

# Data Types in VB.NET

Kavita K. Bharti  
Assistant Professor  
Department of Computer  
Durga Mahavidyalaya, Raipur

In VB.NET, **data type** is used to define the type of a variable or function in a program. Furthermore, the conversion of one data type to another type using the data conversion function.

A **Data Type** refers to which type of data or value is assigning to a variable or function so that a variable can hold a defined data type value. For example, when we declare a variable, we have to tell the compiler what type of data or value is allocated to different kinds of variables to hold different amounts of space in computer memory.

## Syntax:

```
Dim Variable_Name as DataType
```

**VariableName:** It defines the name of the variable that you assign to store values.

**DataType:** It represents the name of the data type that you assign to a variable.

## Different Data Types and their allocating spaces in VB.NET

The following table shows the various data types list in the VB.NET

Data Types	Size	Range
<b>Boolean</b>	A Boolean type depends on the implementing platform	True or False
<b>Byte</b>	1 byte	0 to 255 (unsigned)
<b>Char</b>	2 bytes	0 to 65535 (unsigned)
<b>Date</b>	8 bytes	0:00:0 (midnight) January 1, 0001 to 11:59:59 PM of December 31, 9999.
<b>Decimal</b>	16 bytes	0 to +/- 79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) without any decimal point; And 0 to +/- 7.92281625142264337593543950335 with 28 position to the right of the decimal
<b>Double</b>	8 bytes	-1.79769313486231570E+308 to -4.94- 65645841246544E-324 for negative values; 4.94065645841246544E-324 to 1.79769313486231570E+308, for positive

		values
<b>Integer</b>	4 bytes	-2,147,483,648 to 2,147,483,647 (signed)
<b>Long</b>	8 bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (9.2...E + 18) (signed)
<b>Object</b>	Object size based on the platform such as 4 bytes in 32-bit and 8 bytes in 64-bit platform	It can store any type of data defined in a variable of type Object
<b>SByte</b>	1 byte	-128 to 127 (signed)
<b>Short</b>	2 bytes	-32,768 to 32,767 (signed)
<b>Single</b>	4 bytes	-3.4028235E + 38 to -1.401298E-45 for negative values; And for positive value: 1.401298E-45 to 3.4028235E + 38.
<b>String</b>	String Datatype depend on the implementing platform	It accepts Unicode character from 0 to approximately 2 billion characters.
<b>UInteger</b>	4 bytes	The range start from 0 to 4,294,967,295 (unsigned)
<b>ULong</b>	8 bytes	The range of ULong start from 0 to 18,446,744,073,709,551,615 (1.8...E + 19) (unsigned)
<b>User-Defined (structure)</b>	A user-defined data type depends on the implementing platform	Each member of the structure has its own data type and limits independent of the other members' ranges.
<b>UShort</b>	2 bytes	Range from 0 to 65,535 (unsigned)

Let's use the various data types in a VB.NET program.

### Data\_type.vb

Module Data\_type

Sub Main()

' defining the Data Type to the variables

Dim b As Byte = 1

Dim num As Integer = 5

Dim si As Single

Dim db As Double

Dim get\_date As Date

Dim c As Char

Dim str As String

b = 1

```
num = 20
si = 0.12
db = 2131.787
get_date = Today
c = "A"
str = "Hello Students...."
```

```
Console.WriteLine("Byte is: {0}", b)
Console.WriteLine("Integer number is: {0}", num)
Console.WriteLine("Single data type is: {0}", si)
Console.WriteLine("Double data type is: {0}", db)
Console.WriteLine("Today is: {0}", get_date)
Console.WriteLine("Character is: {0}", b)
Console.WriteLine("String message is: {0}", str)
Console.ReadKey()
```

End Sub

End Module

# Type Conversion Functions in VB.NET

The following functions are available for conversion.

- ❖ **CBool(expression)**: It is used to convert an expression into a Boolean data type.
- ❖ **CByte(expression)**: It is used to convert an expression to a Byte data type.
- ❖ **CChar(expression)**: It is used to convert an expression to a Char data type.
- ❖ **CDate(expression)**: It is used to convert an expression to a Date data type.
- ❖ **Cdbl(expression)**: It is used to convert an expression into a Double data type.
- ❖ **CDec(expression)**: It is used to convert an expression into a Decimal data type.
- ❖ **CInt(expression)**: It is used to convert an expression to an Integer data type.
- ❖ **CLng(expression)**: It is used to convert an expression to a Long data type.
- ❖ **CObj(expression)**: It is used to convert an expression to an Object data type.
- ❖ **CSByte(expression)**: It is used to convert an expression to an SByte data type.
- ❖ **CShort(expression)**: It is used to convert an expression to a Short data type.
- ❖ **CSng(expression)**: It is used to convert an expression into a Single data type.
- ❖ **CStr(expression)**: It is used to convert an expression into a String data type.
- ❖ **CUInt(expression)**: It is used to convert an expression to a UInt data type.
- ❖ **CULng(expression)**: It is used to convert an expression to a ULng data type.
- ❖ **CUShort(expression)**: It is used to convert an expression into a UShort data type.

In the following, program we have performed different conversion.

## **DB\_Conversion.vb**

```
Option Strict On
Module DB_Conversion
    Sub Main()
        'defining the Data type conversion
        Dim dblData As Double
        dblData = 5.78
        Dim A, B As Char
        Dim bool As Boolean = True
        Dim x, Z, B_int As Integer
```

```
A = "A"
B = "B"
B_int = AscW(B)
    Console.WriteLine(" Ascii value of B is {0}", B_int)
x = 1
Z = AscW(A)
Z = Z + x
Console.WriteLine("String to integer {0}", Z)
Console.WriteLine("Boolean value is : {0}", CStr(bool))
Dim num, intData As Integer
    num = CInt(dblData)
intData = CType(dblData, Integer)
Console.WriteLine(" Explicit conversion of Data type " & Str(intData))
Console.WriteLine(" Value of Double is: {0}", dblData)
Console.WriteLine("Double to Integer: {0}", num)
Console.ReadKey()
End Sub
End Module
```