# MATLAB - LOGICAL OPERATIONS

 

MATLAB offers two types of logical operators and functions −

- Element-wise − these operators operate on corresponding elements of logical arrays.
- Short-circuit − these operators operate on scalar, logical expressions.

Element-wise logical operators operate element-by-element on logical arrays. The symbols &, |, and ~ are the logical array operators AND, OR, and NOT.

Short-circuit logical operators allow short-circuiting on logical operations. The symbols && and || are the logical short-circuit operators AND and OR.

## Example

Create a script file and type the following code −

```
a = 5;
b = 20;
   if ( a && b )
      disp('Line 1 - Condition is true');
   end
   if ( a || b )
      disp('Line 2 - Condition is true');
   end
   % lets change the value of  a and b
   a = 0;
   b = 10;
   if ( a && b )
      disp('Line 3 - Condition is true');
   else
      disp('Line 3 - Condition is not true');
   end
   if (~(a && b))

      disp('Line 4 - Condition is true');
   end
```

When you run the file, it produces following result −

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

## Functions for Logical Operations

Apart from the above-mentioned logical operators, MATLAB provides the following commands or functions used for the same purpose −

| Function | Description |
|---|---|
| and$A$, $B$ | Finds logical AND of array or scalar inputs; performs a logical AND of all input arrays A, B, etc. and returns an array containing elements set to either logical 1 *true* or logical 0 *false*. An element of the output array is set to 1 if all input arrays contain a nonzero element at that same array location. Otherwise, that element is set to 0. |
| not$A$ | Finds logical NOT of array or scalar input; performs a logical NOT of input array A and returns an array containing elements set to either logical 1 *true* or logical 0 *false*. An element of the output array is set to 1 |

| | |
|---|---|
| | if the input array contains a zero value element at that same array location. Otherwise, that element is set to 0. |
| or*A*, *B* | Finds logical OR of array or scalar inputs; performs a logical OR of all input arrays A, B, etc. and returns an array containing elements set to either logical 1 *true* or logical 0 *false*. An element of the output array is set to 1 if any input arrays contain a nonzero element at that same array location. Otherwise, that element is set to 0. |
| xor*A*, *B* | Logical exclusive-OR; performs an exclusive OR operation on the corresponding elements of arrays A and B. The resulting element C *i*, *j*, . . . is logical true 1 if A*i*, *j*, . . . or B*i*, *j*, . . . , but not both, is nonzero. |
| all*A* | Determine if all array elements of array A are nonzero or true.<br>• If A is a vector, all*A* returns logical 1 *true* if all the elements are nonzero and returns logical 0 *false* if one or more elements are zero.<br>• If A is a nonempty matrix, all*A* treats the columns of A as vectors, returning a row vector of logical 1's and 0's.<br>• If A is an empty 0-by-0 matrix, all*A* returns logical 1 *true*.<br>• If A is a multidimensional array, all*A* acts along the first non-singleton dimension and returns an array of logical values. The size of this dimension reduces to 1 while the sizes of all other dimensions remain the same. |
| all*A*, *dim* | Tests along the dimension of A specified by scalar *dim*. |
| any*A* | Determine if any array elements are nonzero; tests whether any of the elements along various dimensions of an array is a nonzero number or is logical 1 *true*. The any function ignores entries that are NaN *NotaNumber* .<br>• If A is a vector, any*A* returns logical 1 *true* if any of the elements of A is a nonzero number or is logical 1 *true*, and returns logical 0 *false* if all the elements are zero.<br>• If A is a nonempty matrix, any*A* treats the columns of A as vectors, returning a row vector of logical 1's and 0's.<br>• If A is an empty 0-by-0 matrix, any*A* returns logical 0 *false*.<br>• If A is a multidimensional array, any*A* acts along the first non-singleton dimension and returns an array of logical values. The size of this dimension reduces to 1 while the sizes of all other dimensions remain the same. |
| any*A*, *dim* | Tests along the dimension of A specified by scalar *dim*. |
| false | Logical 0 *false* |
| false*n* | is an n-by-n matrix of logical zeros |
| false*m*, *n* | is an m-by-n matrix of logical zeros. |
| false*m*, *n*, *p*, . . . | is an m-by-n-by-p-by-... array of logical zeros. |
| false*size*(*A*) | is an array of logical zeros that is the same size as array A. |
| false. . . , *'like'* , *p* | is an array of logical zeros of the same data type and sparsity as the logical array p. |
| ind = find*X* | Find indices and values of nonzero elements; locates all nonzero |

| | |
|---|---|
| | elements of array X, and returns the linear indices of those elements in a vector. If X is a row vector, then the returned vector is a row vector; otherwise, it returns a column vector. If X contains no nonzero elements or is an empty array, then an empty array is returned. |
| ind = find$X, k$<br>ind = find$X, k, 'first'$ | Returns at most the first k indices corresponding to the nonzero entries of X. k must be a positive integer, but it can be of any numeric data type. |
| ind = find$X, k, 'last'$ | returns at most the last k indices corresponding to the nonzero entries of X. |
| [row,col] = find$X, \ldots$ | Returns the row and column indices of the nonzero entries in the matrix X. This syntax is especially useful when working with sparse matrices. If X is an N-dimensional array with N > 2, col contains linear indices for the columns. |
| [row,col,v] = find$X, \ldots$ | Returns a column or row vector v of the nonzero entries in X, as well as row and column indices. If X is a logical expression, then v is a logical array. Output v contains the non-zero elements of the logical array obtained by evaluating the expression X. |
| islogical$A$ | Determine if input is logical array; returns true if A is a logical array and false otherwise. It also returns true if A is an instance of a class that is derived from the logical class. |
| logical$A$ | Convert numeric values to logical; returns an array that can be used for logical indexing or logical tests. |
| true | Logical 1 *true* |
| true$n$ | is an n-by-n matrix of logical ones. |
| true$m, n$ | is an m-by-n matrix of logical ones. |
| true$m, n, p, \ldots$ | is an m-by-n-by-p-by-... array of logical ones. |
| true$size(A)$ | is an array of logical ones that is the same size as array A. |
| true$\ldots, 'like', p$ | is an array of logical ones of the same data type and sparsity as the logical array p. |