

PASCAL - BOOLEAN OPERATORS

http://www.tutorialspoint.com/pascal/pascal_boolean_operators.htm

Copyright © tutorialspoint.com

Following table shows all the Boolean operators supported by Pascal language. All these operators work on Boolean operands and produce Boolean results. Assume variable **A** holds true and variable **B** holds false, then –

| Operator | Description | Example |
|----------|---|----------------------------|
| and | Called Boolean AND operator. If both the operands are true, then condition becomes true. | <i>AandB</i> is false. |
| and then | It is similar to the AND operator, however, it guarantees the order in which the compiler evaluates the logical expression. Left to right and the right operands are evaluated only when necessary. | <i>AandthenB</i> is false. |
| or | Called Boolean OR Operator. If any of the two operands is true, then condition becomes true. | <i>AorB</i> is true. |
| or else | It is similar to Boolean OR, however, it guarantees the order in which the compiler evaluates the logical expression. Left to right and the right operands are evaluated only when necessary. | <i>AorelseB</i> is true. |
| not | Called Boolean NOT Operator. Used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make it false. | <i>not AandB</i> is true. |

The following example illustrates the concept –

```
program beLogical;
var
a, b: boolean;
begin
    a := true;
    b := false;

    if (a and b) then
        writeln('Line 1 - Condition is true' )
    else
        writeln('Line 1 - Condition is not true');
    if (a or b) then
        writeln('Line 2 - Condition is true' );

    (* lets change the value of a and b *)
    a := false;
    b := true;

    if (a and b) then
        writeln('Line 3 - Condition is true' )
    else
        writeln('Line 3 - Condition is not true' );

    if not (a and b) then
        writeln('Line 4 - Condition is true' );
end.
```

When the above code is compiled and executed, it produces the following result –

```
Line 1 - Condition is not true  
Line 2 - Condition is true  
Line 3 - Condition is not true  
Line 4 - Condition is true
```

```
Loading [Mathjax]/jax/output/HTML-CSS/jax.js
```