

VISUAL BASIC FOR APPLICATIONS

Damiano SOMENZI

SCHOOL OF ECONOMICS AND MANAGEMENT

Advanced Computer Skills

`damiano.somenzi@unibz.it`



Week 10

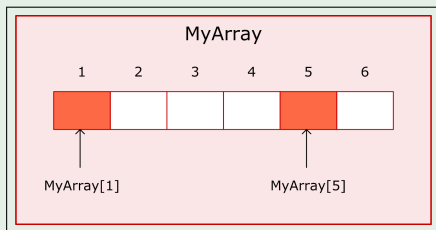
① DATA STRUCTURE: ARRAY

② DATA STRUCTURE: ARRAY AS ARGUMENT

ARRAY

An array holds a set of values of the **same data type**

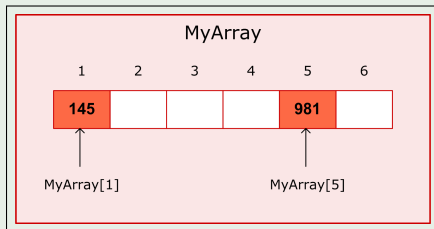
- An array is a **single variable** with many compartments to store values, while a typical variable has only one storage compartment in which it can store only one value



- You refer to the array as a whole when you want to refer to all the values it holds (i.e. "MyArray")
- You refer to its individual elements when you want to refer to an identified value (i.e. "MyArray[5]")

ARRAY EXAMPLE

To store six numbers (integers) you can create an array variable with six compartments, rather than creating six different variables. Each compartment contains one value



ARRAY DECLARATION

- An array variable is a collection of variables that uses the same name but are distinguished by an index value
- Array variables are declared the same way as other variables, using the `Dim` statements. **Generally** it is specified the size of the array
- An array holding 10 integers (fixed-size) is declared as

```
Dim Numbers(1 To 10) As Integer
```

- 1 The array variable is `Numbers`
- 2 It has size 10 (compartments)
- 3 The Lower Bound of the array is 1 and the Upper Bound is 10
- 4 `Numbers(1)`, `Numbers(2)`, .. , `Numbers(10)` variables are Integer data type variables

USING ARRAYS

Code	Array Content	Display
Dim A(1 To 5) As Integer	0 0 0 0 0	
A(1) = 123	123 0 0 0 0	
A(2) = 98	123 98 0 0 0	
A(3) = 341	123 98 341 0 0	
MsgBox(A(2))	123 98 341 0 0	98
MsgBox(A(4))	123 98 341 0 0	0
MsgBox(A(6))	123 98 341 0 0	Out of Range

USING ARRAY

EXAMPLE ONE

```
Sub MyArray()  
    Dim A(1 To 4) As Integer  
    A(1) = 88  
    A(4) = 122  
    MsgBox (A(1) & " - " & A(2) & " - " & A(3) & " - " & A(4) )  
End Sub
```

EXAMPLE TWO

```
Sub MyArray()  
    Dim A(1 To 4) As Integer  
    Dim i As Integer  
    For i = 1 To 4  
        A(i) = Val(InputBox("Enter an Integer"))  
    Next i  
    MsgBox (A(1) & " - " & A(2) & " - " & A(3) & " - " & A(4) )  
End Sub
```

EXAMPLE THREE

```
Sub MyArray()  
    Dim A(1 To 4) As Integer  
    Dim w As Worksheet  
    Dim r As Integer, c As Integer  
    Set w = Worksheets(1)  
    c = 3  
    For r = 1 To 4  
        A(r) = w.Cells(r, c).Value  
    Next r  
    MsgBox (A(1) & " - " & A(2) & " - " & A(3) & " - " & A(4))  
End Sub
```

EXAMPLE FOUR

```
Sub MyArray()  
    Dim A(1 To 4) As Integer  
    Dim w As Worksheet  
    Set w = Worksheets(1)  
    Dim i As Integer  
    Dim r As Integer, c As Integer  
    For i = 1 To 4  
        A(i) = Val(InputBox("Enter an Integer"))  
    Next i  
    c = 2  
    For r = 1 To 4  
        w.Cells(r, c).Value = A(r)  
    Next r  
End Sub
```

EXAMPLE FIVE

```
Sub MyArray()  
    Dim A(1 To 4) As Integer  
    Dim i As Integer  
    For i = 1 To 4  
        A(i) = Val(InputBox("Enter an Integer"))  
    Next i  
    For i = 1 To 4  
        MsgBox ("A(" & i & ") -> " & A(i))  
    Next i  
End Sub
```

EXERCISE 1

Write a procedure **SetOfWords** that creates an array of 5 elements in order to store 5 words. The procedure asks the user for the 5 words, and then it stores them within the array. Furthermore it displays the words

SOLUTION

```
Sub SetOfWords ()
    Dim words(1 To 5) As String
    Dim i As Integer
    For i = 1 To 5
        words(i) = InputBox("Write a word")
    Next i
    For i = 1 To 5
        MsgBox ("word-" & i & " -> " & words(i))
    Next i
End Sub
```

LBOUND FUNCTION

Returns a number representing the smallest available index of an array. The basic syntax is **LBound**(*arrayname*)

UBOUND FUNCTION

Returns a number representing the largest available index of an array. The basic syntax is **Ubound**(*arrayname*)

ARRAY AS ARGUMENT

When an array is declared to be a procedure argument, the array should be declared without the range of indexes specification, just with rounds. Since the smallest and the largest indexes are not available the **LBound** and **UBound** functions should be used

```
Sub SetOfWords(words() As String)  
    Dim i As Integer  
    For i = LBound(words) To UBound(words)  
        ...  
    Next i  
End Sub
```

```
Sub SetOfNumbers(num() As Integer)  
    Dim i As Integer  
    For i = LBound(num) To UBound(num)  
        ...  
    Next i  
End Sub
```

ARRAY AS ARGUMENT

The procedures are called as we know

```
Sub Main()  
  Dim ArrayOfWords(1 To 5) As String  
  Dim ArrayOfNumbers(1 To 20) As Integer  
  ...  
  Call SetOfWords(ArrayOfWords)  
  Call SetOfNumbers(ArrayOfNumbers)  
End Sub
```

ARRAY AS ARGUMENT

EXAMPLE ONE

```
Sub LongWords(words() As String)
    Dim i As Integer
    Dim c As Integer
    c = 0
    For i = LBound(words) To UBound(words)
        If Len(words(i)) > 5 Then c = c + 1
    Next i
    MsgBox ("Number of words larger than 5 -> " & c)
End Sub

Sub PrintWords(words() As String)
    Dim i As Integer
    Dim s As String
    s = ""
    For i = LBound(words) To UBound(words)
        s = s & i & " - " & words(i) & Chr(13)
    Next i
    MsgBox (s)
End Sub

Sub Main()
    Dim SetOfWords(1 To 5) As String
    Dim i As Integer
    Dim r As Range
    Set r = Worksheets(1).Range("A2:A6")
    For i = 1 To 5
        SetOfWords(i) = r.Cells(i, 1).Value
    Next i
    Call LongWords(SetOfWords)
    Call PrintWords(SetOfWords)
End Sub
```

ARRAY AS ARGUMENT

EXAMPLE TWO

```
Sub SetValue(arr() As Double, w As Worksheet, r As Integer, c As Integer)
    Dim i As Integer
    For i = LBound(arr) To UBound(arr)
        w.Cells(r + i, c).Value = arr(i)
        w.Cells(r + i, c).Interior.ColorIndex = 27
    Next i
End Sub
```

```
Sub main()
    Dim dataSet(1 To 3) As Double
    Dim j As Integer, wr As Integer, wc As Integer, wi As Integer
    Dim ws As Worksheet
    For j = 1 To 3
        dataSet(j) = Val(TextBox("Write a number"))
    Next j
    wi = Val(TextBox("Where? (specify a worksheet index)"))
    wr = Val(TextBox("Which row?"))
    wc = Val(TextBox("Which column?"))
    Set ws = Worksheets(wi)
    Call SetValue(dataSet, ws, wr, wc)
End Sub
```

EXERCISE 2

Write a procedure **AverageMarkBySex** which given a range of cells, such that a row refers to a student, his/her sex, and his/her final mark, firstly register the data into three arrays named respectively Student, Sex and Mark, then it computes the average marks for male students and for female students. The procedure can analyze 10 student a time.

A **Main** procedure calls **AverageMarkBySex** and it passes to it the range of data. Data are arranged in the Column B (Student Name), Column C (Student Sex) and Column D (Student Mark).

SOLUTION

```
Sub AverageMarkBySex(st As Range)
Dim Student(1 To 10) As String, Sex(1 To 10) As String, Mark(1 To 10) As Single
Dim m As Single, f As Single Dim mc As Single, fc As Single Dim i As Integer
If st.Count <= 10 Then
    For i = 1 To 10
        Student(i) = st.Cells(i, 1).Value
        Sex(i) = st.Cells(i, 2).Value
        Mark(i) = st.Cells(i, 3).Value
    Next i
    m = 0
    f = 0
    mc = 0
    fc = 0
    For i = 1 To 10
        If Sex(i) = "male" Then
            m = m + 1
            mc = mc + Mark(i)
        ElseIf Sex(i) = "female" Then
            f = f + 1
            fc = fc + Mark(i)
        End If
    Next i
    MsgBox ("Female: " & fc / f)
    MsgBox ("Male average: " & mc / m)
End Sub

Sub main()
Dim data As Range
Set data = Worksheets(1).Range("B1:D10")
Call AverageMarkBySex(data)
End Sub
```