

Arrays: *part1*

An array is a collection of simple variables of the same type to which the computer can efficiently assign a list of values. An array is a consecutive group of memory locations that all have the same name. To refer to a particular location or element in the array, we specify the array name and the array element position number. The Individual elements of an array are identified using an **index**.

Declaring arrays:

Arrays may be declared as **Public** (in a code module), module or local. Module arrays are declared in the general declarations using keyword **Dim** or **Private**. Local arrays are declared in a procedure using **Dim**.

There are two types of arrays in Visual Basic namely: **Fixed size, Dynamic Size**.

Fixed-Size Array: The size of array always remains the same size doesn't change during the program execution. When an upper bound is specified in the declaration. This type is divide to (One Dimensional Array & Multi-Dimensional Array).

Dynamic array : The size of the array can be changed at the run time

For example:

Public A(14) As Integer	“ an array of 15 elements “
Public a(100) As Integer	“ an array of 101 elements “
Dim Student(20) As String	“ an array of 21 elements “
Dim Counter(9) As Integer	“ an array of 10 elements “

Another form:

Dim Num(5) As Integer

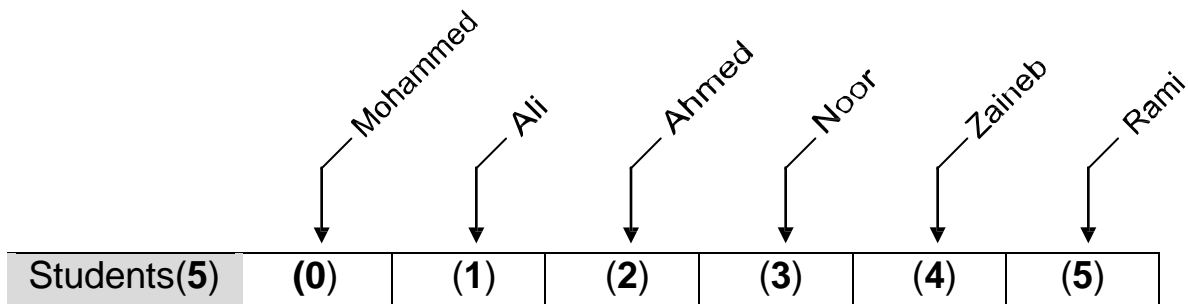
Num is the name of the array, and the number **5** included in the brackets is the upper limit of the array, means the above declaration creates an array with **6** elements, with index numbers running from **0** to **5** .

Num(5)	Num(0)	Num(1)	Num(2)	Num(3)	Num(4)	Num(5)
Value	3.1	7	-10	2	0.33	6

If we want to specify the lower limit, then the brackets should include both the lower and upper limit along with the **To** keyword, an example for this is given below.

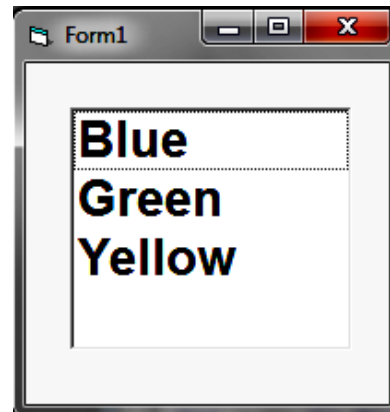
Dim Num(0 To 5) As Integer is equal to **Dim Num(5) As Integer**

Another form: **Dim students(5) As String**



Example1: Print three colors in a **List Box** using fixed **Array**.

Solution:



Code:

Form1:

```
Private Sub Form_Load()
    Dim A(2) As String
    Dim i As Integer
    For i = 0 To 2
        a(0) = "Blue"
        a(1) = "Green"
        a(2) = "Yellow"
        List1.AddItem A(i)
    Next
End Sub
```

Example2: Print four students names in a **List Box** using fixed **Array**.

Solution:

Code:

```
Dim student(3) As String
Dim n As Integer
```

Form1:

```
Private Sub Form_Load()
```

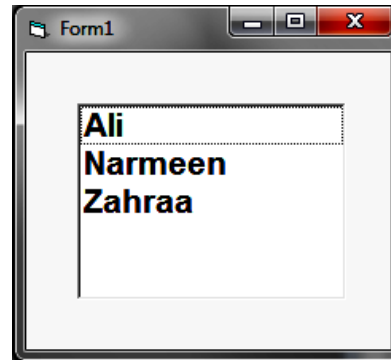
```
For n = 0 To 3
```

```
student(n) = InputBox("Enter the student name", "Enter Name", "")
```

```
List1.AddItem student(n)
```

```
Next
```

```
End Sub
```



Example3: In one dimension **Array** with **10** elements is entered into **ListBox**. Write a program to find the location **i** such that **A (i)** contains the largest value. Display the Largest value and the location into **TextBoxes**.

Solution:

Code:

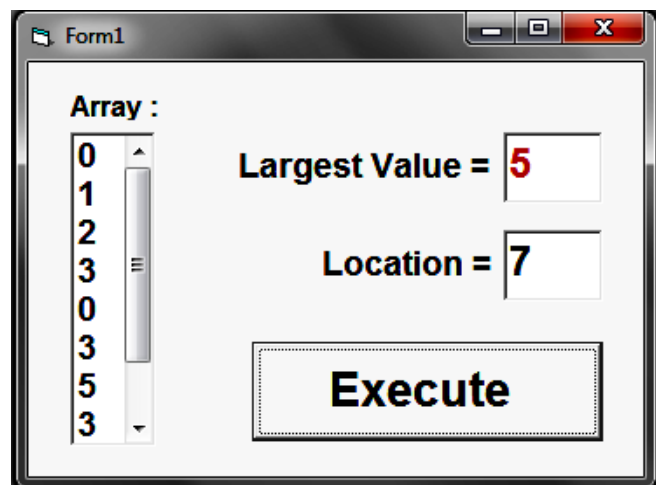
```
Dim A(9) As Integer
Dim i, Location, Max As Integer
```

Command1:

```
Private Sub Command1_Click()
```

```
List1.Clear
```

```
Text1.Text = " "
```



Text2.Text = ""

For i = 0 To 9
A(i) = i * Rnd
List1.AddItem A(i)
Next i



First loop is for adding ten random numbers to the List Box.

Max = A(0)
Location = 0



Assume that **Max** is equal to the first value in the Array just for making a comparison as in the next loop.

For i = 0 To 9
If A(i) > Max Then
Max = A(i)
Location = i + 1
End If
Next i



Second loop is for making comparison between first value in the Array and the next values depending on (i) and note that when Max = A(0) then Location=0+1=1 means this is the first Location for the value lies in the first store "0" in the Array and so-on.

Text1.Text = Max
Text2.Text = Location



Printing

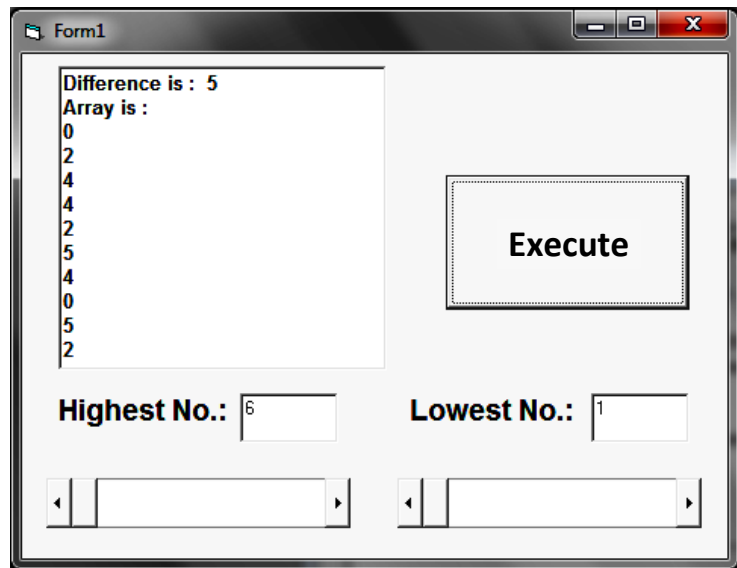
End Sub

Example4: Generate a fixed **Array** of a set of ten random numbers depending on the difference between highest and lowest numbers using **Scroll Bars**. Print the result in a **Text Box**.

Solution:

Code:

```
Dim i, Difference As Integer
Dim Number(9) As Integer
```



Command1:

```
Private Sub Command1_Click()
```

```
Text1.Text = " "
```

```
Difference = HScroll1.Value - HScroll2.Value
```

Means Lowest No. \geq Highest No.
and this is not correct.

```
If Difference < 1 Then
```

```
MsgBox "The low number must be smaller ", vbCritical
```

```
Else
```

Definition of Chr() at
the end of this example

```
Text1.Text = Text1.Text & "Difference is : " & Difference & Chr(13) & Chr(10)
```

```
Text1.Text = Text1.Text & "Array is : " & Chr(13) & Chr(10)
```

```
For i = 0 To 9
```

```
Number(i) = Difference * Rnd
```

```
Text1.Text = Text1.Text & Number(i) & Chr(13) & Chr(10)
```

```
Next i
```

```
End If
```

```
End Sub
```

HScroll1:

```
Private Sub HScroll1_Change()  
Text2.Text = HScroll1.Value  
End Sub
```

HScroll2:

```
Private Sub HScroll2_Change()  
Text3.Text = HScroll2.Value  
End Sub
```

Note: **Text1.text** must be changed to **Multiline** in the properties

Note: Chr() is a short word of Character. In VB, there are some characters that do not appear on the keyboard.

Chr(13) which is mean (**Carriage Return**) which is also mean “return to the beginning of the current line”

العودة لبداية السطر

Chr(10) which is mean (**Line Feed**) which is also mean “advance downward to the next line”

النزول لسطر جديد

في الفيچوال بيسك يجب العودة لبداية السطر و ثم النزول الى سطر جديد عند العمل داخل TextBox