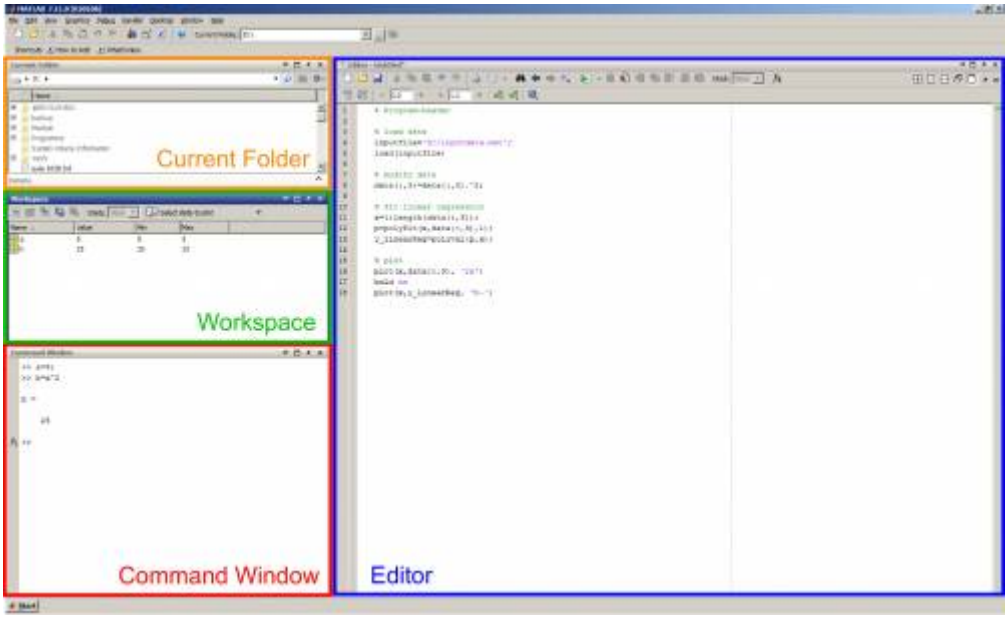
Matlab Crash Course

The Matlab window is shown in the Figure below. The different windows can be moved and resized as desired - they can even be separate windows. There is the window *Current Folder* where the content of the current directory (selectable in the dropdown menu above) is shown, similar to the Microsoft Explorer. In the *Workspace* there are all variables available right now. So when you assign the value 5 to *a*, it appears in that window. All those commands are written in the *Command window*. The *Editor* is used for all kinds of programming. The code in there can be run by pressing F5 or clicking on the `Save and run (F5)` button.

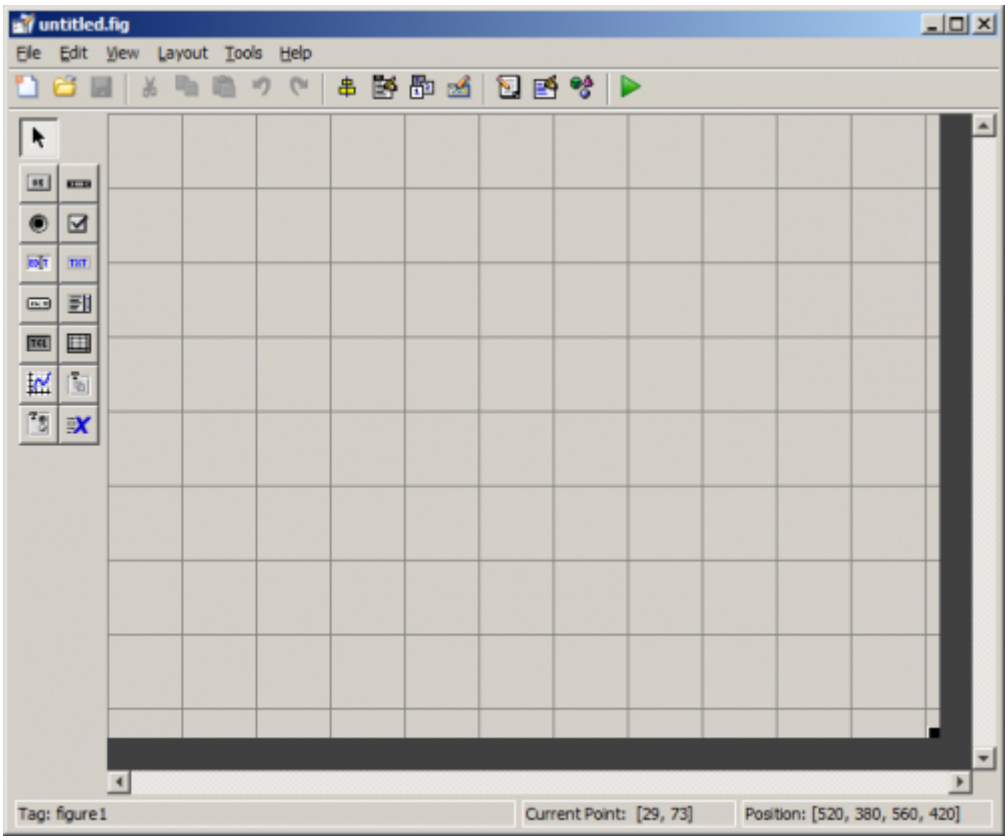


Some important example commands are shown in the following Table:

Command	**Explanation**
a=5;	Assign 5 to the variable a (semicolon supresses command window output)
a=[1,2,3,4,5];	Define a row vector (equal: a=1:5;)
a=[1,2,3,4,5]';	Define a column vector (' transposes a matrix)
b=rand(20,30);	Define 20×30 matrix with random numbers between 0 and 1
a=b(3:5,4:10);	Take row 3,4,5 and columns 4 to 10 from matrix b and assign it to a (a will be a 3×7 matrix)
for k=1:size(a,2) c(k)=a(1,k)*4; end	For-loop from 1 to 7 where values of a (row 1) are multiplied by 4 and assigned to c
c=a(1,:)*4;	The same operation as above but more efficient
x=0:0.01:4*pi;~y=sin(x);	Create vector x from 0 to 4π with increment 0.01 and a vector y containing the corresponding sine values
plot(x,y,'-.bx');	Plots x versus y with a blue (b), dash-dot (-.) line and crosses (x) as markers
save('../vars/myVars.mat', 'a', 'b', 'c')	Saves the variables a, b and c to a .mat file in ../vars/
load('../vars/myVars.mat')	Restores variables a, b and c to Workspace

Interface (GUIDE)

The Matlab GUIDE is a drag-and-drop program for creating graphical user interfaces. It can be opened by typing `guide` in the command window and hit Return; a screenshot of an empty GUI is shown in the Figure below.



This interface consists of all arranged objects (buttons, textboxes, popupmenues, sliders, checkboxes,...) saved in a .fig files (e.g. myProgram.fig) and a corresponding .m files (same filename, e.g. myProgram.m) containing all callback functions.

Each object (e.g. button) has an unique identifiable name (Tag) which can be seen in the Property Inspector (double click on the object) under 'Tag'. If a button is called 'button1', there exists (automatically created) a function in *myProgram.m* called *button1_Callback*. This function is called when the button is clicked.

There exist also other functions than callback functions, depending on what has been done in the interface (e.g. button release, delete, resize function, selection change in a button group, ...)

The state of the interface is stored in one matlab structure, called *handles*. The fields of this structure are object handles, one for each object (e.g. *handles.button1*). This field contains all properties (e.g. size, values, functions) which can be changed or retrieved by `set` and `get`.

In this structure also user data can be stored since this variable is available in all functions (*handles* is an input argument to all callback functions). Example: User wants to load a number from a textfile. This number can be saved in *handles.userNumber*. The handles structure has to be updated at the end of each function when content of it has been changed:

```
guidata(hObject, handles);.
```