

MATLAB Workshop 6 - Scripts and the Workspace

Objectives: Learn about how MATLAB scripts interact with the workspace.

- **Scripts and the Workspace**

An understanding of how the scripts that you are learning to create for MATLAB interact with the MATLAB workspace is critical for control of your workspace and *debugging* your scripts. Frequently, scripts will have errors when first created. The process of removing the errors so that the scripts work properly and provide correct answers is called *debugging*.

This workshop requires that MATLAB scripts named `wkshp6_ac1.m`, `wkshp6_ac3.m`, and `wkshp6_ac4.m` be in the current directory. These scripts are available on the program disk. Please copy them into the current MATLAB directory before proceeding.

(1) Scripts put their variables in the workspace.

Whenever a script is run, all variables that have values assigned are placed into the workspace. To clearly see this, start with a clear workspace. Enter

```
» clear all
```

to completely clear the workspace. Check that this has been done by entering

```
» who
```

```
»
```

No variables should be listed in response. Now run `wkshp6_ac1` by entering

```
» wkshp6_ac1
```

```
Workshop 6, Activities 1 and 2
```

```
Please provide diary name ==>
```

The script header and a request for a diary name appear. This particular script allows you to provide a diary name that will hold the contents of running the script. You can look at the script file later to see how this is done. Of interest is that you can use a script to turn the diary on and off. Enter a diary name of your choosing and press enter. The MATLAB command prompt appears. This means that the script has run!! Yet nothing appeared in the Command Window. So what did happen? Enter

```
» who
```

```
Your variables are:
```

```
diary_name x y z
```

Running the MATLAB script has put variables into your workspace. If you want, you can check to see what the variable types are (`whos`) and what values they contain. You might also want to check on whether the diary file has been placed in the current directory.

- Running a MATLAB script will add all of the script variables to the active workspace.

(2) Scripts will overwrite existing values for variables of the same name in the workspace.

Scripts use the active workspace in the same manner that you do when you operate from the command window. Any command issued in a script is the same as if you used the command directly. Sometimes unexpected results occur. One is that scripts will replace values for variables with the same name with their own values. To see this, enter the following

```
» clear all
```

```
» x = 7
```

```

x =
    7
>> who
Your variables are:
x

```

to completely clear the workspace and then introduce the variable `x` with the value 7.

Now run `wkshp6_ac1` again by entering

```

>> wkshp6_ac1
Workshop 6, Activities 1 and 2
Please provide diary name ==>

```

Respond with a diary name of your choice and enter

```

>> who
Your variables are:
diary_name x          y          z

```

The script again added its own variables. Check on the value for `x`. Enter

```

>> x
x =
    1

```

Running the MATLAB script changed the value of `x` from 7 to 1. This can be dangerous because there is no warning that the script was using a variable name the same as one you already had in the workspace.

- Running a MATLAB script will change the values assigned to variables with the same name.
- Avoid this by using unique variable names in your scripts.

(3) Scripts will use existing variables of the same name in the workspace.

This issue is important in designing and debugging scripts. A common method for developing scripts is to work a sample problem with the diary on and then edit the diary to create a script. This is the method illustrated in Workshop 5. However, there is a problem with this approach in that the workspace contains the variables that were used to develop the script (same names). Improper editing of the diary file to create the script can result in a situation such as the following. Enter

```

>> clear all
>> x = 1;
>> y = 2;
>> z = 3;
>> wkshp6_ac3
Workshop 6, Activity 3
Please provide diary name ==>

```

Respond with a diary name of your choice and press enter. The command prompt appears, indicating that the script has executed without any problems.

Now try (provide a diary name of your choice when requested)

```

>> clear all
>> wkshp6_ac3
Workshop 6, Activity 3
Please provide diary name ==>
??? Undefined function or variable 'x'.
Error in ==> c:\temp\wkshp6_ac3.m
On line 23 ==>      d = x*y;

```

The script that ran just fine a moment ago is now creating error messages!! In particular, the error message indicates that a variable `x`, used in line 23 of the script, has not been defined. This message did not occur on the first running because we had defined a variable `x` in the workspace prior to running the script. The script just used that value. The workspace had been cleared of all variables on the second running - there was no value for `x` available when the script reached the line that needed it. Hence the error message.

- A MATLAB script will use variables and values already in the workspace if not provided in the script.
- Avoid this by using unique variable names in your scripts.

(4) Using the `clear` command inside a script is **dangerous**.

The `clear` command inside a script will clear the workspace just as it does if issued at the command prompt. In illustration, enter

```
» clear all
» x = 1;
» y = 2;
» z = 3;
```

This cleared the workspace and created three variables which you can list by using `who`.

Now enter

```
» wkshp6_ac4
Workshop 6, Activity 4
Please provide diary name ==>
```

The script runs, as evidenced by the appearance of the command prompt. Check to see what variables are in the workspace.

```
» who
»
```

Running the script removed all variables from the workspace. A `clear` command was used inside the script. Look at the script to see where it was used. Using the `clear` command inside a script could erase variables and values that you worked hard to get into the workspace. Good practice is to only issue the `clear` command from the command prompt in the Command Window and never in a script.

- The `clear` command in a script will clear the workspace the same as if issued at the command prompt.
- Avoid using `clear` in your scripts.

Recap: You should have learned

- Scripts run in the current workspace.
- A MATLAB command in a script is the same as a MATLAB command in the command window.
- Scripts can add variables to the workspace.
- Scripts can change variable values in the workspace.
- If a variable is not defined in a script, the script will use the value of a variable of the same name in the workspace.
- `clear` is a dangerous command in a script.