



DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING

---

**Title: Introduction to MATLAB**

---

DATA COMMUNICATION LAB  
CSE 308



GREEN UNIVERSITY OF BANGLADESH

# 1 Objective(s)

- To familiarize students with basic functionality of MATLAB.
- To introduce MATLAB syntax and basic programming constructs.
- To develop proficiency in implementing simple algorithms and functions in MATLAB.

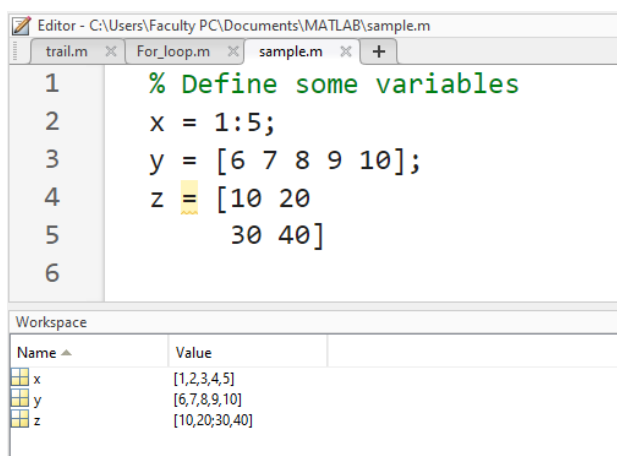
# 2 Workspace

In MATLAB, the **Workspace** refers to the memory space where variables are stored and manipulated during a MATLAB session.

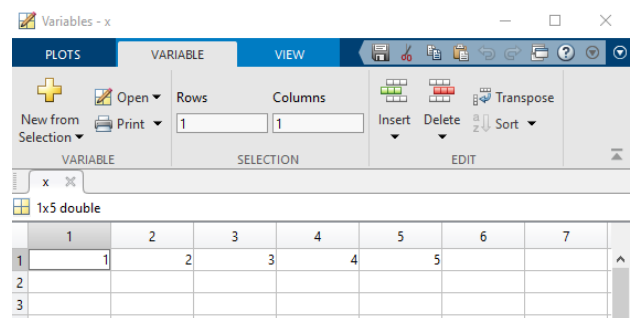
- **Storage of Variables:** When you create variables or load data into MATLAB, they are stored in the workspace. This includes arrays, matrices, strings, structures, and any other data types you use.
- **Variable Browser:** MATLAB provides a Variable Browser tool that allows you to view and manage variables in the workspace more conveniently. You can access it through the MATLAB Desktop environment.

# 3 Basic Codes

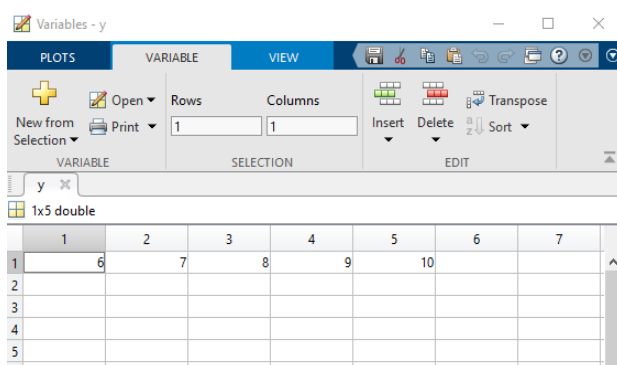
In MATLAB, the if statement is used for conditional execution, allowing code to behave differently based on the evaluation of a logical expression. If the condition is true, the code block following the if statement is executed. Optionally, else and elseif statements can be used to specify alternative code blocks to execute if the condition is false or if additional conditions are true, respectively.



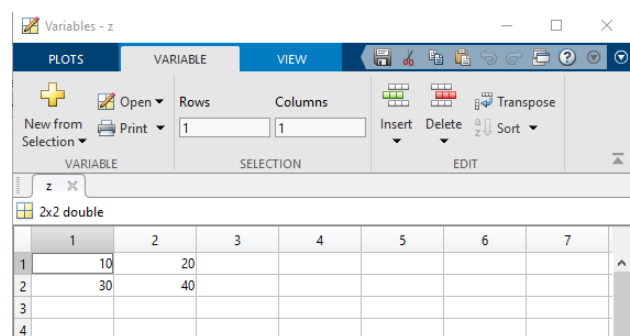
(a) Stored values of variables in Workspace



(b) Values of 'x' in Workspace



(c) Values of 'y' in Workspace



(d) Values of a matrix 'z' in Workspace

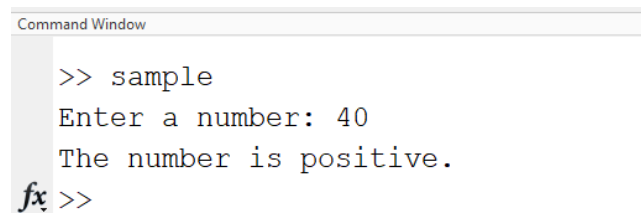
Figure 1: Distribution of different values in Workspace

In short, if statements in MATLAB provide a way to control the flow of execution based on the evaluation of logical conditions. They are fundamental for implementing decision-making processes in MATLAB code.

## If Else

Matlab code using if else	
1	<code>% Input a number</code>
2	<code>num = input('Enter a number: ');</code>
3	
4	<code>% Check if the number is positive, negative, or zero</code>
5	<code>if num &gt; 0</code>
6	<code>    disp('The number is positive.');</code>
7	<code>elseif num &lt; 0</code>
8	<code>    disp('The number is negative.');</code>
9	<code>else</code>
10	<code>    disp('The number is zero.');</code>
11	<code>end</code>

Table 1: MATLAB code using if else statement



```

Command Window

>> sample
Enter a number: 40
The number is positive.
fx >>

```

Figure 2: Output of the code

This MATLAB code prompts the user to input a number. After receiving the input, it checks whether the number is positive, negative, or zero using if-elseif-else conditional statements.

- **disp Function:** In MATLAB, `disp` is a function used to display text or the value of a variable in the command window or script output. In this code, `disp` is used to output messages indicating whether the input number is positive, negative, or zero.
- **Implicit Type Conversion:** MATLAB performs implicit type conversion when necessary. For instance, the `input` function returns a string by default, and MATLAB converts it to a numeric value if used in a numeric context. In this code, the input value is implicitly converted to a numeric type when used in the comparison (`num > 0`, `num < 0`, `num == 0`) without explicit casting.

### Matlab code for plotting

```
1      % Define the range of x values
2      x = -5:0.1:5
3
4      % Define the function (example: y = x^2)
5      y = x.^2
6
7      % Plot the graph
8      plot(x, y);
9      title('Graph of y = x^2');
10     xlabel('x');
11     ylabel('y');
12     grid on; % Show grid
```

Table 2: MATLAB code to plot the graph of  $y = x^2$

## Plotting

This MATLAB code generates a plot of the function  $y = x^2$  over the range of  $x$  values from -5 to 5 with a step size of 0.1. Let's break down each part:

- `x = -5:0.1:5`: This line creates a vector  $x$  containing values from -5 to 5 with a step size of 0.1. This range specifies the  $x$ -axis values for plotting.
- `y = x.^2`: This line calculates the corresponding  $y$ -values for each  $x$ -value in the vector  $x$  by squaring each element. This represents the function  $y = x^2$ .
- `plot(x, y)`: This line creates a 2D plot with  $x$ -values on the horizontal axis and  $y$ -values on the vertical axis. It plots the function  $y = x^2$  using the specified  $x$  and  $y$  values.
  - `title('Graph of y = x^2')`: Sets the title of the plot to "Graph of  $y = x^2$ ".
  - `xlabel('x')`: Labels the horizontal axis as "x".
  - `ylabel('y')`: Labels the vertical axis as "y".
  - `grid on`: Displays a grid on the plot for better visualization and interpretation.

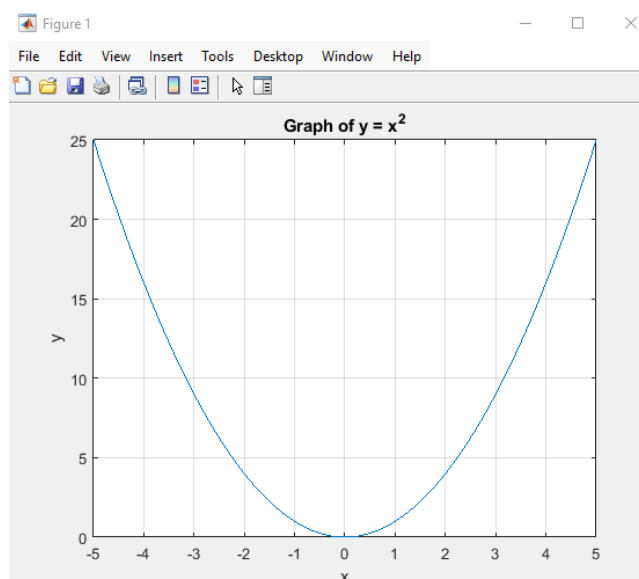


Figure 3: Output of the MATLAB code

This code is a concise and straightforward way to generate a plot of the function  $y = \sin(x)$  in MATLAB. It demonstrates basic plotting functionality, including defining data, creating plots, and labeling axes.

## For Loop

### Matlab code using loop

```
1      % Define the range of x values
2      x = -pi:0.1:pi;
3
4      % Initialize y values
5      y = zeros(size(x));
6
7      % Calculate y values using a for loop
8      for i = 1:length(x)
9          y(i) = sin(x(i));
10     end
11
12     % Plot the graph
13     plot(x, y);
14     title('Graph of y = sin(x)');
15     xlabel('x');
16     ylabel('y');
17     grid on; % Show grid
```

Table 3: MATLAB code to plot the graph of  $y = \sin(x)$

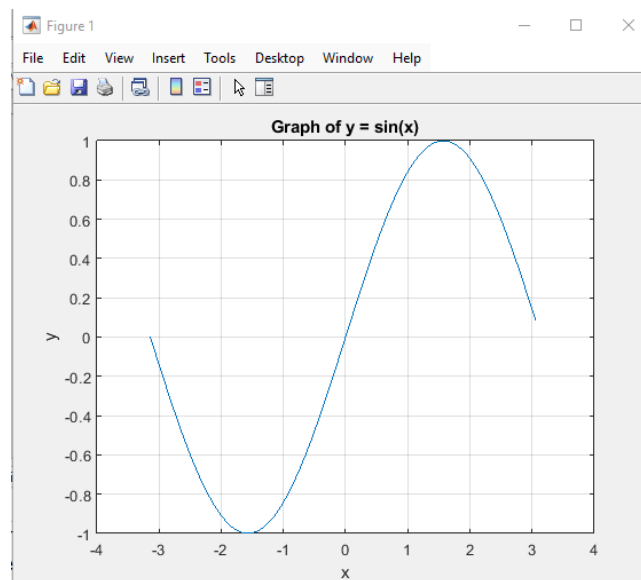


Figure 4: Output of the code

This MATLAB code generates a plot of the sine function  $y = \sin(x)$  over the range of  $x$  values from  $-\pi$  to  $\pi$  with a step size of 0.1. Let's break down each part:

- `x = -pi:0.1:pi;`: This line creates a vector  $x$  containing values from  $-\pi$  to  $\pi$  with a step size of 0.1. This range specifies the  $x$ -axis values for plotting.
- `y = zeros(size(x));`: This line initializes the vector  $y$  with zeros, ensuring it has the same size as the vector  $x$ .

`grid on;` Displays a grid on the plot for better visualization and interpretation.

## Lab Task for Students

- Create a MATLAB script that plots either the sine or cosine function based on user input. The script should allow the user to choose between plotting the sine function ( $\sin(x)$ ) or the cosine function ( $\cos(x)$ ), specify the range of  $x$  values, and plot the selected function.

## 4 Lab Exercise (Submit as a report)

- Complete the given lab task and submit as a report.

## 5 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.