



PROGRAMMING WITH MATLAB

WEEK 4





SCRIPT FILES AND FUNCTIONS



SCRIPT FILES

- All of the built-in commands in MATLAB are script files or functions (sqrt, exp, mean, ...)
- MATLAB allows the user to create his/her own m-files for specific applications or problems
- A script file is simply a collection of executable MATLAB commands. To create a new script file, click on the New Script icon on the left side of the Home Tab

SCRIPT FILES

The screenshot displays the MATLAB Editor interface. The main window is titled "Editor - Untitled" and shows a blank script file. The interface includes a ribbon with tabs for "HOME", "PLOTS", and "APPS". The "HOME" tab is active, showing various toolbars for file operations (New, Open, Save, Find Files, Compare, Import Data, Save Workspace, Clear Workspace), code analysis (Analyze Code, Run and Time, Clear Commands), and development (Simulink, Layout, Preferences, Set Path, Add-Ons, Help, Community, Request Support, Learn MATLAB).

Below the main editor window, the Command Window is visible, displaying the output of a script. The output is as follows:

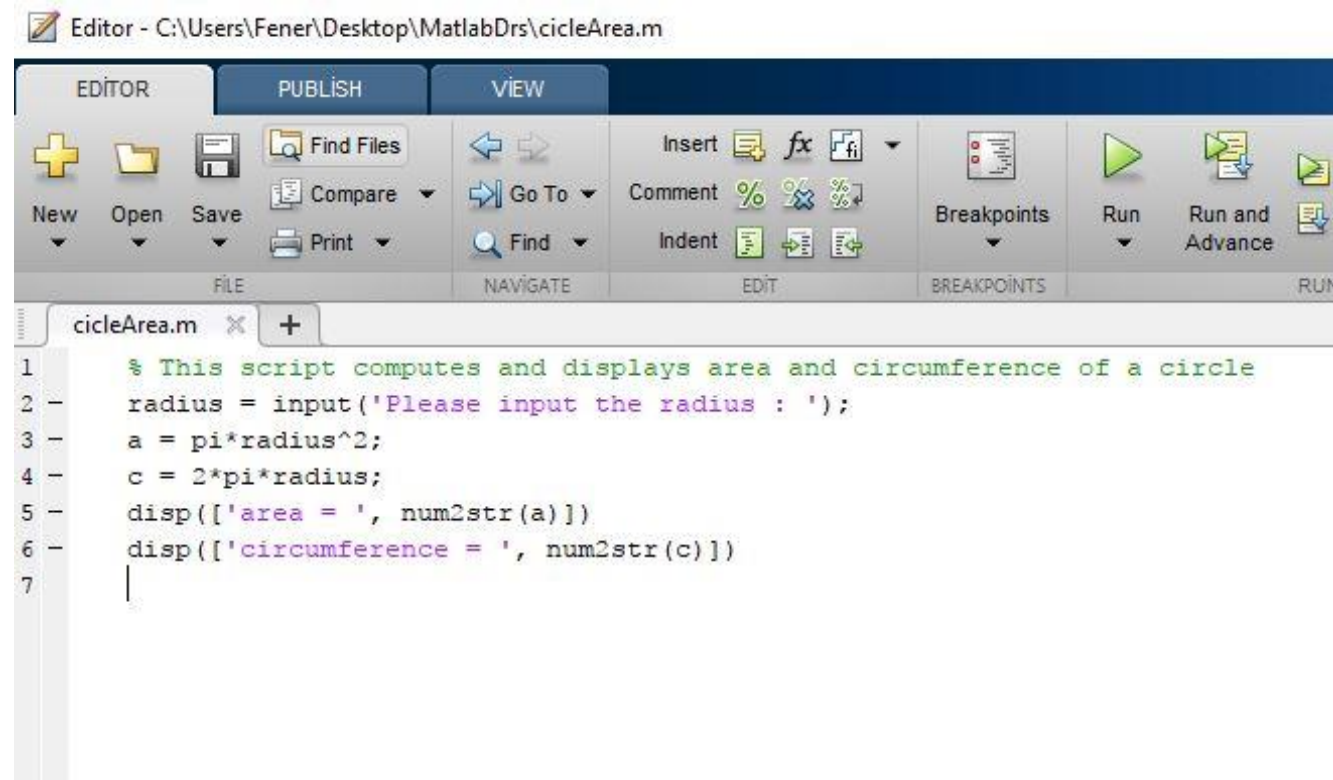
```
Value  
[-15 35 30;10 40 -25;...  
'b'  
[1;7;17;3;11;19;5;13;23]  
2x3 cell  
'f'  
'f1'  
'f2'  
[1;3;2;1;3]  
[1;1;2;3;3]  
3  
2x1 cell  
2x3 cell  
1x2 struct  
3x1 cell  
[0 0 0 0 0 0 0 0]  
1x2 cell  
[4 -3 2.1000]  
[-3 7 6;2 8 -5]  
[1 2 3 4]  
[2 -3 5;3 -9 1;-2 2 -3]  
220  
[3 -42 -26;38 -88 33]
```

The status bar at the bottom of the editor window shows "script" and "Ln 1 Col 1".

SCRIPT FILES

- Type a set of executable commands in the editor window
- Save the file in an appropriate folder
- To run the script file: Hit the Green Run Arrow in the Editor window or type the name of the file at the command prompt in the MATLAB command window

SCRIPT FILES



Editor - C:\Users\Fener\Desktop\MatlabDrs\cicleArea.m

EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Go To Find Insert Comment Indent Breakpoints Run Run and Advance

```
1 % This script computes and displays area and circumference of a circle
2 - radius = input('Please input the radius : ');
3 - a = pi*radius^2;
4 - c = 2*pi*radius;
5 - disp(['area = ', num2str(a)])
6 - disp(['circumference = ', num2str(c)])
7 |
```

SCRIPT FILES

- Now run your script file by clicking on the Green Arrow in the m-file editor window

```
>> cicleArea
```

```
Please input the radius : 5
```

```
area = 78.5398
```

```
circumference = 31.4159
```

USER-DEFINED FUNCTIONS

- **Syntax** : `function [output arguments] = functionname(input arguments)`

The function name should be the same as the file name in which it is saved

```
function [u] = calc(x,y)
```

```
a = x.^3;
```

```
u = a + 5./y;
```

call this function from command window:

```
>> u = calc(3, 5)
```

```
u =
```

```
28
```


USER-DEFINED FUNCTIONS

- Function scope
- Local variables : The variable names given in the function definition are specific (local) to that function.

The variables in a function are deleted after the function is executed.

However, these variables can be passed out of the function as an output argument.

with the “global” command, variables are available for workspace and for other functions declaring these variables as global.

Syntax : global x y z

USER-DEFINED FUNCTIONS

MATLAB functions are often “overloaded”

try the following:

```
>> x = ones(3, 4, 5);
```

```
>> b = size(x);
```

```
>> [m, n] = size(x);
```

```
>> [u, v, w] = size(x);
```

```
>> b2 = size(x,2);
```

You can overload the functions you write yourself by making the input and output arguments variable (see `varargin`, `nargout`, `varargout`, `nargout`)

USER-DEFINED FUNCTIONS

- To create a handle for a function, precede the function name with an @ sign.

```
function [ y ] = fun1(x)
```

```
y = x.^2 + log(x) + 5;
```

```
end
```

```
>> f = @fun1
```

```
>> y = f(3)
```

```
y =
```

```
15.0986
```

USER-DEFINED FUNCTIONS

- **Anonymous functions** :An anonymous function is a one-line expression-based MATLAB function that does not require a program file.

Syntax : `h = @(arglist) anonymous_function`

```
>> cub = @(x) x.^3;
```

```
>> a = cub(5)
```

```
a =
```

```
125
```

USER-DEFINED FUNCTIONS

- You can create an array of function handles by collecting them into a cell or structure array.

```
>> C = {@f1, @fun1, @cos};
```

```
>> C{1}(3)
```

```
ans =
```

```
0.0996
```

or you can use structure array

```
>> S.x = @f1; S.y = @fun1; S.z = @cos;
```

```
>> S.z(pi)
```

```
ans =
```

```
-1
```

USER-DEFINED FUNCTIONS

- **Nested functions:** A nested function is a function that is completely contained within a parent function.

```
function main
```

```
x = 3;
```

```
    nestfun
```

```
        function nestedfun
```

```
            x = x.^2 - 3;
```

```
        end
```

```
end
```

Typically, functions do not require an end statement. However, to nest any function in a program file, all functions in that file must use an end statement.

USER-DEFINED FUNCTIONS

- **Private functions:** Private functions are useful when you want to limit the scope of a function. Private functions reside in subfolder with the special name private, and they are visible only to functions in the parent directory. Then, the function is available only to functions in the folder immediately above the private subfolder, or to scripts called by the functions that reside in the parent folder.