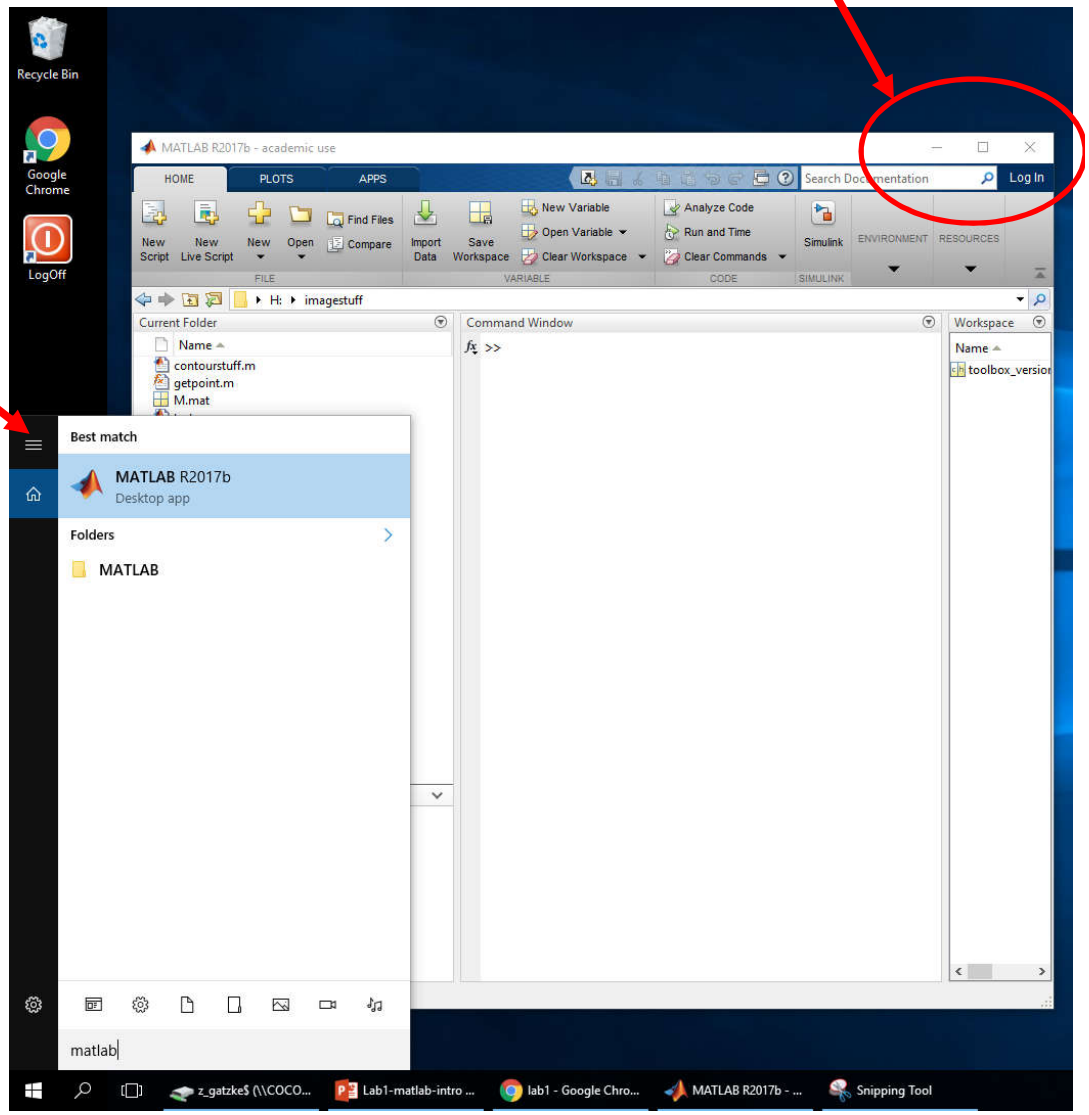
Lab 1
Introduction to MATLAB

Step 0 – Start MATLAB

- Start MATLAB. Click the search icon in lower-left and type matlab into the search bar then click on Matlab
- Use ALT+Tab to switch between windows.
- Use ALT+Shift+Tab to switch the other way.
- You may want to maximize the MATLAB window and switch between MATLAB and this PDF file.
- You may want to have the PDF next to the MATLAB widow.

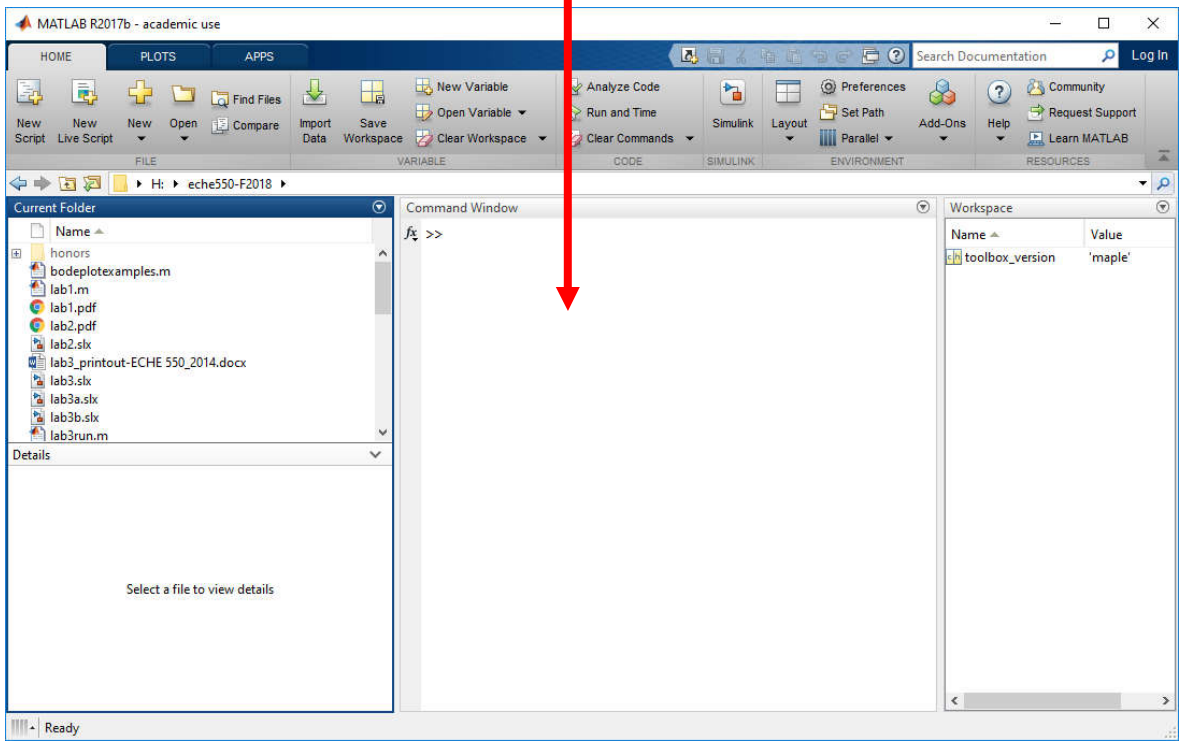


Step 1 –MATLAB Interface

- The main part of the MATLAB interface is the central Command Window
- You can type commands into this window
- At the command prompt >> in the command window type the command:

  pwd

- and hit enter.
- This displays the current directory!

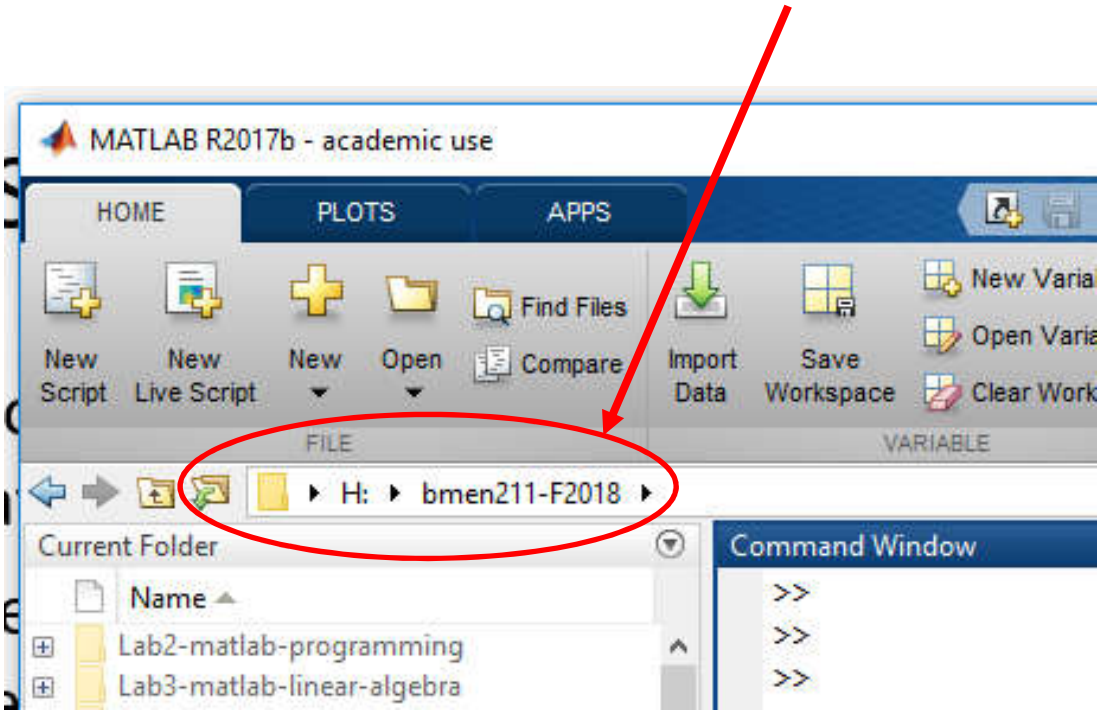


Step 2 – Set the Working Directory

- You need to set where MATLAB saves files!
- Click on the folder icon and select your working folder for this class.
- This is where you save your files!
- MATLAB files mostly have a ".m" extension.
- Example:

  lab1.m

  hw5.m

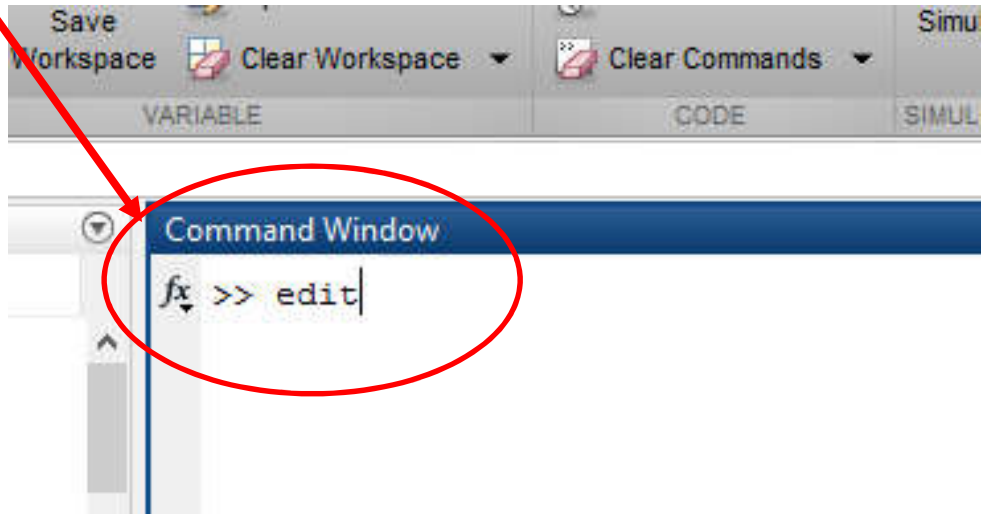


Step 3 – Start the Editor

- In the middle section where you see >> type the word:

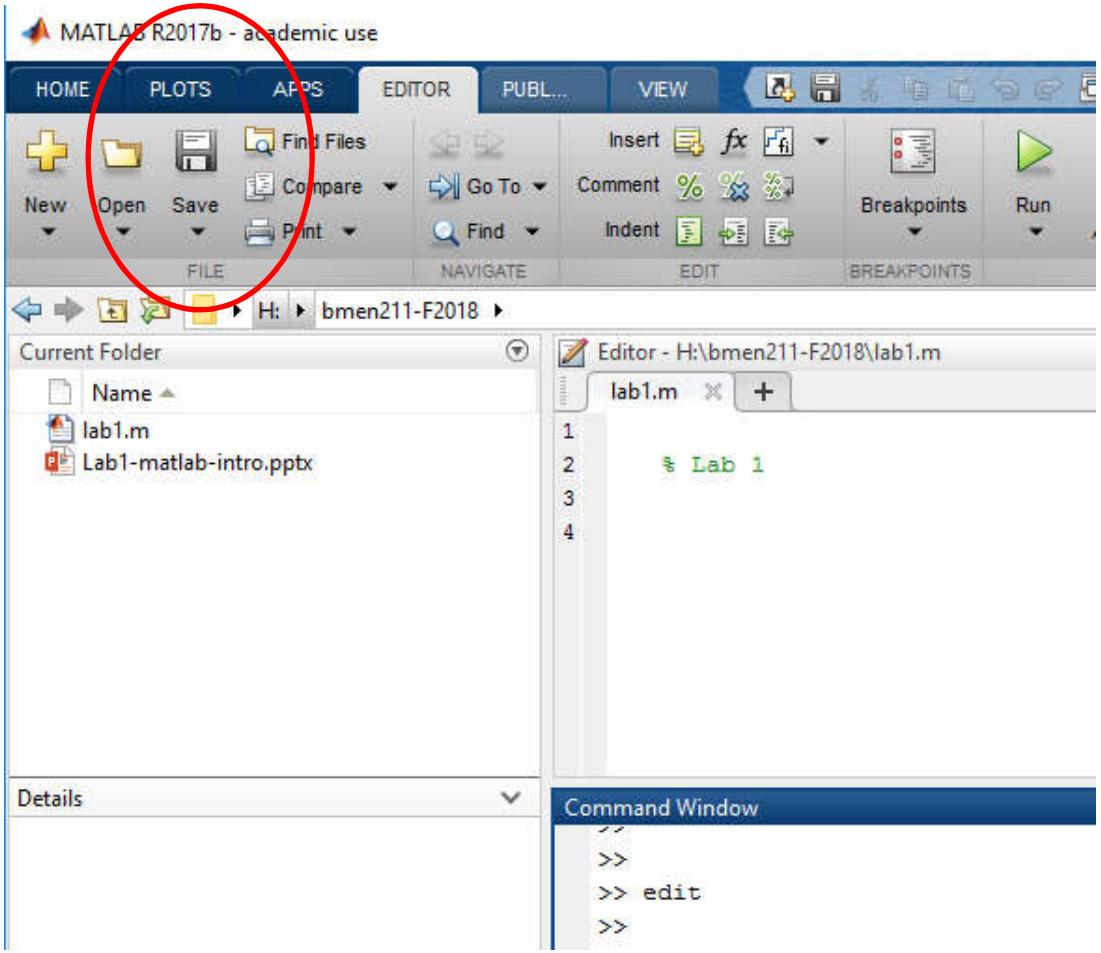
  edit

- Hit return to start the editor.
- The editor is just a text file editor, but it has extra MATLAB functions.
- The editor starts “docked” in the central part of the window.
- The command window is at the bottom now.



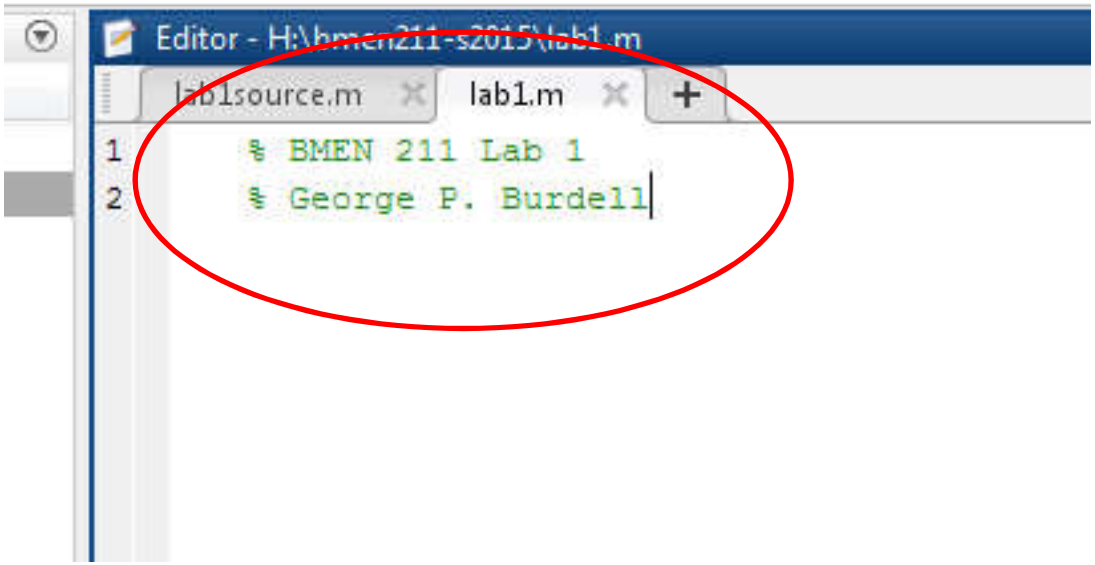
Step 4 – Save Your File

- Your MATLAB should now look like the image.
- Click on “Save” under the Editor tab to save your file.
- YOU CANNOT USE SPACES IN FILE NAMES!
- Give your file the name lab1
- The .m extension will be appended on automatically when it saves.



Step 5 – Comments

- Comments in MATLAB follow the **%** operator.
- Anything after a **%** will be ignored.
- Comments show in green in the editor.
- Add some comments to your file using the editor window.
- Obviously, use your name, not George's!

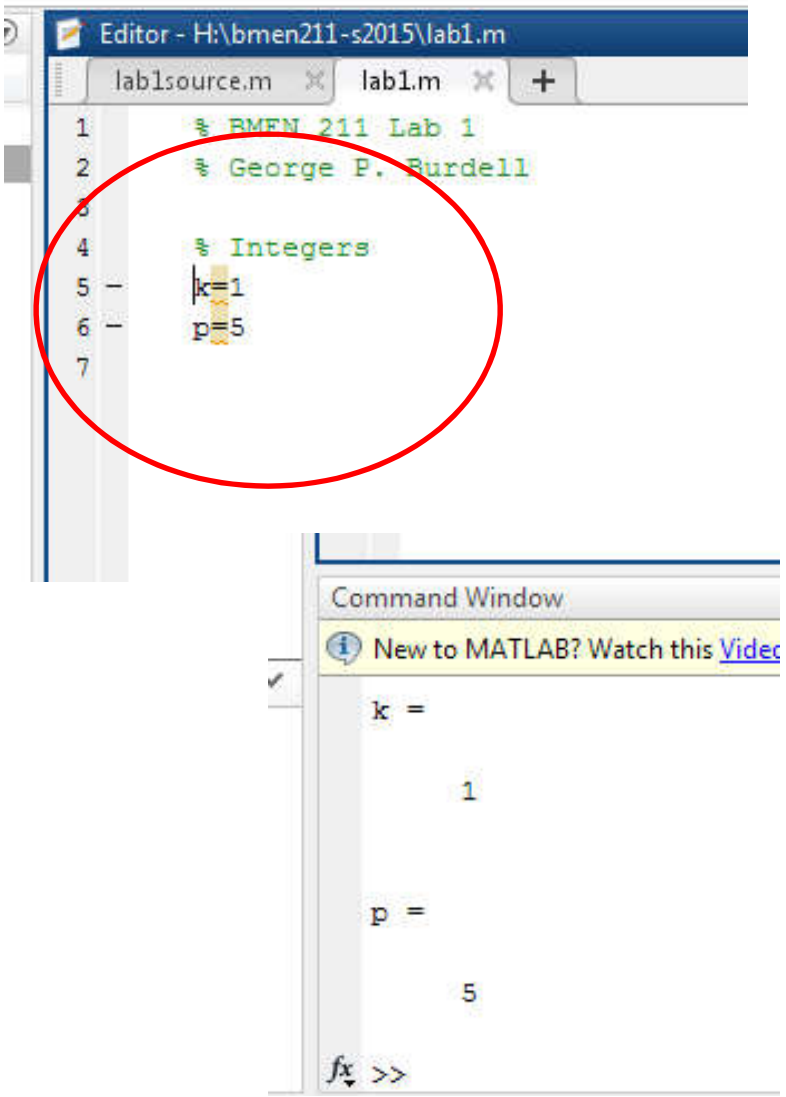


Step 6 – Make Variables with Integers

- MATLAB allows you to assign values to variables.
- In the editor window type the following two lines:

  k=1

  p=5
- Click the green triangle run button. This runs the script and you see the values in the command window.
- NOTE, the variables appear in the Workspace Window.



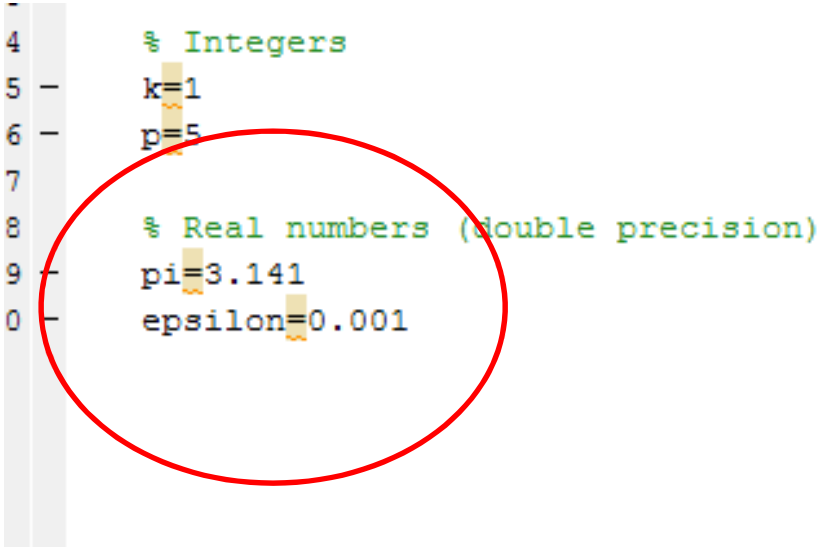
Step 7 – Make Variables with Real Values

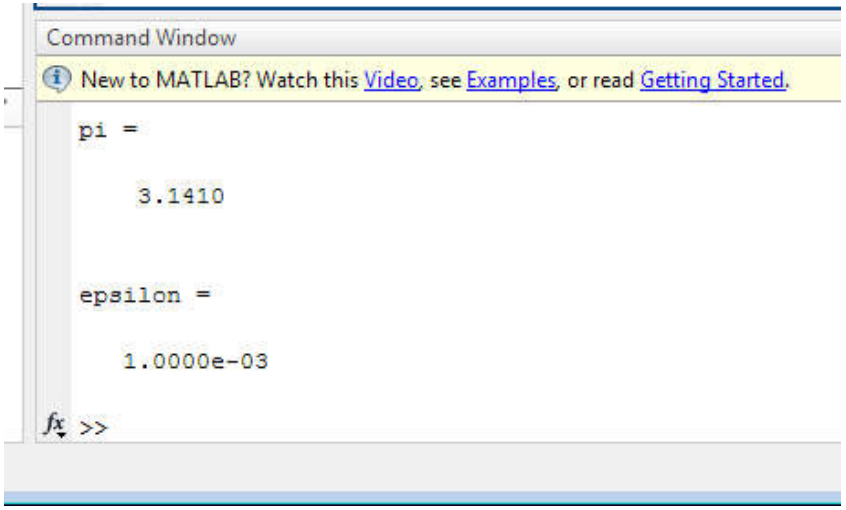
- In the editor window type the following two lines:

  pi=3.141

  epsilon=0.001

- Click the green triangle run button or hit F5. This runs the script and you see the values in the command window.
- You could type commands in the command window, but you will make many mistakes.
- NOTE, the Workspace Window is always updated.





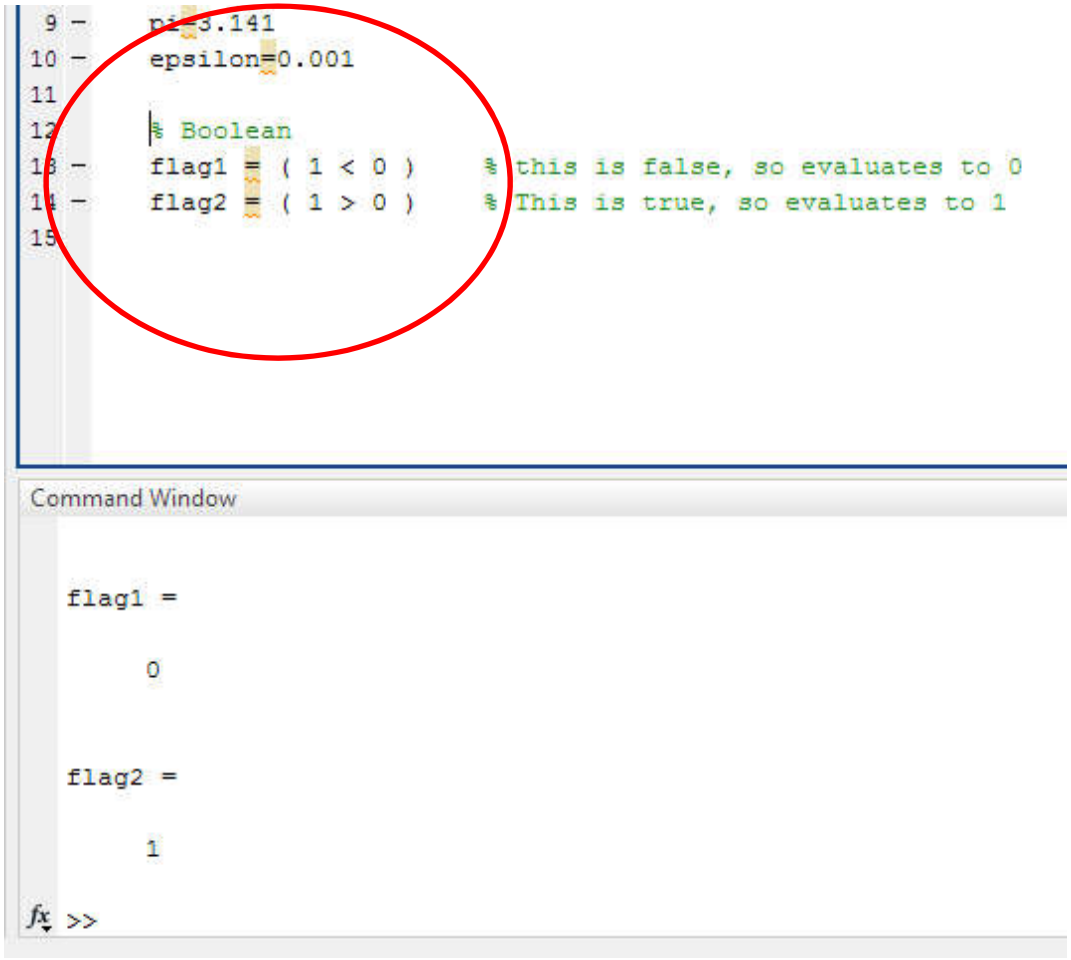
Step 8 – Boolean True/False Values

- Boolean values represent true or false.
- MATLAB uses 1 for TRUE and 0 for FALSE.
- In the editor window type the following two lines:

  flag1 = (1 < 0)

  flag2 = (1 > 0)

- Click the green triangle run button or hit F5.
- NOTE, the variables cannot have spaces or start with a number!

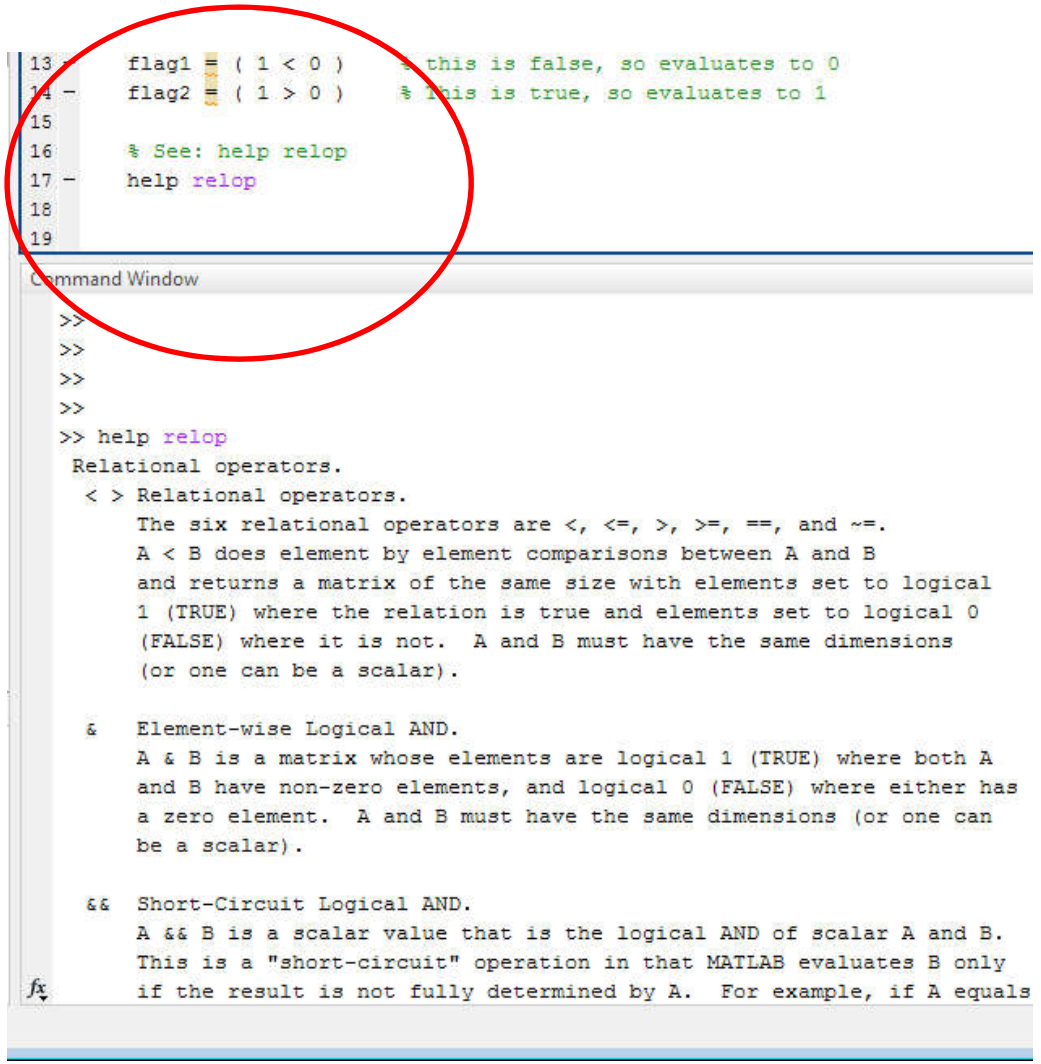


Step 9 – Help

- There are many relational operators in MATLAB.
- In MATLAB the amperstand "&" means logical AND
- In MATLAB, the pipe "|" means logical OR
- MATLAB has a help command. In the editor window type the following line:

  help relop

- Click the green triangle run button or just hit F5.
- Scroll up and down to read.
- Or type in **doc relop**

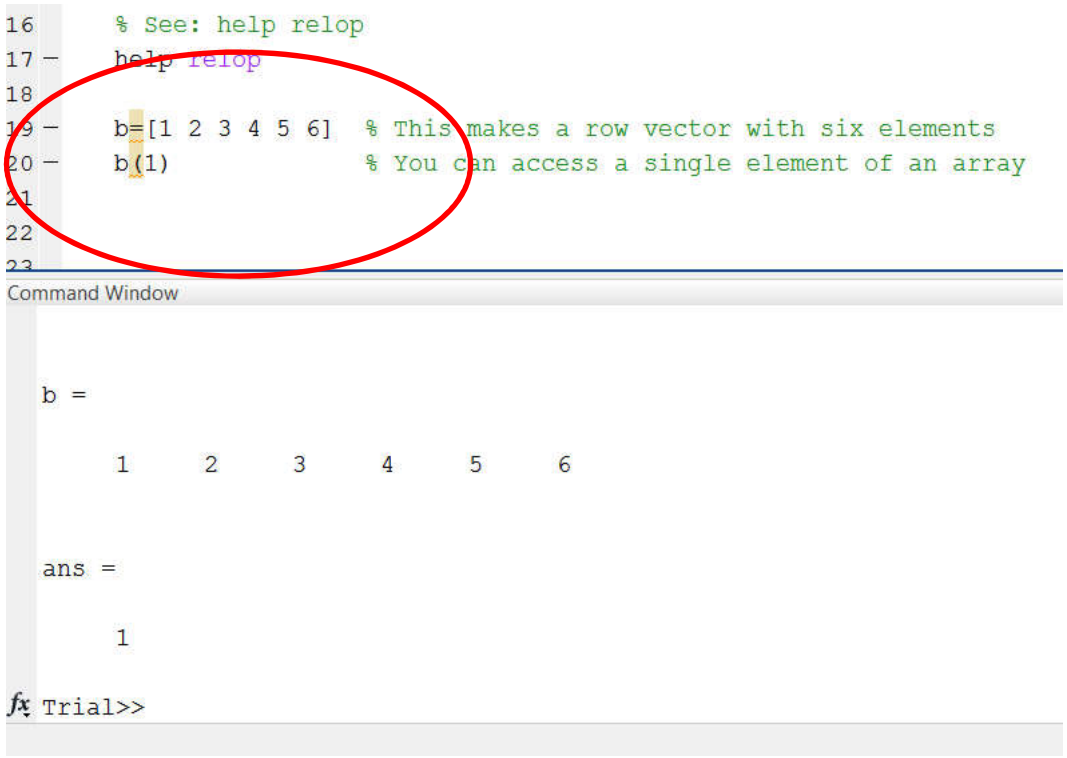


Step 10 – Arrays

- Arrays - Arrays contain multiple pieces of data indexed along one or more dimensions.
- A vector can be seen as a 1D array of real numbers.
- A matrix can be seen as 2D array of real numbers.
- Type the following into the editor window:

  b=[1 2 3 4 5 6]

  b(1)
- Run the file. What does this do?



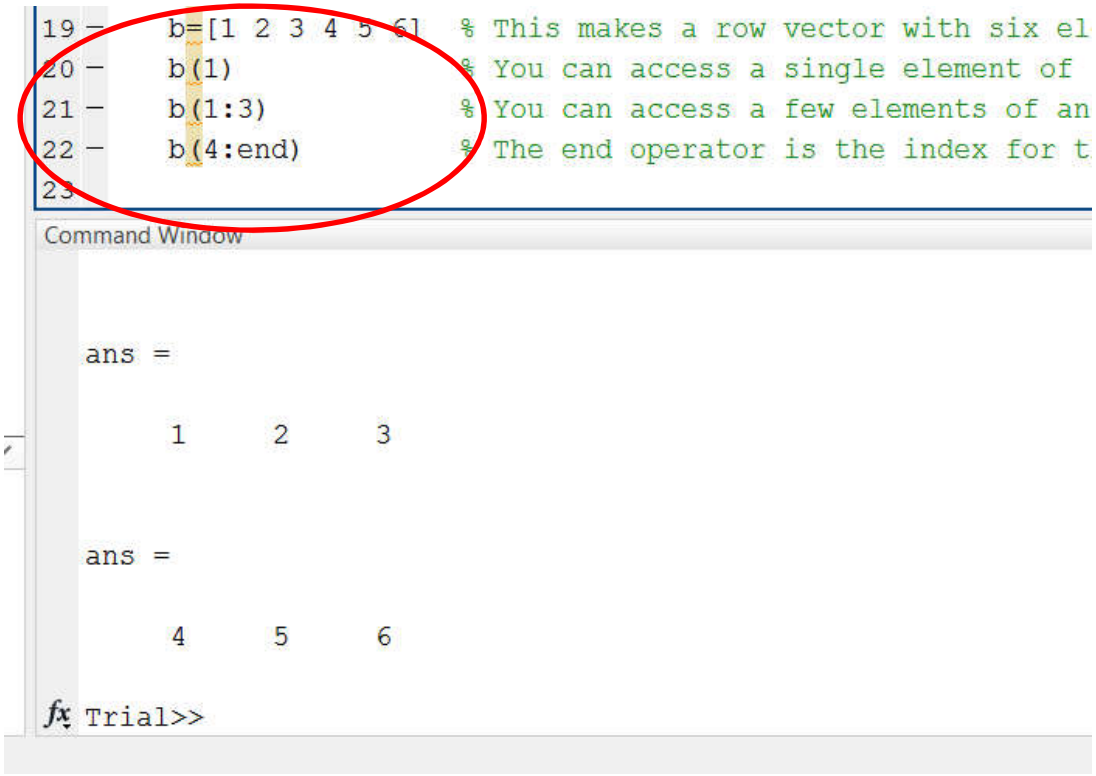
Step 11 – Array indexing

- You can make an array using square brackets.
- You can access information in an array using an index inside parenthesis.
- MATLAB starts the index counter at 1.
- Type the following into the editor window:

  b(1:3)

  b(4:end)

- Run the file. What does this do?



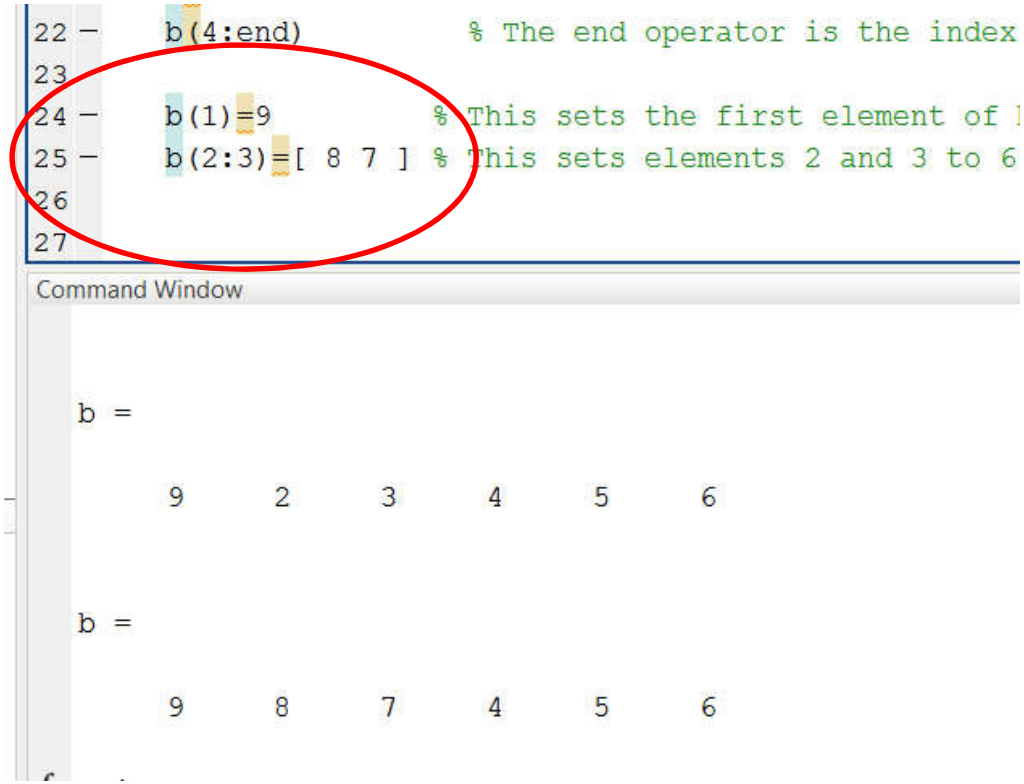
Step 12 – Changing Array Elements

- You can access multiple elements of an array.
- The end operator means the end of the array.
- You can also change elements in an array by assigning new values.
- Type the following into the editor window:

  b(1)=9

  b(2:3)=[8 7]

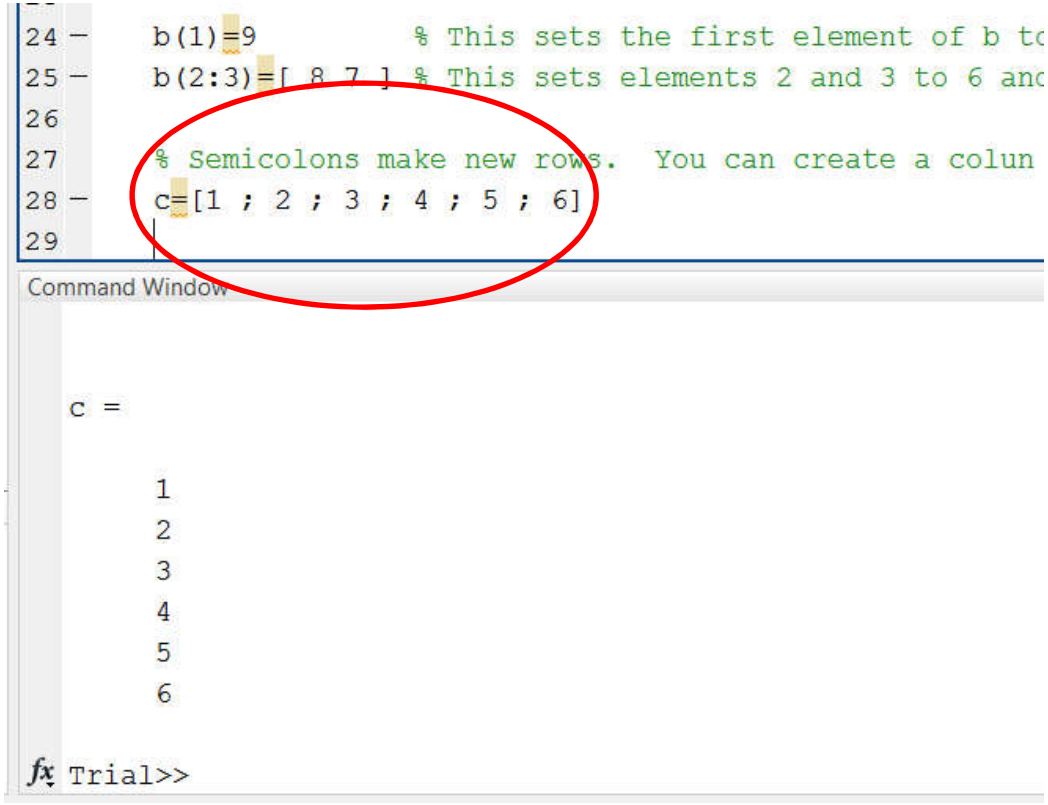
- Note: Both sides must be the same size!



Step 13 – Making New Rows

- To make new rows, use a semicolon between elements rather than spaces.
- Type the following into the editor window:

  c=[1;2;3;4;5;6]
- This makes six rows, creating a column vector with six rows and one column.
- Spaces between values make new columns, semicolons make new rows.

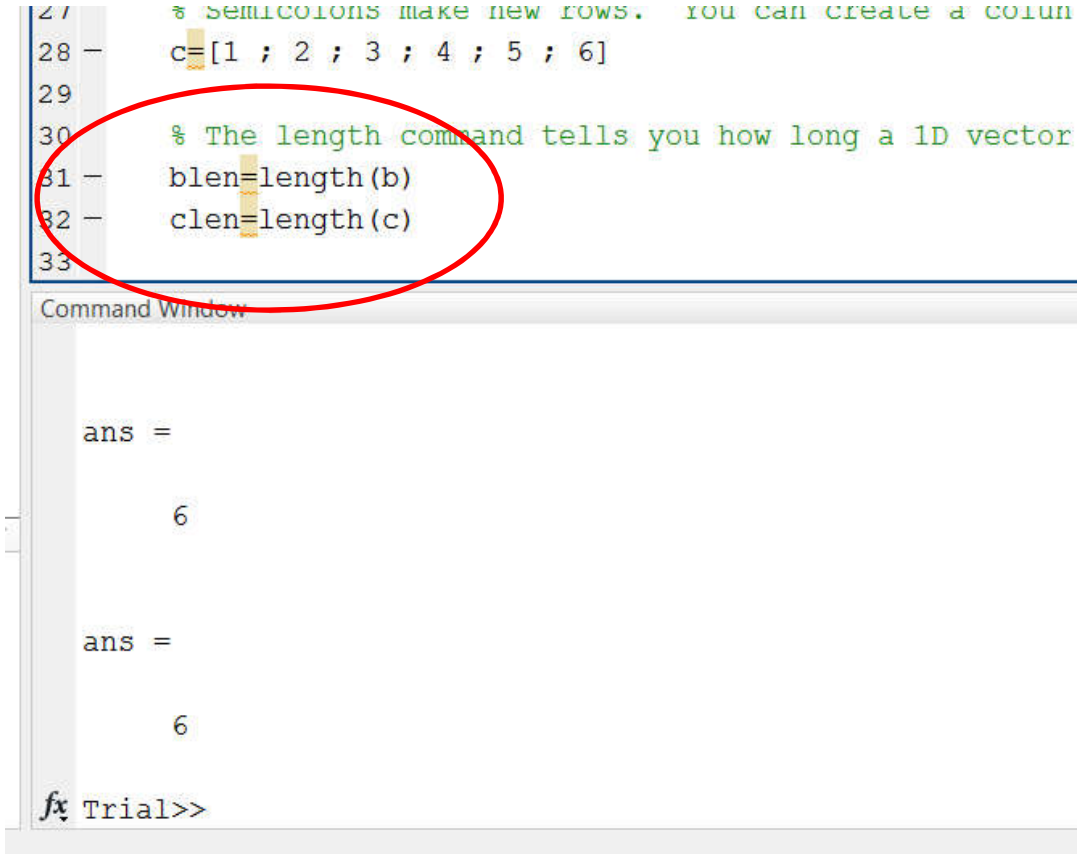


Step 14 – Using the “length” Function

- To find out how long an array is, use the length function.
- Type the following into the editor window:

  blen=length(b)

  clen=length(c)
- This makes two new variables containing the length of b and the length of c.

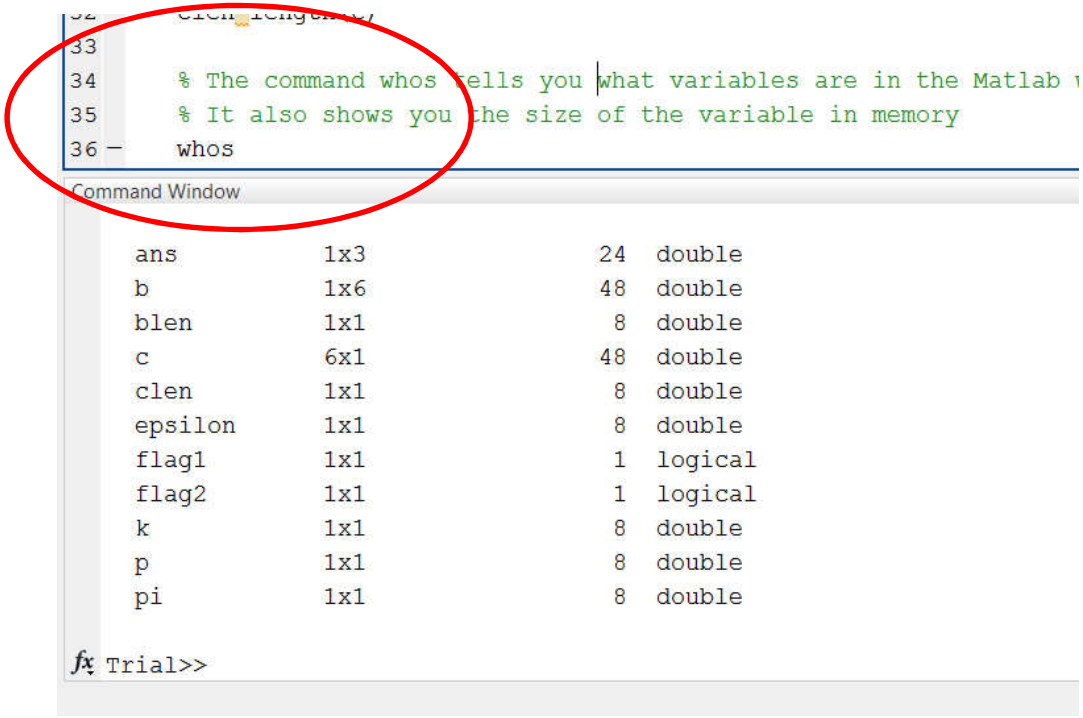


Step 15 – Using the “whos” Command

- To find out what variables are in the memory, the MATLAB workspace window gives some information.
- Type the following into the editor window:

  whos
- This command shows what variables are defined, their size, and what type of information they contain.
- Note, you can get just some variables using wildcards like:

  whos b*



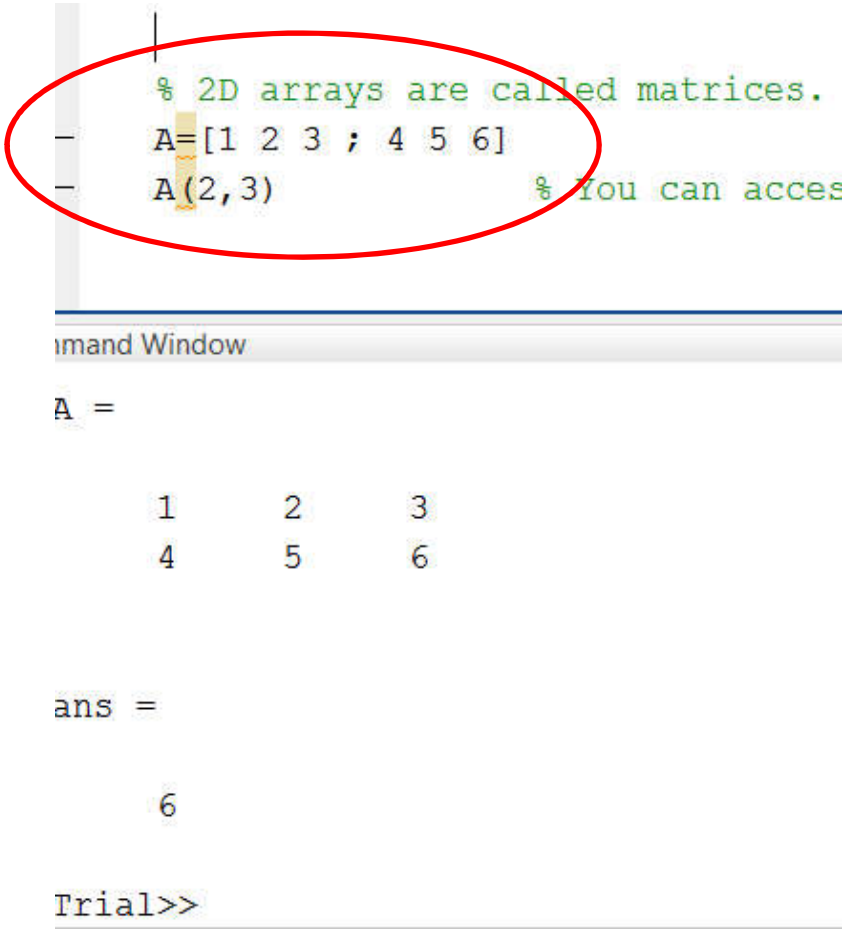
Step 16 – Making 2D Arrays (Matrices)

- 2D arrays are matrices.
- The matrix has two indices.
- The number of rows and number of columns define the size of the matrix.
- Type the following into the editor window:

  A=[1 2 3 ; 4 5 6]

  A(2,3)

- You can access individual element values from a matrix.



Step 17 – Accessing Multiple Matrix Elements

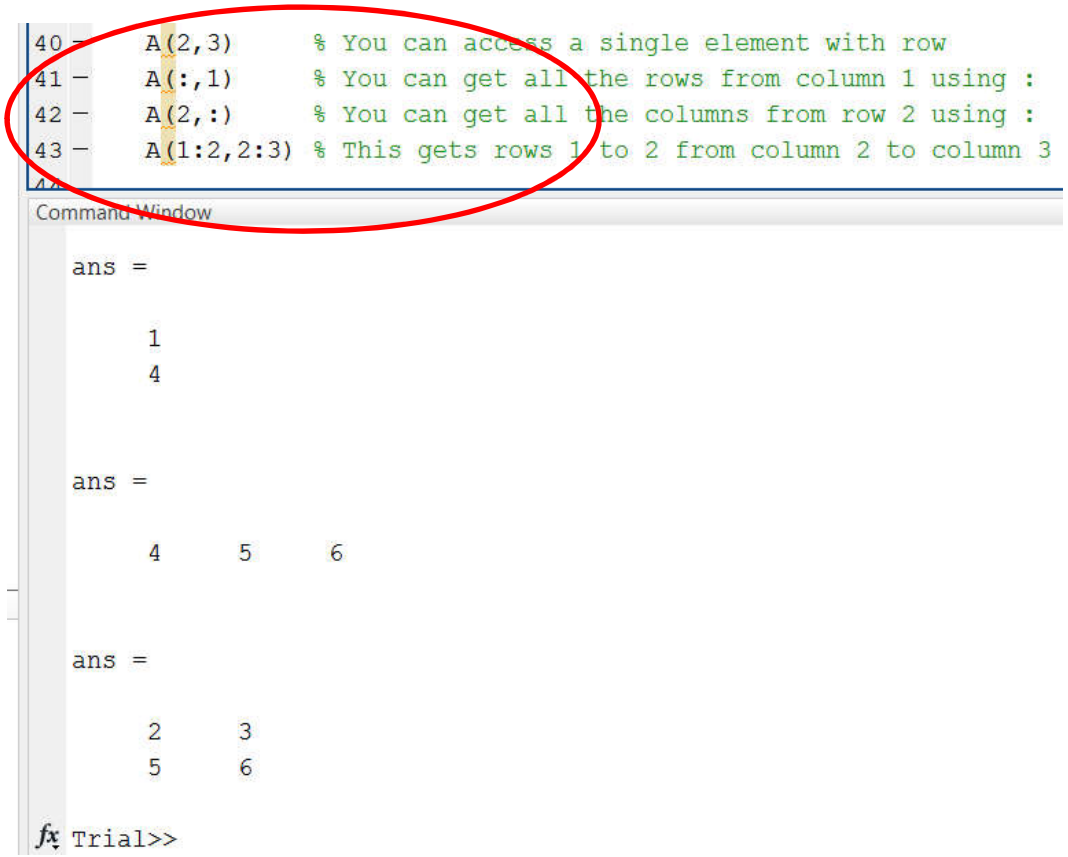
- The colon operator can be used to access multiple 2D array values.
- Type the following into the editor window:

  A(:,1)

  A(2,:)

  A(1:2,2:3)

- Just using a colon access all the rows or all the columns.
- You can use numbers with the colon to access parts of the matrix.



Step 18 – Using the “size” Function

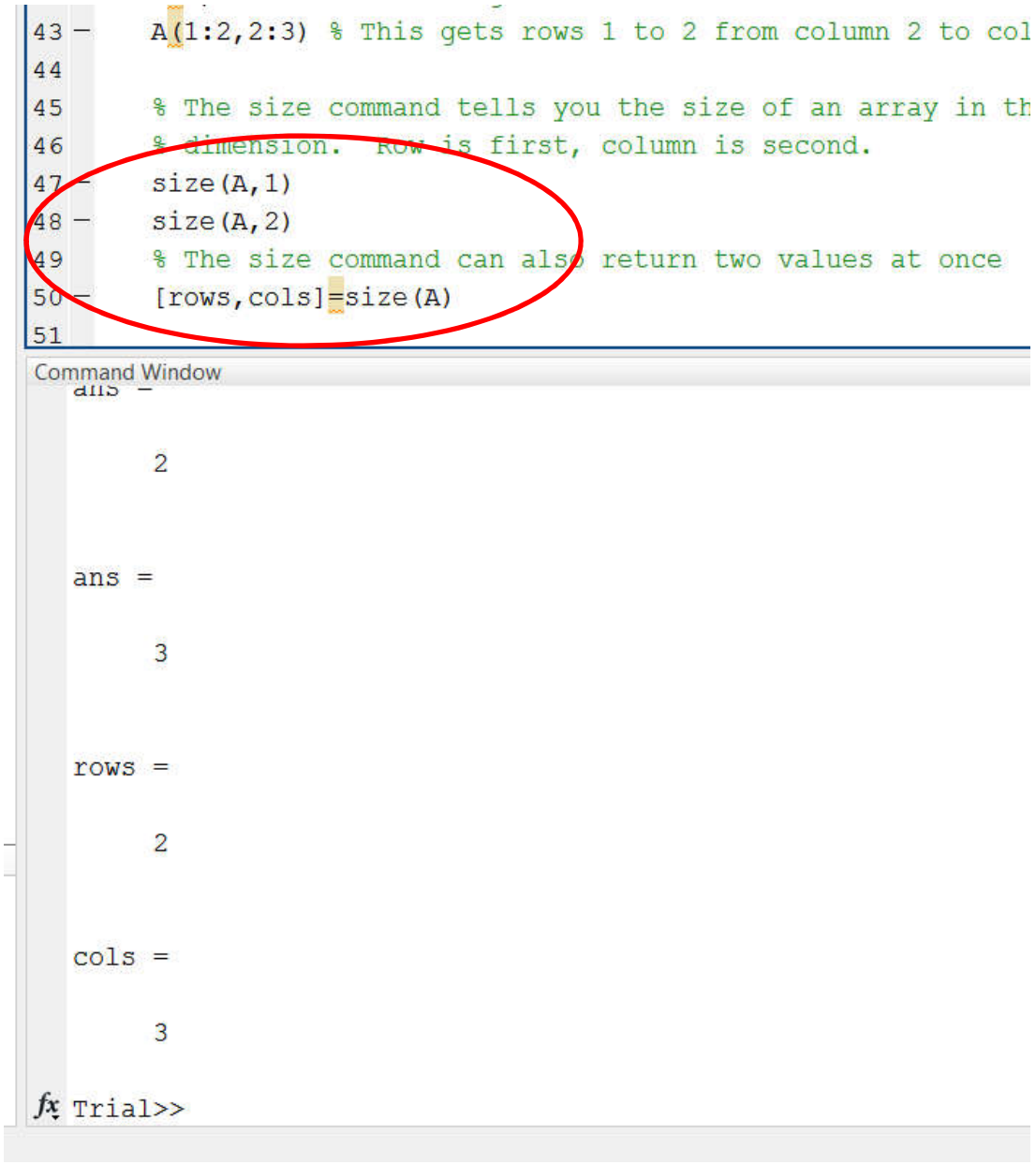
- Sometimes you do not know the size of a matrix.
- The size command can determine the size in one of the dimensions.
- Type the following into the editor window:

  size(A,1)

  size(A,2)

  [rows,cols]=size(A)

- The variables rows and cols will contain the size of matrix A.

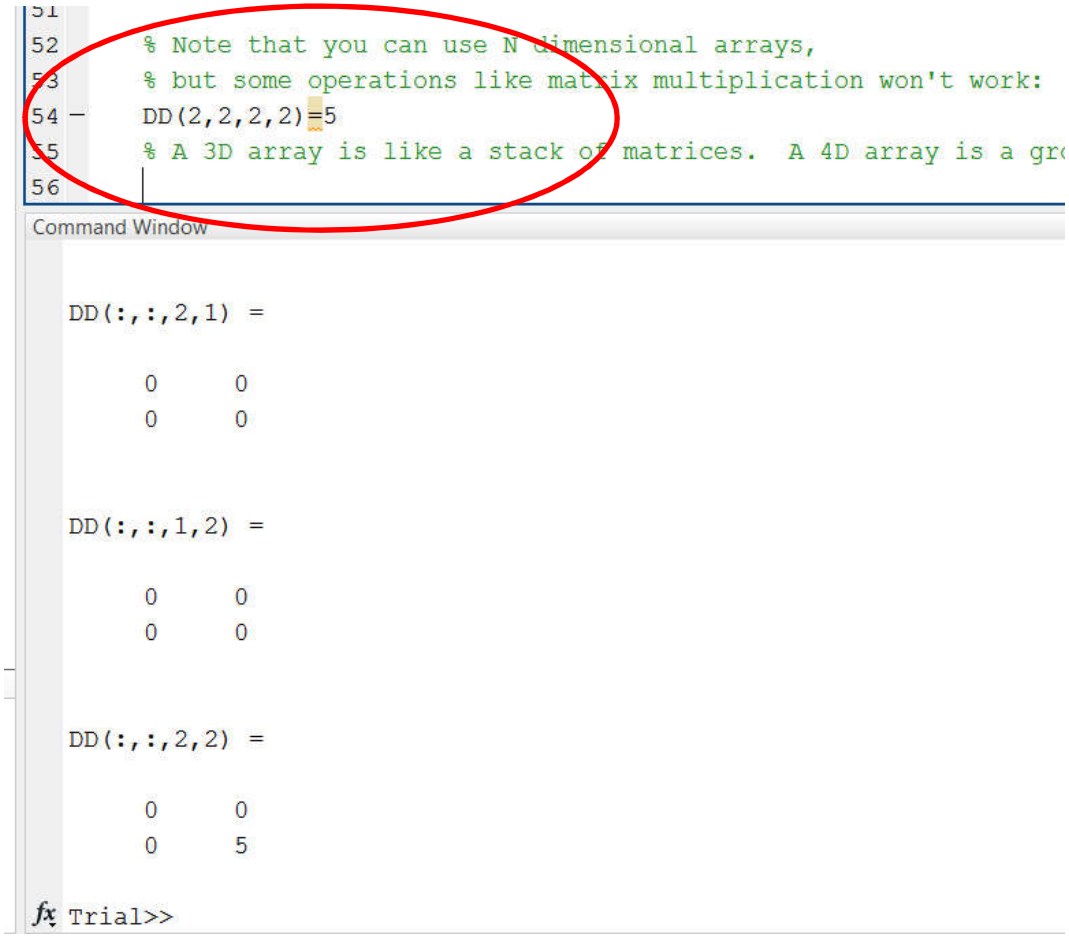


Step 19 – More Complex Arrays

- MATLAB can make arrays with more dimensions.
- A 3D array is like a stack of matrices.
- Type the following into the editor window:

  DD(2,2,2,2)=5

- This makes a four dimensional array, a list of a stack of 2D matrices with four indices.



Step 20 – Using Strings

- Strings are just arrays that contain characters instead of numbers.
- MATLAB uses single quotes (next to enter) to make a string array.
- Type the following into the editor window:

  name='bubba'

  city='Columbia'

  name(2:4)

```
59 –    clear D*
60      % The clear command without a variable name clears a
61
62      % Strings are really just a 1D array of single chara
63 –    name='bubba'
64 –    city='columbia'
65 –    name(2:4)
66
```

Command Window

```
name =

bubba


city =

columbia


ans =

ubb

Trial>>
```

Step 21 – Using the “ones” and “zeros” Functions

- The ones command makes a matrix full of ones.
- The zeros command makes a matrix full of zeros.
- Type the following into the editor window:

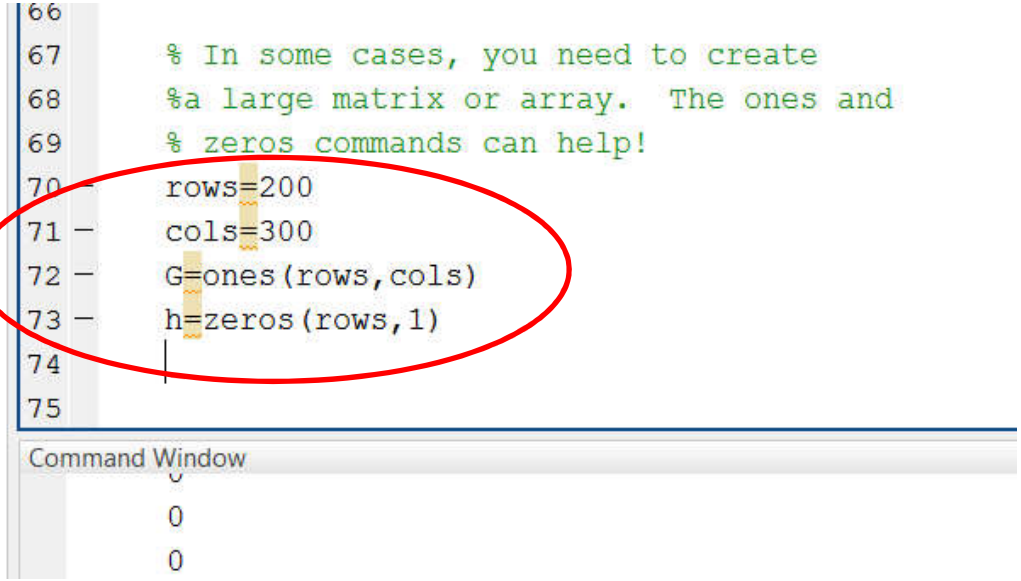
  rows=200

  cols=300

  G=ones(rows,cols)

  h=zeros(rows,1)

- Note: Adding a semicolon after a line suppresses command window output!



Step 22 – Using the “diary” Command

- The diary command writes stuff from the command window to a text file.
- Type the following into the editor window:

  !erase lab1diary.txt

  diary lab1diary.txt

  str=‘Lab 1 – George Burdell, Sect. 1’

  whos

  diary off

- Run your file.

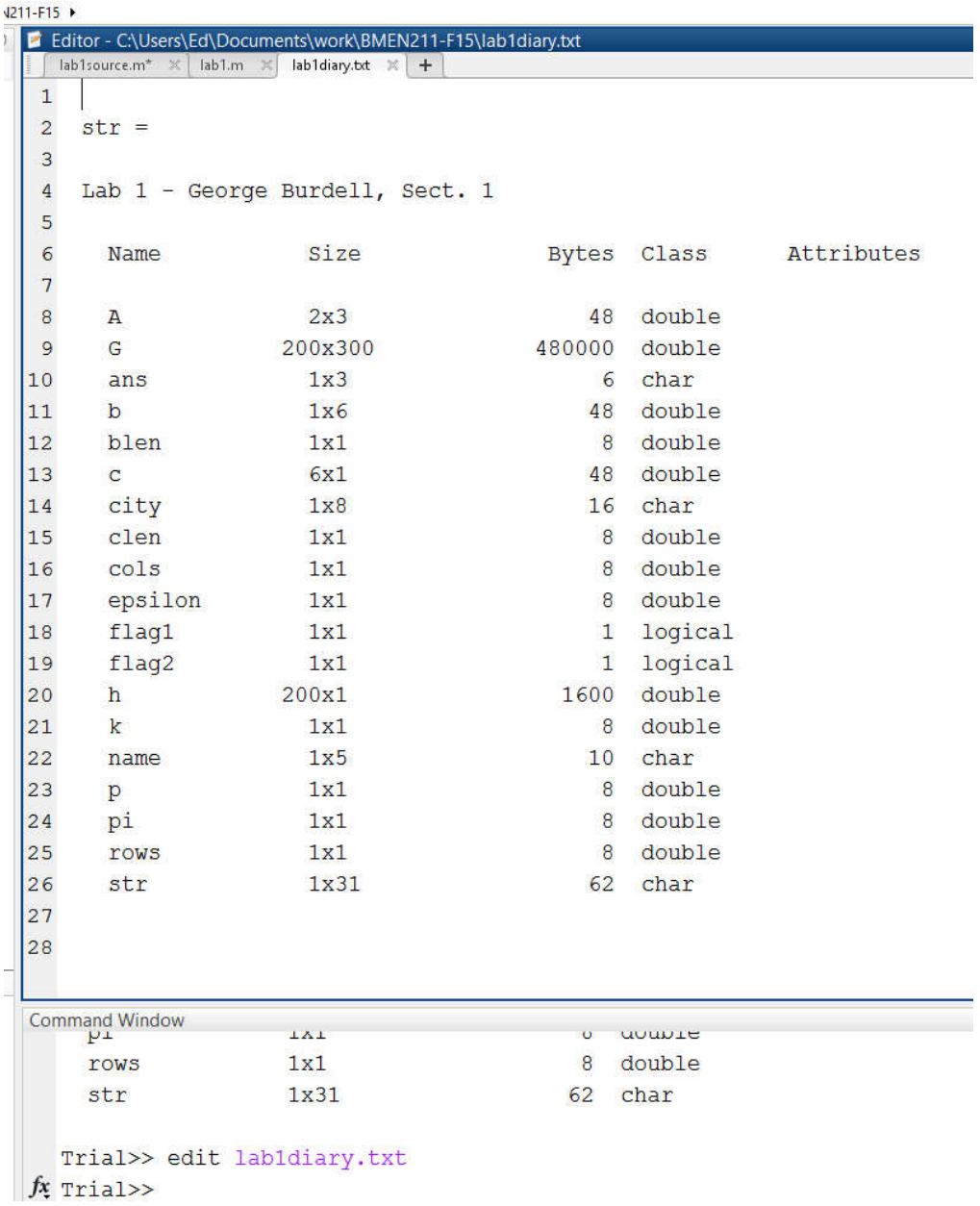
```
8 -    h=zeros(rows,1);
9
0      % The diary command can copy everything done in the editor to a text file
1 -    !erase lab1diary.txt                % This erases any old file
2 -    diary lab1diary.txt                 % This turns on the diary
3
4 -    str='Lab 1 - George Burdell, Sect. 1'% Change this line to your info!
5 -    whos                                % This lists all variables in memory
6 -    diary off   % This stops the diary
7
8
```

Step 23 – Print File

- In the command window, (not the editor window!) type the following

  edit lab1diary.txt

  and hit return.
- It should contain something similar to the image.
- Print out your diary file to turn in, showing that you completed this tutorial. DO NOT print out the .m file



Step 24 – Run Demos

- If you have extra time, go to the command prompt and type

  demos

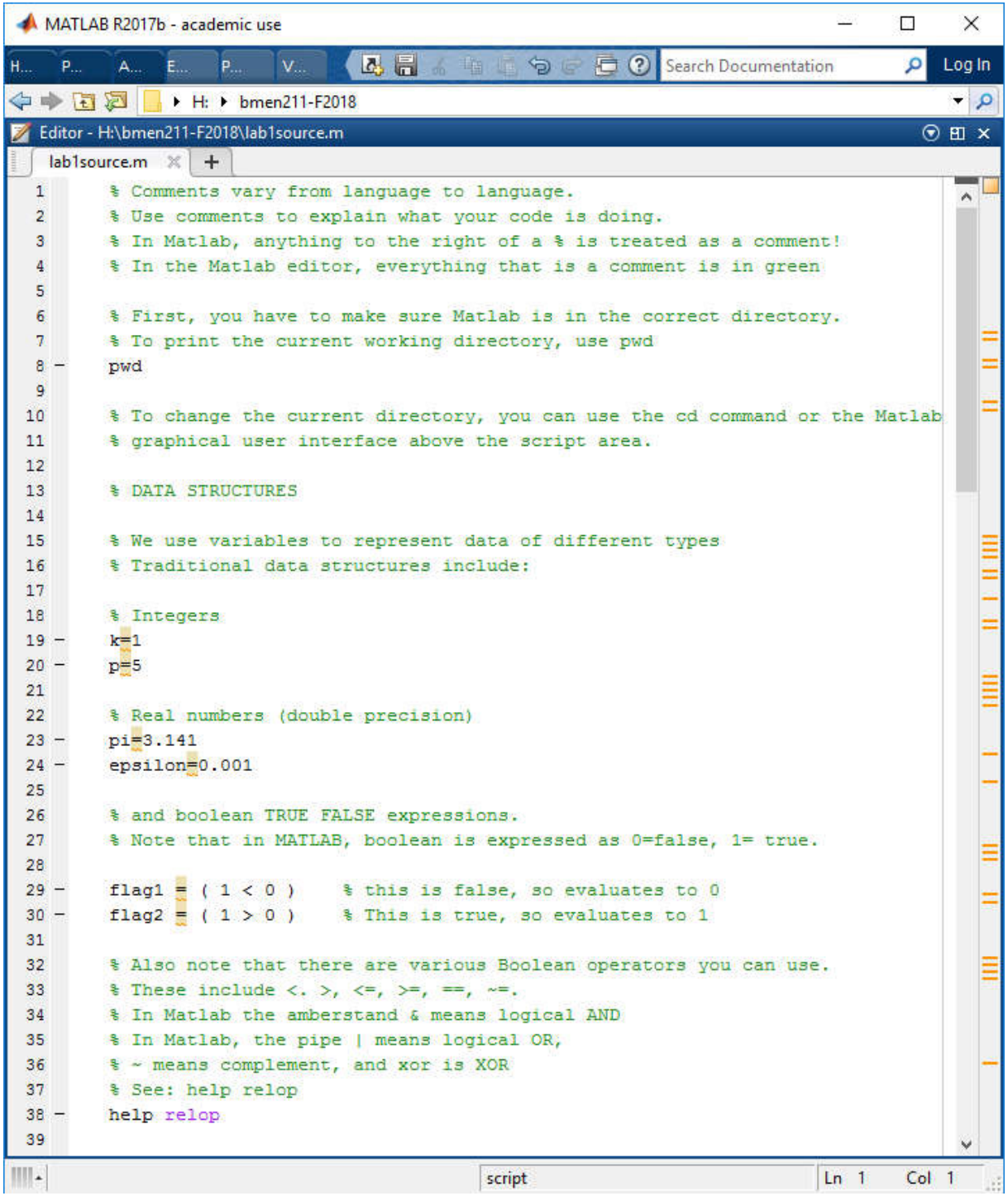
- In the command window and look through the various demos.
- You may want to check out the following:

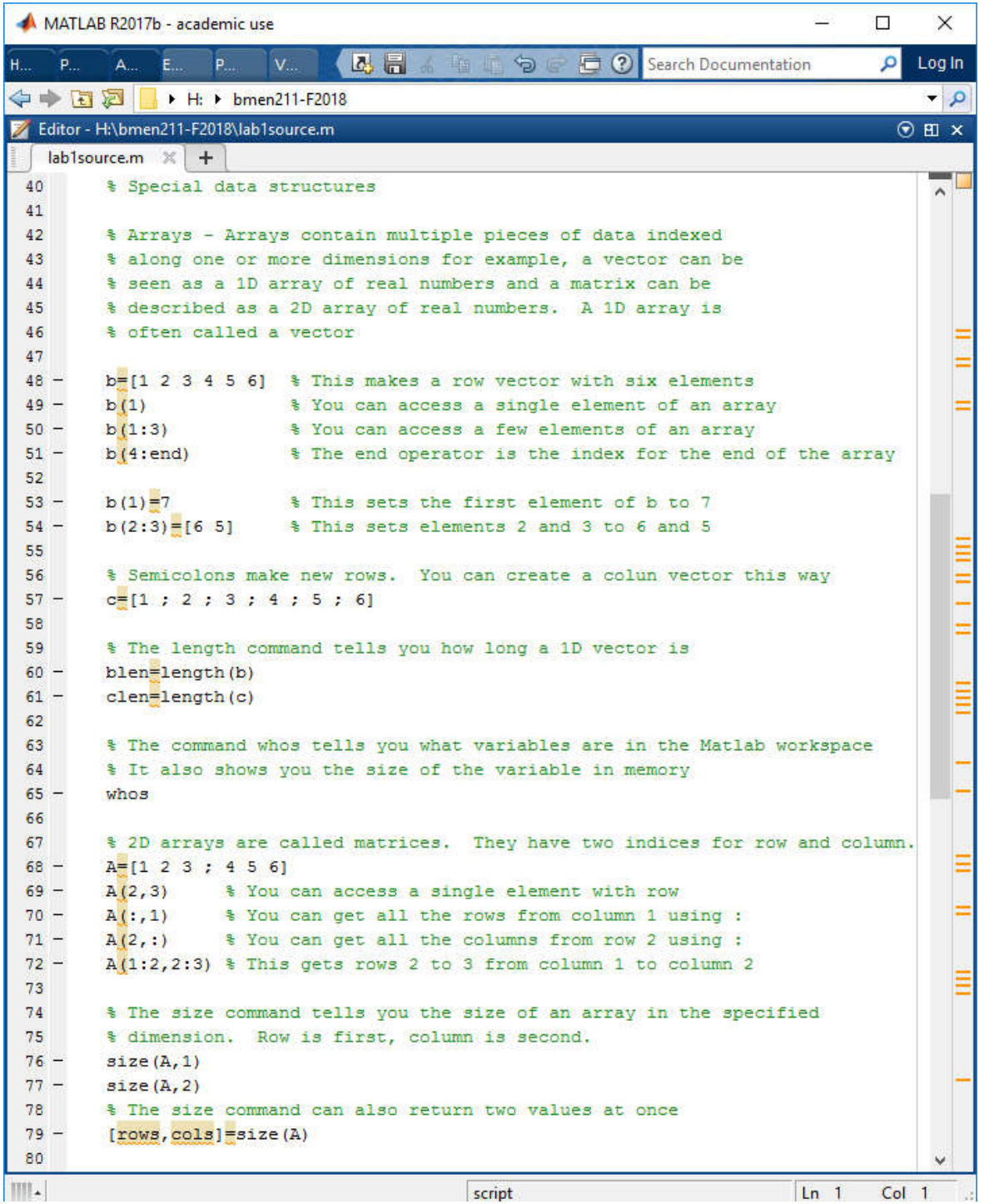
  Bioinformatics Toolbox

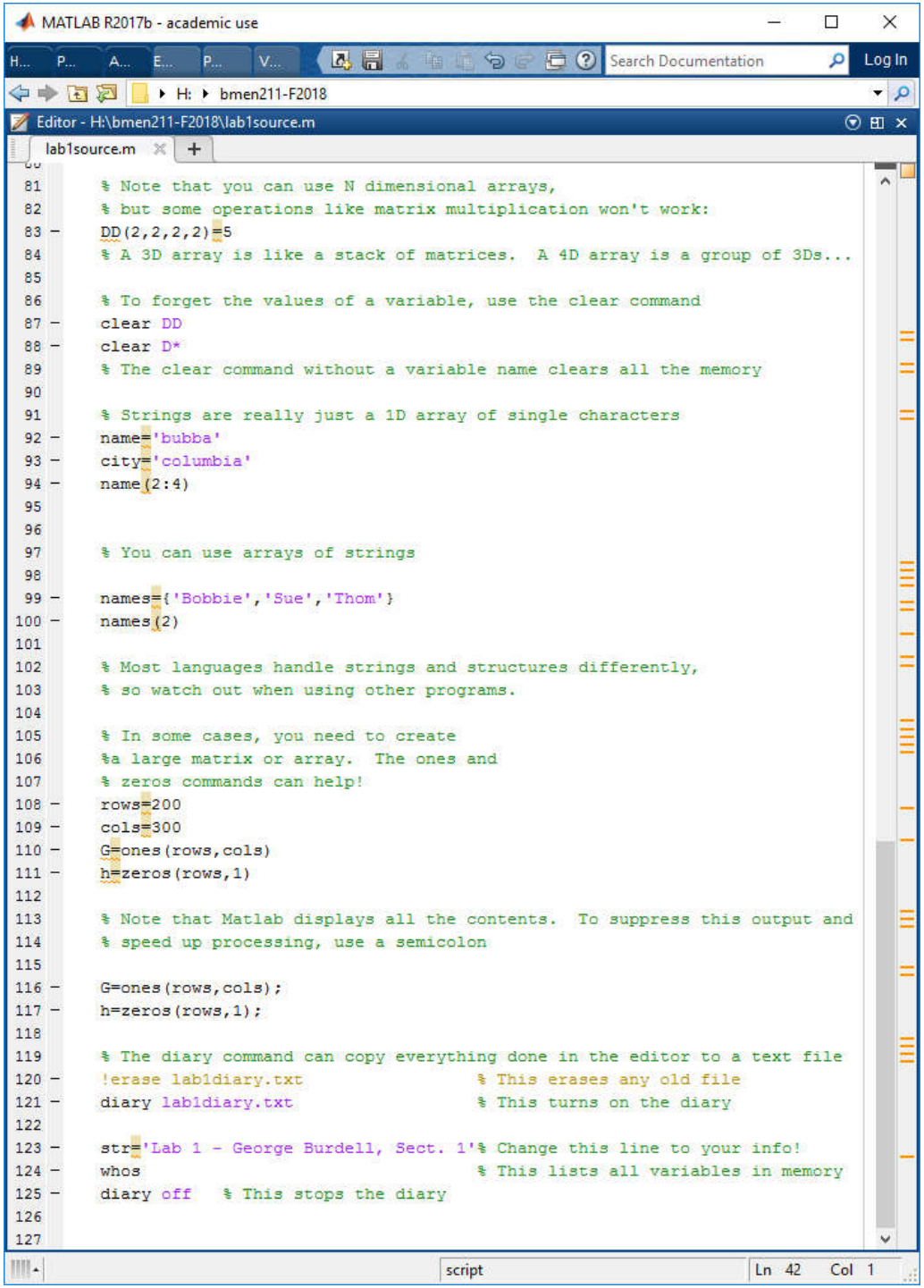
  Image Processing Toolbox

  SimBiology

  Neural Network Toolbox


MATLAB R2017b - academic use
Search Documentation
Log In
H: bmen211-F2018
Editor - H:\bmen211-F2018\lab1source.m
lab1source.m
% Comments vary from language to language.
% Use comments to explain what your code is doing.
% In Matlab, anything to the right of a % is treated as a comment!
% In the Matlab editor, everything that is a comment is in green
% First, you have to make sure Matlab is in the correct directory.
% To print the current working directory, use pwd
pwd
% To change the current directory, you can use the cd command or the Matlab
% graphical user interface above the script area.
% DATA STRUCTURES
% We use variables to represent data of different types
% Traditional data structures include:
% Integers
k=1
p=5
% Real numbers (double precision)
pi=3.141
epsilon=0.001
% and boolean TRUE FALSE expressions.
% Note that in MATLAB, boolean is expressed as 0=false, 1= true.
flag1 = (1 < 0) % this is false, so evaluates to 0
flag2 = (1 > 0) % This is true, so evaluates to 1
% Also note that there are various Boolean operators you can use.
% These include <. >, <=, >=, ==, ~=.
% In Matlab the amberstand & means logical AND
% In Matlab, the pipe | means logical OR,
% ~ means complement, and xor is XOR
% See: help relop
help relop
script
Ln 1
Col 1


MATLAB R2017b - academic use
Search Documentation
Log In
H: bmen211-F2018
Editor - H:\bmen211-F2018\lab1source.m
lab1source.m
% Special data structures
% Arrays - Arrays contain multiple pieces of data indexed
% along one or more dimensions for example, a vector can be
% seen as a 1D array of real numbers and a matrix can be
% described as a 2D array of real numbers. A 1D array is
% often called a vector
b=[1 2 3 4 5 6] % This makes a row vector with six elements
b(1) % You can access a single element of an array
b(1:3) % You can access a few elements of an array
b(4:end) % The end operator is the index for the end of the array
b(1)=7 % This sets the first element of b to 7
b(2:3)=[6 5] % This sets elements 2 and 3 to 6 and 5
% Semicolons make new rows. You can create a colun vector this way
c=[1 ; 2 ; 3 ; 4 ; 5 ; 6]
% The length command tells you how long a 1D vector is
blen=length(b)
clen=length(c)
% The command whos tells you what variables are in the Matlab workspace
% It also shows you the size of the variable in memory
whos
% 2D arrays are called matrices. They have two indices for row and column.
A=[1 2 3 ; 4 5 6]
A(2,3) % You can access a single element with row
A(:,1) % You can get all the rows from column 1 using :
A(2,:) % You can get all the columns from row 2 using :
A(1:2,2:3) % This gets rows 2 to 3 from column 1 to column 2
% The size command tells you the size of an array in the specified
% dimension. Row is first, column is second.
size(A,1)
size(A,2)
% The size command can also return two values at once
[rows,cols]=size(A)
script
Ln 1
Col 1


MATLAB R2017b - academic use
Search Documentation
Log In
H: ▸ bmen211-F2018
Editor - H:\bmen211-F2018\lab1source.m
lab1source.m
81 % Note that you can use N dimensional arrays,
82 % but some operations like matrix multiplication won't work:
83 DD(2,2,2,2)=5
84 % A 3D array is like a stack of matrices. A 4D array is a group of 3Ds...
85
86 % To forget the values of a variable, use the clear command
87 clear DD
88 clear D*
89 % The clear command without a variable name clears all the memory
90
91 % Strings are really just a 1D array of single characters
92 name='bubba'
93 city='columbia'
94 name(2:4)
95
96
97 % You can use arrays of strings
98
99 names={'Bobbie','Sue','Thom'}
100 names(2)
101
102 % Most languages handle strings and structures differently,
103 % so watch out when using other programs.
104
105 % In some cases, you need to create
106 %a large matrix or array. The ones and
107 % zeros commands can help!
108 rows=200
109 cols=300
110 G=ones(rows,cols)
111 h=zeros(rows,1)
112
113 % Note that Matlab displays all the contents. To suppress this output and
114 % speed up processing, use a semicolon
115
116 G=ones(rows,cols);
117 h=zeros(rows,1);
118
119 % The diary command can copy everything done in the editor to a text file
120 !erase lab1diary.txt % This erases any old file
121 diary lab1diary.txt % This turns on the diary
122
123 str='Lab 1 - George Burdell, Sect. 1'% Change this line to your info!
124 whos % This lists all variables in memory
125 diary off % This stops the diary
126
127
script
Ln 42 Col 1