Code Academy
أكاديمية البرمجة

Boolean and Operators

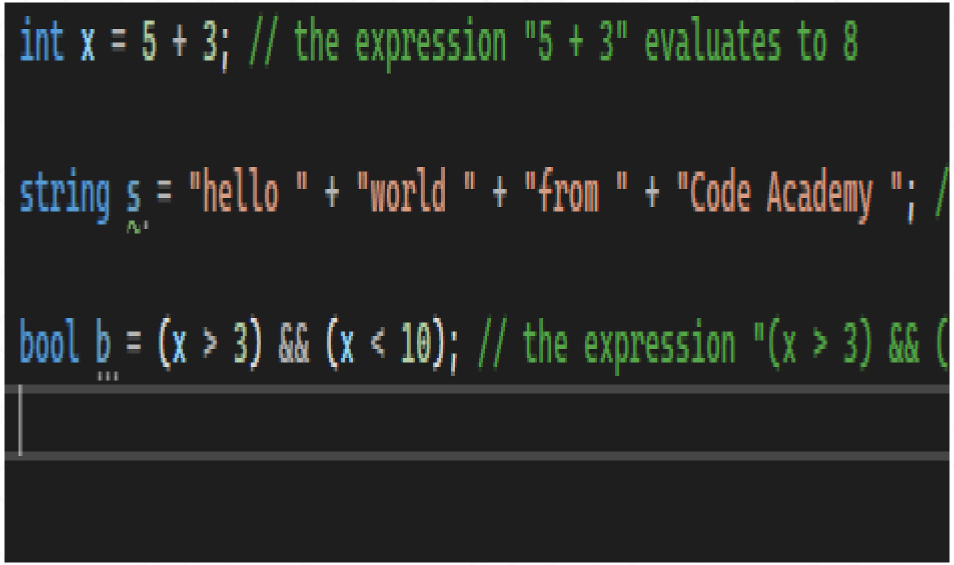
Eng : yousif mohamed

Code Academy
أكاديمية البرمجة

Table of Content

What is an Expression ?

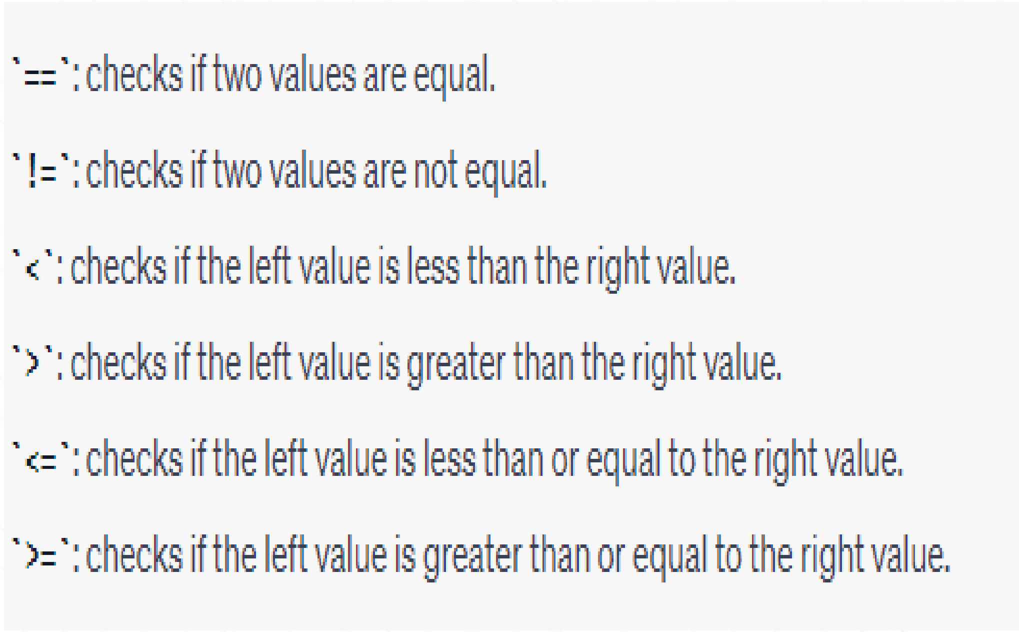
In computer programming, an expression is a combination of one or more values, variables, operators, and function calls that can be evaluated to produce a result.



```
int x = 5 + 3; // the expression "5 + 3" evaluates to 8

string s = "hello " + "world " + "from " + "Code Academy "; /

bool b = (x > 3) && (x < 10); // the expression "(x > 3) && (
```

Comparison Operators

comparison operators are used to compare two values and determine their relationship to each other. These operators return a Boolean value of `true` or `false` depending on whether the comparison is true or false.



`==`: checks if two values are equal.

`!=`: checks if two values are not equal.

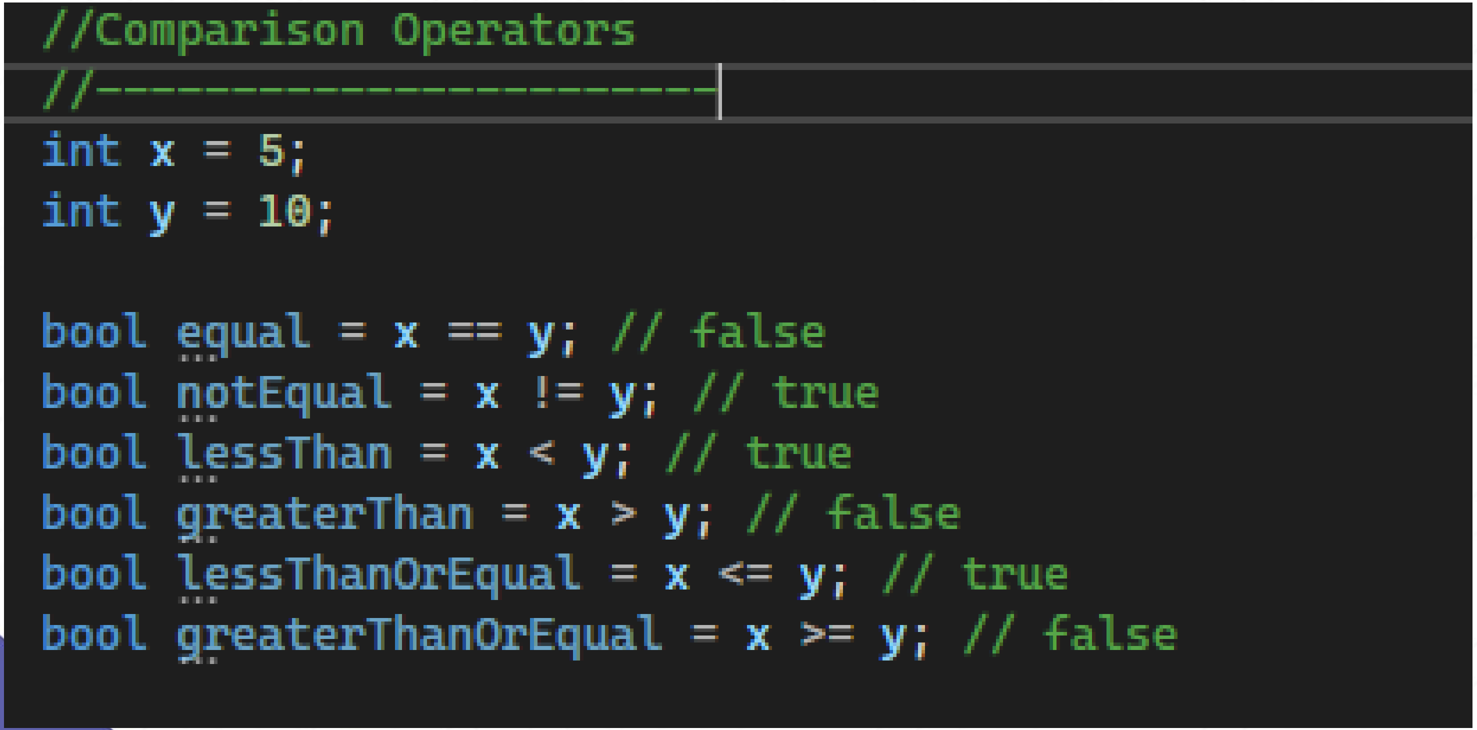
`<`: checks if the left value is less than the right value.

`>`: checks if the left value is greater than the right value.

`<=`: checks if the left value is less than or equal to the right value.

`>=`: checks if the left value is greater than or equal to the right value.

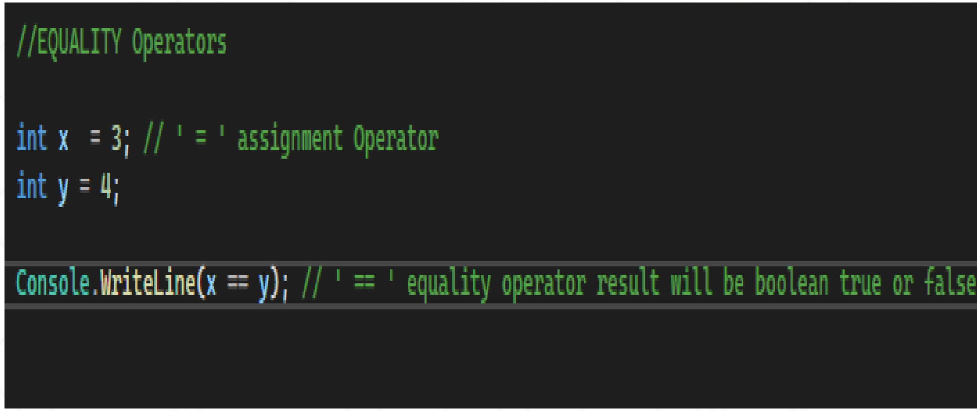
Comparison Operators Examples



```
//Comparison Operators
//------------------------
int x = 5;
int y = 10;

bool equal = x == y; // false
bool notEqual = x != y; // true
bool lessThan = x < y; // true
bool greaterThan = x > y; // false
bool lessThanOrEqual = x <= y; // true
bool greaterThanOrEqual = x >= y; // false
```

Equality Operators ‘ == ’:

In your Program you will need to detect if something is valid under certain Condition or not by saying that if Uservalue is equal to something you will do certain action and we will dive deep in this in upcoming lectures but for now let’s Learn Basics with below example .



```
//EQUALITY Operators

int x  = 3; // ' = ' assignment Operator
int y = 4;

Console.WriteLine(x == y); // ' == ' equality operator result will be boolean true or false
```

Equality Operator is mostly used to get result in boolean form (True or False)

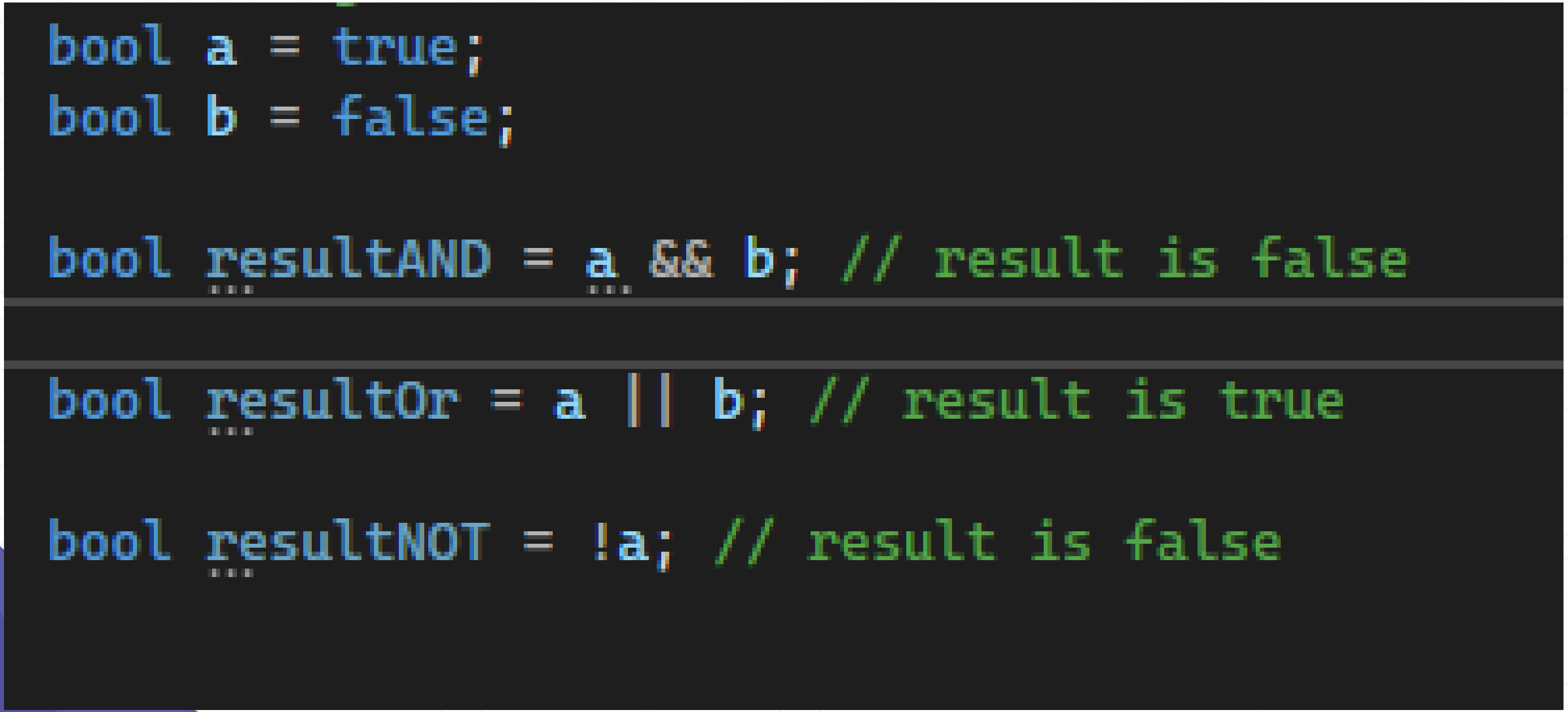
Code Academy
أكاديمية البرمجة

Logical Operators

logical operators are used to combine multiple Boolean expressions and determine their overall truth value. There are three logical operators in C#:

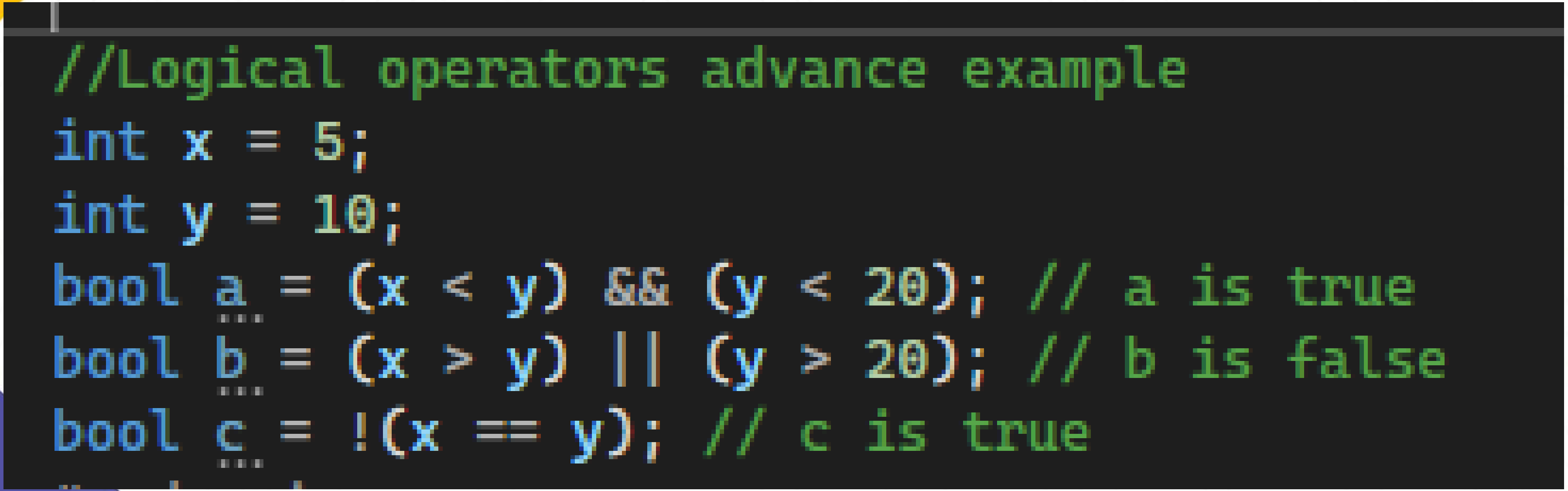
1. `&&` (logical AND): This operator returns `true` if both of its operands are `true`, and `false` otherwise.
2. `||` (logical OR): This operator returns `true` if at least one of its operands is `true`, and `false` otherwise.
3. `!` (logical NOT): This operator negates the Boolean value of its operand. If the operand is `true`, it returns `false`, and if the operand is `false`, it returns `true`.

Logical Operators Basic example :



```
bool a = true;
bool b = false;

bool resultAND = a && b; // result is false

bool resultOr = a || b; // result is true

bool resultNOT = !a; // result is false
```

Logical Operators Advance example :



```
//Logical operators advance example
int x = 5;
int y = 10;
bool a = (x < y) && (y < 20); // a is true
bool b = (x > y) || (y > 20); // b is false
bool c = !(x == y); // c is true
```

Code Academy
أكاديمية البرمجة

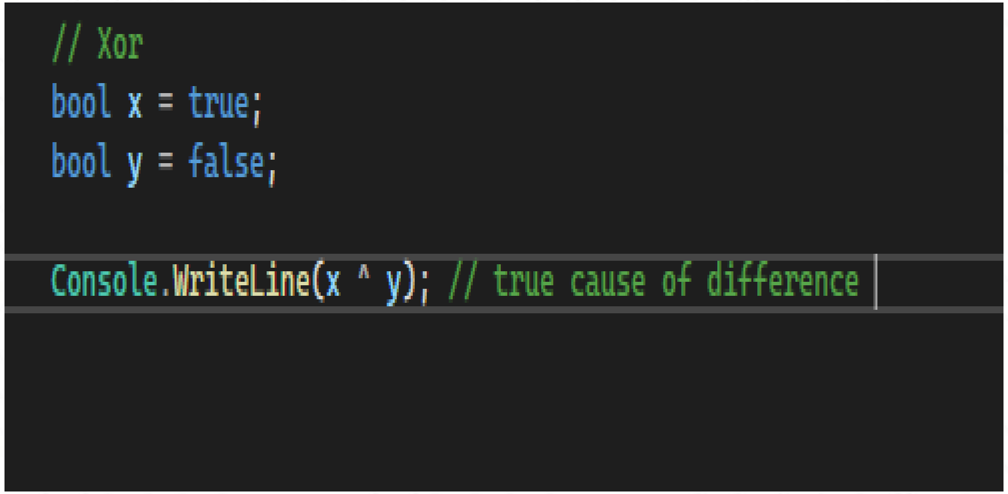
Simple Search Task :

Search about xor operator and write and example with it (10 min)

Code Academy
أكاديمية البرمجة

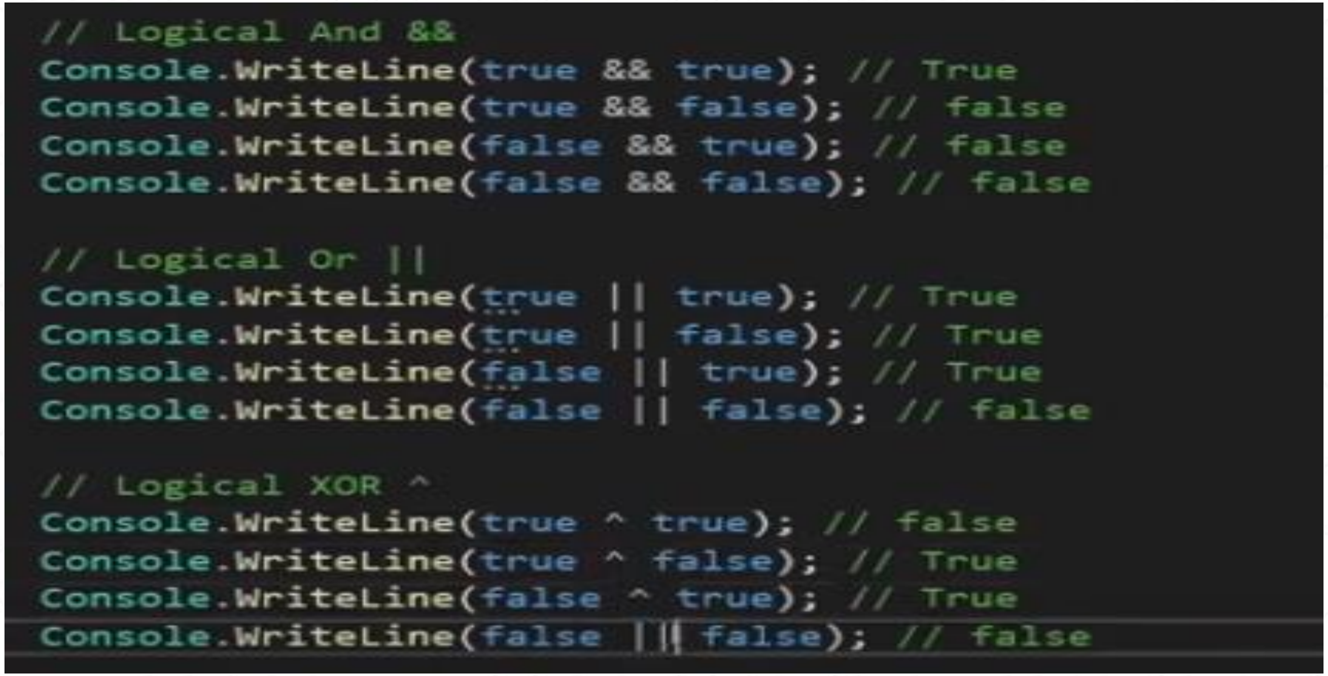
Xor is a logical operator :

Xor is conditioning the 2 expressions if they are different it will give us True otherwise in similarity will give False



```
// Xor
bool x = true;
bool y = false;

Console.WriteLine(x ^ y); // true cause of difference
```

Logical operators Cheat-sheet :

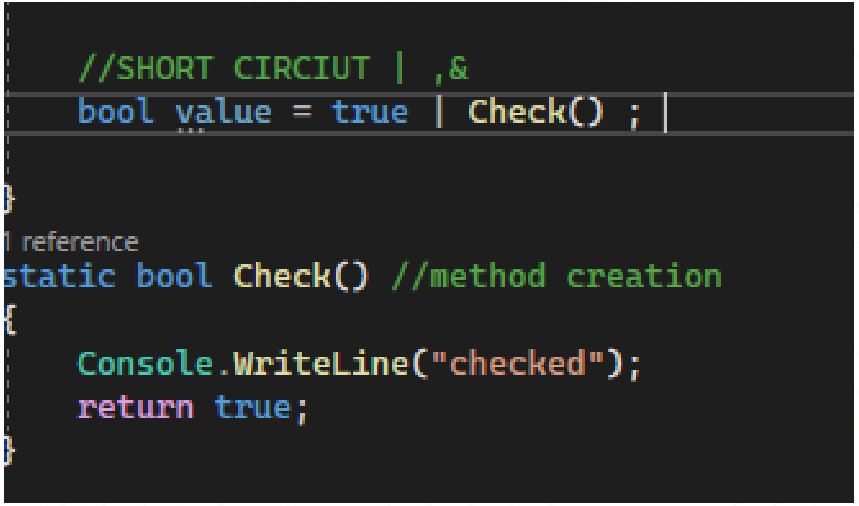


```
// Logical And &&
Console.WriteLine(true && true); // True
Console.WriteLine(true && false); // false
Console.WriteLine(false && true); // false
Console.WriteLine(false && false); // false

// Logical Or ||
Console.WriteLine(true || true); // True
Console.WriteLine(true || false); // True
Console.WriteLine(false || true); // True
Console.WriteLine(false || false); // false

// Logical XOR ^
Console.WriteLine(true ^ true); // false
Console.WriteLine(true ^ false); // True
Console.WriteLine(false ^ true); // True
Console.WriteLine(false || false); // false
```

Short Circle logical operator & - | :

While you are dealing with logical operator ‘ && ’ , ‘ || ’ the compiler doesn’t complete the full expression but it wait to get the First True or False and then break the line with the result but when we use short Circuits the compiler complete all the Line to Satisfy the expression .

It will work same as the && , || but that form is more efficient rather than the other form of the short Circuit



```
    //SHORT CIRCIUT | ,&
    bool value = true | Check() ;

}
1 reference
static bool Check() //method creation
{
    Console.WriteLine("checked");
    return true;
}
```

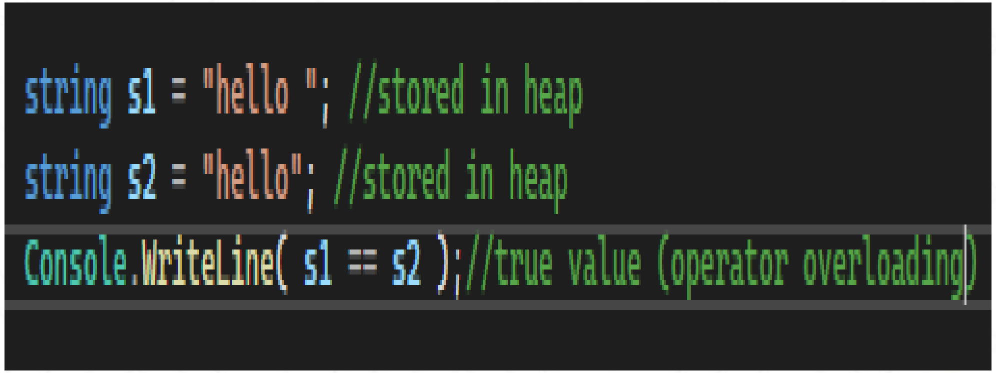
Code Academy
أكاديمية البرمجة

Reference Type Comparison

As we talked before the Reference type store the variable_name(location) in stack but the Value in heap .

So if we tried to compare between the Values of reference type it will give us false because we are comparing between 2 different locations .

But when you try to compare 2 strings (only in string) that will compare the values and that what we call operator overloading

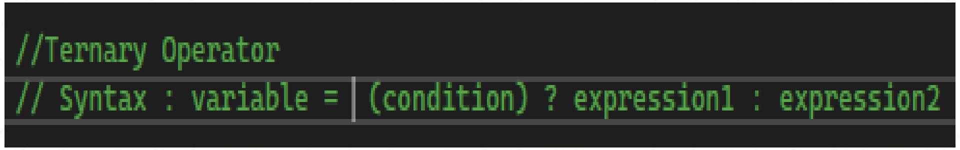


```
string s1 = "hello "; //stored in heap
string s2 = "hello"; //stored in heap
Console.WriteLine( s1 == s2 );//true value (operator overloading)
```

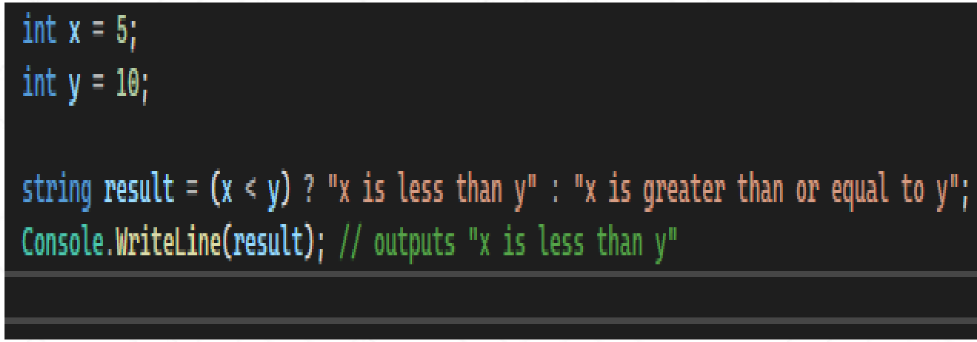
Code Academy
أكاديمية البرمجة

Ternary operator

Ternary operators are used to evaluate a Boolean expression and return one of two values, depending on whether the expression is true or false. These operators are also known as ternary operators because they take three operands: the condition, the expression to return if the condition is true, and the expression to return if the condition is false.



```
//Ternary Operator
// Syntax : variable = (condition) ? expression1 : expression2
```



```
int x = 5;
int y = 10;

string result = (x < y) ? "x is less than y" : "x is greater than or equal to y";
Console.WriteLine(result); // outputs "x is less than y"
```

Code Academy
أكاديمية البرمجة

Session Recap

Code Academy
أكاديميةالبرمجة

Any Questions ?

Thank You

Email : youssef.mohamed@amit-learning.com

Linked-in : https://www.linkedin.com/in/youssif-mohamed-450795157/