

# 1장 MATLAB 시작하기

---

신호처리 응용 I

청주대학교 대학원 전자공학과  
한철수

# 목차

---

- MATLAB의 특징
- MATLAB의 여러 창들의 특성과 목적
- 명령어 창에 대한 상세 소개
- 스칼라의 산술 연산
- 기본적인 수학 내장 함수의 사용
- 스크립트 파일(script file)의 소개 및 사용법

# MATLAB의 특징

---

- Mathworks사의 계산 프로그램, 프로그래밍 언어, 프로그래밍 환경
- Interpreter 방식의 언어
  - Compiler 방식의 언어(C, Pascal, Fortran 등)
- 수학기산 및 가시화(visualization)에 매우 편리함
- 선형대수, 데이터 분석, 신호처리, 수치적분 등 많은 과학 계산용 내장함수를 제공함
- 사용자에게 의한 함수 작성이 편리
- 배우기 쉽고 사용이 편하다.
- 다양한 분야의 광범위한 Toolbox를 제공

# MATLAB의 장점

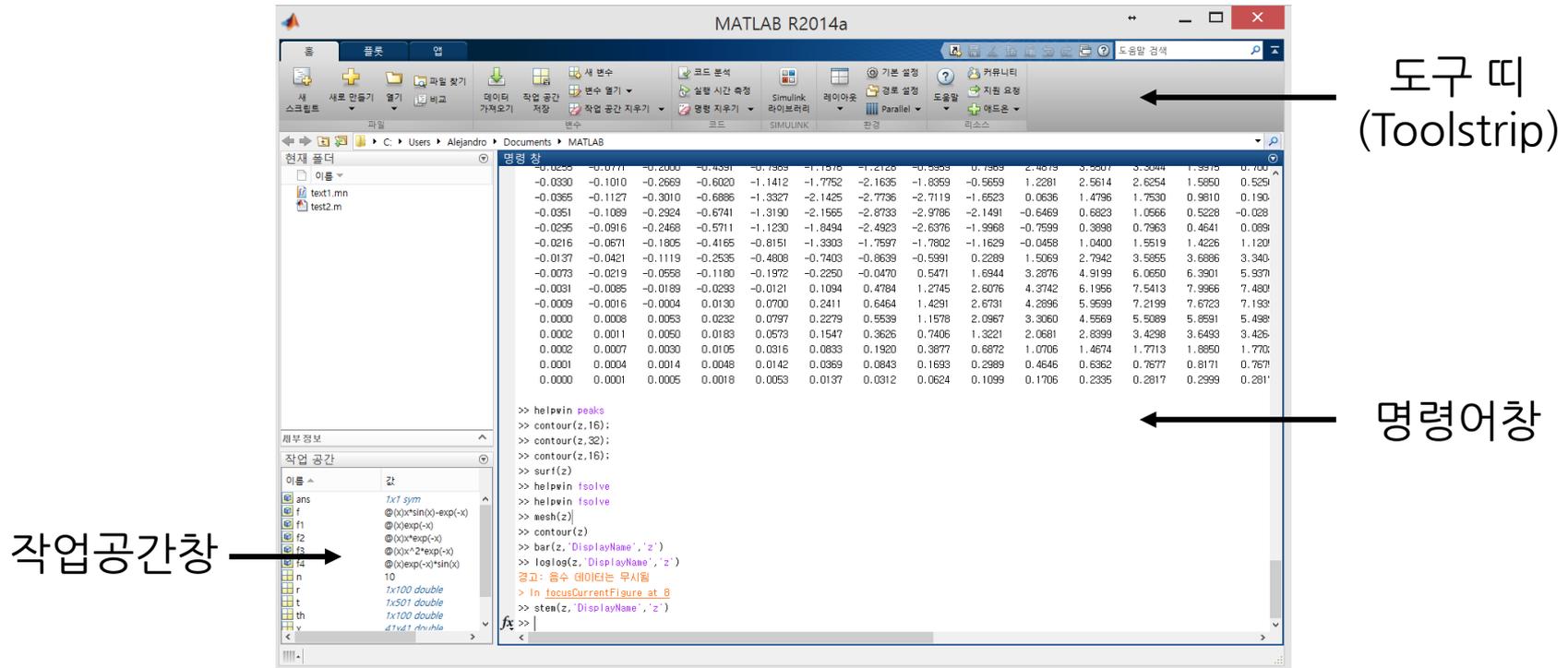
---

- 쉽고 빠른 프로그래밍
  - 고급 언어이면서도 빠르고 쉽게 프로그래밍 가능
- 미리 지정된 풍부한 자료형
  - 복소수, 3차원 행렬, 구조체 등
- 무궁무진한 내장 함수와 분야별 toolbox
- 고급 시각화 툴 지원
- 다양한 플랫폼을 지원하는 M-file
- 풍부한 정보를 제공하는 자료
  - helpwin, doc 등
- 폭 넓은 사용자 그룹

# MATLAB 의 창

- 기본 화면의 구성

- 명령어창 (Command Window)
  - 사용자의 명령을 입력
- 현재 디렉터리창 (Current Directory Window)
- 작업공간창 (Working space)
  - 사용자 변수들에 대한 정보제공



# 그 밖의 창 소개

- 여러가지 보조 창

- 그림 창(Figure Window)

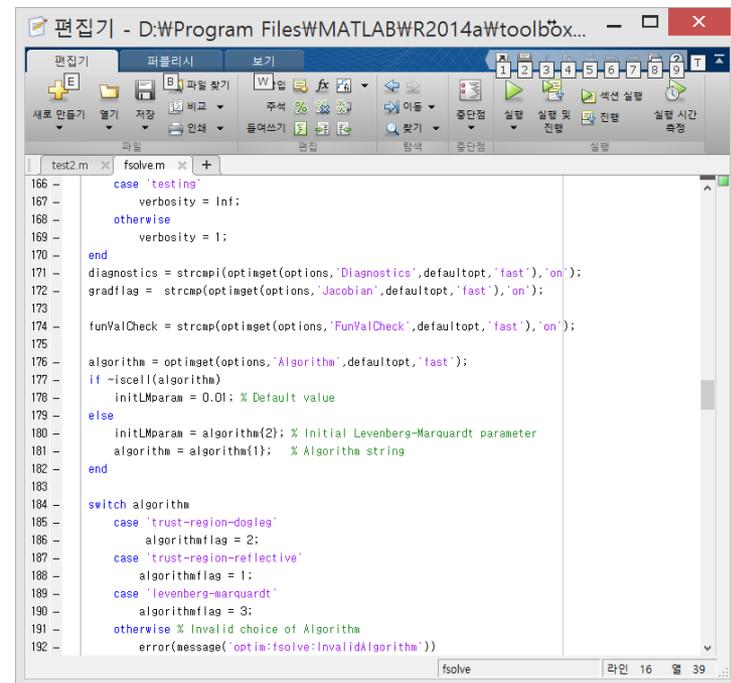
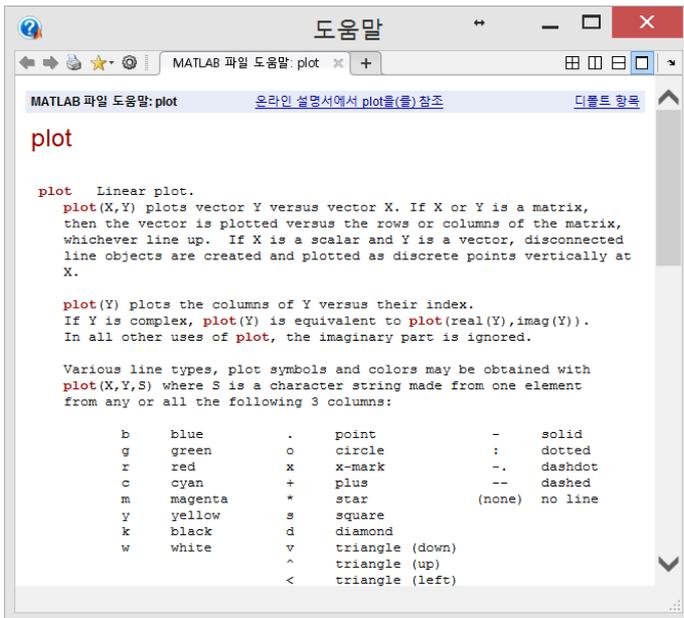
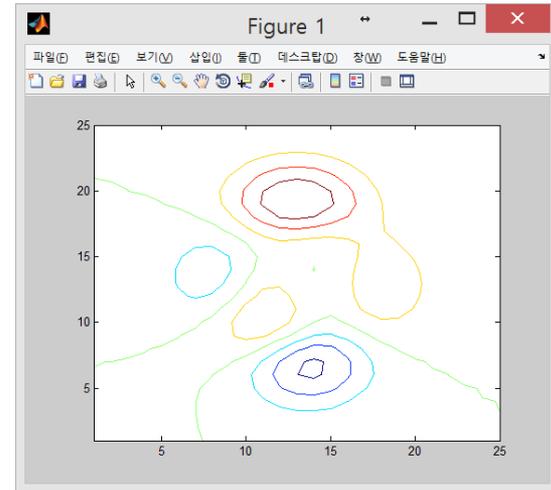
- 그래프 도시

- 편집기 창(Editor Window)

- 스크립트(m-file) 편집

- 도움말 창(Help window)

- 도움말 표시

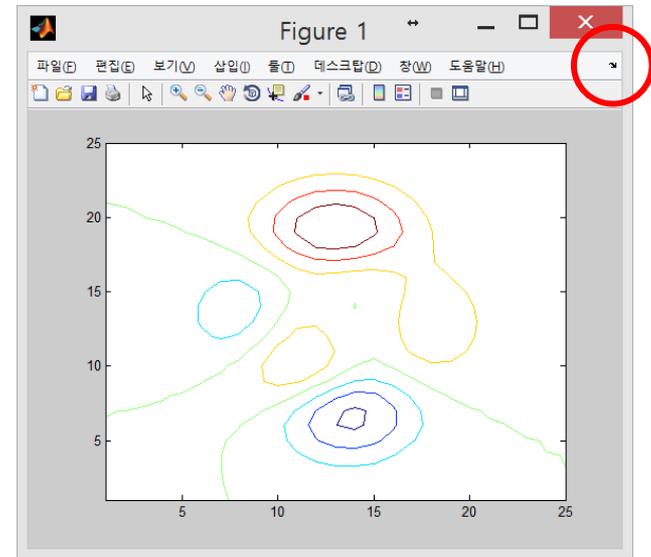


# MATLAB 창의 종류와 목적

창 (Window)	목적
명령어 창 (Command Window)	메인 창으로 변수를 입력하고, 프로그램을 실행함
그림 창 (Figure Window)	그래프 명령어의 실행 결과가 표시됨
편집기 창 (Editor Window)	스크립트 파일과 함수 파일을 생성하고 디버깅함
도움말 창 (Help Window)	도움말 정보를 제공함
명령어 기록 창 (Command History Window)	명령어 창에 입력된 명령어들을 기록함
작업공간 창 (Workspace Window)	저장된 변수에 대한 정보를 제공함
현재 디렉터리 창 (Current Folder Window)	현재 디렉터리에 있는 파일들을 보여줌

# docking 과 undocking

- undocking
- docking



0.0000	0.0000	0.0004	0.0100	0.0100	0.2217	0.0404	1.4231	2.0101	4.2000
0.0000	0.0008	0.0053	0.0232	0.0797	0.2279	0.5539	1.1578	2.0967	3.3060
0.0002	0.0011	0.0050	0.0183	0.0573	0.1547	0.3626	0.7406	1.3221	2.0681
0.0002	0.0007	0.0030	0.0105	0.0316	0.0833	0.1920	0.3877	0.6872	1.0706
0.0001	0.0004	0.0014	0.0048	0.0142	0.0369	0.0843	0.1693	0.2989	0.4646
0.0000	0.0001	0.0005	0.0018	0.0053	0.0137	0.0312	0.0624	0.1099	0.1706

>> helpwin peaks

# 명령어 창에서 작업하기 (1/3)

- (>>) 프롬프트 다음에 명령을 입력 후 Enter 키 입력
  - 여러 문장은 콤마(,)로 구별 가능
  - 이 경우 왼쪽부터 차례로 실행
- 한 줄이 너무 길면(...)을 친 후
  - Enter키를 치면
  - 다음 줄에 연이어 입력 가능.
  - 총 4,096 글자까지 가능
- 위쪽 방향키 (↑)를 누르면 과거 입력했던 명령어를 다시 불러와서 쓸 수 있음.
- 명령어 입력 후 세미콜론(; )을 붙이면
  - 계산과정과 결과가 명령창에 표시되지 않음.
  - (결과가 너무 길거나 이미 알고있을 때 유용)

```
>> x=10, y=35, z=y-x
x =
    10
y =
    35
z =
    25
```



```
>> x=10, y=35, z=y-x
x =
mesh(z)
contour(z)
bar(z, 'DisplayName', 'z')
loglog(z, 'DisplayName', 'z')
stem(z, 'DisplayName', 'z')
helpwin command history
contour(z)
helpdesk
helpwin
helpwin plot
x=10, y=35, z=y-x
fx >> x=10, y=35, z=y-x
```

# 명령어 창에서 작업하기 (2/3)

- clc 명령은 화면의 결과만 모두 지움.
  - 변수내용, 명령어 히스토리는 지워지지 않음.
  - clear와 착각하지 말 것
- 퍼센트(%)는 프로그램을 설명할 때 사용하며 명령과는 무관함
- 콤마(,)는 여러 명령을 구분할 때
- (ans)는 방금 계산한 결과값

```
>> exp(3);           % e3  
>> log10(10^5);    % log10(105)
```

```
>> x=2; y=6+x, x=y+7  
y=  
    8  
x=  
   15
```

# 명령어 창에서 작업하기 (3/3)

---

- 출력 포맷 관련
  - format short:
    - 소수점 이하 4 자리수의 고정소수점으로 표시함
  - format long
    - 소수점 이하 15 자리수의 고정소수점으로 표시함
  - format compact
    - 화면표시에 빈 줄이 없이 뾰뾰하게 표시함
  - format loose
    - 화면표시에 빈 줄을 넣어 표시함

# 스칼라의 산술 연산

- MATLAB에서 사용하는 산술연산자들의 기호

내용	연산자	예
덧셈	+	5+3
뺄셈	-	5-3
곱셈	*	5*3
오른쪽 나눗셈	/	5/3
나머지	mod	5 mod 3
왼쪽 나눗셈	\	5\3(=3/5)
지수연산	^	5^3 (5 <sup>3</sup> 을 의미함)

- 왼쪽 나눗셈(left division)은 오른쪽 나눗셈(right division)의 역수이지만, 배열에서는 왼쪽 나눗셈이 주로 사용된다.

# 연산자 우선순위

- 우선순위가 높은 연산이 먼저 실행됨
- 순위가 같은 연산이 둘 이상 있으면 왼쪽부터 실행됨
- 계산순서를 바꾸기 위해 괄호를 사용할 수 있음

우선순위	연산자	설명
1	()	괄호. 괄호가 중첩되어 있는 경우, 가장 안쪽 괄호부터 실행된다.
2	^	거듭제곱(^)
3	*,/	곱셈, 나눗셈(우선순위가 동등함)
4	+,-	덧셈, 뺄셈

# 연산 순서 예 (1/3)

```
>> 4*50+3*100+10^2-400/2
ans =
    400
>> 2^5/(2^5-1)
ans =
    1.0323
>> 2*(sqrt(5)-1)/(sqrt(5)+1)^2-1
ans =
   -0.7639
>> 10*acos(-1)
ans =
   31.4159
```

**연산 순서:** 수학의 일반 규칙과 동일

1. 괄호 먼저 계산한다
2. 지수 계산
3. 곱하기, 나누기는 왼쪽에서 우측 방향으로
4. 더하기, 빼기는 왼쪽에서 우측방향으로

**;(세미콜론)** 계산결과 표시를 억제함

**,(콤마)** 한 줄에 여러 명령어를 입력하면서  
계산결과를 표시

**%** 프로그램을 설명할 때 사용하며, 명령과는  
무관함

# 연산 순서 예 (2/3)

- MATLAB을 사용하는 가장 간단한 방법은 MATLAB을 계산기처럼 사용하는 것임
- 수식 입력 후 Enter를 치면 ans에 답이 표시됨

```
>> 7+8/2      % 8/2가 먼저 수행됨
ans =
    11
>> (7+8)/2
ans =
    7.5000
>> 4+5/3+2    % 5/3이 먼저 수행됨
ans =
    7.6667
>> 5^3/2      % 5^3이 먼저 수행됨
ans =
    62.5000
>> 27^(1/3)+32^0.2 % 27^(1/3)+32^0.2
ans =
    5
```

```
>> 27^1/3+32^0.2  % (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
>> 7/2
ans =
    3.5000
>> 7₩2 % 역슬래시 대신 ₩가 표시됨(한글윈도우)
ans =
    0.2857
>> ans
ans =
    0.2857
```

# 연산 순서 예 (3/3)

- 의도하는 바에 따라 적절한 괄호가 필요함.

```
>> 2*4^3+5           % 2×43 + 5
ans =
    133
>> (2*4)^3 + 5       % (2×4)3 + 5
ans =
    517
>> (2*4)^3/5         % (2×4)3/5
ans =
    102.4
>> 9^(1/2)           % 91/2
ans =
     3
>> 9^1/2             % 91/2
ans =
    4.5000
```

괄호를 사용하여 의도하는 결과를 바꿀 수 있다.  
연산자 사이에 공백이 있어도 무방함

# 출력형식

- MATLAB의 기본(default) 출력 형식
  - 소수점 이하 네 자리를 가진 고정소수점
- 출력형식은 format 명령어로 바꿀 수 있음.
  - format short
  - format long
  - format short e
  - format long e
  - format short g
  - format long g
  - format bank
  - format compact
  - format loose

```
>> 4+5/3+2  
ans =  
    7.6667
```

- 표시만 지정 한대로 할 뿐이고 실제 변수의 값이 바뀌는 것은 아님.

# format 명령어의 화면 표시형식

명령어	설명	예
format short	0.001 ≤ 수 ≤ 1000인 수를 소수점 이하 네 자리수의 고정소수점으로 표시함. 그 외의 범위의 수는 short e 형식으로 표시함.	>>290/7 ans = 41.4286
format long	0.001 ≤ 수 ≤ 100인 수를 소수점 이하 15자리의 고정소수점으로 표시함. 그 외 범위의 수는 long e의 형식으로 표시함.	>>290/7 ans = 41.42857142857143
format short e	소수점 이하 네 자리수의 과학적 표기법으로 표시함.	>>290/7 ans = 4.1429e+001
format long e	소수점 이하 15 자리수의 과학적 표기법으로 표시함.	>>290/7 ans = 4.142857142857143e+001
format short g	고정 소수점 표시와 부동소수점 표시 중에서 적절한 방법으로 표시하며, 유효 숫자는 5개임.	>>290/7 ans = 41.429
format long g	고정 소수점 표시와 부동소수점 표시 중에서 적절한 방법으로 표시하며, 유효 숫자는 15개임.	>>290/7 ans = 41.4285714285714
format bank	소수점 이하 두 자리 까지만 표시함.	>>290/7 ans = 41.43
format compact	화면에 많은 정보가 표시되도록 하기 위해 빈 줄을 제거함	
format loose	format compact와 반대로 결과 전후에 빈 줄을 삽입함.	

# MATLAB의 기본 수학 내장함수

- MATLAB은 광범위한 내장함수를 포함하고 있고 수식에 사용 가능함
- 내장함수는 함수이름과 괄호 속의 인자(argument)로 구성됨
  - 예) `sqrt(x)` → 수의 제곱근을 계산하는 함수로, 함수이름은 `sqrt`, 인자는 `x`임
- 함수의 인자
  - 수 또는 수치가 할당된 변수 또는 수, 변수, 함수로 구성된 계산 가능한 수식

```
>> sqrt(64)                                % 수 인자
ans = 8

>> sqrt(50+14*3)                            % 수식 인자
ans = 9.5917

>> sqrt(54+9*sqrt(100))                     % 함수가 포함된 인자
ans = 12

>> (15+600/4)/sqrt(121)                     % 함수가 포함된 수식
ans = 15
```

# 기본 수학 내장함수

함수	설명	예
<code>sqrt(x)</code>	제곱근	<pre>&gt;&gt; sqrt(81) ans = 9</pre>
<code>nthroot(x, n)</code>	실수 $x$ 의 실수 $n$ 제곱근. ( $x$ 가 음수이면, $n$ 은 홀수 정수이어야 함)	<pre>&gt;&gt; nthroot(80, 5) ans = 2.4022</pre>
<code>exp(x)</code>	지수함수( $e^x$ )	<pre>&gt;&gt; exp(5) ans = 148.4132</pre>
<code>abs(x)</code>	절대값	<pre>&gt;&gt; abs(-24) ans = 24</pre>
<code>log(x)</code>	자연로그. 밑이 $e$ 인 로그( $\ln$ )	<pre>&gt;&gt; log(1000) ans = 6.9078</pre>
<code>log10(x)</code>	밑이 10인 로그	<pre>&gt;&gt; log10(1000) ans = 3.0000</pre>
<code>factorial(x)</code>	계승함수 $x!$ ( $x$ 는 양의 정수이어야 함)	<pre>&gt;&gt; factorial(5) ans = 120</pre>

# 삼각함수

함수	설명	예
sin(x) sind(x)	각도 x의 사인(x는 radian) 각도 x의 사인(x는 °)	>> sin(pi/6) ans = 0.5000
cos(x) cosd(x)	각도 x의 코사인(x는 radian) 각도 x의 코사인(x는 °)	>> cosd(30) ans = 0.8660
tan(x) tand(x)	각도 x의 탄젠트(x는 radian) 각도 x의 탄젠트(x는 °)	>> tan(pi/6) ans = 0.5774
cot(x) cotd(x)	각도 x의 코탄젠트(x는 radian) 각도 x의 코탄젠트(x는 °)	>> cotd(30) ans = 1.7321

- $\pi = \text{pi}$ 로 입력
- 역삼각함수:  $\text{asin}(x)$ ,  $\text{acos}(x)$ ,  $\text{atan}(x)$ ,  $\text{acot}(x)$
- 쌍곡삼각함수(Hyperbolic):  $\text{sinh}(x)$ ,  $\text{cosh}(x)$ ,  $\text{tanh}(x)$ ,  $\text{coth}(x)$

# 어림함수

함수	설 명	예
round(x)	가장 가까운 정수로 반올림함	<pre>&gt;&gt; round(3.4) ans =      3</pre>
fix(x)	0쪽에 가까운 정수로 어림함	<pre>&gt;&gt; fix(2.6) ans =      2</pre>
ceil(x)	양의 무한대에 가까운 정수로 어림함	<pre>&gt;&gt; ceil(2.2) ans =      3</pre>
floor(x)	음의 무한대 쪽에 더 가까운 정수로 어림함	<pre>&gt;&gt; floor(-2.25) ans =     -3</pre>
sign(x)	부호 함수로서, $x > 0$ 이면 +1을, $x < 0$ 이면 -1을, $x = 0$ 이면 0을 돌려준다.	<pre>&gt;&gt; sign(5) ans =      1</pre>

# 스칼라 변수의 정의

- 할당 연산자 : 우측값을 좌측 변수에 대입

Variable\_name = 수치 값 또는 계산 가능한 식

- 변수 이름의 규칙

- 문자로 시작
- 최대 63글자
- 문자, 숫자, 밑줄글자(\_)로 구성
- 마침표(.), 콤마(,), 세미콜론(;) 포함 불가
- 변수 이름 사이에는 공백 사용불가 (대신 밑줄 글자(\_) 사용)
- 내장함수 이름 (즉 cos, sin, exp, sqrt 등) 변수로 사용 하면 그 함수 사용불가
- 키워드 20개 변수로 사용불가 (iskeyword 함수로 확인 가능)
  - break, case, catch, classdef, continue, else, elseif, end, for, function, global, if, otherwise, parfor, persistent, return, spmd, switch, try, while
- 미리 정의된 변수도 사용가능하나 본래의 값을 다시 사용불가.

# 여러가지 사용 유형들 (1/2)

```
>> x=15           % 15가 변수 x에 할당됨
x =
    15
>> x=3*x-12       % 새로운 값이 x에 할당됨
x =
    33
>> a=12           % 수 12를 a에 할당함
a =
    12
>> B=4            % 수 4를 B에 할당함
B =
     4
>> C=(a-B)+40-a/B*10 % C에 할당됨
C =
    18
>> ABB=72;
>> ABB=9;
>> ABB           % 변수의 현재 값이 출력됨
ABB =
     9
```

```
>> a=12;          % ;로 출력 안됨
>> B=4;
>> C=(a-B)+40-a/B*10; % ;로 출력 안됨
>> C
C =
    18
>> a=12, B=4; C=(a-B)+40-a/B*10 %한줄에
a =
    12
C =
    18
>> x=0.75;
>> E=sin(x)^2+cos(x)^2
E =
     1
```

# MATLAB에서 미리 정의된 변수

상수	설 명	예
eps	두 수 사이의 최소 차이로 $2^{-52}$ 정도	
i, j	허수 즉 $\sqrt{-1}$	>> 1+2j ans = 1.0000 + 2.0000i
Inf	무한히 큰수	>> 10/0 ans = Inf
NaN	Not-a-Number. 0/0과 같이 정의할 수 없는 결과	>> 0/0 ans = NaN
pi	$\pi=3.1415926535897\dots$	>> 2*pi ans = 6.2832

- 위 변수들은 사용자변수로도 사용될 수도 있으나 본래의 의미를 상실하므로 주의

# 여러가지 사용 유형들 (2/2)

```
>> a=4; b=3; sum=a+b
```

```
sum = 7
```

```
>> a=7; sum % a값이 바뀌어도
```

```
sum = 7
```

```
>> Class_1=20; Class_2=15; Class_3=14; Class_4=17; Total_Number=240;
```

```
>> ave = (Class_1 + Class_2 + Class_3 + Class_4) ...
```

```
 / Total_Number * 100
```

```
ans = 27.5
```

```
>> ave=(Class_1+Class_2+Class_3+Class_4)/Total_ ...
```

```
 ??? ave=(Class_1+Class_2+Class_3+Class_4)/Total ...
```

```
Error: The input character is not valid in MATLAB statements or  
expressions.
```

# 변수 관리에 유용한 명령

---

명령어	설 명
clear	메모리에서 모든 변수들을 제거.
clear x y z	메모리에서 변수 x, y, z 만을 제거
who	현재 메모리에 있는 변수들의 목록을 화면에 출력.
whos	현재 메모리에 있는 변수들의 이름과 크기, 바이트와 클래스에 대한 정보를 화면에 출력.

# 유용한 보조 명령

명령어	설 명
clc	명령창의 내용을 모두 지움. (Cleared the Command window)
exist('var')	'var' 이름의 파일이나 변수가 있는지 확인함.
quit	MATLAB 종료
helpwin 명령어	도움말 창을 연다
help 명령어	명령창에서 '명령어'에 간단한 설명을 보여줌.
lookfor 단어	단어와 관련된 명령어를 찾아준다. (명령어를 잘 모를 때 유용)
cd	디렉토리 위치변경하거나 현재 위치를 알려줌.

# 연습문제

---

- MATLAB을 이용하여 다음 계산을 수행하라.

$$(1) 6 \times 2^3 + \frac{4}{5}$$

$$(2) 3 \frac{\sin(2)}{5}$$

$$(3) \frac{27^2}{4} + \frac{4^5}{6^3 - 120}$$

$$(4) \tan^{-1} \frac{12^3}{45}$$

$$(5) \frac{1+i}{1-i}$$

$$(6) 2.73e^{\frac{\pi}{2}i}$$

# 스크립트 파일

---

- 스크립트(script) 파일

- 일련의 명령어들의 목록(프로그램)을 만들어서 파일로 저장
- 언제나 필요할 때 그 파일을 다시 실행할 수 있어서 능률적

- 스크립트 파일의 특징

- MATLAB의 일련의 명령어로 구성된 프로그램
- 명령어 창에서 입력하는 것처럼 파일에 기록된 순서대로 실행됨
- 결과는 명령어 창에 표시 (;을 붙이지 않으면)
- 수정이 가능하고 여러 번 실행도 가능
- 표준 ASCII 파일로서 메모장과 같은 텍스트 편집기에서도 작성/편집 가능
- MATLAB 편집기에서 작성/편집도 가능
- 확장자가 m 이라 M-파일 이라고도 함

# 스크립트파일 새로 작성 및 저장

- 작성방법

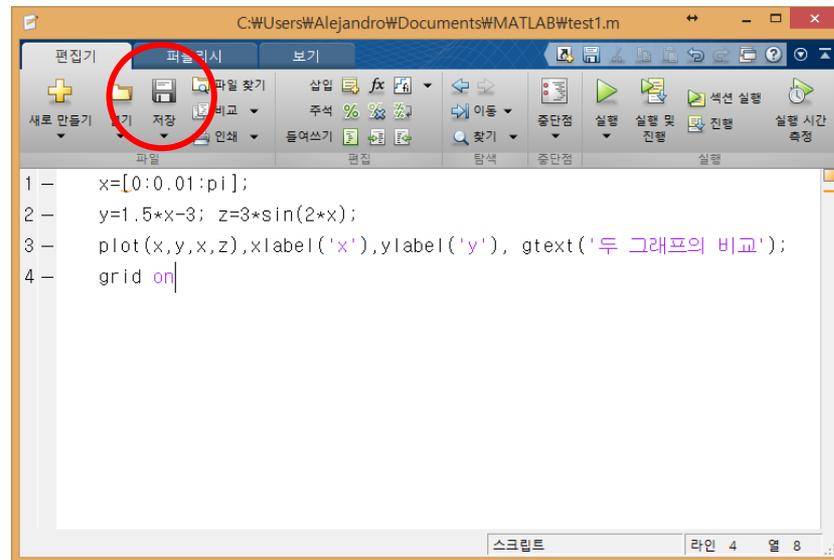
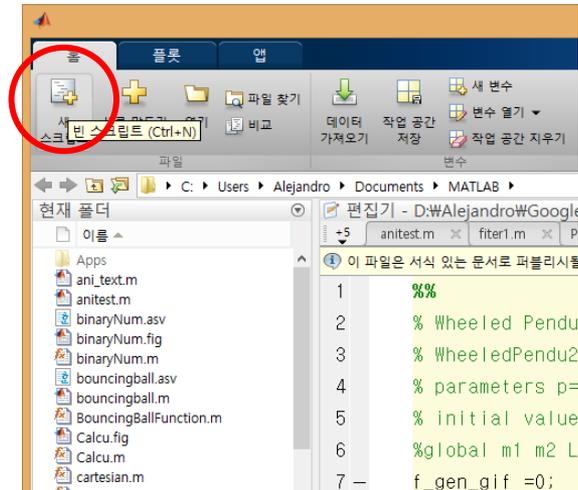
- 도구띠에서 새 스크립트 클릭

- 작성

- 편집기에 줄번호가 나옴.

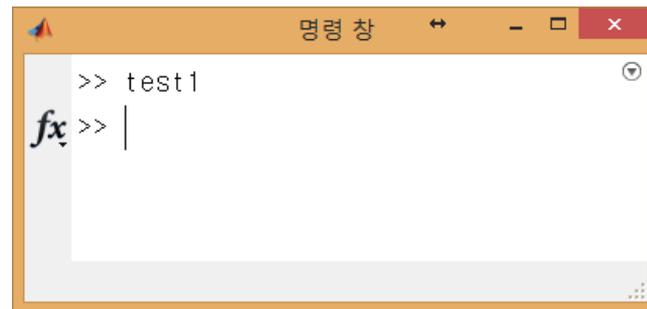
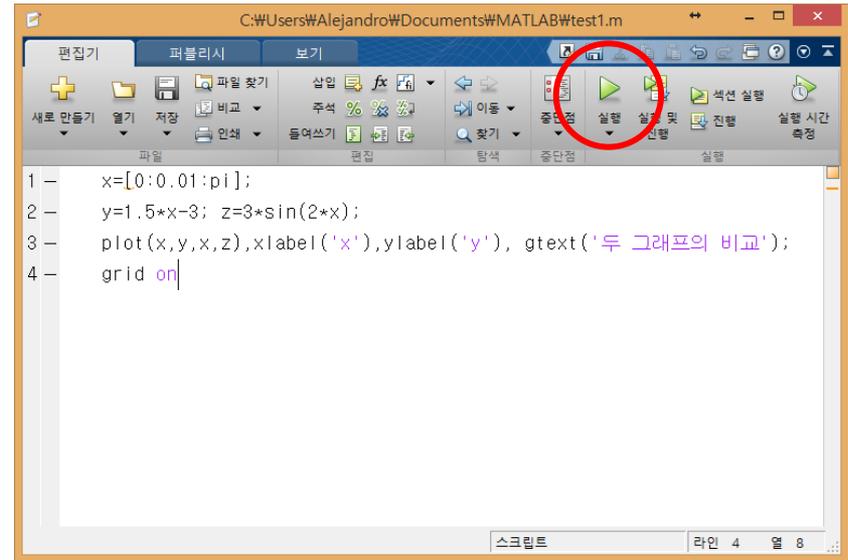
- 저장 방법

- 도구띠의 저장에서 선택



# 실행방법

- 편집기 도구띠에서 실행 클릭
- 명령창에서 m 파일이름 Enter키



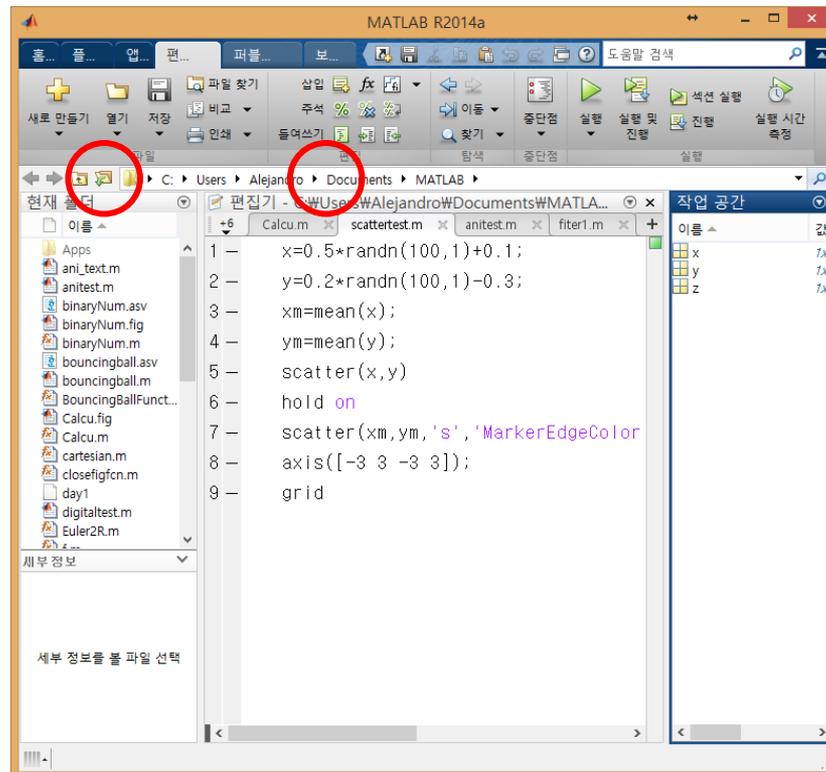
# 참고사항

---

- 스크립트 파일의 이름
  - 변수 이름과 마찬가지로 문자로 시작해야 하며, 문자, 밑줄문자, 숫자를 포함할 수 있음.
  - 이름의 길이는 63글자까지 임.
  - MATLAB 키워드나 내장 함수이름 사용불가.
- 내부 변수 이름
  - 스크립트 파일 내에 스크립트 이름과 같은 변수 사용불가.

# 디렉터리 변경 방법

- 원하는 파일이 있는 곳으로 디렉터리를 옮길 때
  - 현재 디렉터리 직접 수정
  - 폴더 찾아보기 버튼 이용



# cd 명령을 이용한 디렉터리 변경

- cd 명령의 인수로 드라이브 또는 디렉터리 이름을 사용함

```
>> cd('E:\matlab')    % e 드라이브 matlab 디렉터리
>> dir                % 디렉터리 내용 보기

.      test.m  test2.m
..     test1.m

>> test1              % 실행 !
>> cd c:              % 다시 C드라이브로
```

# 응용 예제 (1/4)

- 삼각항등식

$$\cos^2 \frac{x}{2} = \frac{\tan x + \sin x}{2 \tan x}$$

- $x=\pi/5$  를 대입하여 항등식이 성립함을 보여라.

```
>> x=pi/5; % x를 정의
>> LHS=cos(x/2)^2 % 좌변 계산
LHS =
    0.9045

>> RHS=(tan(x)+sin(x))/(2*tan(x)) % 우변 계산
RHS =
    0.9045
```

## 응용 예제 (2/4)

- $t = 0$ 에서 초기온도가  $T_0$ 인 물체를 온도가  $T_s$ 로 일정한 방 안에 놓게 되면, 물체는 다음 식에 따라 온도가 변한다.

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

- 여기서  $T$ 는 시간  $t$ 에서의 물체 온도,  $k$ 는 상수
- 자동차 안에 있던 온도  $120^\circ\text{F}$ 인 소다 캔을 온도가  $38^\circ\text{F}$ 인 냉장고 안에 넣었을 때, 세 시간 후 캔의 온도를 반올림하여 정수로 구하라.
  - 단,  $k = 0.45$  이다.

```
>> Ts=38; T0=120; k=0.45; t=3;           % 변수의 정의
>> T=round(Ts+(T0-Ts)*exp(-k*t))       % 반올림
T =
    59                                   % 결과
```

# 응용 예제 (3/4)

- 연이율  $r$ 로 이자가 연간  $n$ 번 복리로 주어지는 예금계좌에 원금  $P$ 를 투가할 때,  $t$ 년 후의 예금계좌 잔고  $B$ 는

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

- 이자가 매년 복리로 주어진다면, 잔고  $B$ 는

$$B = P(1 + r)^t \quad (2)$$

- 조건 (2)에 5,000 달러를 투자하고 조건 (1) 에도 5,000 달러를 투자했을 때 (2) 계좌의 17년 후의 잔고와 (1)계좌의 잔고가 같아지는 시간을 년과 달로 계산하라. (단,  $r$ 은 모두 8.5%)

- 풀이

- 식 (2)를 이용하여 이자가 매년 복리 지급되는 계좌에 투자된 \$5,000의 17년 후 잔고  $B$ 를 계산한다.
- 매월 복리 이자 공식인 식 (1)로부터, a.에서 계산된  $B$ 에 대한  $t$ 를 계산한다.
- 기간  $t$ 를 년과 월로 환산한다.

# 응용 예제 (4/4)

- 연이율  $r$ 로 이자가 연간  $n$ 번 복리로 주어지는 예금계좌에 원금  $P$ 를 투가할 때,  $t$ 년 후의 예금계좌 잔고  $B$ 는

```
% 예제 1-4 복리
P=5000;r=0.085;ta=17; n=12;
% 단계 a: 식(2) 로 B계산
B=P*(1+r)^ta
% 단계 b: (1)에서 t 계산
t=log(B/P)/(n*log(1+r/n))
% 단계 c: 년 수와 개월 수를 결정
years=fix(t)
months=ceil((t-years)*12)
```

```
>> ex1_4
B =
    2.0011e+04
t =
    16.3737
years =
    16
months =
     5
```

# 질문

---

Q & A