

Looping Control Structure

Sometimes it is required to perform some action or task repeatedly for specified number of times or until some condition is satisfied at that time you need to use Looping control structure. Following are the List of Looping Control Structure:

- (1) Do While ... Loop
- (2) Do Until Loop
- (3) Do ... Loop While
- (4) Do ... Loop Until
- (5) For ... Next
- (6) For Each ... Next
- (7) While ... End While

(1) Do While Loop

Do While ... Loop control structure is used to repeat statements between Do While and Loop statements until the condition specified with the Do While statements evaluates to FALSE. The general syntax of Do While ... Loop structure is given below:

```
Do while Condition
Statement Block
Loop
```

The above structure works as follow:

- (1) First it tests for condition specified with Do While statement.
- (2) If the condition evaluates to TRUE then the statement block within Do While and Loop is executed. After executing the statement block the condition is tested again and the same process is repeated until the condition becomes FALSE.
- (3) If the condition evaluates to FALSE then the statements between Do While and Loop is not executed and control is transferred after the statement Loop.

If the condition evaluates to FALSE at the first trial the statements between Do While and Loop statements will never executes.

It is also known as Entry Controlled Loop or Pretest Loop because the condition is checked before executing the statements.

Example:

Suppose you want to display odd numbers in the range 1 to 10.

```
Dim a As Integer
a = 1
Do While a <= 10
If a Mod 2 <> 0 Then
lblOdd.Text = lblOdd.Text & " " & a
End If
a = a + 1
Loop
```

(2) Do Until Loop

Do Until ... Loop control structure is used to repeat statements between Do Until and Loop statements until the condition specified with the Do Until statements evaluates to TRUE. The general syntax of Do Until ... Loop structure is given below:

```
Do Until Condition  
Statement Block  
Loop
```

The above structure works as follow:

- (1) First it tests for condition specified with Do Until statement.
- (2) If the condition evaluates to FALSE then the statement block within Do Until and loop is executed. After executing the statement block the condition is tested again and the same process is repeated until the condition becomes TRUE.
- (3) If the condition evaluates to TRUE then the statements between Do Until and Loop is not executed and control is transferred after the statement Loop.

Example:

Suppose you want to display odd numbers in the range 1 to 10.

```
Dim a As Integer  
a = 1  
Do Until a >= 10  
If a Mod 2 <> 0 Then  
lblOdd.Text = lblOdd.Text & " " & a  
End If  
a = a + 1  
Loop
```

(3) Do ... Loop While

Do ... Loop While control structure is used to repeat Statement Block until the condition evaluates to FALSE. The general syntax of Do ... Loop While structure is given below:

```
Do  
Statement Block  
Loop While condition
```

The above structure works as follow:

- (1) In this structure first Statement Block is executed.
- (2) After executing the Statement Block it tests for the condition.
- (3) If the condition evaluates to TRUE then the Statement Block is executed. After executing the Statement Block the condition is tested again and the same process is repeated until the condition becomes FALSE.

(4) If the condition evaluates to FALSE then the Statement Block is not executed and control is transferred after the Loop. Thus in Do Loop While structure the Statement Block executes at least once even the condition is FALSE at the First trial.

It is also known as Exit Controlled Loop or Posttest Loop because the condition is checked after executing the statement block.

Example:

Suppose you want to display odd numbers in the range 1 to 10

```
Dim a As Integer
a = 1
Do
If a Mod 2 <> 0 Then
lblOdd.Text = lblOdd.Text & " " & a
End If
a = a + 1
Loop While a <= 10
```

(4) Do ... Loop Until

Do ... Loop Until control structure is used to repeat Statement Block until the condition specified evaluates to TRUE. The general syntax of Do ... Loop Until structure is given below:

```
Do
Statement Block
Loop Until condition
```

The above structure works as follow:

(1) In this structure first Statement Block is executed.

(2) After executing the Statement Block it tests for the condition.

(3) If the condition evaluates to FALSE then the Statement Block is executed. After executing the Statement Block the condition is tested again and the same process is repeated until the condition becomes TRUE.

(4) If the condition evaluates to TRUE then the Statement Block is not executed and control is transferred after the statement Loop.

Thus in Do Loop until structure the statement block executes at least once even the condition is TRUE at the First trial.

Example:

Suppose you want to display odd numbers in the range 1 to 10:

```
lblOdd.Text = ""
Dim a As Integer
a = 1
Do
If a Mod 2 <> 0 Then
lblOdd.Text = lblOdd.Text & " " & a
End If
a = a + 1
Loop Until a >= 10
```

(5) For...Next

For...Next control structure is used to repeat Statement Block for specified number of times. The general syntax of For ... Next statement is given below:

```
For counter=Start-Value To End-Value [step Step-Value]
Statement Block
Next
```

Here,

Start-Value: It indicates from which value the loop will start.

End-Value: It indicates at which value the loop will stop.

Step-Value: it is optional. It indicates by which value the counter will increment or decrement. If not specified then by default its value is 1. You can specify either positive or negative value.

Note: When step value is positive then the start value must be less than the end value. When step value is negative then the start value must be greater than end value.

The For ... Next statement is useful when you know in advance how many times you want to execute the statement block repeatedly.

The above structure works as follows:

(1) First it initializes the counter with the start value.

(2) Executes the statement block.

(3) After executing the Statement Block it will increment or decrement the counter with the value specified in the step value. If step value is not specified then by default it increments or decrements the counter with 1.

This process is repeated until the counter reaches the End Value.

Example:

Suppose you want to find the factorial of number 4:

```
Dim i As Integer
Dim n As Integer = 4
Dim fact As Integer = 1
For i = 1 To n Step 1
    fact = fact * i
Next
Label1.Text = fact
```

(6) For Each...Next:

For each ... Next control structure repeats the Statement Block for all the elements in the array or collection. So there is no need to know the start value and the end value.

The general syntax of For Each ... Next control structure is given below:

```
For Each elements In Array/Collection
Statement Block
Next
```

Example: Suppose you want to multiply each element of array by 2 and then display it.

```
Label1.Text = ""  
Dim i As Integer  
Dim n() As Integer = {1, 2, 3, 4, 5}  
For Each i In n  
    i = i * 2  
    Label1.Text = Label1.Text & " " & i  
Next
```

(7) While ... End While

The general syntax of while End While structure is given below:

```
While condition  
Statement Block  
End While
```

While ... End While control structure is used to repeat statement block until the specified condition evaluates to false.

The above structure works as follow:

- (1) First it tests for condition.
- (2) If the condition evaluates to TRUE then the statement block is executed. After executing the statement block the condition is tested again and the same process is repeated until the condition becomes FALSE.
- (3) If the condition evaluates to FALSE then the statement block is not executed and control is transferred after End While statement.

Example:

Suppose you want to display odd numbers in the range 1 to 10.

```
lblOdd.Text = ""  
Dim a As Integer  
a = 1  
While a < 10  
    If a Mod 2 <> 0 Then  
        lblOdd.Text = lblOdd.Text & " " & a  
    End If  
    a = a + 1  
End While
```