Char Class

Properties

- MinValue
  - This is a constant and shared property of the Char Class
  - It represents the minimum value a Char variable can have
  - The minimum value = Nothing
  - Thus, the declaration of a Char variable without initialization is set to Nothing

Properties

- MaxValue
  - This is a constant and shared property of the Char Class
  - It represents the maximum value a Char variable can have
  - In VB.NET, since a Char datatype is of 2 bytes = 16 bits
  - The maximum value = ChrW(65535)
  - ChrW() represents a UNICODE character and 2^{16}=65536, since the first character begins from 0. Then, the range is 0- 65535
  - Thus, a character variable can be set to any of the above range

Methods

- These are shared methods and return a Boolean Datatype.
- True, if they satisfy the condition else False

1. IsDigit()
2. IsControl()
3. IsLetter()
4. IsLetterOrDigit()
5. IsLower()
6. IsUpper()
7. IsNumber()
8. IsWhiteSpace()

Method Call

- To use the above methods:
- Syntax:

  <variable> = Char.IsMethodName(<argument>)

  - It returns a Boolean Datatype
  - The argument must be of Char Datatype
  - Variable should be Boolean

Conversion Methods

1. ToLower(): Converts a character argument to lower-case and returns a Char Datatype
2. ToUpper():Converts a character argument to upper-case and returns a Char Datatype
3. ToString(): Converts a character argument to String Datatype and thus returns a String datatype

- Syntax:

  <variable>=Char.ToMethodname(<argument>)

  – It returns a Char Datatype
  – The argument must be of Char Datatype

ToString() Method

- Incase of this method, we have a variation available:
  - We can make use of toString() method, present in Object Class which converts any datatype to String
  - Syntax:

    <variable>.toString()

Example

```
Module Module1
    Sub Main()
        Dim c As Char = "a“
        Dim uc, lc As Char
        Console.WriteLine(“Using Char Class =  ” & Char.ToString(c))
        Console.WriteLine(“Using Object Class ” & c.toString())
        uc = Char.ToUpper(c)
        Console.WriteLine(“Upper-case:” & uc)
        lc = Char.ToLower(c)
        Console.WriteLine(“Lower-case:” & lc)
        Console.ReadKey()
    End Sub
End Module
```

Chr() & ChrW()

- Chr() & ChrW() method converts the integer argument to a Char datatype
- Chr() will convert ASCII Code (0-255)
- ChrW() will convert to a wide-character using UNICODE(0-65535)
- Syntax:

  <variable_name>=Chr(<argument>)

  or, <variable_name>=ChrW(<argument>)

  - Here, <variable_name> is of Char Datatype
  - <argument> is of Integer datatype
    - For Chr() – values can be from 0-255
    - For ChrW() – values can be from 0-65535

Asc()

- Asc() & AscW() method converts the character argument to its equivalent ASCII Code which is an integer datatype
- Asc() will result ASCII Code (0-255)
- AscW() will result to a wide-character code i.e. UNICODE(0-65535)
- Syntax:

  <variable_name>=Asc(<argument>)

  or, <variable_name>=AscW(<argument>)

  - Here, <variable_name> is of Integer Datatype
  - <argument> is of Char datatype

```
Module Module1
    Sub Main()
        Dim c As Char = "A"
        Dim d As Integer = 65
        Console.WriteLine(ChrW(d))
        Console.WriteLine(AscW(c))
        Console.WriteLine(Chr(d))
        Console.WriteLine(Asc(c))
        Console.ReadKey()
    End Sub
End Module
```

OUTPUT:
A
65
A
65