
VB Array

Arrays in VB programming language

Array is a collection of values of the same type and name. By using an array you can store many values that can be accessed by their indices.

12	23	34	32	40
----	----	----	----	----

index 0

index 1

index 2

index 3

index4

Commonly, programmers use one and two-dimensional arrays.

Declaring One-dimensional array

```
Dim array_name() As DataType
```

Example:

```
Dim arr() As Integer
```

Allocating One-dimensional array

```
array_name=new Datatype(size){ }
```

Example:

```
arr=new Integer(2){ }
```

Initialize One-dimensional array

```
array_name=new Datatype(size){ val1, val2, val3, ... }
```

or

```
array_name(index0)=val1
```

```
array_name(index1)=val2
```

```
array_name(index2)=val3
```

.....

Example:

Module Module1

Sub Main()

'Declaring and allocating the 1D-array

Dim arr() As Integer

arr = New Integer(2) {1, 2, 3}

'Or

'arr = New Integer(2) {}

'arr(0) = 1

'arr(1) = 2

'arr(2) = 3

'Accessing the array elements

For Each i In arr

 Console.WriteLine(i)

Next

Console.ReadLine()

End Sub

End Module

Accessing one-dimensional array elements with for loop

Example:

Module Module1

```
Sub Main()
```

```
'Declaring and allocating the 1D-array
```

```
Dim arr() As Integer
```

```
arr = New Integer(2) {3, 2, 1}
```

```
'sort the array
```

```
Array.Sort(arr)
```

```
'Accessing the array elements with for loop
```

```
For i = 0 To arr.Length - 1 'Or To arr.GetUpperBound(0)
```

```
    Console.WriteLine(arr(i))
```

```
Next
```

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

Note:Length is used to get the length of the array. Length-1 equals to GetUpperBound.To sort an array, use Array.Sort(array_name) statement.

Exercises

Exercise 1: By using the bubble sort algorithm, write VB.NET code to sort an integer array of 10 elements in ascending.

Solution:

```
Module Module1
```

```
Sub Main()
```

```
Dim arr() As Integer = {12, 23, 1, 21, 12, 12, 32, 45, 3, 5}
bubbleSort(arr, arr.Length)
Dim i As Integer
For i = 0 To arr.Length - 1
Console.WriteLine("{0,4}", arr(i))
Next
```

```
Console.ReadLine()
```

```
End Sub
```

```
Sub bubbleSort(ByVal dataset() As Integer, ByVal n As Integer)
```

```
Dim i, j As Integer
For i = 0 To n Step 1
For j = n - 1 To i + 1 Step -1
If (dataset(j) < dataset(j - 1)) Then
```

```
Dim temp As Integer = dataset(j)
dataset(j) = dataset(j - 1)
dataset(j - 1) = temp
```

```
End If
```

```
Next
```

```
Next
```

```
End Sub
```

```
End Module
```

Exercise 2: Modify the VB.NET code in exercise 1 to sort the array in descending order.

Solution:

```
Module Module1
```

```
Sub Main()
```

```
Dim arr() As Integer = {12, 23, 1, 21, 12, 12, 32, 45, 3, 5}
```

```
bubbleSort(arr, arr.Length)
Dim i As Integer
For i = 0 To arr.Length - 1
Console.WriteLine("{0,4}", arr(i))
Next

Console.ReadLine()

End Sub

Sub bubbleSort(ByVal dataset() As Integer, ByVal n As Integer)
Dim i, j As Integer
For i = 0 To n Step 1
For j = n - 1 To i + 1 Step -1
If (dataset(j) > dataset(j - 1)) Then

Dim temp As Integer = dataset(j)
dataset(j) = dataset(j - 1)
dataset(j - 1) = temp

End If

Next
Next
End Sub

End Module
```

For more VB.NET exercises visit <http://www.worldbestlearningcenter.com>