**Relational and comparison operator.**

( ==, !=, >, <, >=, <= )

Two expressions can be compared using relational and equality operators. For example, to know if two values are equal or if one is greater than the other.

The result of such an operation is either true or false (i.e., a Boolean value).

The relational operators in C++ are:

|  |  |
| --- | --- |
| **operator** | **description** |
| == | equal to |
| != | not equal to |
| < | less than |
| > | greater than |
| <= | less than or equal to |
| >= | greater than or equal to |

Here there are some examples:

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 | (7 == 5) // evaluates to false  (5 > 4) // evaluates to true  (3 != 2) // evaluates to true  (6 >= 6) // evaluates to true  (5 < 5) // evaluates to false |  |

Of course, it's not just numeric constants that can be compared, but just any value, including, of course, variables. Suppose that a=2, b=3 and c=6, then:

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 | (a == 5) // evaluates to false, since a is not equal to 5  (a\*b >= c) // evaluates to true, since (2\*3 >= 6) is true  (b+4 > a\*c) // evaluates to false, since (3+4 > 2\*6) is false  ((b=2) == a) // evaluates to true |  |

Be careful! The assignment operator (operator =, with one equal sign) is not the same as the equality comparison operator (operator ==, with two equal signs); the first one (=) assigns the value on the right-hand to the variable on its left, while the other (==) compares whether the values on both sides of the operator are equal. Therefore, in the last expression ((b=2) == a), we first assigned the value 2 to *b* and then we compared it to *a* (that also stores the value 2), yielding true.