VB.NET - CONSTANTS AND ENUMERATIONS

The **constants** refer to fixed values that the program may not alter during its execution. These fixed values are also called literals.

Constants can be of any of the basic data types like an integer constant, a floating constant, a character constant, or a string literal. There are also enumeration constants as well.

The constants are treated just like regular variables except that their values cannot be modified after their definition.

An **enumeration** is a set of named integer constants.

Declaring Constants

In VB.Net, constants are declared using the **Const** statement. The Const statement is used at module, class, structure, procedure, or block level for use in place of literal values.

The syntax for the Const statement is:

```
[ < attributelist> ] [ accessmodifier ] [ Shadows ]
Const constantlist
```

Where,

- ***attributelist***: specifies the list of attributes applied to the constants; you can provide multiple attributes separated by commas. Optional.
- ***accessmodifier***: specifies which code can access these constants. Optional. Values can be either of the: Public, Protected, Friend, Protected Friend, or Private.
- ***Shadows***: this makes the constant hide a programming element of identical name in a base class. Optional.
- ***Constantlist***: gives the list of names of constants declared. Required.

Where, each constant name has the following syntax and parts:

```
constantname [ As datatype ] = initializer
```

- ***constantname***: specifies the name of the constant
- ***datatype***: specifies the data type of the constant
- ***initializer***: specifies the value assigned to the constant

For example,

```
' The following statements declare constants.
Const maxval As Long = 4999
Public Const message As String = "HELLO"
Private Const piValue As Double = 3.1415
```

Example

The following example demonstrates declaration and use of a constant value:

```
Module constantsNenum
   Sub Main()
      Const PI = 3.14149
      Dim radius, area As Single
      radius = 7
      area = PI * radius * radius
      Console.WriteLine("Area = " & Str(area))
      Console.ReadKey()
   End Sub
End Module
```

When the above code is compiled and executed, it produces the following result:

```
Area = 153.933
```

Print and Display Constants in VB.Net

VB.Net provides the following print and display constants:

Constant	Description
vbCrLf	Carriage return/linefeed character combination.
vbCr	Carriage return character.
vbLf	Linefeed character.
vbNewLine	Newline character.
vbNullChar	Null character.
vbNullString	Not the same as a zero-length string "" ; used for calling external procedures.
vbObjectError	Error number. User-defined error numbers should be greater than this value. For example: Err.Raise*Number* = vbObjectError + 1000
vbTab	Tab character.
vbBack	Backspace character.

Declaring Enumerations

An enumerated type is declared using the **Enum** statement. The Enum statement declares an enumeration and defines the values of its members. The Enum statement can be used at the module, class, structure, procedure, or block level.

The syntax for the Enum statement is as follows:

```
[ < attributelist > ] [ accessmodifier ]  [ Shadows ]
Enum enumerationname [ As datatype ]
   memberlist
End Enum
```

Where,

- ***attributelist***: refers to the list of attributes applied to the variable. Optional.
- ***asscessmodifier***: specifies which code can access these enumerations. Optional. Values can be either of the: Public, Protected, Friend or Private.
- ***Shadows***: this makes the enumeration hide a programming element of identical name in a base class. Optional.
- ***enumerationname***: name of the enumeration. Required
- ***datatype***: specifies the data type of the enumeration and all its members.

- ***memberlist***: specifies the list of member constants being declared in this statement. Required.

Each member in the memberlist has the following syntax and parts:

```
[< attribute list>] member name [ = initializer ]
```

Where,

- ***name***: specifies the name of the member. Required.

- ***initializer***: value assigned to the enumeration member. Optional.

For example,

```
Enum Colors
   red = 1
   orange = 2
   yellow = 3
   green = 4
   azure = 5
   blue = 6
   violet = 7
End Enum
```

Example

The following example demonstrates declaration and use of the Enum variable *Colors*:

```
Module constantsNenum
   Enum Colors
      red = 1
      orange = 2
      yellow = 3
      green = 4
      azure = 5
      blue = 6
      violet = 7
   End Enum
   Sub Main()
      Console.WriteLine("The Color Red is : " & Colors.red)
      Console.WriteLine("The Color Yellow is : " & Colors.yellow)
      Console.WriteLine("The Color Blue is : " & Colors.blue)
      Console.WriteLine("The Color Green is : " & Colors.green)
      Console.ReadKey()
   End Sub
End Module
```

When the above code is compiled and executed, it produces the following result:

```
The Color Red is: 1
The Color Yellow is: 3
The Color Blue is: 6
The Color Green is: 4
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js