

F A現場のための VisualBasic 入門 (基本編)

高橋省三【著】

はじめに

本書は、プログラマーになるのではなく、仕事の関係から VisualBasic をちょっとだけ知っておきたい！現場へ行って修正しなければならない！という方のための VisualBasic 入門書です。
と、思いながら書き始めましたが、だんだんと範囲が広がりました。

単に文法の説明ではなく、VisualBasic 全体の概要と一連の操作を PLC~PC 間通信を実例に取り上げながら説明を進めていきます。

- ・ 極力、ページ数を少なくしています
- ・ 画面 Copy 数を多くし、視覚からわかるようにしています

本書[基本編]は以下の 3 章から構成されています。

- 第 1 章 Windows プログラミングと VisualBasic
- 第 2 章 VisualBasic のコントロール
- 第 3 章 プログラミングの基礎

本書のシリーズとして

- 「RS232C 編」 シリアル通信の基本と計算機リンク・上位リンク・パソコンリンクを紹介
- 「Ethernet 編」 TCP/IP の基本と各社 Ethernet による通信を紹介
- 「Web 編」 IIS 等の Web サーバと Web アプリを紹介
- 「ActiveX 編」 ActiveX の作成方法を紹介
- 「.NET 編」 VisualBasic6.0 と .NET の違いを紹介

を計画していますが、なにぶん余暇を利用したの執筆ですから……。完成しましたら、そちらの方もぜひご覧ください。

2001 年 11 月
高橋省三

P.S

実は、本ファイルはパソコンの中で眠っていました。眠っている間に.NET へと……。

本文中、VisualBasic,Windows98,WindowsNT は米国 Microsoft 社の登録商標です。
その他、本文中にあらわれる会社名、製品名は各社の商標、登録商標です。

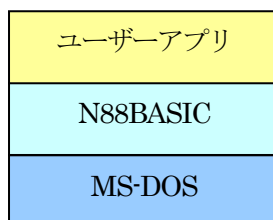
第1章 Windows プログラミングと VisualBasic

1-1 Windows アプリケーションの構造

三菱計算機リンクユニット(以下、AJ71 と略す)オムロン上位リンクユニット(以下、LK201 と略す)等、各社シリアル通信のサンプルプログラムをみると N88BASIC で書かれています。しかし N88BASIC を知らない世代の人たちが増えてきているのも現状なんです…。基本的に N88 と VB との違いは、前提プログラミングとイベント駆動型プログラミングの違いがあります。

MS-DOS (N88)**前提プログラミング**

- ・コンピュータが何か要求を出したら、人間が適切な操作をする



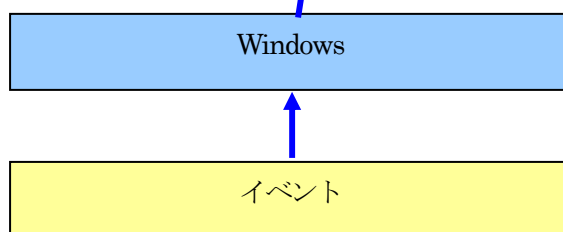
すべての動作はプログラムが制御する

Windows(VB)**イベント駆動型プログラミング**

- ・人間が、あるいは Windows が要求(イベント)を出したらアプリケーションが適切な動作をする



左図は Window の電卓です。マウスを操作し、数字キーをクリックするとその「イベント」が Windows を経由してアクティブな「電卓」に伝えられます。



少し、ぼんやりとした説明になったようです。イベントについては、後の章でまたでてきますのでここでは、**イベント**という言葉覚えていてください。

1-2 VisualBasic の概要

■なぜ、VB？

- ・Windows のOS部を知らなくても、Windows ソフトを開発できる
- ・インタプリタの利便性とコンパイルによる実行速度向上
- ・デザインしている感覚でプログラミングが行える
- ・OCX(ActiveX)を使うことで最新機能や高度な機能を取り込める
- ・書籍やツールなどが多い

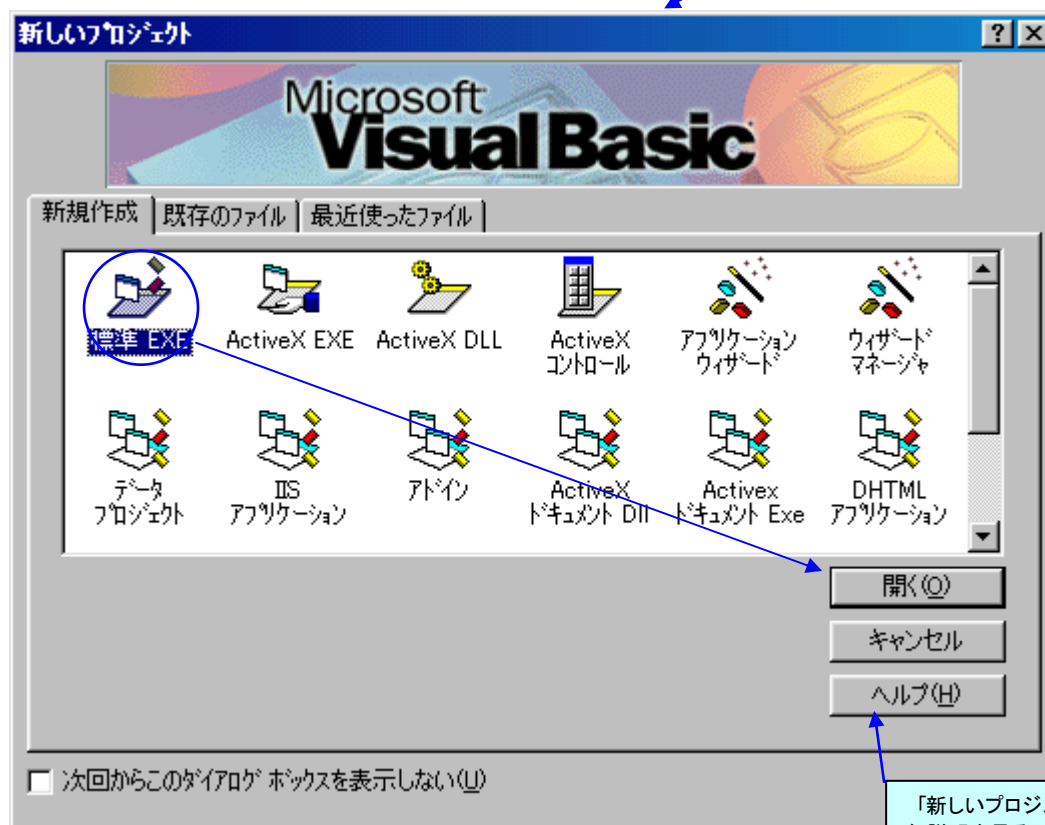
メリットは多くありますが、VisualBasic は万能ではありませんし VisualC++や Delphi の方が優れた部分もあります。しかし、扱いやすさ・気軽に始められることは VisualBasic の大きな点です。

■起動してみよう

スタート-プログラム-Microsoft Visual Studio 6.0-Microsoft Visual Basic 6.0 から起動します



Microsoft Visual Basic 6.0
をクリックします



「新しいプロジェクト」の詳細説明を見ることができます。

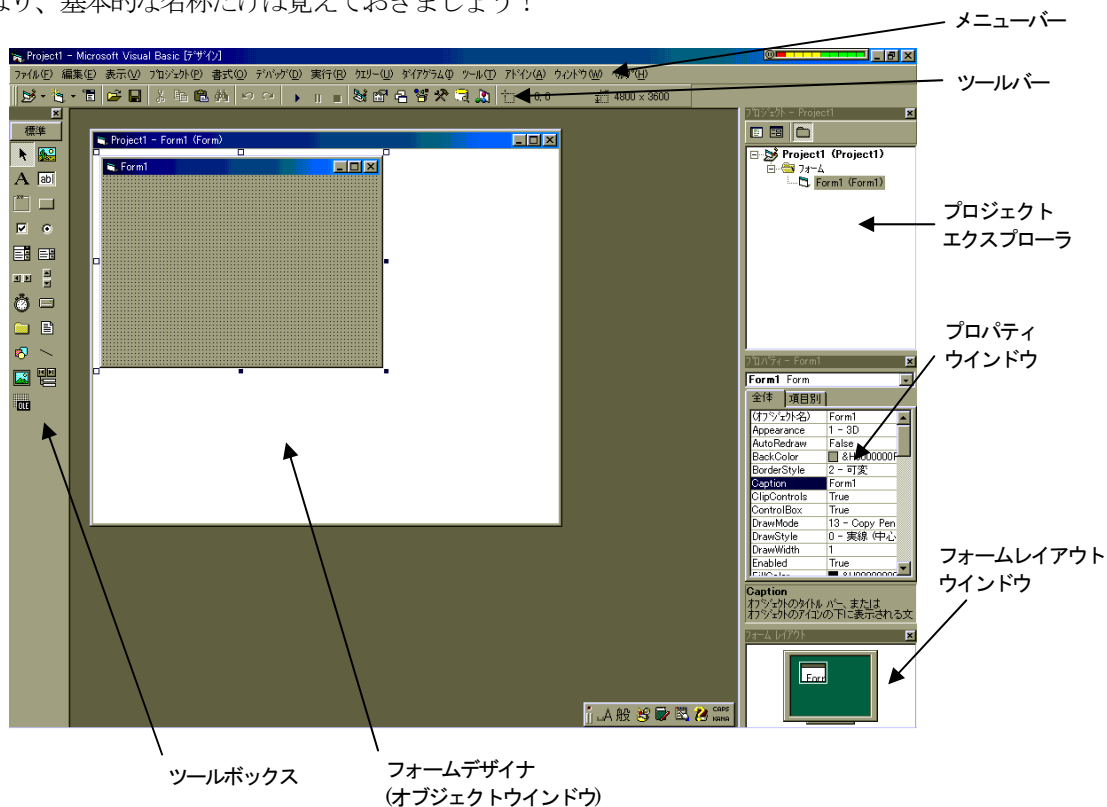


標準の実行可能ファイルが作成されます

標準 EXE

■各ウインドウ名称

やはり、基本的な名称だけは覚えておきましょう！



- ・ メニューバー
アプリケーション作成時に使うコマンド類があります。
- ・ ツールバー
プログラミング環境で頻繁に使うコマンド群
- ・ ツールボックス
ボタン、テキストボックスなどの部品(コントロール)が用意され、Form に貼り付けて使用します
- ・ フォームデザイナー
アプリケーションのユーザーインターフェースを作成するウインドウです。
ツールボックスからさまざまなコントロールを貼り付けてウインドウを作成します。
- ・ プロジェクトエクスプローラ
Form、モジュールなどの構成を表示します。
- ・ プロパティウインドウ
Form に貼り付けたコントロールのプロパティを表示します。
プロパティで、色・サイズなどを変更できます。
- ・ フォームレイアウトウインドウ
実行時の Form レイアウトを確認・設定ができます
- ・ コードウインドウ
Basic コードを記述するウインドウが表示されます



1-3 VisualBasic 全体イメージ

Visual Basic のアプリケーションは、基本的に次の3つのステップに従って作成します。

1. ユーザーインターフェイスを作成します。
2. プロパティを設定します。
3. コードを記述します。

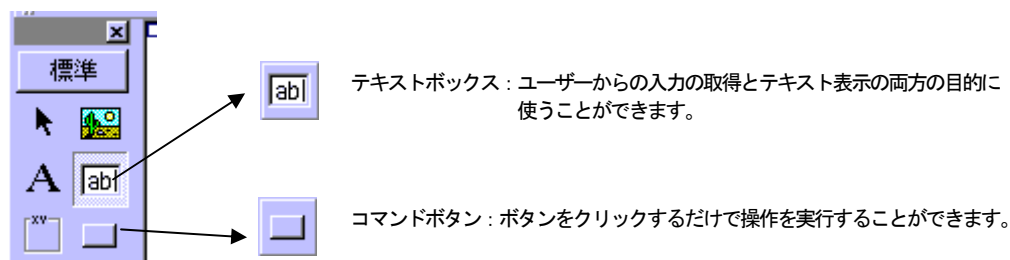
では、実際に作成してみます。

■ユーザーインターフェイス

ユーザーインターフェイスの基本となるフォームを作成します。

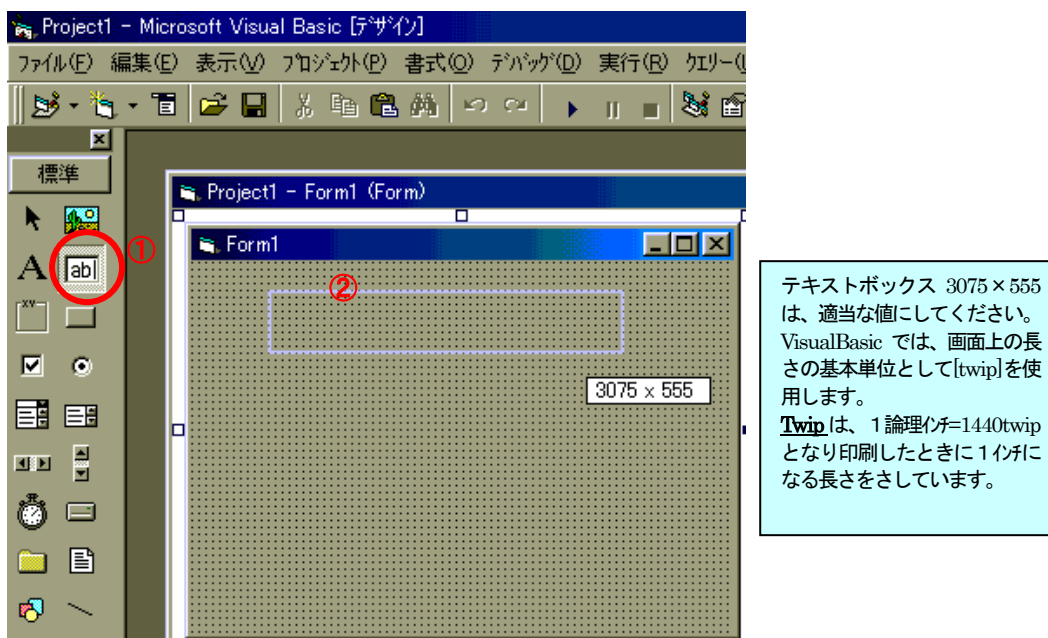
フォームには、ユーザーインターフェイスを構成する**オブジェクト**をフォームに配置します。

これから作成する「Hello Visual Basic!」は、ツールボックスから次の2つのコントロールを使います。(オブジェクトは範囲が広く、フォームやコントロールなどを示します)



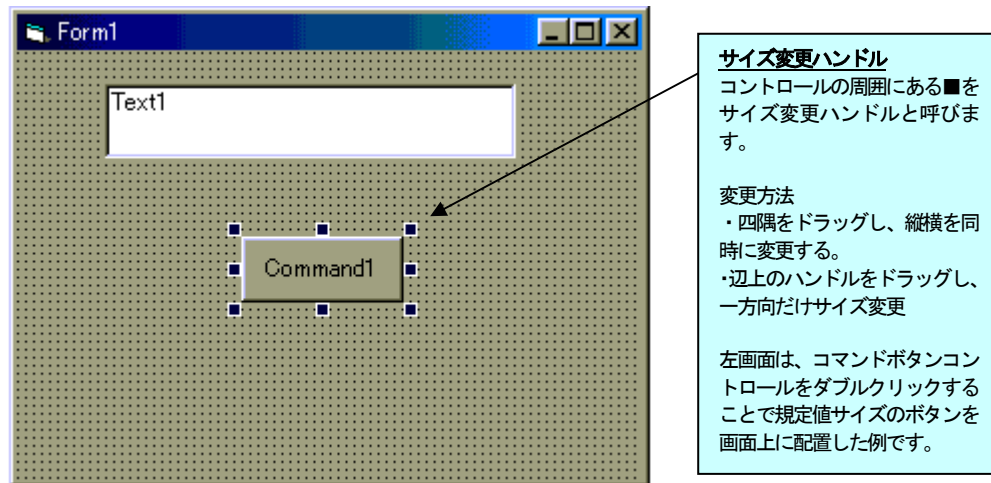
まず、テキストボックスを配置します。

- ① テキストボックスコントロールをクリックします。
- ② マウスポインタをフォーム上に移動し、マウス左ボタンをクリックして目的の大きさになるまでドラッグします。



ツールボックスのコントロールをダブルクリックするだけでも、フォームにコントロールを追加できます。この方法では、コントロールがフォームの中央に規定サイズで配置されますので必要な大きさ・位置に変更します。

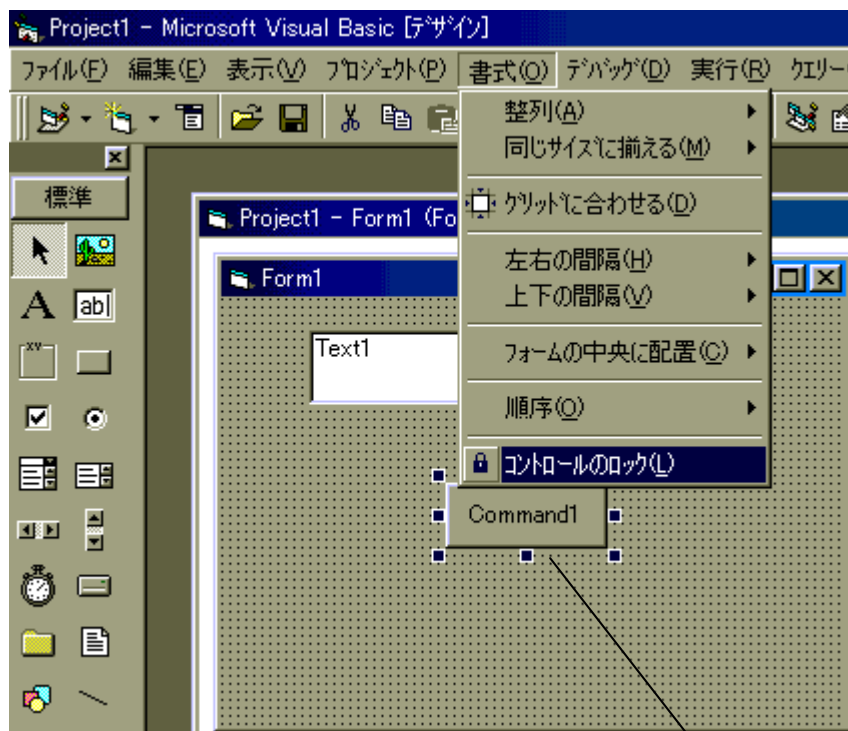
次は、コマンドボタンを同じように配置します。



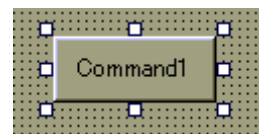
コントロールのロック

フォーム上に配置したコントロールの位置をロックすることができます。

(マウス操作中に不用意にコントロールを移動してしまうことを防止できます)



コントロールのロックを行うと、■が□に変わります。移動する場合は、Ctrlキーを押しながら方向キーを押すと移動することができます。



テキストボックスとコマンドボタンの2つを使用した簡単なユーザーインターフェイスですが、どんなに複雑でも、フォームにコントロールを配置するのは同じです。

これで、ユーザーインターフェイスが完成しましたので次は、プロパティの設定を行います。

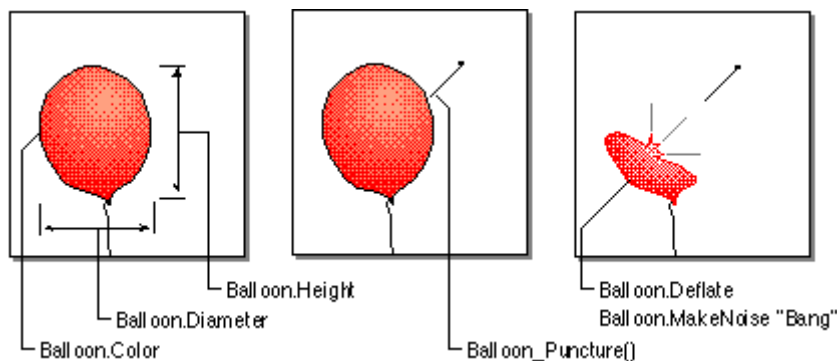
■プロパティの設定

プロパティは**オブジェクト**が持っている属性(特性)です。

設定を変更することで、オブジェクトのサイズ・色・背景・文字書体等を変えることができます。

MSDN ライブラリには、下記の説明がありますので参考にしてください。

オブジェクトはプロパティを持ち、イベントに反応し、メソッドを実行します。



風船(オブジェクト)のプロパティには、高さ(Height)・直径(Diameter)・色(Color)などがあり特有の動作を規定するメソッドがあります。

メソッドには、ヘリウムを入れて膨らませるメソッド(Inflate)・しぼませるメソッド(Deflate)・空中に放すメソッド(Rise)などがあり、さらに風船(オブジェクト)は、外部からの特定の**イベント**に対しあらかじめ定義された反応を示します。

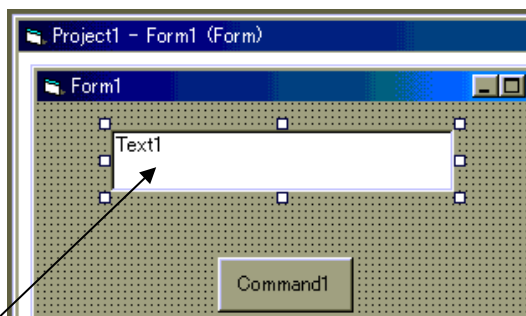
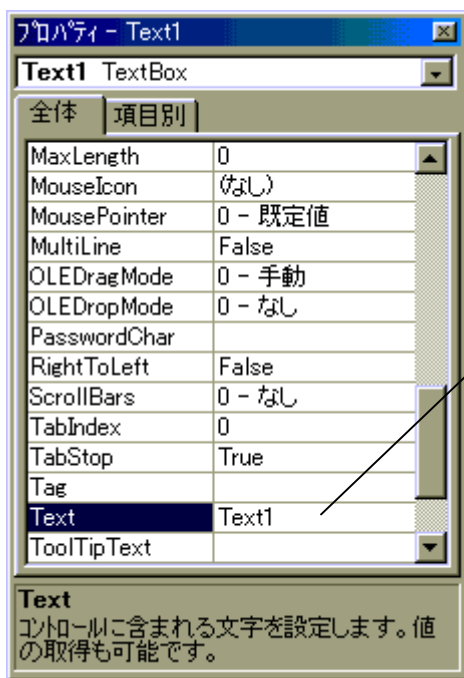
では、実際にフォームのプロパティから見てみましょう。

(オブジェクト名)
 フォームのオブジェクト名を入力します。
 デフォルトで Form1 となっていますのでこのままでも構いません。

Caption
 ユーザーインターフェイスのウィンドウタイトル名を入力します。

その他
 各項目の説明は、プロパティウインドウの下部に表示されていますが、詳細はカーソルを見たい項目へ移動し、[F1]Key を押すと MSDN ライブラリが起動し詳細説明が表示されます。

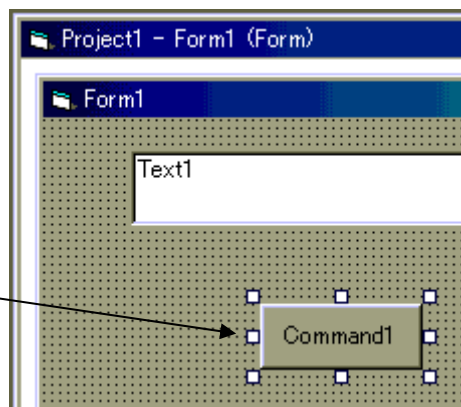
・テキストボックスのプロパティ設定



(オブジェクト名)
テキストボックスのオブジェクト名を入力します。デフォルトでText1となっていますのでこのままでも構いません。
Text
“Text1”という規定値が設定されていますがユーザーインターフェイスからの値取得に使用します。
その他
各項目の説明は、プロパティウインドウの下部に表示されていますが、詳細はカーソルを見たい項目へ移動し、[F1]Key を押すと MSDN ライブラリが起動し詳細説明が表示されます。

ここでは、Text プロパティを空白に設定してください。(Text1 を削除)

・コマンドボタンのプロパティ設定



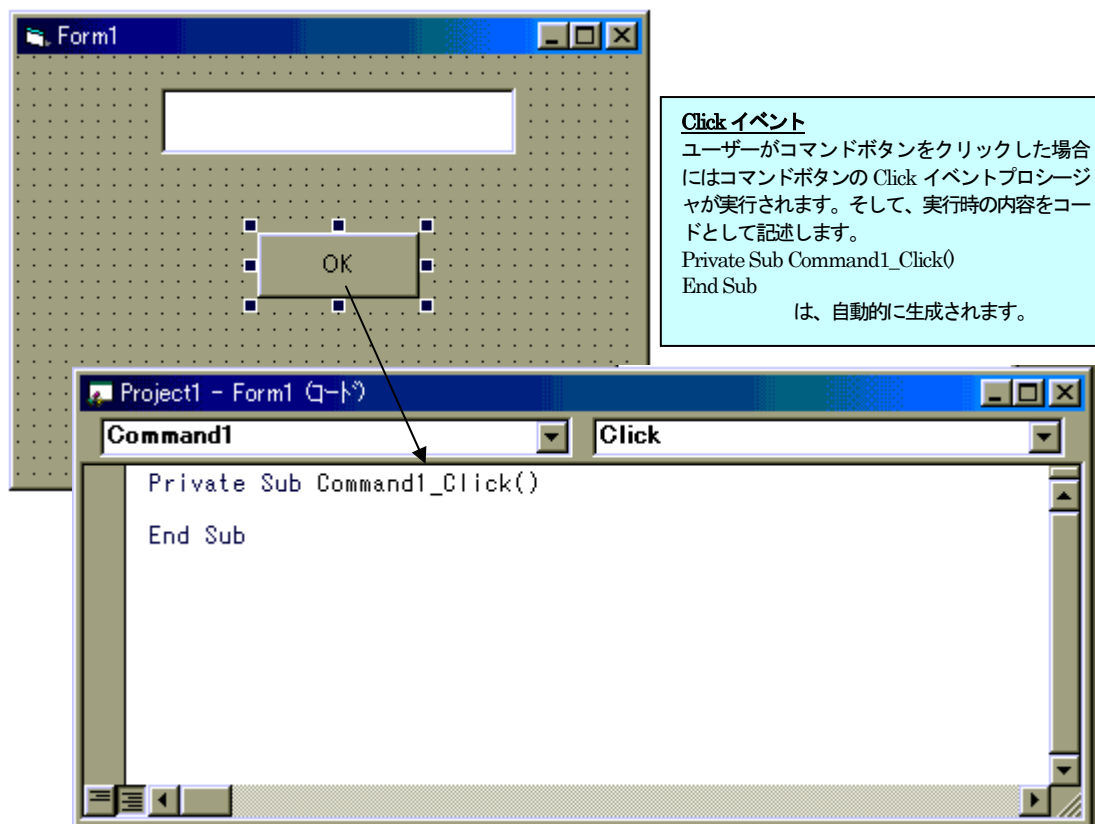
(オブジェクト名)
コマンドボタンのオブジェクト名を入力します。デフォルトでCommand1となっていますのでこのままでも構いません。
caption
Command1 という規定値が設定されていますがボタンの名称ですので分かりやすい名前に変更します。

ここでは、Caption プロパティを”OK”にします。

■コードの記述

Visual Basic では、アプリケーションのコードをコードエディタウィンドウに記述し、コードはステートメント、定数、宣言などの要素から構成されます。

「コマンドボタン」をダブルクリックすると、コードエディタウィンドウが表示されます。



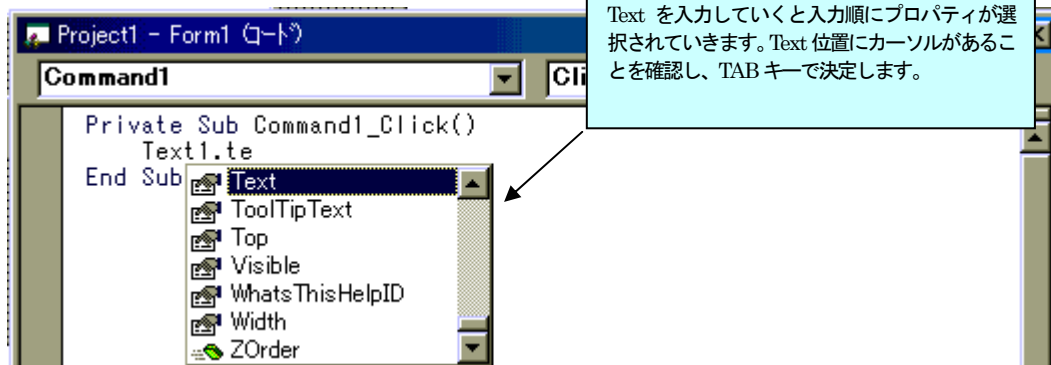
プロシージャとは、Sub(またはFunction)で始まり End で終わるコードブロックで Function プロシージャ、Property プロシージャ、Sub プロシージャの3種類があります。

では、実際に記述してみましょう！

Sub ステートメントと **End Sub** ステートメントの間に次のコードを記述します。

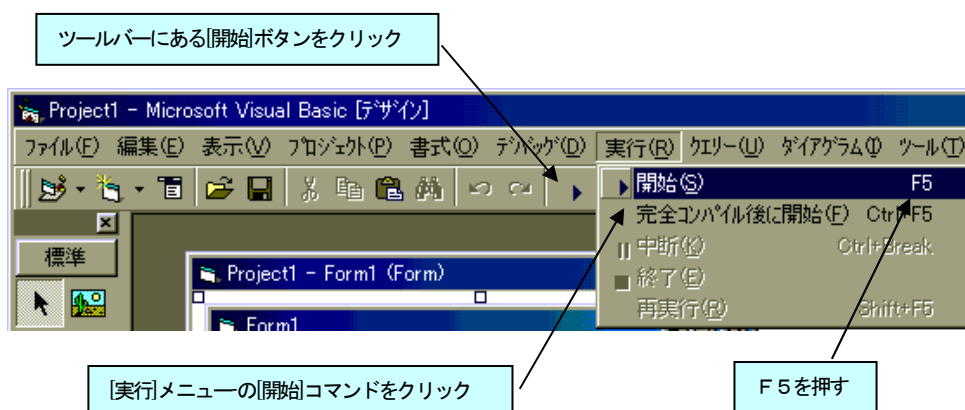
```
Text1.Text = "Hello world"
```

イベントプロシージャは、次のようになります。

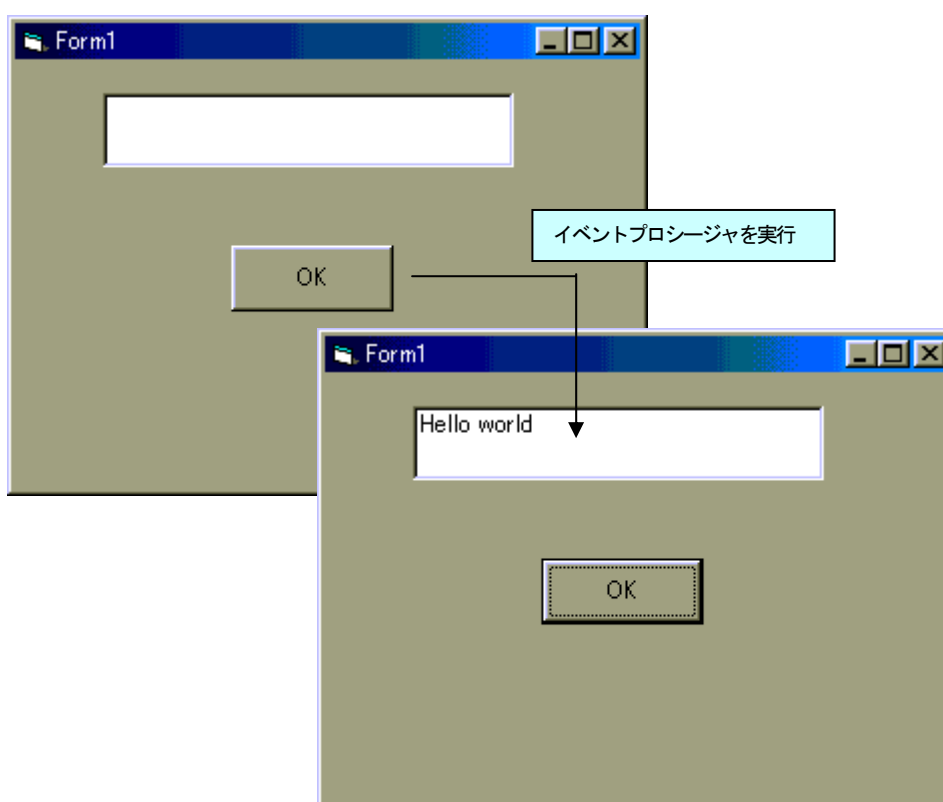


■実行

では、さっそくできあがったアプリケーションを実行してみましょう。



実行の方法は、上記のように3通りありますのでいずれかで実行します。



この動作を N88Basic で記述するとしたら……

- ・ フォームの描画
- ・ 押しボタンの描画
- ・ いつ押されるかわからない Mouse をスキャンするプログラム

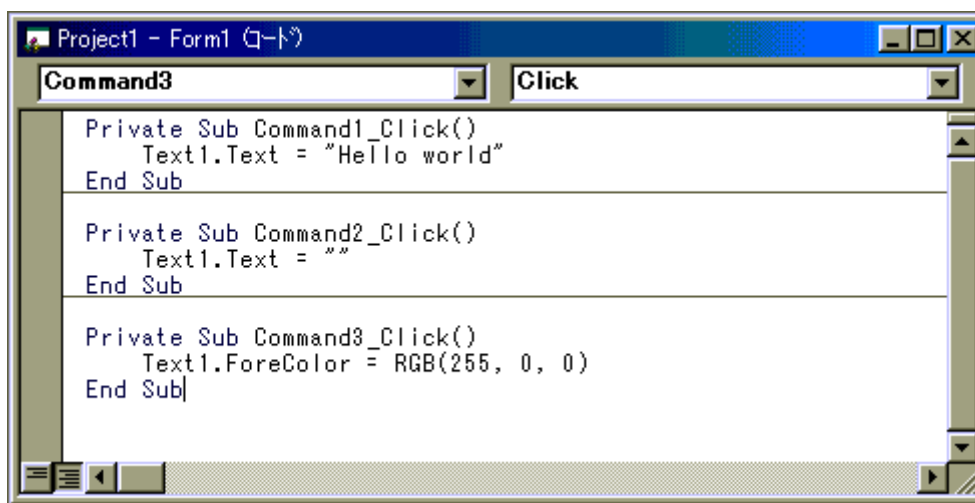
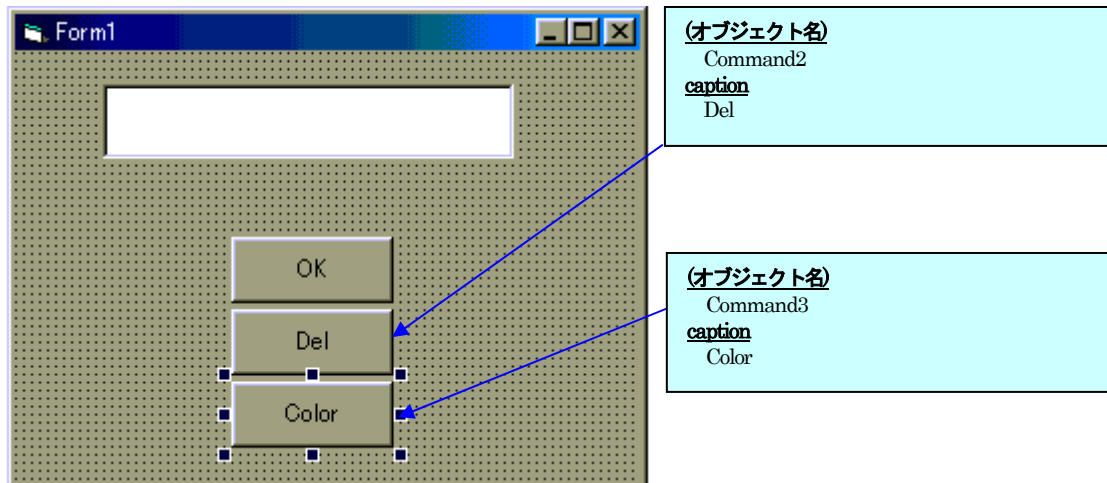
を作成する必要がありますが、Visual Basic でしたらそのような処理を記述する必要がないのです。

では、ここで少し追加をしてみましょう

- ・ Hello world を消去するコマンドボタンを追加
- ・ Hello world の表示色を赤に変更する
- ・

もう、イメージができあがりませんか？

- 「Del」 消去するコマンドボタン・「Color」 表示色変更するコマンドボタンを追加します。



もう、イメージがつかめましたか？

コマンドボタンを押すことによって **イベント** が発生し、Text1 オブジェクト の **プロパティ** が設定されます。

■まとめ

- ・オブジェクト
コントロールやフォームなど、すべてがオブジェクトです
- ・プロパティ
オブジェクトの特性を変えることができます。**Text1.〇〇〇** というようにオブジェクト名+ドット+プロパティ名で記述します。
- ・メソッド
N88BASIC では、命令語にはステートメント・関数がありましたがメソッドはオブジェクトに対する命令語のようなものです。
- ・ステートメント
ここでは出てきませんでしたが、N88BASIC と同様に IF~Then/GOSUB など。

あとは、実践あるのみです。Help を十分に活用して機能を見つけてください。

次の2章では、ActiveX コントロールについて、見てみましょう。

第二章 Visual Basic のコントロール



VisualBasic のツールボックスには、フォームにコントロールを配置するときに使うツールが用意されています。

コントロールの分類

VisualBasic のコントロールは、3種類あります。

・標準コントロール

コマンドボタンコントロール、フレームコントロールなどこの種類のコントロールは、VisualBasic.exe ファイルに保存されています。

・ActiveX コントロール

.ocx という拡張子の付いた、独立したファイルで保存されツールボックスに追加したり、削除したりすることが可能です。

・挿入可能なオブジェクト









Microsoft Excel の Worksheet オブジェクトなど。

旧バージョンでは、.vbx というコントロールがありました。

これらについては、MSDN ライブラリの「VisualBasic の旧バージョンのコントロールのアップデート」を参照ください。

■標準コントロール

アイコン	コントロール名	クラス名	説明
	チェックボックス	CheckBox	真(True)と偽(False)またははい/いいえの選択を表示します。フォームに配置されたチェックボックスから、必要な数だけチェックすることができます。
	コンボボックス	ComboBox	テキストボックスとリストボックスの機能を組み合わせたコントロールです。ユーザーは、選択内容を入力するか、ドロップダウンリストから項目を選択することができます。
	コマンドボタン	CommandButton	ユーザーがクリックしたときに、コマンドまたは操作を実行します。
	データ	Data	既存のデータベースに接続し、そのデータベースの情報をフォームに表示します。
	ディレクトリリストボックス	DirListBox	ディレクトリおよびパスを表示し、ユーザーに選択させます。
	ドライブリストボックス	DriveListBox	使用可能なディスクドライブを表示し、ユーザーに選択させます。
	ファイルリストボックス	FileListBox	ファイルの一覧を表示し、ユーザーに選択させます。
	フレーム	Frame	フレームの中に、別のコントロールを配置することができます。フレームの中に別のコントロールを表示させたり、別のコントロールの機能をフレームに追加することができます。
	水平スクロールバーおよび垂直スクロールバー	HScrollBar および VScrollBar	自動的にスクロールバーが提供されないコントロールに、スクロールバーを追加することができます。通常のコントロールに組み込まれたスクロールバーとは異なります。
	イメージ	Image	ビットマップ、アイコン、Windows メタファイル、JPEG ファイル、または GIF ファイルを表示します。クリップされると、コマンドボタンと同様の働きをします。
	ラベル	Label	テキストを表示します。このテキストは、ユーザーが直接入力したり、変更したりすることはできません。

	ライン	Line	フォーム内で直線を引くときに使います。
	リストボックス	Listbox	項目の一覧を表示し、ユーザーに選択させます。
	OLE コンテナ	OLE	VisualBasic アプリケーションにデータを埋め込みます。
	オプションボタン	OptionButton	複数のオプション(OptionButton)コントロールを組み合わせ、オプショングループとして使います。複数の選択項目を表示し、ユーザーに1つだけを選択させます。
	ピクチャボックス	PictureBox	ビットマップ、アイコン、Windows メタファイル、JPEG ファイル、または GIF ファイルを表示します。テキストを表示したり、ピクチャボックスの中に別のコントロールを配置したりすることもできます。
	シェイプ	Shape	フォーム、フレーム、またはピクチャボックスで長方形、正方形、楕円、または円を描画するときに使います。
	テキストボックス	TextBox	テキストの入力または表示を行う領域を指定します。
	タイマー	Timer	指定された時間間隔で、タイマーイベントを実行します。

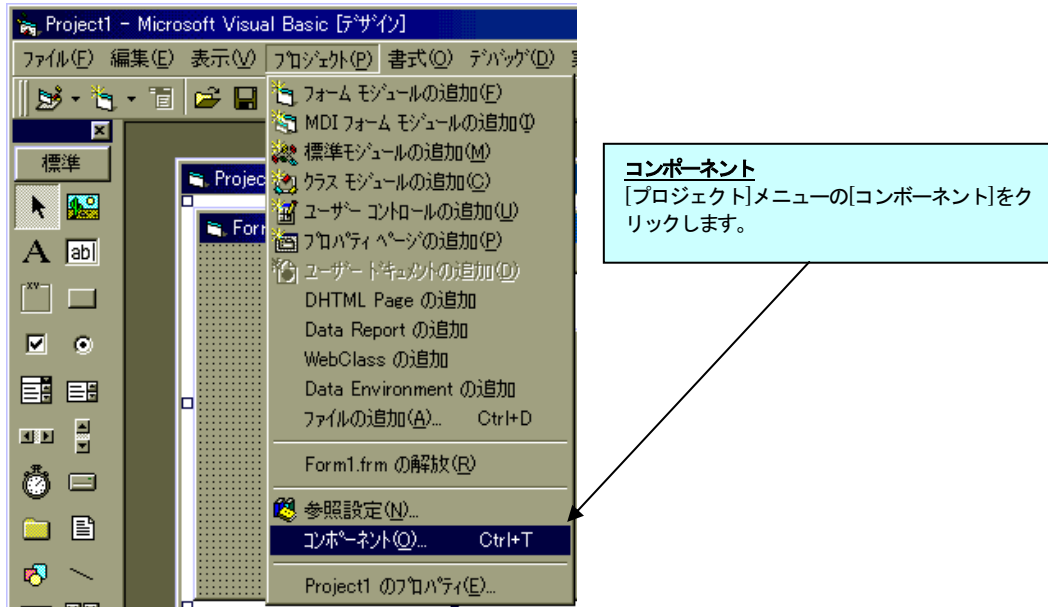
■ActiveXコントロール

Visual Basic には、ActiveX コントロールと呼ばれるコントロールが付属しています。このコントロールは、ファイル名に拡張子.ocx が付き使用する場合は、手動でツールボックスに追加します。Learnig Edition ・ Professional Edition ・ Enterprise Edition の各 Edition によって付属しているコントロールが異なりますので各 Edition で確認してください。

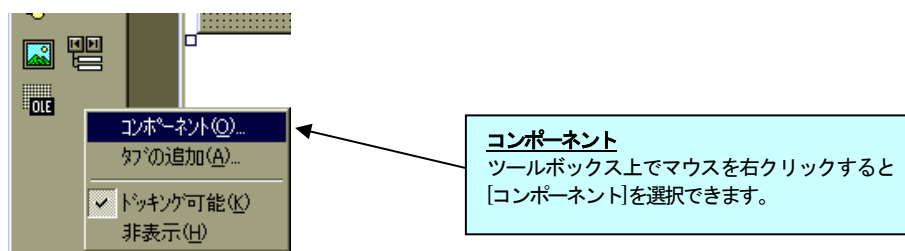
・ActiveX コントロールをツールボックスに追加する

以下の2通りがあります。

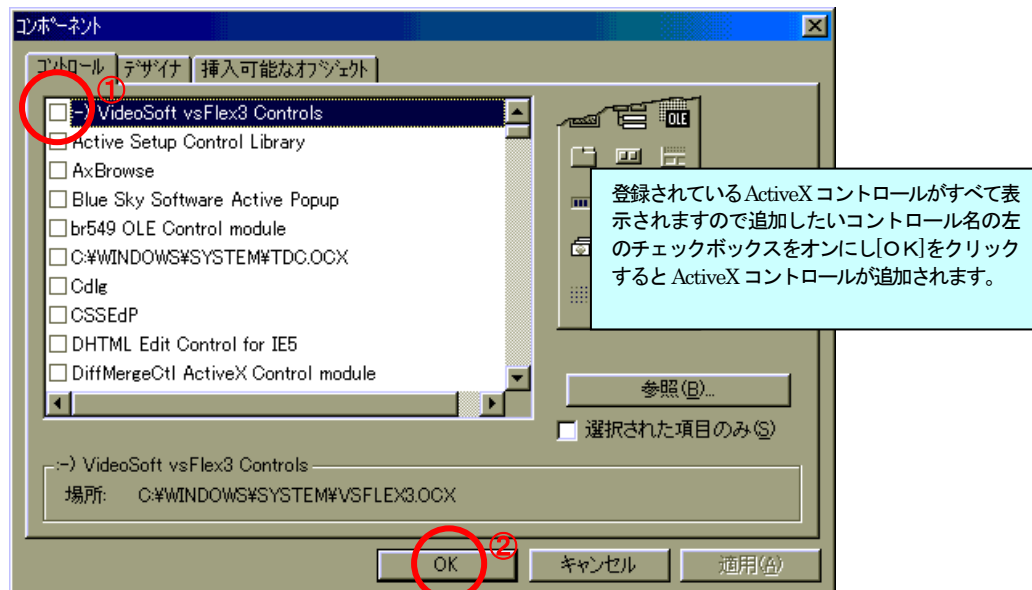
①



②



[コンポーネント] を選択するとコンポーネントダイアログボックスを表示します。



・どんなコントロールがあるの？

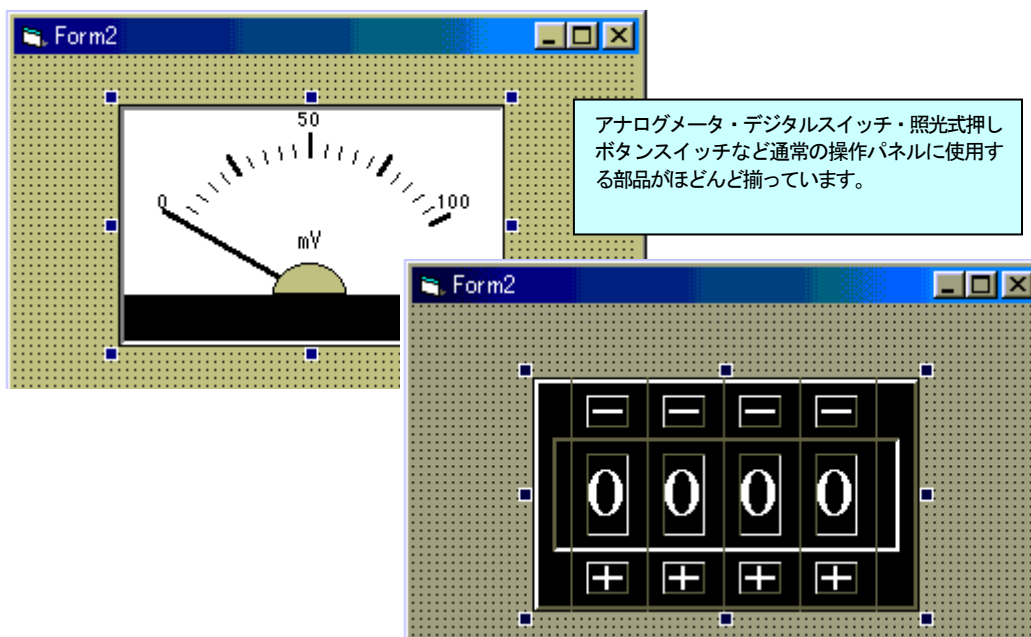
どのようなコントロールがあるかは、Edition によって異なりますのでご自分の PC で確認してください。

コンポーネント名	ファイル名	コントロール
Microsoft ADO Data Control6.0	MSADODC.OCX	ADO データ (ADODC)
Microsoft Chart Control6.0	MSCHART.OCX	チャート (MSChart)
Microsoft Comm Control6.0	MSCOMM32.OCX	コミュニケーション (MSComm)
Microsoft Common Dialog Control6.0	COMDL32.OCX	コモンダイアログ (CommonDialog)
Microsoft Data Bound List Controls6.0	DBLST32.OCX	DB リスト (DBList), DB コンボ (DBCombo)
Microsoft Data Repeater Control6.0	MSDATREP.OCX	データリピータ (DataRepeater)
Microsoft Data Grid Control6.0	MSDATGRD.OCX	データグリッド (DataGrid)
Microsoft Data List Controls6.0	MSDALST.OCX	データリスト (DataList), データコンボ (DataCombo)
Microsoft FlexGrid Control6.0	MSFLXGRD.OCX	フレキシブルグリッド (MSFlexGrid)
Microsoft Hierarchical Flex Grid Control6.0	MSHFLXGD.OCX	階層フレキシブルグリッド (MSHFlexGrid)
Microsoft Internet Transfer Control6.0	MSINET.OCX	インターネットトランスファコントロール (Inet)
Microsoft MAPI Controls6.0	MSMAPI32.OCX	MAPI メッセージ (MAPIMessages) MAPI セッション (MAPISession)
Microsoft MaskedEdit Control6.0	MSMASK32.OCX	マスクエディット (MaskedTextBox)
Microsoft Multimedia Control6.0	MCI32.OCX	マルチメディア MCI (MControl)
Microsoft PictureClip Control6.0	PICCL32.OCX	ピクチャクリップ (PictureClip)
Microsoft RemoteData Control6.0	MSRDC30.OCX	リモートデータ (MSRDC)
Microsoft Rich TextBox Control6.0	RICHTX32.OCX	リッチテキストボックス (RichTextBox)
Microsoft SysInfo Control6.0	SYSDINFO.OCX	システム情報 (SysInfo)
Microsoft TabbedDialog Control6.0	TABCTL32.OCX	タブダイアログ (SSTab)
Microsoft Windows Common Controls6.0	MSCOMCTL.OCX	タブストリップ (TabStrip), ツールバー (Toolbar) ステータスバー (StatusBar), プログレスバー (ProgressBar), ツリービュー (TreeView), リストビュー (ListView), イメージリスト (ImageList), スライダー (Slider), イメージコンボ (ImageCombo)
Microsoft Windows Common Controls-2 6.0	MSCOMCT2.OCX	アニメーション (Animation), アップダウン (UpDown), MonthView, Dtpicker, フラットスクロールバー (FlatScrollbar)
Microsoft Winsock Control6.0	MSWINSCK.OCX	WinSock

・商用コンポーネント

ActiveX コントロール単体で、いろいろな会社から販売されています。

制御装置の操作パネル用のコントロールもありますので必要に応じて参考にしてください。



第3章 プログラミングの基礎

1章と2章で全体像が掴めましたか？

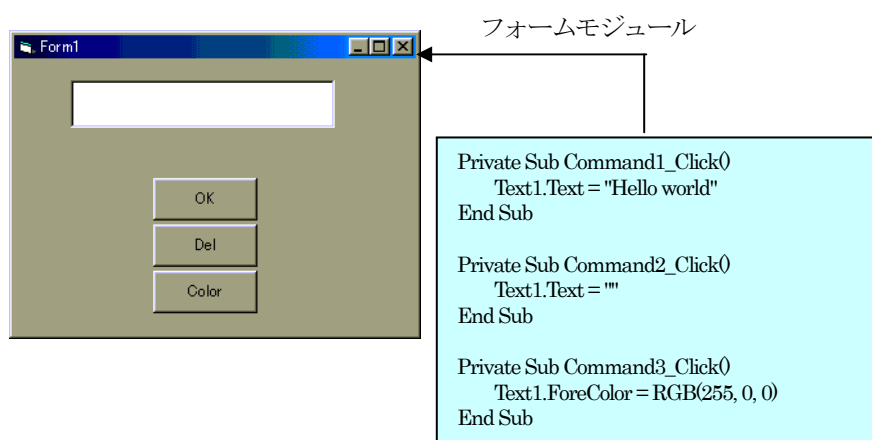
RS232C と MSComm の話に入る前に、プログラミングの基礎について少しだけお付き合いください。といっても、一からの話ではなく概念だけです。(本書は、あくまでも概要を理解していただくため詳細や各関数等の説明は、既に多くの書籍が販売されていますのでそちらを参考にしてください)

・コード

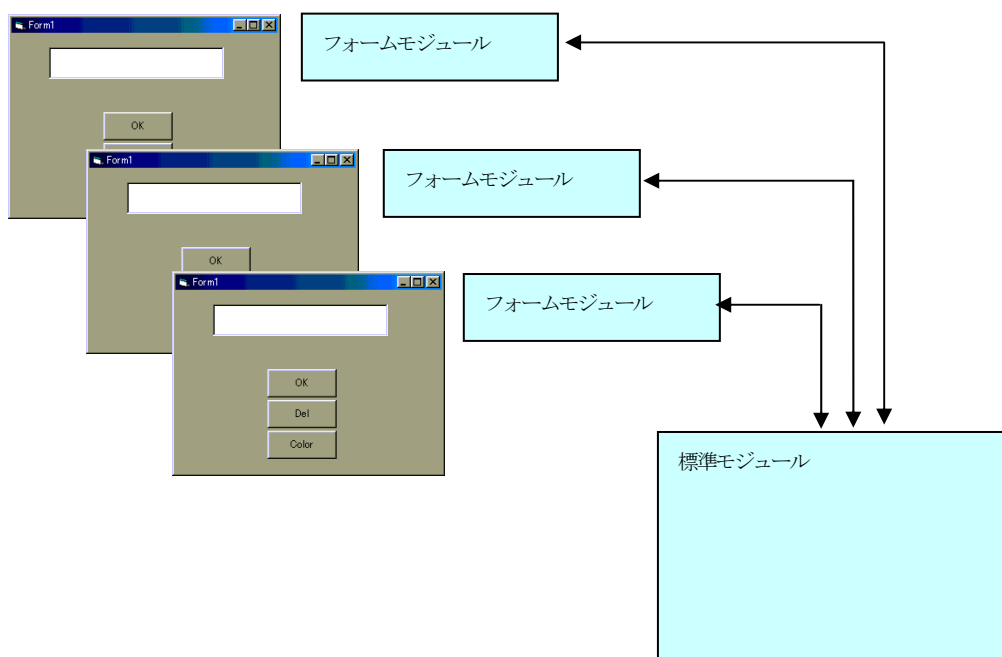
VisualBasic のコードは、モジュールに保存されモジュールには、次の3つがあります。

- ・フォームモジュール
- ・標準モジュール
- ・クラスモジュール

簡単なアプリケーションは、一つのフォームで作成することができ、そのフォームモジュールにすべてのコードを含めることができます。



アプリケーションが大きく複雑になると、フォームが複数になり各フォームで同じコードを重複して記述することがあります。この共通のコードを標準モジュールに共通コードとして記述することができます。



・変数

VisualBasic では、多くのプログラミング言語と同じように、値を格納する「変数」を使います。変数は、名前とデータ型を宣言して使用します。

Dim variablename [As type]

変数は、その宣言する場所によって適用範囲が異なります。プロシージャ内部で変数を宣言すると、その変数は、宣言したプロシージャ内でのみアクセス可能です。このような変数をローカル変数と呼び、その変数が有効である範囲を適用範囲(スコープ)と呼びます。

Form1



```

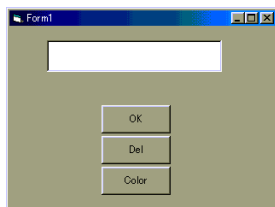
フォームモジュール
Private Sub Command1_Click()
    Dim test as Integer
End Sub
Private Sub Command2_Click()
    Dim test as Integer
End Sub
Private Sub Command3_Click()
    Dim test as Integer
End Sub

```

プロシージャ内で有効な変数

Dim test as Integer が個々のプロシージャで宣言していますが、同じ名前でもすべてが独立した変数です。Command1 で test に数値を代入して Command2 で test を参照しても値を取得することはできないのです。

Form2



```

フォームモジュール
Private formTest as Integer
Private Sub Command1_Click()
    Dim test as Integer
End Sub
Private Sub Command2_Click()
    Dim test as Integer
End Sub
Private Sub Command3_Click()
    Dim test as Integer
End Sub

```

モジュール内で有効な変数

Private formTest as Integer がフォームモジュールの先頭で宣言しています。このように宣言することでモジュール内のすべてのプロシージャでこの変数を参照・変更できます。先頭で宣言する場合に Dim を使用しても動作は同じになりますが Private であることを示すことでコードがわかりやすくなります。この変数も Form1 から参照することはできません。

すべてのモジュールで有効な変数

Private から Public を使用することでアプリケーションのすべてのプロシージャで利用可能になります。

・プロシージャ

VisualBasic では、コードを「プロシージャ」と呼ばれる単位に分割してモジュールを構成します。
(従来のサブルーチンのようなものです)

プロシージャには

Sub プロシージャ	値を返さない
Function プロシージャ	値を返す
Property プロシージャ	値を返したり、代入することが可能

詳細については、この後の章で実際のプログラムをご覧ください。

・オブジェクト

オブジェクトとは、1つの単位として扱うことのできるコードとデータの組み合わせです。コントロールやフォームなどアプリケーションの一部も1つのオブジェクトであり、アプリケーション全体も1つのオブジェクトです。

オブジェクトを使うと、そのオブジェクトが提供する機能を、コードを記述せずに利用することができます。

たとえば、コマンドボタンコントロールを使用するとボタンの表示、ボタンの動作などコードを記述することなくコマンドボタンの機能を使うことができます。

さらに、VisualBasic が持っているコントロールを組み合わせるアプリケーションを作る以外にほかのアプリケーションが提供するオブジェクトを使っても構築することができます。

他のアプリケーションのオブジェクトを利用したアプリケーション

	食費	ガソリン代	宿泊費
1月		12	17
2月		17	11
3月		22	29
4月		14	10
5月		12	17

Excel の Worksheet オブジェクト

Excel の Chart オブジェクト

Word の Document オブジェクト

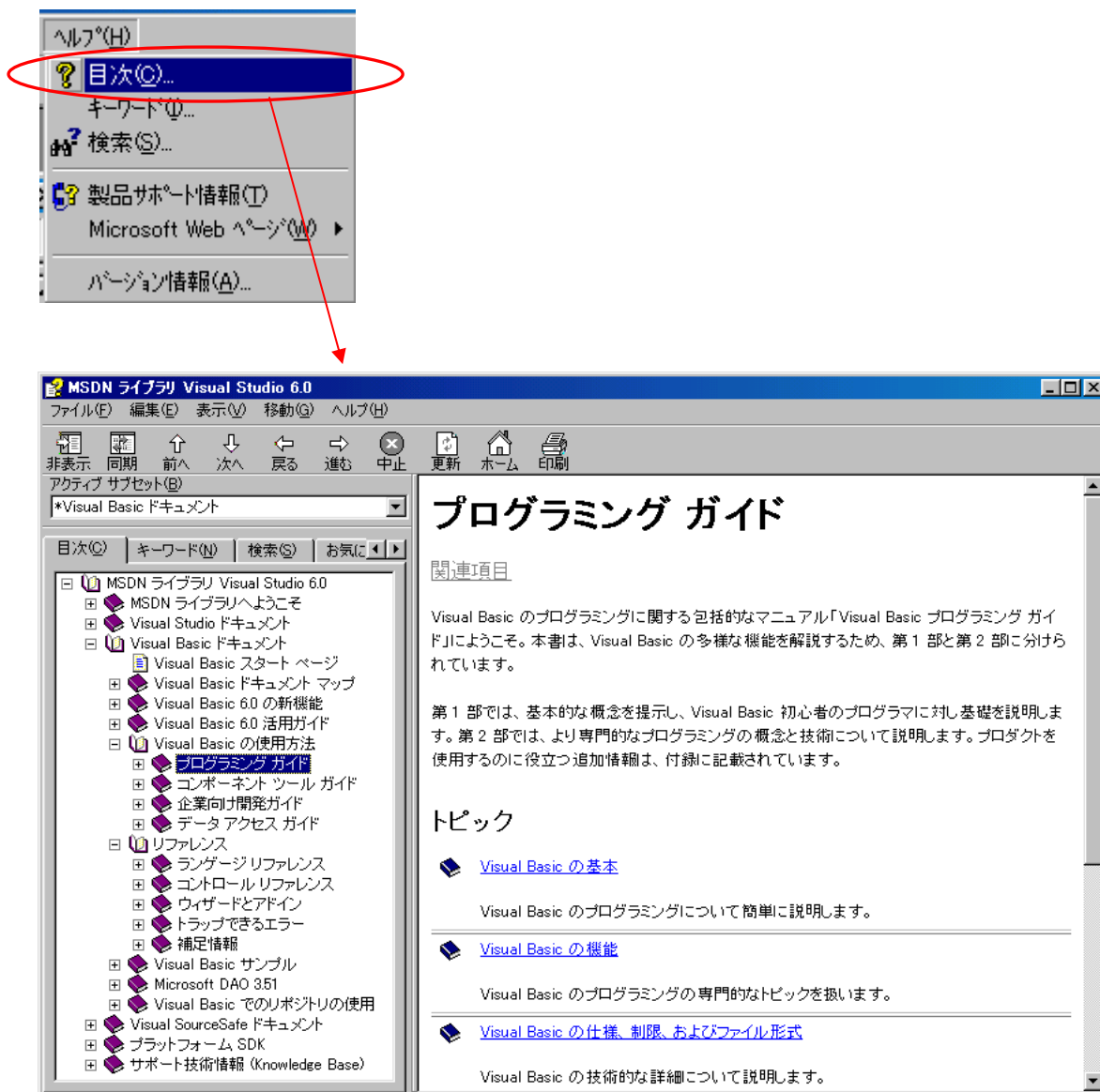
業務経費報告

食費

・MSDNライブラリ

なんとなく、プログラミングの流れが分かりましたか？。詳細は、書店にたくさんの出版物がありますが「MSDNライブラリ」という便利な資料がありますので、一度、ご覧ください。

メニューバー — 目次(C)... をクリックします



実際のプログラミングについては、「RS232C 編」「Ethernet 編」「Web 編」「ActiveX 編」「.NET 編」を通して、紹介していきます。

基本的には、「PLC～PC間通信」にターゲットを絞りますが、プロトコル(通信手順)だけの問題ですので他の機器にも応用可能です。

また、取り上げる PLC は、三菱・オムロン・横河を対象に記載していくつもり？です。