



MATLAB M-FILES

tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

MATLAB (Matrix Laboratory) is a programming language developed by a computer software company MathWorks. M-File also called as the script file is a series of MATLAB commands that will get executed sequentially when the file is executed. This tutorial is designed to give students fluency in working on the M-files in MATLAB. Problem-based examples have also been given in simple and easy way to make your learning fast and effective.

Audience

This tutorial has been prepared for the beginners to help them understand basic to advanced functionality of MATLAB M-File. After completing this tutorial you will find yourself at a moderate level of expertise in using M-files from where you can take yourself to next levels.

Prerequisites

We assume you have a little knowledge of any computer programming and understand concepts like variables, constants, expression, statements, etc. If you have done programming in any other high-level programming language like C, C++ or Java, then it will be very much beneficial and learning MATLAB M-File will be like a fun for you.

Copyright & Disclaimer

© Copyright 2021 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

| | |
|--|-----------|
| About the Tutorial | i |
| Audience..... | i |
| Prerequisites..... | i |
| Copyright & Disclaimer | i |
| Table of Contents | ii |
| 1. MATLAB M-File — Introduction | 1 |
| Program Files..... | 1 |
| 2. MATLAB M-File — Create and Save..... | 3 |
| Create M-file..... | 3 |
| Save M-file | 4 |
| 3. MATLAB M-File — Run..... | 6 |
| 4. MATLAB M-File — Functions..... | 8 |
| 5. MATLAB M-File — Import Data | 11 |

1. MATLAB M-File — Introduction

MATLAB allows you to write a series of commands into a file and execute the file as a complete unit, like writing a function and calling it. It is done using M-file.

M-File also called as the script file is a series of MATLAB commands that will get executed sequentially when the file is executed.

The m-file is saved with **.m** extension.

Program Files

MATLAB allows writing two kinds of program files, which are as follows:

Scripts

Script files are program files with **.m** extension. In these files, you write a series of commands, which you want to execute together. Scripts do not accept inputs and do not return any outputs. They operate on data in the workspace.

A script file looks as follows:

A screenshot of a MATLAB script editor window titled 'firstmfile.m'. The editor shows five lines of MATLAB code:

```
1 - a = 5; b = 7;  
2 - c = a + b  
3 - d = c + sin(b)  
4 - e = 5 * d  
5 - f = exp(-d)
```

Functions

Function files are also program files with **.m** extension. Functions can accept inputs and return outputs. Internal variables are local to the function.

A function file looks as follows:

```
MaxNumber.m x +
1  function value = MaxNumber(a,b)
2  % Summary of this function goes here
3  % Detailed explanation goes here
4  value = a;
5  if (b > value)
6      value = b;
7  end
```

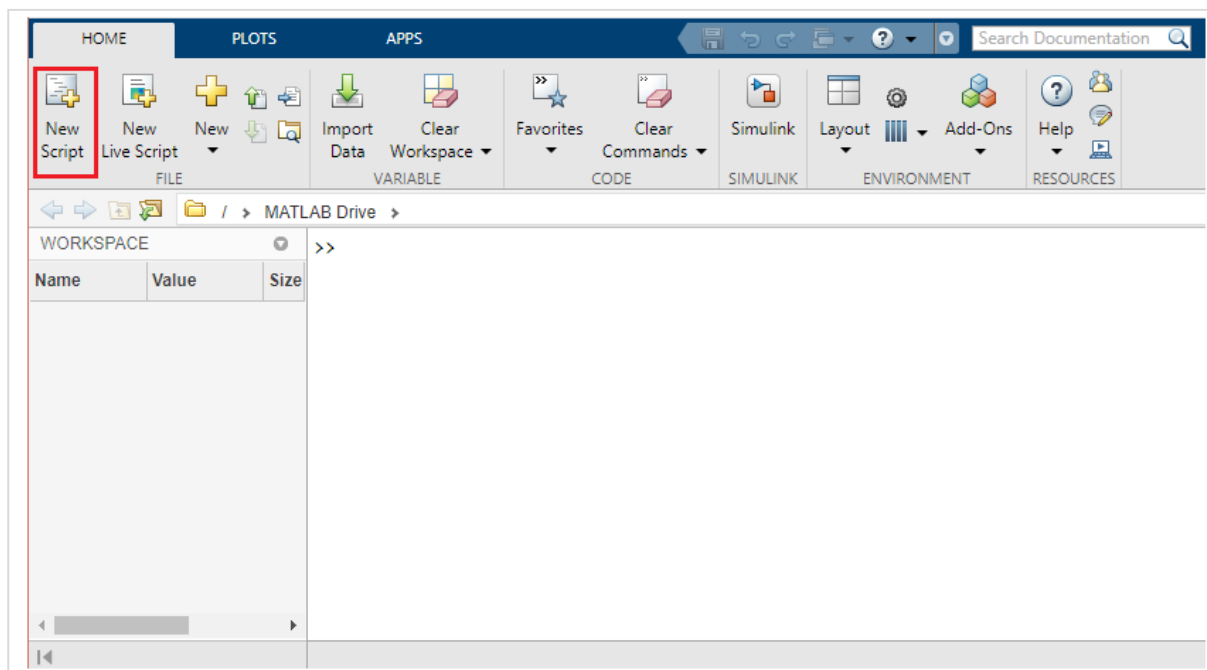
Let us understand how to create and run the m-file in the next chapters.

2. MATLAB M-File — Create and Save

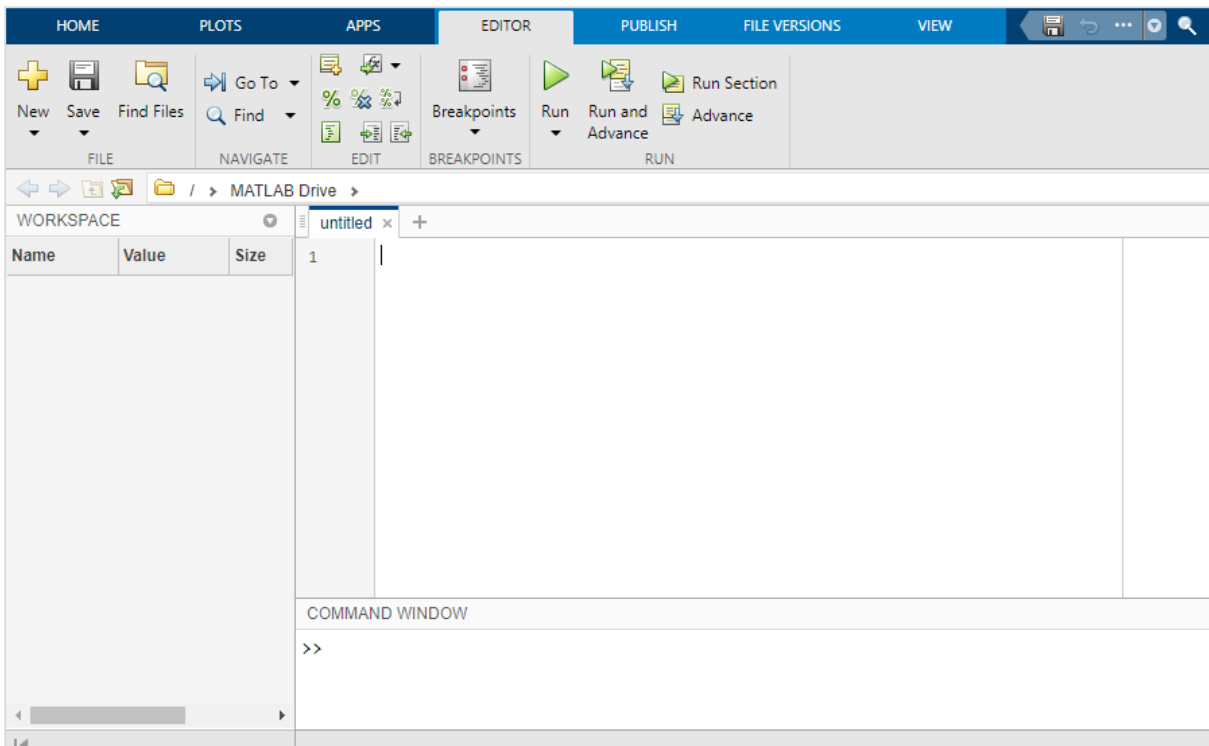
In this chapter, we will learn how to create and save a M-file. Let us begin by understanding about creating a M-file.

Create M-file

To create m-file, we will make use of MATLAB IDE as shown below. Here IDE refers to an integrated development environment.



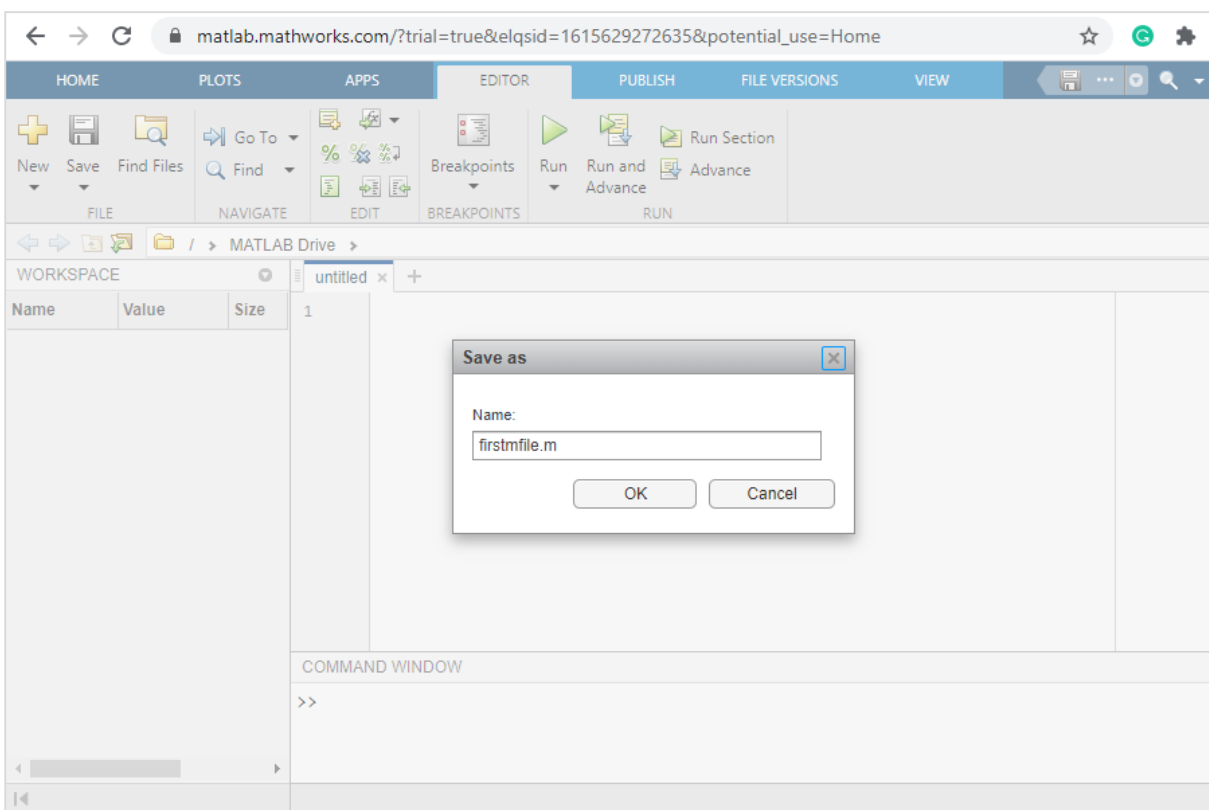
Click on New Script highlighted above to open a new script file.



Save M-file

You will get an untitled file. Let us save the file as **firstmfile.m**.

Click on the save button and it will open a popup, where you can enter the name of the file.



Click on OK to save the file.

Now, you are free to write your commands in the file below:



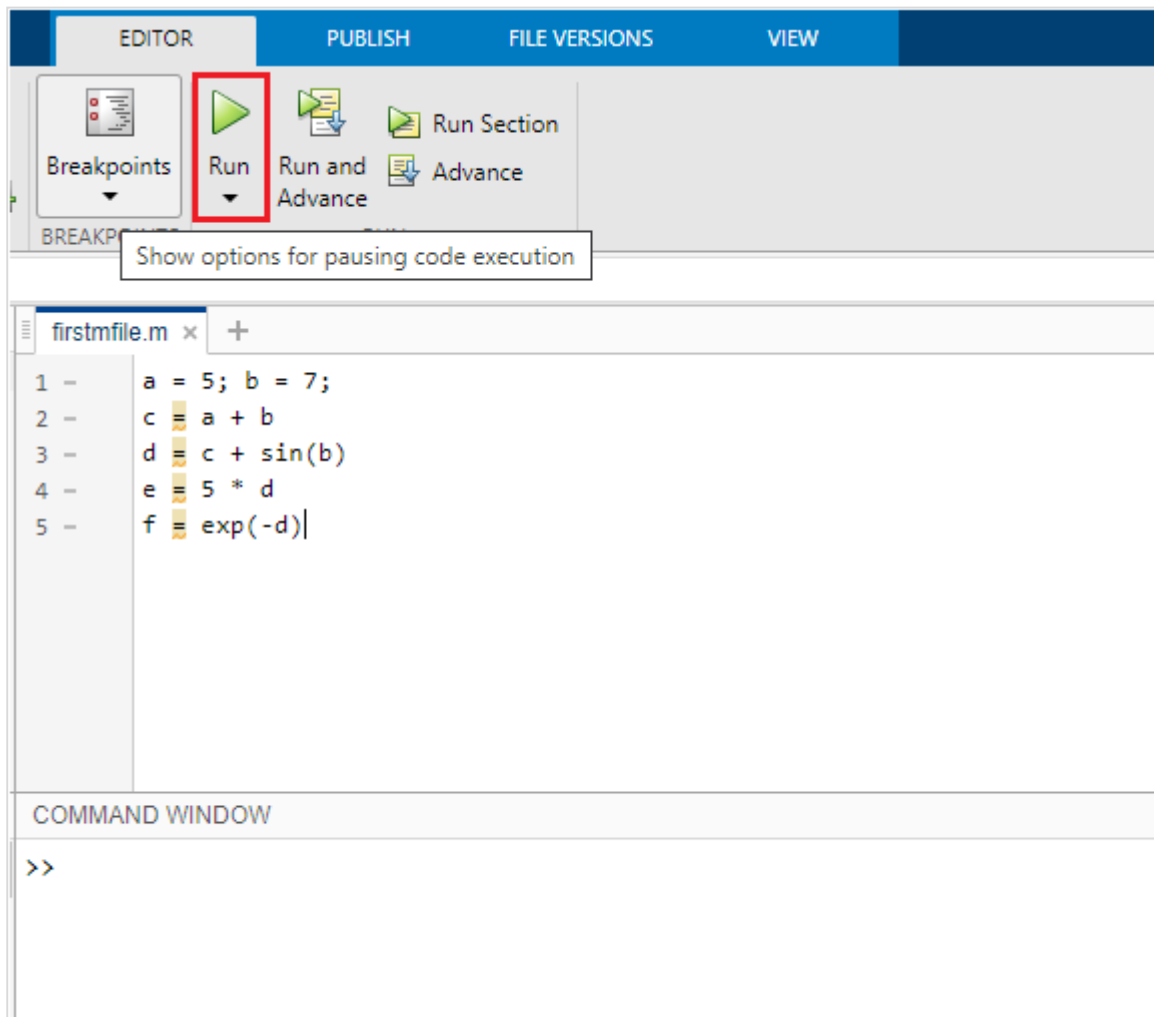
3. MATLAB M-File — Run

In the previous chapter, we have seen how to create and save a m-file. Now in this chapter, we will write a simple code and run the same.

Following is the code which will run inside the **firstmfile.m**

```
a = 5; b = 7;  
c = a + b  
d = c + sin(b)  
e = 5 * d  
f = exp(-d)
```

This is how it will look inside MATLAB editor:



Click on the Run button as highlighted above to see the result inside the command window as shown below:

```
COMMAND WINDOW
>> firstmfile

c =

    12

d =

    12.6570

e =

    63.2849

f =

    3.1852e-06
```

4. MATLAB M-File — Functions

A function is a group of statements that together perform a task. In MATLAB, functions are defined in separate files. The name of the file and name of the function should be the same.

Functions operate on variables within their own workspace, which is also called the local workspace. These functions separate the variables from the workspace which you access at the MATLAB command prompt. This is called the base workspace.

Functions can accept more than one input arguments and may return more than one output arguments.

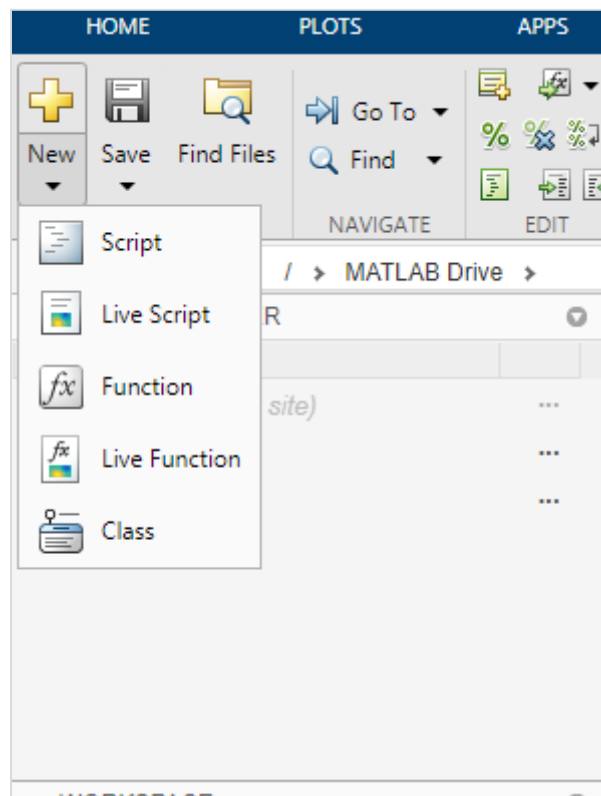
The syntax of a function statement is as follows:

```
function [out1,out2, ..., outN] = myfun(in1,in2,in3, ..., inN)
```

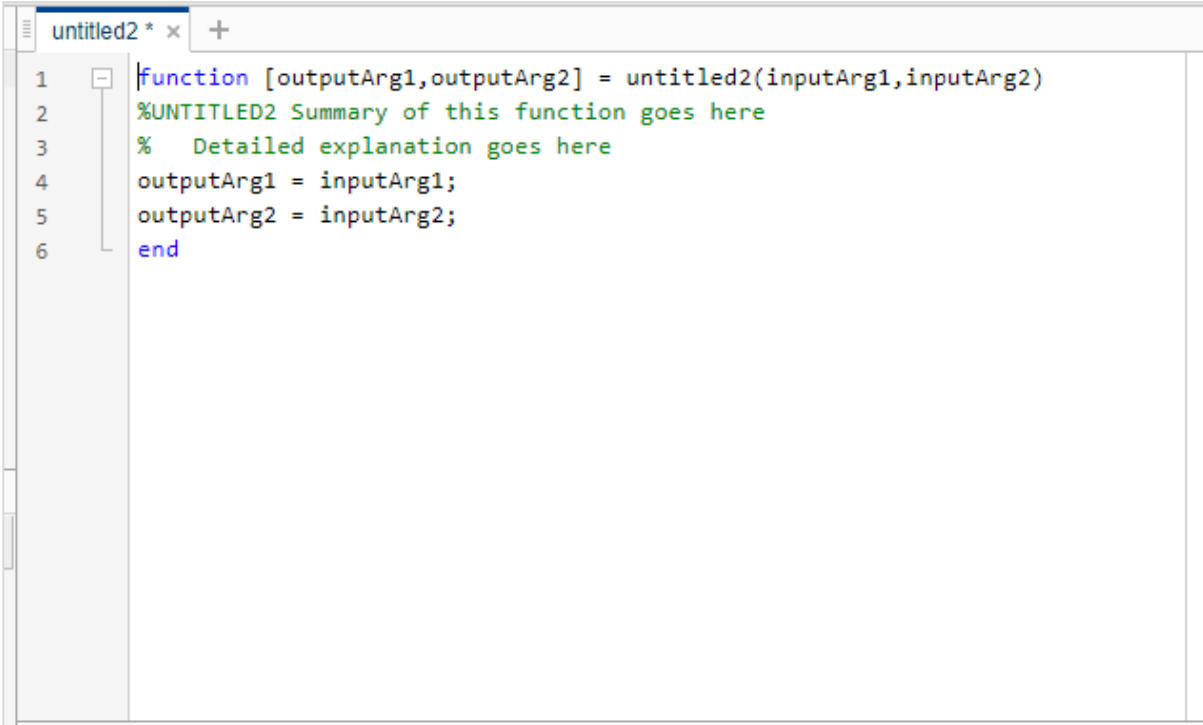
Here out1, out2...outN are output variables. It can be a single variable or comma separated. The variables in1, in2, in3...inN are input variables which can be single variable or comma separated ones. The function in MATLAB starts with the keyword function as shown in the syntax.

While naming your m-file, you should take care that the file name and the function name has to match. You cannot use the name of any built-in function available in MATLAB.

Let us now create a simple function and save it as **.m** file and run it. In MATLAB, the IDE allows you to select what type of file you want to create as shown below:

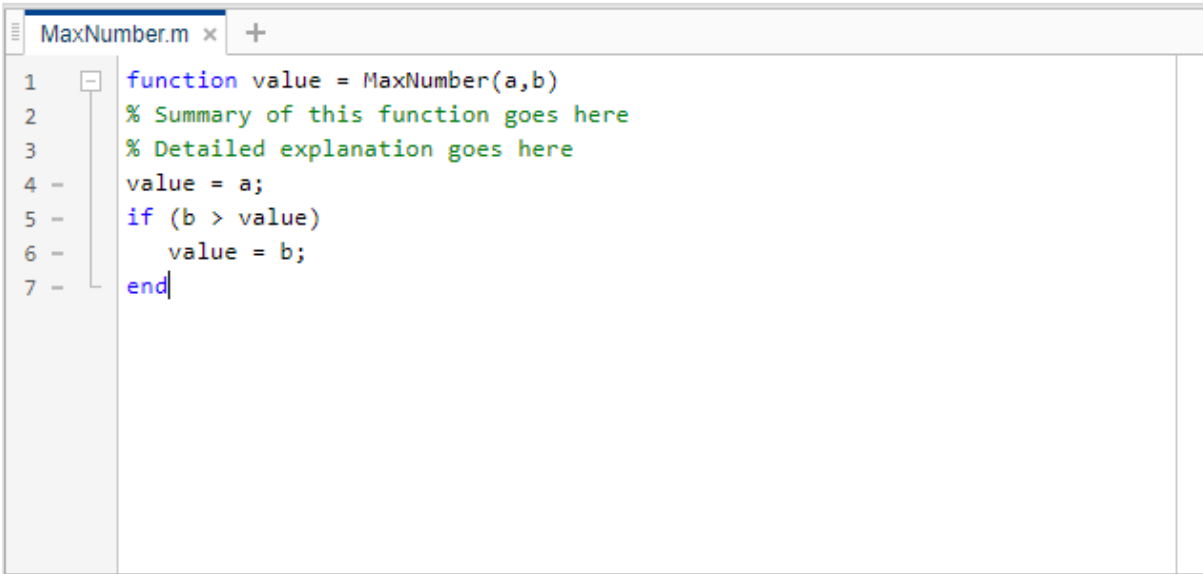


Click on the Function, and it will open a new file as shown below:



```
untitled2 * x +
1 function [outputArg1,outputArg2] = untitled2(inputArg1,inputArg2)
2 %UNTITLED2 Summary of this function goes here
3 % Detailed explanation goes here
4 outputArg1 = inputArg1;
5 outputArg2 = inputArg2;
6 end
```

Now you can update the output variables, the function name and the input variables in the above untitled file and save the file with the same name as the function name.



```
MaxNumber.m x +
1 function value = MaxNumber(a,b)
2 % Summary of this function goes here
3 % Detailed explanation goes here
4 value = a;
5 if (b > value)
6     value = b;
7 end
```

The name of our function is MaxNumber() and it gives the maximum number from the input value passed.

Now let us run the function to get the output. You can call the function by using MaxNumber(100,50).

```
COMMAND WINDOW
>> MaxNumber(100, 50)

ans =

    100

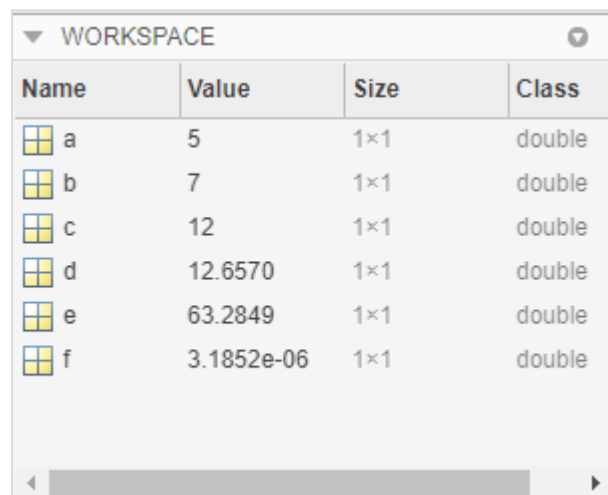
>>
```

5. MATLAB M-File — Import Data

In this chapter, we will understand how to import data from an existing m-file in MATLAB. Consider the m-file firstmfile.m. The contents of the file are as follows:

```
a = 5; b = 7;  
c = a + b  
d = c + sin(b)  
e = 5 * d  
f = exp(-d)
```

When you execute the file, the data variables are available in the workspace, as shown below:

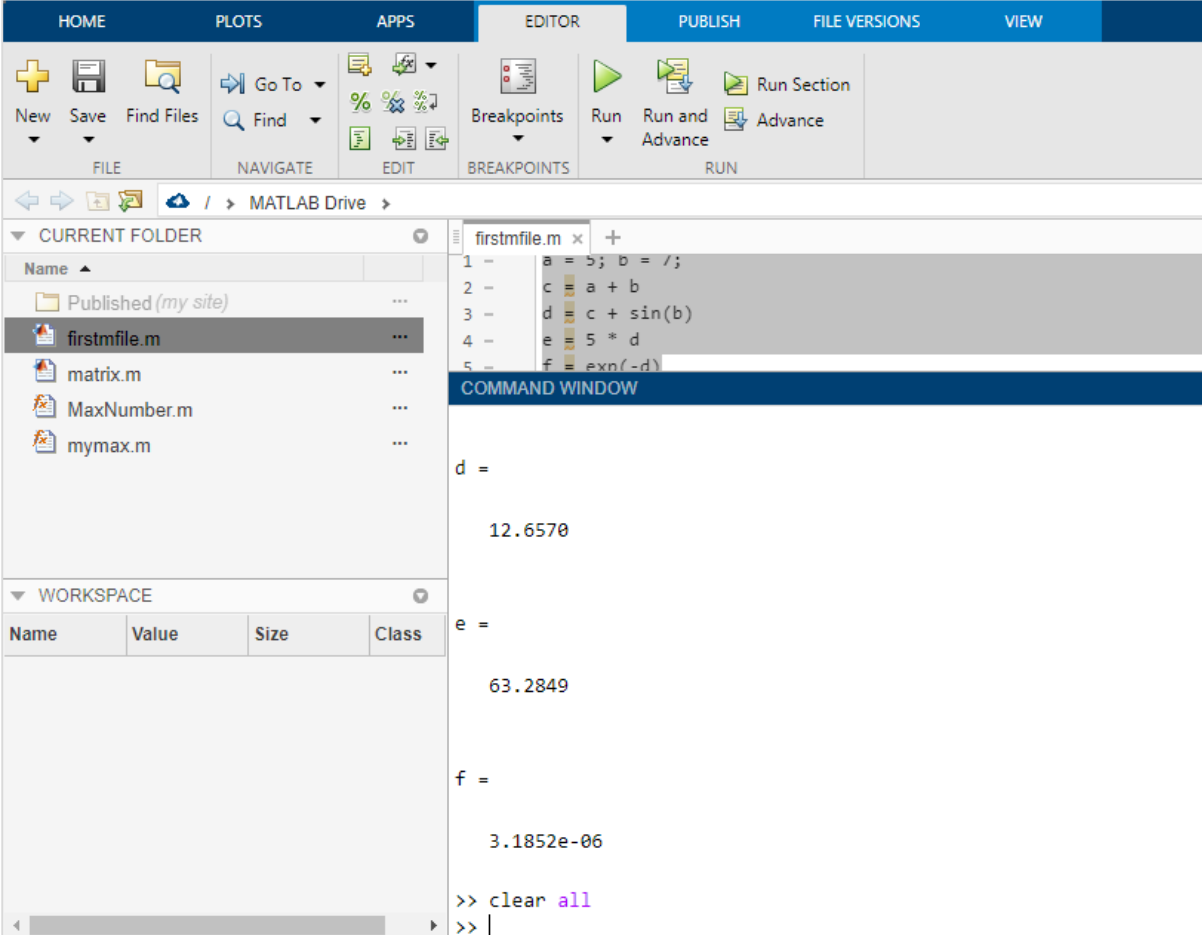


The screenshot shows the MATLAB Workspace window with a table of variables. The table has four columns: Name, Value, Size, and Class. Each row represents a variable with a small grid icon to its left.

| Name | Value | Size | Class |
|------|------------|------|--------|
| a | 5 | 1×1 | double |
| b | 7 | 1×1 | double |
| c | 12 | 1×1 | double |
| d | 12.6570 | 1×1 | double |
| e | 63.2849 | 1×1 | double |
| f | 3.1852e-06 | 1×1 | double |

Let us clear the workspace and the command prompt. Hence, we can now import the file and also check the data available.

To clear workspace, the command is clear all.



The screenshot displays the MATLAB IDE interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, FILE VERSIONS, and VIEW. The ribbon below the menu bar contains icons for New, Save, Find Files, Go To, Find, Breakpoints, Run, Run and Advance, Run Section, and Advance. The main workspace is divided into three sections: CURRENT FOLDER, WORKSPACE, and COMMAND WINDOW.

CURRENT FOLDER: Shows a list of files including Published (my site), firstmfile.m, matrix.m, MaxNumber.m, and mymax.m.

WORKSPACE: A table with columns Name, Value, Size, and Class. It is currently empty.

COMMAND WINDOW: Displays the output of the MATLAB script firstmfile.m. The script code is visible in the Editor window above:

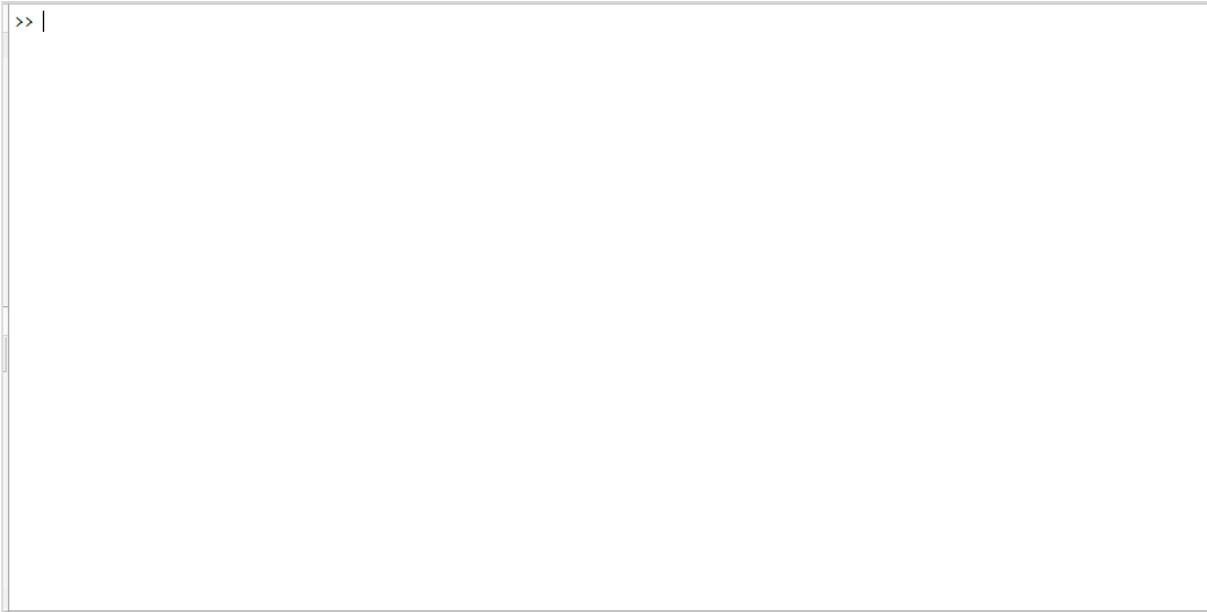
```
1 - a = 5; b = 7;  
2 - c = a + b  
3 - d = c + sin(b)  
4 - e = 5 * d  
5 - f = exp(-d)
```

The Command Window output shows:

```
d =  
  
    12.6570  
  
e =  
  
    63.2849  
  
f =  
  
    3.1852e-06  
  
>> clear all  
>> |
```

The workspace is empty now.

Let us clean the command window by using clc command as shown below:

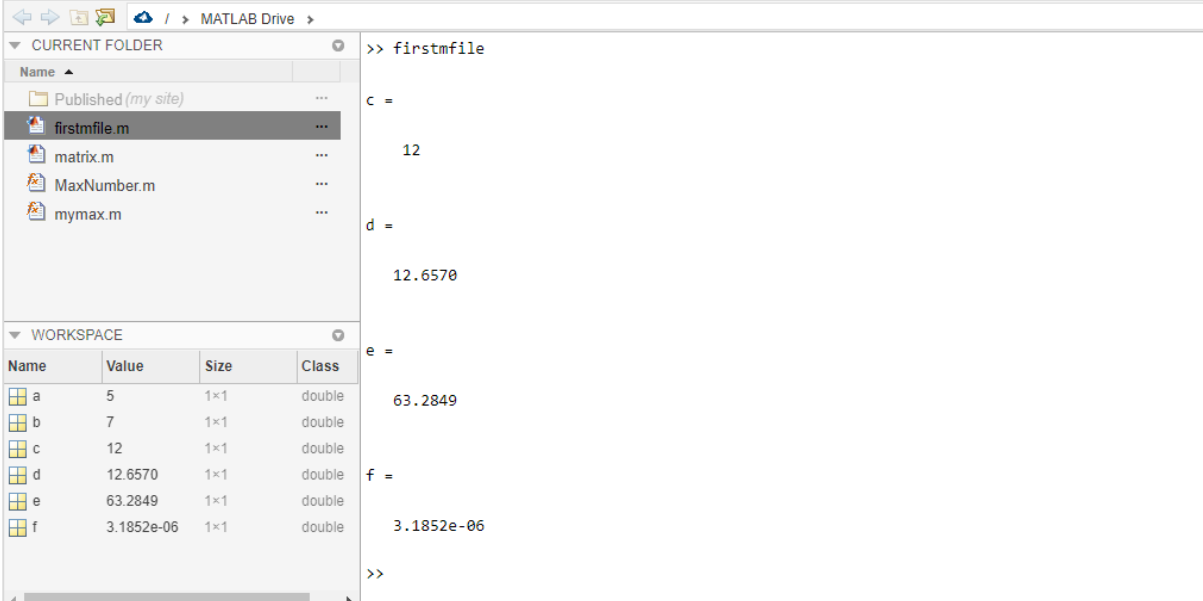


The screenshot shows the MATLAB Command Window with the following text:

```
>> |
```

We have an empty command window.

Now, type the m-file name as shown below. It will execute the code inside the file.



The screenshot shows the MATLAB command window with the current folder set to 'MATLAB Drive'. The file 'firstmfile.m' is selected in the file browser. The command window shows the execution of the command `>> firstmfile`. The output shows the following assignments:

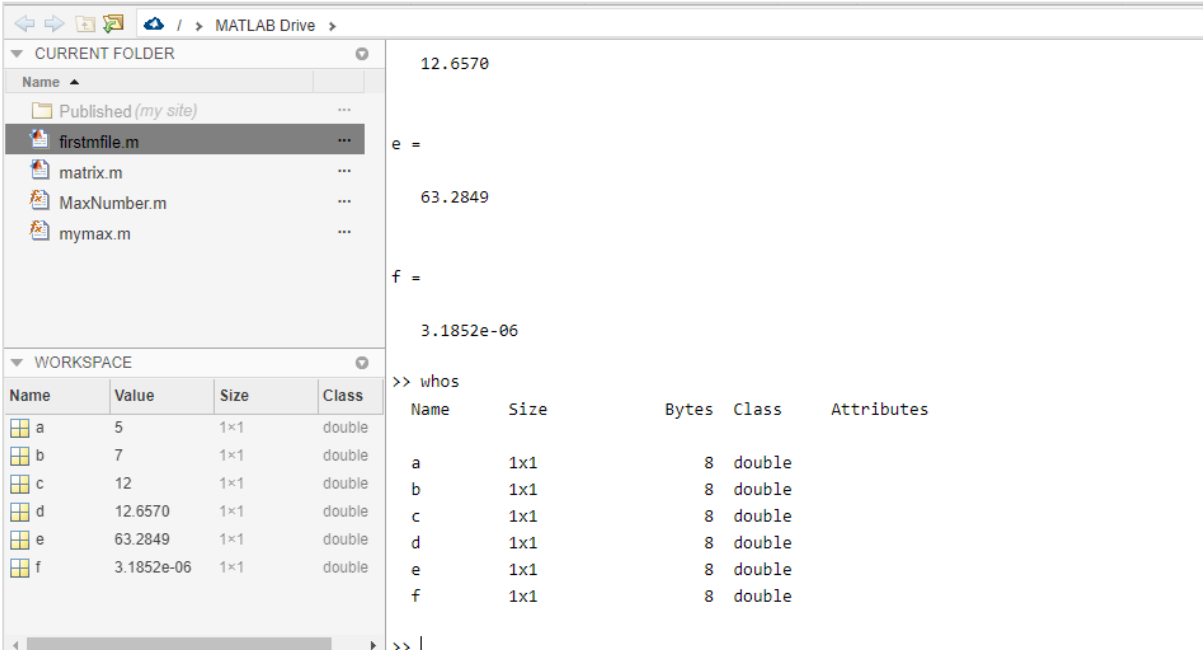
```
c =
    12
d =
    12.6570
e =
    63.2849
f =
    3.1852e-06
>>
```

The workspace table below shows the variables created:

| Name | Value | Size | Class |
|------|------------|------|--------|
| a | 5 | 1×1 | double |
| b | 7 | 1×1 | double |
| c | 12 | 1×1 | double |
| d | 12.6570 | 1×1 | double |
| e | 63.2849 | 1×1 | double |
| f | 3.1852e-06 | 1×1 | double |

The workspace is loaded with all the variables that are used inside the file.

You can also check the details of the variables by using the `whos` command as shown below:



The screenshot shows the MATLAB command window with the current folder set to 'MATLAB Drive'. The file 'firstmfile.m' is selected in the file browser. The command window shows the execution of the command `>> whos`. The output shows the following details:

```
>> whos
Name      Size      Bytes  Class  Attributes
a         1x1         8  double
b         1x1         8  double
c         1x1         8  double
d         1x1         8  double
e         1x1         8  double
f         1x1         8  double
>>
```

The workspace table below shows the variables created:

| Name | Value | Size | Class |
|------|------------|------|--------|
| a | 5 | 1×1 | double |
| b | 7 | 1×1 | double |
| c | 12 | 1×1 | double |
| d | 12.6570 | 1×1 | double |
| e | 63.2849 | 1×1 | double |
| f | 3.1852e-06 | 1×1 | double |