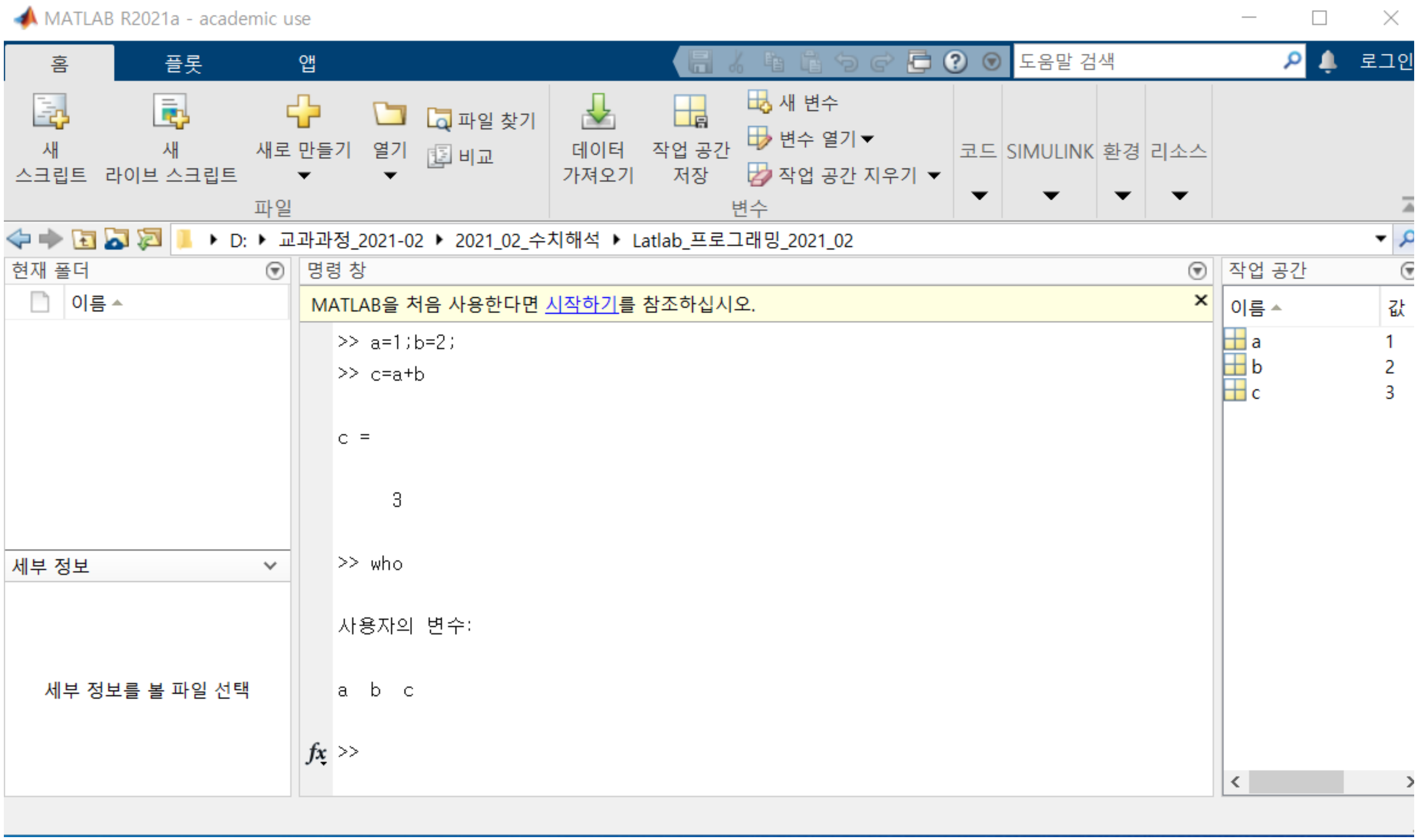
MATLAB 기초 사용법 연습

Prof. Chang-Joo Kim

MATLAB 윈도

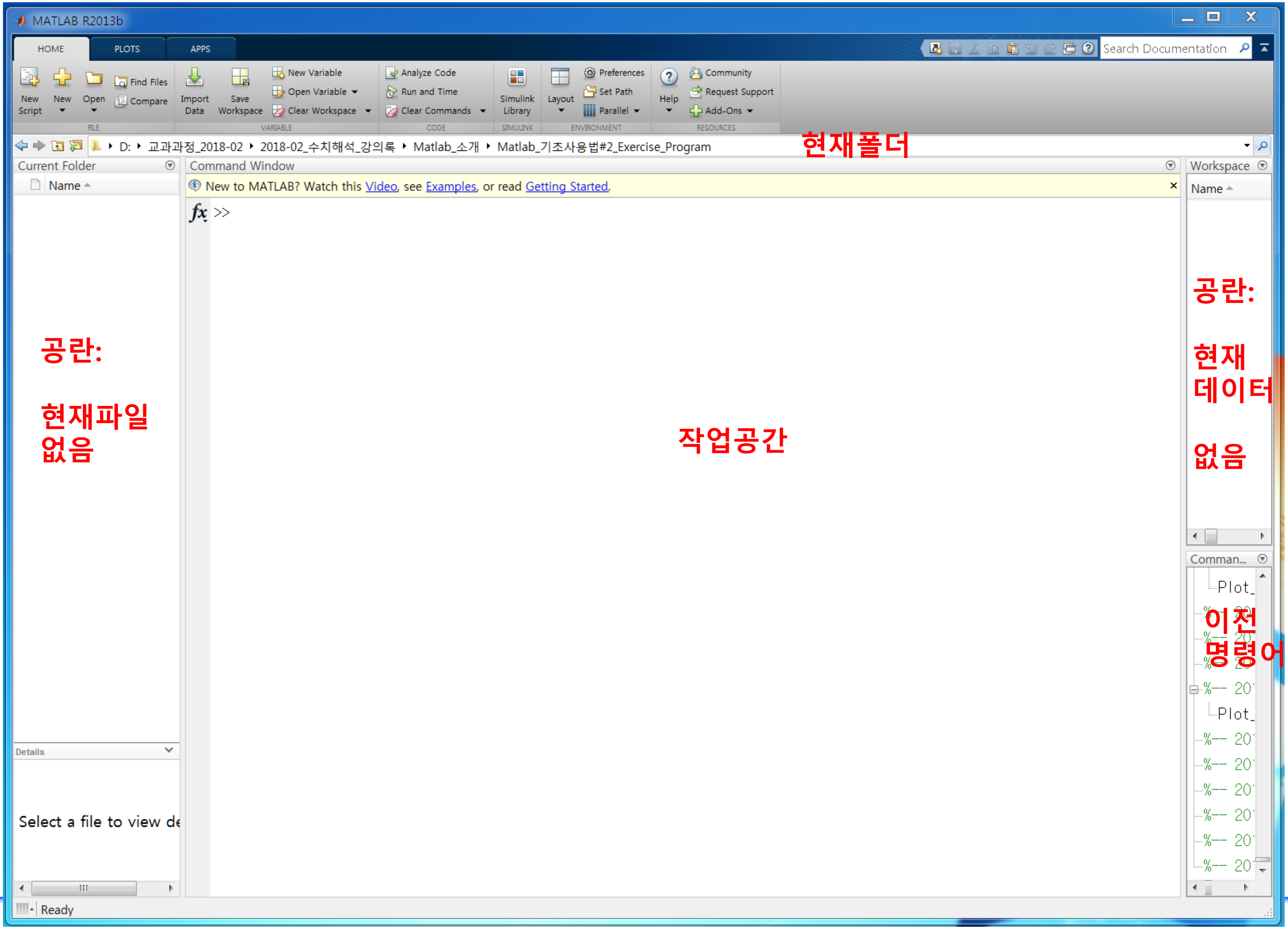


MATLAB을 처음 사용한다면 시작하기를 참조하십시오.

```
>> a=1;b=2;
>> c=a+b

c =

     3

>> who

사용자의 변수:

a  b  c

>>
```

이름	값
a	1
b	2
c	3

초기 화면 : 현재 폴더와 파일



D: ▸ 교과과정_2018-02 ▸ 2018-02_수치해석_강의록 ▸ Matlab_소개 ▸ Matlab_기초사용법#2_Exercise_Program

New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
fx >>
```

Select a file to view de

Ready

현재폴더

공란:

현재파일 없음

작업공간

공란:

현재 데이터

없음

이전 명령어

데이터 입력 (1): 실수 데이터 화면 입력

데이터 입력

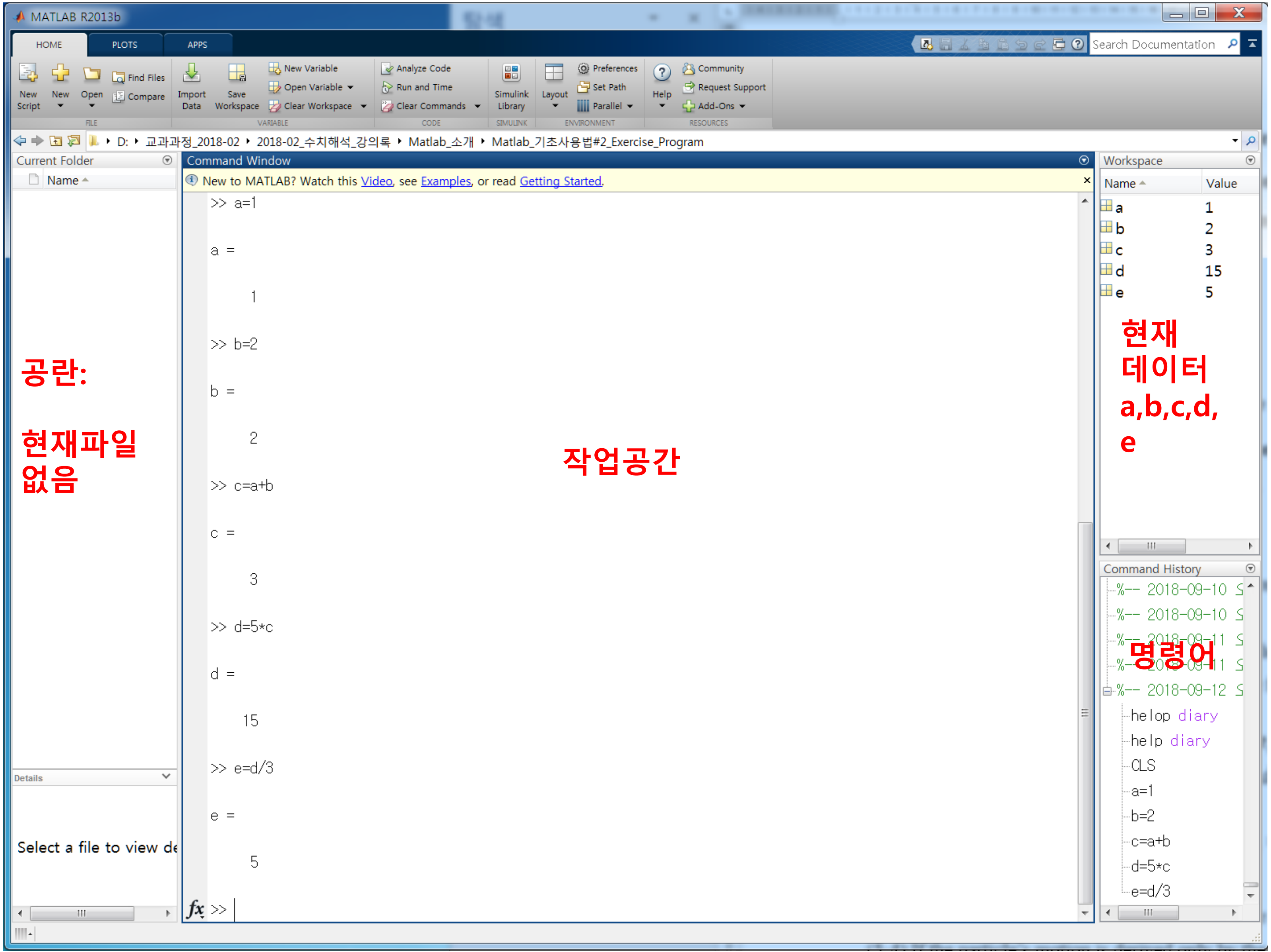
$$a = 1$$

$$b = 2$$

$$c = a + b$$

$$d = 5c$$

$$e = \frac{d}{3}$$



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> a=1

a =

     1

>> b=2

b =

     2

>> c=a+b

c =

     3

>> d=5*c

d =

    15

>> e=d/3

e =

     5

fx >> 
```

Name	Value
a	1
b	2
c	3
d	15
e	5

```
%-- 2018-09-10 
%-- 2018-09-10 
%-- 2018-09-11 
%-- 2018-09-11 
%-- 2018-09-12 
helop diary
help diary
CLS
a=1
b=2
c=a+b
d=5*c
e=d/3
```

공란:

현재파일 없음

작업공간

현재 데이터 a,b,c,d, e

명령어

데이터 입력 (2) : 연산결과를 화면에 출력하지 않음 (;)

데이터 입력

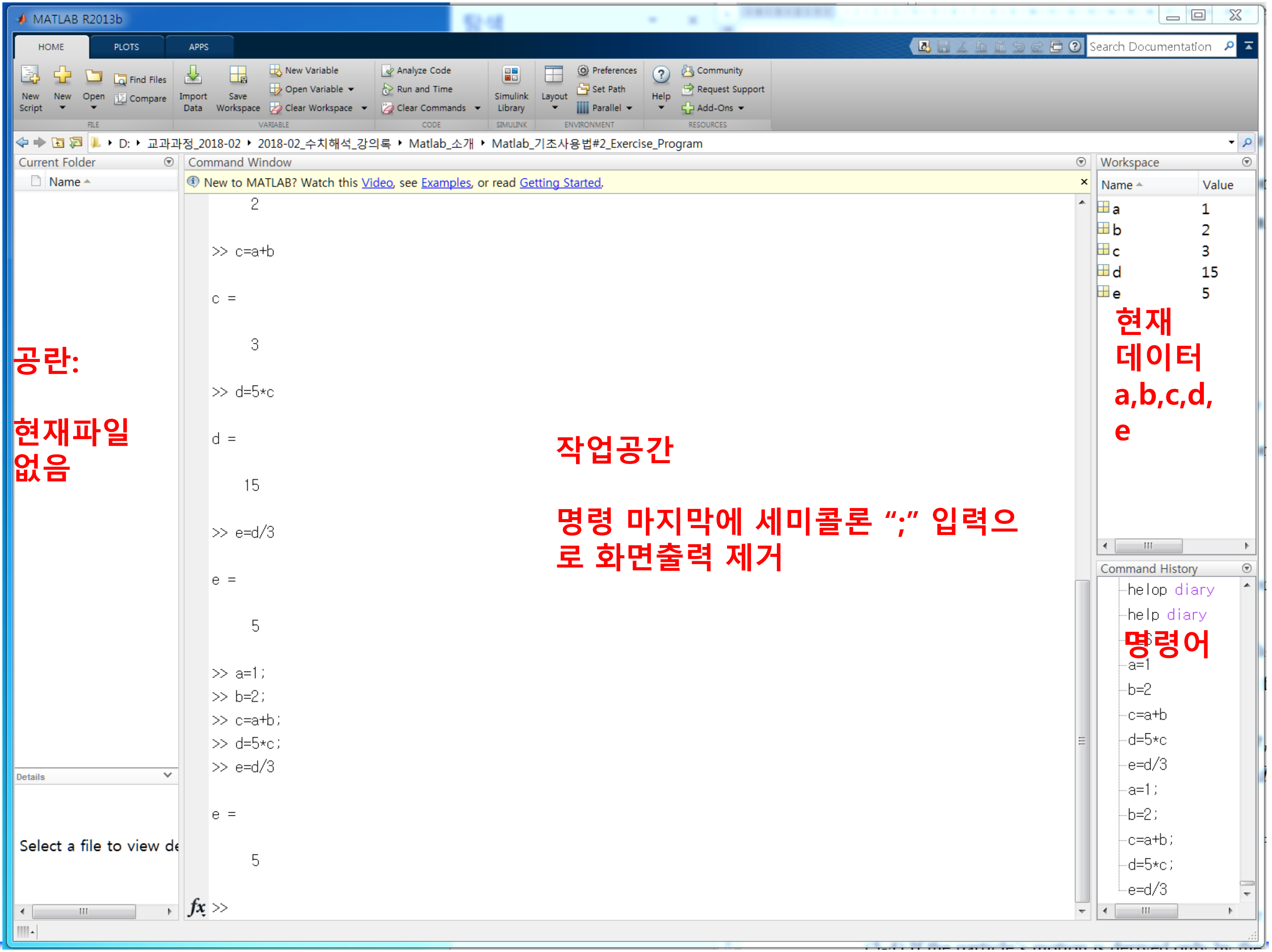
$$a = 1;$$

$$b = 2;$$

$$c = a + b;$$

$$d = 5c;$$

$$e = \frac{d}{3}$$



```
     2

>> c=a+b

c =

     3

>> d=5*c

d =

    15

>> e=d/3

e =

     5

>> a=1;
>> b=2;
>> c=a+b;
>> d=5*c;
>> e=d/3

e =

     5

fx >>
```

Name	Value
a	1
b	2
c	3
d	15
e	5

Command History:

```
helop diary
help diary
a=1
b=2
c=a+b
d=5*c
e=d/3
a=1;
b=2;
c=a+b;
d=5*c;
e=d/3
```

공란:

현재파일 없음

작업공간

명령 마지막에 세미콜론 ";" 입력으로 화면출력 제거

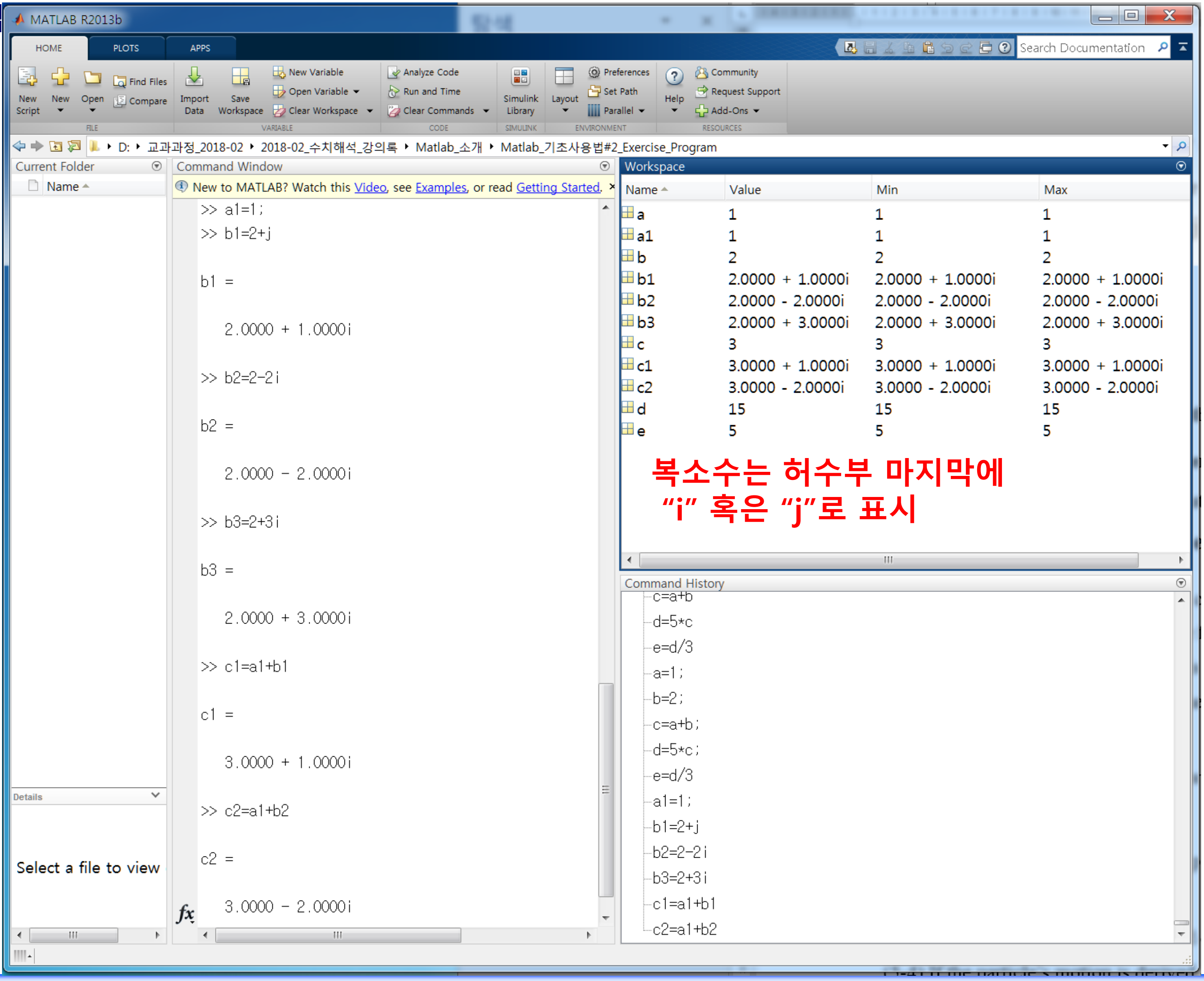
현재 데이터 a,b,c,d, e

명령어

데이터 입력 (3) : 데이터 유형 (정수, 실수, 복소수)

복소수 데이터 입력

$$a1 = 1$$
$$b1 = 2 + j$$
$$b2 = 2 - 2i$$
$$b3 = 2 + 3i$$
$$c1 = a1 + b1$$
$$c2 = a1 + b2$$



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

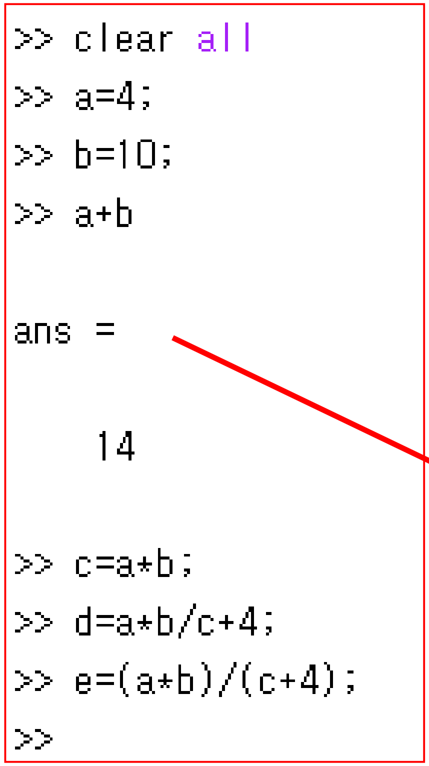
```
>> a1=1;
>> b1=2+j

b1 =

   2.0000 + 1.0000i

>> b2=2-2i

b2 =

   2.0000 - 2.0000i

>> b3=2+3i

b3 =

   2.0000 + 3.0000i

>> c1=a1+b1

c1 =

   3.0000 + 1.0000i

>> c2=a1+b2

c2 =

   3.0000 - 2.0000i
```

Name	Value	Min	Max
a	1	1	1
a1	1	1	1
b	2	2	2
b1	2.0000 + 1.0000i	2.0000 + 1.0000i	2.0000 + 1.0000i
b2	2.0000 - 2.0000i	2.0000 - 2.0000i	2.0000 - 2.0000i
b3	2.0000 + 3.0000i	2.0000 + 3.0000i	2.0000 + 3.0000i
c	3	3	3
c1	3.0000 + 1.0000i	3.0000 + 1.0000i	3.0000 + 1.0000i
c2	3.0000 - 2.0000i	3.0000 - 2.0000i	3.0000 - 2.0000i
d	15	15	15
e	5	5	5

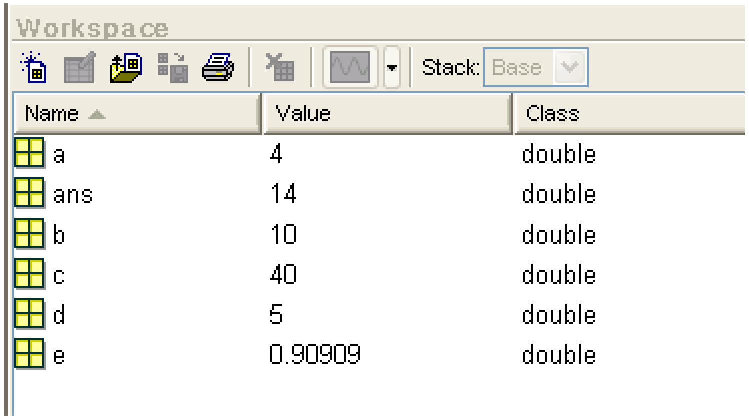
복소수는 허수부 마지막에
“i” 혹은 “j”로 표시

```
c=a+b
d=5*c
e=d/3
a=1;
b=2;
c=a+b;
d=5*c;
e=d/3
a1=1;
b1=2+j
b2=2-2i
b3=2+3i
c1=a1+b1
c2=a1+b2
```

명령창에서 연산 (1)



```
>> clear all
>> a=4;
>> b=10;
>> a+b

ans =

    14

>> c=a+b;
>> d=a+b/c+4;
>> e=(a+b)/(c+4);
>>
```



Name	Value	Class
a	4	double
ans	14	double
b	10	double
c	40	double
d	5	double
e	0.90909	double

ans → 연산결과 (answer) 변수

```
>> ans + 3

ans =

    17
```

- ; 을 붙이면 command window에 표시되지 않음
- 별도의 변수를 지정하지 않고 수식만 입력시 ans 라는 변수에 값이 저장
- 일반적인 연산법칙을 따르며 ^,(*,/),(+,-)순서로 우선적으로 연산됨

명령창에서 연산 (2): Matlab Operation & Commands (1)

- Math Operations

Symbol	Operation	MATLAB form
^	exponentiation:	A^b
*	multiplication:	a*b
/	right division:	a/b
\	left division:	a\b
+	addition:	a+b
-	subtraction:	a-b

- Special Numbers

ans	Temporary variable containing the most recent answer
eps	Specifies the accuracy of floating point precision
i ,j	The imaginary unit
Inf	Infinity
NaN	Indicates an undefined numerical result
pi	The number: = 3.141592….

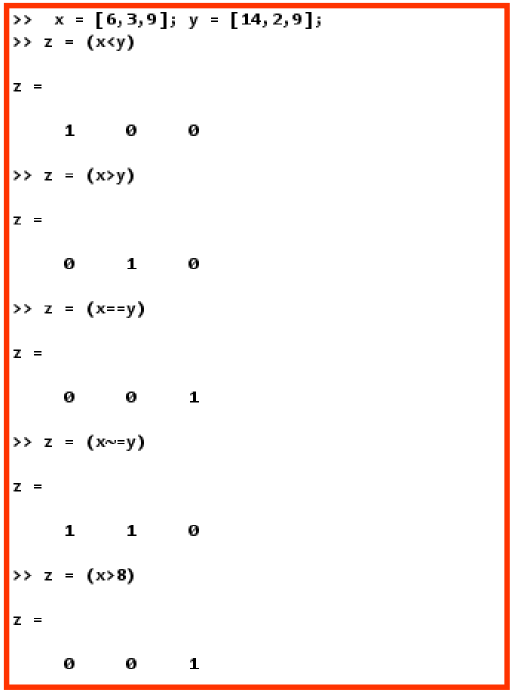
- Math Functions

e^x	exp(x)		
$\sqrt{x}$	sqrt(x)	$\sin^{-1}x$	asin(x)
ln x	log(x)	$\tan^{-1}x$	atan(x)
$\log_{10}x$	log10(x)		
cos x	cos(x)		
sin x	sin (x)		
tan x	tan(x)		
$\cos^{-1}x$	acos(x)		

명령창에서 연산 (3): Matlab Operation & Commands (2)

- **Rational Operators**

Relational operator	Meaning
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to



```
>>  x = [6,3,9]; y = [14,2,9];
>> z = (x<y)

z =

     1     0     0

>> z = (x>y)

z =

     0     1     0

>> z = (x==y)

z =

     0     0     1

>> z = (x~=y)

z =

     1     1     0

>> z = (x>8)

z =

     0     0     1
```

- **Commands**

clc	clears the Command window
clear	Removes all variables form memory
exist('name')	Determines if a file or variable exists having the name 'name'
quit	Stops MATLAB
who	Lists the variable currently in memory
whos	List the current variables and size and indicates if they have imaginary parts
:	Colon: generates and array having regularly spaced elements
,	Comma: separates elements of an array
:	Semicolon: suppress screen printing; also denotes a new row in an array
…	Ellipsis: continues a line

행렬의 표현법 (1)

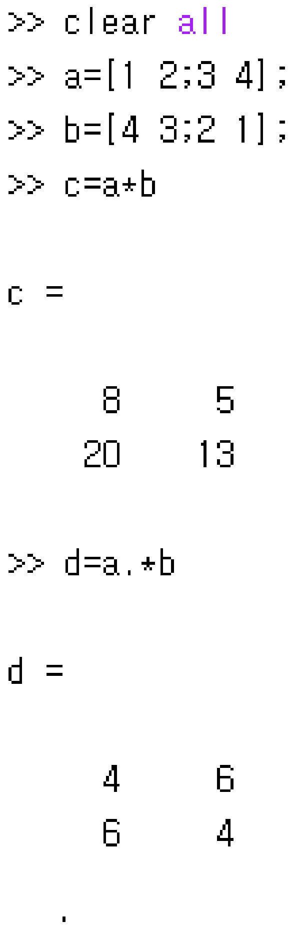
- 행렬 연산 시 일반적인 행렬 연산법칙을 따름

- 행렬의 각 원소끼리 연산을 하고자 할 때는 연산기호 앞에 "." 을 붙임

- M-file에서 행렬작성시엔 ;로 구별하지 않고 새로운 line에 작성해도 별개의 행으로 인식

- 행렬의 특정 원소나 열, 행만을 뽑아 낼 수 있음

```
e=a(1,1)          →         e= 1
f=a(1, :)         →         f= [1 2]
g=a(:,1)          →         g= [ 1
                 3]
a(1,1)=3          →         a= [3, 2 ; 3, 4]
```



```
>> clear all
>> a=[1 2;3 4];
>> b=[4 3;2 1];
>> c=a*b

c =

     8     5
    20    13

>> d=a.*b

d =

     4     6
     6     4
```

행렬의 표현법 (2)

- “ : “ 는 범위를 의미하며 x : y : z 인 경우엔 y씩 만큼 건너 뛰어 x 부터 z까지 의 숫자로 한 행을 만드는 것을 의미

```
h=[0:2:8 ; 0:3:12]
→  h =[0    2    4    6    8
          0    3    6    9   12]
```

```
K=h(1,2:4) -> k=[2 4 6]
```

행렬의 표현법 (3)

- “ : “ 는 범위를 의미하며 x : y : z 인 경우엔 y씩 만큼 건너 뛰어 x 부터 z까지 의 숫자로 한 행을 만드는 것을 의미

```
h=[0:2:8 ; 0:3:12]
→  h =[0    2    4    6    8
        0    3    6    9    12]
```

```
k=h(1,2:4) -> k=[2 4 6]
```

행렬의 표현법 (4): 연산

- **Matrix 연산 : plus**

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

A =

    1    2    3
    4    5    6

>> B=[11 12 13; 14 15 16]

B =

   11    12    13
   14    15    16

>> C= A + B

C =

   12    14    16
   18    20    22

>> C= a + b
??? Undefined function or variable 'a'.

>> who

Your variables are:

A  B  C

>>
```

연산식

$$A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix},\quad B=\begin{pmatrix}11 & 12 & 13\\ 14 & 15 & 16\end{pmatrix}$$

$$C=A+B$$

변수의 대소문자 구분

who: 현재 정의되어 있는 변수명

행렬의 표현법 (5) : 연산

- **Matrix 연산 : scalar product**

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

A =

     1     2     3
     4     5     6

>> c= 20

c =

    20

>> P=c*A

P =

    20    40    60
    80   100   120

>>
```

연산식

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \quad c = 20$$

$$P = cA$$

- **Matrix 연산 : matrix transpose**

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

A =

     1     2     3
     4     5     6


>> b=A'

b =

     1     4
     2     5
     3     6

>>
```

연산식

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$A^T = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

행렬의 표현법 (6) : 연산

▪ Matrix 연산 : matrix product

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

A =

    1    2    3
    4    5    6

>> B=[ 10 20; 30 40; 50 60]

B =

   10    20
   30    40
   50    60

>> c=A*B

c =

  220   280
  490   640

>>
```

연산식

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$B = \begin{pmatrix} 10 & 40 \\ 20 & 50 \\ 30 & 60 \end{pmatrix}$$

$$c = AB$$

▪ Matrix 연산 : 선형 시스템의 해

```
>> A=[6 12 4; 7 -2 3; 2 8 -9]; b=[70; 5; 64];
>> x=inv(A)*b

x =

   3.0000
   5.0000
  -2.0000

>>
```

inv: 역핼렬 계산 내장함수

연산식

$$Ax = b$$

$$A = \begin{pmatrix} 6 & 12 & 4 \\ 7 & -2 & 3 \\ 2 & 8 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} 70 \\ 5 \\ 64 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$x = A^{-1}b$$

행렬 (1) : 행렬의 입력/덧셈

행렬 데이터 입력과 덧셈

$$a1 = [1, 2, 3, 4]$$

$$b1 = [10 \quad 20 \quad 30 \quad 40]$$

$$c1 = a1 + b1$$

$$a2 = \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

$$b2 = [50; 60; 70; 80]$$

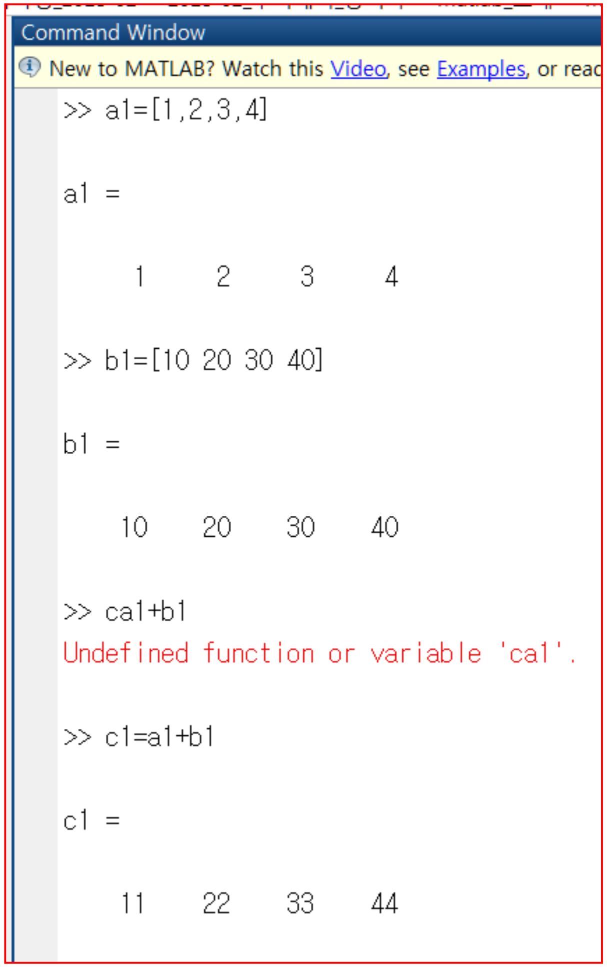
$$c2 = a2 + b2$$

$$a3 = [1, 2, 3; 4, 5, 6]$$

$$b3 = [10, 20, 30; 40, 50, 60]$$

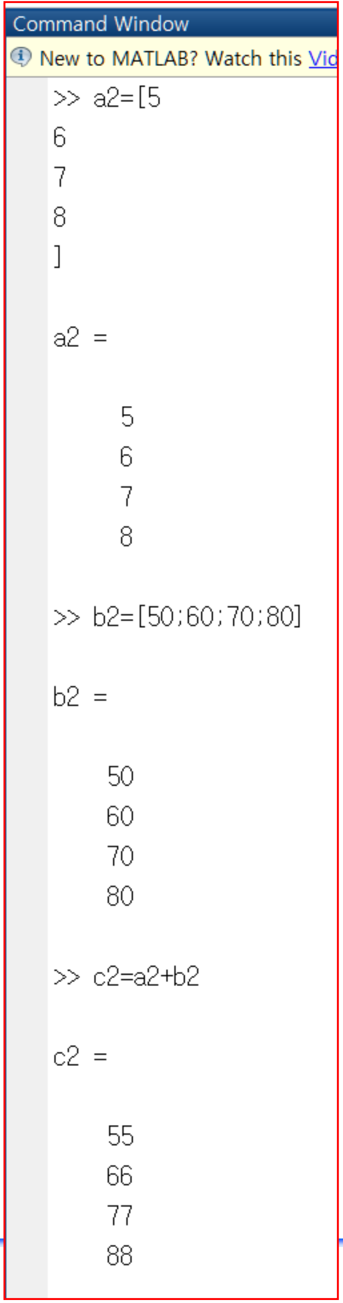
$$c3 = a3 + b3$$

복소수는 허수부 마지막에
"i" 혹은 "j"로 표시


Command Window

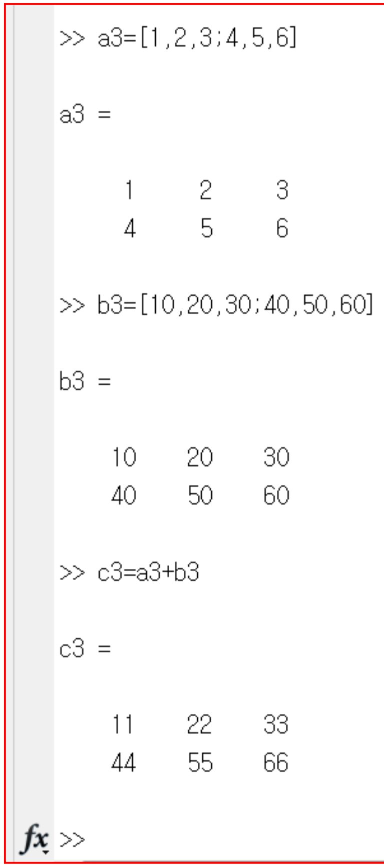
New to MATLAB? Watch this Video, see Examples, or read

```
>> a1=[1,2,3,4]

a1 =

     1     2     3     4

>> b1=[10 20 30 40]

b1 =

    10    20    30    40

>> ca1+b1
Undefined function or variable 'ca1'.

>> c1=a1+b1

c1 =

    11    22    33    44
```


Command Window

New to MATLAB? Watch this Vid

```
>> a2=[5
6
7
8
]

a2 =

     5
     6
     7
     8

>> b2=[50;60;70;80]

b2 =

    50
    60
    70
    80

>> c2=a2+b2

c2 =

    55
    66
    77
    88
```



```
>> a3=[1,2,3;4,5,6]

a3 =

     1     2     3
     4     5     6

>> b3=[10,20,30;40,50,60]

b3 =

    10    20    30
    40    50    60

>> c3=a3+b3

c3 =

    11    22    33
    44    55    66

fx >>
```

행렬 (2) : 행렬의 입력/곱셈

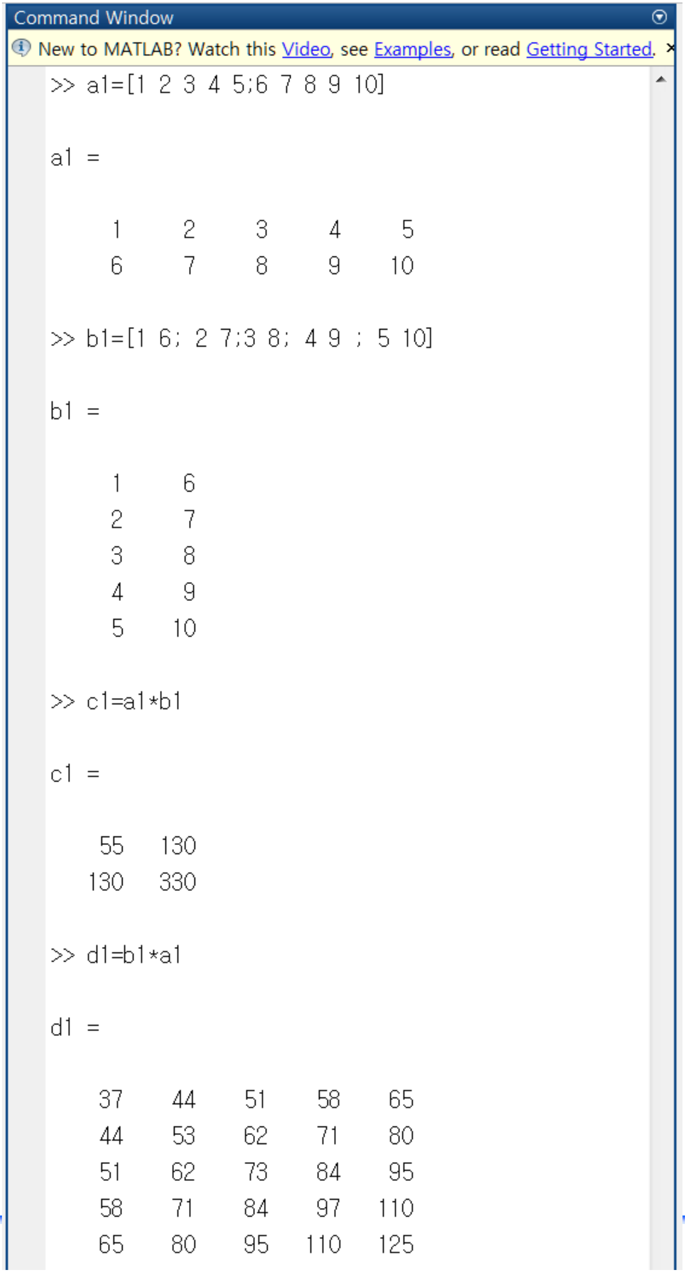
행렬 데이터 입력과 곱셈

$$a1=[1,2,3,4,5;6,7,8,9,10]$$

$$b1=[1\;6;2\;7;3\;8;4\;9;5\;10]$$

$$c1=a1*b1$$

$$d1=b1*a1$$



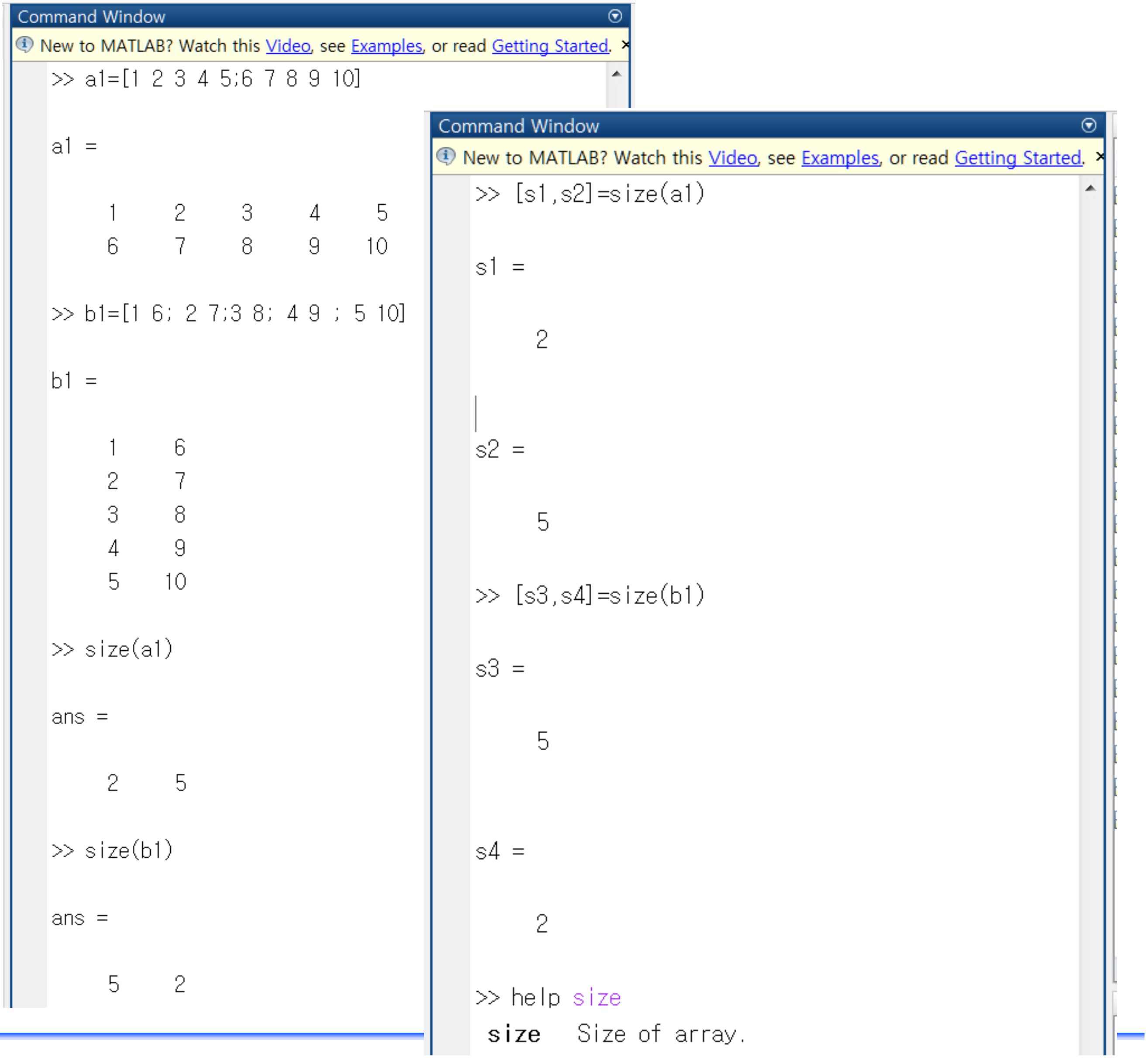
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> a1=[1 2 3 4 5;6 7 8 9 10]

a1 =

     1     2     3     4     5
     6     7     8     9    10

>> b1=[1 6; 2 7;3 8; 4 9 ; 5 10]

b1 =

     1     6
     2     7
     3     8
     4     9
     5    10

>> c1=a1*b1

c1 =

    55   130
   130   330

>> d1=b1*a1

d1 =

    37    44    51    58    65
    44    53    62    71    80
    51    62    73    84    95
    58    71    84    97   110
    65    80    95   110   125
```

행렬 (3) : 행렬의 크기 (size 명령어와 help)

행렬의 크기

$$a1 = [1,2,3,4,5;6,7,8,9,10]$$
$$b1 = [1\ 6;2\ 7;3\ 8;4\ 9;5\ 10]$$
$$size(a1)$$
$$size(b1)$$
$$[s1, s2] = size(a1)$$
$$[s3, s4] = size(b1)$$
$$help\ size$$


Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> a1=[1 2 3 4 5;6 7 8 9 10]

a1 =

     1     2     3     4     5
     6     7     8     9    10

>> b1=[1 6; 2 7;3 8; 4 9 ; 5 10]

b1 =

     1     6
     2     7
     3     8
     4     9
     5    10

>> size(a1)

ans =

     2     5

>> size(b1)

ans =

     5     2
```

Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> [s1,s2]=size(a1)

s1 =

     2


s2 =

     5

>> [s3,s4]=size(b1)

s3 =

     5


s4 =

     2

>> help size
 size   Size of array.
```

행렬 (4) : 행렬의 요소

행렬의 크기

$$a=\begin{bmatrix} 1,2,3,4,5,6,7,8,9,10 \\ 11,12,13,14,15,16,17,18,19,20 \\ 21,22,23,24,25,26,27,28,29,30 \\ 31,32,33,34,35,36,37,38,39,40 \\ 41,42,43,44,45,46,47,48,49,50 \\ 51,52,53,54,55,56,57,58,59,60 \\ 61,62,63,64,65,66,67,68,69,70 \end{bmatrix}$$

$$c1=a(2,5)$$

$$c2=a(1,2:6)$$

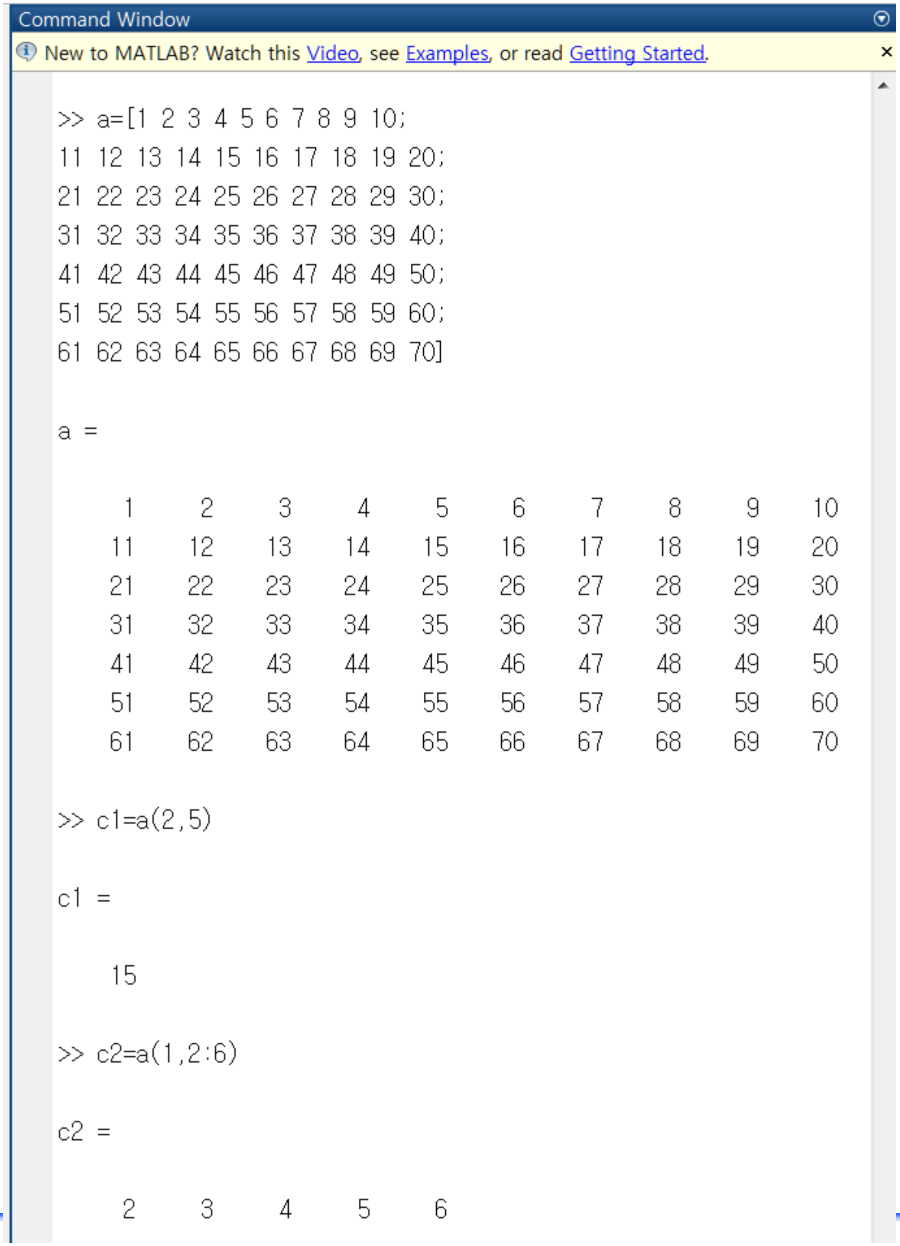
$$c3=a(1:3,2)$$

$$c4=a(2:3,4:7)$$

$$c5=a(1,2:2:6)$$

$$c6=a(1:3:7,2)$$

$$c7=a(1:3:7,2:2:6)$$



Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> a=[1 2 3 4 5 6 7 8 9 10;
11 12 13 14 15 16 17 18 19 20;
21 22 23 24 25 26 27 28 29 30;
31 32 33 34 35 36 37 38 39 40;
41 42 43 44 45 46 47 48 49 50;
51 52 53 54 55 56 57 58 59 60;
61 62 63 64 65 66 67 68 69 70]

a =

     1     2     3     4     5     6     7     8     9    10
    11    12    13    14    15    16    17    18    19    20
    21    22    23    24    25    26    27    28    29    30
    31    32    33    34    35    36    37    38    39    40
    41    42    43    44    45    46    47    48    49    50
    51    52    53    54    55    56    57    58    59    60
    61    62    63    64    65    66    67    68    69    70

>> c1=a(2,5)

c1 =

    15

>> c2=a(1,2:6)

c2 =

     2     3     4     5     6
```

행렬 (5) : 행렬의 요소

행렬의 요소

$$a = \begin{bmatrix} 1,2,3,4,5,6,7,8,9,10 \\ 11,12,13,14,15,16,17,18,19,20 \\ 21,22,23,24,25,26,27,28,29,30 \\ 31,32,33,34,35,36,37,38,39,40 \\ 41,42,43,44,45,46,47,48,49,50 \\ 51,52,53,54,55,56,57,58,59,60 \\ 61,62,63,64,65,66,67,68,69,70 \end{bmatrix}$$

$c1 = a(2,5)$

$c2 = a(1,2:6)$

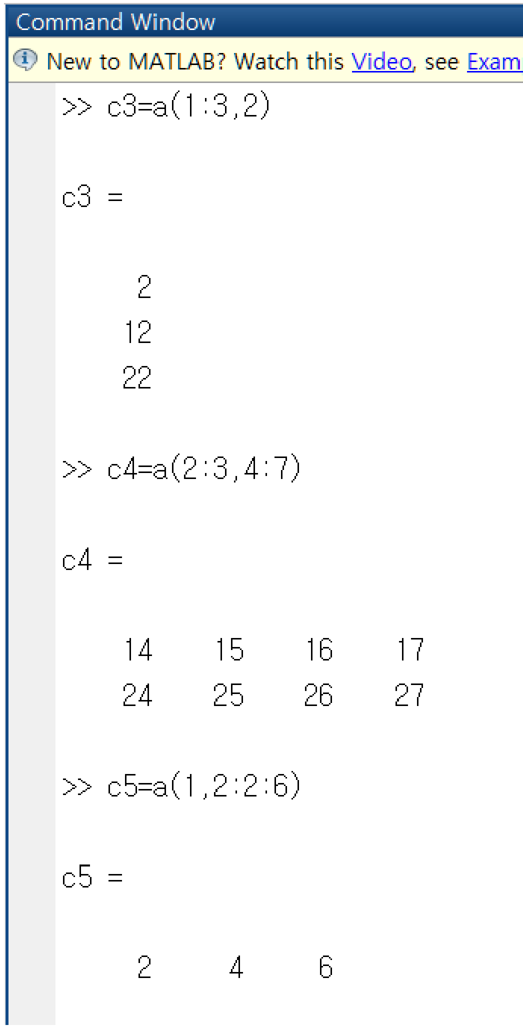
$c3 = a(1:3,2)$

$c4 = a(2:3,4:7)$

$c5 = a(1,2:2:6)$

$c6 = a(1:3:7,2)$

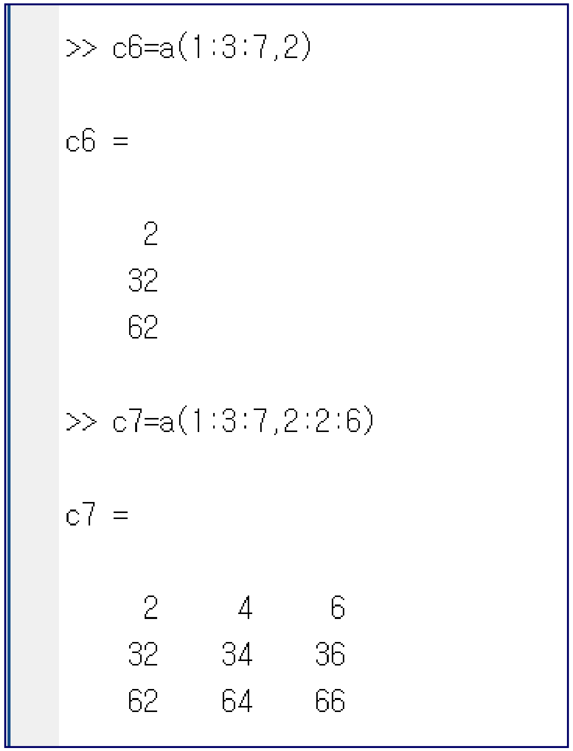
$c7 = a(1:3:7,2:2:6)$



Command Window

New to MATLAB? Watch this Video, see Example

```
>> c3=a(1:3,2)

c3 =

     2
    12
    22

>> c4=a(2:3,4:7)

c4 =

    14    15    16    17
    24    25    26    27

>> c5=a(1,2:2:6)

c5 =

     2     4     6
```



```
>> c6=a(1:3:7,2)

c6 =

     2
    32
    62

>> c7=a(1:3:7,2:2:6)

c7 =

     2     4     6
    32    34    36
    62    64    66
```

행렬 (6) : Empty (null) 행렬

- The empty or null array
  - √ contains no elements. []
  - √ Row and columns can be deleted by setting the selecteed row or column equal to the null array.
    - A(3,:) = [] deletes the third row in **A**.
    - A(:,2:4) = [] deletes the second through fourth columns in **A**.
    - A([1:4],:) deletes the first and fourth row of **A**.

〈Other example 〉

$$\mathbf{A} = \begin{bmatrix} 6 & 9 & 4 \\ 1 & 5 & 7 \end{bmatrix}$$

A(1,5) = 3 $\mathbf{A} = \begin{bmatrix} 6 & 9 & 4 & 0 & 3 \\ 1 & 5 & 7 & 0 & 0 \end{bmatrix}$

B = A(:,5:-1:1) $\mathbf{B} = \begin{bmatrix} 3 & 0 & 4 & 9 & 6 \\ 0 & 0 & 7 & 5 & 1 \end{bmatrix}$

행렬 (7) : 요소별 연산

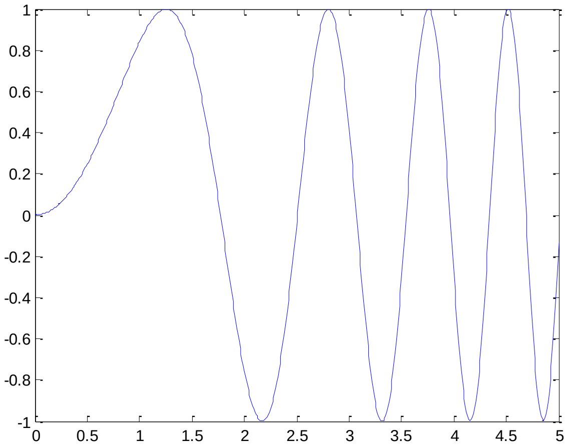
- Element-by-element operations

<example 1>

```
>> x = [1 2 3];
>> y = [4 5 6];
>> x.^y
ans =
     1    32   729

>> y.^2
ans =
    16    25    36

>> 2.^[x y]
ans =
     2     4     8    16    32    64
```

<example 2>

```
>> x = 0:0.01:5 ;
>> y = sin(x^2);
??? Error using ==>
mpower
Matrix must be square.

>> y = sin(x.^2);
```



cf) ==> 2^[1 2 3 4 5 6] = [2^1 2^2 2^3 2^4 2^5 2^6]

명령창에서 프로그램 작성 (1)

- Relational operators
  - √ to make comparisons
  - √ <, <=, >, >=, ==, ~=

- Conditional statements
  - √ to write programs that make decisions
  - √ if, else, elseif

- Loops
  - √ a structure for reputation a calculation a number of times
  - √ for, while

명령창에서 프로그램 작성 (2)

- **Conditional Statements**
  - √ Contain one or more of the if, else, and elseif
  - √ The end statement denotes the end of a conditional statement
  - √ The else and elseif statements may be omitted if not required

```
if  expression
          commands
else if  expression
          commands
else
          commands
end
```

<example>

$$y = \begin{cases} 15\sqrt{4}+10 & if \quad x \ge 9 \\ 10x+10 & if \quad 0 \le x < 9 \\ 10 & if \quad x < 0 \end{cases}$$

```
If  x >=  9
   y = 15*sqrt(4*x) + 10
elseif  x  >=  0
   y = 10*x + 10
else
   y = 10
end
```

명령창에서 프로그램 작성 (3)

- Loops
  - √ Repeat a calculation a given number of times
  - √ for loop: the number of passes is known ahead of time
  - √ while loop: the looping process must terminate when a specified condition is satisfied

<example of a for loop>

```
m = 0;
x(1) = 10;
for k = 2:3:11
   m = m + 1;
   x(m+1) = x(m) + k^2;
end
```

x(1) = 14, x(2) = 39, x(3) = 103, x(4) = 224

<example of a while loop>

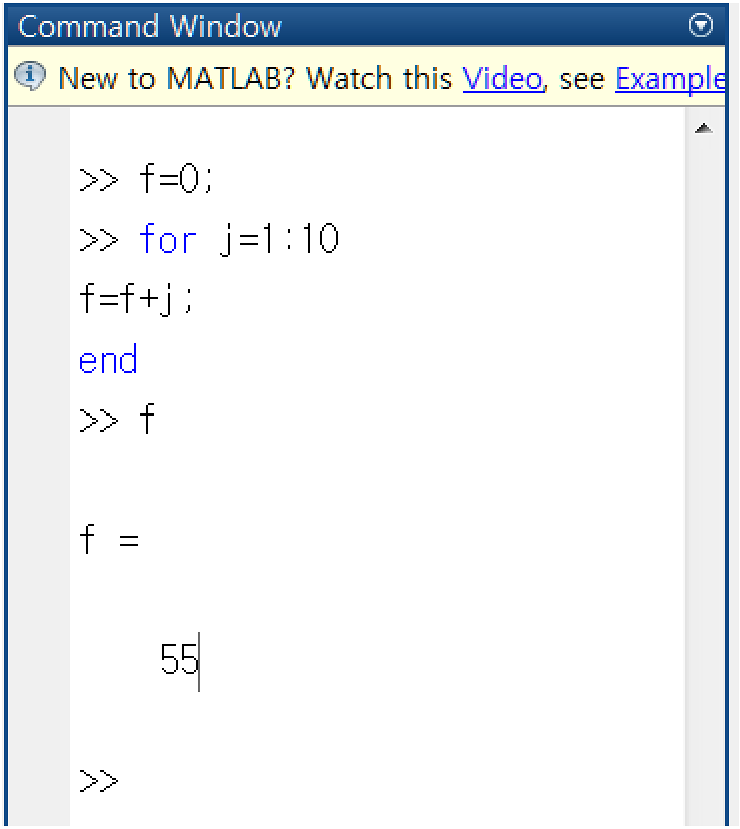
```
x = 5; k = 0;
while x < 25
   k = k + 1;
   y(k) = 3*x;
   x = 2*x - 1;
end
```

y(1) = 15, y(2) = 27, y(3) = 51

명령창에서 프로그램 작성

연산

$$a = \sum_{j=1}^{10} j = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

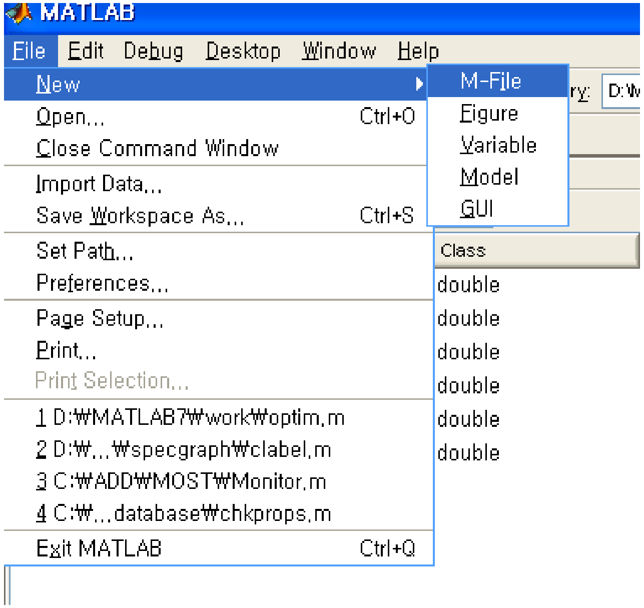


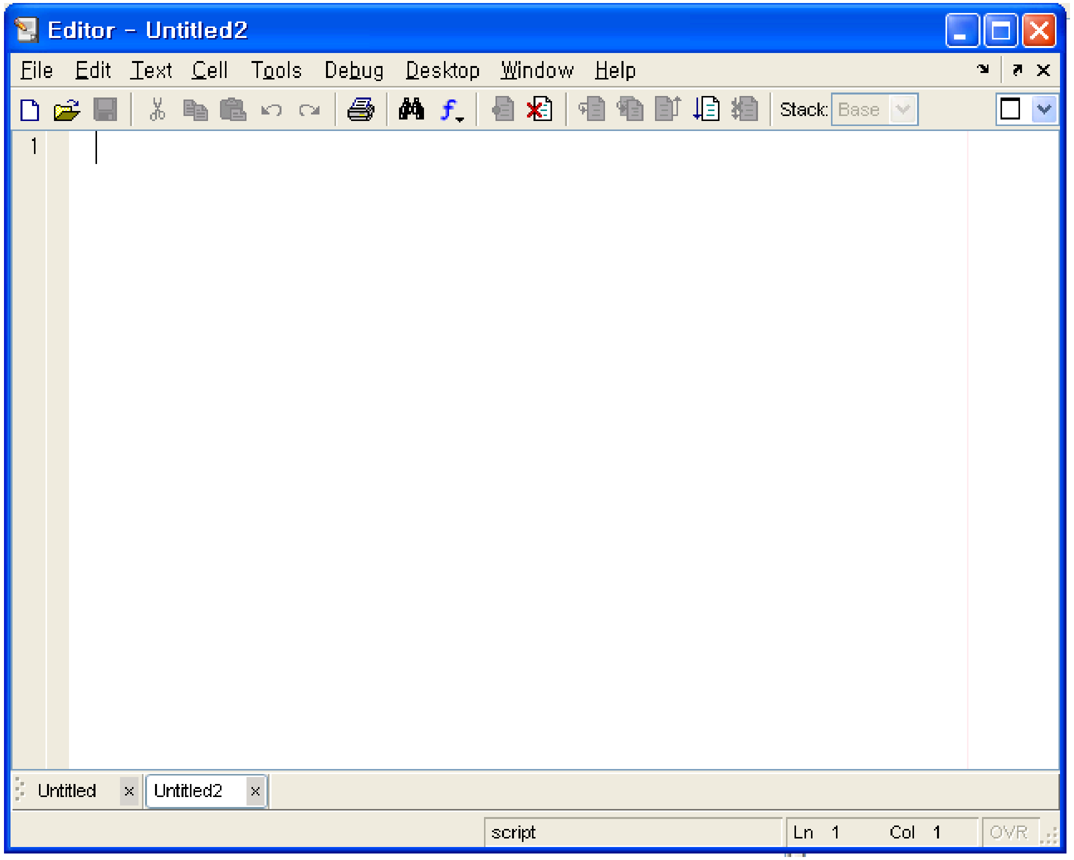
New to MATLAB? Watch this Video, see Example

```
>> f=0;
>> for j=1:10
f=f+j;
end
>> f

f =

    55

>>
```

새로운 m-file 작성 (1)

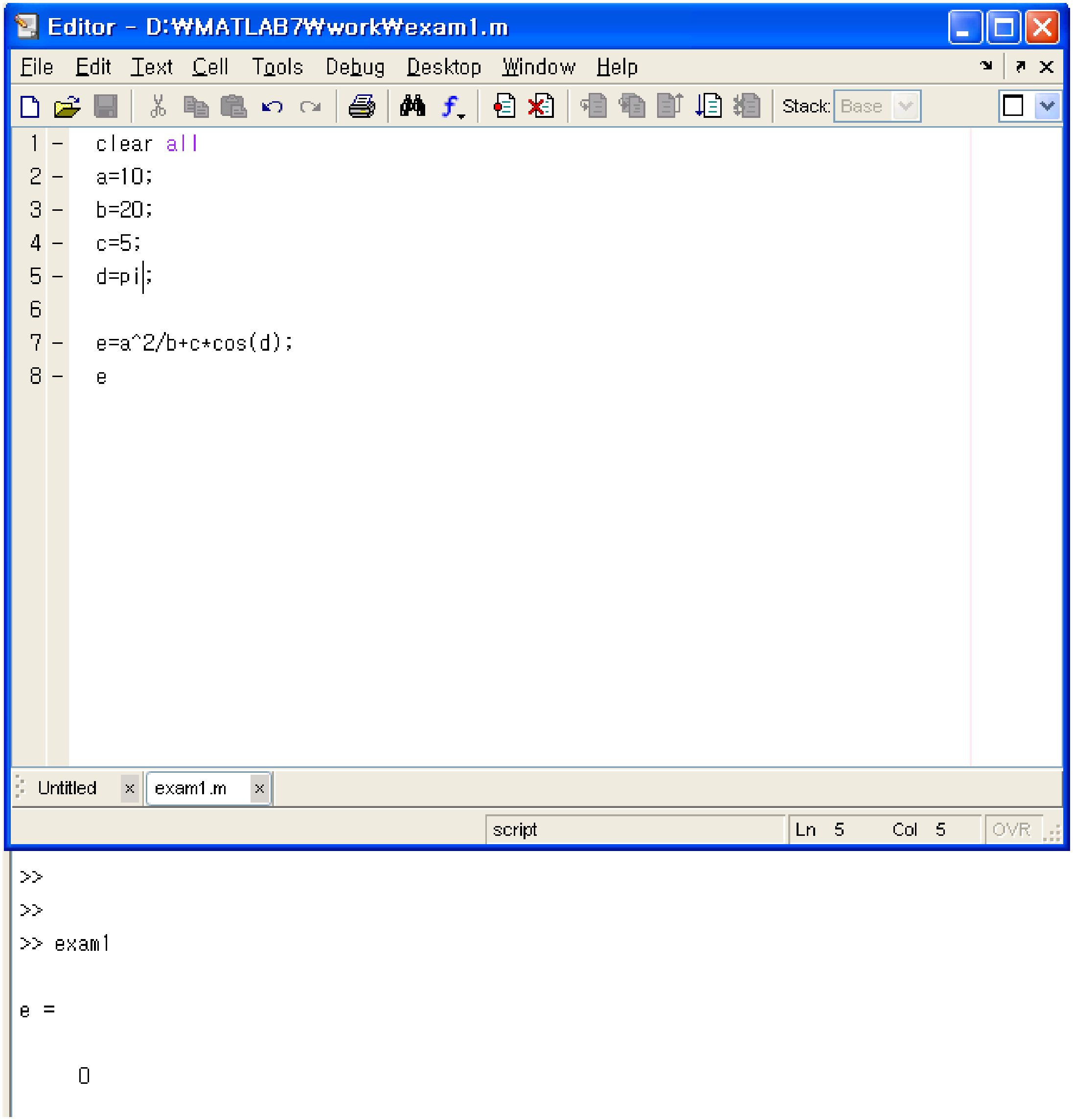
- 별도의 file을 작성하여 저장해 놓은 뒤, 원하는 때 마다 불러서 사용 가능





새로운 m-file 작성 (2)

- Editor에서 수식 작성
- 파일을 저장 후 저장된 파일을 command window에서 입력
- Path로 지정된 폴더 내에 있는 경우 실행 됨
- Path(path,” 파일 경로”) 명령으로 Path 추가 가능
- Path 명령으로 Path 확인 가능
- Menubar의 File->set Path에서 Path 확인 및 편집 가능



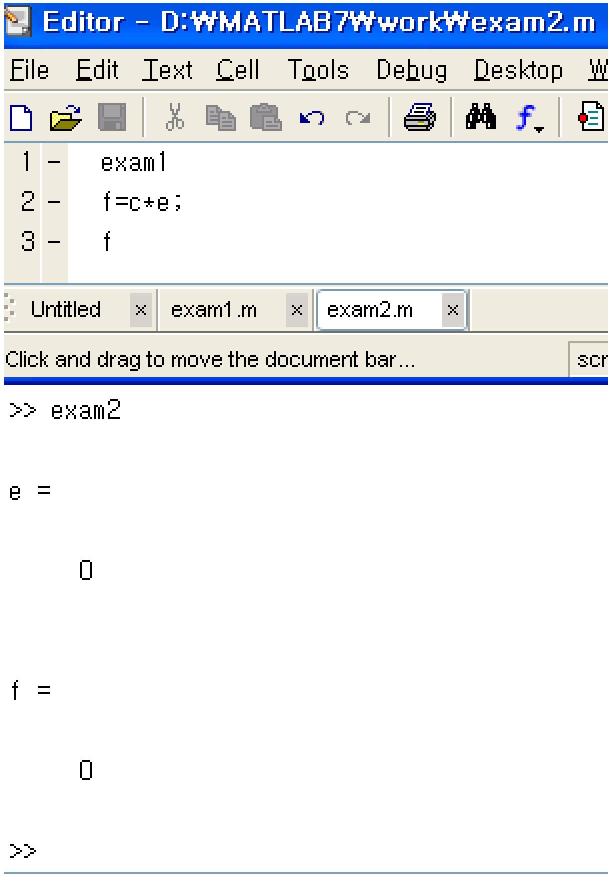
Editor - D:₩MATLAB7₩work₩exam1.m
File Edit Text Cell Tools Debug Desktop Window Help
Stack: Base
Untitled
exam1.m
script
Ln 5
Col 5
OVR

```
clear all
a=10;
b=20;
c=5;
d=pi;

e=a^2/b+c*cos(d);
e
```

```
>>
>>
>> exam1

e =

     0
```

새로운 m-file 작성 (3)

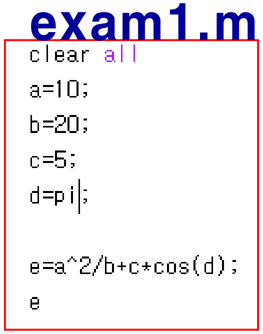
- **M-file 내에 다른 M-file 명을 입력하여도 command window에서 입력한 것과 동일하게 실행 됨**


Editor - D:₩MATLAB7₩work₩exam2.m
File Edit Text Cell Tools Debug Desktop W
Untitled
exam1.m
exam2.m
Click and drag to move the document bar...
scr

```
exam1
f=c*e;
f
```

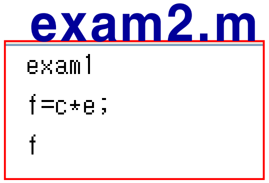
```
>> exam2

e =

     0


f =

     0

>>
```

exam1.m

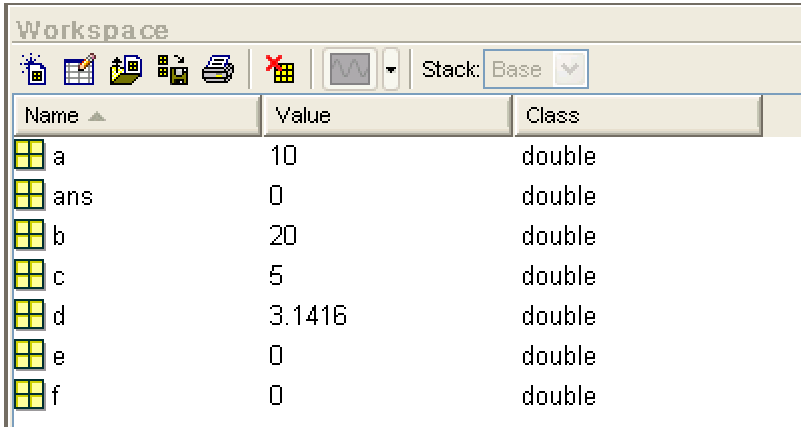


```
clear all
a=10;
b=20;
c=5;
d=pi;

e=a^2/b+c*cos(d);
e
```

exam2.m



```
exam1
f=c*e;
f
```

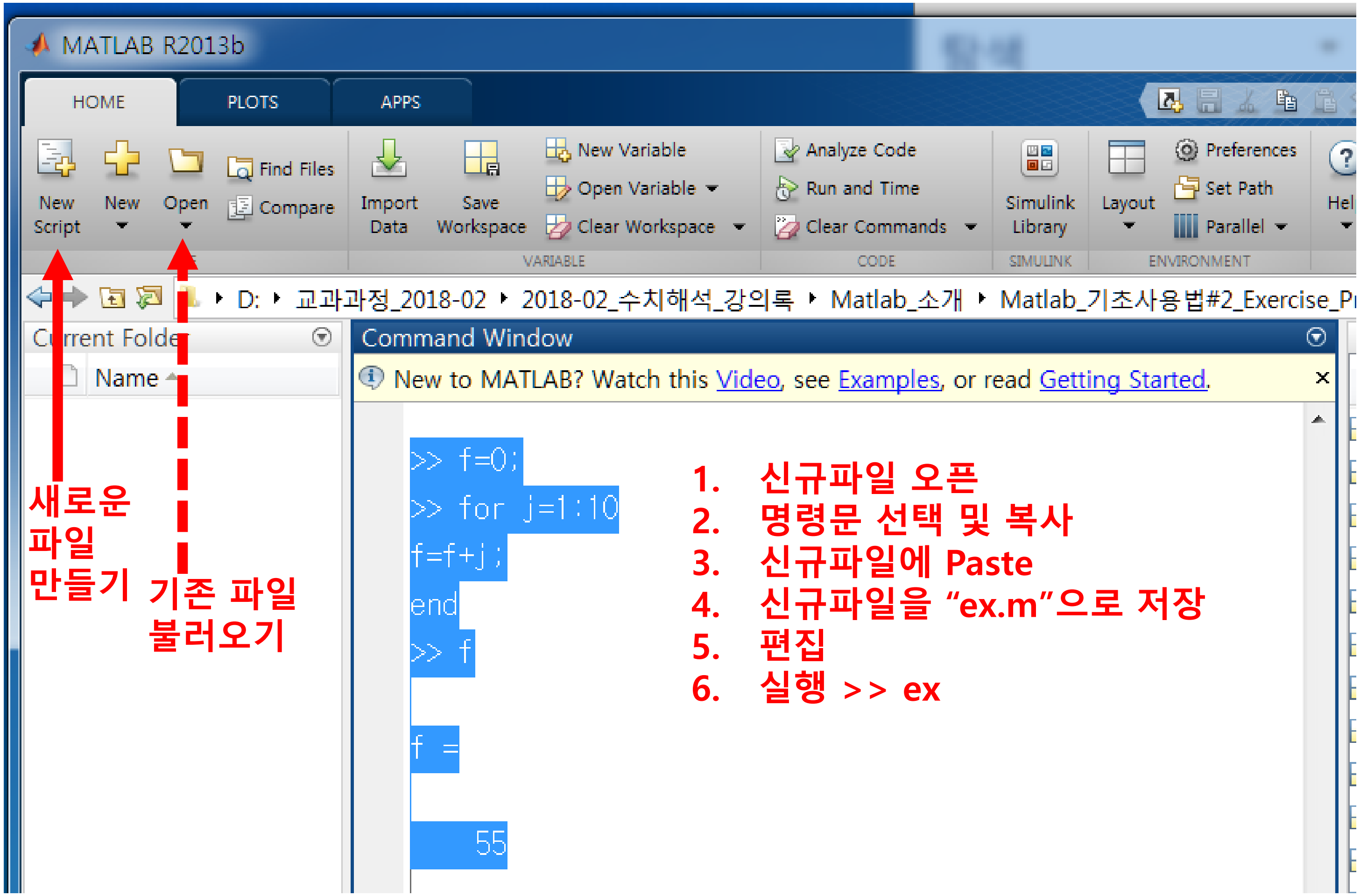


Name	Value	Class
a	10	double
ans	0	double
b	20	double
c	5	double
d	3.1416	double
e	0	double
f	0	double

간단한 m-file 작성 (1)

연산

$$a = \sum_{j=1}^{10} j = 1+2+3+4+5+6+7+8+9+10 = 55$$



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

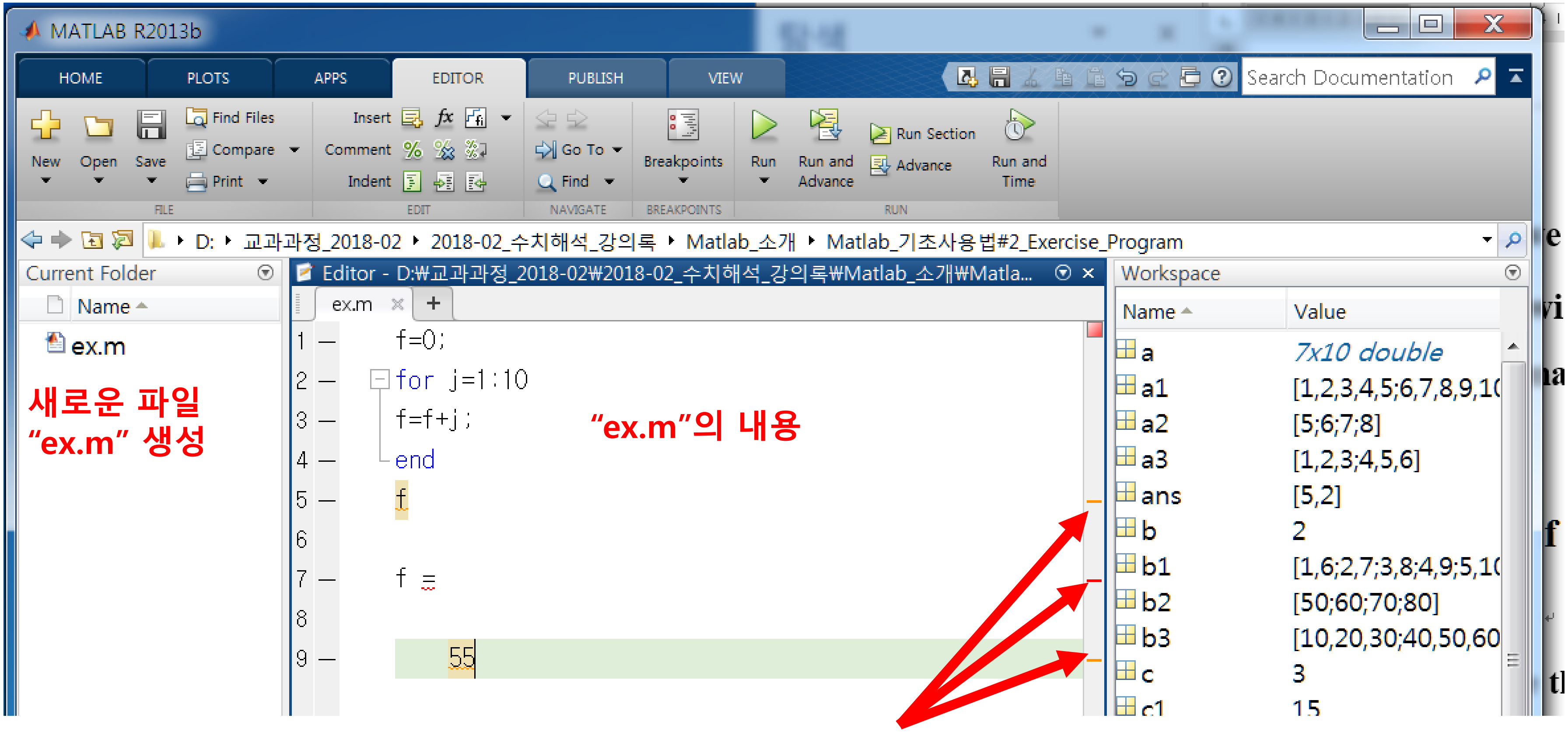
```
>> f=0;
>> for j=1:10
f=f+j;
end
>> f

f =

    55
```

1. 신규파일 오픈
2. 명령문 선택 및 복사
3. 신규파일에 Paste
4. 신규파일을 “ex.m”으로 저장
5. 편집
6. 실행 >> ex

간단한 m-file 작성 (2)

연산

$$a = \sum_{j=1}^{10} j = 1+2+3+4+5+6+7+8+9+10 = 55$$



```
f=0;
for j=1:10
f=f+j;
end
f

f =

    55
```

Name	Value
a	7x10 double
a1	[1,2,3,4,5;6,7,8,9,1
a2	[5;6;7;8]
a3	[1,2,3;4,5,6]
ans	[5,2]
b	2
b1	[1,6;2,7;3,8;4,9;5,1
b2	[50;60;70;80]
b3	[10,20,30;40,50,60
c	3
c1	15

새로운 파일 "ex.m" 생성

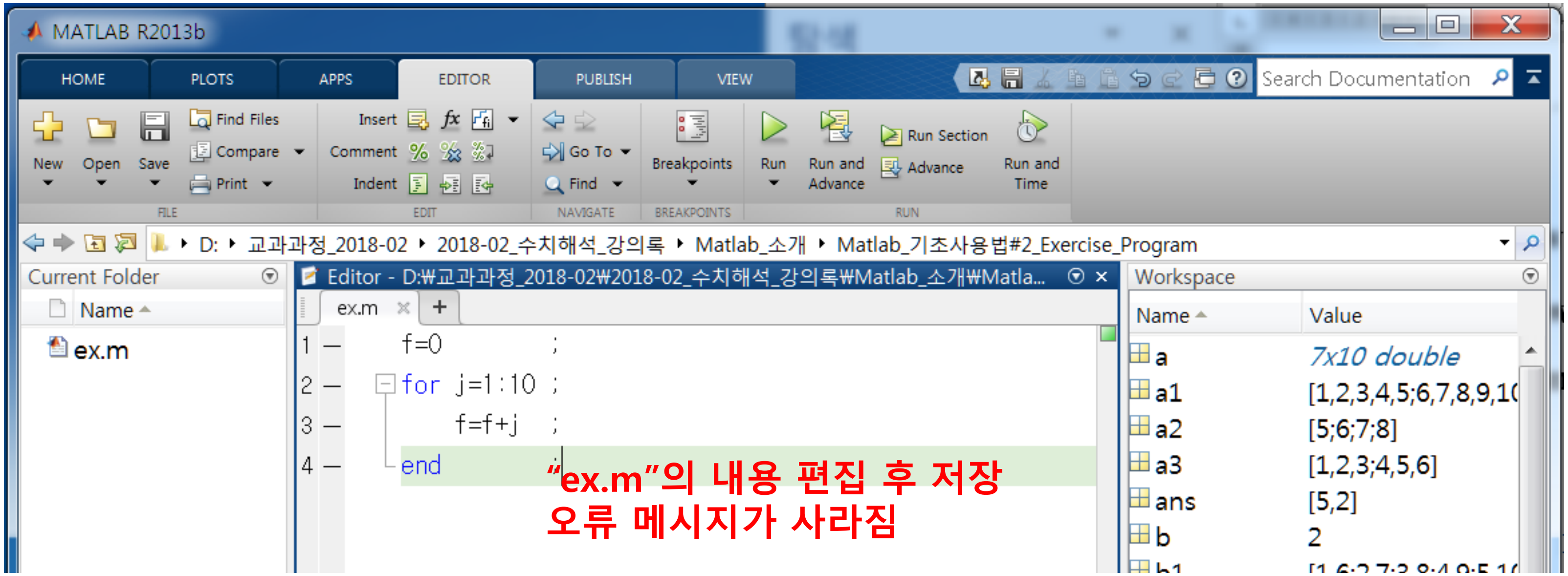
"ex.m"의 내용

오류 메시지

간단한 m-file 작성 (3)

연산

$$a = \sum_{j=1}^{10} j = 1+2+3+4+5+6+7+8+9+10 = 55$$



D: ▸ 교과과정_2018-02 ▸ 2018-02_수치해석_강의록 ▸ Matlab_소개 ▸ Matlab_기초사용법#2_Exercise_Program

Editor - D:₩교과과정_2018-02₩2018-02_수치해석_강의록₩Matlab_소개₩Matla...

```
    f=0          ;
for j=1:10   ;
        f=f+j   ;
end          ;
```

Name	Value
a	*7x10 double*
a1	[1,2,3,4,5;6,7,8,9,1(
a2	[5;6;7;8]
a3	[1,2,3;4,5,6]
ans	[5,2]
b	2

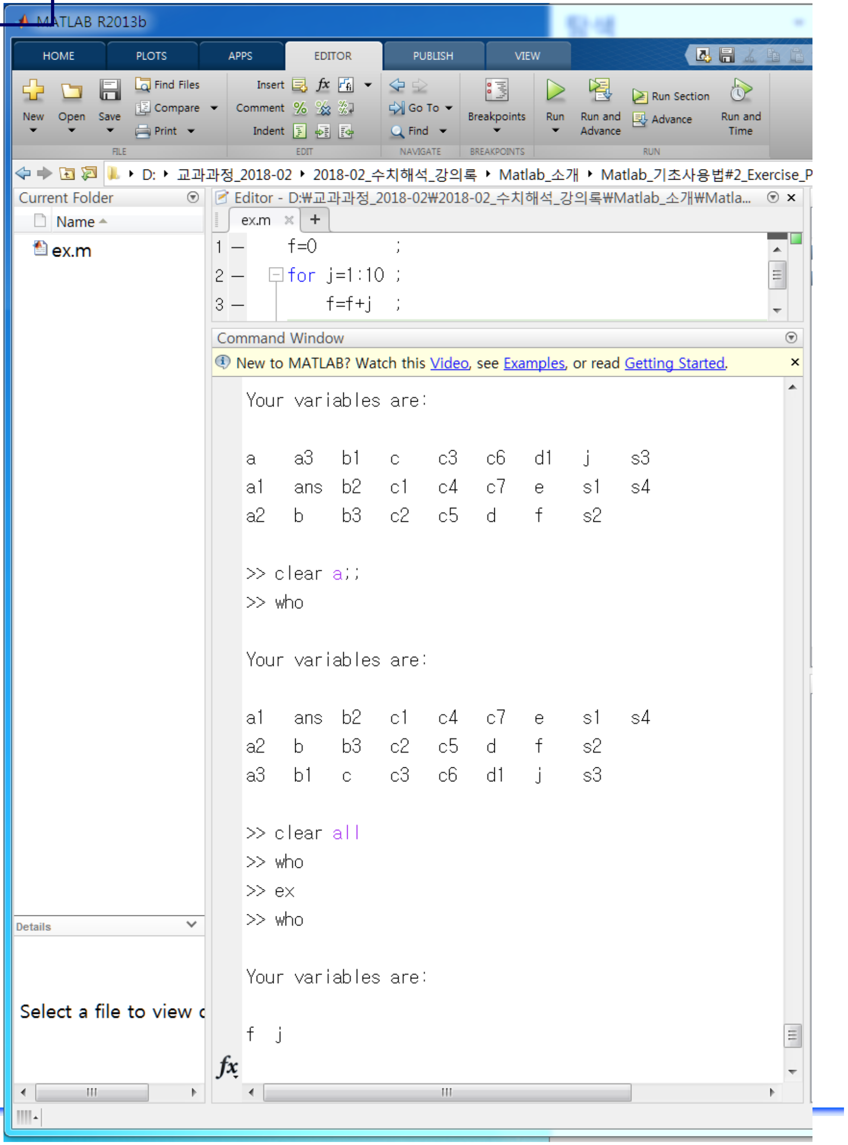
"ex.m"의 내용 편집 후 저장
오류 메시지가 사라짐

간단한 m-file 작성 (4): 실행

연산

$$a = \sum_{j=1}^{10} j = 1+2+3+4+5+6+7+8+9+10 = 55$$

who 명령어: 메모리에 저장된 변수
clear all 명령어 : 모든 변수 삭제



```
f=0        ;
for j=1:10 ;
    f=f+j  ;
```

New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
Your variables are:

a    a3   b1   c    c3   c6   d1   j    s3
a1   ans  b2   c1   c4   c7   e    s1   s4
a2   b    b3   c2   c5   d    f    s2

>> clear a;;
>> who

Your variables are:

a1   ans  b2   c1   c4   c7   e    s1   s4
a2   b    b3   c2   c5   d    f    s2
a3   b1   c    c3   c6   d1   j    s3

>> clear all
>> who
>> ex
>> who

Your variables are:

f  j

fx
```

2차 방정식 m-file (1): 작성

연산

$$\boxed{ax^2 + bx + c = 0}$$

함수의 기본 형태

Function [출력변수] = 함수명(입력변수)

```
%--------------------------------------------------------------
% Finding roots for the 1st order and the 2nd order polynomial
equations
% y = a(1,1)+a(2,1)*x + a(3,1)*x^2
%-------------------------------------------------------------
% Input
%     a(3,1)            : polynomial coefficients
%     epsilon_eisilon : very very small number (0.1^16)
%-------------------------------------------------------------
% Output
%     m           : number of roots
%     root_real   : real parts of roots
%     root_imag   : imaginary parts of roots
%-------------------------------------------------------------
function [m,root_real,root_imag] = root_formula(a,epsilon_eisilon)
%-------------------------------------------------------------
%  Computaion of two roots
%--------------------------------------------------------------
    root_real(1:2,1) = 0.0;   root_imag(1:2,1) = 0.0;
%
    aa = a(3,1);  bb = a(2,1);    cc = a(1,1);
%
%--------------------------------------------------------------
    if abs(aa) < epsilon_eisilon
%--------------------------------------------------------------
              if abs(bb) < epsilon_eisilon
                  m = 0;
                  return
              else
                  m = 1;
                  root_real(1,1) = -cc/bb;
                  return
              end
%--------------------------------------------------------------
    else
%--------------------------------------------------------------
```

```
m  = 2;
        dd = bb*bb - 4.0*aa*cc;
        coef              =  0.5/aa;
        real_part         = -bb*coef;
        imag_part         =  sqrt(abs(dd))*coef;
%
        if dd < 0.0
            root_real(1:2,1)  = real_part;
            root_imag(1,1)    = imag_part;
            root_imag(2,1)    =-imag_part;
        else
            root_imag(1:2,1)  = 0.0;
            root_real(1,1)    = real_part + imag_part;
            root_real(2,1)    = real_part - imag_part;
        end
    end
%---------------------------------------------
```

2차 방정식 m-file (2): 실행

연산

$$x^2 - 1 = 0$$

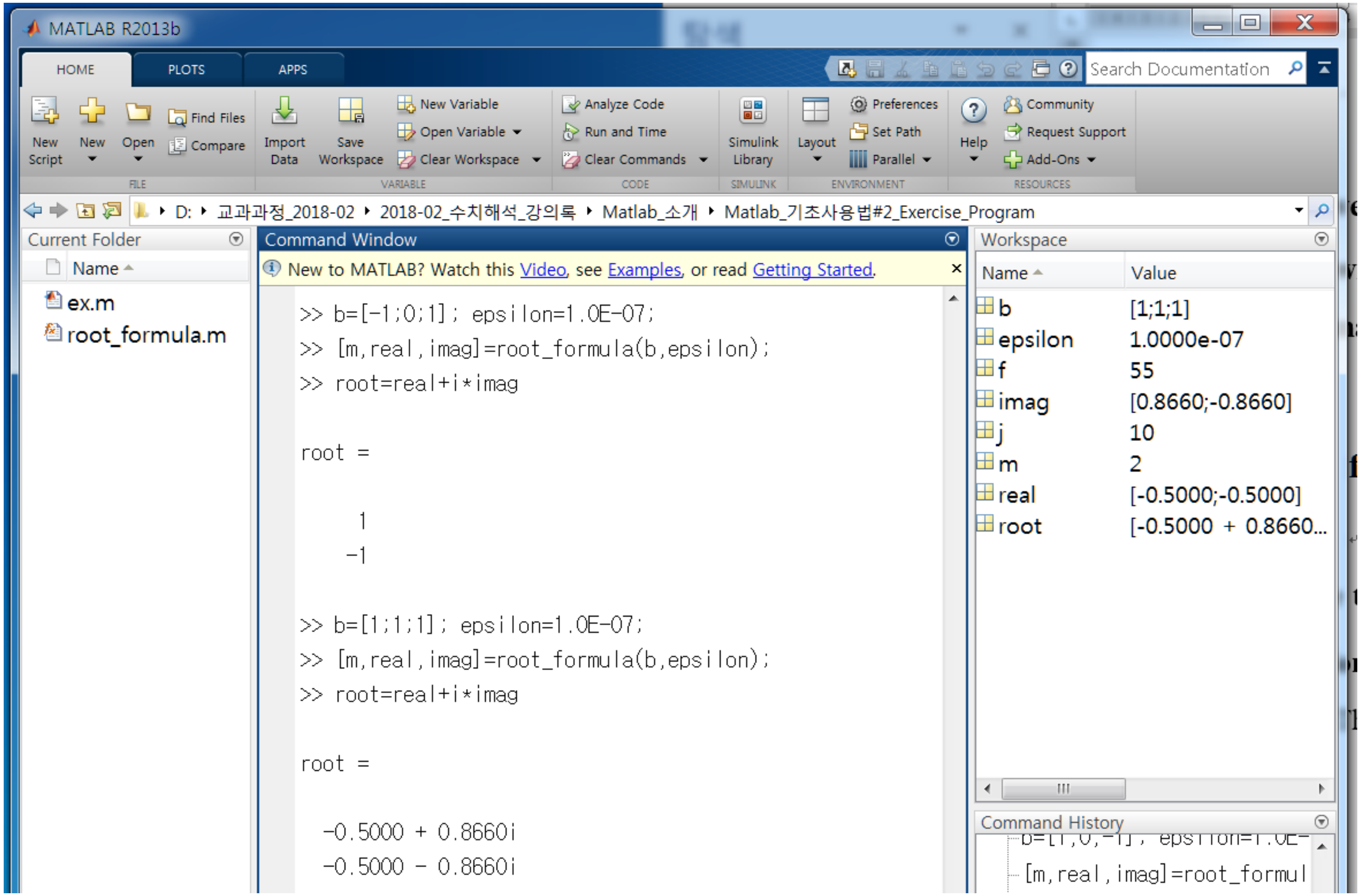
$$x^2 + x + 1 = 0$$

함수의 기본 형태 (주의 파일명과 함수명을 일치시킬 것)

Function [출력변수] = 함수명(입력변수)

실행명령

[출력변수] = 함수명(입력변수)



D: ▸ 교과과정_2018-02 ▸ 2018-02_수치해석_강의록 ▸ Matlab_소개 ▸ Matlab_기초사용법#2_Exercise_Program

Current Folder

- ex.m
- root_formula.m

New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
>> b=[-1;0;1]; epsilon=1.0E-07;
>> [m,real,imag]=root_formula(b,epsilon);
>> root=real+i*imag

root =

     1
    -1

>> b=[1;1;1]; epsilon=1.0E-07;
>> [m,real,imag]=root_formula(b,epsilon);
>> root=real+i*imag

root =

  -0.5000 + 0.8660i
  -0.5000 - 0.8660i
```

Name	Value
b	[1;1;1]
epsilon	1.0000e-07
f	55
imag	[0.8660;-0.8660]
j	10
m	2
real	[-0.5000;-0.5000]
root	[-0.5000 + 0.8660...

Command History

```
b=[1;0;-1]; epsilon=1.0E-
[m,real,imag]=root_formul
```

2-D 데이터 plot (1)

- **Plot (변수a, 변수b)**

  → 변수a를 X축으로, 변수b를 Y축으로 하여 그래프 작성

- **Plot(변수a, 변수b,' 그래프모양 지정')**

  → 표시될 색과 그래프의 모양을 지정 할 수 있음

  r : Red k : Black y : Yellow

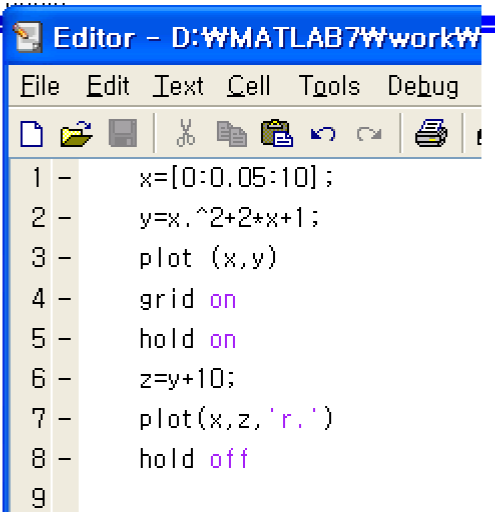
  b : Blue

  - : bar . : dot *:star

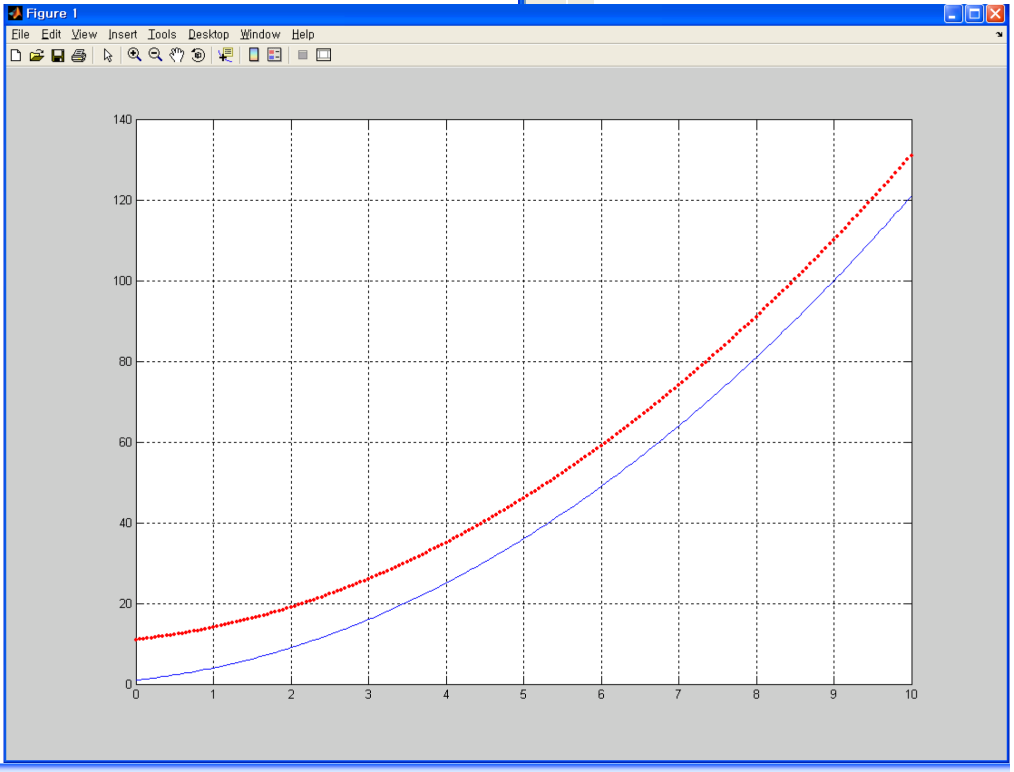
  r* : red star, y- : Yellow bar

- **Grid on/Grid off : Grid 표시 및 숨김**

- **Hold on/Hold ff : 먼저 생성된 그래프 위에 새로운 그래프를 덮어 씌움, 씌우지 않음**



```
x=[0:0.05:10];
y=x.^2+2*x+1;
plot (x,y)
grid on
hold on
z=y+10;
plot(x,z,'r.')
hold off
```



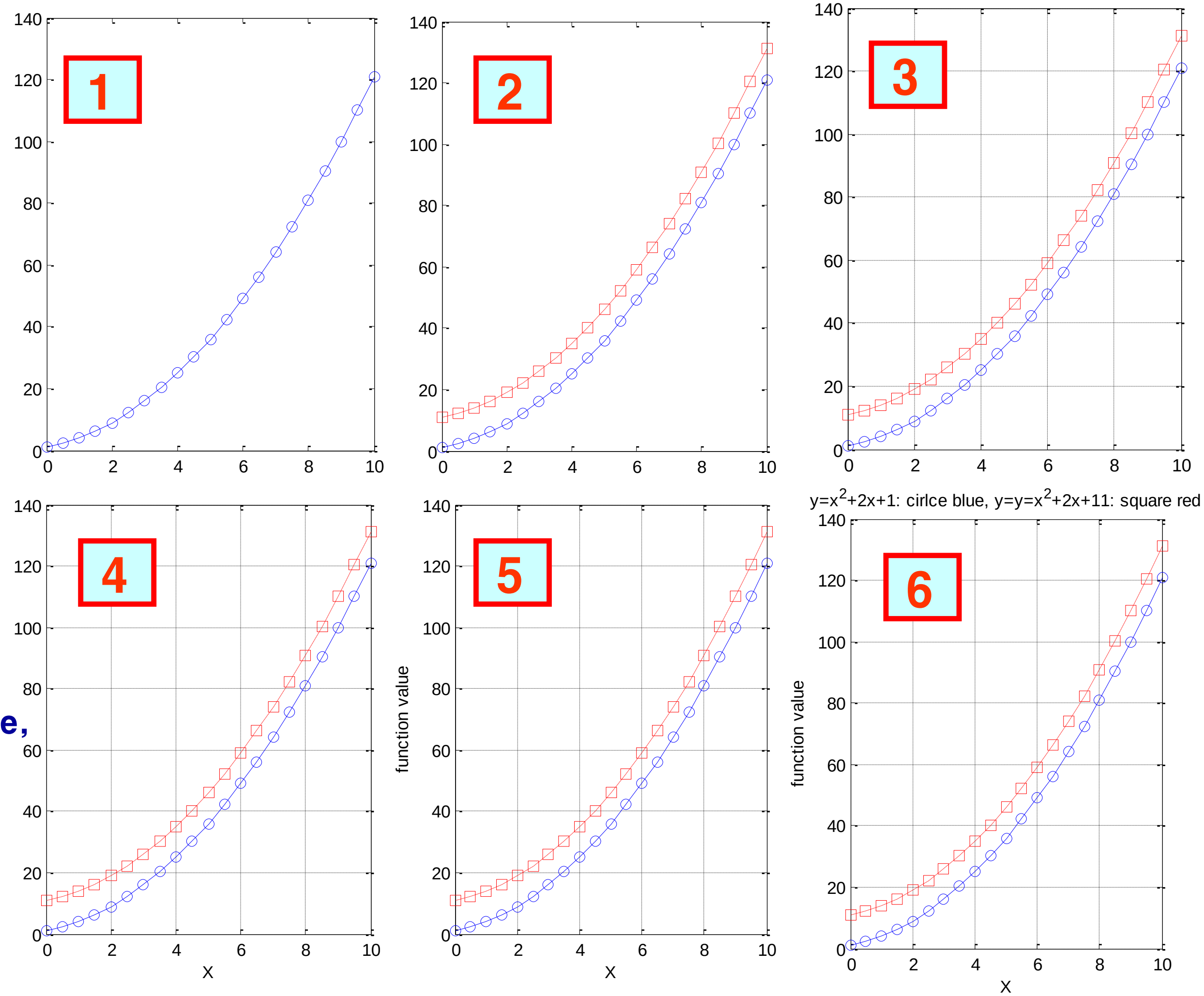
2-D 데이터 plot (2)

- 함수의 Plot 예: 다음의 두 함수를 x=0 에서 x=10까지 구분 그리기

$$y = x^2 + 2x + 1$$

$$y = x^2 + 2x + 11$$

```
>> x=0:0.5:10; % 0.5씩 21구간 나눔
>> y1=x.^2 + 2∗x+1;
>> y2=x.^2 + 2∗x+11;
>> plot(x,y1,'-ob');
>> hold on
>> plot(x,y2,'--sr');
>> grid
>> xlabel('X')
>> ylabel('function value')
>> title('y=x^2+2x+1: cirlce blue,
   y=y=x^2+2x+11: square red')
```



2-D 데이터 plot (3)

- **help 함수: m-file 프로그램의 정보제공**

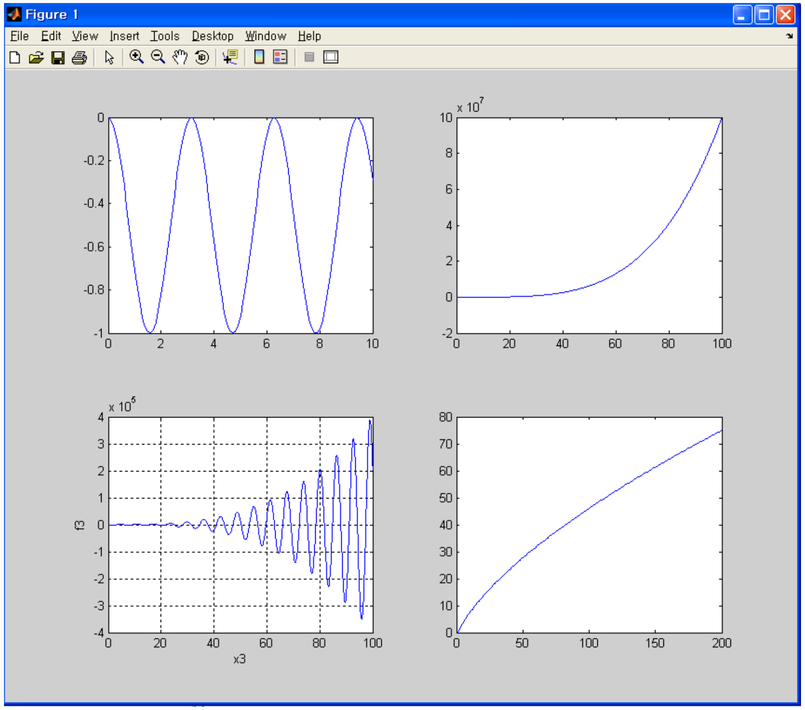
```
>> help plot
 PLOT   Linear plot.
    PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
    then the vector is plotted versus the rows or columns of the matrix,
    whichever line up.  If X is a scalar and Y is a vector, length(Y)
    disconnected points are plotted.

    PLOT(Y) plots the columns of Y versus their index.
    If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
    In all other uses of PLOT, the imaginary part is ignored.

    Various line types, plot symbols and colors may be obtained with
    PLOT(X,Y,S) where S is a character string made from one element
    from any or all the following 3 columns:

           b     blue          .     point              -     solid
           g     green         o     circle             :     dotted
           r     red           x     x-mark             -.    dashdot
           c     cyan          +     plus               --    dashed
           m     magenta       *     star             (none)  no line
           y     yellow        s     square
           k     black         d     diamond
                               v     triangle (down)
                               ^     triangle (up)
                               <     triangle (left)
                               >     triangle (right)
                               p     pentagram
                               h     hexagram
```

2-D 데이터 plot (4)

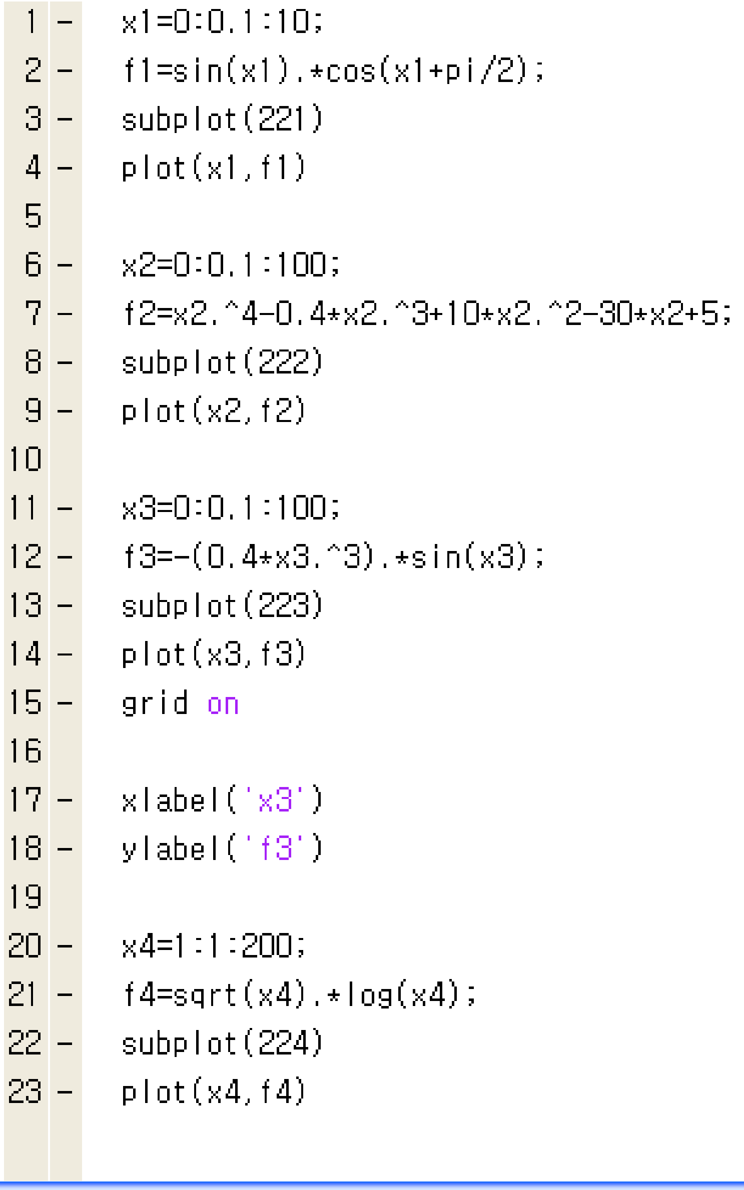
- **Subplot**
  - √ 여러 개의 Plot을 한 화면에 나타낼 때 사용

```
subplot(m,n,i)
     plot(x,y)
```

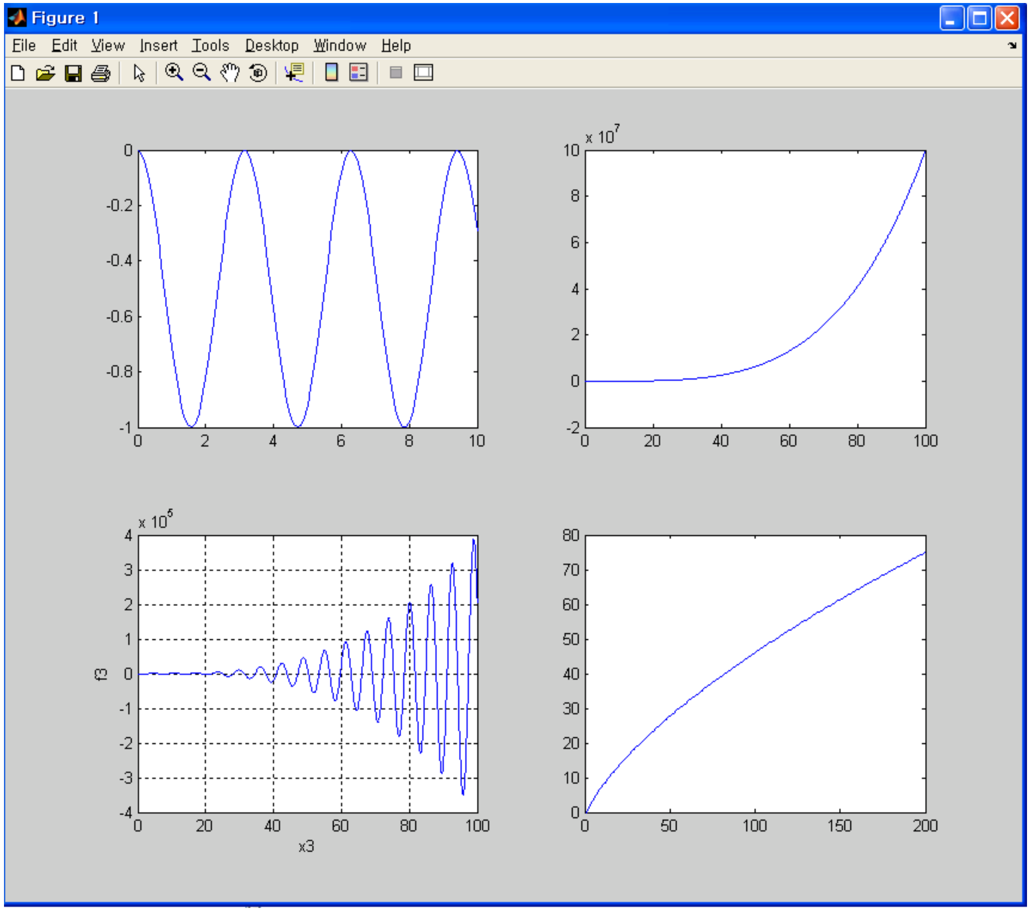
  - √ m X n 개로 나누어진 영역 중 i 번째의 그래프



2-D 데이터 plot (5)



```
x1=0:0.1:10;
f1=sin(x1).*cos(x1+pi/2);
subplot(221)
plot(x1,f1)

x2=0:0.1:100;
f2=x2.^4-0.4*x2.^3+10*x2.^2-30*x2+5;
subplot(222)
plot(x2,f2)

x3=0:0.1:100;
f3=-(0.4*x3.^3).*sin(x3);
subplot(223)
plot(x3,f3)
grid on

xlabel('x3')
ylabel('f3')

x4=1:1:200;
f4=sqrt(x4).*log(x4);
subplot(224)
plot(x4,f4)
```



2-D 데이터 plot (6): Plot Commands

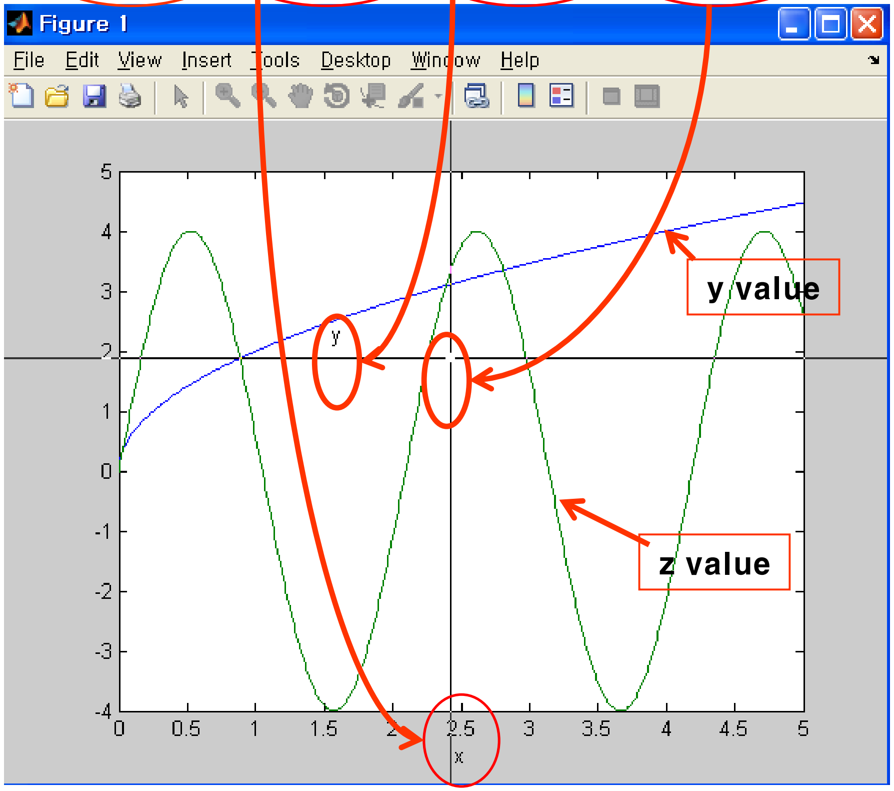
- **Plot Commands**

[x,y] = ginput(n)	Enables the mouse to get *n* points form a plot, and returns the x and y coordinates in the vectors x and y, which have a length *n*.
grid	Puts grid lines on the plot.
gtext (‘text’)	Enables placement of text with the mouse.
plot(x,y)	Generates a plot of the array y versus the array x on rectilinear axes.
title(‘text’)	Puts text in a title at the top of the plot.
xlabel (‘text’)	Adds a text label to the horizontal axis (the abscissa).
ylabel (‘text’)	Adds a text label to the vertical axis (the ordinate).

짝이 되는 변수 끼리 차례대로 적어야 함.

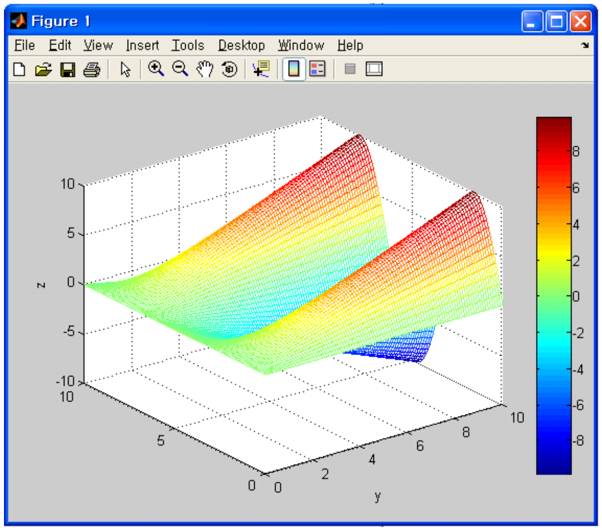
(x,y) 그리고 (x,z)가 짝이 되기 때문에 plot(x,y,x,z)로 표현함

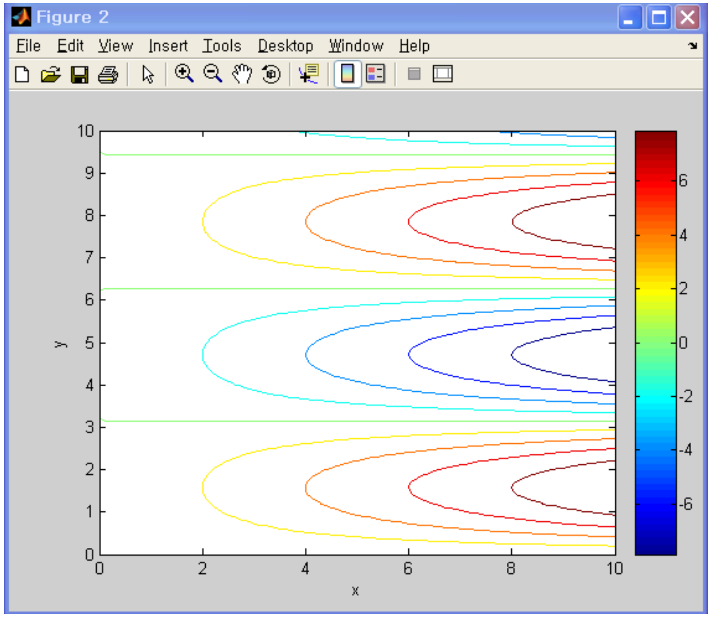
```
>> x = [0 : 0.01 : 5];
>> y = 2*sqrt(x);
>> z = 4*sin(3*x);
>> plot(x,y,x,z),xlabel( 'x' ),gtext( 'y' ),gtext( 'z' )
```

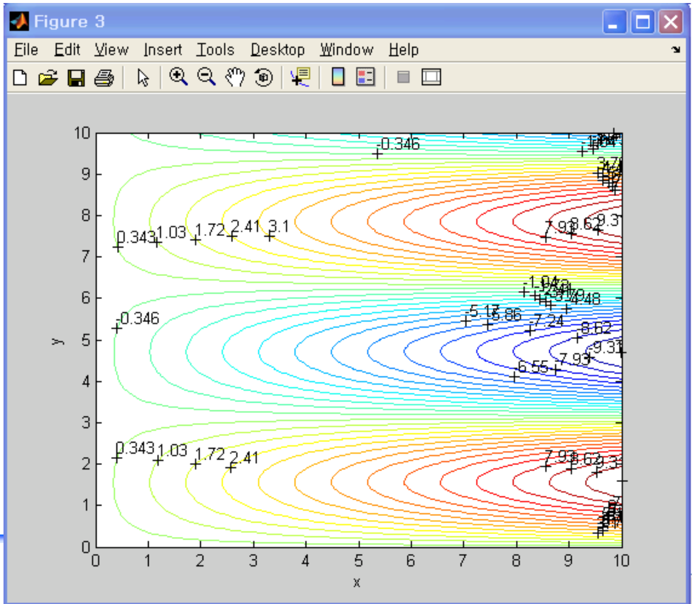


Contour plot (1)

```
x=0:0.1:10;
y=0:0.1:10;
[X,Y]=meshgrid(x,y);
% mesh grid :
% Vector를 행렬로 변환
Z=X.*sin(Y);

figure(1);
mesh(X,Y,Z)
xlabel('x');
xlabel('y');
zlabel('z');
colorbar;
%Color 막대 표현

figure(2);
c=contour(X,Y,Z);
xlabel('x');
ylabel('y');
zlabel('z');
colorbar;

figure(3);
c=contour(X,Y,Z,30);
xlabel('x');
ylabel('y');
zlabel('z');
clabel(c);
%clabel :
%윤곽선 높이 라벨 추가
```







End of Lecture

Appendix A : Matlab Operation & Commands (1)

- Math Operations

Symbol	Operation	MATLAB form
^	exponentiation:	A^b
*	multiplication:	a*b
/	right division:	a/b
\	left division:	a\b
+	addition:	a+b
-	subtraction:	a-b

- Special Numbers

ans	Temporary variable containing the most recent answer
eps	Specifies the accuracy of floating point precision
i ,j	The imaginary unit
Inf	Infinity
NaN	Indicates an undefined numerical result
pi	The number: = 3.141592….

- Math Functions

e^x	exp(x)		
$\sqrt{x}$	sqrt(x)	$\sin^{-1}x$	asin(x)
ln x	log(x)	$\tan^{-1}x$	atan(x)
$\log_{10}x$	log10(x)		
cos x	cos(x)		
sin x	sin (x)		
tan x	tan(x)		
$\cos^{-1}x$	acos(x)		

Appendix A : Matlab Operation & Commands (2)

- Directory and Path Handler

addpath *dirname*	Adds the directory *dirname* to the search path
cd *dirname*	Changes the current directory to *dirname*
dir	Lists all files in the current directory
dir *dirname*	Lists all the files in the directory *dirname*
path	Displays the MATLAB search path
pathtool	Starts the Set Path tool
pwd	Displays the current directory
rmpath *dirname*	Removes the directory *dirname* from the search path
what	Lists the MATLAB-specific files found in the current working directory. Most data files and other non-MATLAB files are not listed. Use *dir* to get a list of all files
what *dirname*	Lists the MATLAB-specific files in directory *dirname*

Appendix A: Matlab Operation & Commands (3)

- Input/output commands

disp(A)	Displays the contents, but not the name, of the array A.
disp(‘text’)	Displays the text string enclosed within single quotes.
format	Controls the screen’s output display format
fprintf	Performs formatted writes to the screen or to a file
x = input(‘text’)	Displays the text in quotes, waits for user input from the keyboard, and stores the value in x.
x = input(‘text’, ‘s’)	Displays the text in quotes, waits for user input from the keyboard, and stores the input as a string in x
k = menu (‘title’, ‘option1’, ‘option2’, …)	Displays a menu whose title is in the string variable ‘title’, and whose choices are ‘option1’, ‘option2’, and so on.

Appendix B: Array Functions

▪ Some Useful Array Functions

Command	Description
cat(n, A, B, C, …)	Creates a new array by concatenating the arrays A,B,C, and so on along the dimension n.
find(x)	Computes an array containing the indices of the nonzero elements of the array x.
[u, v, w] = find(A)	Computes the arrays u and v, containing the row and column indices of the nonzero elements of the matrix A, and the array w, containing the values of the nonzero elements. The array w may be omitted.
length(A)	Computes either the number of elements of A if A is a vector or the largest value of m or n if A is an m × n matrix
linspace(a, b, n)	Creates a row vector of n regularly spaced values between a and b
logspace(a, b, n)	Creates a row vector of n logarithmically spaced values between a and b
max(A)	Returns the algebraically largest element in A if A is a vector. Returns a row vector containing the largest elements in each column if A is a matrix. If any of the elements are complex, max(A) returns the elements that have the largest magnitudes.
[x, k] = max(A)	Similar to max(A) but stores the maximum values in the row vector x and their indices in the row vector k
min(A)	Same as max(A) but returns minimum values.
[x, k] = min(A)	Same as [x, k]=max(A) but returns minimum values.
size(A)	Returns a row vector [m,n] containing the sizes of the m × n array A.
sort(A)	Sorts each column of the array A in ascending order and returns an array the same size as A.
sum(A)	Sums the elements in each column of the array A and returns a row vector containing the sums

Appendix B: Array Operations

▪ Element-by-element operations

Symbol	Operation	Form	Example
+	Scalar – array addition	A + b	[6,3] + 2 = [8,5]
-	Scalar – array subtraction	A – b	[8,3] – 5 = [3,-2]
+	Array addition	A + B	[6,5] + [4,8] = [10,13]
-	Array subtraction	A – B	[6,5] – [4.8] = [2,-3]
.*	Array multiplication	A.*B	[3,5] .* [4,8] = [12,40]
./	Array right division	A./B	[2,5] ./ [4,8] = [2/4, 5/8]
.\	Array left division	A.\B	[2,5] .\ [4,8] = [2\4, 5\8]
.^	Array exponentiation	A.^B	[3,5].^2 = [3^2, 5^2]
			2.^[3,5] = [2^3, 2^5]
			[3,5] .^ [2,4] = [3^2, 5^4]