**Ternary operator "?:" :**

C# includes a special type of decision making operator '?:' called the ternary operator.

Syntax:

var result = Boolean conditional expression ? first statement : second statement

As you can see in the above syntax, ternary operator includes three parts. First part (before ?) includes conditional expression that returns boolean value true or false. Second part (after ? and before :) contains a statement which will be returned if the conditional expression in the first part evalutes to true. The third part includes another statement which will be returned if the conditional expression returns false.

Note :Ternary operator returns a value or expression included in the second or third part of it. It does not execute the statements.

Consider the following example where conditional expression x > y returns false, so it the returns the first statement after ?.

int x = 20, y = 10;

var result = x > y ? "x is greater than y" : "x is less than or equal to y";

Console.WriteLine(result);

The ternary operator can return a value of any data type. So it is advisable to store it in implicitly typed variable - var.

For example, it can return an integer value as shown below.

int x = 20, y = 10;

var result = x > y ? x : y;

Console.WriteLine(result);

Ternary operator can also be used instead of if-else statement. The above example can be written using if-else statement as shown below.

int x = 20, y = 10;

int result = 0;

if (x > y)

result = x;

else if (x < y)

result = y;

Console.WriteLine(result);

**Nested Ternary operator:**

Nested ternary operators are possible by including conditional expression as a second (after ?) or third part (after :) of the ternary operator..Consider the following example.

Example: Nested ternary operator

int x = 2, y = 10;

string result = x > y ? "x is greater than y" : x < y ?

"x is less than y" : x == y ?

"x is equal to y" : "No result";

The ternary operator is right-associative. The expression a ? b : c ? d : e is evaluated as a ? b : (c ? d : e), not as (a ? b : c) ? d : e.

**Points to Remember :**

1. Ternary operator: boolean expression ? first statement : second statement;
2. Ternary operator returns a value, it does not execute it.
3. It can be used to replace a short if-else statement.
4. A nested ternary operator is allowed. It will be evaluated from right to left.