

# **5.** 敘述及物件

\*\*\*\*\*

## **5.1** InputBox 和 MsgBox

## **5.2** 亂數

## **5.3** 陣列

## **5.4** Load、Unload

## 5.1 InputBox 和 MsgBox

### 5.1.1 InputBox 函數

InputBox 是 vb 內建的一個函數，和一般 TextBox(文字方塊)最大的不同點在，程式執行中，文字方塊並沒有辦法讓程式暫停等待使用者輸入資料。以下舉範例說明 InputBox 的用法以及它與 TextBox 的差異。

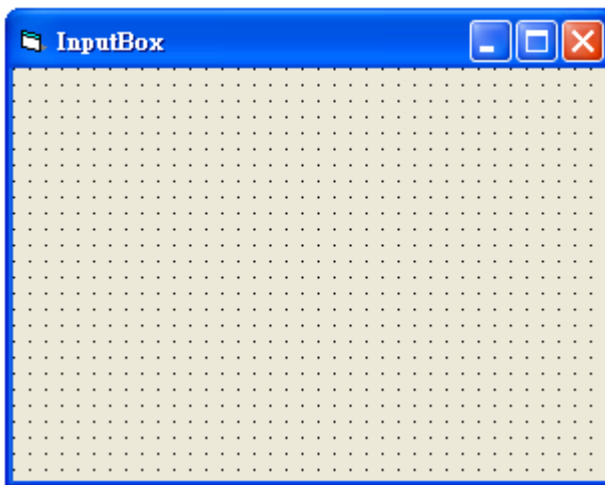


#### InputBox(提示訊息，標題，預設值，Xpos，Ypos)

前 3 個如果是字串就要加雙引號，數字不一定要加，如果只要第 1 和 3 個，第 2 個要留空位(提示訊息，預設值)

**【範例 1】：**將 InputBox 中使用者輸入的內容輸出在表單上

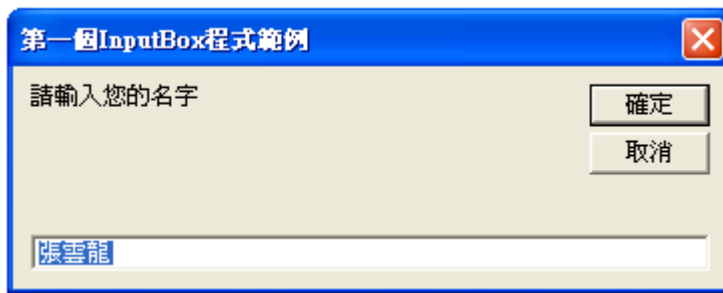
#### 1. 表單配置



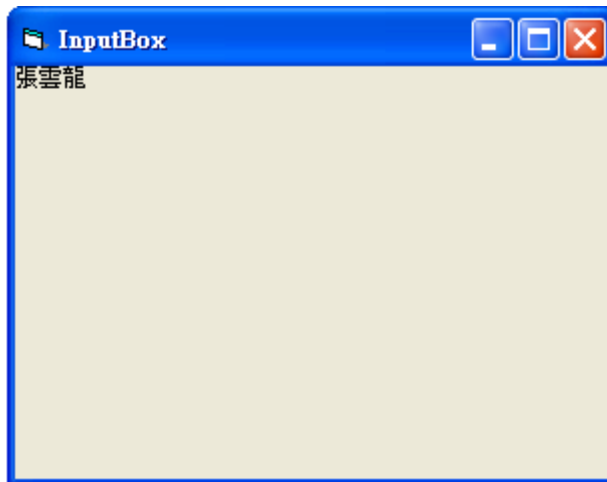
#### 2. 程式碼

```
Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load() 'form_Load()表示程式一啟動就會執行
    Dim sTr As String
    sTr = InputBox("請輸入您的名字", "第一個InputBox程式範例", "張雲龍")
    Form1.AutoRedraw = True '開啓表單自動重繪功能
    Print sTr '輸出姓名
End Sub
```

### 3. 執行程式

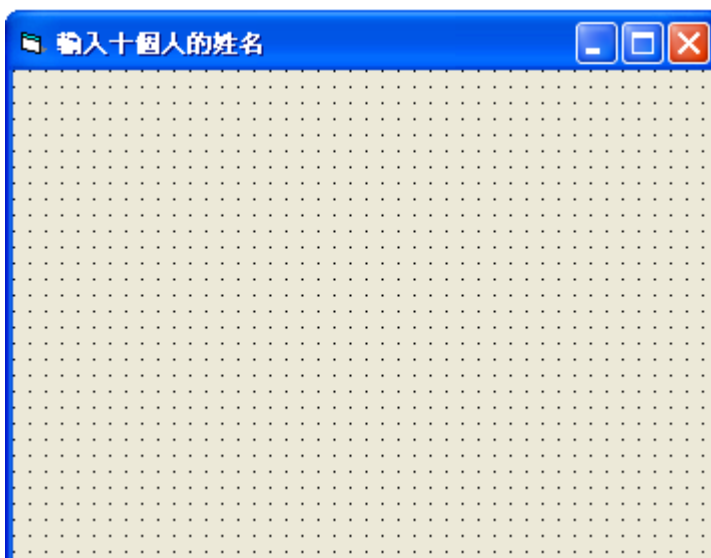


### 4. 按下確定後

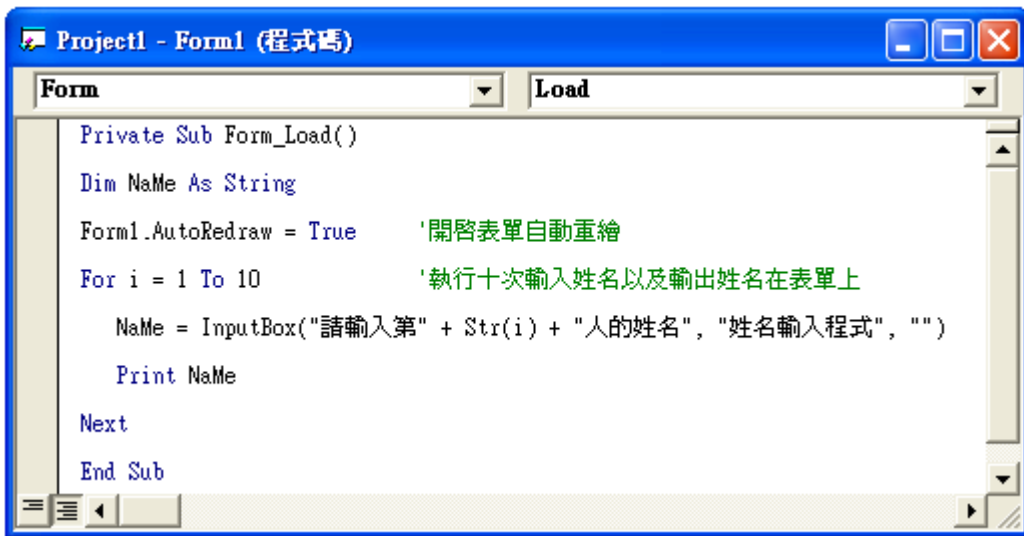


**【範例 2.】**：設計一個利用 InputBox 輸出 10 個姓名的程式

### 1. 表單配置

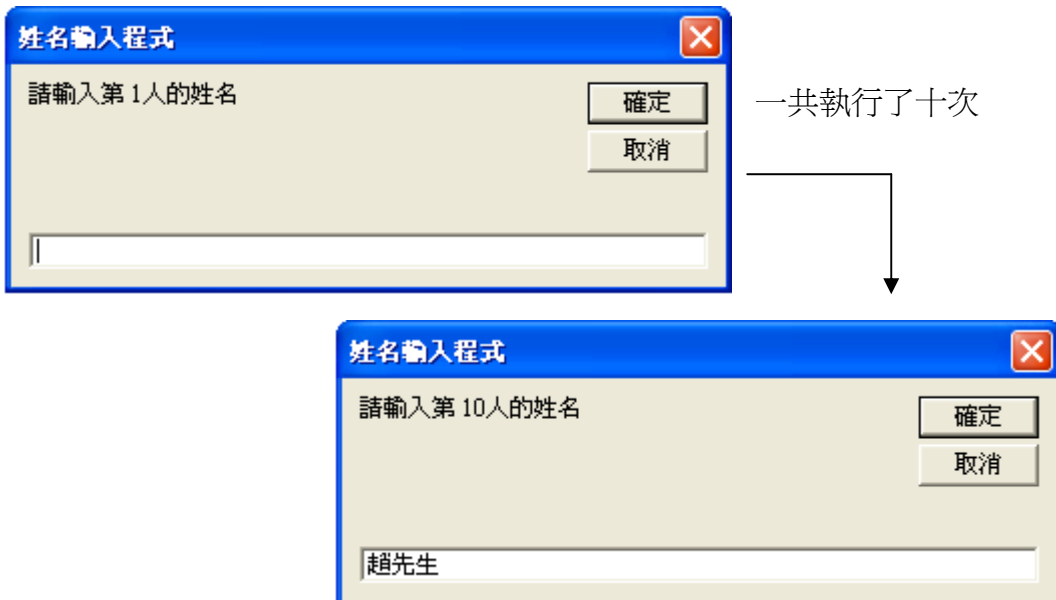


## 2. 程式碼

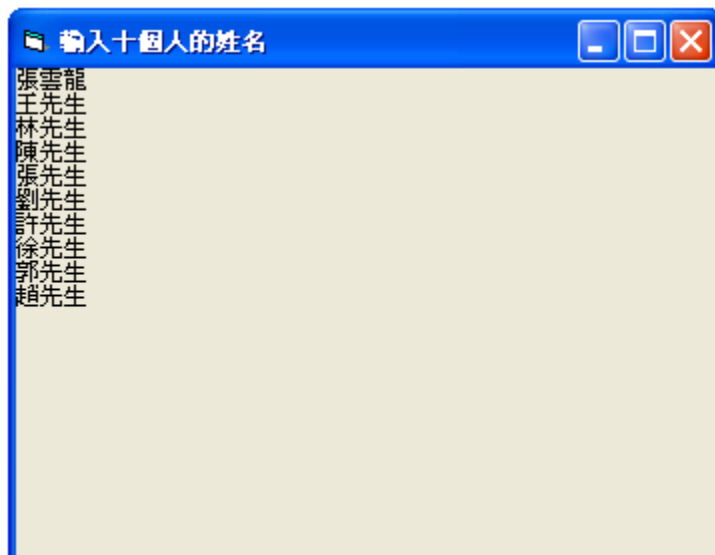


```
Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load()
    Dim NaMe As String
    Form1.AutoRedraw = True '開啓表單自動重繪
    For i = 1 To 10 '執行十次輸入姓名以及輸出姓名在表單上
        NaMe = InputBox("請輸入第" + Str(i) + "人的姓名", "姓名輸入程式", "")
        Print NaMe
    Next
End Sub
```

## 3. 執行結果



## 4. 輸入的十個姓名最後都出現在表單上了



## 5.1.2 MsgBox 函數和敘述

函數和敘述最明顯的差異在於是否需要傳回值，若是需要作選擇的提示視窗，我們必須使用 MsgBox 函數來取得傳回值；如果只是要顯示一個警告視窗，則不需傳回值，就在這個時間，就在這個地點，MsgBox 敘述出場的時機到了。

### 5.1.2.1 MsgBox 函數



MsgBox(提示訊息，按鈕組合和圖示名稱，標題)

【範例】：

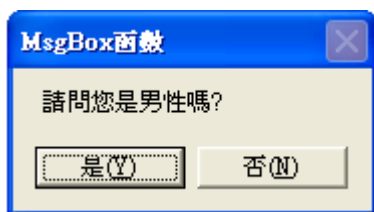
(1) 表單配置



(2) 編寫程式碼：因為是在 Form\_Load()寫，所以要加 Form1.AutoRedraw = True，如果是在 Form\_Activate()就甬加了。

```
Project1 - Form1 (程式碼)
Form Load
Private Sub Form_Load()
    Dim Re As String '宣告字串變數用來接受Msgbox的回傳值
    Form1.AutoRedraw = True '開啓表單重繪功能
    Re = MsgBox("請問您是男性嗎?", vbYesNo, "MsgBox函數") '接收傳回值
    If Re = vbYes Then Print "您是男性" '輸出使用者的選擇
    If Re = vbNo Then Print "您是女性"
End Sub
```

(3)執行結果



(4)選擇『是』的話



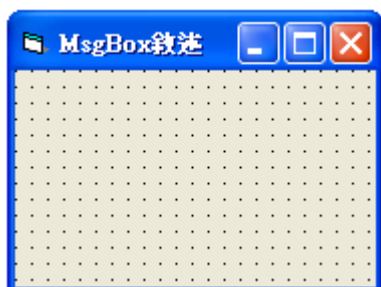
(5)選擇『否』的話



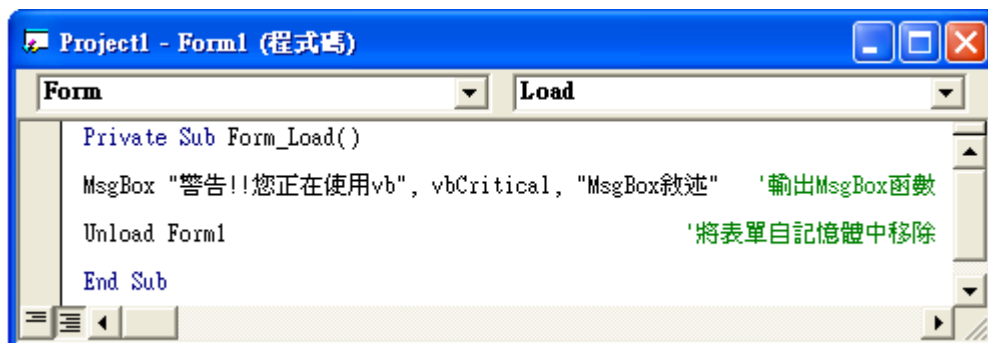
### 5.1.2.2 MsgBox 敘述

【範例】：

(1) 表單配置



(2) 撰寫程式碼



(3) 執行結果，按下確定後即可自動結束程式



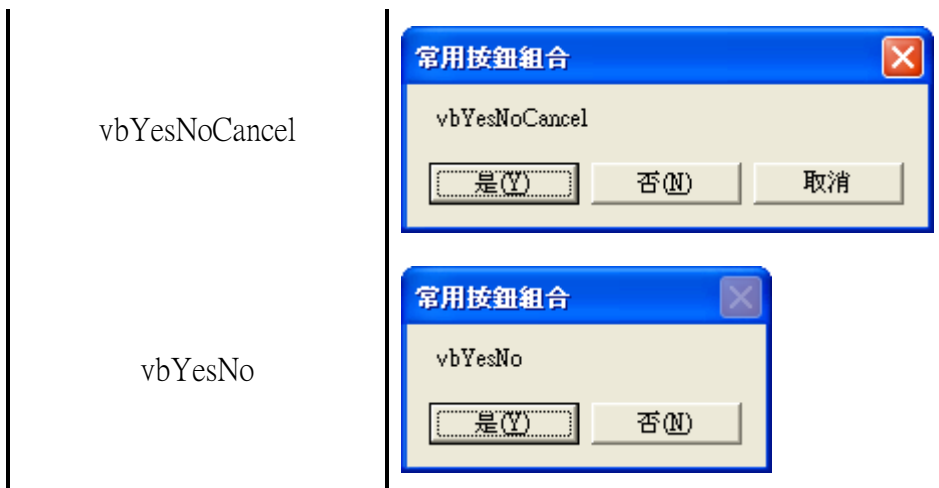
小幫手特區：MsgBox 的按鈕及圖示組合

VB 提供了許多現成的按鈕以及圖示可幫助使用者辨識視窗的作用為何，以下我們將利用前面所說過的 MsgBox 函數範例來說明圖示及按鈕的用法：  
以下列出幾種常見的圖示

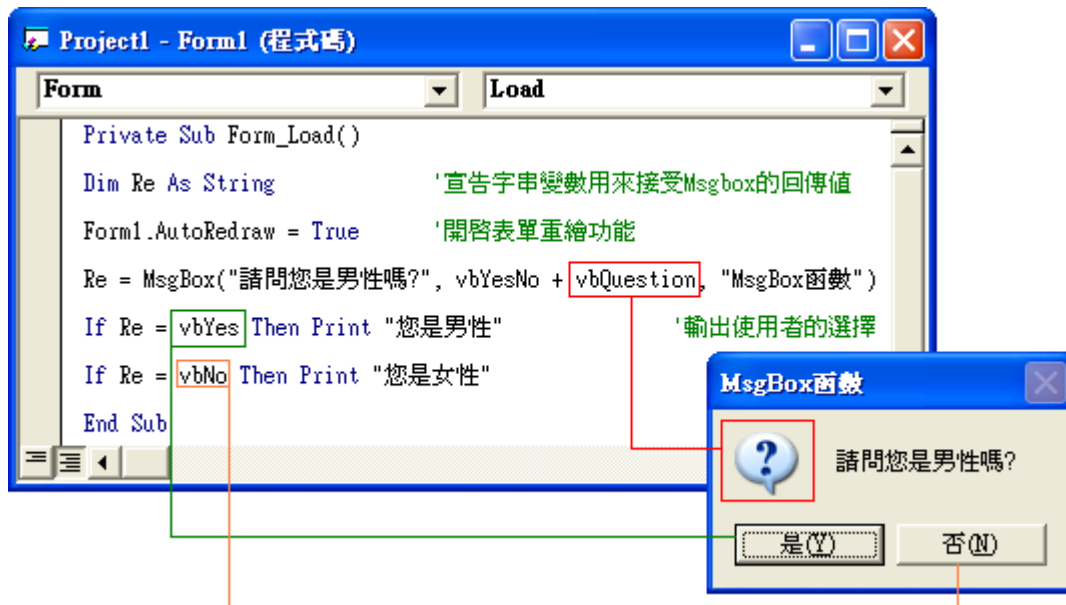
圖示名稱	vbInformation	vbQuestion	vbCritical	vbExclamation
圖示				

以下列出幾種常用的按鈕組合

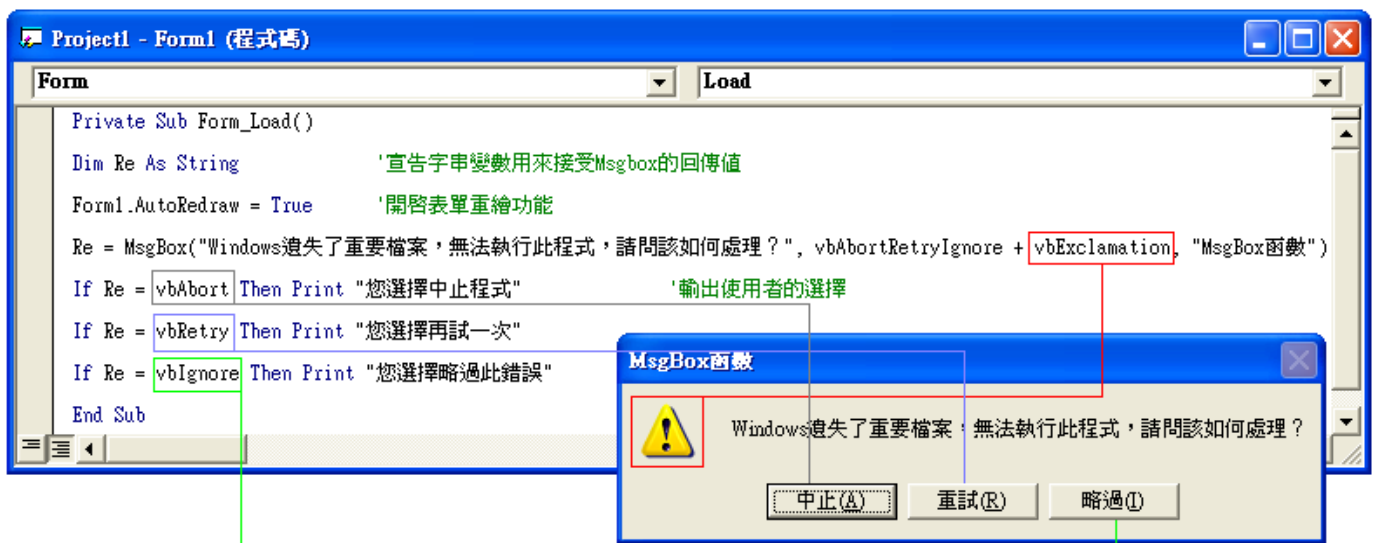
按鈕組合名稱	按鈕組合圖示
vbOKOnly	
vbOKCancel	
vbAbortRetryIgnore	



【範例 1】： 增加問號圖示



【範例 1】： 變更程式按鈕及圖示





## 5.2 亂數

系統內部有一個亂數表，當使用者需要使用亂數時，則從亂數中某一個位置開始存取，在 vb 中，亂數的值大於等於 0，小於 1 之間的任一小數。



變數 = Rnd

例：A = Rnd

由於程式每次一開始所選取的亂數表位置是相同的，所以避免出現亂數重覆的現象，必須使用 `Randomize` 來避免此種情形的發生。

在使用亂數時，由於值很小 ( $0 \leq \text{Rnd} < 1$ )，我們經常都會將亂數乘上特定倍數，但為了去除小數點，我們有幾個函數可以利用，以下將說明常用的幾個函數

`Fix()` 無條件捨去小數點

`Int()` 等於或小於的最大整數

【範例】：

`Fix(Rnd*10)`

`Fix(0.5) = 0`

`Int(0.5) = 0`

`Fix(-0.5) = 0`

`Int(-0.5) = - 1`

程式範例我們在下一節再一起舉例

## 5.3 陣列

陣列和一般的變數其實沒有什麼不同，只是陣列將 N 個型態相同的變數放在記憶體中連續的位置，加速程式的存取，也增加程式設計師的便利。

### 5.3.1 一維陣列

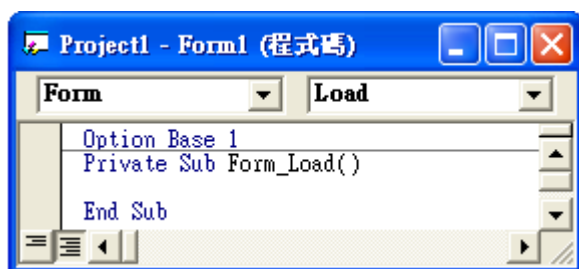
陣列的使用方法其實很容易，語法如下：

#### 語法 1

**Dim 陣列名稱(N) As 資料型態**

例：Dim A(10) As integer                   ‘代表宣告 A(0)~~A(10)共 11 個陣列位置

當然也可以強程式將陣列改爲由 1 開始，只要在一般中宣告『Option Base 1』即可將陣列的起始位置由 1 開始，如下所示



```
Project1 - Form1 (程式碼)
Form Load
Option Base 1
Private Sub Form_Load()
End Sub
```

陣列的第二種宣告方法如下：

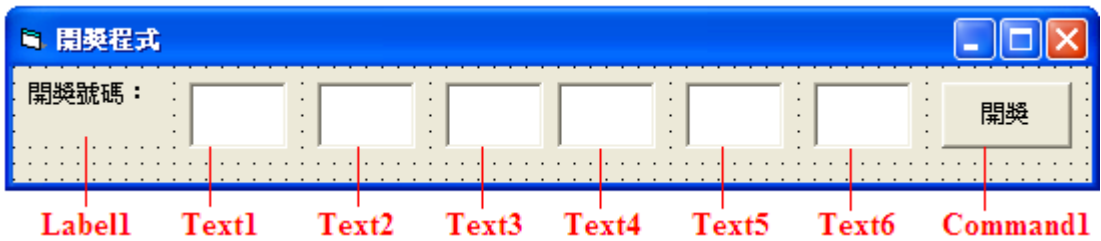
#### 語法 2

**Dim 陣列名稱(起始值 To 結束值) As 資料型態**

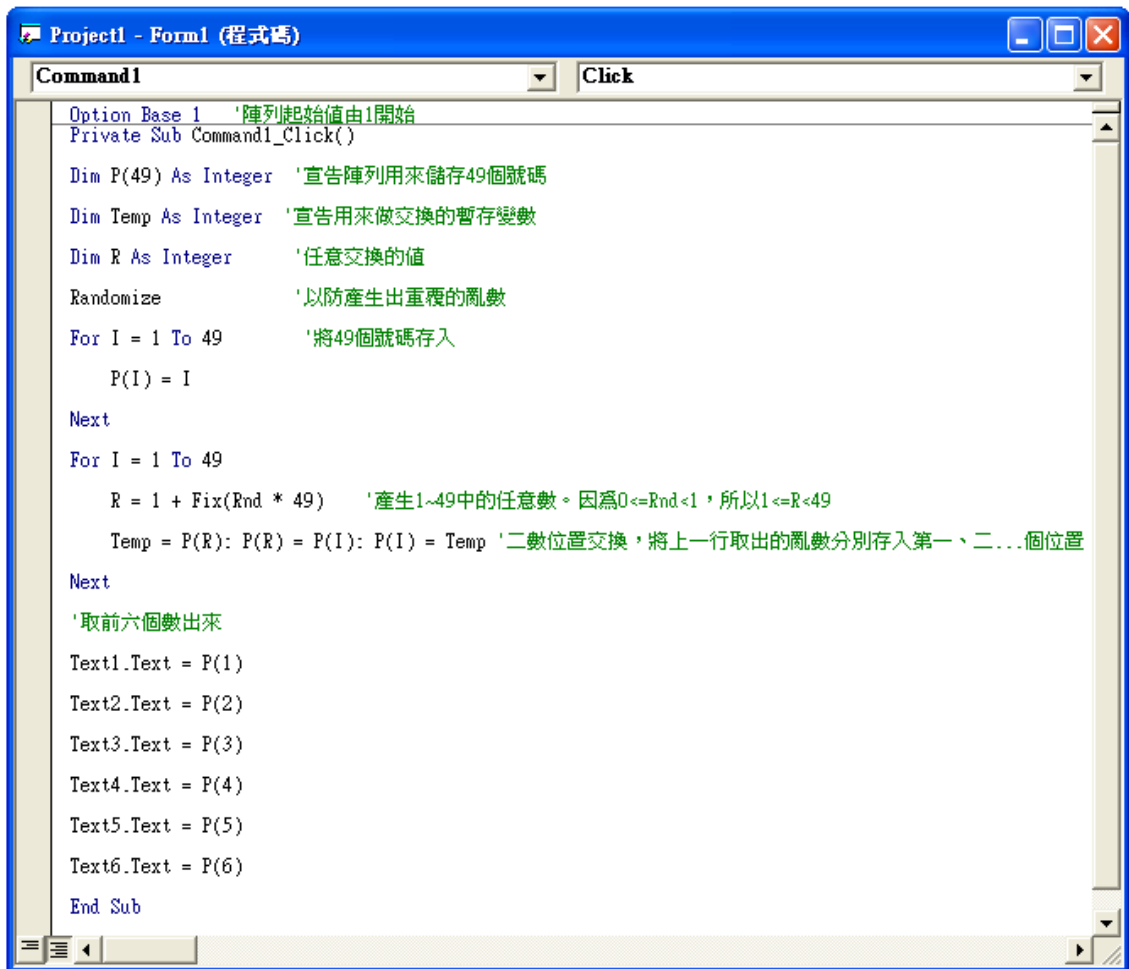
例：Dim A(2 to 10) As Integer           ‘代表宣告了 A(2)~~A(10)共 9 個陣列位置

## 【範例】：樂透開獎

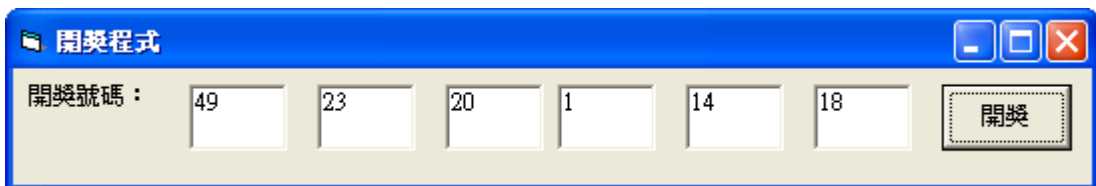
### 1. 表單配置



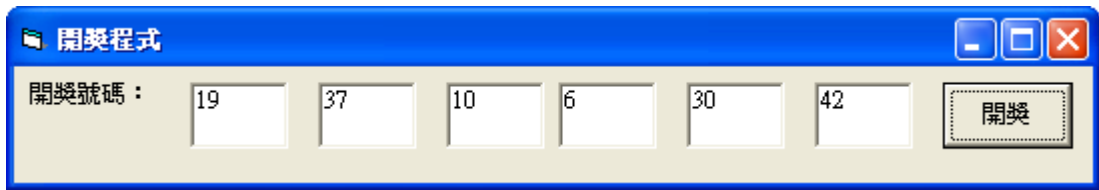
### 2. 撰寫程式碼



### 3. 執行結果，按下“開獎”後



每一次開獎的數字都不相同



### 5.3.2 多維陣列

多維陣列和一維陣列其實很像，只是一維陣列只有一行，而多維陣列多了幾列，其他都一樣，**多維陣列最大可以到達 60 維**，如下所示。

一維陣列 Dim A(4)

0	1	2	3	4
---	---	---	---	---

二維陣列 Dim A(1,4)

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4

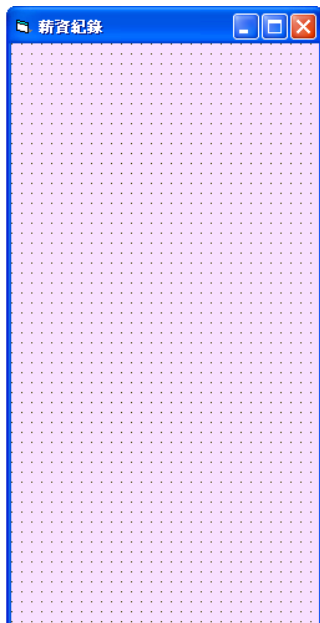
當然可以用到三維~~四維~~都沒有關係，例如五維陣列的宣告方式就是



**Dim 陣列名稱(N1,N2,N3,N4,N5) As 資料型態**

**【範例】**：我們要紀錄 3 年當中，每個月的收入，即可以多維陣列來紀錄

#### 1. 版面配置



## 2. 撰寫程式碼

Option Base 1

'陣列由 1 開始(一定只能放在『一般宣告』區)

---

Private Sub Form\_Load()

Dim GetMoney(3, 12) As Currency       '宣告一個二維陣列，型態為貨幣

Form1.AutoRedraw = True               '開啓表單重繪功能

'=====開始輸入薪資=====

GetMoney(1, 1) = 11000

GetMoney(1, 2) = 12000

GetMoney(1, 3) = 13000

GetMoney(1, 4) = 14000

GetMoney(1, 5) = 15000

GetMoney(1, 6) = 16000

GetMoney(1, 7) = 17000

GetMoney(1, 8) = 18000

GetMoney(1, 9) = 19000

GetMoney(1, 10) = 11000

GetMoney(1, 11) = 11100

GetMoney(1, 12) = 11200

GetMoney(2, 1) = 21000

GetMoney(2, 2) = 22000

GetMoney(2, 3) = 23000

GetMoney(2, 4) = 24000

GetMoney(2, 5) = 25000

GetMoney(2, 6) = 26000

GetMoney(2, 7) = 27000



End If

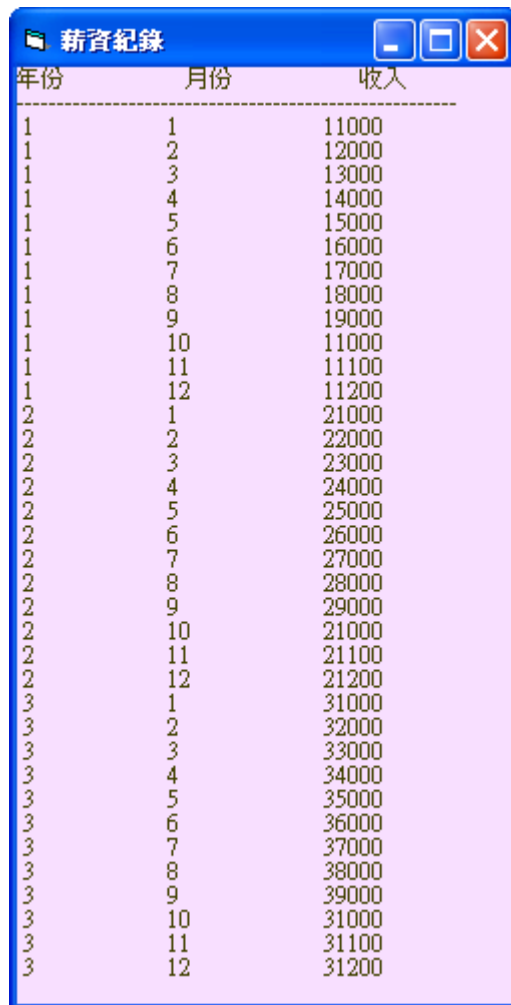
Next j

Next i

End Sub

---

#### 4. 執行結果

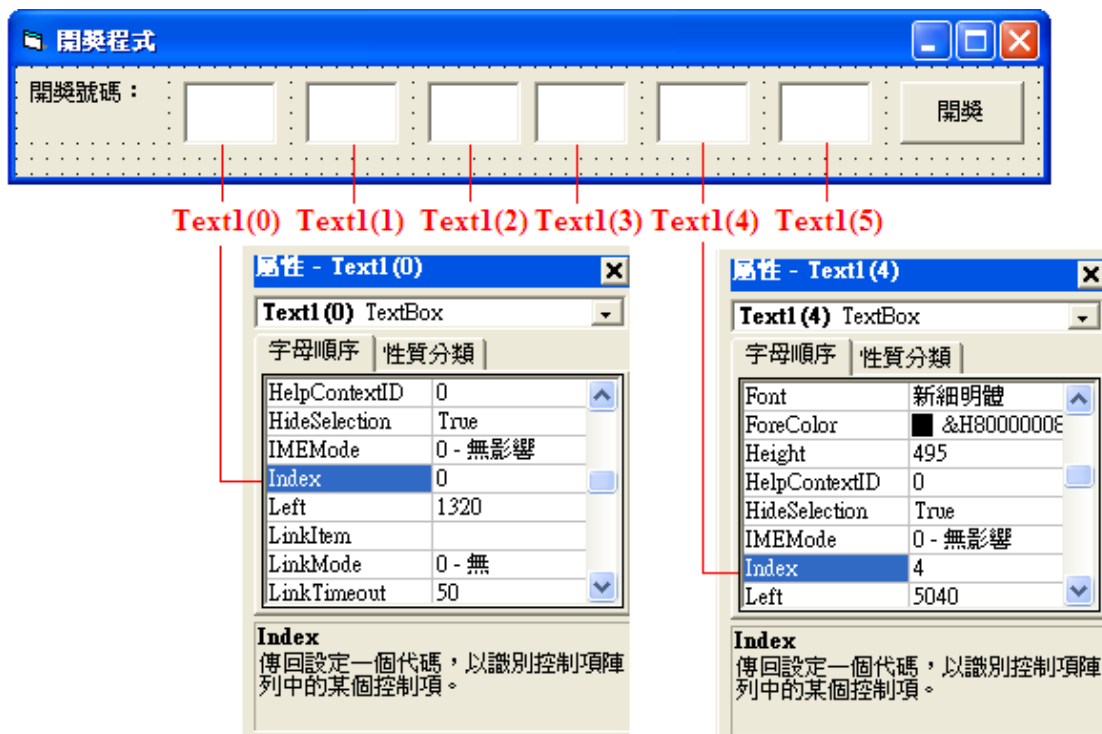


年份	月份	收入
1	1	11000
1	2	12000
1	3	13000
1	4	14000
1	5	15000
1	6	16000
1	7	17000
1	8	18000
1	9	19000
1	10	11000
1	11	11100
1	12	11200
2	1	21000
2	2	22000
2	3	23000
2	4	24000
2	5	25000
2	6	26000
2	7	27000
2	8	28000
2	9	29000
2	10	21000
2	11	21100
2	12	21200
3	1	31000
3	2	32000
3	3	33000
3	4	34000
3	5	35000
3	6	36000
3	7	37000
3	8	38000
3	9	39000
3	10	31000
3	11	31100
3	12	31200

### 5.3.3 控制項陣列

看了上述的程式，你會發現，陣列真的很方便，尤其是用在處理大量資料時，較單一命名的變數方便許多。其實陣列的好處不只於此，陣列的觀念也可以應用在控制項中。在此，我們將上一節的開獎程式，改寫成控制項陣列。控制項陣列很簡單，以下面要用的範例來看，只要把第一個 Text1 複製、貼上，即會出現一個視窗寫著”已經有一個名為’Text1’的控制項，您是否想要建立一個控制項陣列？”

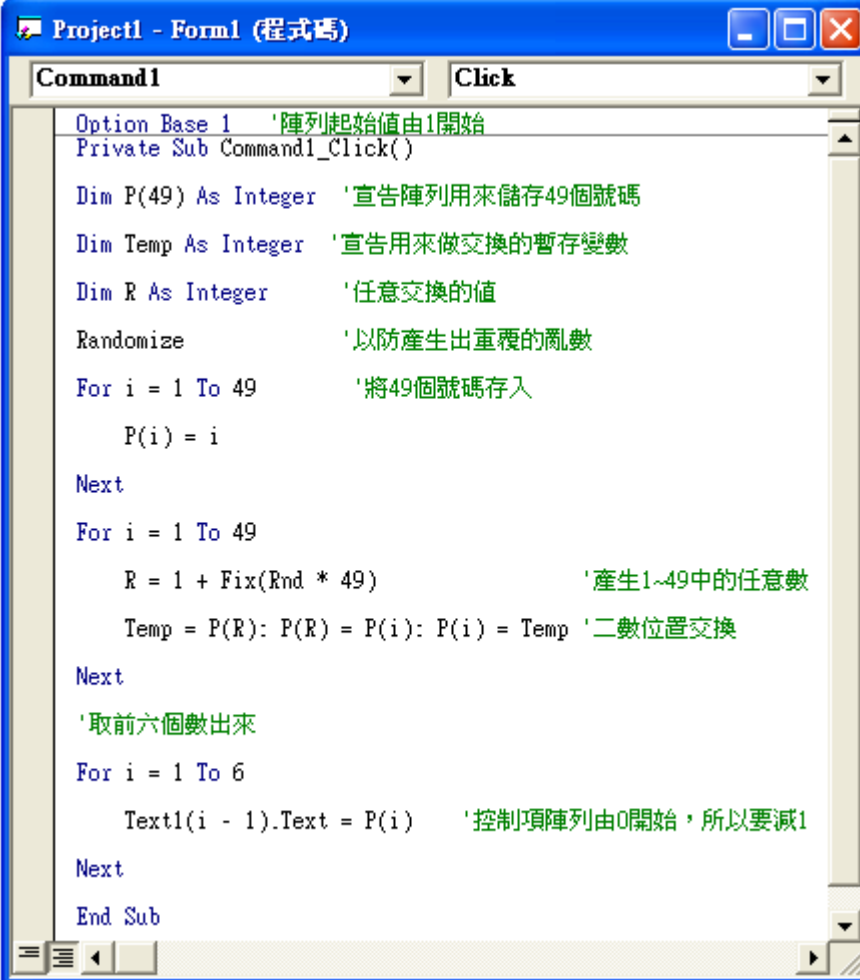
#### 1. 版面



※Index 屬性是控制項陣列中最重要的一环，它代表著控制項在陣列中的位置，靠著它，您可以將控制項當成陣列般方便的使用。



## 2. 撰寫程式碼



```
Option Base 1 '陣列起始值由1開始
Private Sub Command1_Click()
    Dim P(49) As Integer '宣告陣列用來儲存49個號碼
    Dim Temp As Integer '宣告用來做交換的暫存變數
    Dim R As Integer '任意交換的值
    Randomize '以防產生出重覆的亂數
    For i = 1 To 49 '將49個號碼存入
        P(i) = i
    Next
    For i = 1 To 49
        R = 1 + Fix(Rnd * 49) '產生1~49中的任意數
        Temp = P(R): P(R) = P(i): P(i) = Temp '二數位置交換
    Next
    '取前六個數出來
    For i = 1 To 6
        Text1(i - 1).Text = P(i) '控制項陣列由0開始，所以要減1
    Next
End Sub
```

※ 您一定可以發現，程式是不是又少了幾行，而且可以用 For 迴圈來簡化程式的長度，不論是在撰寫或維護上都較為容易。將來若撰寫大型程式時，將會發現，程式碼愈短，就等於程式的容量愈小，其中的優勢，是很明顯的。

### 5.3.4 動態陣列

所謂的動態陣列指的是『在程式行中，可以任意增加或減少陣列的大小』，這樣可以節省不必要的記憶體空間浪費，例如當設計一個報名系統時，您無法事先知道報名的人數會有多少，假設預計 1000 個，結果報名人數超出，就會發生無法報名的情況。如果報名人數只有 990 個，那麼便浪費了十個陣列空間。這時候動態陣列的好處就顯現出來了。以下將說明動態的使用方法：



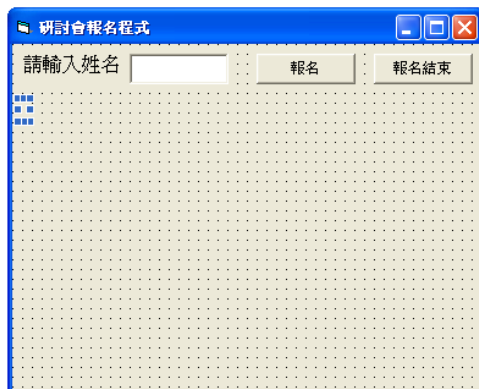
**語法** ReDim 陣列名稱(N) As 型態

改變陣列大小的方式，只要將 N 的值修改成想要的值即可。但陣列中的資料會全部消失，如欲保留陣列中的資料，必須加入一個保留字『Preserve』，使用方法如下：

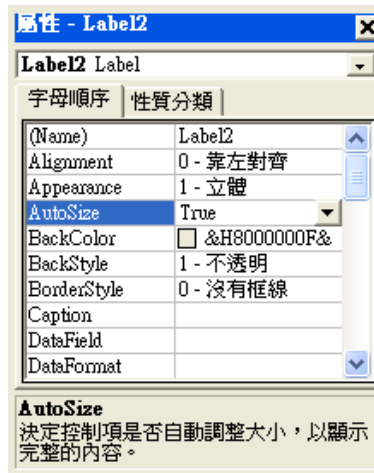
**ReDim Preserve 陣列名稱 (N) As 型態**

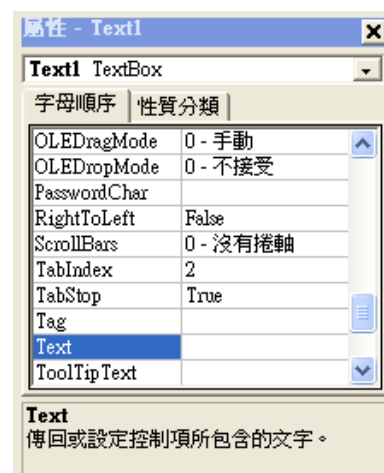
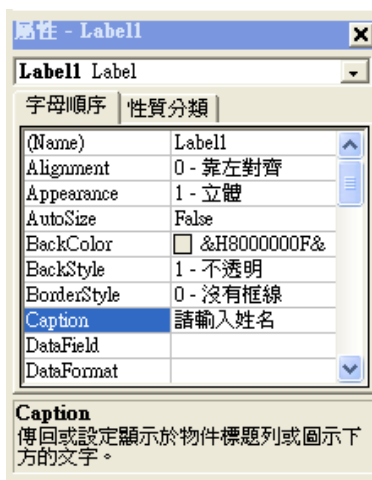
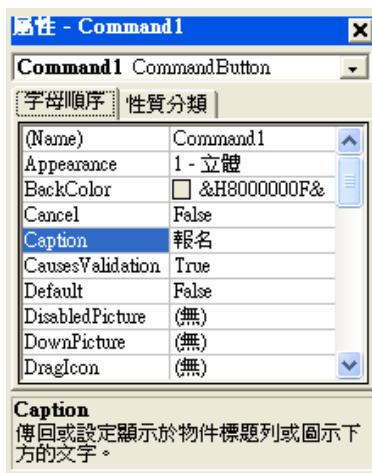
**【範例】：**研討會報名程式

#### 1. 版面配置



#### 2. 屬性





### 3. 撰寫程式碼

Dim i As Integer '計算人數的整數變數宣告  
 Dim People\_Name() As String '儲存姓名的字串變數宣告

```
Private Sub Command1_Click() '報名
  People_Name(i) = Text1.Text '將輸入的姓名存入陣列
  Text1.Text = "" '將清掉 Text1 中已輸入的姓名
  i = i + 1 '人數加 1
  ReDim Preserve People_Name(i) As String '重設陣列的上限
End Sub
```

```
Private Sub Command2_Click() '報名結束
  Dim j As Integer '迴圈要用的變數
  For j = 0 To i - 1 '將陣列中的姓名輸出
    Label2.Caption = Label2.Caption + People_Name(j) + Chr(10) + Chr(13)
  Next
```

輸出人數

```
Label2.Caption = Label2.Caption + Chr(10) + Chr(13) + "報名人數：" + Str(i)  
End Sub
```

---

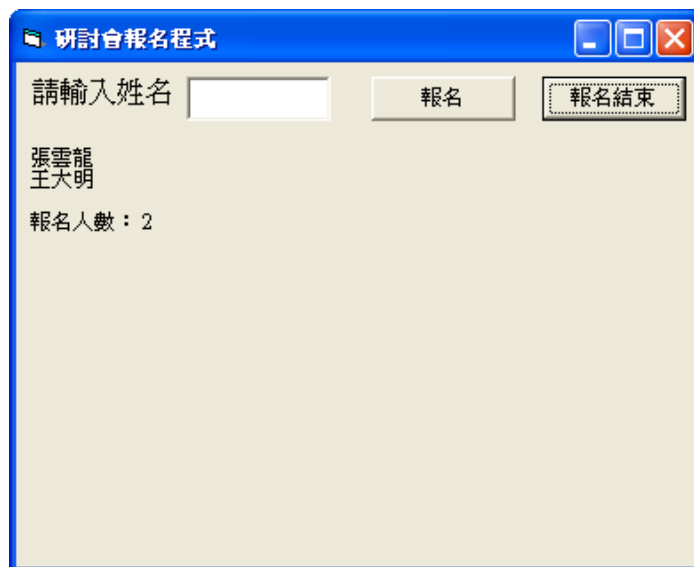
```
Private Sub Form_Load()
```

```
ReDim Preserve People_Name(0) As String 將變數的上限為零(預設值)
```

```
End Sub
```

---

#### 4.執行結果





## 小幫字特區：Chr()函數 、Asc () 函數

### ASCII (American Standard Code for Information Interchange)

#### 美國資訊交換標準碼

電腦系統都有一套制定每個符號所對應的代碼 code，ASCII 碼是一套 7 位元編碼系統，代表英國字元的編碼，ASCII 是從 0 一直編到 127，總共有 128 個字元。


ASCII 碼是電腦的內碼，每一個內碼都對應到一個符號，這些符號有英文字母。例：大寫 A 對應的 ASCII 碼是 65、數字 7 對應的 ASCII 碼是 55、特殊符號#對應的 ASCII 碼是 35、還有系統控制碼像是數字 DELETE 對應的 ASCII 碼是 127。多數電腦都用 ASCII 碼表現純文字 Text-Only，便利資料能在各電腦間做交換。

第一部 ASCII 碼大致可以分作三部分組成。

由 00H 到 1FH 共 32 個，一般用來通訊或作為控制之用，有些字元可顯示於螢幕，有些則無法顯示在螢幕上，但能看到其效果(例如換行字元、歸位字元)。如下表：

ASCII 碼					ASCII 碼				
十進位	十六進位	字元	控制字元	意義	十進位	十六進位	字元	控制字元	意義
000	00		NULL	空字元	016	10	▶	DLE	
001	01	☺	SOH		017	11	◀	DC1	
002	02	☺	STX		018	12	↑	DC2	
003	03	♥	ETX		019	13	!!	DC3	
004	04	♦	EOT		020	14	¶	DC4	
005	05	♣	ENQ		021	15	§	NAK	
006	06	♠	ACK		022	16	-	SYN	
007	07	•	BELL	鈴聲	023	17	↓	ETB	
008	08	■	BS	倒退鍵	024	18	↑	CAN	
009	09		HT	定位鍵	025	19	↓	EM	
010	0A		LF	line feed	026	1A	→	SUB	檔案結束
011	0B	♂	VT	home	027	1B	←	ESC	escape
012	0C	♀	FF	form feed	028	1C	L	FS	向右游標
013	0D		CR	carriage return	029	1D	↔	GS	向左游標
014	0E	♪	SO		030	1E	▲	RS	向上游標
015	0F	☼	SI		031	1F	▼	US	向下游標

第二部分是 20H 到 7FH 共 96 個，這 95 個字元是用來表示阿拉伯數字、英文字母大小寫和底線、括號等符號，都可以顯示在螢幕上。如下表：

ASCII 碼		字元	ASCII 碼		字元	ASCII 碼		字元	ASCII 碼		字元
十進位	十六進位		十進位	十六進位		十進位	十六進位		十進位	十六進位	
032	20		056	38	8	080	50	P	104	68	h
033	21	!	057	39	9	081	51	Q	105	69	i
034	22	"	058	3A	:	082	52	R	106	6A	j
035	23	#	059	3B	;	083	53	S	107	6B	k
036	24	\$	060	3C	<	084	54	T	108	6C	l
037	25	%	061	3D	=	085	55	U	109	6D	m
038	26	&	062	3E	>	086	56	V	110	6E	n
039	27	'	063	3F	?	087	57	W	111	6F	o
040	28	(	064	40	@	088	58	X	112	70	p
041	29	)	065	41	A	089	59	Y	113	71	q
042	2A	*	066	42	B	090	5A	Z	114	72	r
043	2B	+	067	43	C	091	5B	[	115	73	s
044	2C	,	068	44	D	092	5C	\	116	74	t
045	2D	-	069	45	E	093	5D	]	117	75	u
046	2E	.	070	46	F	094	5E	^	118	76	v
047	2F	/	071	47	G	095	5F	_	119	77	w
048	30	0	072	48	H	096	60	`	120	78	x
049	31	1	073	49	I	097	61	a	121	79	y
050	32	2	074	4A	J	098	62	b	122	7A	z
051	33	3	075	4B	K	099	63	c	123	7B	{
052	34	4	076	4C	L	100	64	d	124	7C	
053	35	5	077	4D	M	101	65	e	125	7D	}
054	36	6	078	4E	N	102	66	f	126	7E	~
055	37	7	079	4F	O	103	67	g	127	7F	

第三部分由 80H 到 0FFH 共 128 個字元，一般稱為『擴充字元』，這 128 個擴充字元是由 IBM 制定的，並非標準的 ASCII 碼。這些字元是用來表示框線、音標和其他歐洲非英語系的字母。

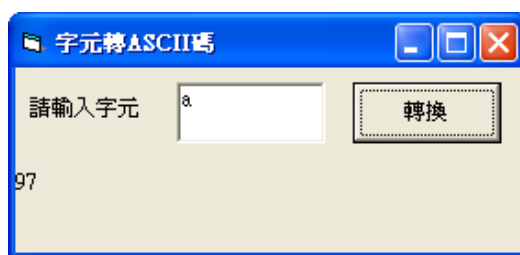
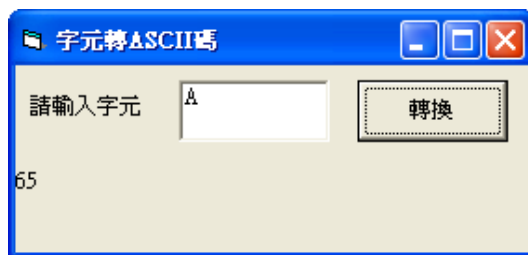
ASCII 碼			ASCII 碼			ASCII 碼			ASCII 碼		
十進位	十六進位	字元	十進位	十六進位	字元	十進位	十六進位	字元	十進位	十六進位	字元
128	80	Ç	160	A0	á	192	C0	Ł	224	E0	α
129	81	ü	161	A1	í	193	C1	ł	225	E1	β
130	82	é	162	A2	ó	194	C2	ṽ	226	E2	γ
131	83	â	163	A3	ú	195	C3	ł̇	227	E3	π
132	84	ä	164	A4	ñ	196	C4	—	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	+	229	E5	σ
134	86	â	166	A6	ª	198	C6	ł̈	230	E6	μ
135	87	ç	167	A7	º	199	C7	ł̋	231	E7	ṽ
136	88	è	168	A8	¿	200	C8	Ł	232	E8	φ
137	89	ë	169	A9	ƒ	201	C9	ł̌	233	E9	θ
138	8A	è	170	AA	¬	202	CA	ł̍	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	ł̎	235	EB	δ
140	8C	î	172	AC	¼	204	CC	ł̏	236	EC	∞
141	8D	ì	173	AD	ı	205	CD	=	237	ED	φ
142	8E	Ä	174	AE	<<	206	CE	ł̐	238	EE	€
143	8F	Å	175	AF	>>	207	CF	ł̑	239	EF	∩
144	90	É	176	B0	⋮	208	D0	ł̒	240	F0	≡
145	91	æ	177	B1	⋮	209	D1	ł̓	241	F1	±
146	92	Æ	178	B2	⋮	210	D2	ł̔	242	F2	≥
147	93	ô	179	B3		211	D3	ł̕	243	F3	≤
148	94	ö	180	B4	┆	212	D4	ł̖	244	F4	┌
149	95	ò	181	B5	┆	213	D5	ł̗	245	F5	└
150	96	û	182	B6	┆	214	D6	ł̘	246	F6	÷
151	97	ù	183	B7	┆	215	D7	ł̙	247	F7	≈
152	98	ÿ	184	B8	┆	216	D8	ł̚	248	F8	°
153	99	ö	185	B9	┆	217	D9	ł̛	249	F9	•
154	9A	ü	186	BA		218	DA	ł̜	250	FA	·
155	9B	ƒ	187	BB	┆	219	DB	ł̝	251	FB	√
156	9C	f	188	BC	┆	220	DC	ł̞	252	FC	ⁿ
157	9D	¥	189	BD	┆	221	DD	ł̟	253	FD	²
158	9E	₣	190	BE	┆	222	DE	ł̠	254	FE	■
159	9F	f	191	BF	┆	223	DF	ł̡	255	FF	

在程式當中，我們所使用的 Chr(10) 代表的意義為換行(line feed)也就是跳到下一行中的相同位置。而 Chr(13)代表的意義為回到行首，如下文章中，假設游標位置目前在第一行的『黃』字右方，則輸入 Chr(10)則會將游標位置移到第二行的『疑』字右方，此時若再輸入 Chr(13)，則會將游標位置移到『床』的左方。

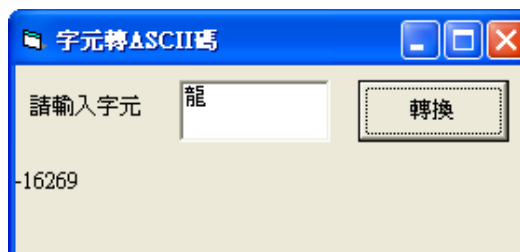
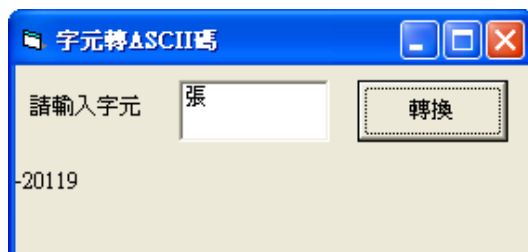
白日依山盡，黃河入海流，欲窮千里目，更上一層樓。  
床前明月光，疑似地上霜，舉頭望明月，低頭吃便當。

ASCII()則是將括號中的字元轉換成其對應的 ASCII 碼，當然，只能轉換 Ascii 碼所存在的字元。

如下程式範例，大小寫的 a 在 Ascii 碼中所代表的數字並不相同



中文字的編碼並不使用 Ascii 碼，所以會發生錯誤。





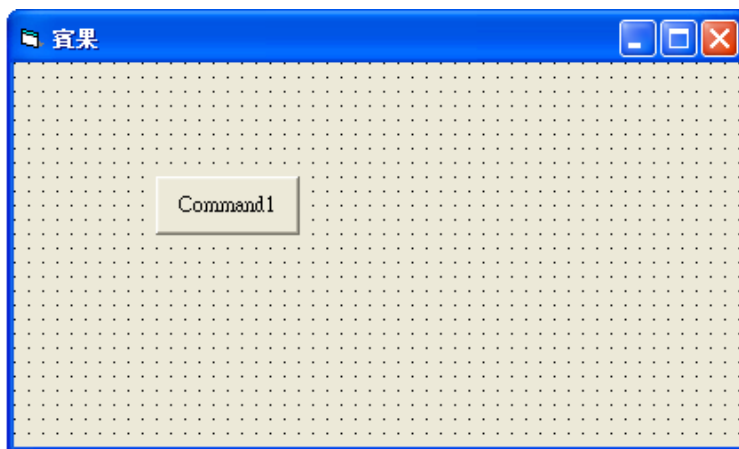
## 5.5 Load、Unload

當我們需要大量的控制項時，我們若是一個又一個的編排實在很耗費時間，我們可以利用控制項陣列的特性，適當的使用 Load 指令來呼叫控制項陣列，節省許多時間。Load 出來的物件而不會直接顯示在表單上，程式設計師必須在適當的時間將其 Visible 屬性設為 True 才會顯示在表單上。

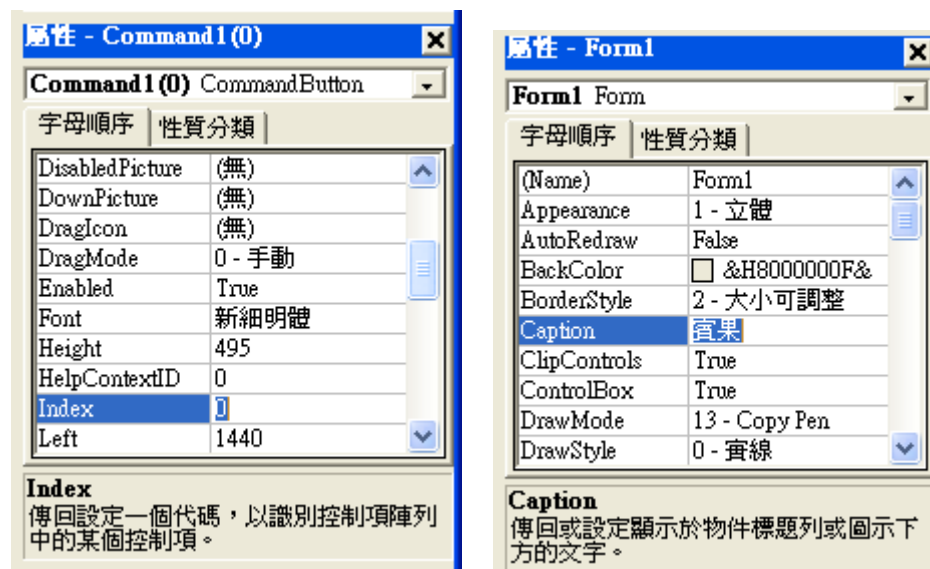
Unload 指令則是將此物件由記憶體中移除，但 Unload 會直接將物件從表單上移除。

**【範例】：**設計一個賓果遊戲 1~25 的程式

### 1. 表單配置



### 2. 屬性設定



### 3. 撰寫程式碼

```
Private Sub Command1_Click(Index As Integer)
```

```
Dim Bingo As Integer '計算同一條線上有幾個數字被點選
```

```
Command1(Index).Enabled = False '將按下的數字變成灰色(不能再按)
```

=====以下程式碼是用來檢查有沒有連成直線=====

For i = 0 To 20 Step 5

    If Command1(i).Enabled = False Then                   '假如被檢查的數字是灰色

        Bingo = Bingo + 1                                   '被點選的數字加 1

    End If

Next

If Bingo = 5 Then   '假如一條線上有五個被點選就賓果

    MsgBox "賓果"

    Unload Me   '將表單自記憶體中移除

End If

Bingo = 0   '將 Bingo 清除內容

For i = 1 To 21 Step 5

    If Command1(i).Enabled = False Then

        Bingo = Bingo + 1

    End If

Next

If Bingo = 5 Then

    MsgBox "賓果"

    Unload Me

End If

Bingo = 0

For i = 2 To 22 Step 5

    If Command1(i).Enabled = False Then

        Bingo = Bingo + 1

    End If

Next

```
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
For i = 3 To 23 Step 5
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
For i = 4 To 24 Step 5
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
```

'=====以下程式碼是用來檢查有沒有連成橫線=====

```
For i = 0 To 4
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
For i = 5 To 9
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
For i = 10 To 14
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
```

```
    MsgBox "賓果"

    Unload Me

End If

Bingo = 0

For i = 15 To 19

    If Command1(i).Enabled = False Then

        Bingo = Bingo + 1

    End If

Next

If Bingo = 5 Then

    MsgBox "賓果"

    Unload Me

End If

Bingo = 0

For i = 20 To 24

    If Command1(i).Enabled = False Then

        Bingo = Bingo + 1

    End If

Next

If Bingo = 5 Then

    MsgBox "賓果"

    Unload Me

End If

Bingo = 0

'=====以下程式碼是用來檢查有沒有連成斜線=====

For i = 0 To 24 Step 6
```

```
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
For i = 4 To 20 Step 4
    If Command1(i).Enabled = False Then
        Bingo = Bingo + 1
    End If
Next
If Bingo = 5 Then
    MsgBox "賓果"
    Unload Me
End If
Bingo = 0
End Sub
```

---

```
Private Sub Form_Load()
    Dim Number(25), R, Temp As Integer
    Randomize '避免亂數重覆
    Form1.ScaleWidth = 5600 '設定表單的大小
    Form1.ScaleHeight = 5600
```

```

For i = 1 To 24                '載入另外 24 個按鍵
    Load Command1(i)
    Command1(i).Visible = True '並將其顯示在表單上
Next
For i = 1 To 25                '將 1~25 放入陣列中
    Number(i) = i
Next
For i = 1 To 25                '將陣列中的數字任意洗牌
    R = 1 + Int(Rnd * 25)
    Temp = Number(R): Number(R) = Number(i): Number(i) = Temp
Next
For i = 0 To 24                '設定 25 個按鍵的屬性
    Command1(i).Caption = Number(i + 1) '將 1 到 25 顯示在按鍵上
    Command1(i).Width = 800        '按鍵的寬
    Command1(i).Height = 800       '按鍵的高
    Command1(i).Left = 800 + (i Mod 5) * 800 '按鍵在橫向的排列位置
    Command1(i).Top = 800 + (i \ 5) * 800   '按鍵在垂直的排列位置
Next
End Sub

```

#### 4. 執行結果

窗体标题: 宜果

11	15	3	7	19
14	13	22	24	18
4	16	1	10	23
17	9	2	21	6
12	5	25	20	8