The following are some things I have learned to keep in mind when working with *MATLAB* - perhaps they will be useful to you as well. I have made no effort to organize them in any structured way.

Script Files and Function Files

If you are a non-programmer like me, I guess you can think of both script files and function files as "programs" that *MATLAB* executes in the **workspace**, which is manipulated through the command widow. A function file can also execute in a script file as long both are saved in the same directory (folder). The main differences are two: 1) A script file is "self-contained," in the sense that it can be simply run in the command window by typing the name of the file and hitting *Enter*. On the other hand, a function file usually has numerical **arguments** (inputs, one or more) that must be given values when "calling" the file in the command window. 2) A script file uses **global variables** (see below) that are either defined within the script file or which already exist in the workspace. Any variables defined within a script become global variables in the workspace after the script is run. Any variables in a function file must either have values "passed to" the function when calling it, or given values within the function file itself. Any variables defined within the function file are **local variables** that will not exist within the workspace after the function is run.

Global and Local Variables

Global variables are variables that have been defined in the command window or in a script file. Once defined they "live on" in the workspace until cleared using the `clear` command. At any time one can find out the global variables that exist by using the `who` command. *It is generally a good idea when writing a script file to begin with the command line* `clear all;` *to clear all variables, and then define needed variables within the script itself.* **Local variables** are variables that are used within function files. They must be assigned values within the function file or have values passed to them when calling the function.

More on Function Files

When designing a function file you should determine the variables that will be the arguments ("inputs") of the function, and what variables the function should return (whose values will be the "outputs" of the function). The letters to be used inside the function file for those variables should be used in naming the function. So, for example, suppose that we wish to create a function called `hpstr`, whose arguments will be a, b and c, and that will return two variables s and t that will be defined within the function file. Then the first line (other than comments) of our function file would be

```
function [s,t]=hpstr(a,b,c);
```

Here the `(a,b,c)` tells us that the function will have three arguments passed to it, and those arguments will be values that are assigned to the local variables a, b and c. The `hpstr` is the name of the function, and needs to be used when calling the function in the *MATLAB* workspace (command window). The `[s,t]` tells us that the function will return two values, the values assigned to the local variables s and t by the time the function file is done executing.

When calling the function, the values of s and t can be assigned to any global variables in the workspace. So, for example, if we include the command line `[x,y]=hpstr(3,-7,1)`, the function will take in the values 3, -7 and 1 and assign them to the local variables a, b and c within the `hpstr` function file. Those values will then presumably be used to find values for the local variables s and t, but the function call `[x,y]=hpstr(3,-7,1)` then assigns those values to the global variables x and y in the workspace.

Some Useful Commands and Keystrokes

- If a script or function file is not running, the very first thing you should check is whether the path in the command window is for where you have the file stored. *Always check that when you get an error message that is not clear what it is about.*
- You may try running a script or function that gets "stuck." You can stop its execution using *Ctrl-C.*
- In the command window you can use the uparrow key to recall any previous statements (each press of the uparrow takes you back one statement/command).
- Putting a semicolon at the end of any *MATLAB* command suppresses the output of that command. Almost all lines in a script or function file of any complexity should have semicolons, or the user will be inundated with results of all the variable assignments and calculations when the file is run or called.

- The first command of script files should be `clear all` so that values of global variables from the workspace aren't used inadvertently.