```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%  MATLAB Brush Up Course: Session 1  %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%   by JOAN MARGALEF   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%  INTRODUCTION TO MATLAB: ENVIRONMENT & MATRICES

% This session is designed to introduce fundamental concepts
% in MATLAB. It covers the basics of MATLAB desktop elements, including the
% workspace, command window, and editor. The practical exercises focus on
% basic mathematical operations, string concatenation, script creation, and
% advanced matrix operations. Through these problems, learners will gain a
% solid understanding of MATLAB’s capabilities in numerical computation,
% matrix manipulation, and script writing.


%% 0. Matlab Desktop: Explained

% Current Folder: working directory

% Workspace: keep track of the variables

% Command Window: execute commands
    % 1. Command prompt (>>): is the beginning of the command line.
    % 2. To execute a command by pressing ENTER.
    % 3. variable = expression: assigns to a variable.

% Editor: to edit a "Script"


% Let's close the editor to understand the other elements...

% RUN:
2 + 3  % The most recent statement that was not assigned stored ans.
a = 50 % Assign a name to the var: no need to define the variable type
A = 55 % Case Sensitive
a = 100 % Previous Value get's replaced
b = a^2; % Create new variables from other ones
c = a + b; % Semicolon suppresses output
disp(c)  % Display value, also can be seen by clicking in Workspace

message='I use MATLAB!'; % String var between single quotes
disp(message)


%% Practice 0

% 1. Assigning Numerical Values
% a. Create a variable 'x' and assign the value 10 to it.
% b. Create another variable 'y' and assign the value 20 to it.

% 2. Performing Mathematical Operations
% a. Create a variable 'sum' that stores the sum of 'x' and 'y'.
% b. Create a variable 'product' that stores the product of 'x' and 'y'.
% c. Display the results of bothusing the disp() function.


% 3. LEARN HOW TO CONCATENATE STRING VARs. Create two string variables
% 'name' and 'surname' (using yours) and execute
% full_name = [name surname];

% 4. Display the content of `full_name` using the disp() function.
```

```
%% 1. What is a Script? Some Algebra

% Command Window: Evaluates single expressions.
% Editor: Script stores a sequence of instructions in M-files (*.m).
% Running a script: MATLAB executes commands in the M-file.


% IMPORTANT:
% Before running it might be required to save the script
% Name of the script can't have spaces and weird symbols
% Using double comment (%%) it creates sections that can be run separetely


% USEFUL COMMANDS
clear; % Clear WorkSpace
clc; % Clear Command Window

% It is common practice to remove everything from the workspace and
% command window before running a new script.
% This is done to avoid potentially conflicting variable names and
% function definitions




% Other ones
pwd % Print the current directory.
ls % List all of the files in the current directory.
lookfor disp % Search help for keywords.
help disp  % Provide information of a function


%% Practice 1

% 1. Create a script and save a file "yourname.m" in the working directory.

% 2. Type three commented lines: your name, current date, and a short
%    description of the file.

% 3. Put 'clear' and 'clc' on top.

% 4. Create a Section.
%  - Calculate the square root of 16 using sqrt()
%  - Compute the exponential of 3 using exp()
%  - Calculates the natural logarithm of 10 log()
%  - Round value of 2.65 using round()

% 5. Create another section
%  - Execute in command window "pi" (predeterminated assigned)
%  - Calculate sinus of "2*pi" using sin()
%  - Calculate cosinus of "2/3*pi" using cos()

% 6. Run sections separetely and together
```

```matlab
%% 2. Matrices

clc;
clear;

% Creating a Row Vector
r = [1 2 3];
r = [1, 2, 3];


% Creating a Column Vector
c = [1; 2; 3];

% Transpose
r
r'
c
c'

% Creating Vector as a Sequence
seq=[0:0.5:5]

linspace(0,5) % Row vector of 100 linearly equally spaced
linspace(0,10,3) % Row vector of n linearly equally spaced

% Creating a Matrix
M = [1 2 3; 4 5 6; 7 8 9];

% Zeros, Ones, and Identity Matrices
 z= zeros(1, 2) % Creates an n-by-m matrix of zeros.
 o= ones(2, 3) % Creates an n-by-m matrix of ones.
 e= eye(4,5)     % Creates an n-by-n identity matrix.
 nanmat= NaN(3,3) % NaN stands for Not a Number (0/0 or No data)



% Manipulating Matrices and Vectors

% Create 2x2 matrices A and B
A = [1 1; 1 1]
B = [2 1; 1 1]


% Transpose of A and B
A'
B'

% Inverse
A^(-1)
inv(A)

% Matrix Addition
A + B

% Matrix Subtraction
A - B

% Matrix Multiplication
A * B % Matrix product of A and B
A .* B % Element-wise multiplication of A and B


% Concatenation
[A B] % Horizontal concatenation of A and B
[A; B] % Vertical concatenation of A and B
```

```
% Concatenation by repetition
repmat(B,1,2) % 1 Vertical 2 Hor
repmat(B,2,1) % 2 Vertical 1 Hor
repmat(B,2,2) % 2 Vertical 2 Hor


A(1, 1) % Access first element of A
B(2, :) % Access second row of B  (":" means ALL)
A(:, 1) % Access first column of A
diag(A) % Returns diagonal as a vector
A(1,1)=5
A
a=B(2,2)

C=eye(5)
C(5,3:end)  % Access elemnets of row 5, columns 3 till end
C(1,[1,3])  % Access first row, first and third element


%% Practice 2

% 1. Create a vector from 1 to 100 in steps of 2.

% 2. Create a 3x2 matrix with first row all being 1, second 2,
% and third 3

% 3. Create a 4x5 matrix 'A' with all elements being 7.

% 4. Take the first (top-left) 3x3 Matrix from 'A'and rename it
% as 'A'.

% 5. Create a 3x3 identity matrix 'B'. Give its inverse.

% 6. Concatenate matrix 'A' and 'B' horizontally

% 7. Sum, subtract, multiply, and multiple elementwise A and B.
```

```matlab
%%  3. Matrix Properties


b=[1 4 1];

% Sorting Elements
sort(b)

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

% Size of Matrix B
size(B) % Returns the size of matrix B as a two-element row vector.
size(B,1) % Returns the number of rows in matrix B (First Dimension).
size(B,2) % Returns the number of columns in matrix B (Second Dimension).

% Reshaping Matrix B into a 1x9 matrix
reshape(B, 1, 9) % Reshapes matrix B into a 1x9 matrix.

% Summing Elements
sum(b)
sum(B)  %Sum by Column
sum(B,1)  %Sum by Column
sum(B, 2) %Sum by Row

% Cumulative Sum
cumsum(B, 1) % Returns the cumulative sum of elements along column
cumsum(B, 2) % Returns the cumulative sum of elements along rows


% Minimum and Maximum Values
min(B) % Returns a row vector containing the minimum element of each column
max(B) % Returns a row vector containing the maximum element of each column



% Determinant
det(B)

% Trace (sum of diagonal elements).
trace(B)

% Finds the eigenvalues
eig(B)

% Rank
rank(B)

% Extracts the lower triangular
tril(B)

% Extracts the upper triangular
triu(B) %


%% Practice 3

% 1. Create a 'X' as [4 7; 2 6], calculate its determinant.

% 2. Create a 'Y' as [3 1 4; 1 5 9; 2 6 5]. Find eigenvalues
%    and the trace.

% 3. Using matrix 'B' ([1 2 3; 6 5 4; 8 7 9]), find the maximum value in
%    each row.

% 4. Create a 4x4 matrix 'Z' as [10 9 8 7; 6 5 4 3; 2 1 0 -1; -2 -3 -4 -5],
%    extract its upper triangular part, and then find the rank.
```