Exercise # 1 – The Matlab Environment

Tutor in charge of this HW: Ayelet Sarel. E-mail your solutions to: [matlab.wis@gmail.com](mailto:matlab.wis@gmail.com)

The mail subject should be: 'matlab intro 2021B exercise #1'. Attach only the .m script file to your mail (submit all questions in the same script).

# Submit exercises only in groups of two or three students.

NOTE: All the following instructions should be implemented in your script except of sub-questions where it is indicated that you should use the command line.

# Question 1 – working environment

1. Read carefully the submission guidelines that appear in the homepage of the course website: (<http://www.weizmann.ac.il/midrasha/courses/MatlabIntro/>). Login instructions including username and password will be published in the forum welcome message soon: (<https://groups.google.com/g/matlabwis2021a>)
2. Create a new folder called C:\matlab\_course\HW1 (you can do it outside of MATLAB).
3. Create a new script named “ex1\_<studentID1>\_<studentID2>.m” and save it in the directory C:\matlab\_course\HW1.
4. Your script should display on the screen (in the Command Window) the name of the current directory using the "disp" and "pwd" functions (note: you should use a semicolon at the end of each code line).
5. Try running the script from the command line by simply writing its name. Did the script manage to run? Did you get an error? Why?
6. Include the directory of your script in the Matlab path (go to Home → Environment → Set Path → Add Folder).

Did the script manage to run now?

1. Remove the directory of your script in the Matlab path using the command "rmpath". Try running the script from the command line again. Did you get an error?
2. Change the current directory to C:\matlab\_course\HW1 (using the command ‘cd’ in the command line) and run the script again. Did you get an error now? Why?

From now on always check what is your current directory in Matlab – it is important for saving and loading variables and creating the desired working environment as you will see next.

# Question 2 - variables handling

1. Create a variable called *Resistance* that has the value 5. You should specify the units of the variable in a comment (ohm).
2. Create a variable called *Current* that has the value 3 (amp);
3. Calculate the value of the variable *Voltage* (*Voltage = Resistance \* Current*)
4. Using "disp" or "fprintf" (google it) - display your result in the following format: “The voltage is: <your result>”.
5. Use the function whos in the command line (not in your script). What is the variable type of

*Current*? (Write the answer as a comment in your script).

1. Display all variables in the workspace using the function who.
2. Save the variable *Voltage* (using the function save) in file *save\_voltage.mat*.
3. Delete the variable *Voltage* from the workspace using the function clear (in your command line).
4. Now display all the variables in the workspace using the function who. How many variables are in the working space? (Write the answer as a comment in your script).
5. Load the saved variable *Voltage* from your file *save\_voltage.mat* (in your command line). How many variables exist in the Workspace now? (Write the answer as a comment in your script)
6. Use clear in the command line. How many variables exist in the Workspace? Write the answer as a comment in your script).
7. Copy & paste the following line into your command line:

*Current =7; Resistance =5;*

How many variables exist in the Workspace now? (Write the answer as a comment in your script)

1. Re-run your script (using the run button ). What is the *Voltage* now? Why didn’t it change to 35? What should be done to get the result 7\*5=35? (Write the answer as a comment in your script)
2. Insert the command “clear” to your code between the creation of the *Resistance* and *Current* variables, and the *Voltage* calculation. Now re-run your script. Why did you get an error? (Write the answer as a comment in your script).
   * Delete the “clear” command from your script.

*In general do not use functions like “clear”, “clc”, “close” in the middle of your submitted scripts. You might clear your outputs and those won’t be checked…*

1. Why in your opinion is it recommended to insert “clear”/“clc” commands at the beginning of the script? **Describe a simple scenario** in which the use of the function “clear” in the beginning of a script will prevent a run time error. (Write the answer as a comment in your script)

*From now on, please write ‘clear’ in the beginning of your all your assignments.*

# Question 3: debugging and fixing errors

1. Copy & paste the following code to your script:

% this code multiplies two numbers and displays the result: num1=10;

num2=13; num2=4;

disp(['If we multiply ',...

num2str(num), ' and ', num2str(num2), ' we get: '... num2str(num1\*num2)]));

1. The code has (i) a **syntax** error, (ii) a **run time** error and (iii) a **warning**. MATLAB marks the detected syntax errors in red, and the detected warnings in orange. Fix only the syntax error, make sure there are no additional red marks and run your script. Did it run without an error?
2. Hover with the cursor over “num2”, what is the Matlab warning? What is the difference between syntax errors and warnings in Matlab? Fix the code to eliminate the warning (only the warning). Make sure the result stays the same. Re-run your script.
3. You should still get an error (a **run time error)**. In the error message you see the reason and get a hint of where the script is wrong. What was the problem? Try to fix the code and make sure the script runs without errors.
4. What is the difference between a **syntax** error and a **run time** error? In your answer think about how we can detect these two different types of errors.