

Looping Statement inVB.NET

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

Do While Loop

There may be a situation when you need to execute a block of code several times. In general, statements are executed sequentially. **A loop statement allows us to execute a statement or group of statements multiple times.**

It repeats the enclosed block of statements while a Boolean condition is true or until the condition becomes True. It could be terminated at any time with the Exit Do statement.

The syntax for this loop construct is –

Do { While | Until } condition

[statements]

[Continue Do]

[statements]

[Exit Do]

[statements]

Loop

-or-

Do

[statements]

[Continue Do]

[statements]

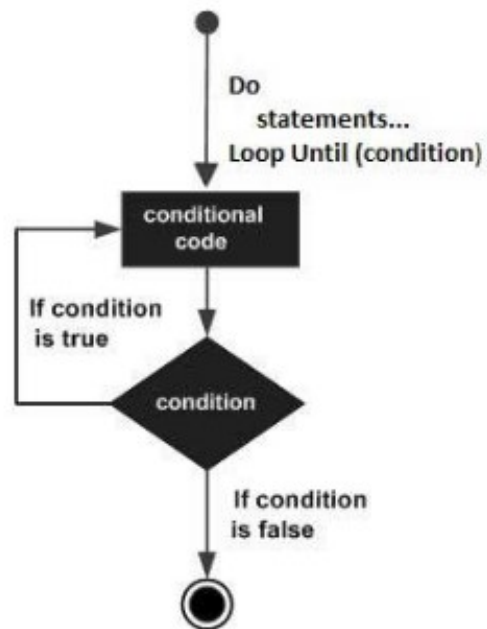
[Exit Do]

[statements]

Loop { While | Until } condition

*Anything written between square brackets [] are optional.

Flow Diagram



Module loops

```
Sub Main()
```

```
    Dim a As Integer = 100
```

```
    do loop execution
```

```
    Do
```

```
        Console.WriteLine("value of a: {0}", a)
```

```
        a = a + 1
```

```
    Loop While (a < 120)
```

```
    Console.ReadLine()
```

```
End Sub
```

```
End Module
```

OUTPUT

Value of a = 100

Value of a = 101

.

.

Value of a = 120

The program would behave in same way, if you use an Until statement, instead of While –

```
Module loops
```

```
Sub Main()
```

```
    Dim a As Integer = 100
```

```
    do loop execution
```

```
    Do
```

```
        Console.WriteLine("value of a: {0}", a)
```

```
        a = a + 1
```

```
    Loop Until (a = 120)
```

```
    Console.ReadLine()
```

```
End Sub
```

```
End Module
```

For....Next loop

It repeats a group of statements a specified number of times and a loop index counts the number of loop iterations as the loop executes.

The syntax for this loop construct is –

```
For counter [ As datatype ] = start To end [ Step step ]
```

```
    [ statements ]
```

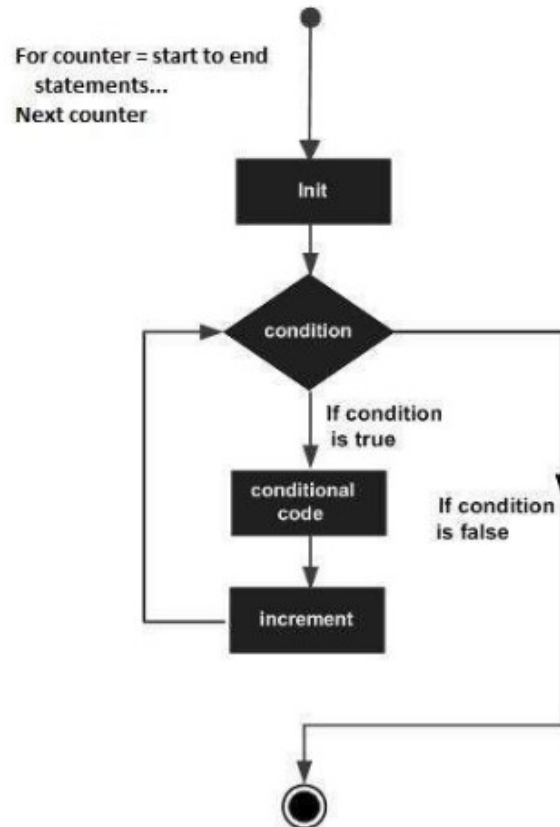
```
    [ Continue For ]
```

```
    [ statements ]
```

```
    [ Exit For ]
```

```
[ statements ]  
Next [ counter ]
```

Flow Diagram



Exmple

```
Module loops  
Sub Main()  
  Dim a As Byte  
  For a = 10 To 20  
    Console.WriteLine("value of a: {0}", a)  
  Next  
  Console.ReadLine()  
End Sub  
End Module
```

Output

value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15
value of a: 16
value of a: 17
value of a: 18
value of a: 19
value of a: 20

If you want to use a step size of 2, for example, you need to display only even numbers, between 10 and 20 –

Module loops

```
Sub Main()  
    Dim a As Byte  
    For a = 10 To 20 Step 2  
        Console.WriteLine("value of a: {0}", a)  
    Next  
    Console.ReadLine()  
End Sub  
End Module
```

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

value of a: 10
value of a: 12
value of a: 14
value of a: 16
value of a: 18
value of a: 20