
Table of Contents

Clean up from previous section	1
This is a header comment. This will not be computed by the MATLAB compiler.	1
Your first program!	1
Working with variables	1
Arrays	3
Displaying to the command line	4
Solving equations	5
Conditional Statements	5
Loops	6
Functions	7
Graphing	7
Tips & Tricks	9

Clean up from previous section

```
clc;  
clear;
```

This is a header comment. This will not be computed by the MATLAB compiler.

This is a normal comment.

```
% Use comments to help clarify your code for anyone else who is reading it
```

Your first program!

```
fprintf(2, 'Your first program!\n');  
  
fprintf('Hello world!'); %% Print to screen  
  
    Your first program!  
    Hello world!
```

Working with variables

```
fprintf('\n');  
fprintf(2, '\nWorking with variables\n');  
  
% Variables MUST start with a letter and cannot be MATLAB keywords  
% to check run the 'iskeyword' command  
fprintf('Is ''if'' a MATLAB keyword?');  
iskeyword if % Returns 1 to indicate it is a MATLAB keyword  
fprintf('Is ''tree'' a MATLAB keyword?');  
iskeyword tree % Returns 0 to indicate it is NOT a MATLAB keyword  
  
% Declaring variables
```

```
sup = 1 % Without suppression
nosup = 2; % With suppression
add = 1 + 2; % Variable with addition
sub = 4 - 3; % Variable with subtraction
mult = 2 * 2; % Variable with multiplication
div = 1 / 3; % Variable with division
modu = mod(23,5); % Variable with modulus (remainder after division)
string = 'Hello world!'; % Variable containing a string

% Calculations using variables
a = 1; % Declare a variable
b = 2; % Declare a second variable

v_add = a + b; % Variable addition
v_sub = a - b; % Variable subtraction
v_mult = a * b; % Variable multiplication
v_div = a / b; % Variable division

% OUTPUTS
fprintf('nosup = %d\n',nosup);
fprintf('add = %d\n',add);
fprintf('sub = %d\n',sub);
fprintf('mult = %d\n', mult);
fprintf('div = %.2f\n',div);
fprintf('mod = %d\n\n', modu);
fprintf('string = %s\n\n', string);
fprintf('v_add = %d\n',v_add);
fprintf('v_sub = %d\n',v_sub);
fprintf('v_mult = %d\n', v_mult);
fprintf('v_div = %.2f\n',v_div);
```

```
Working with variables
Is 'if' a MATLAB keyword?
ans =

    1

Is 'tree' a MATLAB keyword?
ans =

    0

sup =

    1

nosup = 2
add = 3
sub = 1
mult = 4
div = 0.33
```

```

mod = 3

string = Hello world!

v_add = 3
v_sub = -1
v_mult = 2
v_div = 0.50

```

Arrays

```

fprintf(2, '\nArrays\n');

array_1d = [1,2,3,4] % One dimensional array declaring each variable
array_1d = 1:1:4 % One dimensional array
array_2d = [1,2,3,4;5,6,7,8] % Two dimensional array declaring each variable
array_2d = [1:1:4;5:1:8] % Two dimensional array

% Arrays of ones
array_ones_1by4 = ones(1,4)
array_ones_2by4 = ones(2,4)
array_ones_3by4 = ones(3,4)
array_ones_4by4 = ones(4)

array_zeros_4by4 = zeros(4)

array_rand_4by4 = rand(4)

% Array arithmetic
array_1d = 2 .* array_1d % INCLUDE THE PERIOD WHEN DOING CALCULATIONS INVOLVING A

```

Arrays

```

array_1d =

    1    2    3    4

```

```

array_1d =

    1    2    3    4

```

```

array_2d =

    1    2    3    4
    5    6    7    8

```

```

array_2d =

    1    2    3    4

```

```

5      6      7      8

array_ones_1by4 =

1      1      1      1

array_ones_2by4 =

1      1      1      1
1      1      1      1

array_ones_3by4 =

1      1      1      1
1      1      1      1
1      1      1      1

array_ones_4by4 =

1      1      1      1
1      1      1      1
1      1      1      1
1      1      1      1

array_zeros_4by4 =

0      0      0      0
0      0      0      0
0      0      0      0
0      0      0      0

array_rand_4by4 =

0.8013    0.5747    0.2467    0.6609
0.2278    0.8452    0.6664    0.7298
0.4981    0.7386    0.0835    0.8908
0.9009    0.5860    0.6260    0.9823

array_1d =

2      4      6      8

```

Displaying to the command line

```
fprintf(2, '\nDisplaying to the command line\n');
```

```
% Printing to seperate lines IMPROPERLY
fprintf('This is line 1. ');
fprintf('This is line 2. ');
% Printing to seperate lines PROPERLY. '\n' is a line break.
fprintf('\n');
fprintf('This is line 1.\n');
fprintf('This is line 2.\n');

fprintf('I printed an integer: %d\n', add); % Print an integer
fprintf('I printed a string: "%s"\n', string); % Print a string
fprintf('I printed a float: %.2f\n', div); % Print a float. Specify accuracy
fprintf('I printed an apostrophe: Queen's\n'); % Print an apostrophe
```

```

    This is line 1.This is line 2.
    This is line 1.
    This is line 2.
    I printed an integer: 3
    I printed a string: "Hello world!"
    I printed a float: 0.33
    I printed an apostrophe: Queen's

```

Solving equations

```
fprintf(2, '\nSolving Equations\n')

f = @(x) -x.^2 - 2.*x + 10;

y = fzero(f,1);

fprintf('The root of x^2 - 2x + 10 is %.2f.\n', y);
```

```

    Solving Equations
    The root of x^2 - 2x + 10 is 2.32.

```

Conditional Statements

```
fprintf(2, '\nConditional Statements\n');

value = 6;
fprintf('The value is %d\n', value);

if value < 5;
    fprintf('%d < 5\n', value);
elseif value >= 5 && value < 10
    fprintf('5 >= %d < 10\n', value);
else
    fprintf('This is a weird number\n');
end
```

```
Conditional Statements
The value is 6
5 >= 6 < 10
```

Loops

```
fprintf(2, '\nLoops\n');

% For loops
for i = 1:10
    fprintf('We have traversed the loop %d times\n', i); % Print the number of times
end

array_1d = ones(1,4)

fprintf('array_1d = [ %d', array_1d(1));
for ii = 2:4
    array_1d(ii) = array_1d(ii-1) + 1;
    fprintf(' %d', array_1d(ii));
end
fprintf(' ]\n');

for iii = 1:length(array_1d)
    if array_1d(iii) ~= 4
        fprintf('array_1d(%d) = %d\n', iii, array_1d(iii));
    else
        break;
    end
    fprintf('array_1d(%d) = %d\n', iii, array_1d(iii));
end

% While loops
i=0;
while i<6
    i = i+1;
end

% Use while loops when you need to run a loop an indeterminable amount of
% times, and for loops for a determinable amount of times
```

```
Loops
We have traversed the loop 1 times
We have traversed the loop 2 times
We have traversed the loop 3 times
We have traversed the loop 4 times
We have traversed the loop 5 times
We have traversed the loop 6 times
We have traversed the loop 7 times
We have traversed the loop 8 times
We have traversed the loop 9 times
We have traversed the loop 10 times
```

```
array_1d =  
  
    1    1    1    1  
  
array_1d = [ 1 2 3 4 ]  
array_1d(1) = 1  
array_1d(2) = 2  
array_1d(3) = 3
```

Functions

```
fprintf(2, '\nFunctions\n');  
  
multiply(3,2);
```

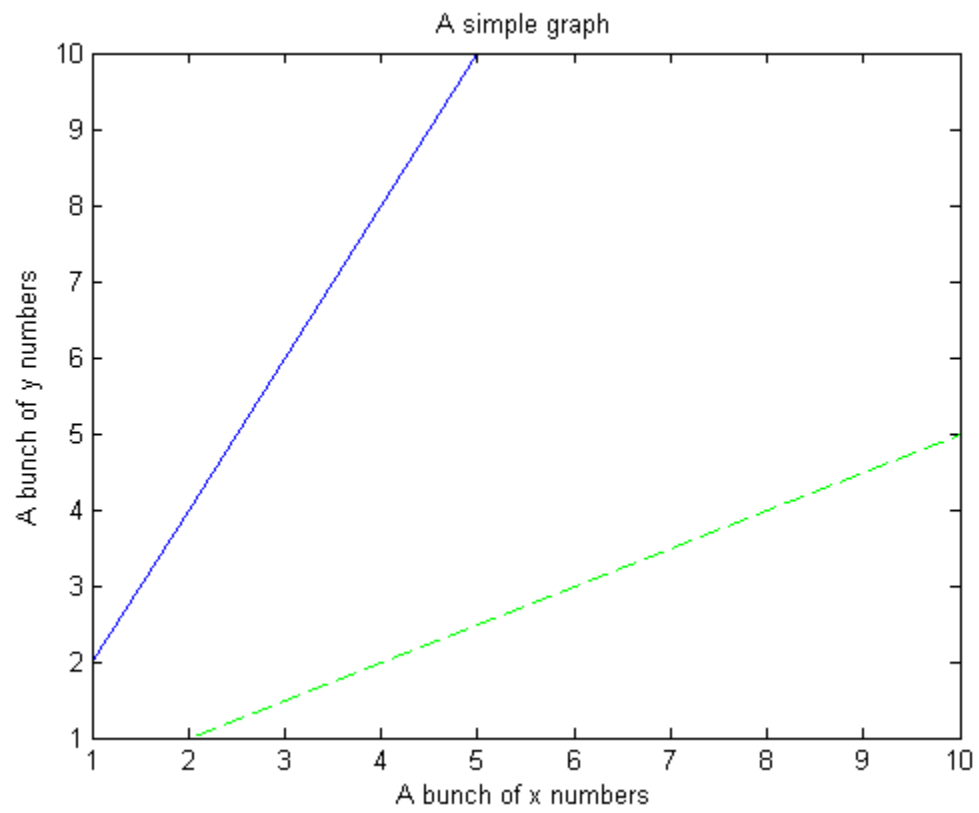
```
Functions  
6
```

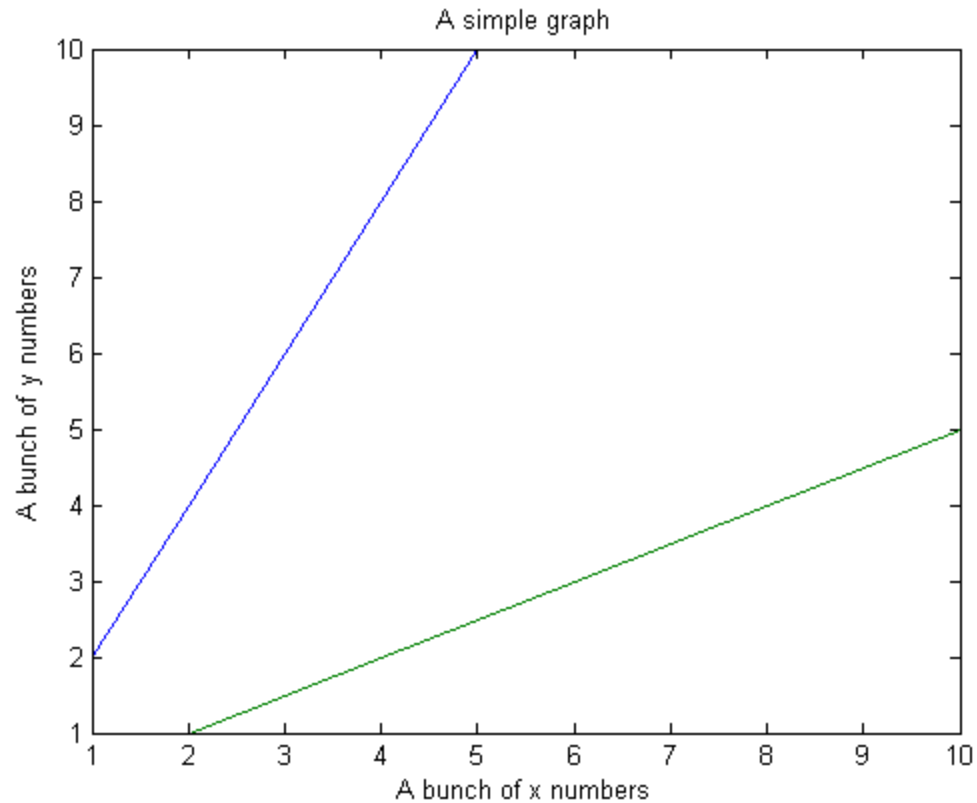
Graphing

```
fprintf(2, '\nGraphing\n');  
  
x1_axis = 1:1:5; % Create an array for the x axis numbers  
y1_axis = 2:2:10; % Create an array for the y axis numbers  
  
figure(1); % Declare the figure number  
plot(x1_axis,y1_axis); % Plot the x and y coordinates on the graph  
xlabel('A bunch of x numbers'); % Label the x axis  
ylabel('A bunch of y numbers'); % Label the y axis  
title('A simple graph'); % Title the graph  
  
x2_axis = 2:2:10;  
y2_axis = 1:1:5;  
  
hold on  
plot(x1_axis,y1_axis); % Plot the x and y coordinates on the graph  
plot(x2_axis,y2_axis,'g--'); % Plot the x and y coordinates on the graph  
xlabel('A bunch of x numbers'); % Label the x axis  
ylabel('A bunch of y numbers'); % Label the y axis  
title('A simple graph'); % Title the graph  
hold off  
  
hold on  
  
figure(3)  
plot(x1_axis,y1_axis,x2_axis,y2_axis); % Plot the x and y coordinates on the graph  
xlabel('A bunch of x numbers'); % Label the x axis  
ylabel('A bunch of y numbers'); % Label the y axis  
title('A simple graph'); % Title the graph
```

hold off

Graphing





Tips & Tricks

```
% Comment the shit out of your code. If you know how to solve the problem,  
% but don't know how to code to get the answer, comment what you think you  
% should do
```

```
% DON'T GIVE A VARIABLE AND A FUNCTION THE SAME NAME
```

```
% the 'help' command is your bestfriend, use it! Search in the top corner
```

```
% KEYBOARD SHORT CUTS
```

```
% Use 'Ctrl' + 'S' to save your work
```

```
% Use 'Ctrl' + 'C' to terminate a script (from the command window)
```

```
% COMMAND WINDOW
```

```
% 'ans' gives the most recent answer
```

```
% 'clc' clears the command window
```

```
% USEFUL HELP FILES
```

```
% Search 'elfun' for help with stuff like sin, cos, exp, etc
```

Published with MATLAB® R2014a