



Basics & Syntax

Basic Commands

<code>clc</code>	Clears the command window.
<code>clear</code>	Removes all variables from the workspace.
<code>clear x y</code>	Removes variables <code>x</code> and <code>y</code> from the workspace.
<code>who</code>	Lists current variables in the workspace.
<code>whos</code>	Lists variables and their sizes, bytes, and class.
<code>help command</code>	Displays help for the specified <code>command</code> .
<code>doc command</code>	Opens the documentation page for the specified <code>command</code> .
<code>exit</code> or <code>quit</code>	Closes MATLAB.

Data Types

<code>double</code>	Default numeric type (double-precision floating point).
<code>single</code>	Single-precision floating point.
<code>int8</code> , <code>int16</code> , <code>int32</code> , <code>int64</code>	Signed integer types.
<code>uint8</code> , <code>uint16</code> , <code>uint32</code> , <code>uint64</code>	Unsigned integer types.
<code>char</code>	Character array (string).
<code>logical</code>	Boolean type (true or false).
<code>cell</code>	Cell array (can hold different data types).
<code>struct</code>	Structure array (fields with different data types).

Operators

<code>=</code>	Assignment operator.
<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>^</code>	Arithmetic operators (addition, subtraction, multiplication, division, exponentiation).
<code>./</code> , <code>.-</code> , <code>.*</code> , <code>./</code> , <code>.^</code>	Element-wise arithmetic operators.
<code>==</code> , <code>~=</code> , <code>></code> , <code><</code> , <code>>=</code> , <code><=</code>	Relational operators (equal, not equal, greater than, less than, greater than or equal, less than or equal).
<code>&&</code> , <code> </code> , <code>~</code>	Logical operators (AND, OR, NOT).
<code>:</code>	Colon operator (creates sequences).

Arrays and Matrices

Array Creation

<code>[1 2 3]</code>	Creates a row vector.
<code>[1; 2; 3]</code>	Creates a column vector.
<code>[1 2; 3 4]</code>	Creates a 2x2 matrix.
<code>1:5</code>	Creates a row vector from 1 to 5 (step size 1).
<code>1:0.5:5</code>	Creates a row vector from 1 to 5 (step size 0.5).
<code>linspace(0, 10, 5)</code>	Creates a row vector with 5 equally spaced points between 0 and 10.
<code>zeros(2, 3)</code>	Creates a 2x3 matrix of zeros.
<code>ones(2, 3)</code>	Creates a 2x3 matrix of ones.
<code>eye(3)</code>	Creates a 3x3 identity matrix.

Array Indexing

<code>A(i, j)</code>	Accesses the element in the <i>i</i> -th row and <i>j</i> -th column of matrix <code>A</code> .
<code>A(i, :)</code>	Accesses the <i>i</i> -th row of matrix <code>A</code> .
<code>A(:, j)</code>	Accesses the <i>j</i> -th column of matrix <code>A</code> .
<code>A(1:3, 2:4)</code>	Accesses a submatrix of <code>A</code> .
<code>A(:)</code>	Accesses all elements of <code>A</code> as a column vector.

Matrix Operations

<code>A + B</code>	Matrix addition.
<code>A - B</code>	Matrix subtraction.
<code>A * B</code>	Matrix multiplication.
<code>A .* B</code>	Element-wise multiplication.
<code>A / B</code>	Matrix right division (<code>A * inv(B)</code>).
<code>A \ B</code>	Matrix left division (<code>inv(A) * B</code>).
<code>A .^ B</code>	Element-wise exponentiation.
<code>A'</code>	Transpose of matrix <code>A</code> .
<code>inv(A)</code>	Inverse of matrix <code>A</code> .

Control Flow & Functions

Conditional Statements

```

if condition
    % statements
elseif condition
    % statements
else
    % statements
end
    
```

Looping

```

for i = 1:10
    % statements
end

while condition
    % statements
end
    
```

Functions

```

function [output1, output2] =
my_function(input1, input2)
    % statements
    output1 = ...;
    output2 = ...;
end

Anonymous functions:

f = @(x) x.^2;
result = f(5); % result is 25
    
```

Switch Statement

```
switch expression
case value1
    % statements
case value2
    % statements
otherwise
    % statements
end
```

Plotting

Basic Plots

<code>plot(x, y)</code>	Creates a 2D line plot of <code>y</code> versus <code>x</code> .
<code>scatter(x, y)</code>	Creates a scatter plot of <code>y</code> versus <code>x</code> .
<code>bar(x, y)</code>	Creates a bar chart.
<code>histogram(data)</code>	Creates a histogram of the data.
<code>pie(data)</code>	Creates a pie chart.

Plot Customization

<code>title('Plot Title')</code>	Sets the title of the plot.
<code>xlabel('X-axis Label')</code>	Sets the label for the x-axis.
<code>ylabel('Y-axis Label')</code>	Sets the label for the y-axis.
<code>legend('Data Series')</code>	Adds a legend to the plot.
<code>grid on</code>	Turns the grid on.
<code>axis([xmin xmax ymin ymax])</code>	Sets the axis limits.
<code>hold on</code>	Keeps the current plot and axes properties so that subsequent plotting commands add to the existing plot.
<code>hold off</code>	Releases the current plot and axes.

3D Plotting

<code>plot3(x, y, z)</code>	Creates a 3D line plot.
<code>surf(X, Y, Z)</code>	Creates a surface plot.
<code>mesh(X, Y, Z)</code>	Creates a mesh plot.
<code>contour(X, Y, Z)</code>	Creates a contour plot.