MATLAB - OPERATORS

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. MATLAB is designed to operate primarily on whole matrices and arrays. Therefore, operators in MATLAB work both on scalar and non-scalar data. MATLAB allows the following types of elementary operations −

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operations
- Set Operations

Arithmetic Operators

MATLAB allows two different types of arithmetic operations −

- Matrix arithmetic operations
- Array arithmetic operations

Matrix arithmetic operations are same as defined in linear algebra. Array operations are executed element by element, both on one-dimensional and multidimensional array.

The matrix operators and array operators are differentiated by the period . symbol. However, as the addition and subtraction operation is same for matrices and arrays, the operator is same for both cases. The following table gives brief description of the operators −

Show Examples

Operator	Description
+	Addition or unary plus. A+B adds the values stored in variables A and B. A and B must have the same size, unless one is a scalar. A scalar can be added to a matrix of any size.
-	Subtraction or unary minus. A-B subtracts the value of B from A. A and B must have the same size, unless one is a scalar. A scalar can be subtracted from a matrix of any size.
*	Matrix multiplication. C = A*B is the linear algebraic product of the matrices A and B. More precisely, $$C(i,j) = \sum_{k=1}^{n} A(i,k)B(k,j)$$ For non-scalar A and B, the number of columns of A must be equal to the number of rows of B. A scalar can multiply a matrix of any size.
.*	Array multiplication. A.*B is the element-by-element product of the arrays A and B. A and B must have the same size, unless one of them is a scalar.
/	Slash or matrix right division. B/A is roughly the same as B*inv$A$. More precisely, B/A = $A'\backslash B'$'.
./	Array right division. A./B is the matrix with elements Ai,j/Bi,j. A and B must have the same size, unless one of them is a scalar.

****	Backslash or matrix left division. If A is a square matrix, A\B is roughly the same as inv$A$*B, except it is computed in a different way. If A is an n-by-n matrix and B is a column vector with n components, or a matrix with several such columns, then X = A\B is the solution to the equation $AX = B$. A warning message is displayed if A is badly scaled or nearly singular.
.	Array left division. A.\B is the matrix with elements Bi, j/Ai, j. A and B must have the same size, unless one of them is a scalar.
^	Matrix power. X^p is X to the power p, if p is a scalar. If p is an integer, the power is computed by repeated squaring. If the integer is negative, X is inverted first. For other values of p, the calculation involves eigenvalues and eigenvectors, such that if [V,D] = eig$X$, then X^p = V*D.^p/V.
.^	Array power. A.^B is the matrix with elements Ai, j to the Bi, j power. A and B must have the same size, unless one of them is a scalar.
'	Matrix transpose. A' is the linear algebraic transpose of A. For complex matrices, this is the complex conjugate transpose.
.'	Array transpose. A.' is the array transpose of A. For complex matrices, this does not involve conjugation.

Relational Operators

Relational operators can also work on both scalar and non-scalar data. Relational operators for arrays perform element-by-element comparisons between two arrays and return a logical array of the same size, with elements set to logical 1 *true* where the relation is true and elements set to logical 0 *false* where it is not.

The following table shows the relational operators available in MATLAB:

Show Examples

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Logical Operators

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 and, 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.

Show Examples

Bitwise Operations

Bitwise operators work on bits and perform bit-by-bit operation. The truth tables for &, |, and ^ are as follows −

p	q	p & q	p | q	p ^ q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

Assume if A = 60; and B = 13; Now in binary format they will be as follows:

A = 0011 1100

B = 0000 1101

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A = 1100 0011

MATLAB provides various functions for bit-wise operations like 'bitwise and', 'bitwise or' and 'bitwise not' operations, shift operation, etc.

The following table shows the commonly used bitwise operations:

Show Examples

Function	Purpose
bitand*a, b*	Bit-wise AND of integers *a* and *b*
bitcmp*a*	Bit-wise complement of *a*
bitget*a, pos*	Get bit at specified position *pos*, in the integer array *a*
bitor*a, b*	Bit-wise OR of integers *a* and *b*
bitset*a, pos*	Set bit at specific location *pos* of *a*
bitshift*a, k*	Returns *a* shifted to the left by *k* bits, equivalent to multiplying by 2^k. Negative values of k correspond to shifting bits right or dividing by $2^{\|k\|}$ and rounding to the nearest integer towards negative infinite. Any overflow bits are truncated.
bitxor*a, b*	Bit-wise XOR of integers *a* and *b*
swapbytes	Swap byte ordering

Set Operations

MATLAB provides various functions for set operations, like union, intersection and testing for set

membership, etc.

The following table shows some commonly used set operations −

Show Examples

Function	Description
intersect$A$, B	Set intersection of two arrays; returns the values common to both A and B. The values returned are in sorted order.
intersect$A$, B,$'rows'$	Treats each row of A and each row of B as single entities and returns the rows common to both A and B. The rows of the returned matrix are in sorted order.
ismember$A$, B	Returns an array the same size as A, containing 1 *true* where the elements of A are found in B. Elsewhere, it returns 0 *false*.
ismember$A$, B,$'rows'$	Treats each row of A and each row of B as single entities and returns a vector containing 1 *true* where the rows of matrix A are also rows of B. Elsewhere, it returns 0 *false*.
issorted$A$	Returns logical 1 *true* if the elements of A are in sorted order and logical 0 *false* otherwise. Input A can be a vector or an N-by-1 or 1-by-N cell array of strings. **A is considered to be sorted if A** and the output of sort$A$ are equal.
issorted$A$,$'rows'$	Returns logical 1 *true* if the rows of two-dimensional matrix A is in sorted order, and logical 0 *false* otherwise. **Matrix A is considered to be sorted if A** and the output of sortrows$A$ are equal.
setdiff$A$, B	Sets difference of two arrays; returns the values in A that are not in B. The values in the returned array are in sorted order.
setdiff$A$, B,$'rows'$	Treats each row of A and each row of B as single entities and returns the rows from A that are not in B. The rows of the returned matrix are in sorted order. The 'rows' option does not support cell arrays.
setxor	Sets exclusive OR of two arrays
union	Sets union of two arrays
unique	Unique values in array