Script File

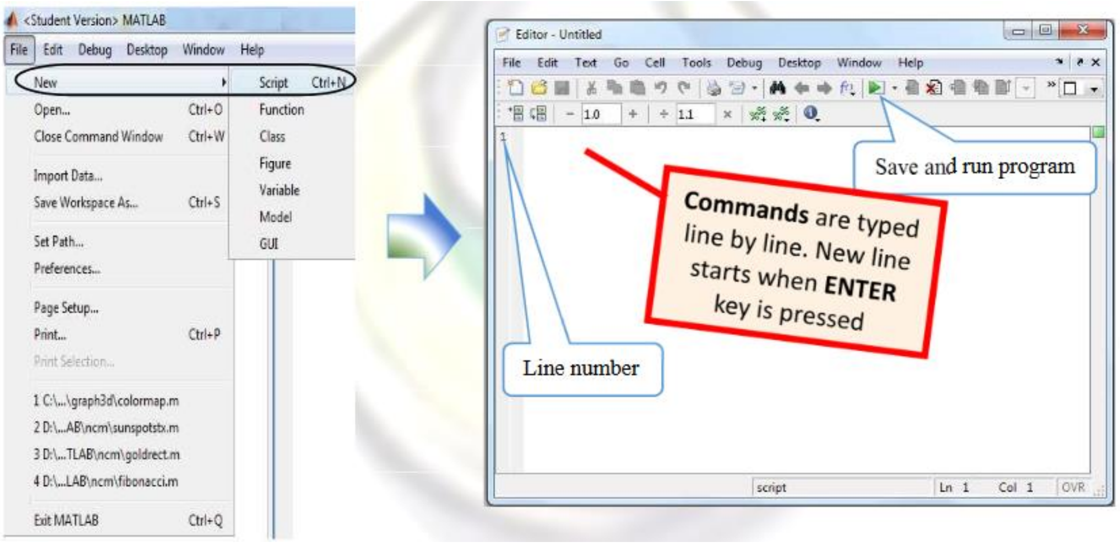
For simple problems, entering commands at the MATLAB prompt in the Command window is simple and efficient. However, when the number of commands increases, or you want to change the value of one or more variables, reevaluate a number of commands, typing at the MATLAB becomes tedious. You will find that for most uses of MATLAB, you will want to prepare a script, which is a sequence of commands written to a file. Then, by simply typing the script file name at a MATLAB prompt, each command in the script file is executed as if it were entered at the prompt.

Script File: Group of MATLAB commands placed in a text file with a text editor. MATLAB can open and execute the commands exactly as if they were entered at the MATLAB prompt. The term "script" indicates that MATLAB reads from the "script" found in the file. Also called "M-files," as the filenames must end with the extension '.m', e.g. example1.m.

M-files Scripts: are text files and may be created and modified with any text editor. The steps to create a script are:

1)Click on icon on the MATLAB toolbar.
2)Press keys (Ctrl + N)
3)Form (File → New → Script)

- A new window will activate called the Editor as shown.


<Student Version> MATLAB
File Edit Debug Desktop Window Help
New
Open... Ctrl+O
Close Command Window Ctrl+W
Import Data...
Save Workspace As... Ctrl+S
Set Path...
Preferences...
Page Setup...
Print... Ctrl+P
Print Selection...
1 C:\...\graph3d\colormap.m
2 D:\...AB\ncm\sunspotstx.m
3 D:\...TLAB\ncm\goldrect.m
4 D:\...LAB\ncm\fibonacci.m
Exit MATLAB Ctrl+Q
Script Ctrl+N
Function
Class
Figure
Variable
Model
GUI
Editor - Untitled
File Edit Text Go Cell Tools Debug Desktop Window Help
1.0
1.1
1
script
Ln 1
Col 1
OVR

Save and run program

Commands are typed line by line. New line starts when **ENTER** key is pressed

Line number

- When finished, save the file using File → Save or click on icon .

  The rules for filenames are the same as for variables (they must start with a letter, after that there can be letters, digits, or the underscore, etc.). By default, scripts will be saved in the **Work Directory**. If you want to save the file in a different directory, the Current Directory can be changed.

Example: write a program (vector.m) to generate a vector with 10 elements and find:
a.The largest element and its position.
b.The smallest element and its position.

Solve:

```
clc
clear
V=[7 2 1 3 5 69 3 4 1 15]
[Vmax,Pmax]=max(V)
[Vmin,Pmin]=min(V)
```

```
>>vector
V =
    7     2     1     3     5    69     3     4     1    15
Vmax =
   69
Pmax =
    6
Vmin =
    1
Pmin =
    3
```

❖ Input and Output Statements:

The script would be much more useful if it were more general; for example, if the value of the radius could be read from an external source rather than being assigned in the script. Also, it would be better to have the script print the output in a nice, informative way. Statements that accomplish these tasks are called input/output statements, or I/O for short. With examples of input and output statements will be shown here from the Command Window, these statements will make the most sense in scripts.

```
>> r = input ('Enter the radius: ')
Enter the radius: 7
r = 7
```

If character or string input is desired, ‘s’ must be added after the prompt:

```
 >> name = input ('Enter your Name: ', 's' )
Enter your Name: Ahmed name = Ahmed
```

MATLAB gave an error message and repeated the prompt. However, if the input function is used to enter number but the user instead enters a letter or vice versa

```
>> n = input ('Enter your Age: ')
```

```
 Enter your Age: k
??? Error using ==> input
 Undefined function or variable 'k'.
```

❖ Output Statements(disp and fprintf) function:

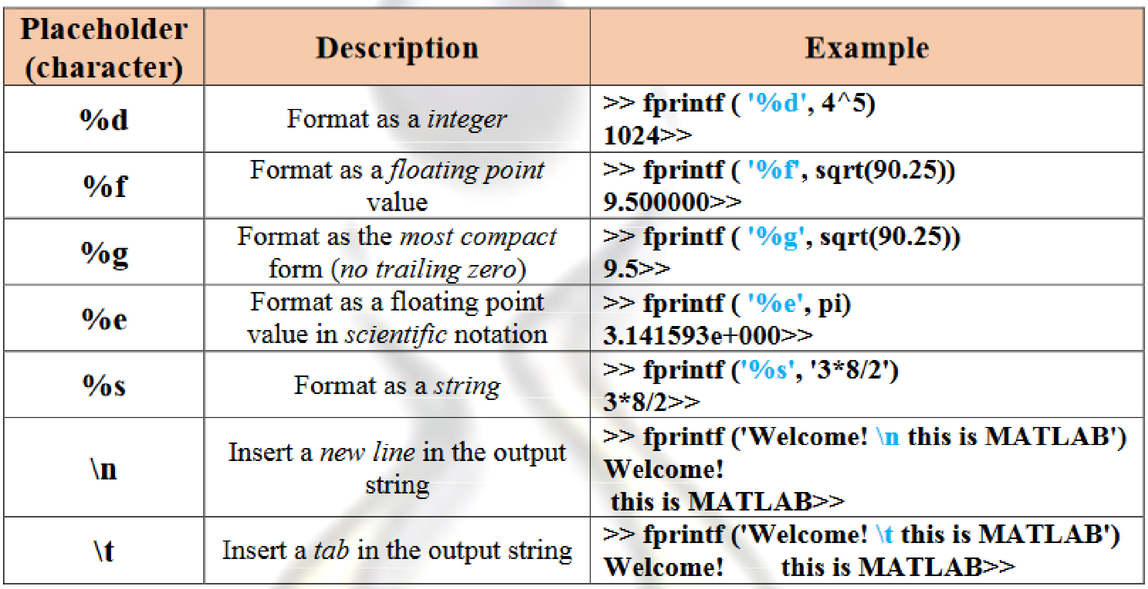
The simplest output function in MATLAB is disp, which is used to display the result of an expression or a string without assigning any value to the default variable ans. However, disp does not allow formatting. For examples:

```
>> disp ('Hello')
Hello
>> disp (6^4)
1296
>> disp ([2:8])
2 3 4 5 6 7 8
>> disp ([1:5 ; 5:5:25])
1  2  3   4    5
5 10 15 20 25
```

Formatted output can be printed to the screen using the fprintf function. For example:

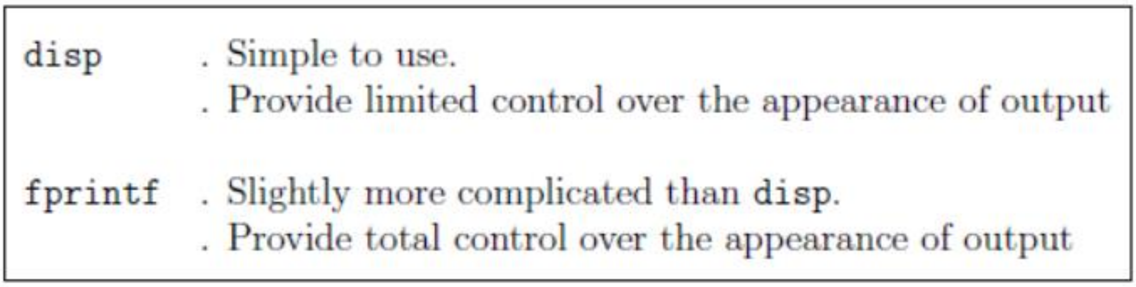
```
>> fprintf ('The 7! value is %d\n' , factorial(7))
The 7! value is 5040
>> fprintf ('Hello \n\n This is MATLAB \n')
Hello

This is MATLAB
```

The fprintf function, first a string (called the format string) is passed, which contains any text to be printed as well as formatting information for the expressions to be printed. In this example, the %d is an example of format information. The %d is sometimes called a placeholder; it specifies where the value of the expression that is after the string is to be printed. The character in the placeholder is called the conversion character, and it specifies the type of value that is being printed. There are others of the simple placeholders:



Placeholder (character)	Description	Example
%d	Format as a *integer*	>> fprintf ('%d', 4^5) 1024>>
%f	Format as a *floating point* value	>> fprintf ('%f', sqrt(90.25)) 9.500000>>
%g	Format as the *most compact* form (*no trailing zero*)	>> fprintf ('%g', sqrt(90.25)) 9.5>>
%e	Format as a floating point value in *scientific* notation	>> fprintf ('%e', pi) 3.141593e+000>>
%s	Format as a *string*	>> fprintf ('%s', '3*8/2') 3*8/2>>
\n	Insert a *new line* in the output string	>> fprintf ('Welcome! \n this is MATLAB') Welcome! this is MATLAB>>
\t	Insert a *tab* in the output string	>> fprintf ('Welcome! \t this is MATLAB') Welcome! this is MATLAB>>

The main differences between these two commands can be summarized as follows :



disp
. Simple to use.
. Provide limited control over the appearance of output

fprintf
. Slightly more complicated than disp.
. Provide total control over the appearance of output

Example.

```
% This script file calculates the average of points
% scored in three games.
% The point from each game are assigned to a variable
% by using the `input' command.
```

Solve:

```
game1 = input('Enter the points scored in the first game ');
game2 = input('Enter the points scored in the second game ');
game3 = input('Enter the points scored in the third game ');
average = (game1+game2+game3)/3
```

The following shows the command prompt when this script file (saved as example) is executed.

```
>> example
>> Enter the points scored in the first game 15
>> Enter the points scored in the second game 23
>> Enter the points scored in the third game 10
average =
16
```

Example: Write a program to test the degrees of 40 students and find: (named file “students”)
1)The number of students those degrees above 80
2)The number of students those degrees between 60 and 79
3)The number of students those failed and their degrees

Solve:

```
clear
clc
Degree= randi([25,100],1,40)
DegreeAbove80=sum(Degree>80);
Degree60to79=sum(Degree>=60 & Degree<80);
Failures=sum(Degree<50);
fprintf('\n')
disp('------------------------------------------------')
fprintf('The no. of students with degree above 80 : %d\n',DegreeAbove80)
fprintf('The no. of students with degrees 60 to 79 : %d\n',Degree60to79)
fprintf('The no. of failures students: %d\n',Failures)
fprintf('And their degrees are :\n')
disp(Degree(Degree<50))
```