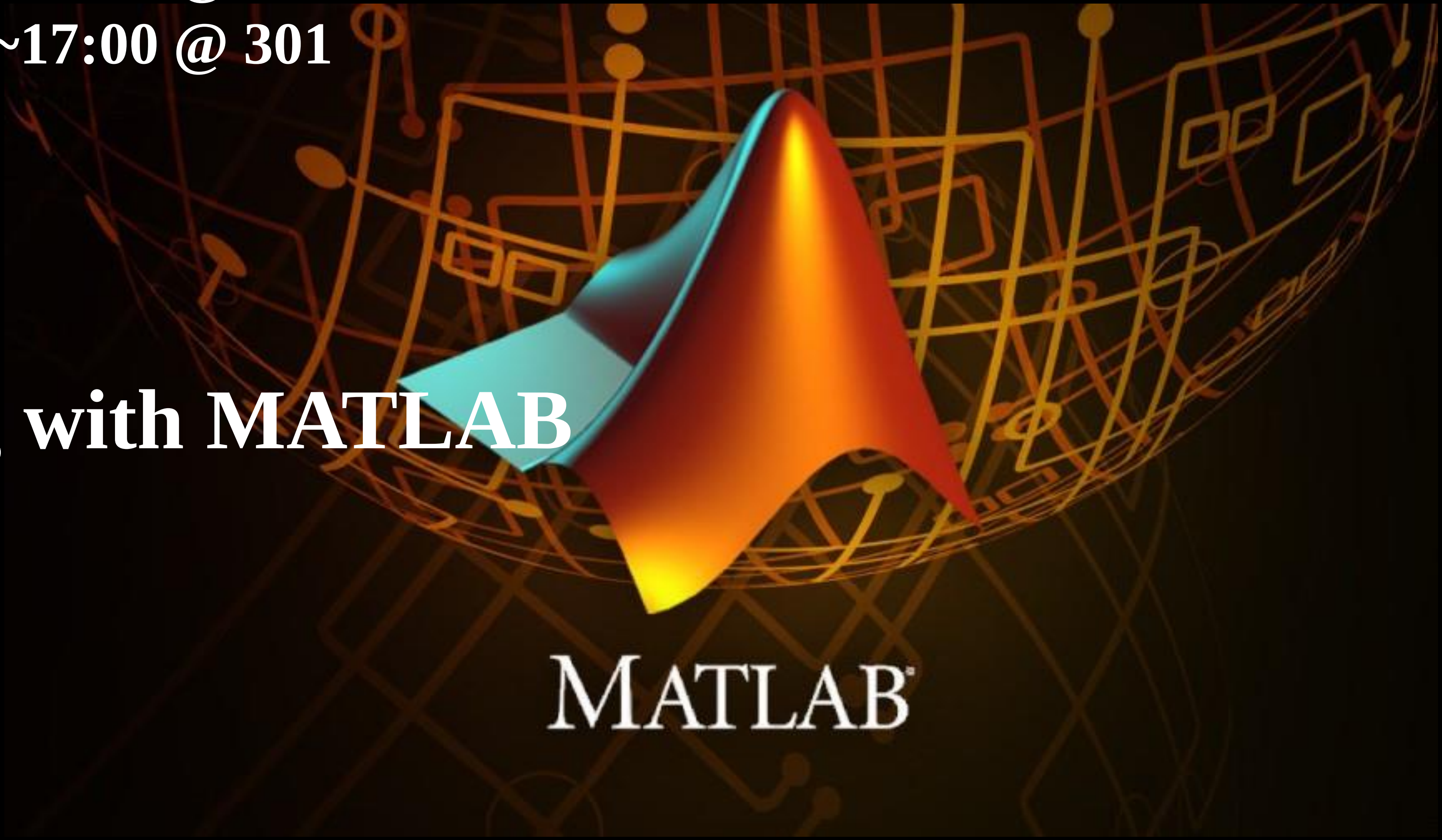
Introduction to MATLAB
Mon. 15:00~17:00 @ 301
Wed. 15:00~17:00 @ 301

02.
Starting with MATLAB



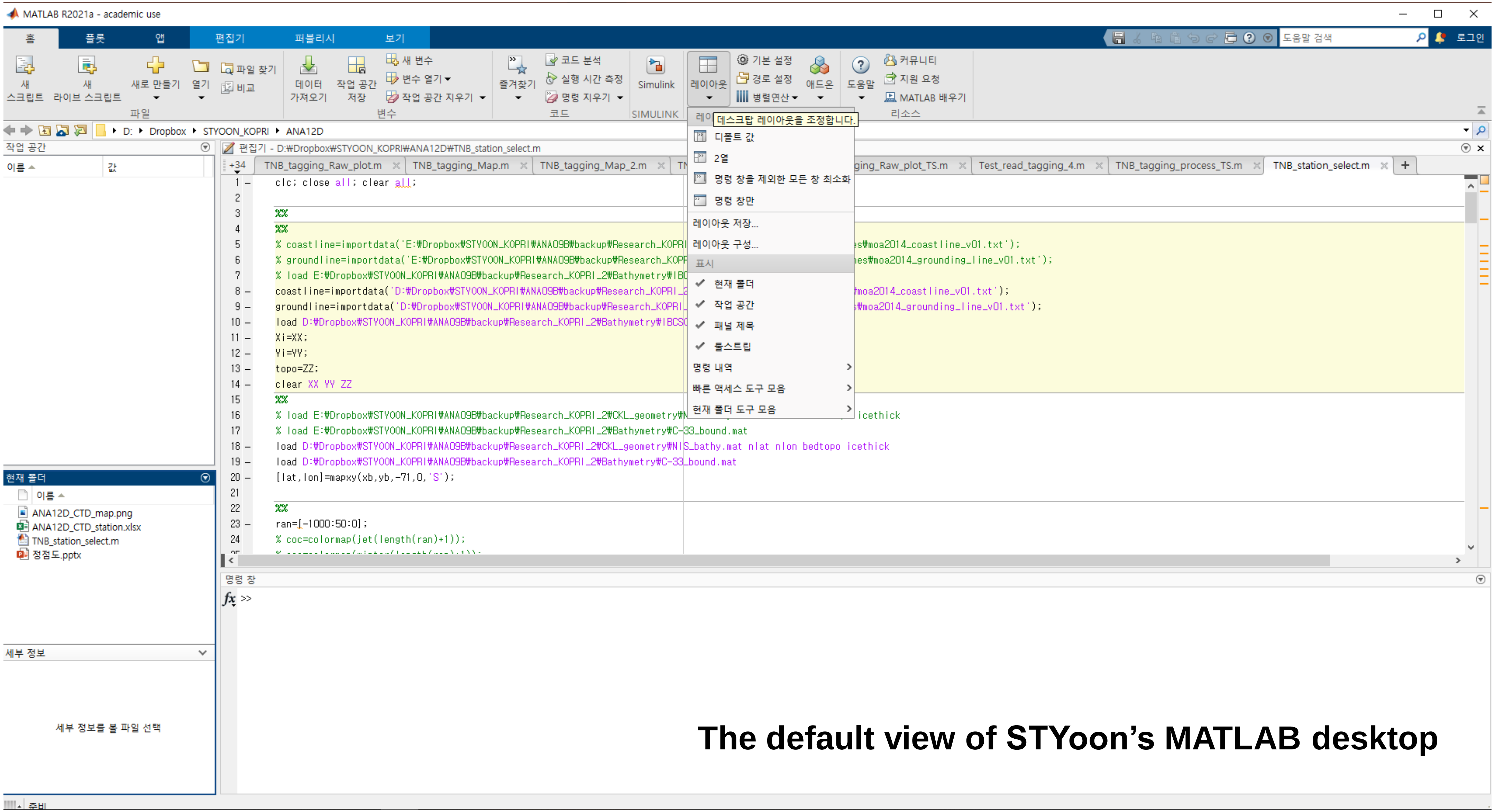
Seung-Tae Yoon

Department of Oceanography, KNU

※ **In this week,**

- ✓Learn about a kind of layouts in MATLAB
- ✓Use MATLAB as a calculator (using the built-in functions)
- ✓Make script files

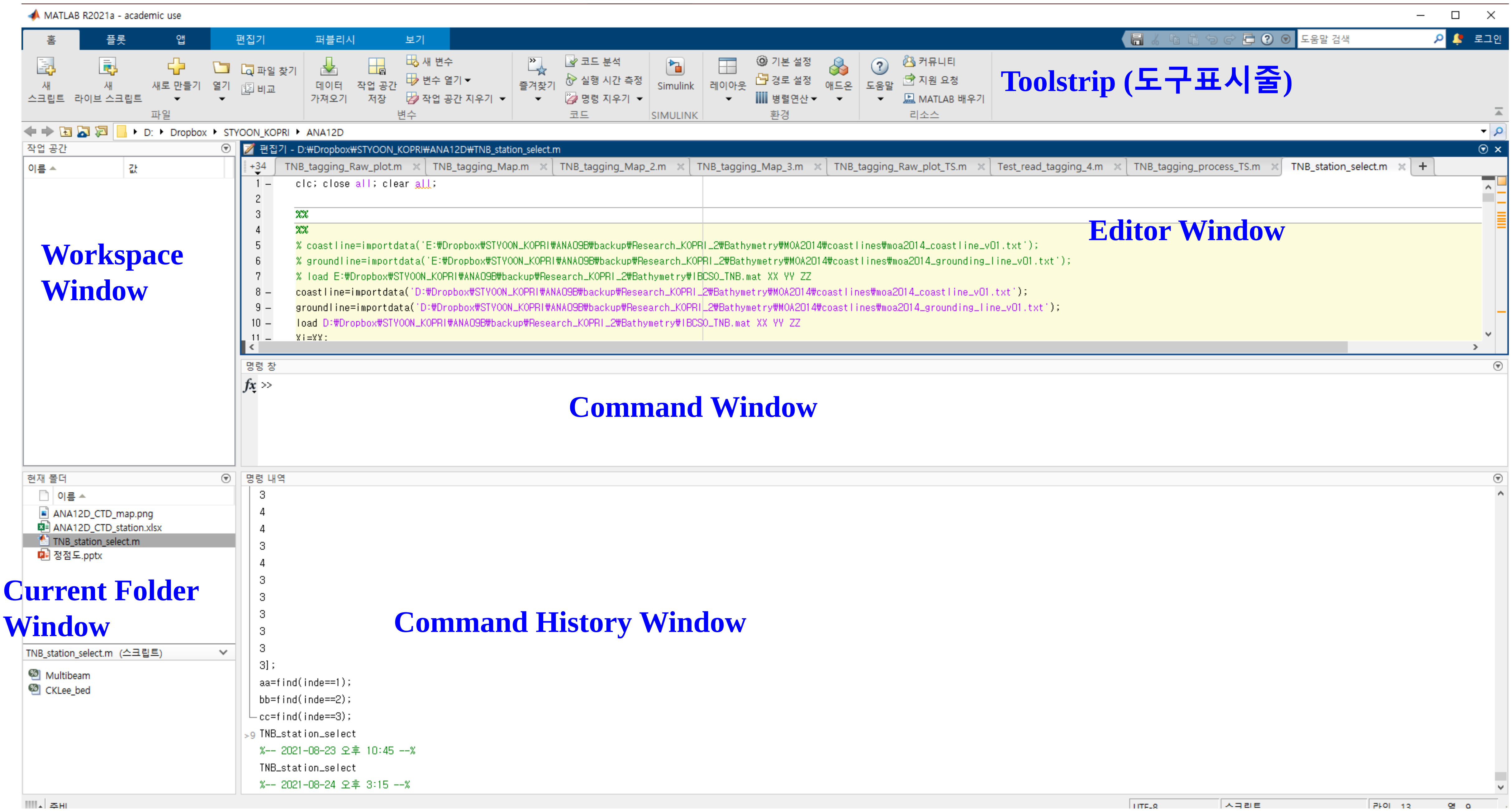
1.1 Starting MATLAB, MATLAB Windows



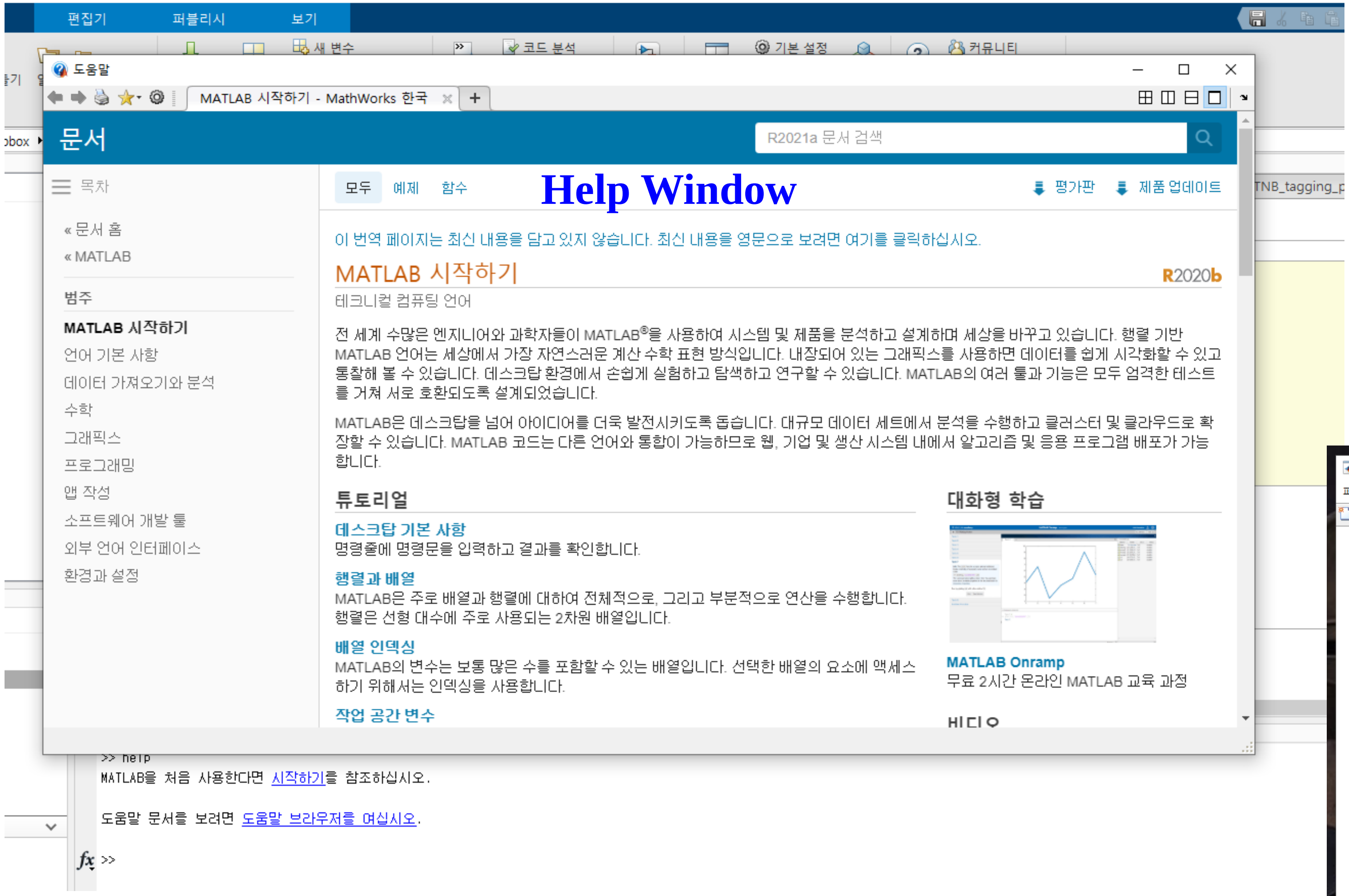
The default view of STYoon’s MATLAB desktop

Window	(Korean)	Purpose
Command Window	명령 창	Main window, enters variables, runs programs.
Workspace Window	작업공간	Provides information about the variables that are stored.
Current Folder Window (Detailed information)	현재 폴더 (세부정보)	Shows the files in the current folder.
Editor Window	편집기	Creates and debugs script and function files.
Command History Window	명령내역	Logs commands entered in the Command Window.
Figure Window	그림창	Contains output from graphic commands.
Help Window	도움말	Provides help information.

※ debugging: In computer programming and software development, debugging is the process of finding and resolving bugs within computer programs, software, or systems.

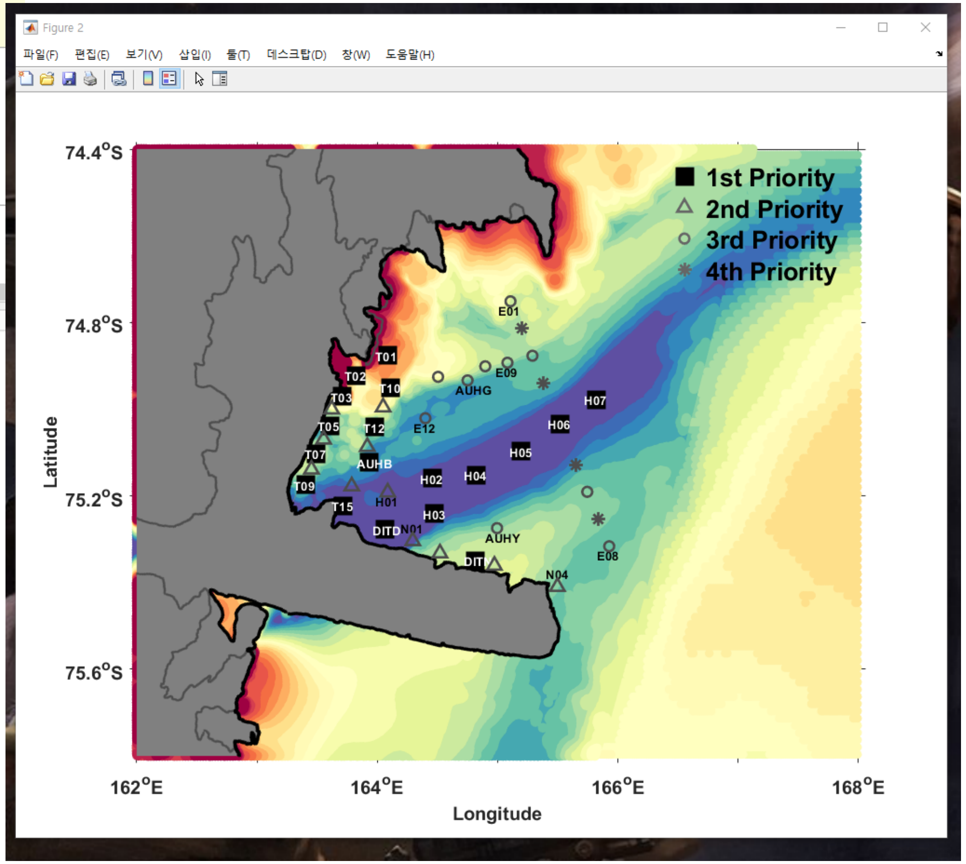


MATLAB R2021a - academic use
홈
플롯
앱
편집기
퍼블리시
보기
Toolstrip (도구표시줄)
D: Dropbox STYOON_KOPRI ANA12D
작업 공간
Workspace Window
편집기 - D:₩Dropbox₩STYOON_KOPRI₩ANA12D₩TNB_station_select.m
Editor Window
clc; close all; clear all;
명령 창
Command Window
현재 폴더
ANA12D_CTD_map.png
ANA12D_CTD_station.xlsx
TNB_station_select.m
정점도.pptx
Current Folder Window
TNB_station_select.m (스크립트)
Multibeam
CKLee_bed
명령 내역
aa=find(inde==1);
bb=find(inde==2);
cc=find(inde==3);
TNB_station_select
%-- 2021-08-23 오후 10:45 --%
TNB_station_select
%-- 2021-08-24 오후 3:15 --%
Command History Window

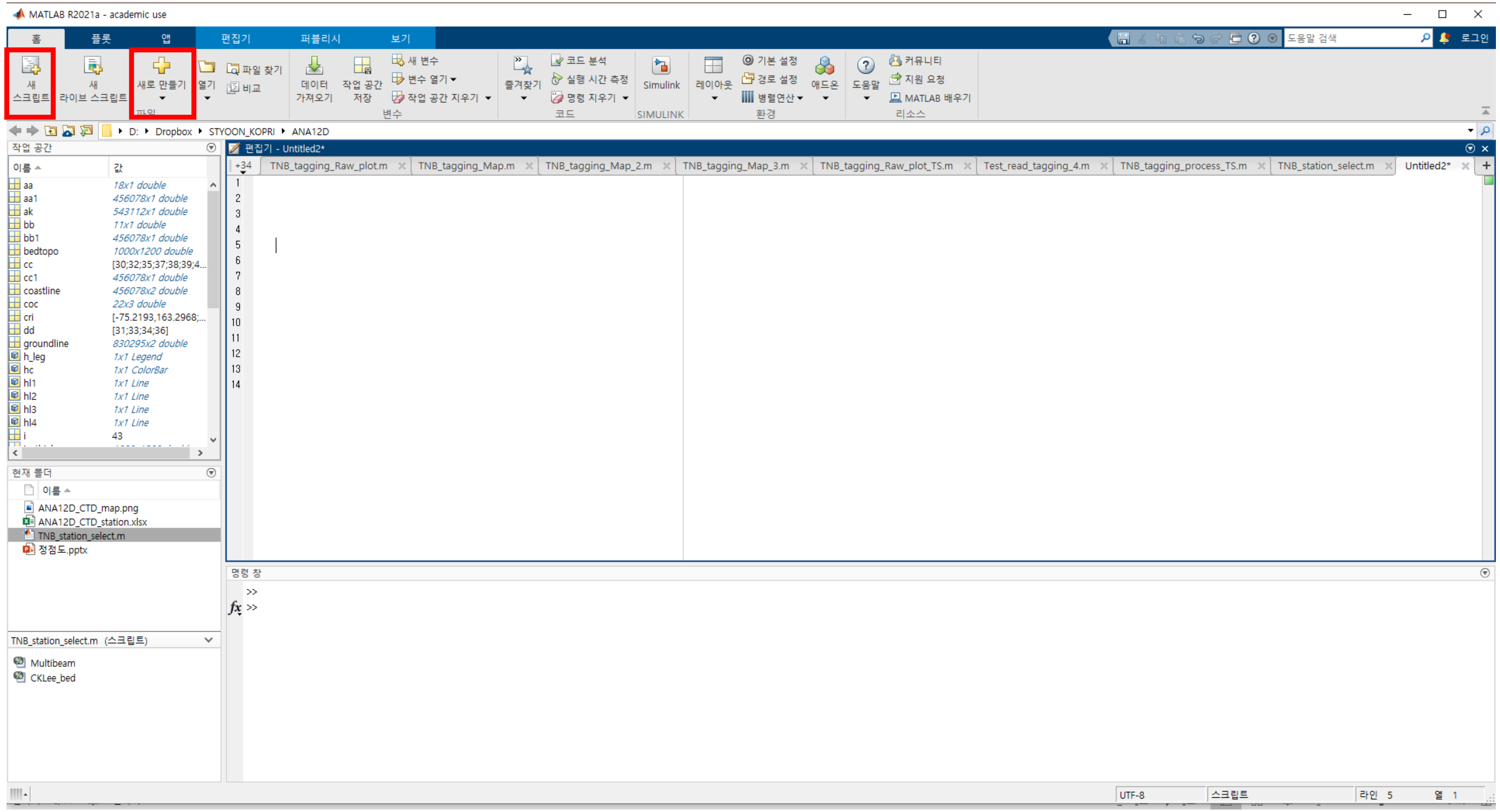


>> help help

Figure Window

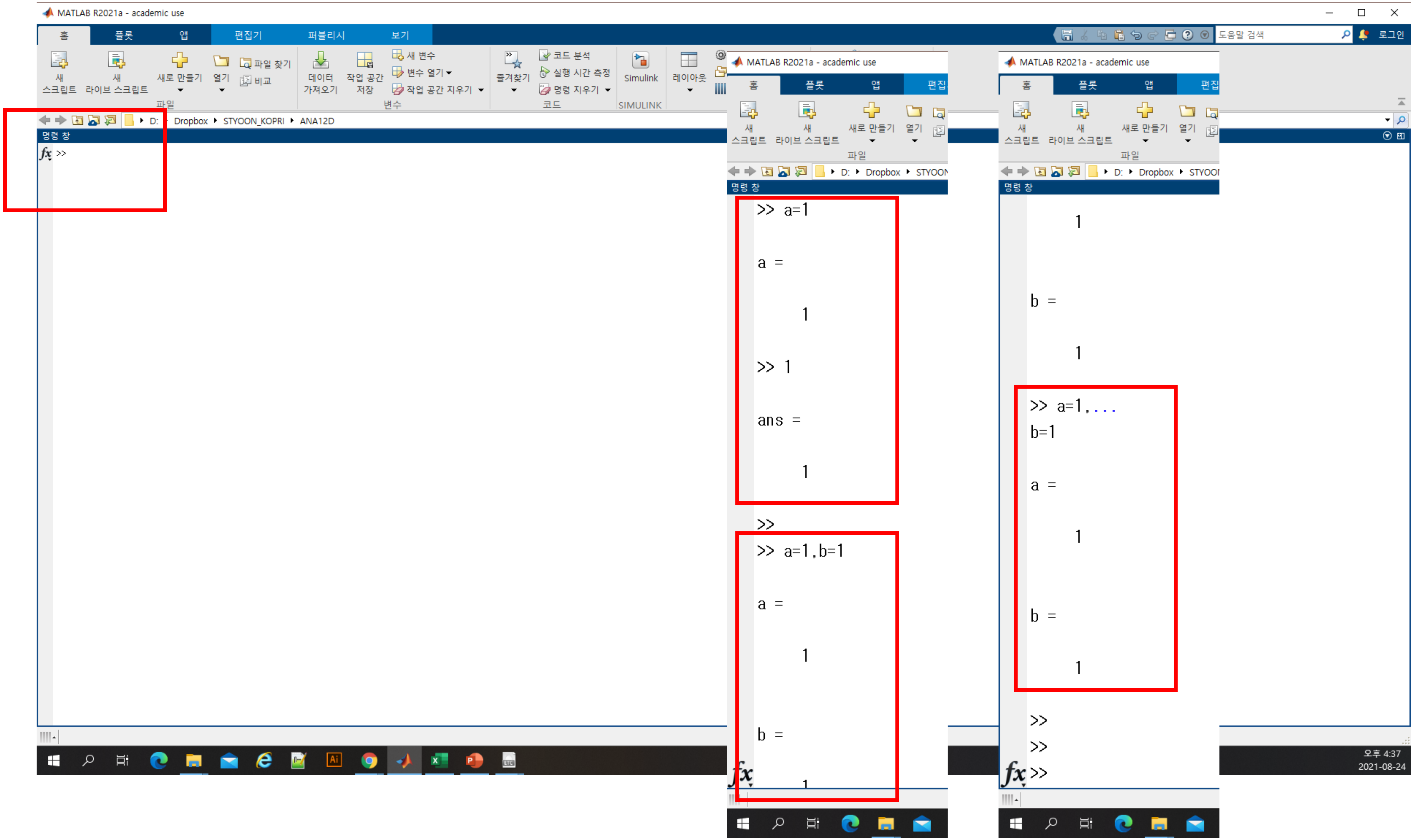


- ***Editor Window***: The Editor Window is used for writing and editing programs. This Window is opened by clicking on the New Script icon in the Toolstrip or by clicking on the New icon and then selecting Script from the menu that opens.



1.2 Working in the Command Window

- To type a command the cursor is placed next to the command prompt (>>).
- Once a command is typed and the **Enter** key is pressed, the command is executed. However, only the last command is executed. Everything executed previously (that might be still displayed) is unchanged.
- Several commands can be typed in the same line. This is done by typing a comma (,) between the commands. When the **Enter** key is pressed, the commands are executed in order from left to right.
- It is not possible to go back to a previous line that is displayed in the Command Window, make a correction, and then re-execute the command.
- A previously typed command can be recalled to the command prompt with the up-arrow key (↑). When the command is displayed at the command prompt, it can be modified if needed and then executed. The down-arrow key (↓) can be used to move down the list of previously typed commands.
- If a command is too long to fit in one line, it can be continued to the next line by typing three periods … (called an ellipsis) and pressing the **Enter** key. The continuation of the command is then typed in the new line. The command can continue line after line up to a total of 4,096 characters.

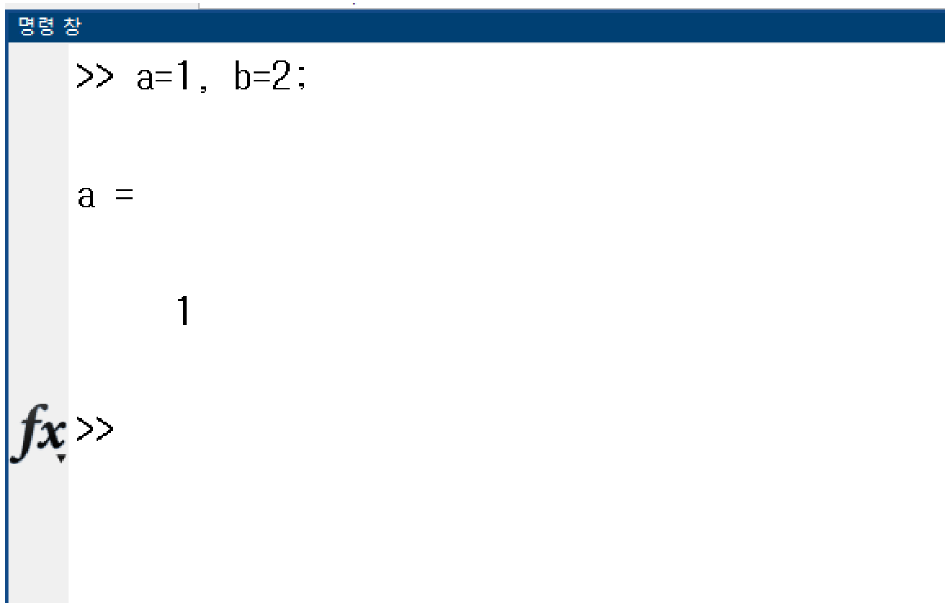

MATLAB R2021a - academic use
>>
>> a=1
a =
1
>> 1
ans =
1
>>
>> a=1,b=1
a =
1
b =
1
1
b =
1
>> a=1,...
b=1
a =
1
b =
1
>>
>>
>>

▪ The semicolon (;)

: When a command is typed in the Command Window and the **Enter** key is pressed, the command is executed. Any output that the command generates is displayed in the Command Window. If a semicolon (;) is typed at the end of a command, the output of the command is not displayed.

: Typing a semicolon is useful when the result is obvious or known, or when the output is very large.

: If several commands are typed in the same line, the output from any of the commands will not be displayed if a semicolon instead of a comma is typed between the commands.



▪ Typing %

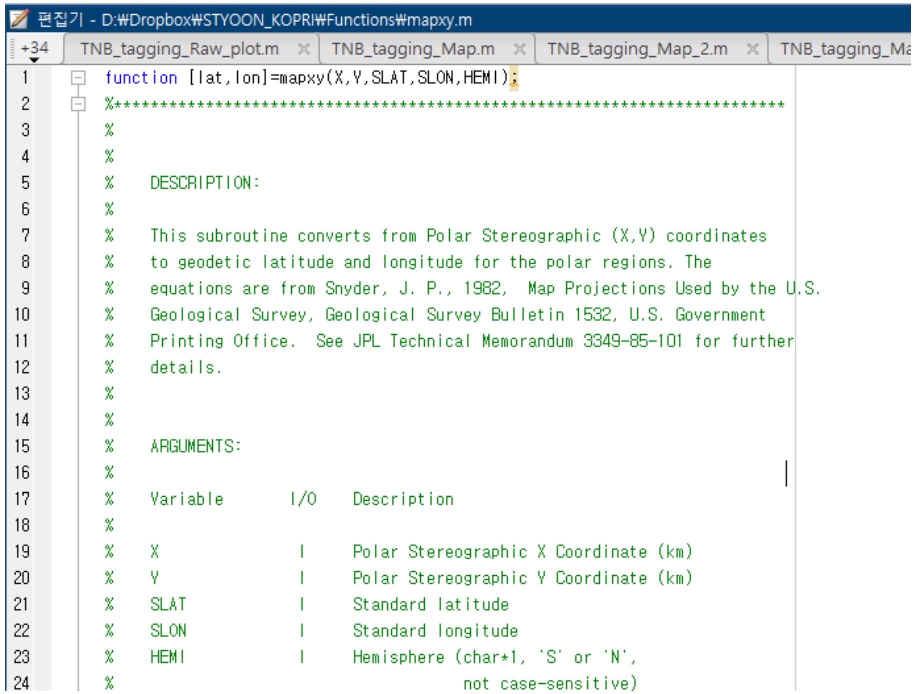
: When the symbol % (percent) is typed at the beginning of a line, the line is designated as a comment. This means that when the **Enter** key is pressed the line is not executed.

: The % character followed by text (comment) can also be typed after a command (in the same line). This has no effect on the execution of the command.

: Usually there is no need for comments in the Command Window. Comments, however, are frequently used in a program to add descriptions or to explain the program (see Chapters 4 and 6).

: **shift + 5**; **ctrl + r**

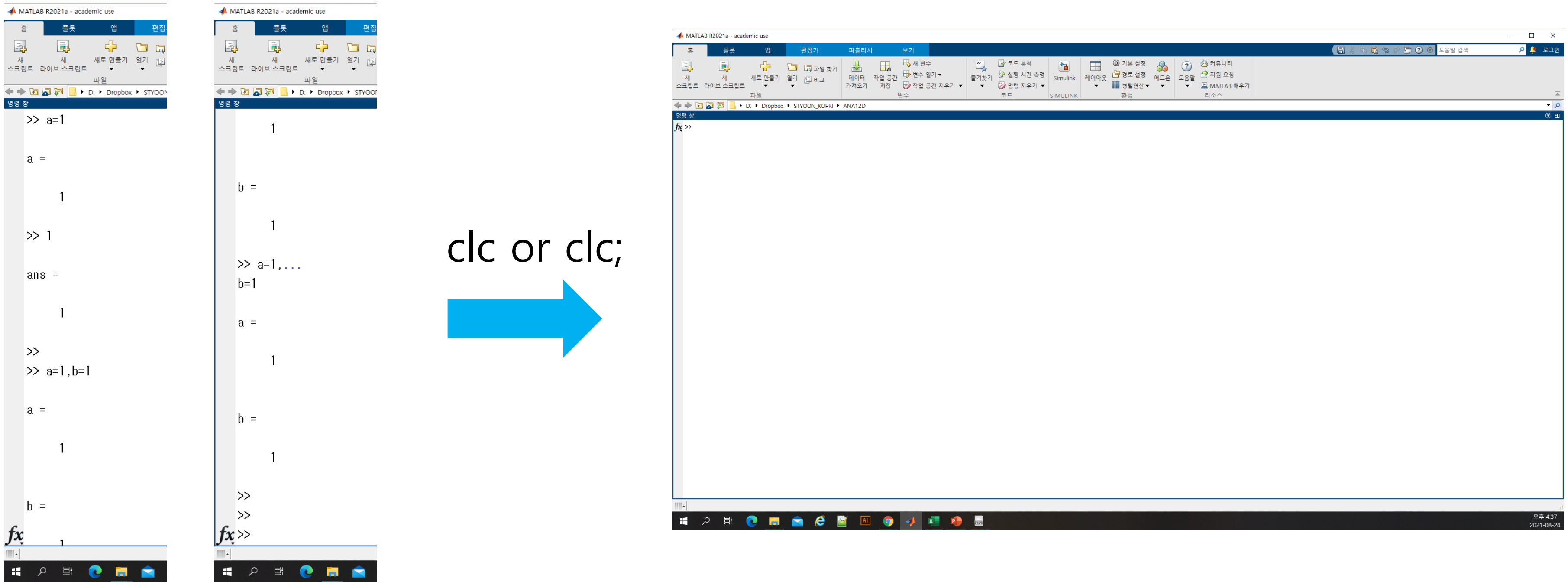
: **ctrl + t**



▪ **The** clc **command:**

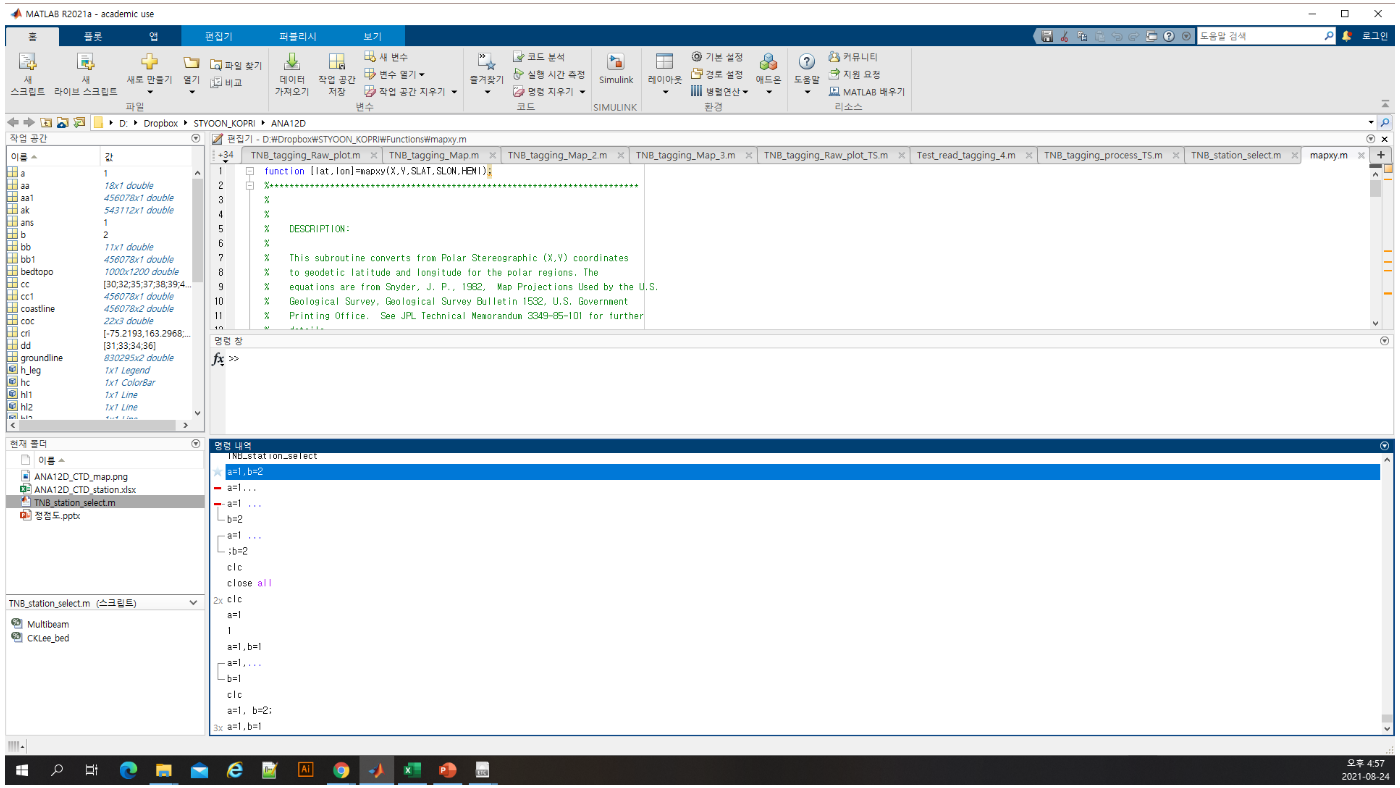
: The clc command (type clc and press **Enter**) clears the Command Window. After typing in the Command Window for a while, the display may become very long. Once the clc command is executed, a clear window is displayed.

: The command does not change anything that was done before. For example, if some variables were defined previously (see Section 1.6), they still exist and can be used. The up-arrow key (↑) can also be used to recall commands that were typed before.



▪ The Command History Window

: The Command History Window lists the commands that have been entered in the Command Window. This includes commands from previous sessions. A command in the Command History Window can be used again in the Command Window. By double-clicking on the command, the command is reentered in the Command Window and executed. It is also possible to drag the command to the Command Window, make changes if needed, and then execute it. The list in the Command History Window can be cleared by selecting the lines to be deleted and then right-clicking the mouse and selecting Delete Selection. The whole history can be deleted by right-clicking the mouse and selecting choose Clear Command History in the menu that opens.



1.3 Arithmetic Operations with scalars

Operation	Symbol	Example
Addition	+	5 + 3
Subtraction	–	5 – 3
Multiplication	*	5 * 3
Right division	/	5 / 3
Left division	\	5 \ 3 = 3 / 5
Exponentiation	^	5 ^ 3 (means $5^3 = 125$)

5/3=3₩5

1.3.1 Order of precedence

Precedence	Mathematical Operation
First	Parentheses. For nested parentheses, the innermost are executed first.
Second	Exponentiation.
Third	Multiplication, division (equal precedence).
Fourth	Addition and subtraction.

※ The first four terms of the Taylor series for sin(pi/4)

$$\sin x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

Tutorial 1-1: Using MATLAB as a calculator.

```
>> 7+8/2
ans =
    11
>> (7+8)/2
ans =
    7.5000
>> 4+5/3+2
ans =
    7.6667
>> 5^3/2
ans =
   62.5000
>> 27^(1/3)+32^0.2
ans =
     5
>> 27^1/3+32^0.2
ans =
    11
>> 0.7854-(0.7854)^3/(1*2*3)+0.785^5/(1*2*3*4*5)...
-(0.785)^7/(1*2*3*4*5*6*7)
ans =
    0.7071
>>
```

Type and press **Enter**.

8/2 is executed first.

Type and press **Enter**.

7+8 is executed first.

5/3 is executed first.

5^3 is executed first, /2 is executed next.

1/3 is executed first, 27^(1/3) and 32^0.2 are executed next, and + is executed last.

27^1 and 32^0.2 are executed first, /3 is executed next, and + is executed last.

Type three periods ... (and press **Enter**) to continue the expression on the next line.

The last expression is the first four terms of the Taylor series for sin(π/4).

1.4 Display Formats

- The user can control the format in which MATLAB displays output on the screen.

Command	Description	Example

`format short`	Fixed-point with 4 decimal digits for: $0.001 \le number \le 1000$ Otherwise display format `short e`.	`>> 290/7`
		`ans =`
		`41.4286`

`format long`	Fixed-point with 15 decimal digits for: $0.001 \le number \le 100$ Otherwise display format `long e`.	`>> 290/7`
		`ans =`
		`41.428571428571431`

`format short e`	Scientific notation with 4 decimal digits.	`>> 290/7`
		`ans =`
		`4.1429e+001`

`format long e`	Scientific notation with 15 decimal digits.	`>> 290/7`
		`ans =`
		`4.142857142857143e+001`

`format short g`	Best of 5-digit fixed or floating point.	`>> 290/7`
		`ans =`
		`41.429`

`format long g`	Best of 15-digit fixed or floating point.	`>> 290/7`
		`ans =`
		`41.4285714285714`

`format bank`	Two decimal digits.	`>> 290/7`
		`ans =`
		`41.43`

`format compact`	Eliminates blank lines to allow more lines with information displayed on the screen.	
`format loose`	Adds blank lines (opposite of `compact`).	

```
>> format compact
>> 290/7
ans =
   41.4286
>> format loose
>> 290/7

ans =

   41.4286
```

1.5 Elementary Math Built-in Functions

- In addition to basic arithmetic operations, expressions in MATLAB can include functions. MATLAB has a very large library of built-in functions. A function has a name and an argument in parentheses.

명령 창

```
>> format compact
>> sqrt(64)
ans =
     8
>> sqrt(50+14*3)
ans =
    9.5917
>> sqrt(54+9*sqrt(100))
ans =
    12
>> (15+600/4)/sqrt(121)
ans =
    15
>> 
```

Ex)

Function	Description	Example

`sqrt(x)`	Square root.	`>> sqrt(81)`
		`ans =`
		`9`

`nthroot(x,n)`	Real nth root of a real number x. (If x is negative n must be an odd integer.)	`>> nthroot(80,5)`
		`ans =`
		`2.4022`

`exp(x)`	Exponential (e^x).	`>> exp(5)`
		`ans =`
		`148.4132`

Function	Description	Example

`abs(x)`	Absolute value.	`>> abs(-24)`
		`ans =`
		`24`

`log(x)`	Natural logarithm.
	Base e logarithm (ln).
		`>> log(1000)`
		`ans =`
		`6.9078`

`log10(x)`	Base 10 logarithm.	`>> log10(1000)`
		`ans =`
		`3.0000`

`factorial(x)`	The factorial function $x!$
	(x must be a positive integer.)
		`>> factorial(5)`
		`ans =`
		`120`

명령 창

```
>> (80)^(1/5)
ans =
    2.4022
>> (80)^1/5
ans =
    16
fx >>
```

※ The number e, also known as Euler's number, is a mathematical constant approximately equal to 2.71828, and can be characterized in many ways.

※ Help & Edit

편집기
퍼블리시
보기
새 변수
코드 분석
기본 설정
커뮤니티
도움말
MATLAB 시작하기 - MathWorks 한국
문서
R2021a 문서 검색
목차
« 문서 홈
« MATLAB
범주
MATLAB 시작하기
언어 기본 사항
데이터 가져오기와 분석
수학
그래픽스
프로그래밍
앱 작성
소프트웨어 개발 툴
외부 언어 인터페이스
환경과 설정
모두
예제
함수
평가판
제품 업데이트
이 번역 페이지는 최신 내용을 담고 있지 않습니다. 최신 내용을 영문으로 보려면 여기를 클릭하십시오.
MATLAB 시작하기
R2020b
테크니컬 컴퓨팅 언어
전 세계 수많은 엔지니어와 과학자들이 MATLAB®을 사용하여 시스템 및 제품을 분석하고 설계하며 세상을 바꾸고 있습니다. 행렬 기반 MATLAB 언어는 세상에서 가장 자연스러운 계산 수학 표현 방식입니다. 내장되어 있는 그래픽스를 사용하면 데이터를 쉽게 시각화할 수 있고 통찰해 볼 수 있습니다. 데스크탑 환경에서 손쉽게 실험하고 탐색하고 연구할 수 있습니다. MATLAB의 여러 툴과 기능은 모두 엄격한 테스트를 거쳐 서로 호환되도록 설계되었습니다.
MATLAB은 데스크탑을 넘어 아이디어를 더욱 발전시키도록 돕습니다. 대규모 데이터 세트에서 분석을 수행하고 클러스터 및 클라우드로 확장할 수 있습니다. MATLAB 코드는 다른 언어와 통합이 가능하므로 웹, 기업 및 생산 시스템 내에서 알고리즘 및 응용 프로그램 배포가 가능합니다.
튜토리얼
데스크탑 기본 사항
명령줄에 명령문을 입력하고 결과를 확인합니다.
행렬과 배열
MATLAB은 주로 배열과 행렬에 대하여 전체적으로, 그리고 부분적으로 연산을 수행합니다. 행렬은 선형 대수에 주로 사용되는 2차원 배열입니다.
배열 인덱싱
MATLAB의 변수는 보통 많은 수를 포함할 수 있는 배열입니다. 선택한 배열의 요소에 액세스하기 위해서는 인덱싱을 사용합니다.
작업 공간 변수
대화형 학습
MATLAB Onramp
무료 2시간 온라인 MATLAB 교육 과정
>> help
MATLAB을 처음 사용한다면 시작하기를 참조하십시오.
도움말 문서를 보려면 도움말 브라우저를 여십시오.
>>

※ **Base rule of logarithm change**

$$\log_b(a) = \frac{\log_x(a)}{\log_x(b)}$$

명령 창

```
>> log(exp(1))
ans =
      1
>> log2(2)
ans =
      1
>> log10(10)
ans =
      1
>> aa=log10(9)/log10(3)
aa =
      2
>>
```

▶ **log3(9)**

Ex)

※ One radian is defined as the angle subtended from the center of a circle which intercepts an arc equal in length to the radius of the circle.

$$1\ \text{rad} = 1 \cdot \frac{180^\circ}{\pi} \approx 57.2958^\circ$$

$$2.5\ \text{rad} = 2.5 \cdot \frac{180^\circ}{\pi} \approx 143.2394^\circ$$

$$\frac{\pi}{3}\ \text{rad} = \frac{\pi}{3} \cdot \frac{180^\circ}{\pi} = 60^\circ$$

Function	Description	Example

`sin(x)`
`sind(x)`
	Sine of angle x (x in radians).
	Sine of angle x (x in degrees).
		`>> sin(pi/6)`
		`ans =`
		`0.5000`

`cos(x)`
`cosd(x)`
	Cosine of angle x (x in radians).
	Cosine of angle x (x in degrees).
		`>> cosd(30)`
		`ans =`
		`0.8660`

`tan(x)`
`tand(x)`
	Tangent of angle x (x in radians).
	Tangent of angle x (x in degrees).
		`>> tan(pi/6)`
		`ans =`
		`0.5774`

`cot(x)`
`cotd(x)`
	Cotangent of angle x (x in radians).
	Cotangent of angle x (x in degrees).
		`>> cotd(30)`
		`ans =`
		`1.7321`

```
>> cos(60)
ans =
   -0.9524
>> cos(pi/3)
ans =
    0.5000
>> cosd(pi/3)
ans =
    0.9998
>> cosd(60)
ans =
    0.5000
fx >>
```

- The inverse trigonometric functions are asin(x), acos(x), atan(x), acot(x) for the angle in radians; and asind(x), acosd(x), atand(x), acotd(x) for the angle in degrees.
- pi is equal to π.

- The hyperbolic trigonometric functions are sinh(x), cosh(x), tanh(x), and coth(x).

(1) $\sinh x = \dfrac{e^x - e^{-x}}{2}$,[Read as 'hyperbolic sine x']

(2) $\cosh x = \dfrac{e^x + e^{-x}}{2}$,[Read as 'hyperbolic cosine x']

(3) $\tanh x = \dfrac{\sinh x}{\cosh x} = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

(4) $\coth x = \dfrac{\cosh x}{\sinh x} = \dfrac{e^x + e^{-x}}{e^x - e^{-x}}$

(5) $\operatorname{cosech} x = \dfrac{1}{\sinh x} = \dfrac{2}{e^x - e^{-x}}$

(6) $\sec hx = \dfrac{1}{\cosh x} = \dfrac{2}{e^x + e^{-x}}$

- $\sinh 0 = 0, \quad \cosh 0 = 1, \quad \tanh 0 = 0$

Ex)

Function	Description	Example

`round(x)`	Round to the nearest integer.	`>> round(17/5)`
		`ans =`
		`3`

Function	Description	Example

`fix(x)`	Round toward zero.	`>> fix(13/5)`
		`ans =`
		`2`

`ceil(x)`	Round toward infinity.	`>> ceil(11/5)`
		`ans =`
		`3`

`floor(x)`	Round toward minus infinity.	`>> floor(-9/4)`
		`ans =`
		`-3`

`rem(x,y)`	Returns the remainder after x is divided by y.	`>> rem(13,5)`
		`ans =`
		`3`

`sign(x)`	Signum function. Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.	`>> sign(5)`
		`ans =`
		`1`

명령 창

```
>> fix(-3.2)
ans =
    -3
>> ceil(-3.2)
ans =
    -3
>> floor(-3.2)
ans =
    -4
>> fix(3.2)
ans =
     3
>> ceil(3.2)
ans =
     4
>> floor(3.2)
ans =
     3
>>
```

1.6 Defining Scalar Variables

- A variable is a name made of a letter or a combination of several letters (and digits) that is assigned a numerical value. Once a variable is assigned a numerical value, it can be used in mathematical expressions, in functions, and in any MATLAB statements and commands.

1.6.1 The Assignment Operator

Variable_name = A numerical value, or a computable expression

- When the **Enter** key is pressed the numerical value of the right-hand side is assigned to the variable, and MATLAB displays the variable and its assigned value in the next two lines.

```
명령 창
>> x=2;
>> x=x*2+7-12
x =
    -1
>>
```

- The use of previously defined variables to define a new variable is demonstrated next.

- If a variable already exists, typing the variable’s name and pressing the **Enter** key will display the variable and its value in the next two lines.

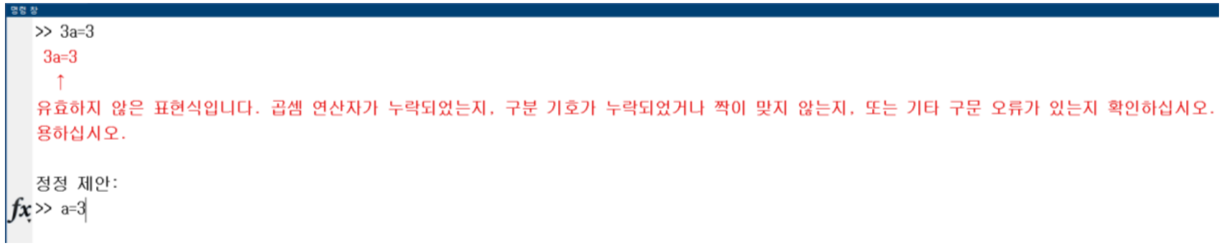
```
명령 창
>> a=12;
>> B=4;
>> C=(a-B)+40-a/B*10;
>> C
C =
    18
>>
```

```
명령 창
>> a=12, B=4; C=(a-B)+40-a/B*10
a =
    12
C =
    18
>>
```

- Several assignments can be typed in the same line. The assignments must be separated with a comma (spaces can be added after the comma).
- When the **Enter** key is pressed, the assignments are executed from left to right and the variables and their assignments are displayed.
- A variable is not displayed if a semicolon is typed instead of a comma.

1.6.2 Rules about Variable Names

- A variable can be named according to the following rules:



- *Must begin with a letter.*

- *Can be up to 63 characters long.*

- *Can contain letters, digits, and the underscore character.*

- *Cannot contain punctuation characters (e.g., period (마침표), comma, semicolon).*

- *MATLAB is case-sensitive: it distinguishes between uppercase and lowercase letters. For example, AA, Aa, aA, and aa are the names of four different variables.*

- *No spaces are allowed between characters (use the underscore where a space is desired).*

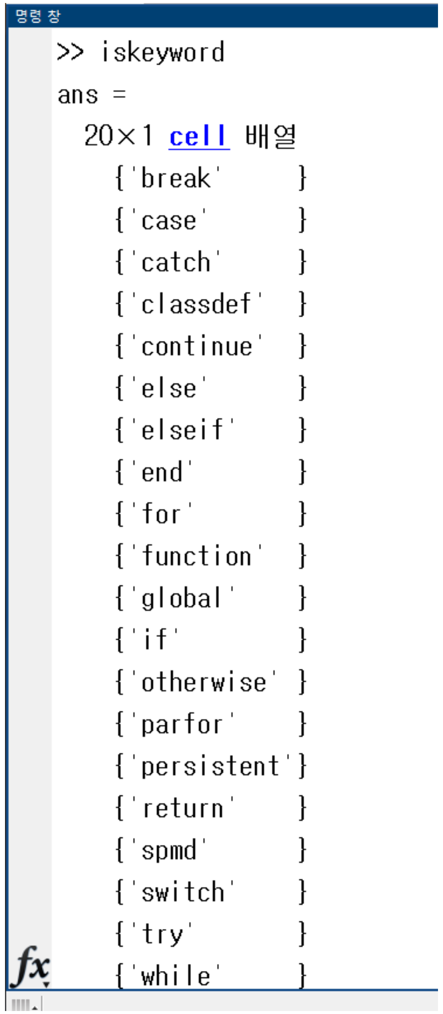
- *Avoid using the name of a built-in function for a variable (i.e., avoid using cos, sin, exp, sqrt, etc.). Once a function name is used to for a variable name, the function cannot be used.*

1.6.3 Predefined Variables and Keywords

- There are 20 words, called keywords, that are reserved by MATLAB for various purposes and cannot be used as variable names. These words are:

```
break case catch classdef  function global if otherwise
parfor persistent return spmd switch try while
```

- When typed, these words appear in blue. An error message is displayed if the user tries to use a keyword as a variable name. (The keywords can be displayed by typing the command `iskeyword`.)



- A number of frequently used variables are already defined when MATLAB is started. Some of the predefined variables are:

`ans`	A variable that has the value of the last expression that was not assigned to a specific variable (see Tutorial 1-1). If the user does not assign the value of an expression to a variable, MATLAB automatically stores the result in `ans`.
`pi`	The number π.
`eps`	The smallest difference between two numbers. Equal to 2^(–52), which is approximately 2.2204e–016.
`inf`	Used for infinity.
`i`	Defined as $\sqrt{-}$, which is: 0 + 1.0000i.
`j`	Same as i.
`NaN`	Stands for Not-a-Number. Used when MATLAB cannot determine a valid numeric value. Example: 0/0.

- The predefined variables can be redefined to have any other value.

- The variables pi, eps, and inf, are usually not redefined since they are frequently used in many applications.

- Other predefined variables, such as i and j, are sometime redefined (commonly in association with loops) when complex numbers are not involved in the application.

1.7 Useful Commands for Managing Variables

- The following are commands that can be used to eliminate variables or to obtain information about variables that have been created. When these commands are typed in the Command Window and the **Enter** key is pressed, either they provide information, or they perform a task as specified below.

Command	Outcome
clear (clear all)	Removes all variables from the memory (workspace)
clear x y z	Removes only variables x, y, and z from the memory
who	Displays a list of the variables currently in the memory
whos	Displays a list of the variables currently in the memory and their sizes together with information about their bytes and class

명령 창

```
>> who

사용자의 변수:

Nam2    aa      bb1       coastline  groundline  hl2   icethick  lat   out2   topo  yb
Real    aa1     bedtopo   coc        h_leg       hl3   ind1      lon   ran    xb    yy
Xi      ak      cc        cri        hc          hl4   ind2      nlat  size   xx
Yi      bb      cc1       dd         hl1         i     inde      nlon  size2  y

>>
```

명령 창

```
>> whos
  Name             Size                 Bytes  Class                                  Attributes

  Nam2             43x1                  4740  cell
  Real             43x2                   688  double
  Xi               21600x1801       311212800  double
  Yi               21600x1801       311212800  double
  aa               18x1                   144  double
  aa1              456078x1           3648624  double
  ak               543112x1           4344896  double
  bb               11x1                    88  double
  bb1              456078x1           3648624  double
  bedtopo          1000x1200          9600000  double
  cc               10x1                    80  double
  cc1              456078x1           3648624  double
  coastline        456078x2           7297248  double
  coc              22x3                   528  double
  cri              2x2                     32  double
  dd               4x1                     32  double
  groundline       830295x2          13284720  double
  h_leg            1x1                      8  matlab.graphics.illustration.Legend
  hc               1x1                      8  matlab.graphics.illustration.ColorBar
  hl1              1x1                      8  matlab.graphics.chart.primitive.Line
```

1.8 Script Files

- So far all the commands were typed in the Command Window and were executed when the **Enter** key was pressed.

- Although every MATLAB command can be executed in this way, using the Command Window to execute a series of commands — especially if they are related to each other (a program) — is not convenient and may be difficult or even impossible. The commands in the Command Window cannot be saved and executed again. In addition, the Command Window is not interactive.

- This means that every time the **Enter** key is pressed only the last command is executed, and everything executed before is unchanged. If a change or a correction is needed in a command that was previously executed and the result of this command is used in commands that follow, all the commands have to be entered and executed again.

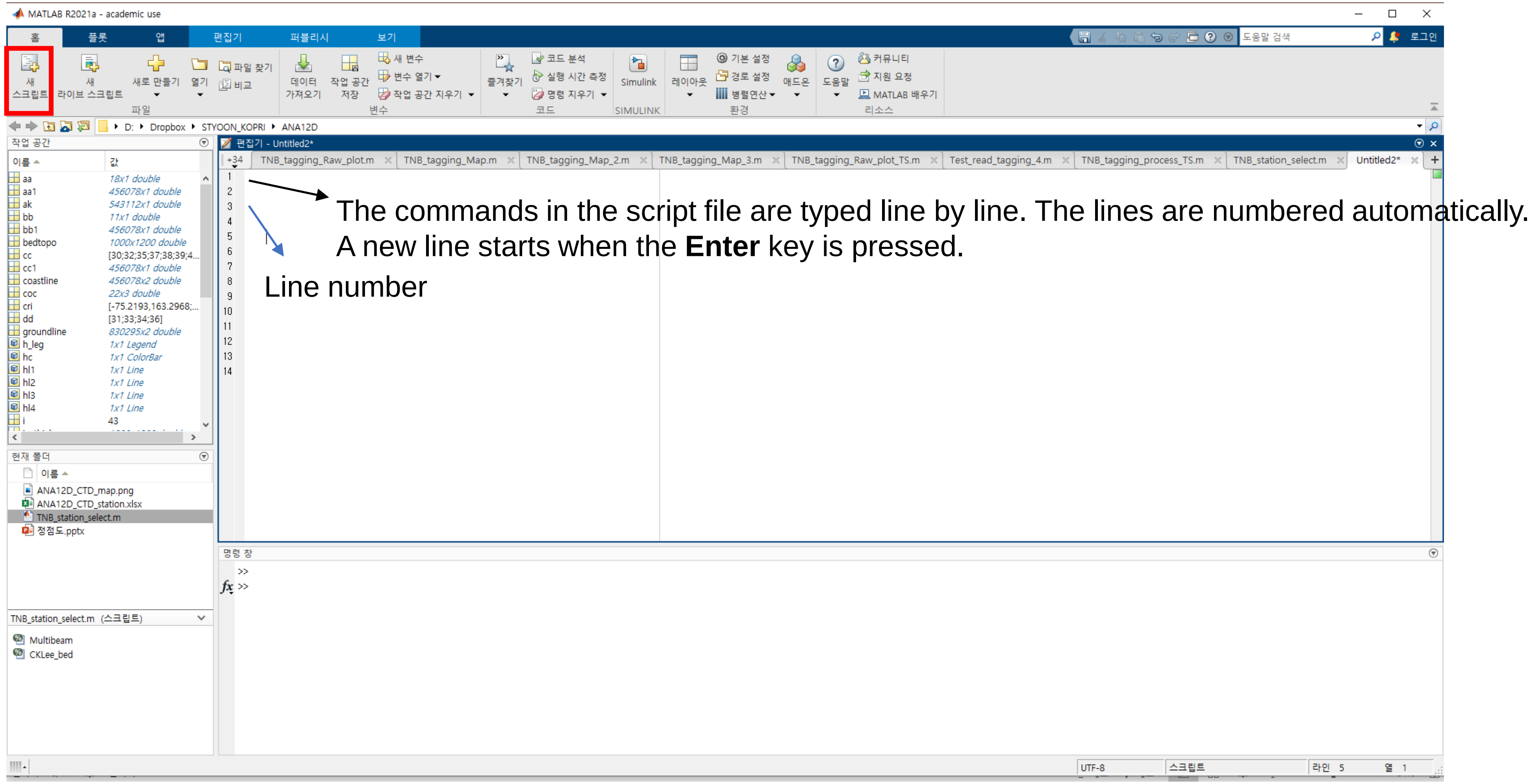
- A different (better) way of executing commands with MATLAB is first to create a file with a list of commands (program), save it, and then run (execute) the file. When the file runs, the commands it contains are executed in the order that they are listed. If needed, the commands in the file can be corrected or changed and the file can be saved and run again. Files that are used for this purpose are called script files.

1.8.1 Notes about Script Files

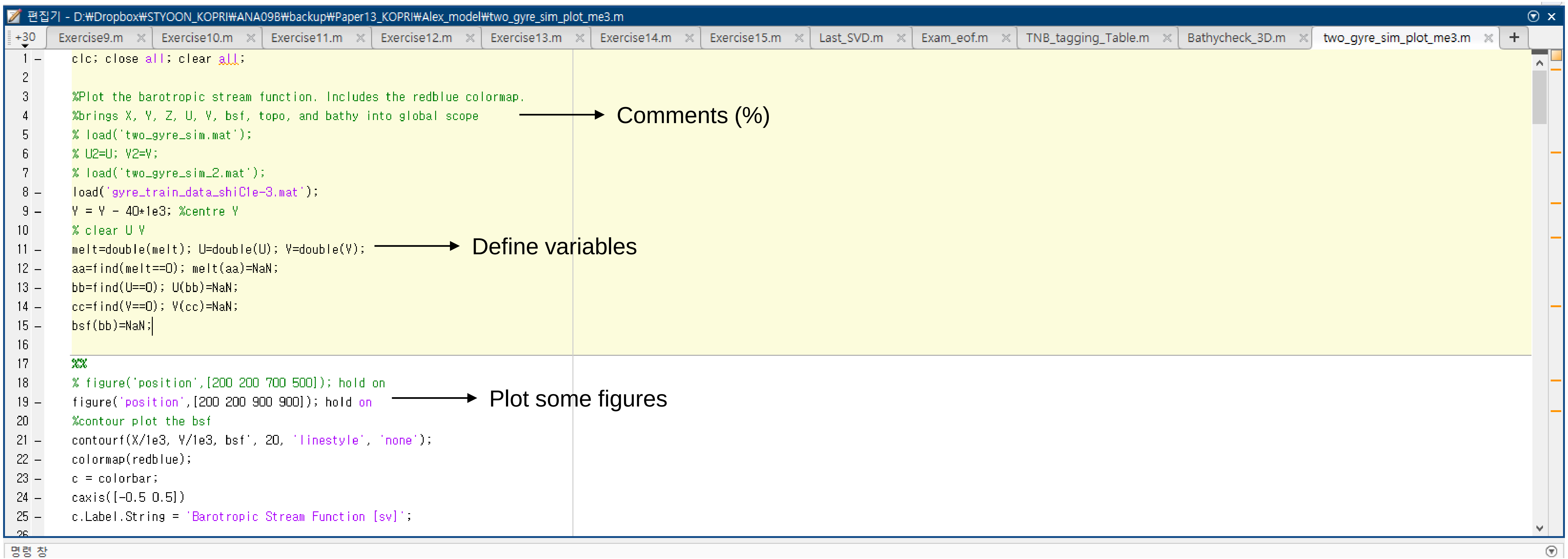
- A script file is a sequence of MATLAB commands, also called a program.
- When a script file runs (is executed), MATLAB executes the commands in the order they are written, just as if they were typed in the Command Window.
- When a script file has a command that generates an output (e.g., assignment of a value to a variable without a semicolon at the end), the output is displayed in the Command Window.
- Using a script file is convenient because it can be edited (corrected or otherwise changed) and executed many times.
- Script files can be typed and edited in any text editor and then pasted into the MATLAB editor.
- Script files are also called M-files because the extension .m is used when they are saved.

1.8.2 Creating and Saving a Script File

- In MATLAB script files are created and edited in the Editor/Debugger Window.



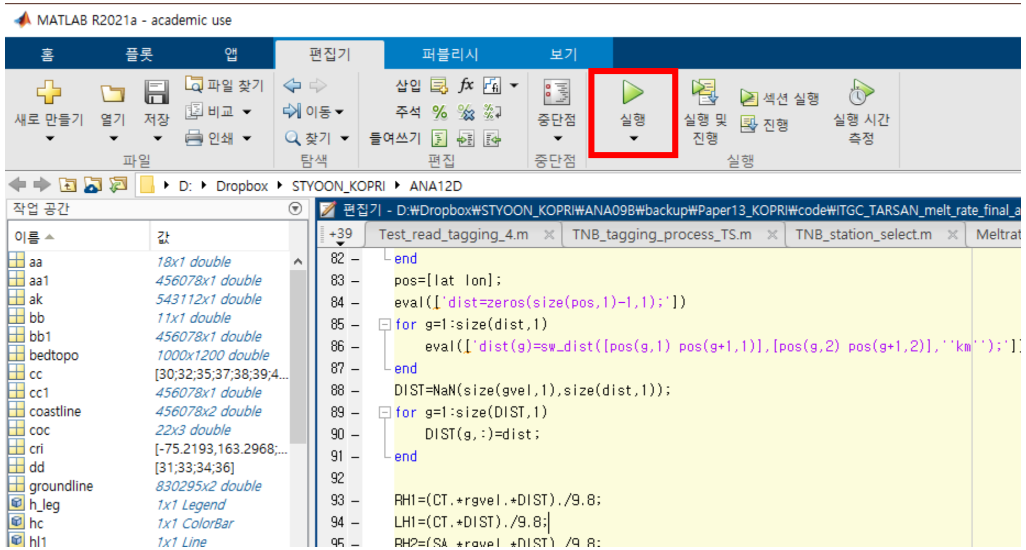
Ex)



- Before a script file can be executed it has to be saved. This is done by clicking **Save** in the Toolstrip and selecting **Save As...** from the menu that opens. When saved, MATLAB adds the extension .m to the name.

- The rules for naming a script file follow the rules of naming a variable (must begin with a letter, can include digits and underscore, *no spaces*, and up to 63 characters long). The names of user-defined variables, predefined variables, and MATLAB commands or functions should not be used as names of script files.

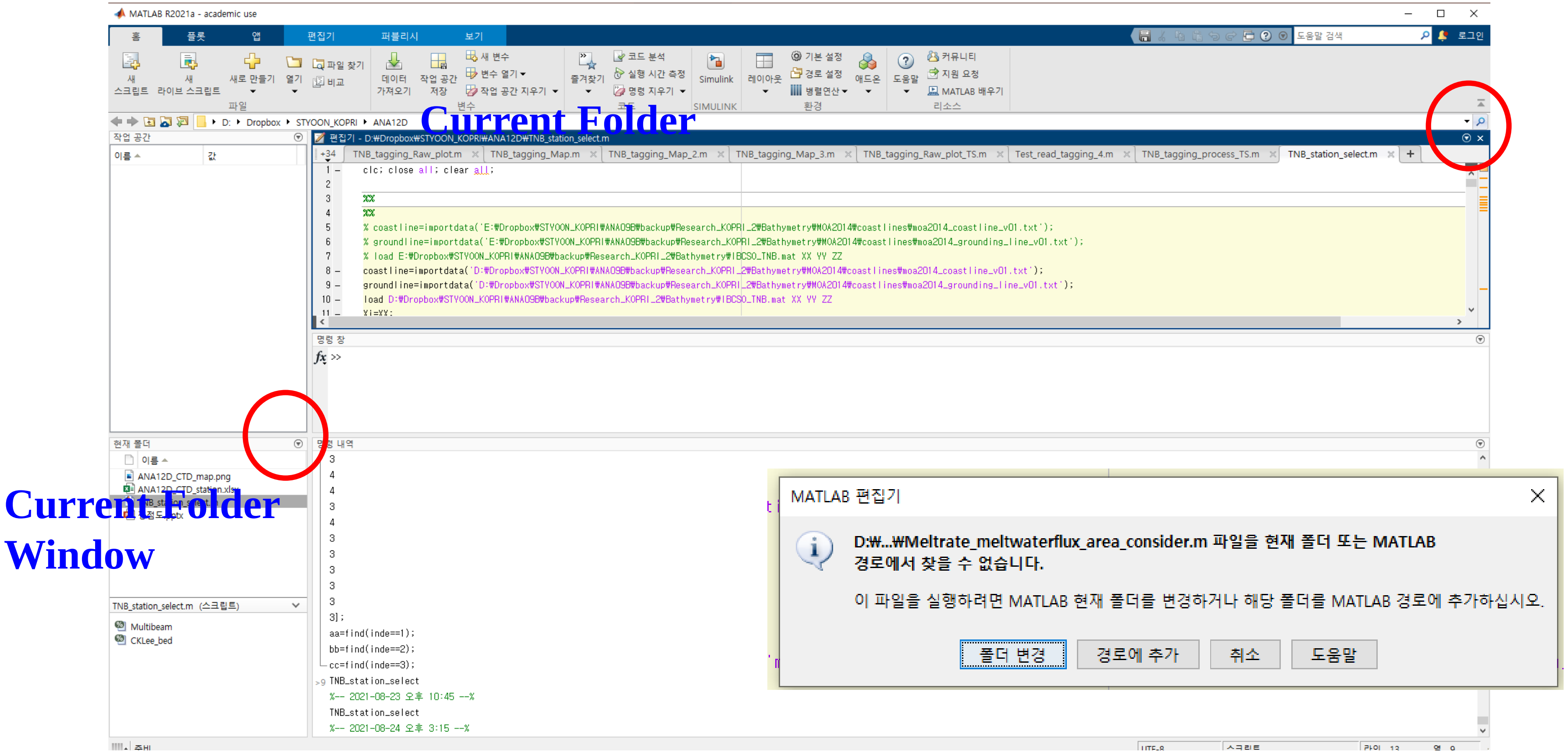
1.8.3 Running (Executing) a Script File



- A script file can be executed either directly from the Editor Window by clicking on the **Run** icon or by typing the file name in the Command Window and then pressing the **Enter** key.
- Execution: F5; Execute partially: F9

- For a file to be executed, MATLAB needs to know where the file is saved.
- The file will be executed if the folder where the file is saved is the current folder of MATLAB or if the folder is listed in the search path.

1.8.4 Current Folder



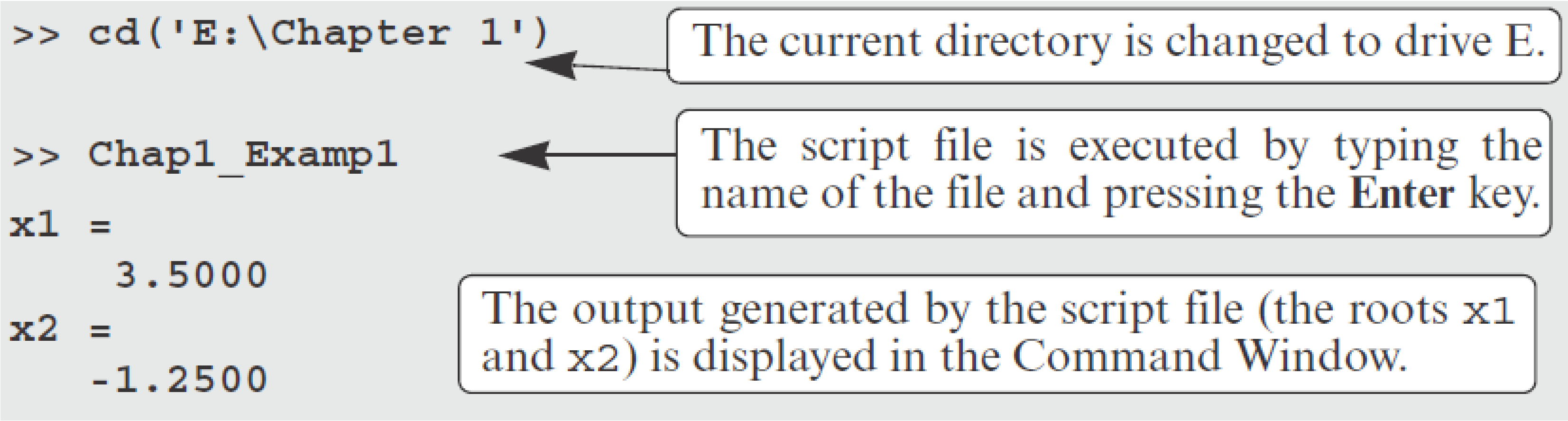
- If an attempt is made to execute a script file by clicking on the **Run** icon (in the Editor Window) when the current folder is not the folder where the script file is saved, then the prompt opens.

- The user can then change the current folder to the folder where the script file is saved, or add it to the search path.

- The current folder can also be changed in the Current Folder Window, and by choosing the drive and folder where the files is saved.

- An alternative simple way to change the current folder is to use the cd command in the Command Window.
- To change the current folder to a different drive, type `cd`, space, and then the name of the directory followed by a colon : and press the **Enter** key. For example, to change the current folder to drive E (e.g., the flash drive) type `cd E:`.
- If the script file is saved in a folder within a drive, the path to that folder has to be specified.

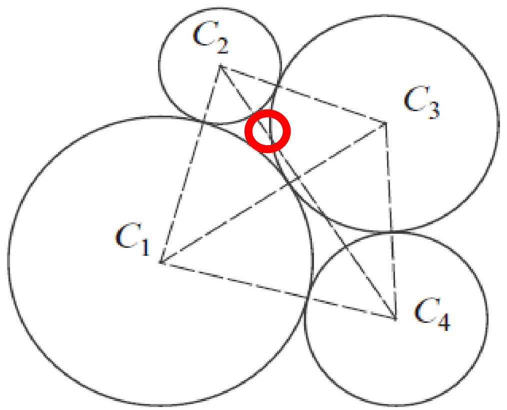
Ex)

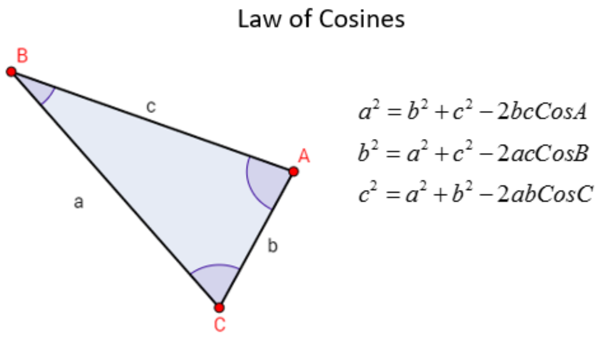


1.9 Examples of MATLAB Applications

Sample Problem 1-1 $$\cos^2\frac{x}{2} = \frac{\tan x + \sin x}{2\tan x}$$ X=π/5

Sample Problem 1-2





Sample Problem 1-3 $$T = T_S + (T_0 - T_S)e^{-kt}$$

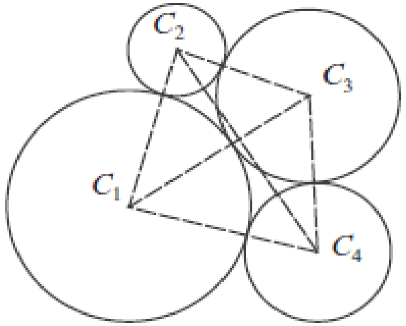
Sample Problem 1-4 $$B = P\left(1 + \frac{r}{n}\right)^{nt} \quad (1) \qquad B = P(1+r)^t \quad (2)$$

Sample Problem 1-2: Geometry and trigonometry

Four circles are placed as shown in the figure. At each point where two circles are in contact, they are tangent to each other. Determine the distance between the centers C_2 and C_4.

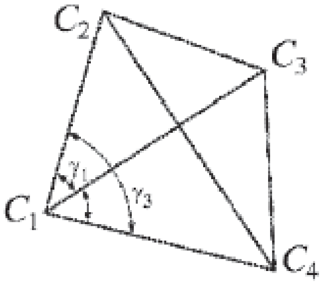
The radii of the circles are:

$R_1 = 16$ mm, $R_2 = 6.5$ mm, $R_3 = 12$ mm, and $R_4 = 9.5$ mm.



Solution

The lines that connect the centers of the circles create four triangles. In two of the triangles, $\Delta C_1C_2C_3$ and $\Delta C_1C_3C_4$, the lengths of all the sides are known. This information is used to calculate the angles γ_1 and γ_2 in these triangles by using the law of cosines. For example, γ_1 is calculated from:



$$(C_2C_3)^2 = (C_1C_2)^2 + (C_1C_3)^2 - (C_1C_2)(C_1C_3)\cos\gamma_1$$

Next, the length of the side C_2C_4 is calculated by considering the triangle $\Delta C_1C_2C_4$. This is done, again, by using the law of cosines (the lengths C_1C_2 and C_1C_4 are known and the angle γ_3 is the sum of the angles γ_1 and γ_2).

Sample Problem 1-3: Heat transfer

An object with an initial temperature of T_0 that is placed at time $t = 0$ inside a chamber that has a constant temperature of T_S will experience a temperature change according to the equation

$$T = T_S + (T_0 - T_S)e^{-kt}$$

where T is the temperature of the object at time t, and k is a constant. A soda can at a temperature of 120° F (after being left in the car) is placed inside a refrigerator where the temperature is 38° F. Determine, to the nearest degree, the temperature of the can after three hours. Assume $k = 0.45$. First define all of the variables and then calculate the temperature using one MATLAB command.

Sample Problem 1-4: Compounded interest

The balance B of a savings account after t years when a principal P is invested at an annual interest rate r and the interest is compounded n times a year is given by:

$$B = P\left(1 + \frac{r}{n}\right)^{nt} \qquad (1)$$

If the interest is compounded yearly, the balance is given by:

$$B = P(1 - r)^t \qquad (2)$$

Suppose \$5,000 is invested for 17 years in one account for which the interest is compounded yearly. In addition, \$5,000 is invested in a second account in which the interest is compounded monthly. In both accounts the interest rate is 8.5%. Use MATLAB to determine how long (in years and months) it would take for the balance in the second account to be the same as the balance of the first account after 17 years.

Sample Problem 1-1

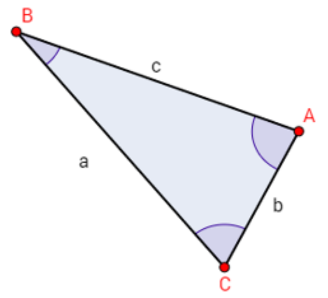
```
>> x=pi/5;
>> LHS=cos(x/2)^2
LHS =
    0.9045
>> RHS=(tan(x)+sin(x))/(2*tan(x))
RHS =
    0.9045
```

Define x.

Calculate the left-hand side.

Calculate the right-hand side.

Law of Cosines



$$a^2 = b^2 + c^2 - 2bcCosA$$
$$b^2 = a^2 + c^2 - 2acCosB$$
$$c^2 = a^2 + b^2 - 2abCosC$$

Sample Problem 1-3

```
>> Ts=38;  T0=120; k=0.45; t=3;

>> T=round(Ts+(T0-Ts)*exp(-k*t))

T =
    59
```

Round to the nearest integer.

Sample Problem 1-2

C2C4=33.5051

```
% Solution of Sample Problem 1-2
R1=16; R2=6.5; R3=12; R4=9.5;
C1C2=R1+R2; C1C3=R1+R3; C1C4=R1+R4;
C2C3=R2+R3; C3C4=R3+R4;
Gama1=acos((C1C2^2+C1C3^2-C2C3^2)/(2*C1C2*C1C3));
Gama2=acos((C1C3^2+C1C4^2-C3C4^2)/(2*C1C3*C1C4));
Gama3=Gama1+Gama2;

C2C4=sqrt(C1C2^2+C1C4^2-2*C1C2*C1C4*cos(Gama3))
```

Define the R's.

Calculate the lengths of the sides.

Calculate γ_1, γ_2, and γ_3.

Calculate the length of side C_2C_4.

Sample Problem 1-4

```
% Solution of Sample Problem 1-4
P=5000;  r=0.085;  ta=17; n=12;
B=P*(1+r)^ta
t=log(B/P)/(n*log(1+r/n))

years=fix(t)
months=ceil((t-years)*12)
```

Step (a): Calculate B from Eq. (2).

Step (b): Solve Eq. (1) for t, and calculate t.

Step (c): Determine the number of years.

Determine the number of months.

```
>> format short g
B =
       20011
t =
      16.374
years =
    16
months =
     5
```

The values of the variables B, t, years, and months are displayed (since a semicolon was not typed at the end of any of the commands that calculate the values).

1.10 Problems

1. Calculate:

   (a) $\left(5-\frac{19}{7}+2.5^3\right)^2$ (b) $7\times3.1+\frac{\sqrt{120}}{5}-15^{5/3}$

2. Calculate:

   (a) $\sqrt[3]{8+\frac{80}{2.6}}+e^{3.5}$ (b) $\left(\frac{1}{\sqrt{75}}+\frac{73}{3.1^3}\right)^{1/4}+55\times0.41$

3. Calculate:

   (a) $\frac{23+\sqrt[3]{45}}{16\times0.7}+\log_{10}589006$ (b) $(36.1-2.25\pi)(e^{2.3}+\sqrt{20})$

4. Calculate:

   (a) $\frac{3.8^2}{2.75-41\times25}+\frac{5.2+1.8^5}{\sqrt{3.5}}$ (b) $\frac{2.1\times10^6-15.2\times10^5}{3\cdot\sqrt[3]{6\times10^{11}}}$

5. Calculate:

   (a) $\frac{\sin(0.2\pi)}{\cos(\pi/6)}+\tan 72°$ (b) $(\tan 64°\cos 15°)^2+\frac{\sin^2 37°}{\cos^2 20°}$

6. Define the variable z as $z = 4.5$; then evaluate:

   (a) $0.4z^4+3.1z^2-162.3z-80.7$ (b) $\left(z^3-23\right)/\left(\sqrt[3]{z^2+17.5}\right)$

7. Define the variable t as $t = 3.2$; then evaluate:

   (a) $\frac{1}{2}e^{2t}-3.81t^3$ (b) $\frac{6t^2+6t-2}{t^2-1}$

8. Define the variables x and y as $x = 6.5$ and $y = 3.8$; then evaluate:

   (a) $\left(x^2+y^2\right)^{2/3}+\frac{xy}{y-x}$ (b) $\frac{\sqrt{x+y}}{(x-y)^2}+2x^2-xy^2$

9. Define the variables a, b, c, and d as:

$c = 4.6$, $d = 1.7$, $a = cd^2$, and $b = \frac{c+a}{c-d}$; then evaluate:

(a) $e^{d-b} + \sqrt[3]{c+a} - (ca)^d$ (b) $\frac{d}{c} + \left(\frac{ct}{b}\right)^2 - c^d - \frac{a}{b}$

10. Two trigonometric identities are given by:

(a) $\cos^2 x - \sin^2 x = 1 - 2\sin^2 x$ (b) $\frac{\tan x}{\sin x - 2\tan x} = \frac{1}{\cos x - 2}$

For each part, verify that the identity is correct by calculating the values of the left and right sides of the equation, substituting $x = \pi / 10$.

11. Two trigonometric identities are given by:

(a) $(\sin x + \cos x)^2 = 1 + 2\sin x\cos x$ (b) $\frac{1-2\cos x - 3\cos^2 x}{\sin^2 x} = \frac{1-3\cos x}{1-\cos x}$

For each part, verify that the identity is correct by calculating the values of the left and right sides of the equation, substituting $x = 20^\circ$.

12. Define two variables: *alpha* = π/8, and *beta* = π/6. Using these variables, show that the following trigonometric identity is correct by calculating the values of the left and right sides of the equation.

$$\tan(\alpha + \beta) = \frac{\tan\alpha + \tan\beta}{1 - \tan\alpha\tan\beta}$$

13. Given: $\int x^2 \cos x\,dx = 2x\cos x + (x^2 - 2)\sin x$. Use MATLAB to calculate the following definite integral: $\int_{\pi/6}^{\pi/3} x^2 \cos x\,dx$.

```
a1=(2*pi/3)*cos(pi/3)+((pi/3)^2-2)*sin(pi/3)
a2=(2*pi/6)*cos(pi/6)+((pi/6)^2-2)*sin(pi/6)
a1-a2
```

Function	Description	Example

`round(x)`	Round to the nearest integer.	`>> round(17/5)`
		`ans =`
		`3`

Function	Description	Example

`fix(x)`	Round toward zero.	`>> fix(13/5)`
		`ans =`
		`2`

`ceil(x)`	Round toward infinity.	`>> ceil(11/5)`
		`ans =`
		`3`

`floor(x)`	Round toward minus infinity.	`>> floor(-9/4)`
		`ans =`
		`-3`

`rem(x,y)`	Returns the remainder after x is divided by y.	`>> rem(13,5)`
		`ans =`
		`3`

`sign(x)`	Signum function. Returns 1 if $x > 0$, -1 if $x < 0$, and 0 if $x = 0$.	`>> sign(5)`
		`ans =`
		`1`

명령 창

```
>> fix(-3.2)
ans =
    -3
>> ceil(-3.2)
ans =
    -3
>> floor(-3.2)
ans =
    -4
>> fix(3.2)
ans =
     3
>> ceil(3.2)
ans =
     4
>> floor(3.2)
ans =
     3
>>
```

22. A total of 4217 eggs have to be packed in boxes that can hold 36 eggs each. By typing one line (command) in the Command Window, calculate how many eggs will remain unpacked if every box that is used has to be full. (Hint: Use MATLAB built-in function `fix`.)

23. A total of 777 people have to be transported using buses that have 46 seats and vans that have 12 seats. Calculate how many buses are needed if all the buses have to be full, and how many seats will remain empty in the vans if enough vans are used to transport all the people that did not fit into the buses. (Hint: Use MATLAB built-in functions `fix` and `ceil`.)

24. Change the display to `format long g`. Assign the number 7E8/13 to a variable, and then use the variable in a mathematical expression to calculate the following by typing one command:
    (*a*) Round the number to the nearest tenth. 10^-1
    (*b*) Round the number to the nearest million. 10^-6

29. The number of combinations $C_{n,r}$ of taking r objects out of n objects is given by:

$$C_{n,r} = \frac{n!}{r!(n-r)!}$$

In the Powerball lottery game the player chooses five numbers from 1 through 59, and then the Powerball number from 1 through 35.
Determine how many combinations are possible by calculating $C_{59,5}C_{35,1}$.
(Use the built-in function `factorial`.)

33. The greatest common divisor is the largest positive integer that divides the numbers without a remainder. For example, the greatest common divisor of 8 and 12 is 4. Use the MATLAB Help Window to find a MATLAB built-in function that determines the greatest common divisor of two numbers. Then use the function to show that the greatest common divisor of:
(*a*) 91 and 147 is 7.
(*b*) 555 and 962 is 37.

+ least common multiple
(a) 24 and 27 is 216
(b) 14 and 18 is 126

38. Use the Help Window to find a display format that displays the output as a ratio of integers. For example, the number 3.125 will be displayed as 25/8. Change the display to this format and execute the following operations:

(*a*) $5/8+16/6$ (*b*) $1/3-11/13+2.7^2$

39. Gosper's approximation for factorials is given by:

$$n! = \sqrt{\left(2n+\frac{1}{3}\right)\pi}\, n^n e^{-n}$$

Use the formula for calculating 19!. Compare the result with the true value obtained with MATLAB's built-in function `factorial` by calculating the error (*Error*=(*TrueVal*-*ApproxVal*)/*TrueVal*).

32. Radioactive decay of carbon-14 is used for estimating the age of organic material. The decay is modeled with the exponential function $f(t) = f(0)e^{kt}$, where t is time, $f(0)$ is the amount of material at $t = 0$, $f(t)$ is the amount of material at time t, and k is a constant. Carbon-14 has a half-life of approximately 5,730 years. A sample taken from the ancient footprints of Acahualinca in Nicaragua shows that 77.45% of the initial ($t = 0$) carbon-14 is present. Determine the estimated age of the footprint. Solve the problem by writing a program in a script file. The program first determines the constant k, then calculates t for $f(t) = 0.7745f(0)$, and finally rounds the answer to the nearest year.

37. The velocity v and the falling distance d as a function of time of a skydiver that experience the air resistance can be approximated by:

$$v(t) = \sqrt{\frac{mg}{k}}\tanh\left(\sqrt{\frac{kg}{m}}\,t\right) \quad \text{and} \quad d(t) = \frac{m}{k}\ln\left[\cosh\left(\sqrt{\frac{kg}{m}}\,t\right)\right]$$

where $k = 0.24$ kg/m is a constant, m is the skydiver mass, $g = 9.81$ m/s^2 is the acceleration due to gravity, and t is the time in seconds since the skydiver starts falling. Determine the velocity and the falling distance at $t = 8$ s for a 95-kg skydiver

ln=log_e (natural logarithm)

40. According to Newton's law of universal gravitation, the attraction force between two bodies is given by:

$$F = G\frac{m_1 m_1}{r^2}$$

where m_1 and m_2 are the masses of the bodies, r is the distance between the bodies, and $G = 6.67 \times 10^{-11}$ N-m^2/kg^2 is the universal gravitational constant. Determine how many times the attraction force between the sun and the Earth is larger than the attraction force between the Earth and the moon. The distance between the sun and Earth is 149.6×10^9 m, the distance between the moon and Earth is 384.4×10^6 m, $m_{Earth} = 5.98 \times 10^{28}$ kg, $m_{sun} = 2.0 \times 10^{30}$ kg, and $m_{moon} = 7.36 \times 10^{22}$ kg.

In next time,

- Starting with MATLAB
  - Exercises

- Creating Arrays
  - Matrix

If you have any questions,

Please contact styoon@knu.ac.kr