ帳は Windows のエディターの原点のようなもので、一度起動して少し練習をしてみる価値はあります。

3.3 tbasic 内蔵エディターの特徴

tbasic 内蔵エディター特有の機能について2つだけ説明します。実際の操作法は、プログラムの実行法等のところで、説明します。

- マルチウインドウです。

メモ帳は違いますが、これは 現在のエディターは殆どそうです。

- いくつかのオプション設定で、キーワードなどを強調したり、色を変えたりすることができます。

具体的な例をひとつあげます。次は、常用対数の表を表示するプログラムです。


```

'-----
' 常用対数表 log10(x)
'-----
Print 常用対数表 log10(x)
For i=1 to 100
  Print i/10; Tab(5); \":\"; log10(i/10)
Next i
End

```

このプログラムを見ると、上から3行が緑の色で表示されています。これはコメントと言われる行であることを示しています。また、4行目の"で囲まれた部分が紺色で表示されています。これはこの部分が文字列と言われるものであることを示しています。さらに、5行目の For と to そして、6行目の Next さらに、7行目の End が強調文字で表示されています。これらの強調文字で表された語は、キーワードと言われる BASIC での特別な語であることを示しています。

このように幾つかの部分、強調文字や、色を変えることで、プログラムの意味を分かりやすくすることができます。

更に、似たものですが、次の例を見てみましょう。

```

'-----
' 常用対数表 log10(x)
'-----
Print 常用対数表 log10(x)
For i=1 to 100
  Print i/10; Tab(5); \":\"; log10(i/10)
Next i
End

```

この画面では、上から4行目の「常用対数表」の文字が赤字で表されています。これは、BASIC のプログラムの命令文の中に、日本語が含まれていることを警告しているものです。BASIC のプログラムで日本語が含まれて良いのは、コメント部分と文字列の中身だけで、それ以外だとエラーになります。赤字で表示されているのは、そのエラーの警告です。

このように、プログラムを作成する上で、分かりやすいように表示がされるようになっています。

もちろん、この設定が煩わしい場合は、環境設定でこのようにならないようすることも出来ます。

3.4 実行画面と編集画面の違い

実行画面はエディターではありませんので、実行画面で文書を作ることは無いでしょう。でも計算結果が表示されるのは実行画面です。実行画面は、文字の量に依存しますが、最大 1000000 行表示できます。、ダイレクトモードとして、この画面に文字を入力することも出来ます。

実行画面に文字を入力するのは、ダイレクトモードとしての使用以外は想定されていませんが、実行画面のコピーは重要な利用法の一つです。

計算結果を文書に保存したり、ワープロの中で利用したいこともあるでしょう。その場合、プログラムの中でファイルに保存し、そのファイルを利用することも出来ます。しかし、一番簡単なのは単に計算結果を実行画面に表示させ、その結果をコピーして、それを文書に貼り付ける（ペーストする）ことです。

それには、画面を範囲指定して、コピーすれば良いわけですが、この実行画面での範囲指定はマウスでしか出来ないことに注意して下さい。

範囲指定は、キーによるもの（Shift カーソル）が便利ですが、この際利用する↑↓が履歴利用のために確保されています。範囲指定の時だけ外すように実行画面を作ることも原理的には可能なのですが、かえって混乱するので、出来ないようにしてあります。ですから、

実行画面での範囲指定は
マウスのドラッグで

行ってください。

4 tbasic でのプログラムの実行

Sample プログラムの実行の項では、既に出来上がっているプログラムを開き、それを実行する方法について説明しました。tbasic に限らず、言語を使う場合、新たにプログラムを作成し、エディターで入力し、実行し、それを保存するというのも重要な部分です。

ここでは、既にプログラムが出来上がっていたとして、それを入力し、実行し、保存する方法を説明します。勿論、プログラムを作ることがプログラミングの主要な部分ですから、プログラムの作成方法を飛ばして話を進めることに、不満を感じる人もいるかも知れません。

でも、プログラミングの具体的方法についての説明は「初級 Basic」以降の主要な話題です。そしてそこでは余り tbasic に依存しない一般的な BASIC についての話を主にしたいと思っています。ですから、この「tbasic の操作法」の中で tbasic での手順についてを説明をします。

4.1 Hello Basic!

プログラミングの第一歩の例として、よく挙げられるプログラムとして、Hello World というプログラムがあります。ここでもその例に倣ってそれから始めましょう。

この Hello World プログラムは "Hello World" を画面に表示するプログラムのことです。最も簡単なプログラムの、最も有名なものです。ここでは Basic の勉強ですから、"Hello Basic!" にしましょう。

例 4.1. "Hello Basic!" を画面に表示するプログラム。

このプログラムはダイレクトモード実習を行った人には簡単に想像が出来るでしょう。以下がそのプログラムです。簡単です。

```
Print "Hello Basic!"
```

この一行をダイレクトモードで実行すると、



となります。

ダイレクトモードでの入力は tbasic を終了すると、消えてしまい、同じことをしようとすると、また手で入力しなければなりません。このくらいのものであれば、何の問題ありませんが、複雑なものであれば、一度入力したものを活用したいと思うのが自然です。それがプログラムの一つの重要な要素です。一度プログラムを作り、それを保存しておけば、それを実行することで、その内容を、再び入力しないで済みます。

それではこのプログラムを入力して実行し、保存する、手順を説明しましょう。手順は次の通りです。

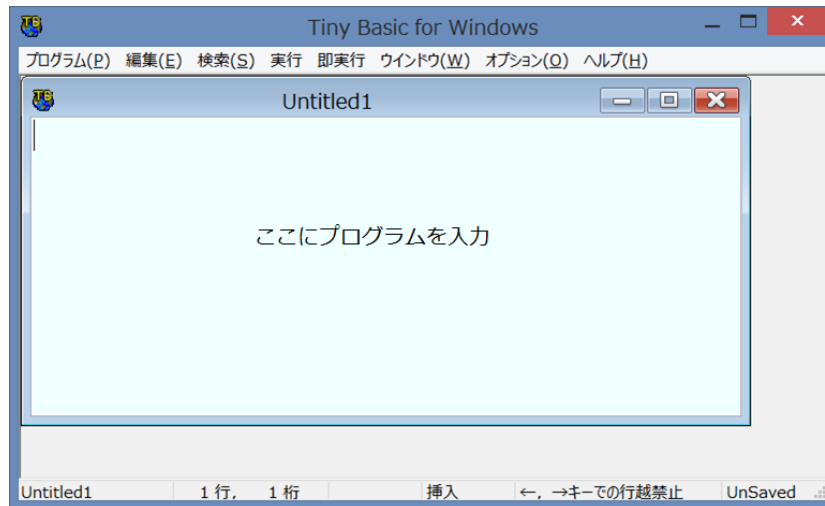
■ プログラム入力・実行・保存の手順

- (1) エディタープログラムの新規作成をクリック
- (2) エディターで実際にプログラムを入力
- (3) 実行キーでプログラムを実行
- (4) そのプログラムをディスクに保存

実際に行ってみましょう。

■ エディタープログラムの新規作成をクリック

tbasic エディターのメニューバーでプログラム（P）→新規作成（N）をクリックします。すると、なにも書かれていなしエディターウインドウが開きます。



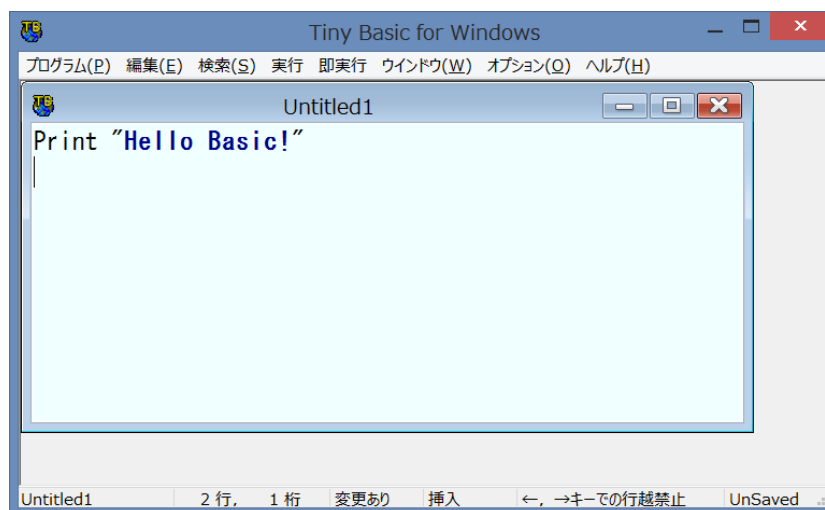
■ プログラムの入力

この白い部分に実際に上のプログラムをキーボードから入力して下さい。簡単ですから、すぐ出来るでしょう。ここで、

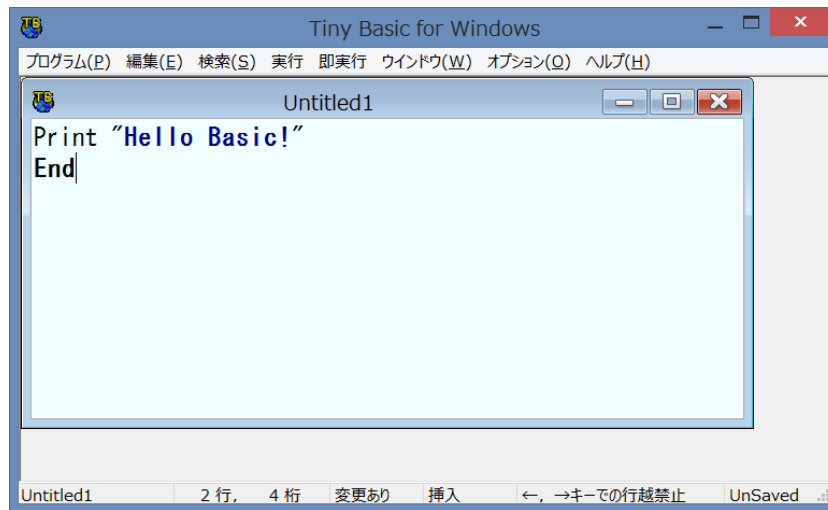
入力はすべて半角直接入力で行う

ことに注意しましょう。

標準設定では、改行をすれば、全角文字があれば、青色で表示されます。ですから、入力後取りあえず改行して、全角文字が無いのを確認するのもよいかもしれません。



こんな感じです。また、プログラムのマナーということで言うと、End 文を書いた方が良いでしょう。



こちらの方が良いです。これでプログラム入力は終わりです。

ダイレクトモードの説明でも言いましたが、ここで Print は Basic のコマンドの一つです。Basic ではコマンドは大文字・小文字は関係ありません。ですから、

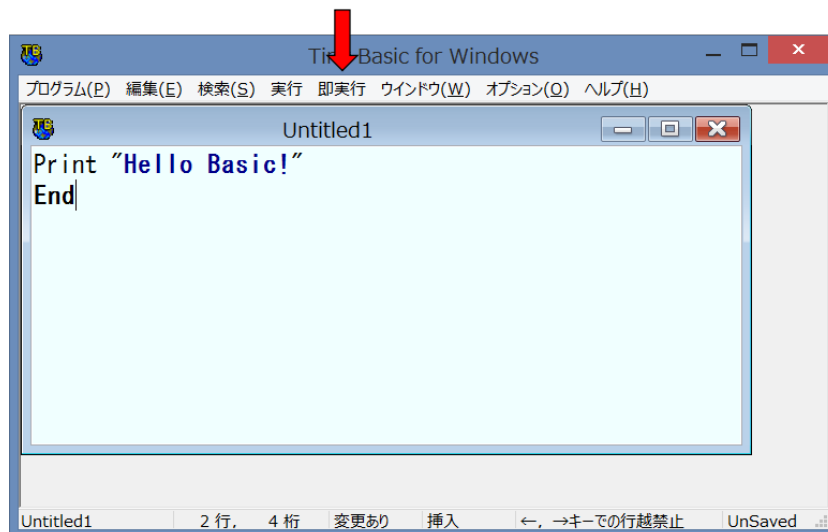
PRINT "Hello Basic!"

でも構いません。しかし、Hello Basic の部分はコマンドでなく、画面表示用の文字列です。ですから、HELLO BASIC! と書くと（特に問題はありませんが、）実行結果が違います。

■ プログラムの実行

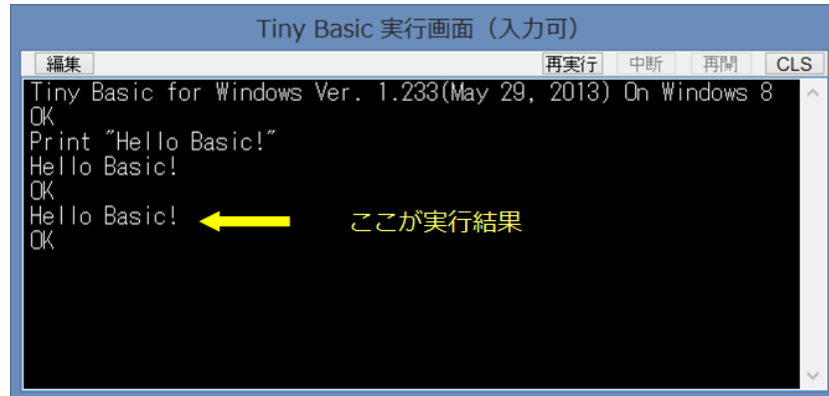
それではこのプログラムを実行してみましょう。それにはメニューバーで即実行を押すだけです。

ここをクリック



■ プログラムの実行結果

これでプログラムが実行され、結果が実行画面に表示されます。



もし、何かエラーでこの通りにならないかもしれません。その場合、正しく入力されていないことが考えられます。エラーの場合、その表示がでますから表示をよく見て、エラーと指摘された、少し前を注意して確認して下さい。

エラーがあった場合、「何行何桁でエラー」と表示されますが、そこはエラーの場所と言うよりは、エラーと確認された位置です。ですから、実際はその少し前にあるのが普通です。

■ プログラムの保存

それではこのプログラムをディスクに保存してみましょう。そのためには、2つ準備が必要です。それは

プログラム保存の手順

- 保存する名前を決める
- 保存する場所を決める

です。

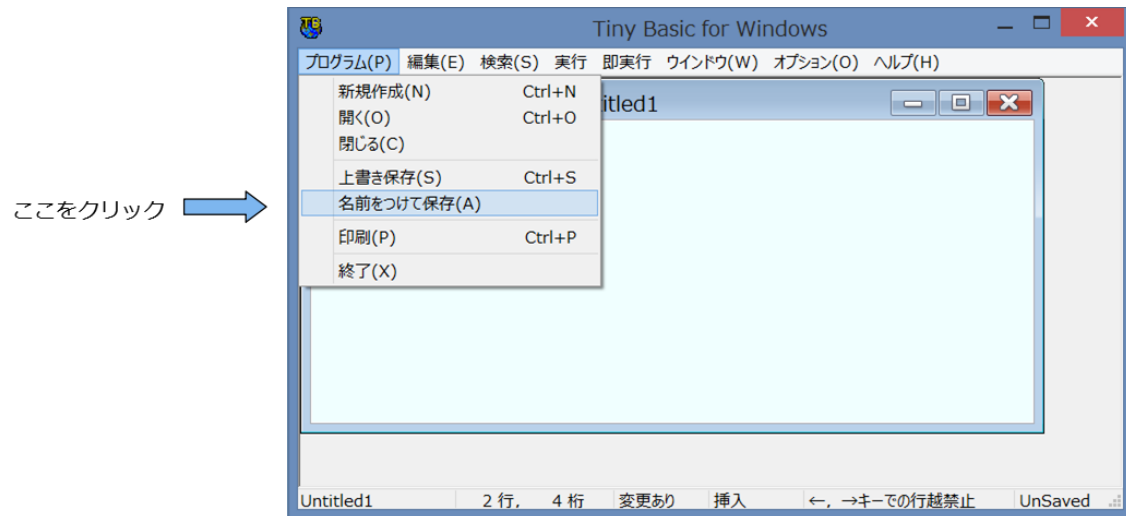
保存する名前は、どのようなものでもよいのですが、ここでは `hello.tbt` という名前にしましょう。ここで、`.tbt` と言うのは拡張子で（正しくは.を除いた `tbt` が拡張子です。）、`tbasic` のプログラムは最後が `.tbt` とするように推奨しています。そして、普通に保存すると自動的に、`.tbt` は付くようになっていますので、実際の名前は、

`hello`

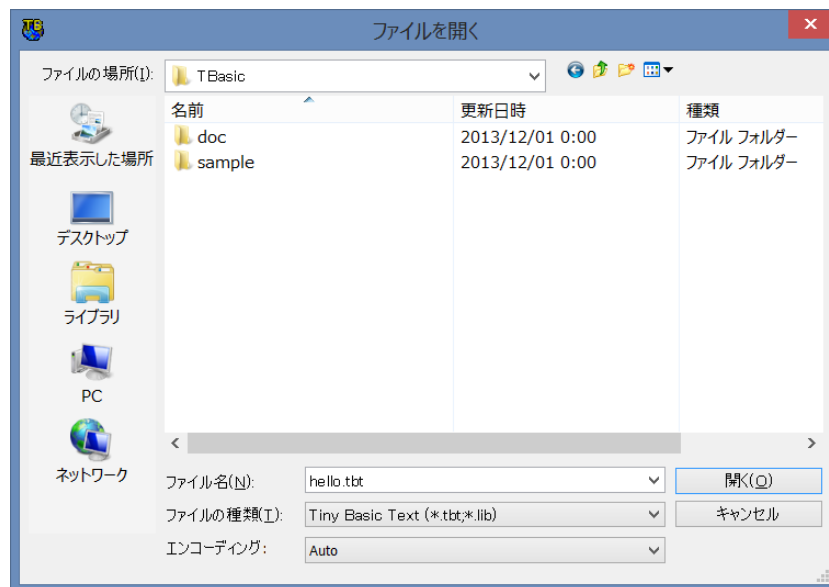
となります。プログラムの名前は漢字も使えますが、最初は半角英数字文字を使うことにしましょう。

次に保存する場所ですが、これもどこでもよいのですが、デフォルトの場所にしましょう。デフォルトの場所とは、特に指定をしないときの場所です。環境設定を行わない場合、`tbasic` をインストールしたフォルダ、これもデフォルトだと `TBasic` フォルダの中になります。

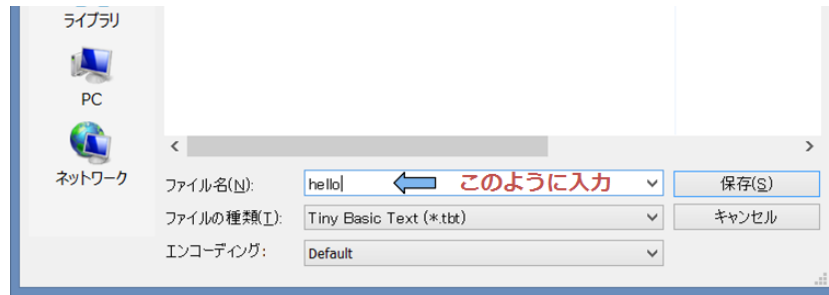
保存するには、メニューバーでプログラム（P）－名前を付けて保存（A）をクリックします。



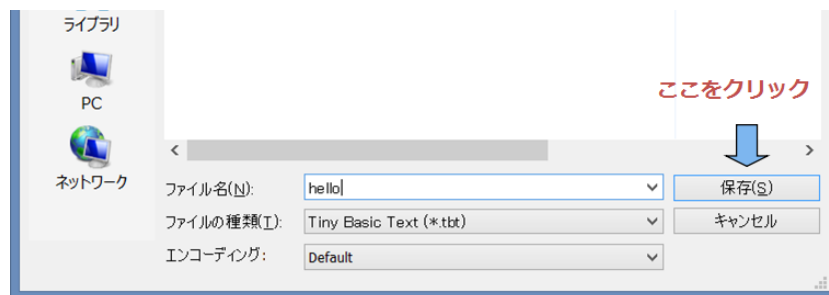
そうすると、



このような画面になりますから、下の部分ファイル名（N）とある入力ボックスの Untitled を目的の名前 hello とします。

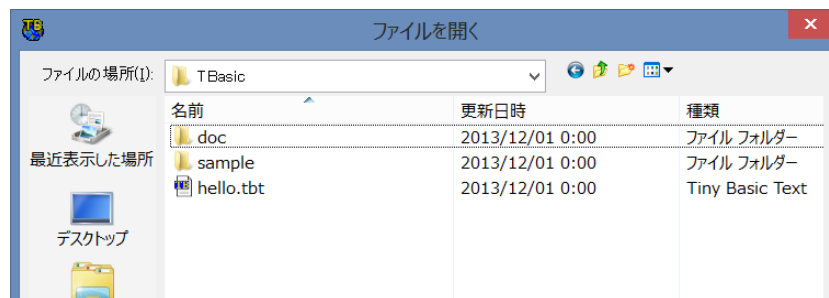


そしてその右にある保存ボタンを押します。



これで保存が終了です。

保存が成功したことを確認するには、メニューバーのプログラム (P) - 開く (O) をクリックして、



と hello.tbt があることが確認できれば成功です。コンピュータの設定によってはファイル名が hello となっている場合もあります。

4.2 本格的な入力に挑戦

例 1 の Hello Basic! は余りにも簡単すぎると思った人もいるかもしれません。そこで少し本格的なプログラムの入力に挑戦してみましょう。

例 4.2. ペテルブルグのパラドックスのシミュレーション

次はペテルブルグのパラドックスと言われる問題をプログラミングしたものです。


```
' ペテルブルグのパラドックス
',
Cls
Price = 20
Gain = 0
MaxCount = 0
NumofGames = 10000
For i=1 to NumofGames
    Randomize
    Count = 0
    Do While 1=1
        Result= Round(Rnd(1))
        if Result = 1 then
            Syoukin = 2^Count
            MaxCount= Max(MaxCount,Count)
            Exit do
        end if
        Count = Count +1
    Loop
    Gain = Gain+ Syoukin - Price
Next i
Print "1回の代金 "; Price;" ドルのとき"
Print "決算金額 "; Gain;"ドル"
Print "最大賞金 ";2^MaxCount;"ドル"
End
```

このペテルブルグ（サンクトペテルブルク）のパラドックスについては色々なホームページでの解説があります。知らなかった人は読んでみてはいかがですか。

4.3 最後に

色々なことに対して言われていますが、

創造は模倣から始まる

ということはプログラミングについても当てはまります。色々なプログラムを見ることはプログラミングの良い勉強です。プログラムをただ入力しただけでは余り勉強にならないと思うかもしれません。しかし、それでもその入力の過程で学ぶことも色々あります。

是非、上のプログラムを入力して、実際に実行してみてください。プログラミングが初めての人にとっては、中々大変なことでしょう。それは恐らくエラーが何度も出て中々巧く動かないと予想されるからです。実は何度もエラーを経験して、そのエラーを発見するのが良い勉強なのです。

今回は、入力ミスのないように入力するという練習ですが、意外と難しいです。

もし、手間を省く方法を思いついて、すぐに上のプログラムを実行できる人は、入力法については修了です。