

1 Defining Variables & Basic Arithmetic

To create a variable, use the `=` operator.

```
% Define variables
a = 10;
b = 3;
c = a + b      % The result (13) will display in the Command Window
```

Pro Tip: A semicolon (`;`) at the end of a line performs the calculation but hides the output from the screen.

2 Creating Matrices

Matrices are the building blocks of MATLAB. Use **square brackets** to define them:

- Use **spaces** to separate numbers in a row.
- Use **semicolons** to start a new row.

```
% Create a 2x2 matrix
A = [1 2; 3 4]
B = [5 6; 7 8]
```

3 Matrix Multiplication

In MATLAB, the `*` operator performs standard linear algebra matrix multiplication (row-by-column).

$$C = A \times B$$

```
% Standard matrix multiplication
C = A * B
```

4 Solving Linear Systems (The Backslash)

One of MATLAB's most powerful features is solving systems of linear equations in the form $Ax = B$. Instead of calculating an inverse, we use the **backslash operator** (`\`), also known as *mldivide*. To find the vector x :

```
% Solve for x in the equation Ax = B
x = A \ B
```

Note: While you could mathematically use `inv(A)*B`, the backslash operator is faster and more accurate for computers to process.

5 Summary Checklist

Task	MATLAB Command
Assign a Variable	<code>x = 5</code>
Create a Matrix	<code>A = [1, 2; 3, 4]</code>
Create a Column Vector	<code>v = [1; 2; 3]</code>
Matrix Multiplication	<code>A * B</code>
Solving Linear Systems ($Ax = B$)	<code>x = A \ B</code>

*Created with the assistance of Google Gemini.