

Operadores lógicos y condicionales

Operador de negación lógico !

El operador `!` calcula la negación del operando, eso quiere decir que genera `true` si el operando se evalúa como `false` y `false`, si se evalúa como `true`.

```
bool passed = false;
Console.WriteLine(!passed); // output: True
Console.WriteLine(!true);   // output: False
```

Operando AND lógico &

El operador `&` calcula el operador AND lógico de sus operandos. El resultado de `x & y` es `true` si `x` e `y` se evalúan como `true`. De lo contrario, el resultado es `false`.

El operando `&` evalúa ambos operandos, incluso si el izquierdo da como resultado `false`

```
bool SecondOperand() {
    Console.WriteLine("Second operand is evaluated.");
    return true;
}

bool a = false & SecondOperand();
Console.WriteLine(a);
// Output:
// Second operand is evaluated.
// False

bool b = true & SecondOperand();
Console.WriteLine(b);
// Output:
// Second operand is evaluated.
// True
```

Operando IR exclusivo lógico ^

El operador `^` calcula el operador OR exclusiva lógica, también conocido como el operador XOR lógico, de sus operandos. El resultado de `x ^ y` es `true` si `x` se evalúa como `true` e `y` se evalúa como `false` o `x` se evalúa como `false` e `y` se evalúa como `true`. De lo contrario, el resultado es `false`. Es decir, para los operandos `bool`, el operador `^` calcula el mismo resultado como el operador de desigualdad `!=`.

```
Console.WriteLine(true ^ true);   // output: False
Console.WriteLine(true ^ false);  // output: True
```

```
Console.WriteLine(false ^ true);    // output: True
Console.WriteLine(false ^ false);   // output: False
```

Operador OR |

El operador | calcula el operador OR lógico de sus operandos. El resultado de `x | y` es `true` si `x` o `y` se evalúan como `true`. De lo contrario, el resultado es `false`.

El operador | evalúa ambos operandos, incluso aunque el izquierdo se evalúe como `true`, de modo que el resultado debe ser `true` con independencia del valor del operando derecho.

```
bool SecondOperand() {
    Console.WriteLine("Second operand is evaluated.");
    return true;
}

bool a = true | SecondOperand();
Console.WriteLine(a);
// Output:
// Second operand is evaluated.
// True

bool b = false | SecondOperand();
Console.WriteLine(b);
// Output:
// Second operand is evaluated.
// True
```

Operador AND lógico condicional &&

El operador AND lógico condicional &&, también denominado operador AND lógico "de cortocircuito", calcula el operador AND lógico de sus operandos. El resultado de `x && y` es `true` si `x` y `y` se evalúan como `true`. De lo contrario, el resultado es `false`. Si `x` se evalúa como `false`, `y` no se evalúa.

```
bool SecondOperand() {
    Console.WriteLine("Second operand is evaluated.");
    return true;
}

bool a = false && SecondOperand();
Console.WriteLine(a);
// Output:
// False

bool b = true && SecondOperand();
Console.WriteLine(b);
// Output:
```

```
// Second operand is evaluated.  
// True
```

Operador OR lógico condicional ||

El operador OR lógico condicional `||`, también denominado operador OR lógico "de cortocircuito", calcula el operador OR lógico de sus operandos. El resultado de `x || y` es `true` si `x` o `y` se evalúan como `true`. De lo contrario, el resultado es `false`. Si `x` se evalúa como `true`, `y` no se evalúa.

```
bool SecondOperand() {  
    Console.WriteLine("Second operand is evaluated.");  
    return true;  
}  
  
bool a = true || SecondOperand();  
Console.WriteLine(a);  
// Output:  
// True  
  
bool b = false || SecondOperand();  
Console.WriteLine(b);  
// Output:  
// Second operand is evaluated.  
// True
```