

# Computer Programming: Starting with MATLAB

Asst. Prof. Dr. Yalçın İşler  
Izmir Katip Celebi University

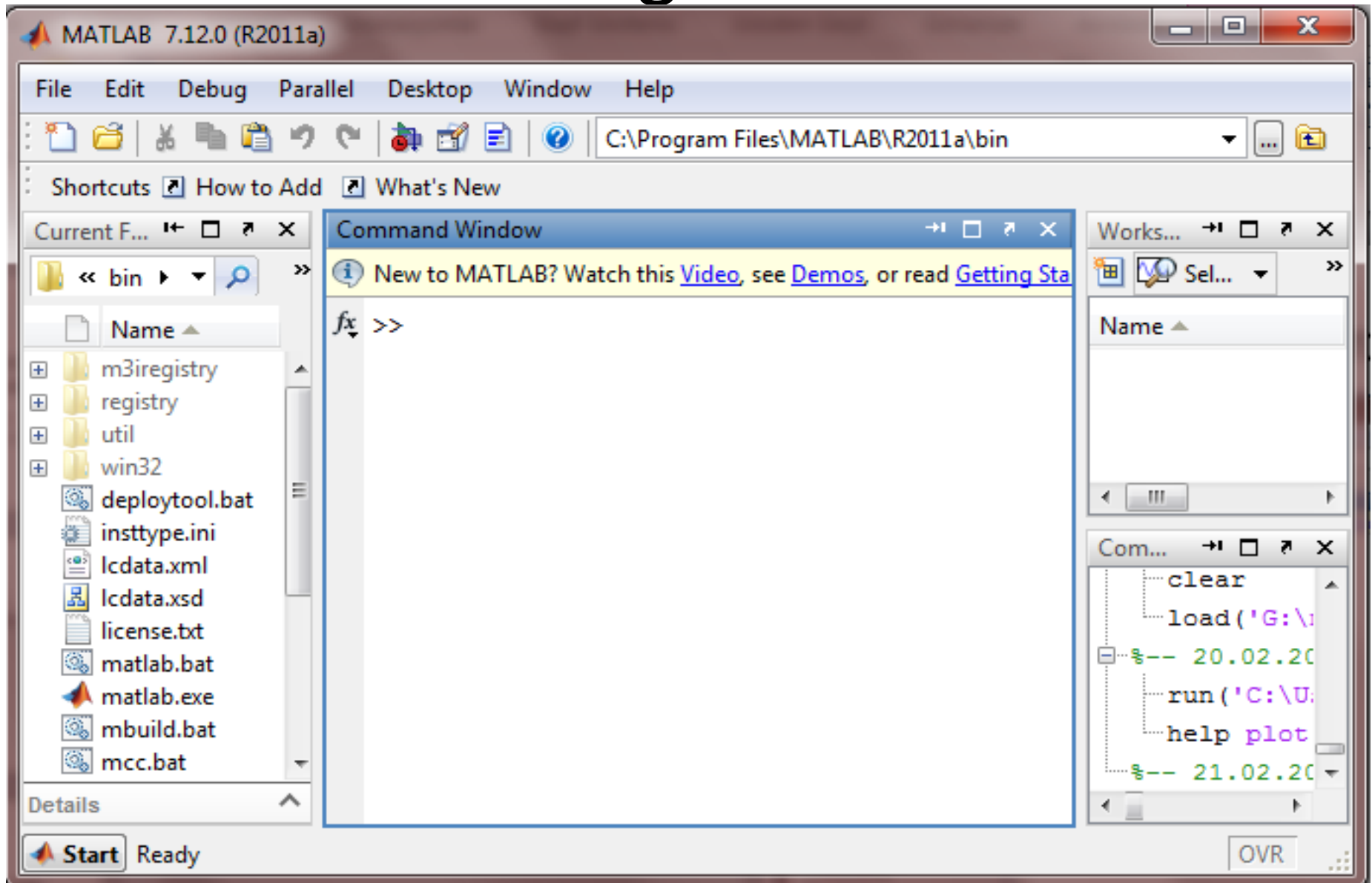
# Outline

- Matlab Environment
- Working in the Command Window
- Using MATLAB as a Calculator
- Display Formats
- Elementary Functions
- Variables
- Script Files
- Examples

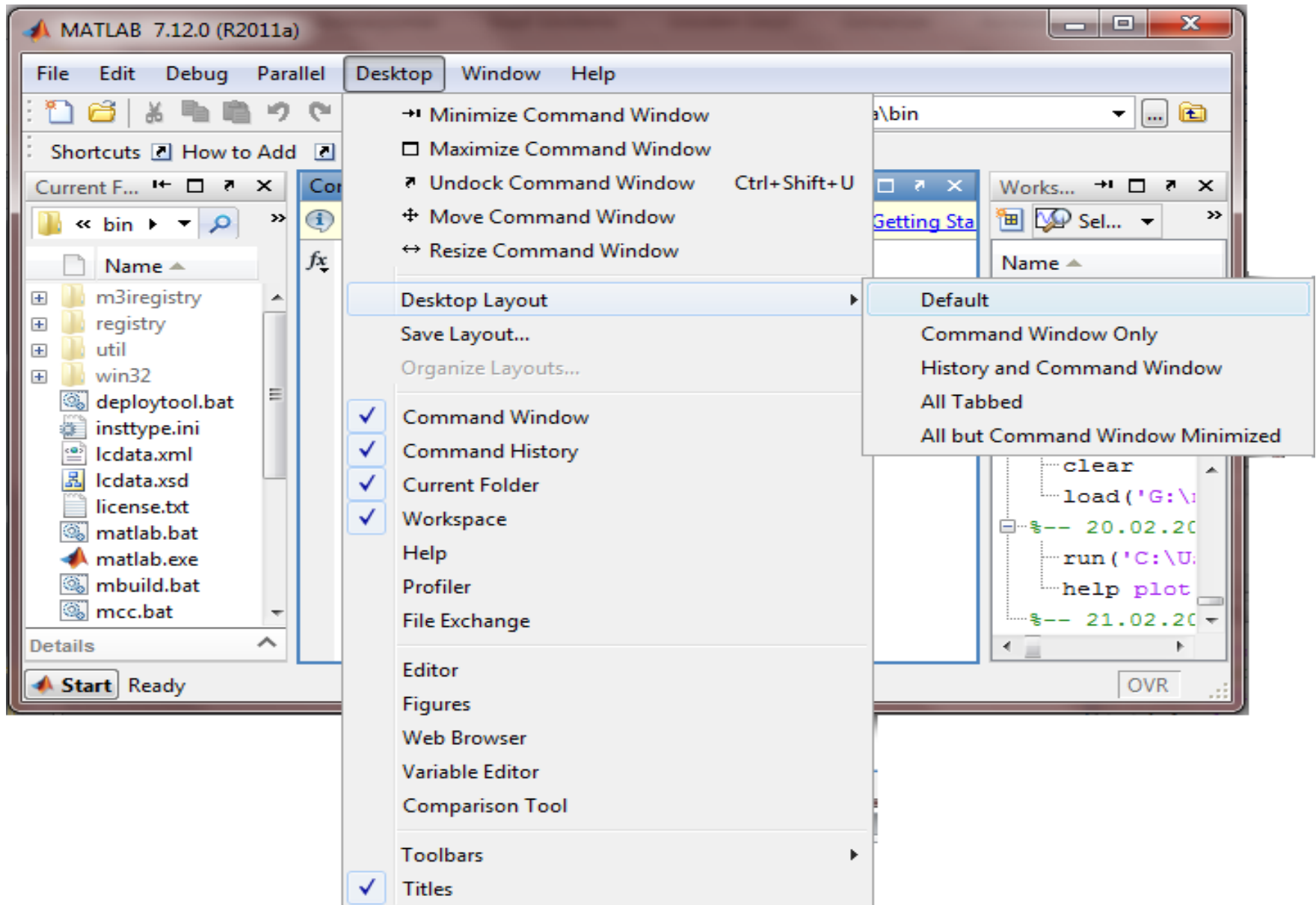
# MATLAB

- MATrix LABoratory
- Widely used in universities and colleges
- Powerful environment for technical computing
- No need having prior experience in programming
- Based on Numerical analysis
- Symbolic calculations

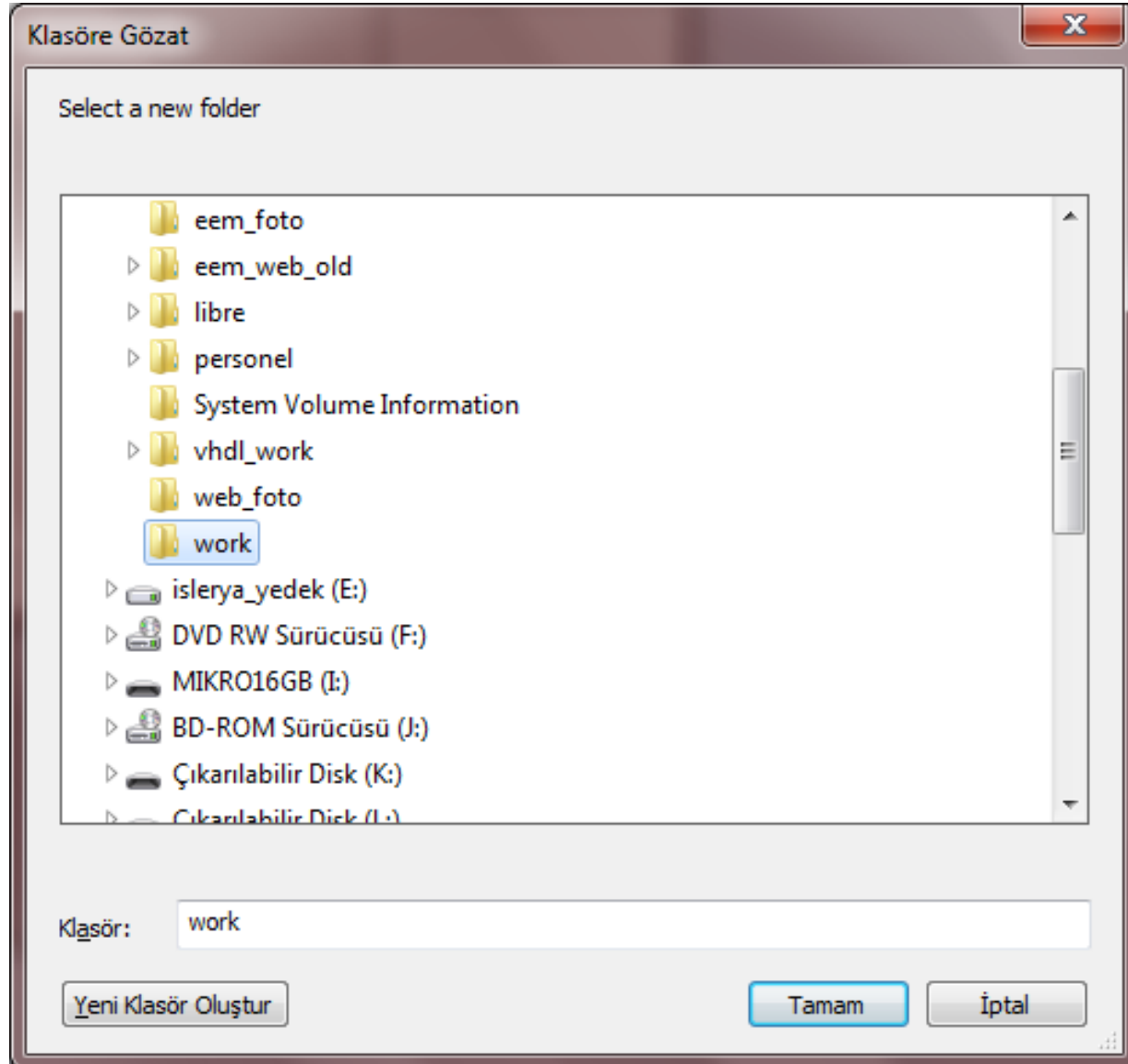
# Starting MATLAB



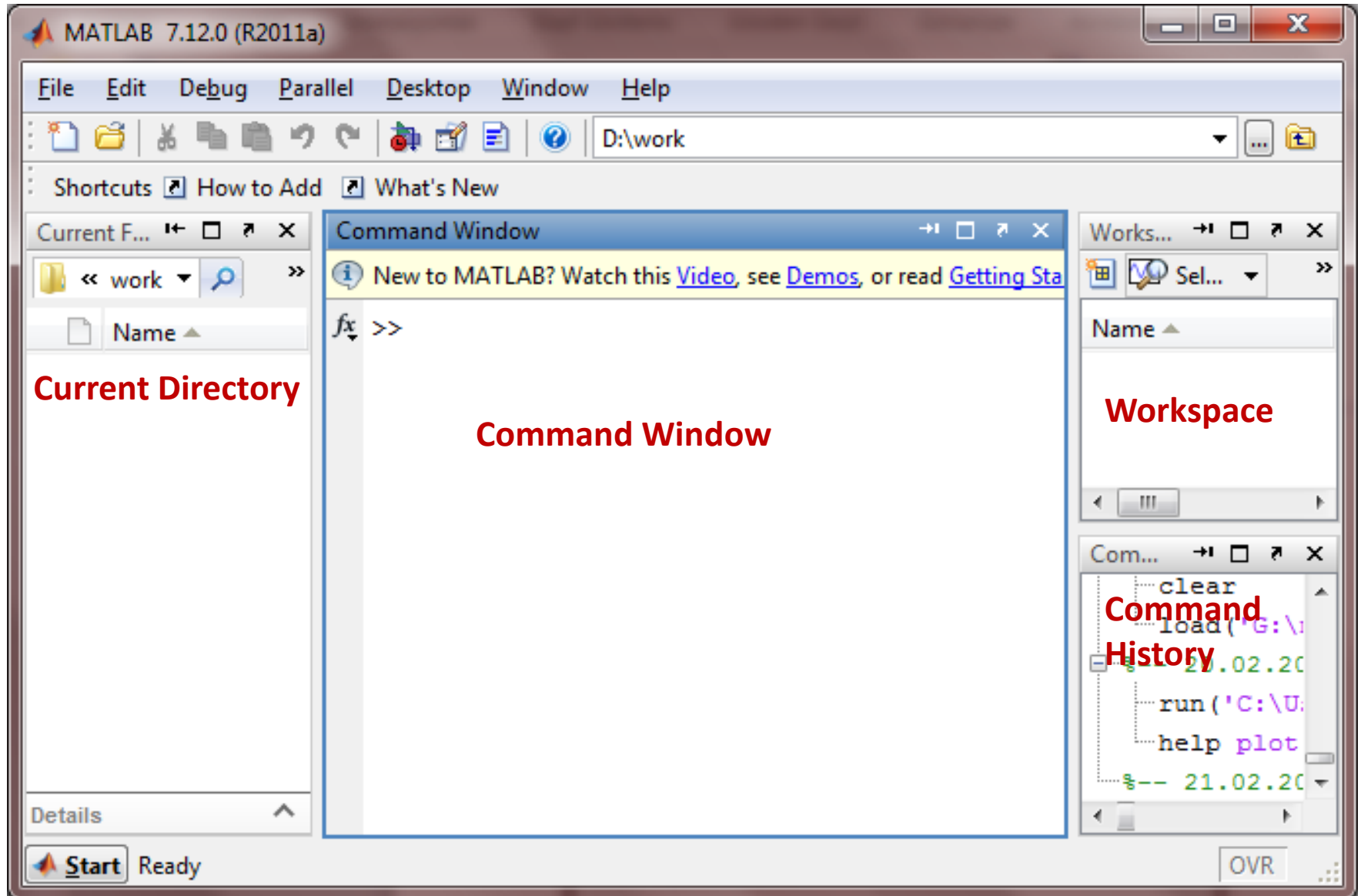
# Choosing Desktop Layout



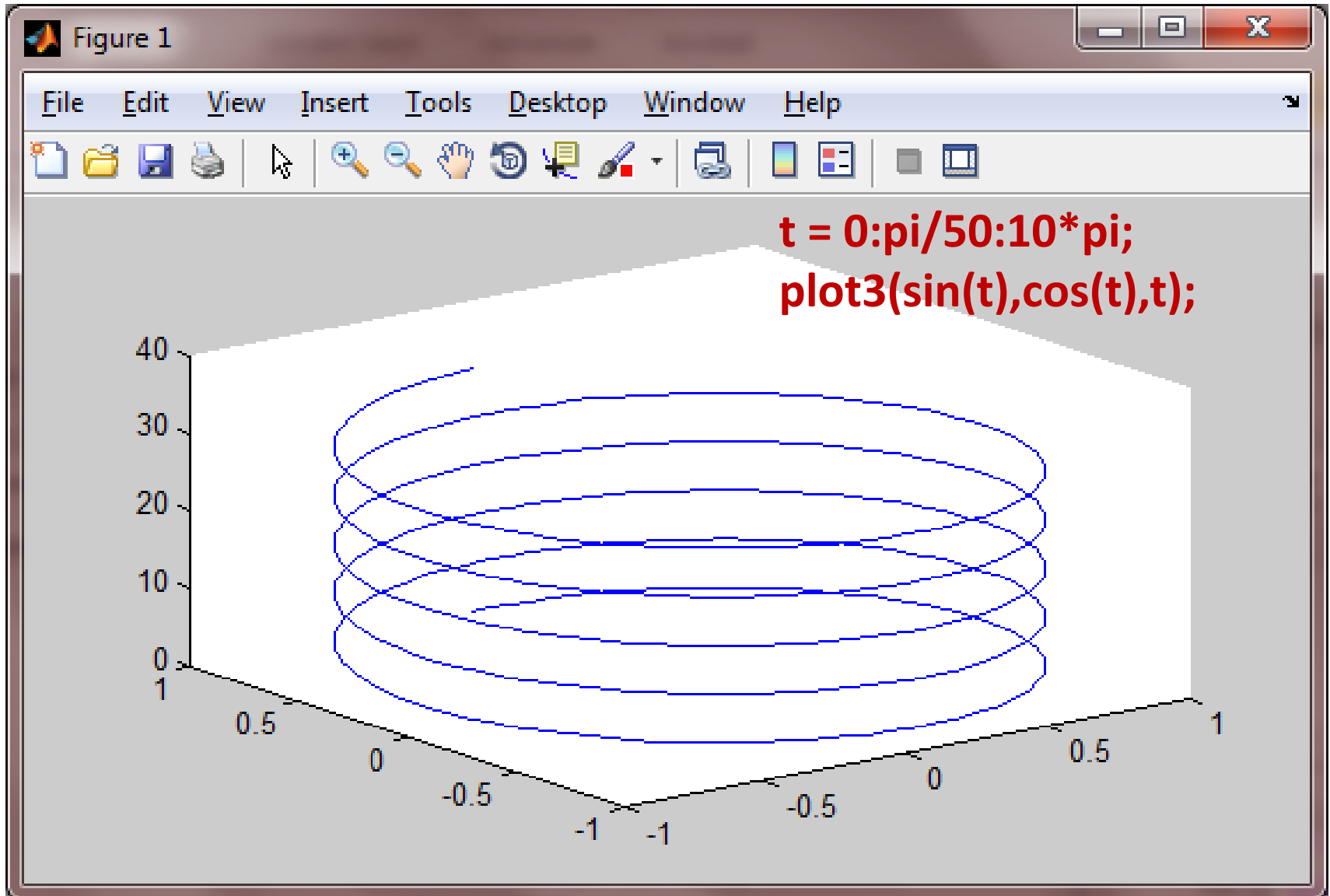
# Changing Current Folder



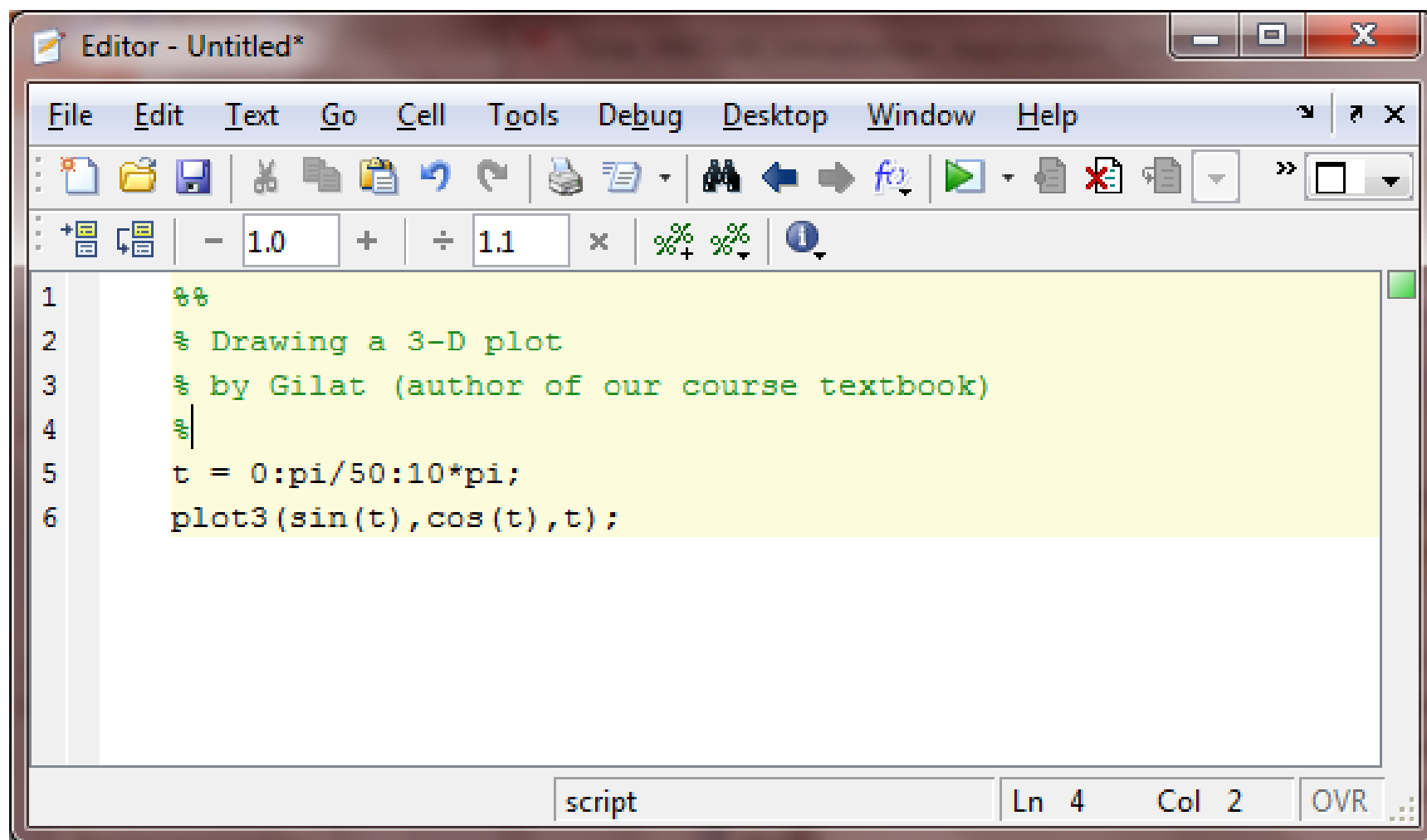
# MATLAB Environment



# Figure Window



# Editor Window



# Working with Command Window

The screenshot shows the MATLAB 7.12.0 (R2011a) interface. The Command Window is the central focus, displaying the following commands and outputs:

```
>> 2*4+3  
ans =  
    11  
  
>> 2*4+3;  
>> |
```

Annotations in red text with blue arrows point to specific parts of the Command Window:

- Write a command and the press Enter key** points to the first command `>> 2*4+3`.
- Answer will be shown (ans means answer)** points to the output `ans = 11`.
- Write a command with ending by a semicolon and the press Enter key** points to the second command `>> 2*4+3;`.
- Answer will NOT be shown due to the semicolon** points to the second command `>> 2*4+3;`.
- >> means that Matlab is ready for a new command** points to the prompt `>>` on the line following the semicolon command.

The Command Window also displays a message: "New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)".

The Command Window also shows a list of commands in the Command History window:

```
plot3(sin  
clc  
clear all  
clc  
2*4+3  
2*4+3;
```

# Running Command

- After running a command (writing the command, and pressing the Enter-key), you cannot change it. To correct the command, you should give a new correct command 😊
- A previous command can be seen, by pressing Up-Arrow key. Before pressing Enter key, you can move the cursor by using Left- and Right-Arrow keys, and then you can make changes.

# Some Notes

- If a semicolon (;) is typed at the end of a command, the answer will not be shown, but the command will be executed, of course.
- No matter which command is written after the percent symbol (%), it will not be executed, which is called commenting.
- `clc`: Clear the command window

# Arithmetic Operations

Operation	Symbol	Example
Addition	+	3+2
Subtraction	-	3-2
Multiplication	*	3*2
Division (right division)	/	3/2
Left division	\	3\2
Exponentiation	^	3^2

$x \setminus y$  equals to  $y / x$  for scalars.

Precedence	Operation
1	( )
2	^
3	* /\
4	+ -

if precedences are equal, the left-side operation will be executed first.

# Display formats

Try >> 290 / 7 after

Command	Description	The example output
format short	Fixed-point with 4-decimal digits between 0.001 and 1000, otherwise «long e» format.	41.4286
format long	Fixed-point with 14-decimal digits between 0.001 and 1000, otherwise «long e» format.	41.42857142857143
format short e	Scientific notation with 4 decimal digits	4.1429e+001
format long e	Scientific notation with 15 decimal digits	4.142857142857143e+001
format short g	Best of 5-digit fixed or floating points	41.429
format long g	Best of 15-digit fixed or floating points	41.4285714285714
format bank	Two decimal points	41.43
format compact	Eliminates empty lines	
format loose	Adds empty lines (opposite of compact)	

# Elementary Functions

Function	Description	Example
sqrt(x)	Square root	sqrt(81)
nthroot(x,n)	Real n-th root of a real number x	nthroot(80,5)
exp(x)	$e^x$	exp(5)
abs(x)	Absolute value	abs(-24)
log(x)	Natural logarithm (base e = ln)	log(1000)
log10(x)	Base 10 logarithm	log10(1000)
factorial(x)	$x!$	factorial(5)

# Trigonometric Functions

Function	Description
$\sin(x)$ $\text{sind}(x)$ $\text{asin}(x)$ $\text{asind}(x)$	Sine and inverse sine
$\cos(x)$ $\text{cosd}(x)$ $\text{acos}(x)$ $\text{acosd}(x)$	Cosine and inverse cosine
$\tan(x)$ $\text{tand}(x)$ $\text{atan}(x)$ $\text{atand}(x)$	Tangent and inverse tangent
$\cot(x)$ $\text{cotd}(x)$ $\text{acot}(x)$ $\text{acotd}(x)$	Cotangent and inverse cotangent
$\pi$	$\pi$

Hyperbolic trigonometric functions are also present.

# Rounding Functions

Function	Description	Example
<code>round(x)</code>	Rounds to the nearest integer	<code>round(17/5) → 3</code>
<code>fix(x)</code>	Rounds towards zero	<code>fix(13/5) → 2</code>
<code>ceil(x)</code>	Rounds towards infinity	<code>ceil(11/5) → 3</code>
<code>floor(x)</code>	Rounds towards minus infinity	<code>floor(-9/4) → -3</code>
<code>rem(x, y)</code>	Returns the remainder after x is divided by y	<code>rem(17,5) → 2</code>
<code>sign(x)</code>	Signum function, returns 1 if $x > 0$ , -1 if $x < 0$ , and 0 if $x = 0$	<code>sign(17) → 1</code>

# Variables

- A variable, is a name made of a letter or a combination of several small and/or capital letters, digits, and underscore character (\_), contains a numerical value.
- A variable name must begin with a letter (Turkish characters and spaces are NOT allowed)
- Case-sensitive: AA, Aa, aA, and aa are all different.
- Avoid using the names of built-in function. If you prefer giving a function name to a variable, you can use the variable but the function.
- Forbidden to use keywords: *break, case, catch, continue, else, elseif, end, for, function, global, if, otherwise, persistent, return, switch, try, while*

# Assignment Operator

The image shows the MATLAB 7.12.0 (R2011a) interface. The Command Window displays the following code and output:

```
>> x=1  
  
x =  
  
    1  
  
>> y=2;  
>> (1+2)/3  
  
ans =  
  
    1  
  
>> x=1,y=2;(1+2)/3  
  
x =  
  
    1  
  
ans =  
  
    1
```

The Workspace window shows the following variables and their values:

Name	Value
ans	1
x	1
y	2

The Command History window shows the following commands:

```
%-- 21.02.2012 04:36 --%  
x=1  
y=2;  
(1+2)/3  
x=1,y=2;(1+2)/3
```

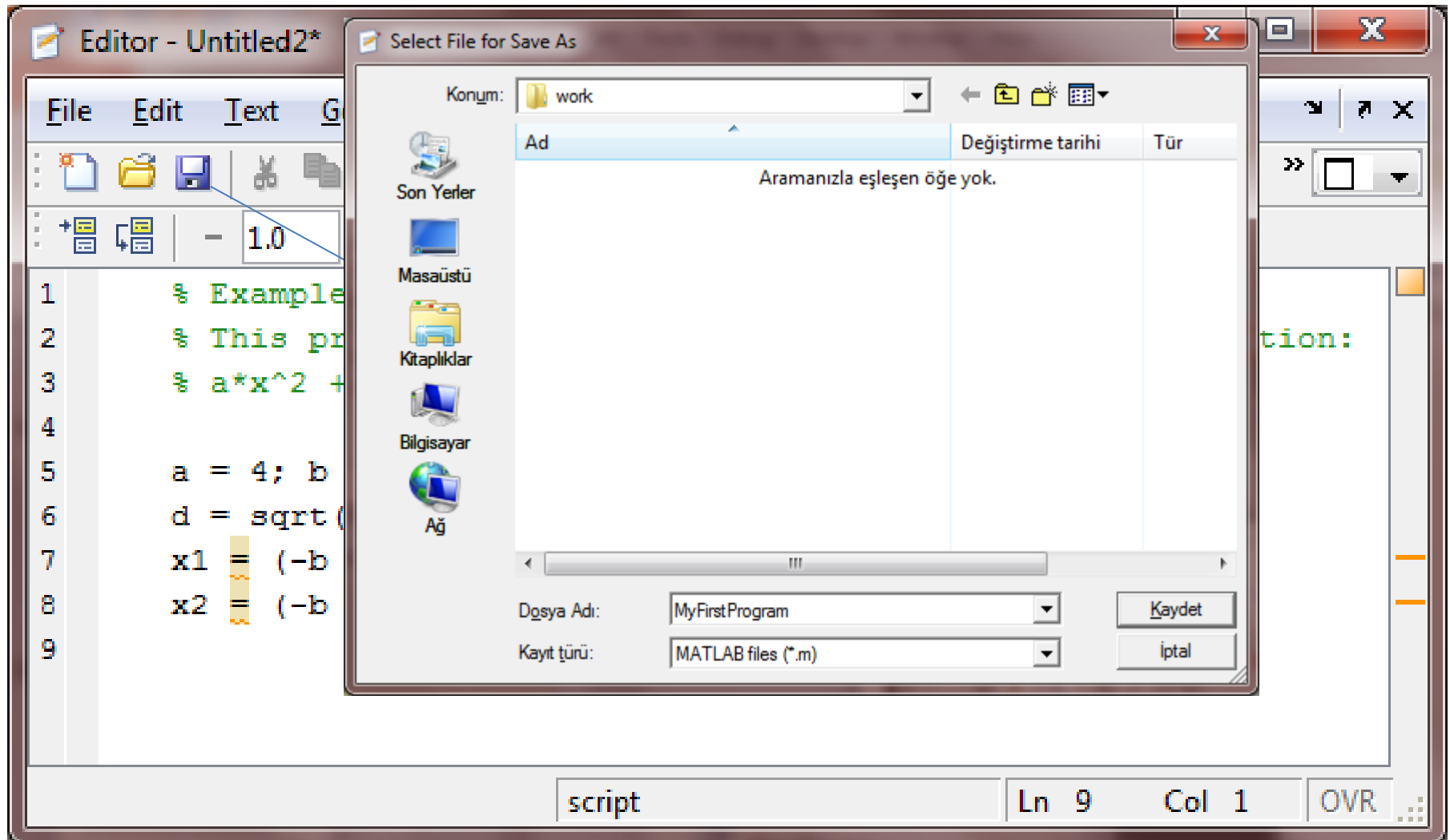
# Predefined Variables

- ans: A variable that has the value of the last expression that was not assigned to a specific variable.
- pi: The number of  $\pi$ .
- eps: The smallest difference between two numbers that Matlab can recognize.
- inf: Used for infinity.
- i: Defined as  $\sqrt{-1}$  (to show imaginary part of complex numbers)
- j: Same as i.
- NaN: Not a Number (for example, 0/0).

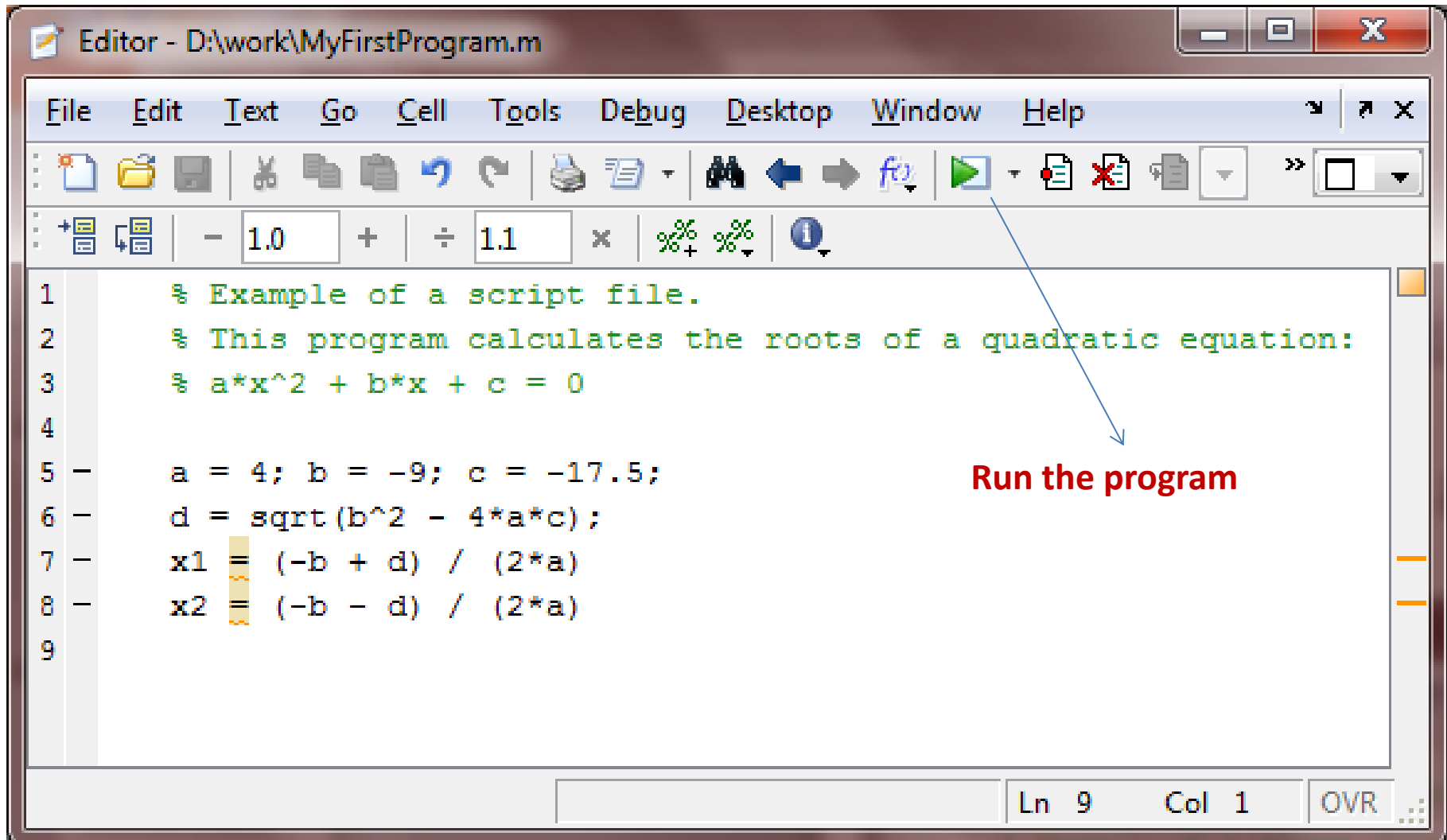
# Script Files

- A script file is a sequence of MATLAB commands, also called a program during all remaining lecture sessions of this course.
- When a script file runs (is executed), MATLAB executes the commands in the order as if they are typed in the Command Window.
- Using a script file is convenient because it can be edited and executed many times even after turn off and turn on the computer.
- Script files can be typed and edited in any text editor (preferably Matlab Editor).
- Script files are also called M-files because the extension .m is used when they are saved.

# Creating and Saving a Script File



# Running a Script File



Editor - D:\work\MyFirstProgram.m

File Edit Text Go Cell Tools Debug Desktop Window Help

1 % Example of a script file.  
2 % This program calculates the roots of a quadratic equation:  
3 %  $a*x^2 + b*x + c = 0$   
4  
5 - a = 4; b = -9; c = -17.5;  
6 - d = sqrt(b^2 - 4\*a\*c);  
7 - x1 = (-b + d) / (2\*a)  
8 - x2 = (-b - d) / (2\*a)  
9

Run the program

Ln 9 Col 1 OVR

# Running an Already-Saved Script

The screenshot shows the MATLAB 7.12.0 (R2011a) environment. The Command Window displays the execution of the script `MyFirstProgram`. The output shows the assignment of variables `x1` and `x2`. A blue arrow points from the text `MyFirstProgram` in the Command Window to the red text **Run the program by typing its name**. The Workspace window shows the current state of variables: `a` (4), `b` (-9), `c` (-17.5000), `d` (19), `x1` (3.5000), and `x2` (-1.2500). The Command History window shows the sequence of commands executed, including `MyFirstProgram`.

```
>> MyFirstProgram

x1 =

    3.5000

x2 =

   -1.2500

fx >>
```

Name	Value
a	4
b	-9
c	-17.5000
d	19
x1	3.5000
x2	-1.2500

```
y=2;
(1+2)/3
x=1,y=2;(1+2)/3
clc
MyFirstProgram
clear
clc
MyFirstProgram
```

# Laboratory Session

Do sample applications in Chapter 1 of the  
textbook.

# Homework #3

Not later than the next week:

Solve problems 3, 4, 7, 12, 13, 16, 19, 20, 25, and 26 from the Chapter 1 of the textbook using Matlab.