**C# for loop:**

The for keyword indicates a loop in C#. The for loop executes a block of statements repeatedly until the specified condition returns false.

Syntax:

for (variable initialization; condition; steps)

{

//execute this code block as long as condition is satisfied

}

As per the syntax above, the for loop contains three parts: initialization, conditional expression and steps, which are separated by a semicolon.

* variable initialization: Declare & initialize a variable here which will be used in conditional expression and steps part.
* condition: The condition is a boolean expression which will return either true or false.
* steps: The steps defines the incremental or decremental part

Consider the following example of a simple for loop.

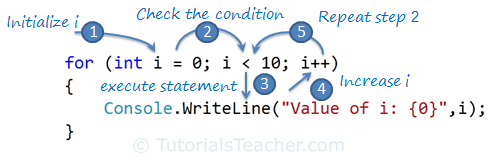
for (int i = 0; i < 10; i++)

{

Console.WriteLine("Value of i: {0}", i);

}

The below figure illustrates the execution steps of above example.

[](http://www.tutorialsteacher.com/Content/images/csharp/for-loop.png)

for loop execution steps

As you can see in the above example, first step is to declare & initialize an int type variable. The second step is to check the condition. The third step is to execute the code block if the 'if' condition returns true. The fourth step is to increment the int variable and last step is to evaluate the condition again and repeat the steps.

It is not necessary to put the initialization, condition and steps into brackets. You can initialize a variable before the 'for' loop, and the condition and steps can be defined inside the for loop.

int i = 0;

for(;;)

{

if (i < 10)

{

Console.WriteLine("Value of i: {0}", i);

i++;

}

else

break;

}

**Infinite loop:**

Be careful with infinite loop. It will be an infinite loop if for loop does not contain initialization, condition or steps part. Also, make sure that conditional expression will return false at some point of time to stop the looping.

for ( ; ; )

{

Console.Write(1);

}

The control variable for the for loop can be of any numeric data type, such as double, decimal, etc.

Example: for loop

for (double d = 1.01D; d < 1.10; d+= 0.01D)

{

Console.WriteLine("Value of i: {0}", d);

}

The steps part in a for loop can either increase or decrease the value of a variable.

for(int i = 10; i> 0;i--)

{

Console.WriteLine("Value of i: {0}", i);

}

**Break:**

You can also exit from a for loop by using the break keyword.

Example: break in for loop

for (int i = 0; i < 10; i++)

{

if( i == 5 )

break;

Console.WriteLine("Value of i: {0}", i);

}

**Nested For loop:**

C# allows a for loop inside another for loop.

for (int i = 0; i < 10; i++)

{

for(int j =i; j< 10; j++)

Console.WriteLine("Value of i: {0}, J: {1} ", i,j);

}

**Points to Remember:**

1. The for loop executes the block of code repeatedly.
2. The for loop has three steps: initialization, condition and increment/decrement.
3. The for loop can use control variable of any numeric data type.
4. Use break keyword to stop the execution and exit from for loop.
5. Nested for loop is allowed in C#.