

Forum: [Hack Forums](#)

Thread: [\[TuT\] Using Arrays in VB 2008.](#)

MySocksAreWet - March 26, 2010, 10:41 pm

Lets start witha defination.

An array has a single name that all members of the array share, but each member has its own unique index number. This is used to refer to each item of data numerically, instead of that shitload of different text names.

Here's a nice & neat array:

Code Option Strict Off

Option Explicit On

```
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
```

```
        arrQuestions(0) = "Do you think that bitch is fat?"
```

```
        arrQuestions(1) = "Do you even have her number?"
```

```
        arrQuestions(2) = "This chick looks 35 are you sure shes 17?"
```

```
        MsgBox(arrQuestions(1))
```

```
    End
```

```
End Sub
```

```
End Class
```

Here is a comparsion of a using an array, and not using one. First the ladder (second).

```
Code Option Strict Off
```

```
    Option Explicit On
```

```
    Public Class Form1
```

```
        Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
            Dim Total
```

```
            Dim JanFood = 90
```

```
            Dim FebFood = 122
```

```
            Dim MarFood = 125
```

```
            Dim AprFood = 78
```

```
            Dim MayFood = 132
```

```
            Dim JunFood = 123
```

```
            Dim JulFood = 143
```

```
            Dim AugFood = 112
```

```
            Dim SeptFood = 167
```

```
            Dim OctFood = 123
```

```
Dim NovFood = 132
Dim DecFood = 125
```

```
Total = JanFood + FebFood + MarFood + AprFood + MayFood + JunFood + JulFood + AugFood +
SeptFood + OctFood + NovFood + DecFood
```

```
MsgBox(Total)
```

```
End
```

```
End Sub
End Class
```

Or you can make your program sweet and less time consuming like this:

```
Code Option Strict Off
Option Explicit On
```

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
    Dim arrMonthFoodBill(11)
    Dim Total
```

```
    arrMonthFoodBill(0) =
    arrMonthFoodBill(1) =
    arrMonthFoodBill(2) =
    arrMonthFoodBill(3) =
    arrMonthFoodBill(4) =
    arrMonthFoodBill(5) =
    arrMonthFoodBill(6) =
    arrMonthFoodBill(7) =
    arrMonthFoodBill(8) =
    arrMonthFoodBill(9) =
    arrMonthFoodBill(10) =
    arrMonthFoodBill(11) =
```

```
    For i = 0 To 11
        Total = Total + arrMonthFoodBill(i)
    Next i
```

```
    MsgBox(Total)
```

```
End Sub
End Class
```

Now also remember this, it's called the 'Zero Effect'. When declaring elements in an array, ALWAYS START WITH ZERO.

For example, if your array consists of 15 elements, you would type 14, since you need to start at 0.

Code 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14

arrBitchesIHave(14)

Thank you for reading.

cybr - March 26, 2010, 10:42 pm

Simple, and easy to understand. Nice job.

JIMP - March 26, 2010, 10:51 pm

Good post.Nice job brother.

MySocksAreWet - March 26, 2010, 11:17 pm

Thread has been updated too.

MySocksAreWet - March 27, 2010, 3:17 pm

Only 13 views come on guys this took a while...

cybr - March 27, 2010, 3:29 pm

For example, if your array consists of 15 elements, you would type 14, since you need to start at 0.

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14

arrBitchesIHave(14)

MySocksAreWet - March 27, 2010, 3:30 pm

(Mar 27, 2010 03:29 PM)iM TheoriuM Wrote: [\[CITE\]](#) For example, if your array consists of 15 elements, you would type 14, since you need to start at 0.

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14

arrBitchesIHave(14)

I'll add that thanks.