IF

IF determines whether a numeric or string value is within a given range

Syntax

IF(expression1, ComparisonArgument, expression2)

Arguments

- expression1 is the argument parameter (or expression) whereas expression2 is the value which is compared with expression1.
- ComparisonArgument determines the comparison of the expressions. Possible arguments are:
  - EQ -> Equal to (=)
  - NE -> Not Equal to ()
  - LT -> Lower Then (<)
  - GT -> Greater Then (>)
  - LE -> Lower or Equal to (<=)
  - GE -> Greater of Equal to (=>)
- Please note that in addition to the two letter operators you can also use the logical operators itself.

Remarks

1. The function returns 0 if FALSE and 1 if TRUE
2. If the expressions are string values, the comparison yields results according to Visual Basic conventions for string comparison.
3. IF() returns a Boolean value (0 or 1). CONSTRAINTs have a general drawback that they need to be checked continuously during the model assembling process. If one is not fulfilled any more, the Modeller must adapt the template which is often rather time consuming. IF() offers an elegant solution if frequent adaptations of the system of equations are expected in multi-case calculations. By making one equation consisting of several parts (or ranges), we can set the invalid parts equal to zero by multiplying them by:

IF(x,GT,lower bound value)*IF(x,LT,upper bound value)

or
IF(x>lower bound value)*IF(x<upper bound value)

A disadvantage is that the complete equation must be evaluated including all of its parameters instead of only the valid part. For Concept Exploration, however, it may be the best solution. If such a calculation is in progress, user interaction is not further required if validity is dealt with in this way.

Quick links: Functions overview | Attribute overview | Constants overview | Dimensions overview