

Introduction to VB.net

Lecture 5

Nasrin Khodapanah



VB.NET

Agenda

- Subs & Functions
 - EX05
 - Lab3

Subs & Functions



Subs & Functions

A function is more or less the same thing as a Sub - a segment of code you create yourself, and that can be used whenever you want it. The difference is that a Function returns a value, while a Sub doesn't.

Subs

```
Module Module1
```

```
    Sub main()
```

```
        DisplayMessage()
```

```
    End Sub
```

```
    Sub DisplayMessage()
```

```
        Console.WriteLine("Hello everybody")
```

```
    End Sub
```

```
End Module
```

Functions

```
Module Module1
```

```
    Sub main()
```

```
        Console.WriteLine(DisplayAnotherMessage)
```

```
    End Sub
```

```
Function DisplayAnotherMessage()
```

```
    Dim message As String = "This "
```

```
    message += "is "
```

```
    message += "another "
```

```
    message += "message "
```

```
    Return message
```

```
End Function
```

```
End Module
```

Subs & Functions

Arguments

```
Sub Add(ByVal Val1 As Integer, ByVal Val2 As Integer)
```

Calling the sub
`add(val1, val2)`

Subs & Functions

```
Function Add(ByVal val1 As Integer, ByVal val2 As Integer) As Integer
```

```
    Dim result As Integer
```

```
    'Add values
```

```
    result = val1 + val2
```

```
    'Return result
```

```
    Return result
```

```
End Function
```


Subs & Functions

Optional Arguments

```
Function Add2(ByVal val1 As Integer, ByVal val2 As  
    Integer, Optional ByVal val3 As Integer = 0) As Integer  
    Return val1 + val2 + val3
```

End Function

ex: call it in the main with 2 values, then with 3 values

comments: ''' (three quotes)

Subs & Functions

''' <summary>

''' Add passed values

''' </summary>

''' <param name="Val1">First value to add</param>

''' <param name="Val2">Second value to add</param>

''' <param name="Val3">Third value to add, Optional</param>

''' <returns>The addition of the three values</returns>

''' <remarks></remarks>

ByVal, ByRef.

- ByVal and ByRef change how parameters are received.
- A parameter passed ByVal—by value—can be changed in the new method. Its value will not be changed elsewhere.
- ByRef, by reference, means the variable location itself is copied.
- The default in Visual Basic is to pass arguments by value.

ByVal, ByRef.

```
Sub Main()  
    Dim value As Integer = 1  
  
    ' The integer value doesn't change here when passed ByVal.  
    test1(value)  
    Console.WriteLine(value)  
  
    ' The integer value DOES change when passed ByRef.  
    test2(value)  
    Console.WriteLine(value)
```

```
Sub test1(ByVal test As Integer)  
    test = 16  
End Sub
```

```
Sub test2(ByRef test As Integer)  
    test = 16  
End Sub
```

Detail: When the integer value is passed to test1, its value is only changed inside the test1 subroutine. In Main the value is unchanged.

Note: ByVal passes a copy of the bytes of the variable (the value of it). It does not copy a storage location.

Detail: In test2, the reference to the integer is copied, so when the value is changed, it is reflected in the Main sub.

Finally, The value is changed to 16 in the Main subroutine after test2 returns.

EX05



Exo - Subs & Functions

Transform the last exo using subs and functions for each of the options

Sub / Functions for Add

Sub / Functions for Delete

Sub / Functions for Display

Sub / Functions for Count

Sub / Functions for CountLongest