



MATLAB Notes

@Soumil Gupta

▼ Basic Commands →

- type `clc` to clear the command window.
- type `clearvars` to clear all the workspace variables.
- type `whos` to see the detailed list of all variables right on the command window. notably, it shows the type of the variable also. alternatively, double click on the variable on the workspace to get more information about that variable (helpful with matrices)
- type `var = 'hello'` to create a string in MATLAB. the data type of this `var` is `char`.
- use `,` to separate statements. e.g.: `x=1, y=2` .
- add a `;` at the end of your command to suppress the output.
- type `var = "hello"` to create a string in MATLAB. the data type of this `var` is `string`.
- press the `up arrow` on your keyboard to see the list of all the previous commands you've run.

▼ Create a new script →

- A script is different from the command window. It is a like coding a C file.
- Go to `Home → New → Script`.
- To run the file, hit `Save → Run`
- Keep in mind that if you don't want a variable to be printed when the program is run, then use the `;` operator at the end of the line.
- Hit `Ctrl + Enter` to run the program.
- Use `%` to add comments.
- Always start your script with `clc, clearvars, close all`

▼ Arrays and Matrices →

- Vectors and Arrays are terms which can be used interchangeably in MATLAB.
- `x = 1: 10` creates a horizontal array of size 10 (1 row by 10 columns) with elements from 1 to 10.
- `x = 1:1:10` creates a horizontal array with elements starting from 1, with a jump value of 1 and ending at 10.
- To change the dimension of the array (i.e. transpose), use the operator `'` e.g.: `x'`
- `linspace`: To generate a horizontal array with numbers evenly spaced, use `linspace` command. `linspace(start number, end number, number of elements)`. By default, the number of elements generated is 100.
- Manually assign an array: `x = [10 19 -1 0]`. The space serves the purpose of the comma. You can also use a comma `,`
- Make a 2D Array / Matrix: `A = [1 , 2 ; 3, 4]`. The `;` marks the start of a new row.
- All Linear Algebra Rules are followed by MATLAB. e.g, `A+2` adds 2 to all the elements of the matrix *A*.
- **Element Wise Operation**: What if you want to run operations not on the matrix itself but on the elements of the matrix, for example, you want to square all the elements of a matrix A: type `A.^2`. The `.` operator refers to the element-wise operations.

- Make Zero Matrix: `0 = zeros(3)`. For custom size: `0 = zeros(3,1)`, this gives a 3 row by 1 column matrix with all entries equal to 0.
- Make All elements of matrix 1: `I = ones(3)`. For custom size: `I = ones(3,1)`, this gives a 3 row by 1 column matrix with all entries equal to 1.
- Make Identity Matrix: `I = eye(3)`
- Access elements of a matrix the same way you did in C, but use `()` instead of `[]`. e.g, `A(2,3)`. For a 1D array you just to have to specify one index. You are free to change the value at any time by accessing the element.
- **IMPORTANT:** MATLAB uses 1—based indexing instead of the traditional 0—based indexing in other languages.
- `A(end)` gives you the last element of the array A
- `A(end-10)` gives you the 10th element from the end of the array A .
- Extract entire row/column from a matrix: To extract entire row, e.g: `A(2, :)`. To extract entire column, e.g: `A(:,2)`.
- `A(2,1:2)` gives all the element of second row from first to second column.

▼ Math Functions →

- click on the f_x option to get the list of all math functions available on MATLAB
- type `help function_name` to get all the required parameters for the function.
- use `plot(x, y)` to plot the graph. the first parameter will be on the x-axis and the second parameter will be on the y-axis.

```
x = linspace(0,5);
y=(-(x-3).^2)+10;
plot(x,y);
max(y) %gives the maximum value of y
min(y) %gives the minimum value of y
```

- type `doc function_name` to open the documentation of the function.

