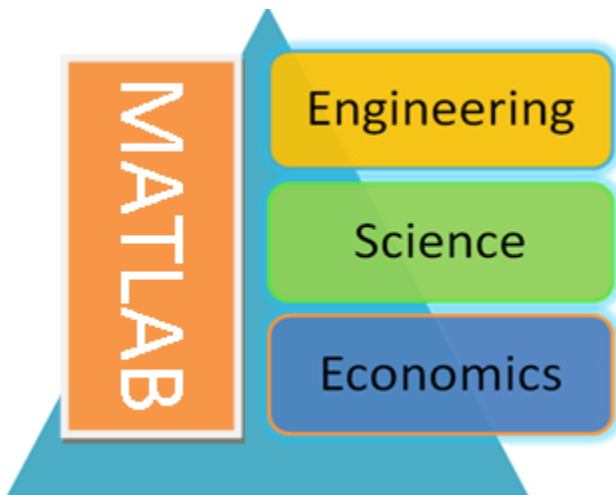
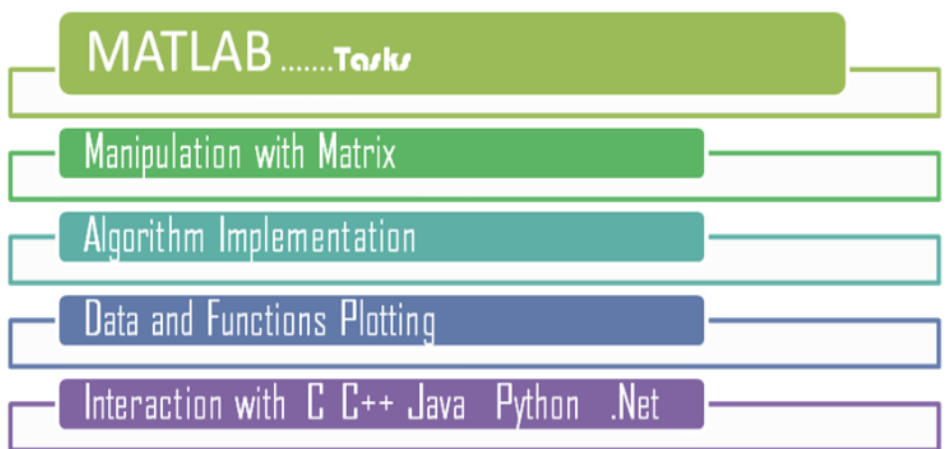


## What is MATLAB?

- MATLAB is a software package for high-performance mathematical computation, visualization, and programming environment. It provides an interactive environment with hundreds of built-in functions for technical computing, graphics, and animations.
- MATLAB stands for Matrix Laboratory. MATLAB was written initially to implement a simple approach to matrix software developed by the LINPACK (Linear system package) and EISPACK (Eigen system package) projects.
- MATLAB is a modern programming language environment, and it has refined data structures, includes built-in editing and debugging tools, and supports object-oriented programming.
- MATLAB is Multi-paradigm. So, it can work with multiple types of programming approaches, such as Functional, Object-Oriented, and Visual.
- MATLAB is used in various disciplines of engineering, science, and economics.



- MATLAB allows several types of tasks, such as manipulations with matrix, algorithm implementation, data, and functions plotting, and can interact with programs written in other programming languages.



- MATLAB is a dynamic and weakly typed programming language.
- MATLAB environment handles tasks of the declaration of the data type of the variables and provision for an appropriate amount of storage for the variables.

## **History of MATLAB**

The development of the MATLAB started in the late 1970s by Cleve Moler, the chairman of the Computer Science department at the University of New Mexico. Cleve wanted to make his students able to use LINPACK & EISPACK (software libraries for numerical computing, written in FORTRAN), and without learning FORTRAN. In 1984, Cleve Moler with Jack Little & Steve Bangert rewrote MATLAB in C and founded MathWorks. These libraries were known as JACKPAC at that time, later these were revised in 2000 for matrix manipulation and named as LAPACK.

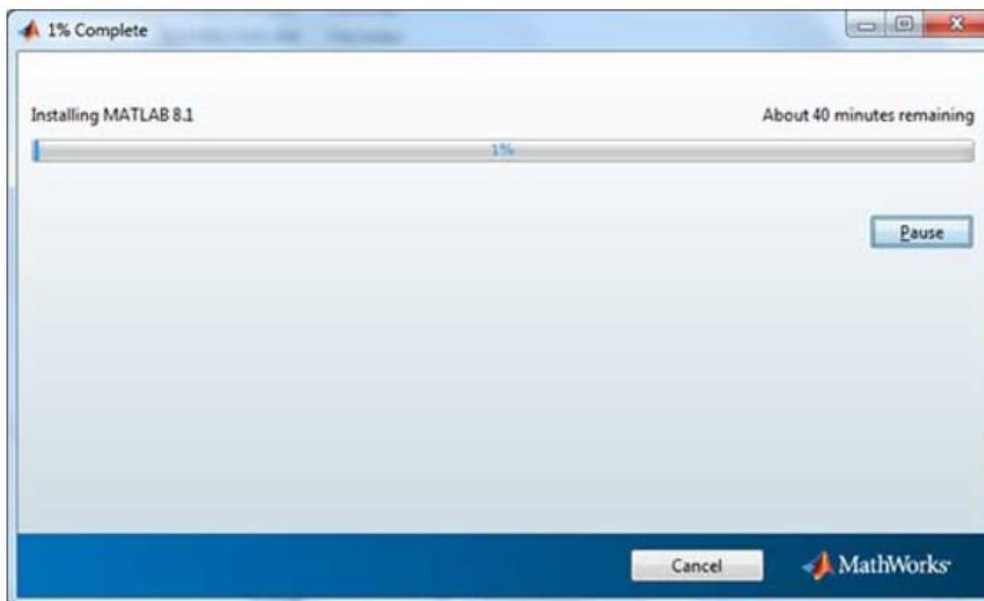
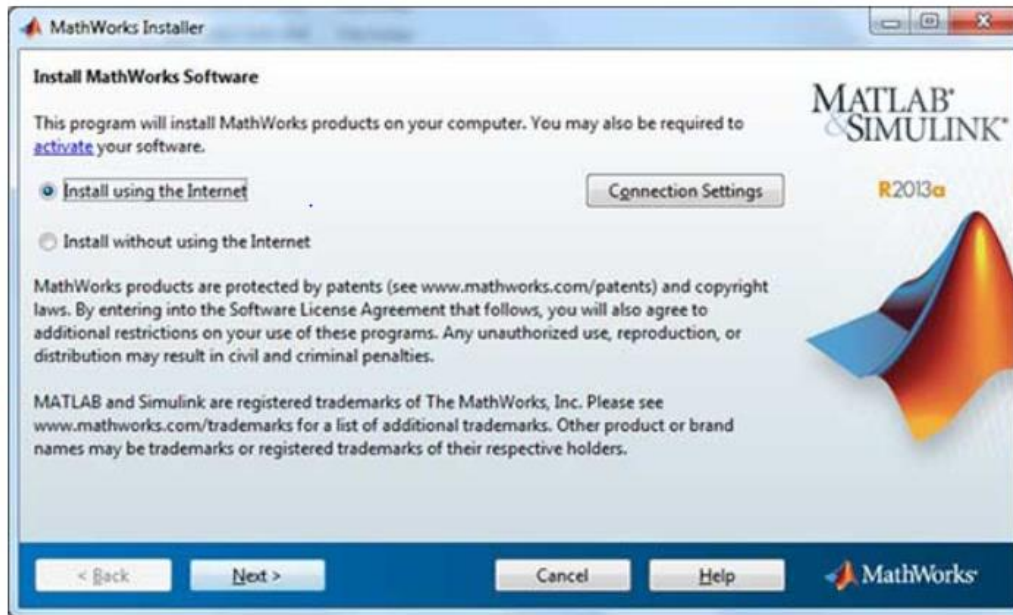
## **Features of MATLAB**

Following are the basic features of MATLAB –

- It is a high-level language for numerical computation, visualization and application development.
- It also provides an interactive environment for iterative exploration, design and problem solving.
- It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
- It provides built-in graphics for visualizing data and tools for creating custom plots.
- MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.
- It provides tools for building applications with custom graphical interfaces.
- It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

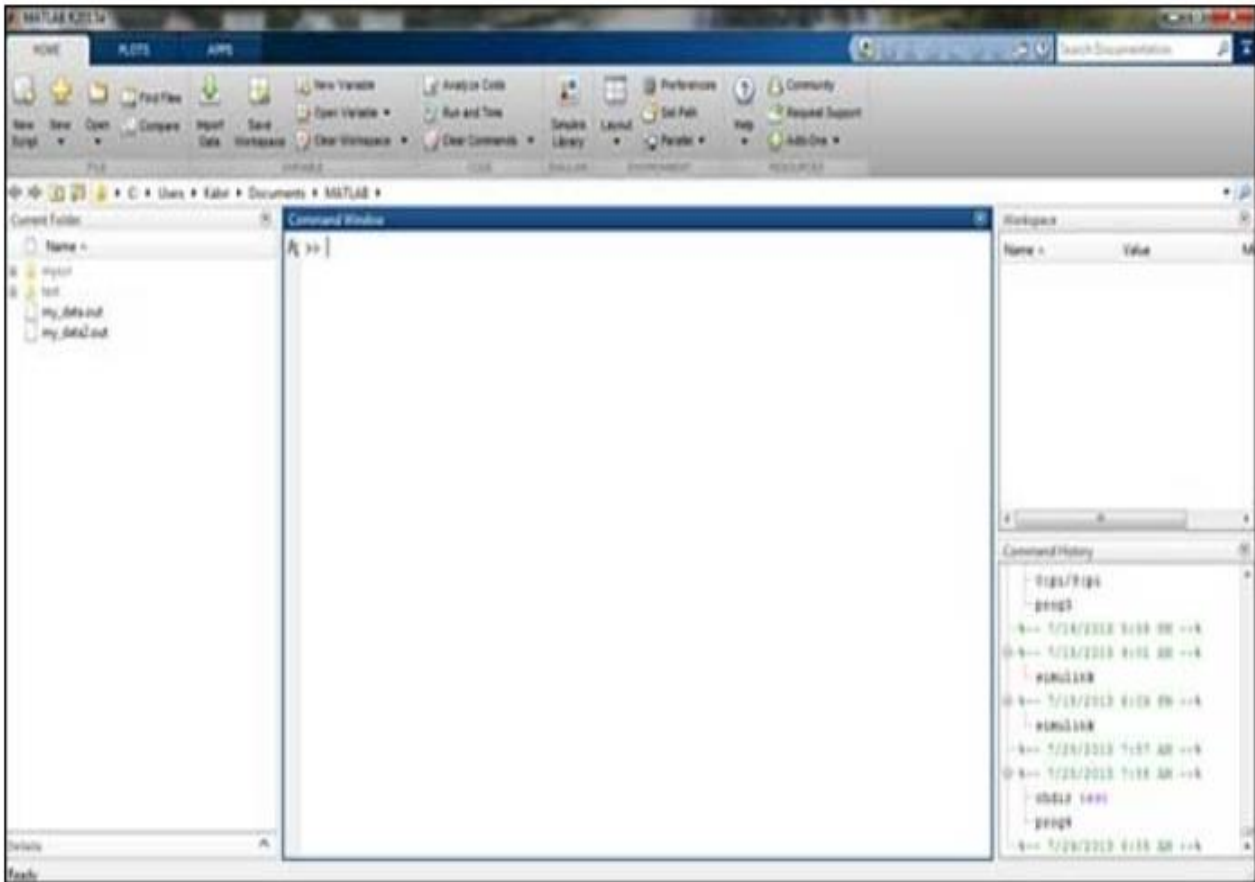
## **Local Environment Setup**

- Setting up MATLAB environment is a matter of few clicks. The installer can be downloaded from [here](#).
- MathWorks provides the licensed product, a trial version and a student version as well. You need to log into the site and wait a little for their approval.
- After downloading the installer the software can be installed through few clicks.



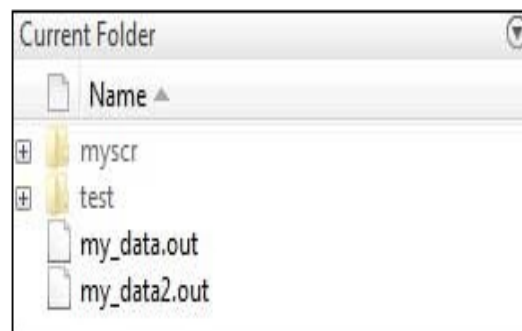
## Understanding the MATLAB Environment

MATLAB development IDE can be launched from the icon created on the desktop. The main working window in MATLAB is called the desktop. When MATLAB is started, the desktop appears in its default layout.

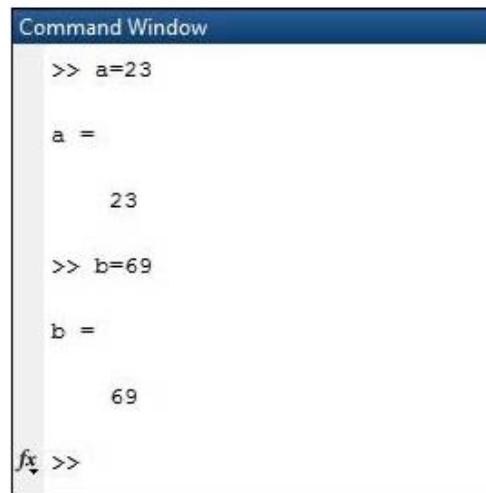


The desktop has the following panels –

- **Current Folder** – This panel allows you to access the project folders and files.

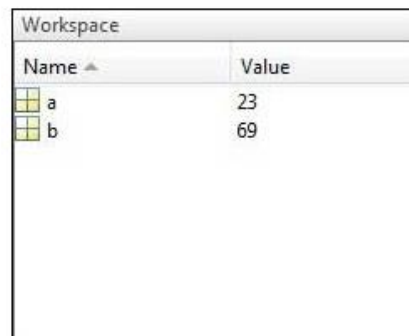


- **Command Window** – This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).



A screenshot of the MATLAB Command Window. The title bar is blue and labeled 'Command Window'. The window contains the following text: '>> a=23', 'a =', '23', '>> b=69', 'b =', '69', and at the bottom, a function icon followed by '>>'. The background is white with a light gray border.

- **Workspace** – The workspace shows all the variables created and/or imported from files.



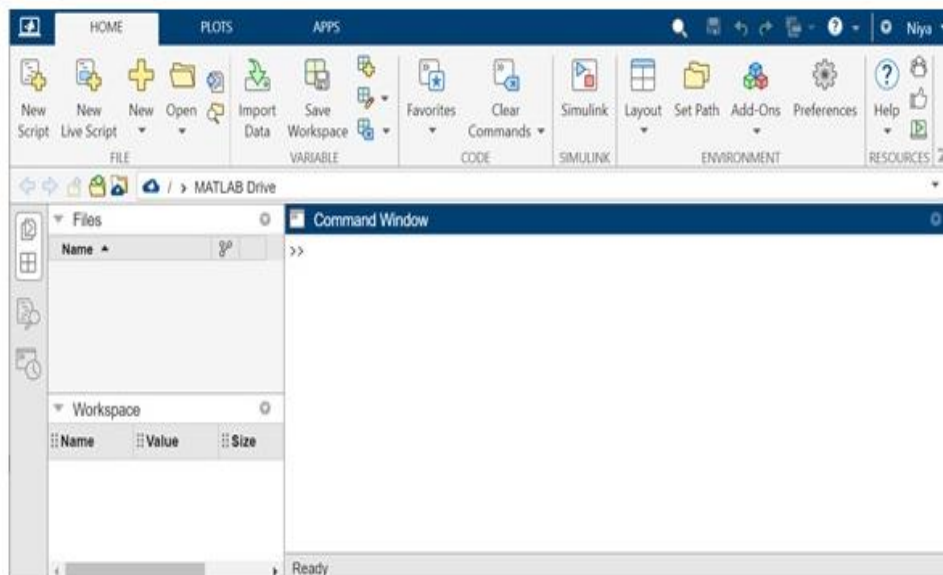
A screenshot of the MATLAB Workspace window. The title bar is labeled 'Workspace'. It contains a table with two columns: 'Name' and 'Value'. There are two rows of data: one for variable 'a' with value '23', and one for variable 'b' with value '69'. Each row has a small yellow square icon to the left of the variable name.

Name ▲	Value
a	23
b	69

- **Command History** – This panel shows or return commands that are entered at the command line.

```
%-- 7/14/2013 5:58 PM --%
%-- 7/15/2013 9:01 AM --%
|simulink
%-- 7/15/2013 6:09 PM --%
|simulink
%-- 7/25/2013 7:57 AM --%
%-- 7/25/2013 7:58 AM --%
|chdir test
|prog4
%-- 7/29/2013 8:55 AM --%
|a=23
|b=69
```

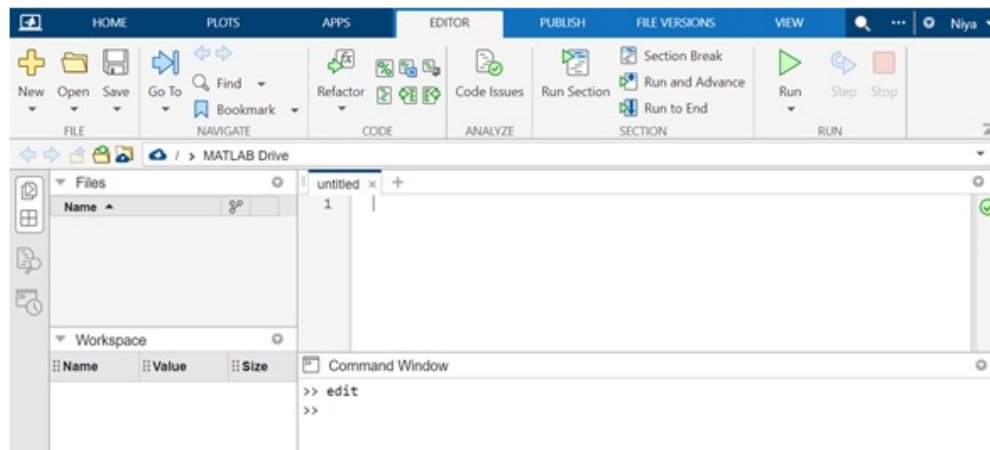
Matlab comes with a built-in powerful editor that helps to write your code and also to compile , debug and see the output. When you log in to Matlab this will be the screen that is displayed to you.



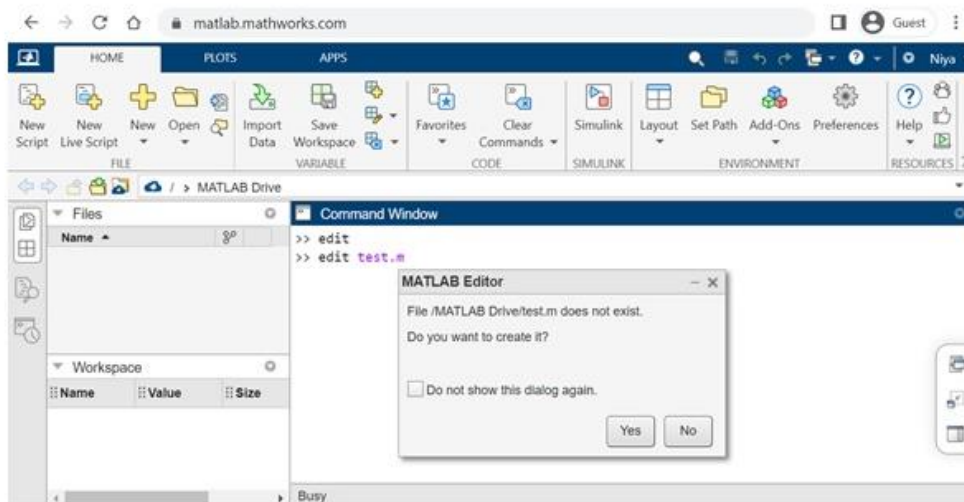
To open a editor you can type edit or type edit along with filename in command window.For example

```
>>edit  
>>edit test.m
```

The matlab files are saved with the .m extension. Let us first just type edit in the command window and see the editor opening.



Now, let us type edit test.m in the command window and see the editor opening.

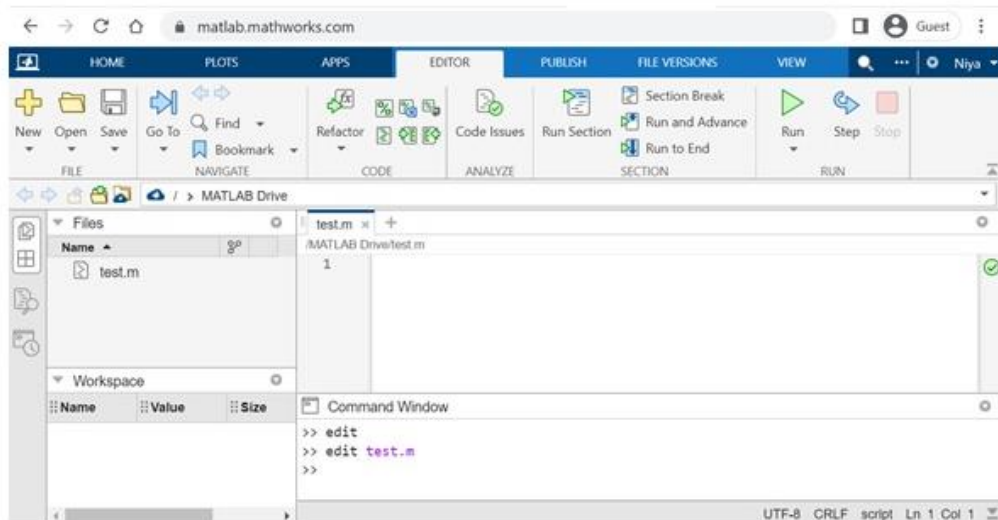


On enter you will see the dialog box asking to confirm since the file is new. The file test.m is not existing and will be created, if it exists it will be opened directly.

Click on Yes and it will open a new file test.m in the editor as shown below.



Click on Yes and it will open a new file test.m in the editor as shown below.

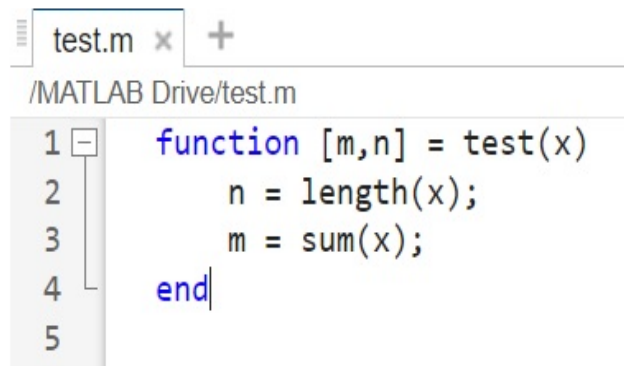


Now, let us write a small piece of code in the editor and run the same.

```
function [m,n] = test(x)
    n = length(x);
    m = sum(x);
end
```

The function name: test will take the vector as input and return the sum of the vector and the length of it.

Let us enter the function inside the editor test.m, as shown below.



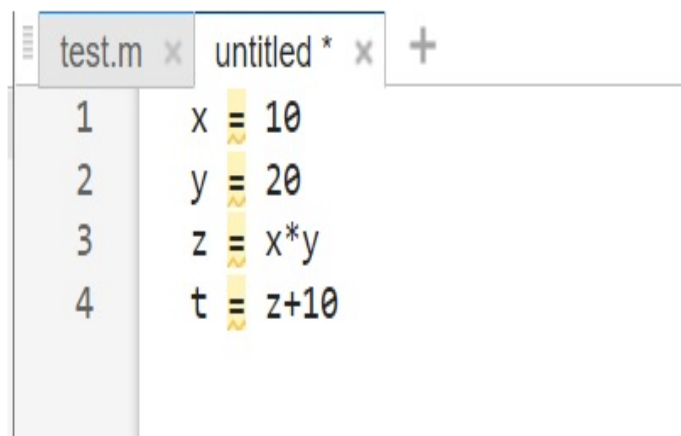


Save the changes and in the command line you can execute as shown below.

```
>> a = [10, 20, 30, 40]
a = 10    20    30    40
>> [m, l] = test(a)
m = 100
l = 4
>>
```

We are giving the vector  $a = [10, 20, 30, 40]$ , to the function test. It returns m and l where m is the sum of the vectors and l is the length.

Another thing to do in the editor is you can write the instruction inside the editor as shown below –



So we have the above instruction , let me select line no 1 to 3 and execute it as shown below.



```
>> x = 10
y = 20
z = x*y
x = 10
y = 20
z = 200
>>
```

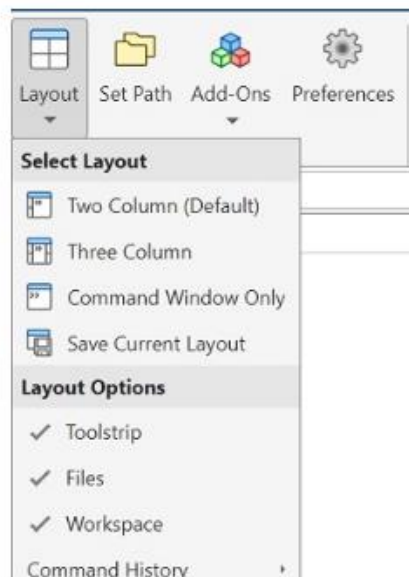
## Opening of Workspace Browser in Matlab

Open matlab, most of the time workspace is by default selected as part of the layout, in case you don't see it. Here are a few ways to open a workspace for you.

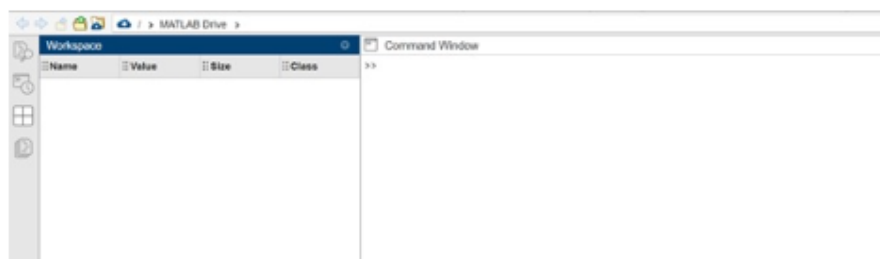
Go to Home tab, inside the Environment section click on layout as shown below –



Open layout and it should show the following details



Incase workspace is not selected for you , select it and you should see workspace as part of the layout as shown below.



You can make use of the command: workspace in the matlab command window and once it executes it will open the workspace.

```
>>workspace
```

## Creating and Editing Variables in Workspace

Let us create a few variables in the command window and see the same being stored in the workspace.

Below we have created a matrix A in command window

```
>> A = [1 2 3; 4 5 6; 7 8 9]
```

```
A =
```

```
1    2    3
4    5    6
7    8    9
```

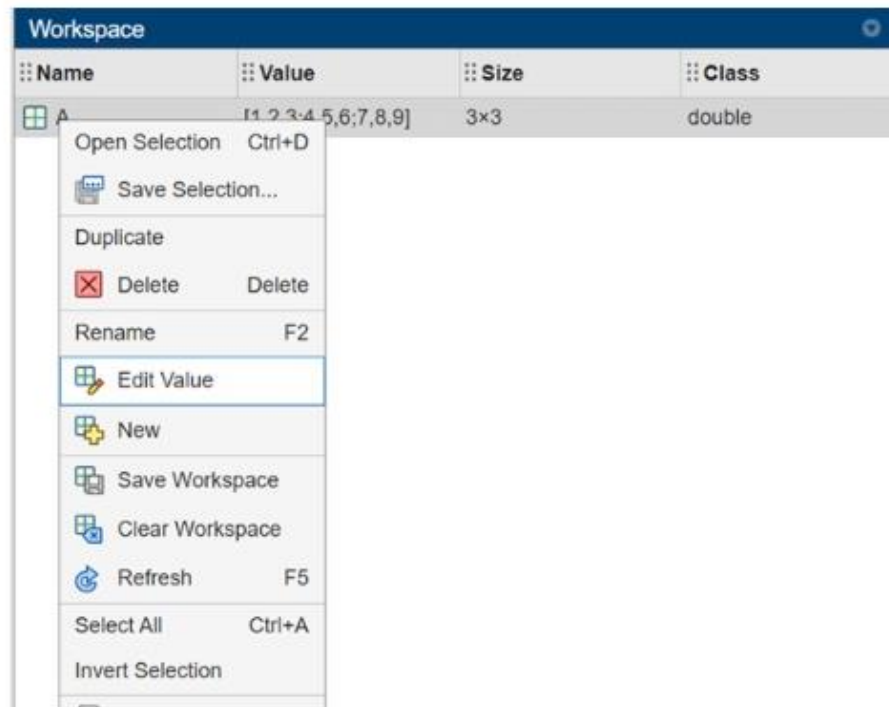
```
>>
```

Once it is executed you will see variable A in workspace.

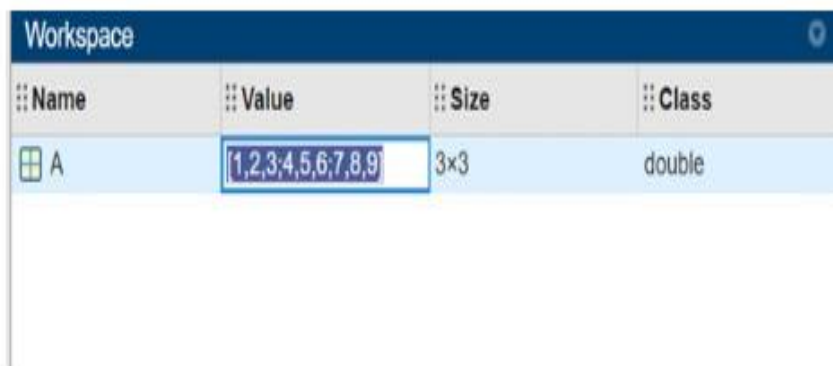
Workspace				Command Window
Name	Value	Size	Class	>> A = [1 2 3; 4 5 6; 7 8 9]
A	[1,2,3;4,5,6;7,8,9]	3x3	double	A =  1    2    3 4    5    6 7    8    9  >>

The workspace stored the Name, value, size and class of the variable A created.

To edit the value of the variable A in workspace, simply right click and select Edit Value as shown below.



On you click on Edit Value you will see you can edit the value as shown below –



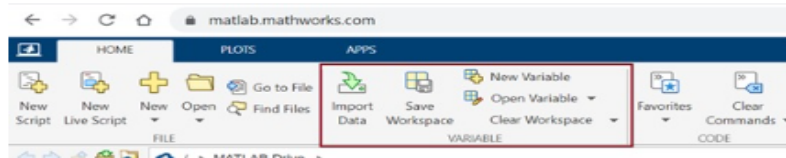
## Save and Load Workspace Variables

The workspace is cleared when you exit from matlab. But you can save the workspace in case you want to use it in future.

The file is saved with the .mat extension. When you want it next, open the file in your matlab session.

Here are a few ways to save the workspace variables.

Go to Home tab and you should see the VARIABLE section as shown below

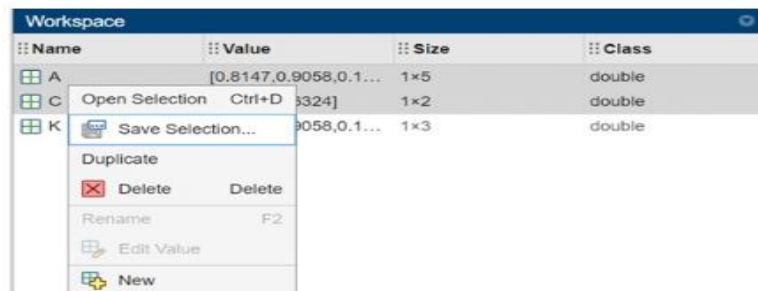


Click on Save Workspace. It will open the file save dialog window as shown below



Save the file with the .mat extension

You can also select a subset of variables in the workspace , right click and click on save selection as shown below –



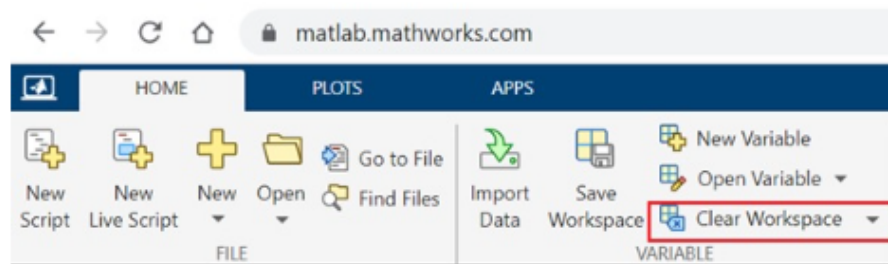
Another way to save the full workspace is to right-click on the workspace and click on save workspace as shown below.

## Clearing Workspace

With clear workspace it will remove all the variables present in the workspace. By default when the matlab session is closed the workspace is cleared.

But if you still want to clear the workspace when the session is active, here are a few ways to do it.

Inside Home tab, you can make use of clear workspace as shown below.



You can also make use of the command `clear` inside the matlab command window to clear and remove all items from workspace.

```
clear // removes all the variables present in the workspace

clear A // will only remove the variable A from the workspace

clear -regexp ^test ^a; // makes use of regular expression and removes all the
variables that start with test and a.
```

## Using clearvars Command

This method will remove all the variables present in the workspace. If you specify the variable names only those are removed.

### Example

```
clearvars
clearvars A B C
```



