

MATLAB for Beginners

Connect with me: [Youtube](#) | [LinkedIn](#) | [WhatsApp Channel](#) | [Web](#) | [Facebook](#) | [Twitter](#)

Relational operators in MATLAB

Greater Than (>) and Less Than (<) operators:

```
a = 5;
b = 7;

if a < b
    disp('a is less than b');
end

if b > a
    disp('b is greater than a');
end
```

Output:

```
a is less than b. b is greater than a
```

Greater Than or Equal To (>=) and Less Than or Equal To (<=) operators:

```
a = 5;
b = 7;
if a <= b
    disp('a is less than or equal to b');
end

if b >=a
    disp('b is greater than or equal to a');
end
```

Output:

```
a is less than or equal to b.
b is greater than or equal to a.
```

=== Equality (==) and Inequality (~=) operators:

```

a = 5;
b = 7;

if a == b
    disp('a is equal to b');
else
    disp('a is not equal to b');
end

    if a ~= b
disp('a is not equal to b');
else
    disp('a is equal to b');
end

```

Output:

```

a is not equal to b.
a is not equal to b
...

#### example

Example: Calculating the Mean and Standard Deviation of a Data Set

% Define the data set
data = [4, 8, 15, 16, 23, 42, 5, 9, 17, 29];

% Calculate the mean
mean_val = mean(data);
disp(mean_val); % Output: 17.9

% Calculate the standard deviation
std_val = std(data);
disp(std_val); % Output: 11.1423

mean function to calculate the mean value of the data set, and the std function to
calculate the standard deviation of the data set.

Sure, here's a simple MATLAB example for beginners:

```

```
% This program calculates the average of three numbers
```

```
% Input the three numbers a = input('Enter the first number: '); b = input('Enter the second number: '); c =
input('Enter the third number: ');
```

```
% Calculate the average avg = (a + b + c) / 3;
```

```
% Display the result fprintf('The average of %f, %f, and %f is %f\n', a, b, c, avg);
```

This program prompts the user to enter three numbers, calculates the average of the three numbers, and displays the result.

To run this program, save the code in a file with a `.m` extension (e.g., `average.m`), and then run the file in MATLAB by typing the name of the file (without the `.m` extension) in the MATLAB command window.

For example, if you save the code in a file called `average.m`, you can run it by typing `average` in the command window and pressing Enter.

Sure, here's a simple MATLAB example for beginners:

Example 1: Matrix multiplication

css

Copy code

```
% Define two matrices
```

```
A = [1 2 3; 4 5 6; 7 8 9];
```

```
B = [10 11 12; 13 14 15; 16 17 18];
```

```
% Multiply matrices
```

```
C = A * B;
```

```
% Print result
```

```
disp(C)
```

Example 5: Sorting an array

scss

Copy code

```
% Define an array
```

```
A = [5 2 9 3 7];
```

```
% Sort the array
```

```
sorted_A = sort(A);
```

```
% Print the sorted array
```

```
disp(sorted_A);
```

Example 4: Generating random numbers

```
% Generate a random number between 0 and 1
```

```
x = rand();
```

```
% Generate an array of random numbers between 0 and 1
```

```
y = rand(1, 10);
```

```
% Generate an array of random integers between 1 and 10
```

```
z = randi([1, 10], 1, 10);
```

```
% Display the results  
disp(x);  
disp(y);  
disp(z);
```

Example 5: Reading and writing to files

```
% Write some data to a file  
data = [1 2 3; 4 5 6; 7 8 9];  
dlmwrite('data.txt', data);
```

```
% Read the data from the file  
data = dlmread('data.txt');
```

```
% Display the data  
disp(data);
```

Example 2: Looping through an array

matlab

Copy code

```
array = [1, 2, 3, 4, 5];
```

```
for i = 1:length(array)  
    disp(array(i));  
end
```

Example 5: Generating a random matrix

matlab

Copy code

```
rows = 3;  
cols = 4;
```

```
A = rand(rows, cols);
```

```
disp(A);
```

Example 1: Calculating the sum of an array of numbers

matlab

Copy code

```
% Define the array of numbers  
numbers = [2 4 6 8 10];
```

```
% Calculate the sum of the array using the sum() function  
total = sum(numbers);
```

```
% Display the total  
disp(total);  
Vector Addition:
```

To add two vectors in MATLAB, you can use the '+' operator. Here's an example:

css

Copy code

```
a = [1, 2, 3];  
b = [4, 5, 6];  
c = a + b;
```

This will create a new vector 'c' that contains the result of adding the corresponding elements of 'a' and 'b'. In this case, the result will be [5, 7, 9].

Finding roots of a function:

To find the roots of a function in MATLAB, you can use the 'roots' function. Here's an example:

css

Copy code

```
p = [1, -4, 4];  
r = roots(p);
```

This will find the roots of the polynomial $p(x) = x^2 - 4x + 4$, which has a double root at $x = 2$. The 'roots' function returns a vector of the roots, in this case [2, 2].

Creating a vector of numbers from 1 to 10:

matlab

Copy code

```
x = 1:10
```

Creating a matrix of ones with 3 rows and 4 columns:

matlab

Copy code

```
A = ones(3, 4)
```

Defining a function that computes the area of a circle given its radius:

matlab

Copy code

```
function area = circle_area(radius)  
    area = pi * radius^2;  
end
```

Finding the maximum value in a vector:

matlab

Copy code

```
x = [1 5 3 8 2];  
max_val = max(x)
```

Sorting a vector in descending order:

matlab

Copy code

```
x = [1 5 3 8 2];  
sorted_x = sort(x, 'descend')
```

Computing the sum of the first 10 positive integers:

matlab

Copy code

```
sum = 0;  
for i = 1:10
```

```
sum = sum + i;  
end
```

In this example, we define the dimensions of a rectangle (width and height), calculate its area using the multiplication operator (``*``), and display the result using the ``disp`` function.

This is a simple example, but it demonstrates how to define variables, perform arithmetic operations, and display output in MATLAB.

Sure, here's a real-life coding example of MATLAB for beginners:

Problem Statement:

You are given a dataset containing the temperatures in Celsius and Fahrenheit for a number of cities. You need to write a MATLAB script to convert the Celsius temperatures to Fahrenheit temperatures and output the results in a table.

Solution:

Here's an example of how you can write a MATLAB script to solve this problem:

```
% Load the data from a CSV file data = readtable('temperatures.csv');  
  
% Convert the Celsius temperatures to Fahrenheit fahrenheit = (data.Celsius * 9/5) + 32;  
  
% Create a table to store the results results = table(data.City, data.Celsius, fahrenheit, 'VariableNames', {'City',  
'Celsius', 'Fahrenheit'});  
  
% Display the results disp(results);
```

Explanation:

1. The first line loads the data from a CSV file named 'temperatures.csv' and stores it in a table called 'data'.
2. The second line calculates the Fahrenheit temperatures using the Celsius temperatures in the 'Celsius' column of the 'data' table.
3. The third line creates a new table called 'results' to store the original city names, Celsius temperatures, and Fahrenheit temperatures. The 'VariableNames' argument specifies the column names for the table.
4. The fourth line displays the 'results' table in the MATLAB command window.

This is just a simple example, but it shows how MATLAB can be used to process and analyze data. You can modify this script to work with other datasets or perform different calculations based on your needs.

Sure, here's a simple real-life coding example using MATLAB for beginners:

Problem statement: Suppose you're working for a company that sells products online. You've been given a dataset that contains information about the sales of different products, such as the product name, quantity sold, and price. Your task is to calculate the total revenue generated by each product and display the results.

Solution using MATLAB:

1. Load the data from a CSV file:

```
% Load the data from a CSV file data = readtable('sales_data.csv');
```

2. Calculate the revenue for each product:

```
% Calculate the revenue for each product revenue = data.Quantity .* data.Price;
```

3. Group the revenue by product name:

```
% Group the revenue by product name grouped_revenue = groupsummary(data, 'Product', 'sum', 'Revenue', @sum);
```

4. Display the results:

```
% Display the results disp(grouped_revenue);
```

Output:

```
Product sum_Revenue
```

```
Bicycle 2000 Camera 3200 Headphones 800 Laptop 6000 Phone 4500
```

This code reads the sales data from a CSV file, calculates the revenue for each product, groups the revenue by product name, and displays the results. It uses built-in MATLAB functions, such as `readtable` and `groupsummary`, to perform the data manipulation and aggregation.

Sure, here's a real-life mathematics coding example of MATLAB for beginners:

1. Solving Equations:

MATLAB can be used to solve mathematical equations, both algebraic and differential equations. Here's an example of using MATLAB to solve a system of linear equations:

```
% Define the system of linear equations A = [1 2 3; 4 5 6; 7 8 10]; b = [6; 15; 28];
```

```
% Solve the system of linear equations x = A\b;
```

```
% Display the solution disp(x);
```

2. Plotting Graphs:

MATLAB can also be used to plot mathematical functions and graphs. Here's an example of using MATLAB to plot the sine and cosine functions:

```
% Define the range of x values x = linspace(-pi, pi);
```

```
% Compute the sine and cosine values y_sin = sin(x); y_cos = cos(x);
```

```
% Plot the sine and cosine functions plot(x, y_sin, x, y_cos); legend('sin(x)', 'cos(x)');
```

3. Calculus:

MATLAB can also be used for calculus, such as computing derivatives and integrals. Here's an example of using MATLAB to compute the derivative of a function:

```
% Define the function f = @(x) x.^2 + sin(x);
```

```
% Define the point at which to compute the derivative x0 = 1;
```

```
% Compute the derivative using the central difference method h = 0.01; df_dx = (f(x0 + h) - f(x0 - h))/(2*h);
```

```
% Display the derivative disp(df_dx);
```

These are just a few examples, but MATLAB can be used for many other mathematical applications, such as optimization, linear algebra, and statistics.

Interpolation:

MATLAB can be used for interpolation, which is the process of estimating values between known data points. You can use MATLAB to interpolate data and create smooth curves. Here's an example of using MATLAB to interpolate a set of data

points and plot the resulting curve:

```
% Define the data points
x = 0:0.1:1;
y = sin(x);

% Interpolate the data
xi = 0:0.01:1;
yi = interp1(x, y, xi, 'spline');

% Plot the data and interpolated curve
plot(x, y, 'o', xi, yi);
```

Integration:

MATLAB can also be used for numerical integration, which is the process of approximating the area under a curve. You can use MATLAB to integrate functions and calculate numerical values. Here's an example of using MATLAB to integrate a function:

```
% Define the function
f = @(x) x.^2.*sin(x);

% Integrate the function
a = 0;
b = pi/2;
I = integral(f, a, b);

% Display the result
disp(I);
```

Sure, here is a real-life mathematics coding example of MATLAB for beginners:

Solving a system of linear equations:

MATLAB can be used to solve systems of linear equations, which arise in many real-life applications, such as optimization problems, circuit analysis, and economics. Here's an example of using MATLAB to solve a system of linear equations:

```
% Define the coefficients and constants A = [1, 2, 3; 4, 5, 6; 7, 8, 9]; b = [10; 11; 12];
```

```
% Solve the system of equations x = A \ b;
```

```
% Display the solution disp(x);
```

This code defines a system of three linear equations in three variables, represented by the matrix equation $Ax = b$, where A is the coefficient matrix, x is the vector of unknowns, and b is the vector of constants. The code uses the backslash operator to solve for x and then displays the solution.

You can also use MATLAB to visualize the solution of a system of linear equations. Here's an example of using MATLAB to plot the solution of a system of two linear

equations:

```
% Define the coefficients and constants A = [1, 2; 3, 4]; b = [5; 6];
```

```
% Solve the system of equations x = A \ b;
```

```
% Plot the solution plot(x(1), x(2), 'o'); hold on; x1 = linspace(0, 5, 100); x2 = (b(2) - A(2,1)*x1)/A(2,2); plot(x1, x2); xlabel('x1'); ylabel('x2');
```

This code defines a system of two linear equations in two variables, represented by the matrix equation $Ax = b$. The code uses the backslash operator to solve for x and then plots the solution in a 2D plot, along with the two equations.

```
% Solve the equation  $x^2 + 2x + 1 = 0$ 
```

```
syms x
```

```
eqn =  $x^2 + 2*x + 1 == 0$ ;
```

```
sol = solve(eqn, x);
```

```
disp(sol); % Output: -1
```

```
% Solve the quadratic equation  $ax^2 + bx + c = 0$ 
```

```
a = 1;
```

```
b = 2;
```

```
c = 1;
```

```
x1 =  $(-b + \sqrt{b^2 - 4*a*c}) / (2*a)$ ;
```

```
x2 =  $(-b - \sqrt{b^2 - 4*a*c}) / (2*a)$ ;
```

```
disp(x1); % Output: -1
```

```
disp(x2); % Output: -1
```

Linear Regression: Linear regression is a common statistical technique used to model the relationship between two variables. In Matlab, the polyfit function can be used to perform linear regression. For example, you can use linear regression to model the relationship between a company's advertising spending and its sales revenue.

```
% Generate sample data
```

```
x = [1 2 3 4 5];
```

```
y = [5 7 9 11 13];
```

```
% Perform linear regression
```

```
p = polyfit(x,y,1);
```

```
% Plot the data and the linear regression line
```

```
plot(x,y,'o')
```

```
hold on
```

```
plot(x,polyval(p,x))
```

Principal Component Analysis: Principal component analysis (PCA) is a common technique used in data analysis to reduce the dimensionality of a dataset. In Matlab, the pca function can be used to perform PCA. For example, you can use PCA

to reduce the dimensionality of a dataset representing customer purchases in a grocery store.

```
% Generate sample data
```

```
X = randn(100,3)*[1 0.5 0.1; 0.5 1 0.2; 0.1 0.2 1];
```

```
% Perform PCA
```

```
[coeff,score,latent] = pca(X);
```

```
% Plot the first two principal components
```

```
scatter(score(:,1),score(:,2))
```

```
## Social Links
```

- [WhatsApp Channel](https://whatsapp.com/channel/0029VaC3BC160eBZZSs3CW0c)
- [Web](https://yasirbhutta.github.io/)
- [Youtube](https://www.youtube.com/yasirbhutta)
- [Facebook](https://www.facebook.com/yasirbhutta786)
- [Twitter](https://twitter.com/yasirbhutta)