
アプリケーション理解

Excel VBA

【目次】

リストボックスを利用する	3
リストボックスに表示するデータの準備.....	3
ユーザーフォームの準備	3
一覧表の行が増えてもいいようにしてみよう	5
選択したデータを読んでもみる.....	6
売上入力を作ってみよう	7

リストボックスを利用する

ユーザーフォームでは、一覧から選択する「リストボックス」を利用できます。これができるば、一覧から選択してデータを入力できるので、入力間違いを防ぐことができます。

リストボックスに表示するデータの準備

リストボックスでは、シートに準備してあるデータを読み込んで表示できます。そこでまずデータをシート上に作ってみましょう。

最終的にユーザーフォームからの出力を Sheet1 に出したいので、一覧データは Sheet2 に準備しましょう。

Sheet2 を表示して、以下の様なデータを入力してみましょう。

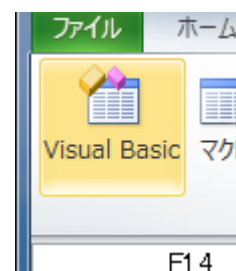
	A	B
1	商品名	価格
2	消しゴム	40
3	エンピツHB	40
4	エンピツ2B	40
5	蛍光オブティクス	85
6	ヤマト アラビック	115
7	フリクションノック07	204
8	ゼブラノックUK0.7	72
9	強力付箋	442
10		
11		

データの内容は、他のものでも構いません。ただのサンプルです。

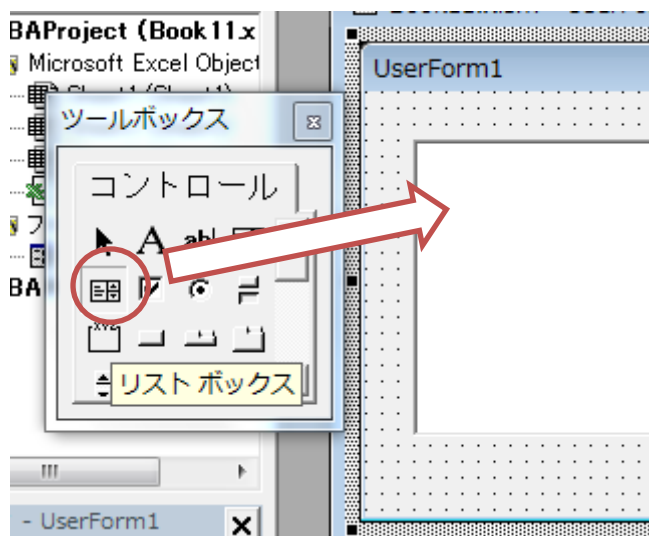
ユーザーフォームの準備

今度は、ユーザーフォームを作成して、リストボックスを貼り付けて、画面を作成します。

開発タブから、Visual Basic アイコンを開き挿入メニューから、ユーザーフォームを追加します。



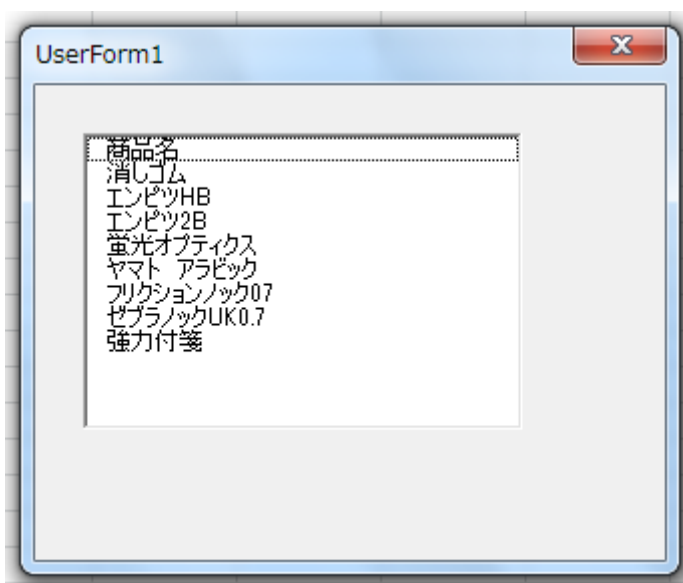
今回は、リストボックスをはりますので、一覧のアイコンをフォームに貼ります。



貼っただけでは、中にはなにも表示されません。この中に先ほど Sheet2 で入力した内容を割り当てます。データを貼り付けるのではなく、どこのデータを利用するか登録するとリストボックスが自動で読み込んでくれます。

```
Private Sub UserForm_initialize()  
    ListBox1.RowSource = "Sheet2!A1:A9"  
End Sub
```

これで、1行目から9行目までを読み込んで、表示してくれます。



マウスでクリックすると、選択行に色が付いて、選択しているのがわかります。

一覧表の行が増えてもいいようにしてみよう

上のプログラムでは、1行目から9行目で固定です。これだと、あとから一覧表が増えても、リストボックスは9行のままです。

そこで前回利用した、最後の行を探すロジックを使って、行の一番最後を探して、そこまでを登録するように改造してみましょう。

```
Dim lastRow As Integer
lastRow = Worksheets("Sheet2").Cells(Rows.Count, 1).End(xlUp).Row
ListBox1.RowSource = "Sheet2!A1:A" & lastRow
```

lastRow という変数を作成し、最後の行を探します。ここに Rows.Count とは、このシートの一番下の行ということです。そこから、xlUp つかり上の方向に突き当たりまで探した行の番号を探しています。

ところで、これだと価格が読み込まれていません。どうすれば価格まで読み込めるようになるでしょうか？

そうですね、A1 から B9 まで読むようにすればいいわけですから

```
ListBox1.RowSource = "Sheet2!A1:B" & lastRow
```

B に変えればいいですね。でも、これだと表示されません。

実は、リストボックスは列数があって、指定しないと1列になっています。そこで列数を2列に変更します。

```
ListBox1.ColumnCount = 2
ListBox1.RowSource = "Sheet2!A1:B" & lastRow
```

これで、2列表示されるようになります。

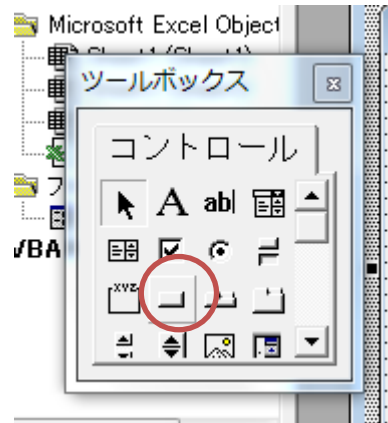
選択したデータを読んでみる

今度はコマンドボタンを貼って、選択したものをシートに書き出してみましよう。

コマンドボタンを貼ります。

コマンドボタンの caption を購入に変えて、おきましょう。
そして、購入ボタンをダブルクリックしてプログラムを書き込みます

ListBox1 がリストボックスコントロールで
ListBox1.ListIndex が現在選択されている
行番号です。



```
Private Sub CommandButton1_Click()  
    If ListBox1.ListIndex <= 0 Then Exit Sub  
    With Sheets("sheet1")  
        .Range("A1") = ListBox1.List(ListBox1.ListIndex, 0)  
        .Range("B1") = ListBox1.List(ListBox1.ListIndex, 1)  
    End With  
End Sub
```

最初の if 文は、タイトル行が選択されていたり、なにも選択されていない時のもので、その時はなにもしないでサブルーチンを終わります。

そして、新しく With 構文が出てきました。これは、何回もコントロールの名前を書くのを省略するためのもので、With Sheets("Sheet1")と書けば、次に End With が来るまでの間、. (ドット) で書き始めた文の頭に Sheets("Sheet1")が補完されます。ですから、.Range は自動的に Sheets("Sheet1").Range となります。同じコントロールの中のものを何度も使用するときに使います。

売上入力を作ってみよう

今は全部 Sheet1 の 1 行目に出力していますが、これを改造して、みましょう。

	A	B	C	D
1	城西高校購買部販売記録			
2	品名	価格		
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				

```
Private Sub CommandButton1_Click()  
    Dim lastRow As Integer  
  
    If ListBox1.ListIndex <= 0 Then Exit Sub  
  
    With Sheets("sheet1")  
        lastRow = .Cells(Rows.Count, 1).End(xlUp).Row + 1  
        .Range("A" & lastRow) = ListBox1.List(ListBox1.ListIndex, 0)  
        .Range("B" & lastRow) = ListBox1.List(ListBox1.ListIndex, 1)  
    End With  
End Sub
```

これで、選択して購入とすればデータが増えていきます。

課題：個数を入力できるように改造しよう