

MATLAB[®]

The Language of Technical Computing

Version 6.5.0.180913a Release 13
June 18, 2002



Copyright 1984–2002, The MathWorks, Inc.

목차

- Matlab이란 무엇인가?
- Matlab 특징
- Matlab 시작
- 행렬
- 연산자
- 꼭 알아야할 명령어
- M-file 프로그래밍
- 명령어의 흐름제어

목차(cont'd)

- M-file Debugging
- subfunction
- 다차원배열
- M-file작성시 유용한 함수들
- Matlab Graphics
- 자주쓰는 그래픽 관련 함수
- Matlab을 이용한 신호처리 예제
- 예제의 해결 포인트

Matlab은 무엇인가?

- 강력한 수학적 능력을 보유한 프로그램
- Matrix Laboratory의 약자이며 Cleve Moler에 의해 Fortran으로 작성되었으나, 현재는 미국의 Mathworks사에 의해 C++로, 버전 6부터는 JAVA를 이용해 작성되고있음

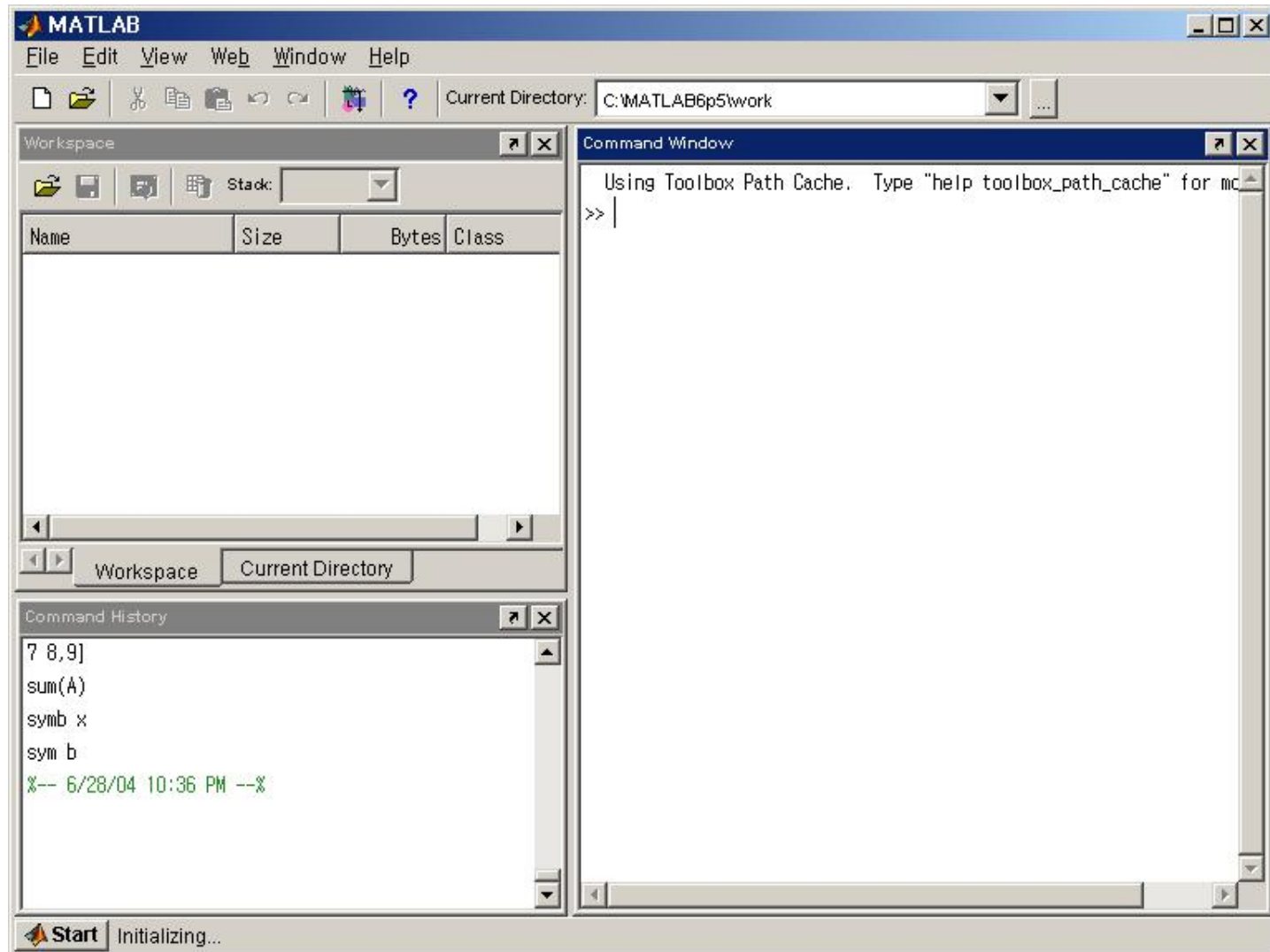
Matlab은 무엇인가?(cont'd)

- 이용범위
 - 수학과 관련된 계산
 - 알고리즘 개발
 - 상황 모델링과 데이터 분석
 - 여러가지 과학과 공학적 요소의 그래픽적 표현
 - GUI에 의한 애플리케이션 개발

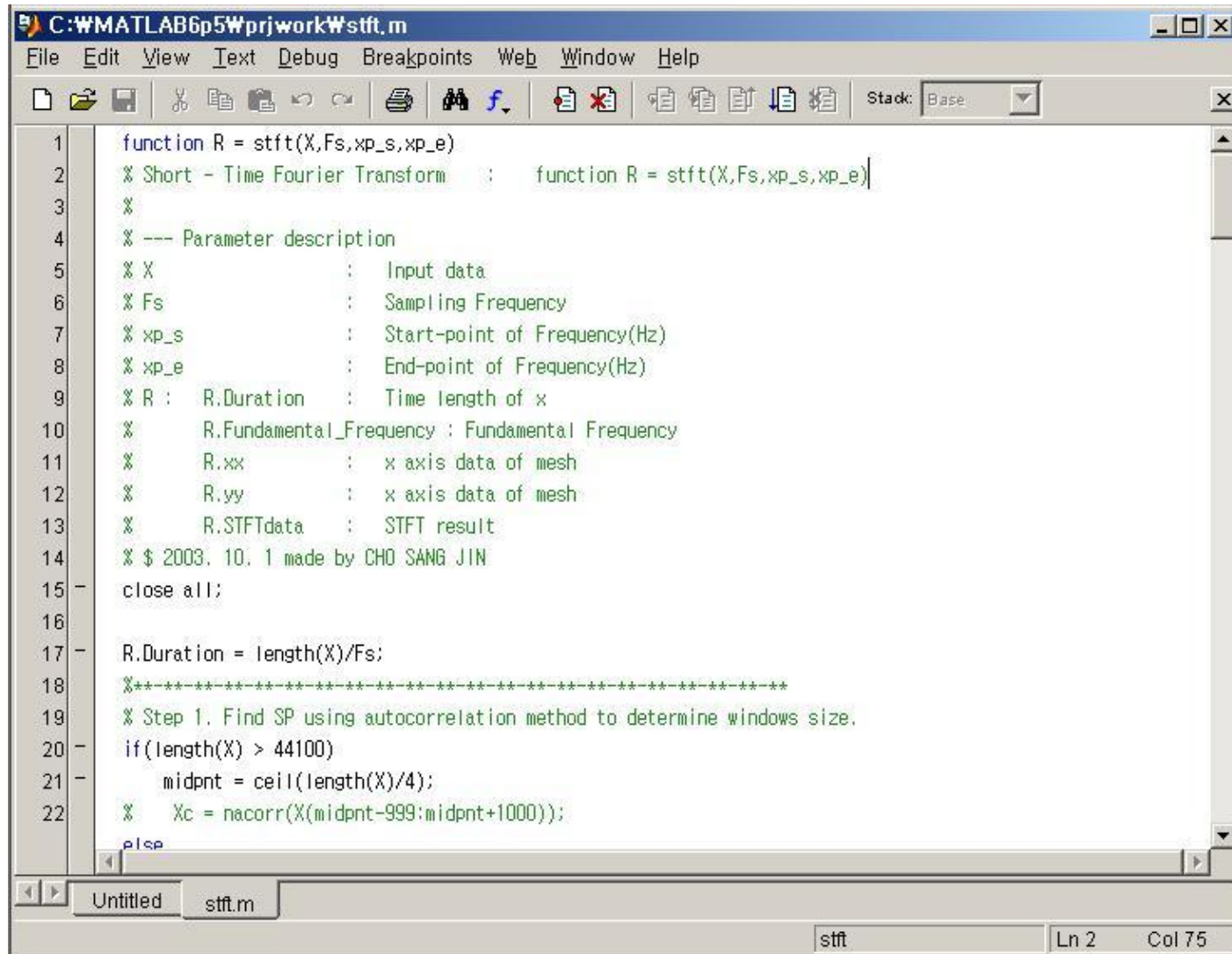
Matlab 특징

- 행렬 데이터
- M-File을 이용한 프로그래밍
- Toolbox
- Symbol로 이루어진 수식 계산
- GUI 프로그래밍
- Simulink

Matlab 시작



Matlab 시작(cont'd)

A screenshot of the MATLAB 6.5 SP1 editor window. The title bar shows the file path 'C:\WMATLAB6p5Wprjwork\stft.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. The toolbar contains various icons for file operations, editing, and debugging. The main editor area displays the source code for the 'stft' function. The code includes a function signature, comments describing parameters and the function's purpose, and the beginning of the function body. The status bar at the bottom indicates the current file is 'stft' and the cursor is at line 2, column 75.

```
1 function R = stft(X,Fs,xp_s,xp_e)
2 % Short - Time Fourier Transform : function R = stft(X,Fs,xp_s,xp_e)
3 %
4 % --- Parameter description
5 % X : Input data
6 % Fs : Sampling Frequency
7 % xp_s : Start-point of Frequency(Hz)
8 % xp_e : End-point of Frequency(Hz)
9 % R : R.Duration : Time length of x
10 % R.Fundamental_Frequency : Fundamental Frequency
11 % R.xx : x axis data of mesh
12 % R.yy : x axis data of mesh
13 % R.STFTdata : STFT result
14 % $ 2003. 10. 1 made by CHO SANG JIN
15 close all;
16
17 R.Duration = length(X)/Fs;
18 %*****
19 % Step 1. Find SP using autocorrelation method to determine windows size.
20 if(length(X) > 44100)
21 midpnt = ceil(length(X)/4);
22 % Xc = nacorr(X(midpnt-999:midpnt+1000));
else
```

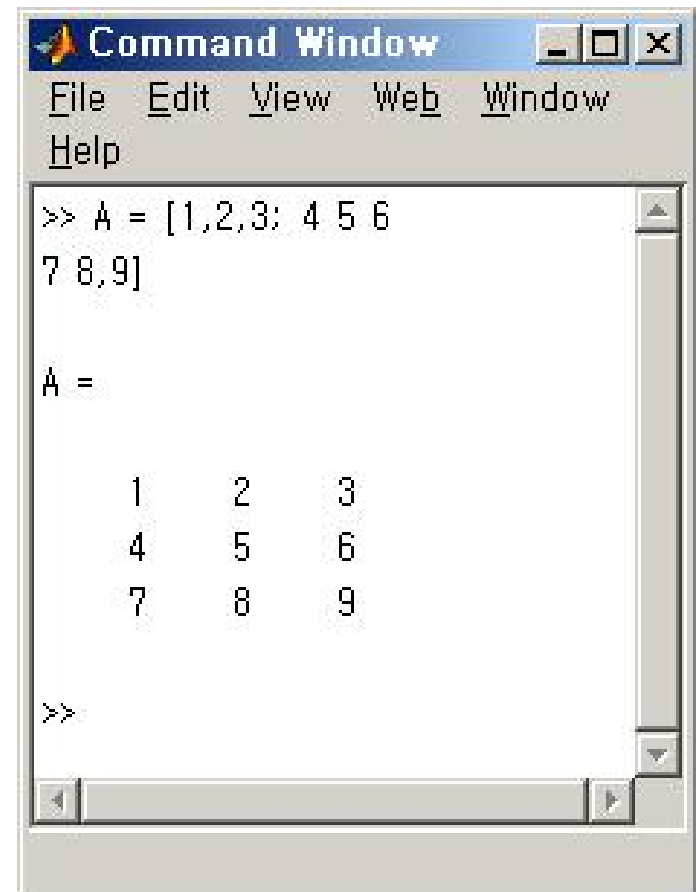
행렬 선언

- 변수명(variable) = 수식(expression)
 - 변수 : double, char, struct, cell, sparse
 - 수식 : Matlab이 해석할 수 있는 numeric , nonnumeric형의 수식
- 행렬을 만드는 방법
 - 명확한 행렬요소 작성
 - 외부 데이터를 load
 - M-file로 행렬을 생성

행렬 연산

- 행렬 데이터 입력
 - 열구분 - 빈칸, 콤마
 - 행구분 - 세미콜론, 엔터

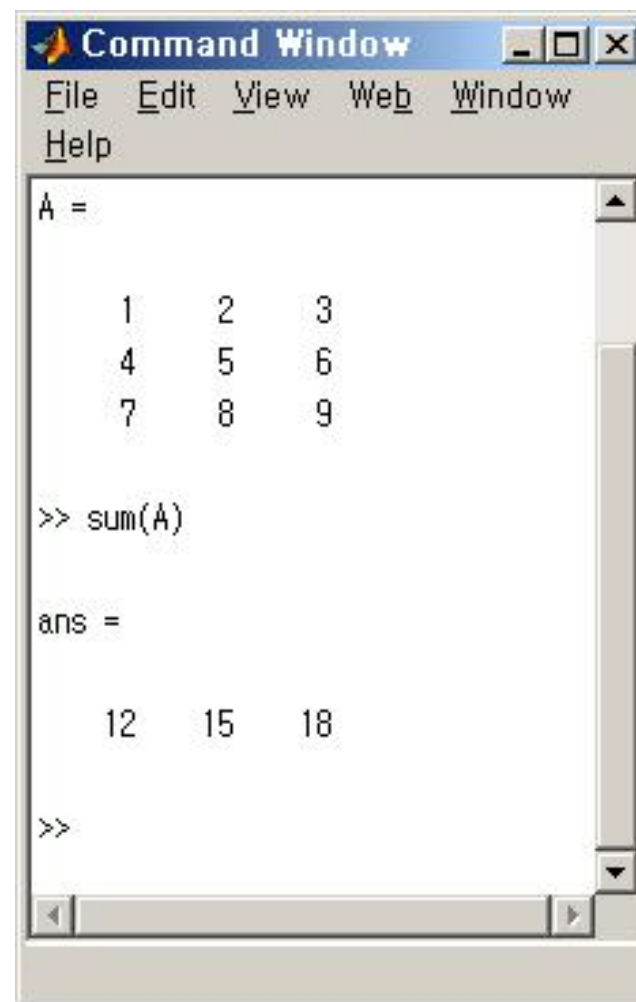
$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



The image shows a MATLAB Command Window. The title bar is 'Command Window'. The menu bar includes 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The command prompt shows the input: `>> A = [1,2,3; 4 5 6 7 8,9]`. Below the command, the output displays the matrix `A =` followed by a 3x3 matrix of values: 1, 2, 3 in the first row; 4, 5, 6 in the second row; and 7, 8, 9 in the third row. The prompt `>>` is visible at the bottom.

행렬 연산(cont'd)

- 행렬의 합
 - 내장함수 : `sum`
 - 기본적으로 Matlab의 모든 연산과 함수의 입력 데이터 처리는 “열벡터” 방향이다.
- 행렬의 Transpose
 - 연산자 : `'`
 - 복소수의 공액



```
Command Window
File Edit View Web Window Help

A =

     1     2     3
     4     5     6
     7     8     9

>> sum(A)

ans =

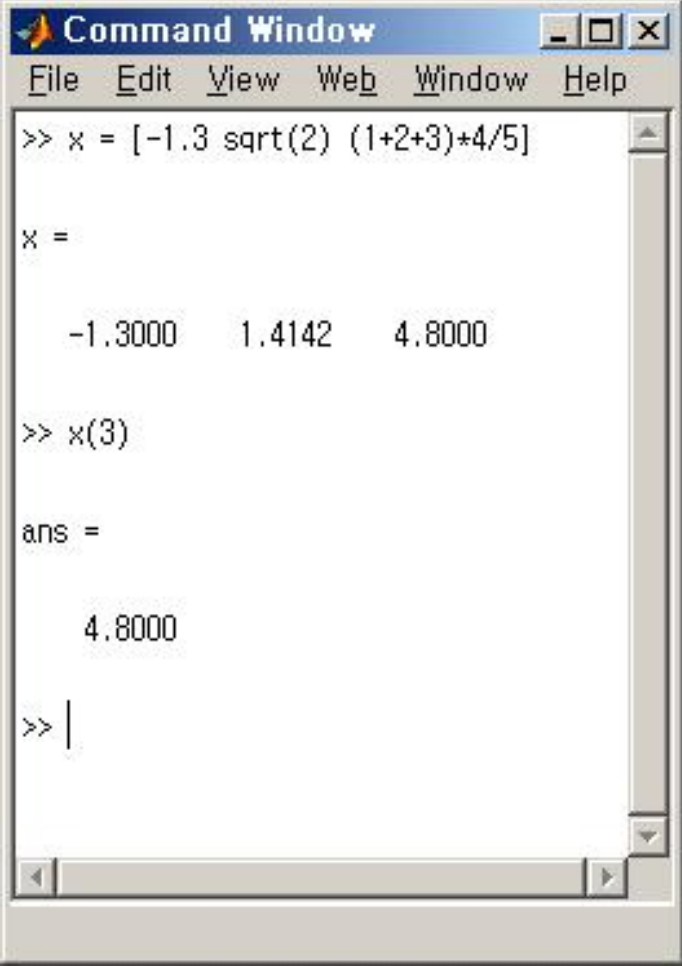
    12    15    18

>>
```

The screenshot shows the MATLAB Command Window interface. At the top is a title bar with the MATLAB logo and the text "Command Window". Below the title bar is a menu bar with options: File, Edit, View, Web, Window, and Help. The main area of the window displays the following text: "A =" followed by a 3x3 matrix of numbers: 1, 2, 3 in the first row; 4, 5, 6 in the second row; and 7, 8, 9 in the third row. Below this, the command ">> sum(A)" is entered. The output "ans =" is shown, followed by the row-wise sums: 12, 15, 18. At the bottom, the prompt ">>" is visible, indicating the command prompt is ready for the next input.

행렬 연산(cont'd)

- 행렬의 원소들은 임의의 Matlab식이 될 수 있다.
- 행렬의 원소를 얻으려면 “()”안에 해당원소의 위치를 넣는다.



A screenshot of the MATLAB Command Window. The window has a title bar 'Command Window' and a menu bar with 'File', 'Edit', 'View', 'Web', 'Window', and 'Help'. The command prompt shows the following sequence of commands and outputs:

```
>> x = [-1.3 sqrt(2) (1+2+3)*4/5]

x =

    -1.3000    1.4142    4.8000

>> x(3)

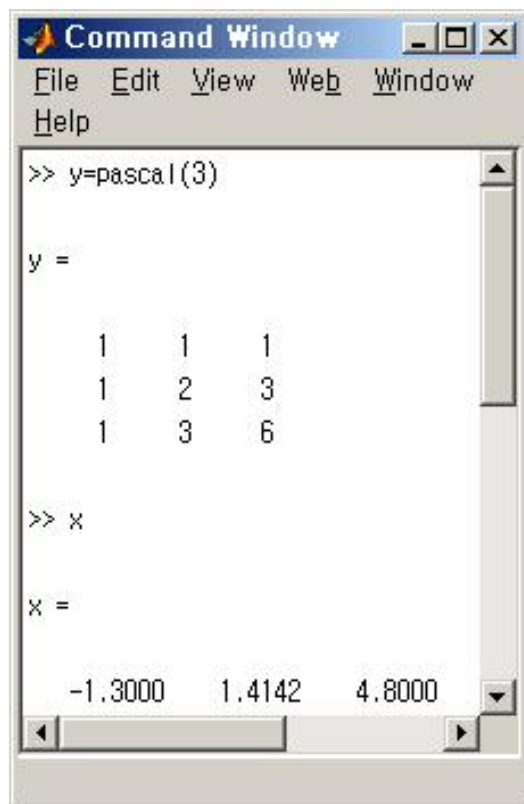
ans =

    4.8000

>> |
```

행렬 연산(cont'd)

- 두개의 행렬을 하나로 합칠 경우 행렬의 차원을 일치시켜야 한다. -> size함수 이용



```
>> y=pascal(3)

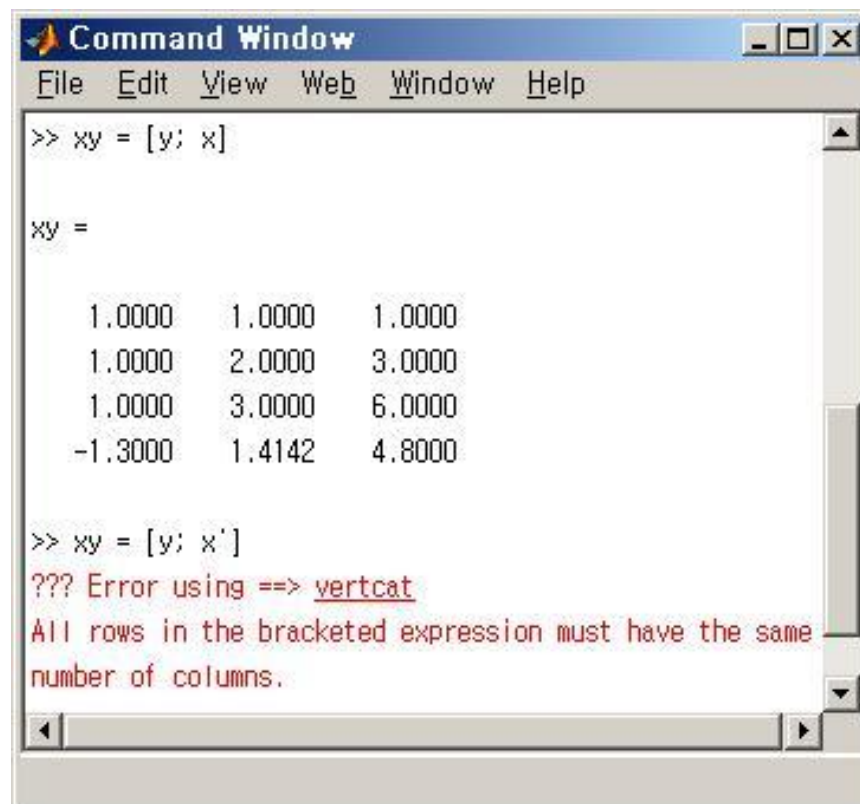
y =

     1     1     1
     1     2     3
     1     3     6

>> x

x =

-1.3000    1.4142    4.8000
```



```
>> xy = [y; x]

xy =

     1.0000     1.0000     1.0000
     1.0000     2.0000     3.0000
     1.0000     3.0000     6.0000
    -1.3000     1.4142     4.8000

>> xy = [y; x']
??? Error using ==> vertcat
All rows in the bracketed expression must have the same
number of columns.
```

Matlab의 연산자

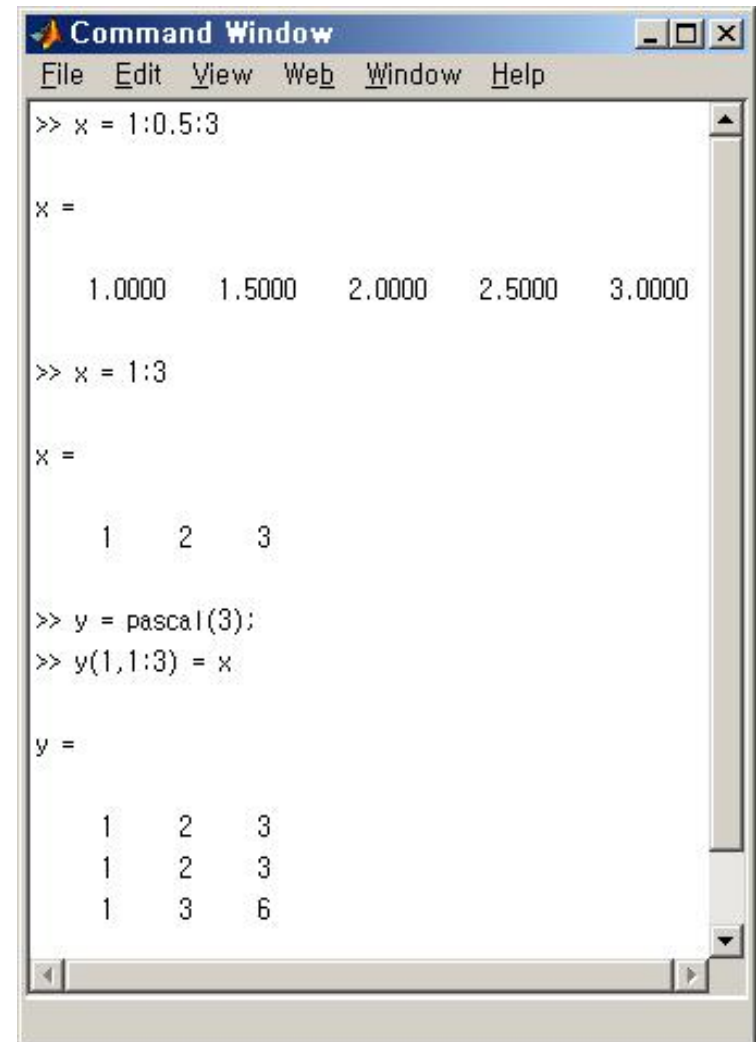
- 연산자의 종류
 - 산술연산자 - 사칙연산
 - 관계연산자 - 같다, 크다, 작다
 - 논리연산자 - AND, OR
- 연산자의 우선순위
 - 산술->관계->논리
 - 왼쪽에서 오른쪽으로, ()가 우선

산술연산자

1	Transpose (.')	Power (.^)	Complex conjugate transpose (')		
2	Unary plus (+)	Unary minus (-)			
3	곱셈 (*)	오른쪽 나눗셈 (./)	왼쪽 나눗셈 (.\)	행렬에서의 좌측과 동일한 연산	
4	덧셈 (+)	뺄셈 (-)			
5	Colon 연산자 (:)				

증가연산자, Colon

- 용법
 - 변수 = 최소값 : 증가치 : 최대값
 - 변수 = 최소값 : 최대값
 - 행렬의 인덱스를 표기할 수도 있음



The screenshot shows the MATLAB Command Window with the following commands and outputs:

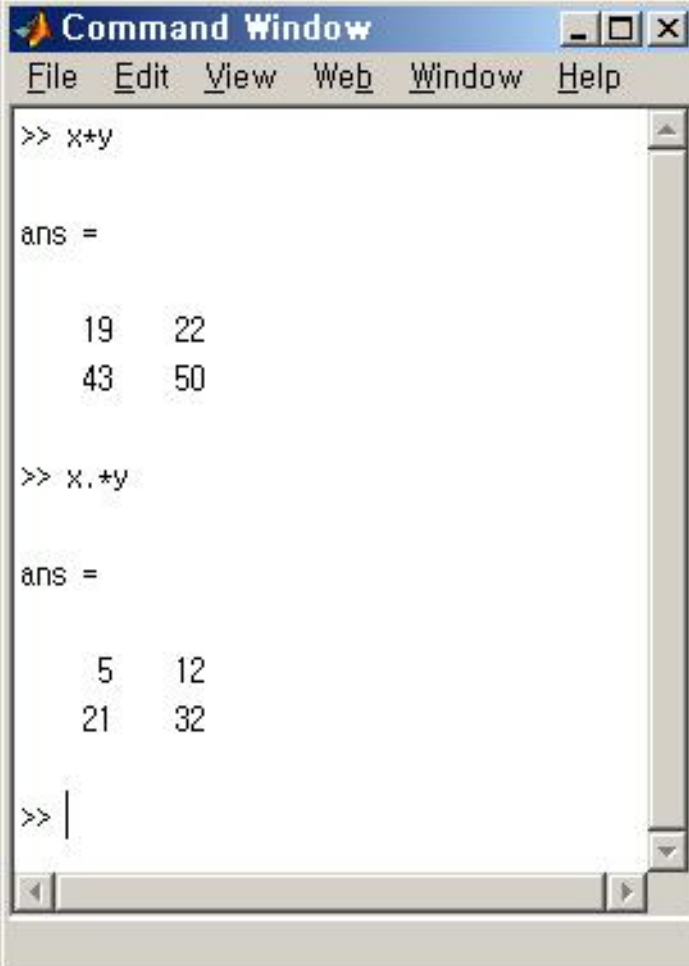
```
>> x = 1:0.5:3  
  
x =  
  
    1.0000    1.5000    2.0000    2.5000    3.0000  
  
>> x = 1:3  
  
x =  
  
     1     2     3  
  
>> y = pascal(3);  
>> y(1,1:3) = x  
  
y =  
  
     1     2     3  
     1     2     3  
     1     3     6
```

점연산자, .

- 행렬의 각각의 요소에 해당 연산을 수행케 함

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

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



```
Command Window
File Edit View Web Window Help

>> x+y

ans =

    19    22
    43    50

>> x.*y

ans =

     5    12
    21    32

>> |
```

관계연산자

관계연산자의 종류	해당 연산자에 대한 설명
<	보다 작다(less than)
<=	작거나 같다(less than or equal to)
>	보다 크다(greater than)
>=	크거나 같다(greater than or equal)
==	같다(equal to)
~=	같지 않다(not equal to)

논리연산자

논리 연산자의 종류	해당 연산자의 설명
&	AND
	OR
~	NOT

논리함수

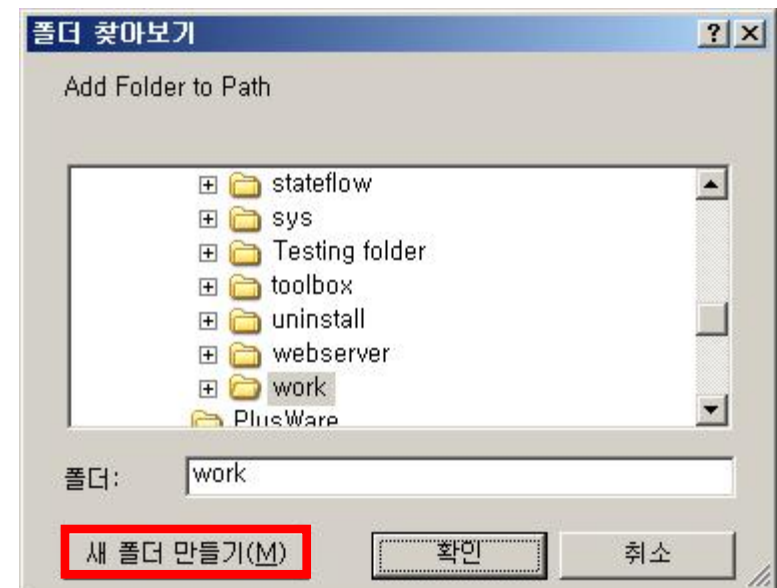
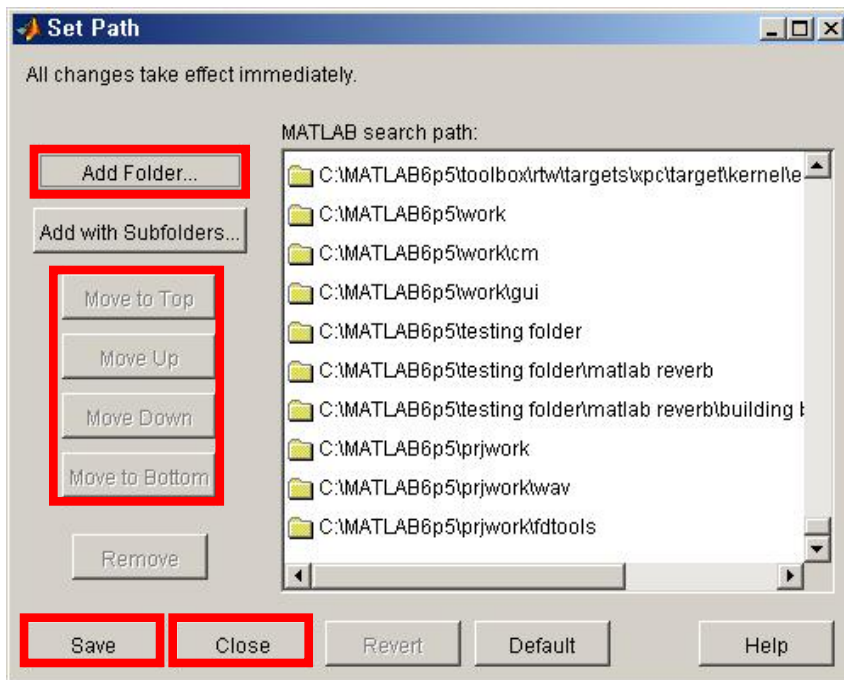
함수의 종류	해당 함수에 대한 설명
xor	Exclusive OR 연산을 수행
all	벡터의 모든 원소가 “참(nonzero)”인 경우에만 “1”을 반환
any	벡터의 임의의 원소가 “참(nonzero)”인 경우에만 “1”을 반환
isnan	벡터의 원소 중에서 “NaN”이 위치한 곳에 “1”을 반환
isfinite	벡터의 원소 중에서 유한한 크기를 갖는 수치, 또는 “NaN”이 아닌곳에 “1”을 반환
isinf	벡터의 원소 중에서 무한한 크기를 갖는 곳에 “1”을 반환

꼭 알아야 할 Matlab 명령어

- help: 명령어에 대한 도움말을 보여줌
- helpdesk: Document CD인스톨 후 사용가능
- lookfor: 함수나 명령어를 찾음
- dir(or ls), delete, type: 도스 명령어와 동일
- who(s): 변수의 정보를 알려줌
- size: 변수의 차원에 대한 정보를 알려줌
- clear: 변수를 메모리에서 삭제
- clc: command window의 내용을 삭제

Matlab 작업공간 만들기

- “File-Set path”선택 후 “Add Folder”로 자신이 주로 작업할 폴더를 선택하거나 새로 만든다.
- 폴더를 아래/위로 옮길 수 있다.



M-file 프로그래밍

- M-file이란 Matlab언어로 쓰여진 파일
 - script mode : 연속된 Matlab 명령어
 - function mode : 입/출력 매개 변수 사용

Script mode의 M-file	Function mode의 M-file
파일 이름은 어떠한 것으로도 사용할 수 있다.	Function이름과 파일 이름이 같아야 한다.
base workspace에 있는 데이터에 대해서 연산	Function workspace에 있는 데이터에 대해서 연산

Function mode의 M-file

- M-file작성 단계
 1. Matlab Editor실행 – “New File” 아이콘 클릭
 2. “function 출력변수 = 함수이름(입력변수)”의 형태로 함수를 정의
 3. “%”으로 시작하는 주석을 단다.
 - “help 함수이름”하면 함수 정의 이후부터 주석이 출력된다.
 4. 함수 본체 작성

Function mode의 M-file의 예

```
C:\WMATLAB6p5Wprjwork\stft.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base

1 function R = stft(X,Fs,xp_s,xp_e)
2 % Short - Time Fourier Transform : function R = stft(X,Fs,xp_s,xp_e)
3 %
4 % --- Parameter description
5 % X          : Input data
6 % Fs         : Sampling Frequency
7 % xp_s       : Start-point of Frequency(Hz)
8 % xp_e       : End-point of Frequency(Hz)
9 % R : R.Duration : Time length of x
10 % R.Fundamental_Frequency : Fundamental Frequency
11 % R.xx       : x axis data of mesh
12 % R.yy       : x axis data of mesh
13 % R.STFTdata : STFT result
14 % $ 2003, 10, 1 made by CHO SANG JIN
15 close all;
16
17 R.Duration = length(X)/Fs;
18 %*****
19 % Step 1. Find SP using autocorrelation method to determine windows size.
20 if(length(X) > 44100)
```

함수정의

주석문

이하 함수 본체

stft Ln 6 Col 45

명령어의 흐름 제어

- 명령어의 흐름을 제어하는 방법 4가지
 - if, else와 elseif
 - switch
 - for
 - while
- 명령어의 끝은 “end”로 표기

if문

if 논리문

명령어 문장

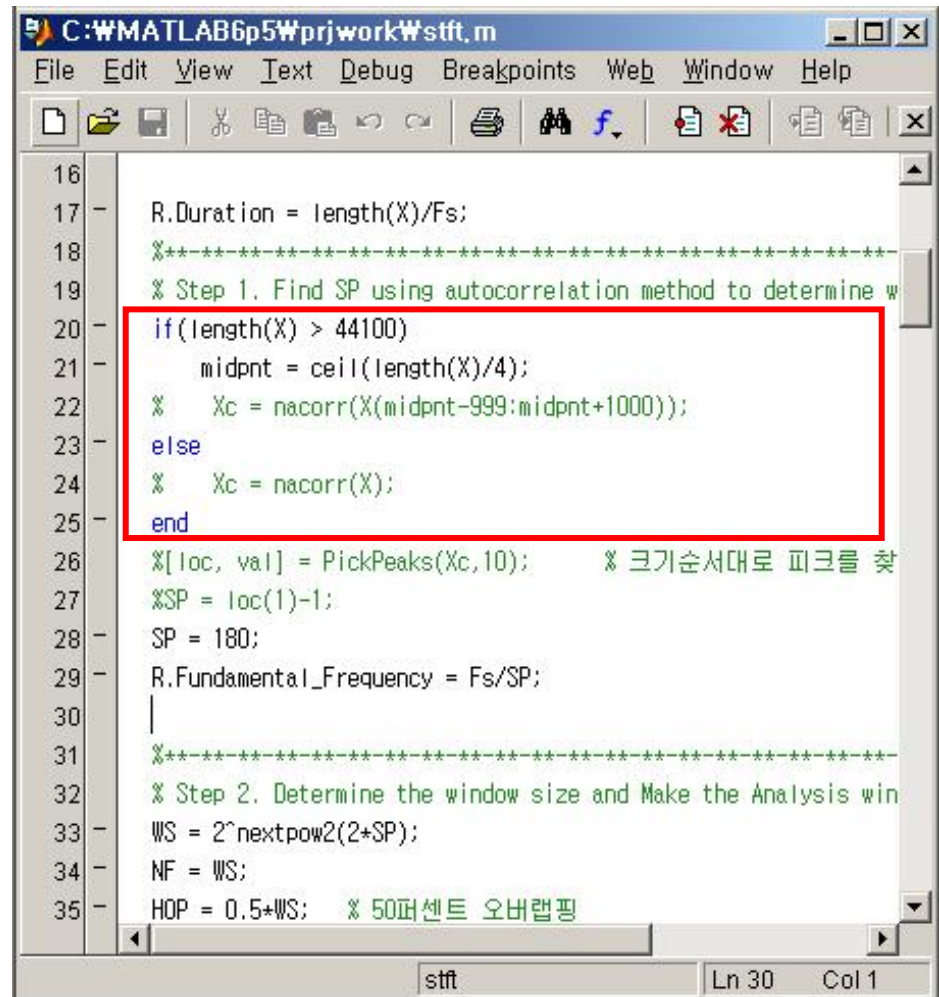
elseif 논리문

명령어 문장

else

명령어 문장

end



The image shows a MATLAB script editor window titled 'C:\WMATLAB6p5WprjworkWstft.m'. The script contains several lines of code. Lines 19 through 25 are enclosed in a red rectangular box, highlighting an if-else statement. The code is as follows:

```
16  
17 - R.Duration = length(X)/Fs;  
18 %*****  
19 % Step 1. Find SP using autocorrelation method to determine w  
20 - if(length(X) > 44100)  
21 -     midpnt = ceil(length(X)/4);  
22 %     Xc = ncorr(X(midpnt-999:midpnt+1000));  
23 - else  
24 %     Xc = ncorr(X);  
25 - end  
26 % [loc, val] = PickPeaks(Xc,10); % 크기순서대로 피크를 찾  
27 % SP = loc(1)-1;  
28 - SP = 180;  
29 - R.Fundamental_Frequency = Fs/SP;  
30 |  
31 %*****  
32 % Step 2. Determine the window size and Make the Analysis win  
33 - WS = 2^nextpow2(2*SP);  
34 - NF = WS;  
35 - HOP = 0.5*WS; % 50퍼센트 오버래핑
```

The status bar at the bottom indicates the file is 'stft' and the cursor is at 'Ln 30 Col 1'.

switch문

switch 변수 또는 식

case 값1

명령어 문장

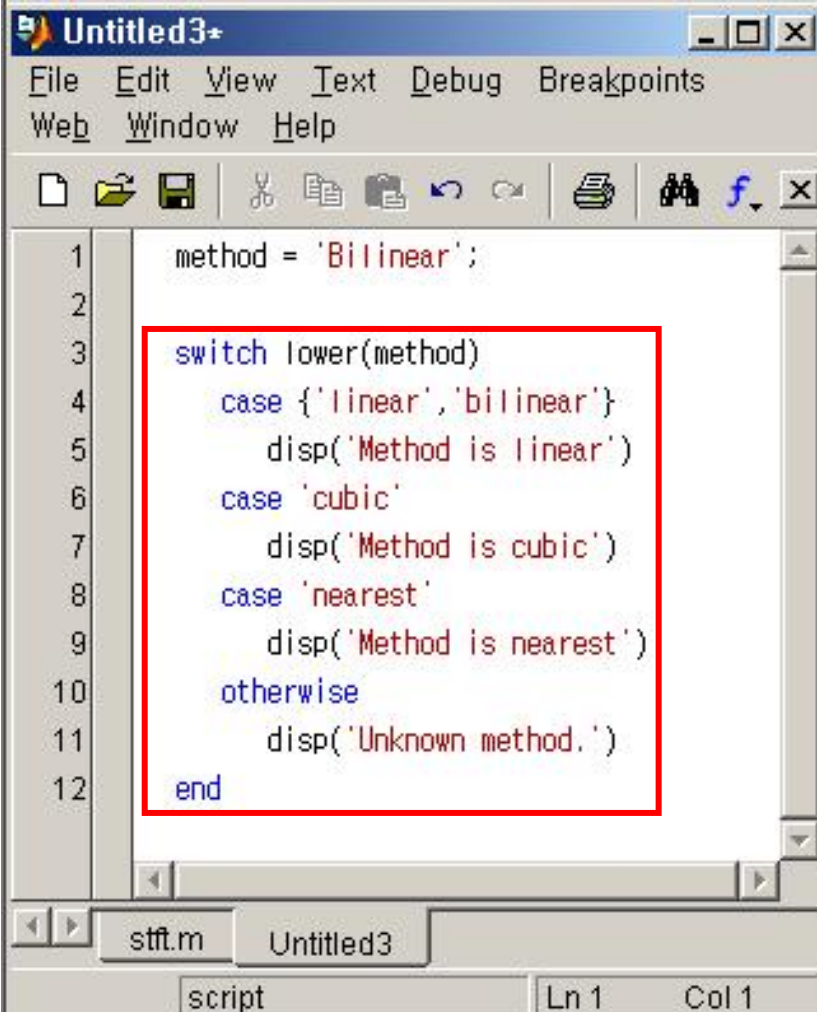
case 값2

명령어 문장

otherwise

명령어 문장

end



The image shows a MATLAB window titled 'Untitled3'. The window contains a script with the following code:

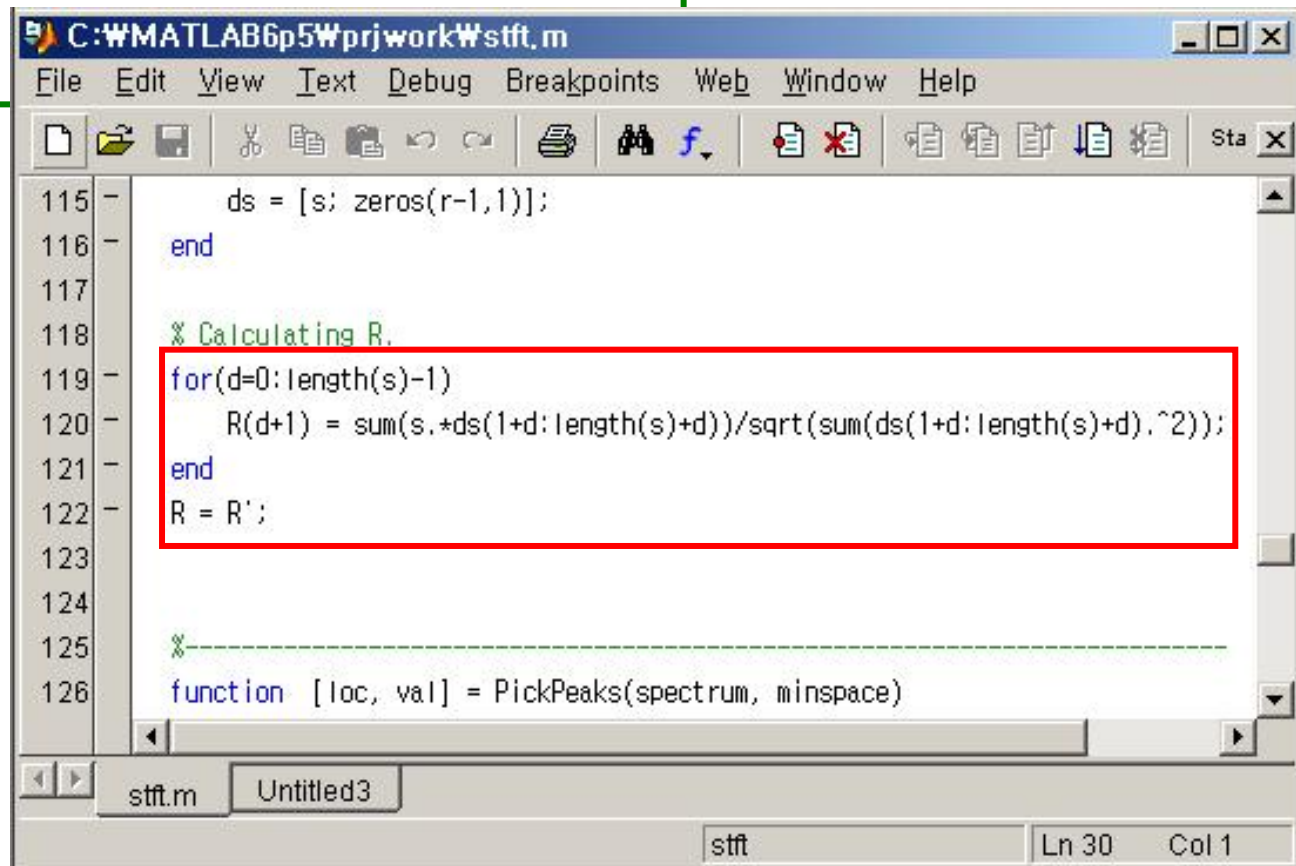
```
1 method = 'Bilinear';  
2  
3 switch lower(method)  
4     case {'linear','bilinear'}  
5         disp('Method is linear')  
6     case 'cubic'  
7         disp('Method is cubic')  
8     case 'nearest'  
9         disp('Method is nearest')  
10    otherwise  
11        disp('Unknown method.')  
12    end
```

The code is displayed in a window with a menu bar (File, Edit, View, Text, Debug, Breakpoints, Web, Window, Help) and a toolbar. The code is highlighted with a red box. The window also shows a status bar at the bottom with 'stf.m', 'Untitled3', 'script', 'Ln 1', and 'Col 1'.

for문

for 인덱스 = 초기치:증가치:최종치
명령어 문장

end



A screenshot of the MATLAB editor window. The title bar shows the file path 'C:\W\MATLAB6p5\prj\work\stft.m'. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. The toolbar contains various icons for file operations and editing. The script editor shows the following code:

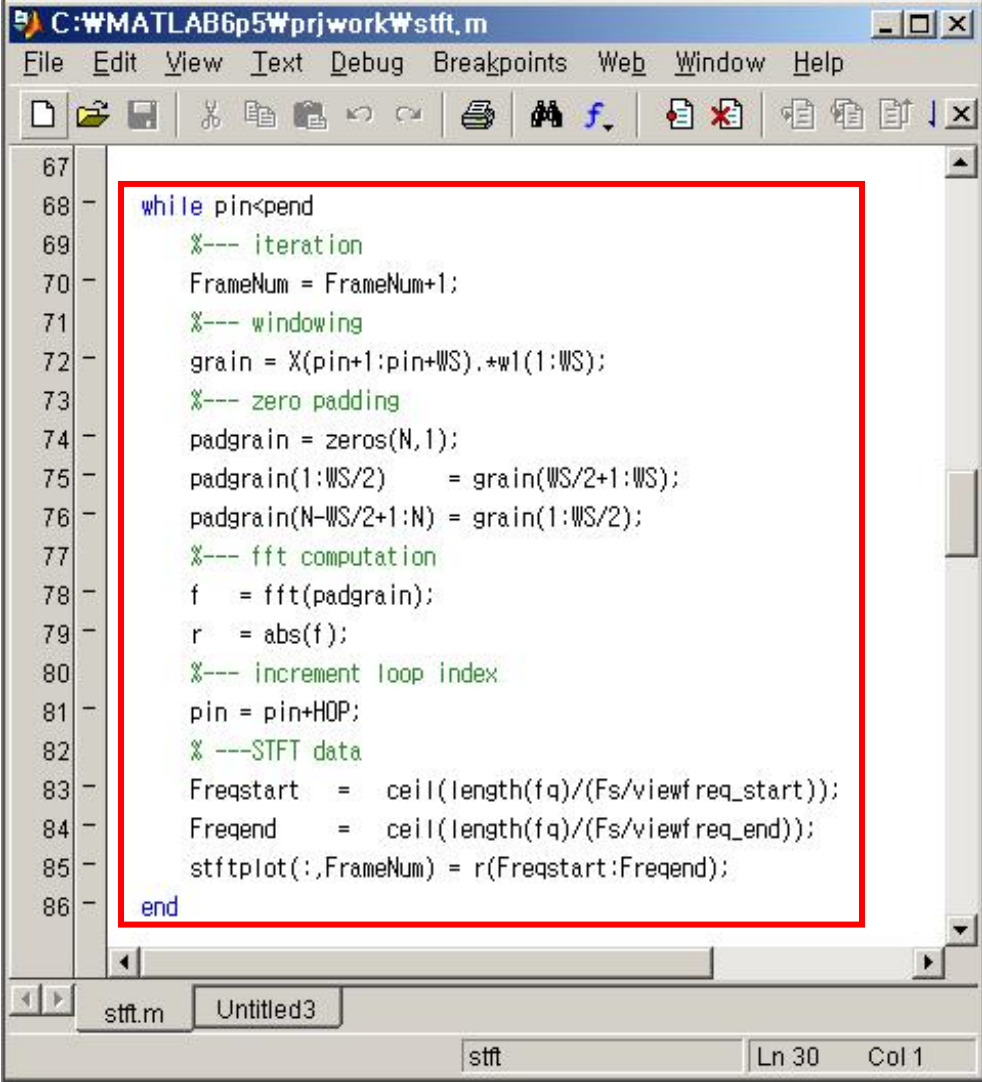
```
115 - ds = [s; zeros(r-1,1)];
116 - end
117
118 % Calculating R.
119 - for(d=0:length(s)-1)
120 -     R(d+1) = sum(s.*ds(1+d:length(s)+d))/sqrt(sum(ds(1+d:length(s)+d).^2));
121 - end
122 - R = R';
123
124
125 %-----
126 function [loc, val] = PickPeaks(spectrum, minspace)
```

The for loop from line 119 to 121 is highlighted with a red rectangular box. The status bar at the bottom indicates the current file is 'stft' at line 30, column 1.

while문

while 논리적인 조건
명령어 문장
end

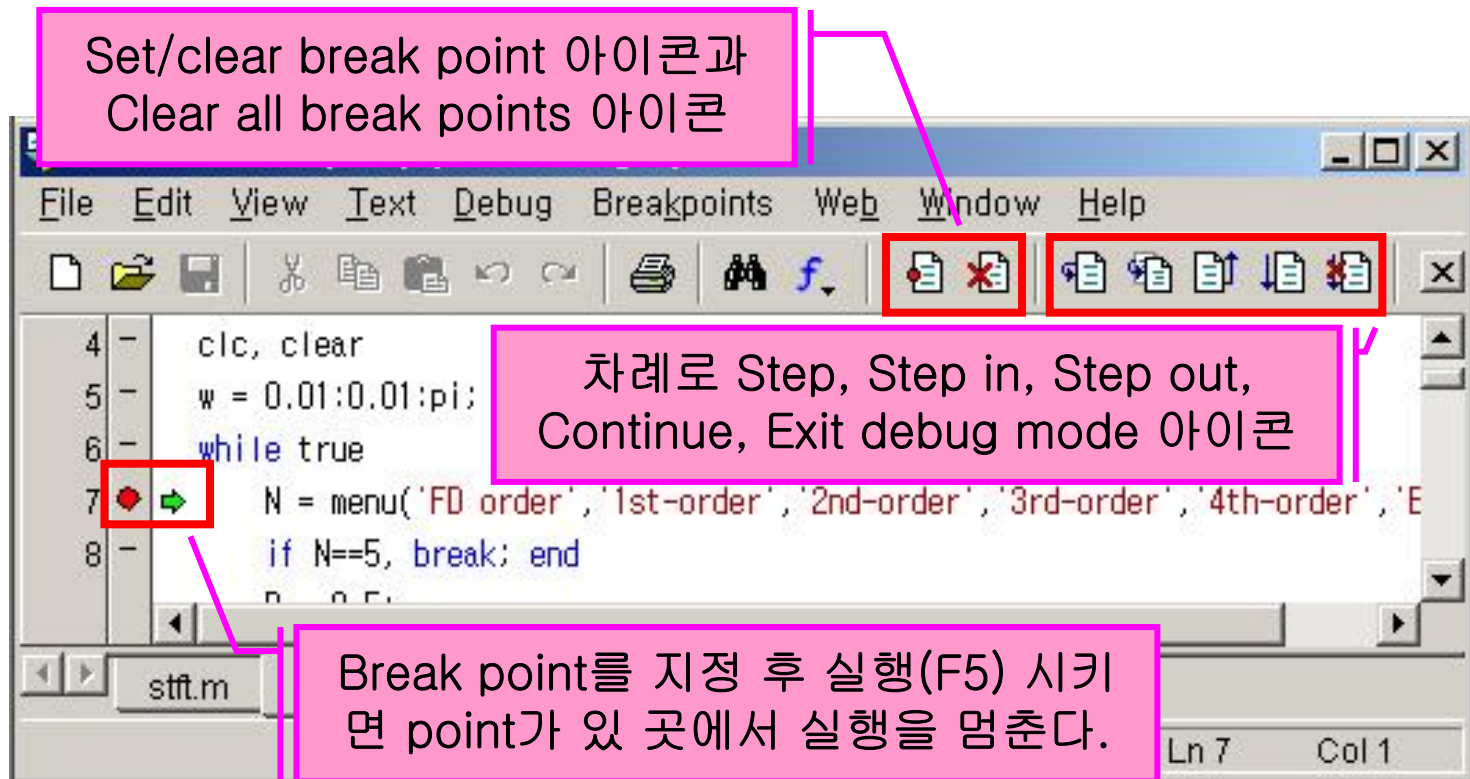
Matlab에도 C언어
에 있는 “break”
명령어가 있는데,
기능은 서로 같다
고 보아도 된다.



```
C:\WMATLAB6p5Wprjwork\stft.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons]
67
68 while pin<pend
69     %--- iteration
70     FrameNum = FrameNum+1;
71     %--- windowing
72     grain = X(pin+1:pin+WS).*w1(1:WS);
73     %--- zero padding
74     padgrain = zeros(N,1);
75     padgrain(1:WS/2) = grain(WS/2+1:WS);
76     padgrain(N-WS/2+1:N) = grain(1:WS/2);
77     %--- fft computation
78     f = fft(padgrain);
79     r = abs(f);
80     %--- increment loop index
81     pin = pin+HOP;
82     % ---STFT data
83     Freqstart = ceil(length(fq)/(Fs/viewfreq_start));
84     Freqend = ceil(length(fq)/(Fs/viewfreq_end));
85     stftplot(:,FrameNum) = r(Freqstart:Freqend);
86 end
```

M-file의 debugging

- 의심이 가는 곳에 “break point” 지정
- debugging 아이콘을 이용하여 debugging



Subfunction

- function mode M-file의 첫번째 함수를 “*primary function*”이라 하고, 이 함수의 이름이 M-file의 이름이 된다.
- primary function안에 새롭게 정의된 함수를 “*subfunction*”이라고 한다.
- subfunction은 해당 primary function안에서, 또는 같은 파일 안에 있는 subfunction에서만 참조가 가능하다.

Subfunction의 사용 예

```
Untitled5+
File Edit View Text Debug Breakpoints
Web Window Help

1  function [avg, med] = newstats(in)
2
3  n=length(in);
4
5  avg=newmean(in,n);
6
7  med=newmedian(in,n);
8
9  function a = newmean(in,n)
10 a = sum(in)/n;
11
12 function m = newmedian(v,n)
13 w=sort(v);
14 if rem(n,2)==1
15     m=w((n+1)/2);
16 else
17     m=(w(n/2)+w(n/2+1))/2;
18 end
19
```

primary
function

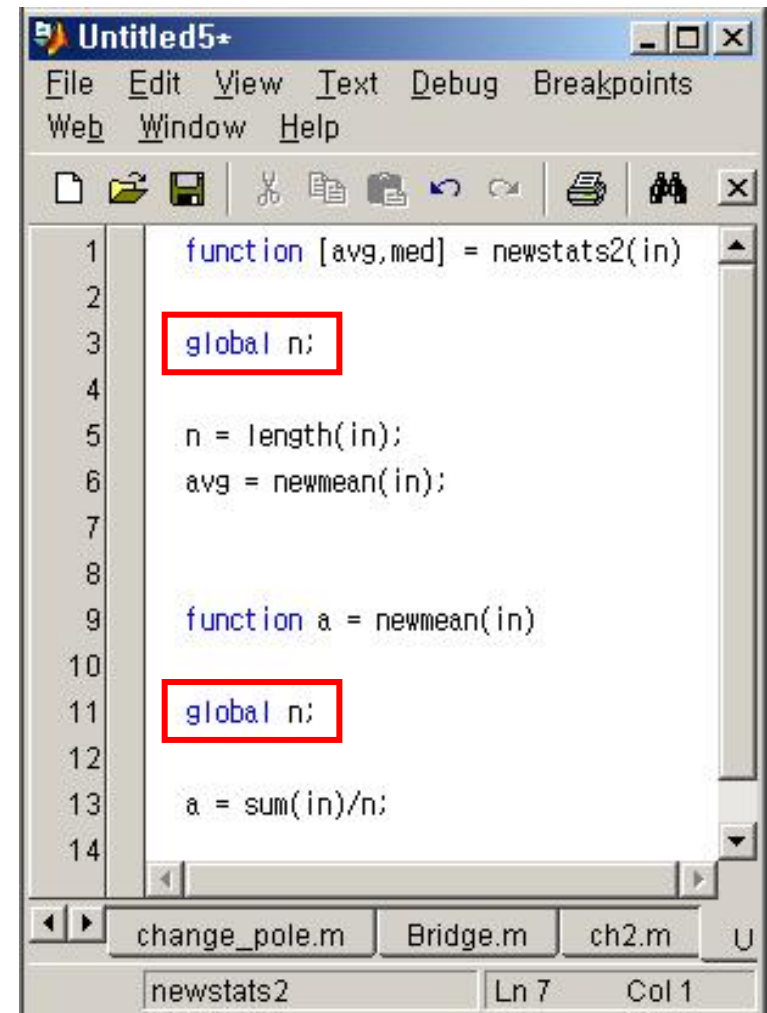
subfunction
호출

subfunction

primary function
에서 사용되는 변
수와 subfunction
에서 사용하는 변
수는 이름이 같더
라도 서로 별개의
것이다.

Global 변수

- primary function
과 subfunction에
동일하게 “global”
변수를 선언하여
야 같은 이름의
변수를 서로 사용
할 수 있다.



```
Untitled5*
File Edit View Text Debug Breakpoints
Web Window Help

1 function [avg,med] = newstats2(in)
2
3 global n;
4
5 n = length(in);
6 avg = newmean(in);
7
8
9 function a = newmean(in)
10
11 global n;
12
13 a = sum(in)/n;
14
```

change_pole.m Bridge.m ch2.m U

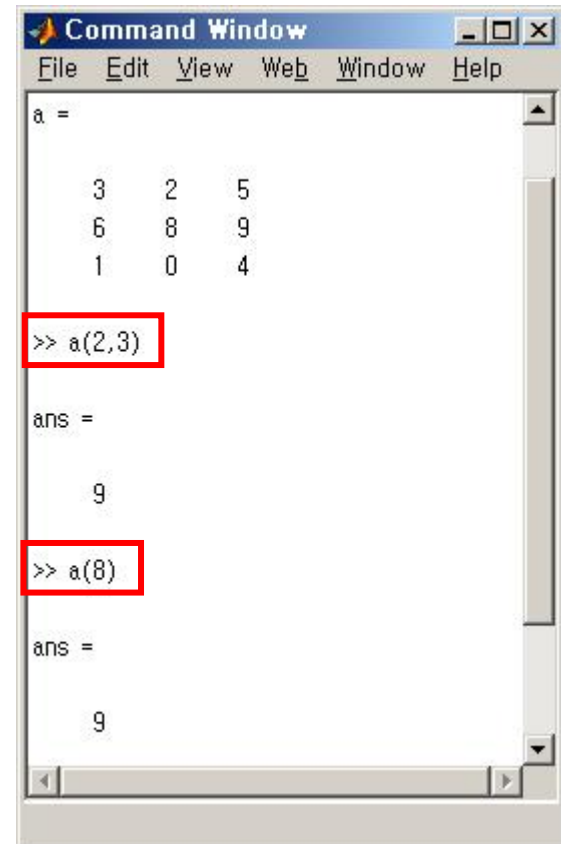
newstats2 Ln 7 Col 1

다차원 배열

- Matlab에서 다차원 배열은 일반적인 2차원 배열에 대한 확장
- 표기법
 - 변수(차원1, 차원2, 차원3, ...)
 - 차원1은 행, 차원2는 열, 차원3은 페이지에 대한 인덱스를 나타낸다.

배열 인덱싱

$$A = \begin{bmatrix} 3 & 2 & 5 \\ 6 & 8 & 9 \\ 1 & 0 & 4 \end{bmatrix} \equiv A = \begin{bmatrix} 3 \\ 6 \\ 1 \\ 2 \\ 8 \\ 0 \\ 5 \\ 9 \\ 4 \end{bmatrix}$$



The image shows a MATLAB Command Window. At the top, the title bar says "Command Window". Below it is a menu bar with "File", "Edit", "View", "Web", "Window", and "Help". The main area shows the following text:

```
a =  
  
     3     2     5  
     6     8     9  
     1     0     4  
  
>> a(2,3)  
  
ans =  
  
     9  
  
>> a(8)  
  
ans =  
  
     9
```

The commands `>> a(2,3)` and `>> a(8)` are highlighted with red boxes. The output for `a(2,3)` is 9, and the output for `a(8)` is also 9.

다차원 배열을 만들어 주는 함수

- `ones(D1,D2,D3,...)`
 - 모든 원소가 “1”로 이루어진 배열
- `zeros(D1,D2,D3,...)`
 - 모든 원소가 “0”으로 이루어진 배열
- `randn(D1,D2,D3,...)`
 - 정규분포된 난수 발생
- `rand(D1,D2,D3,...)`
 - 균등하게 분포된 난수 발생

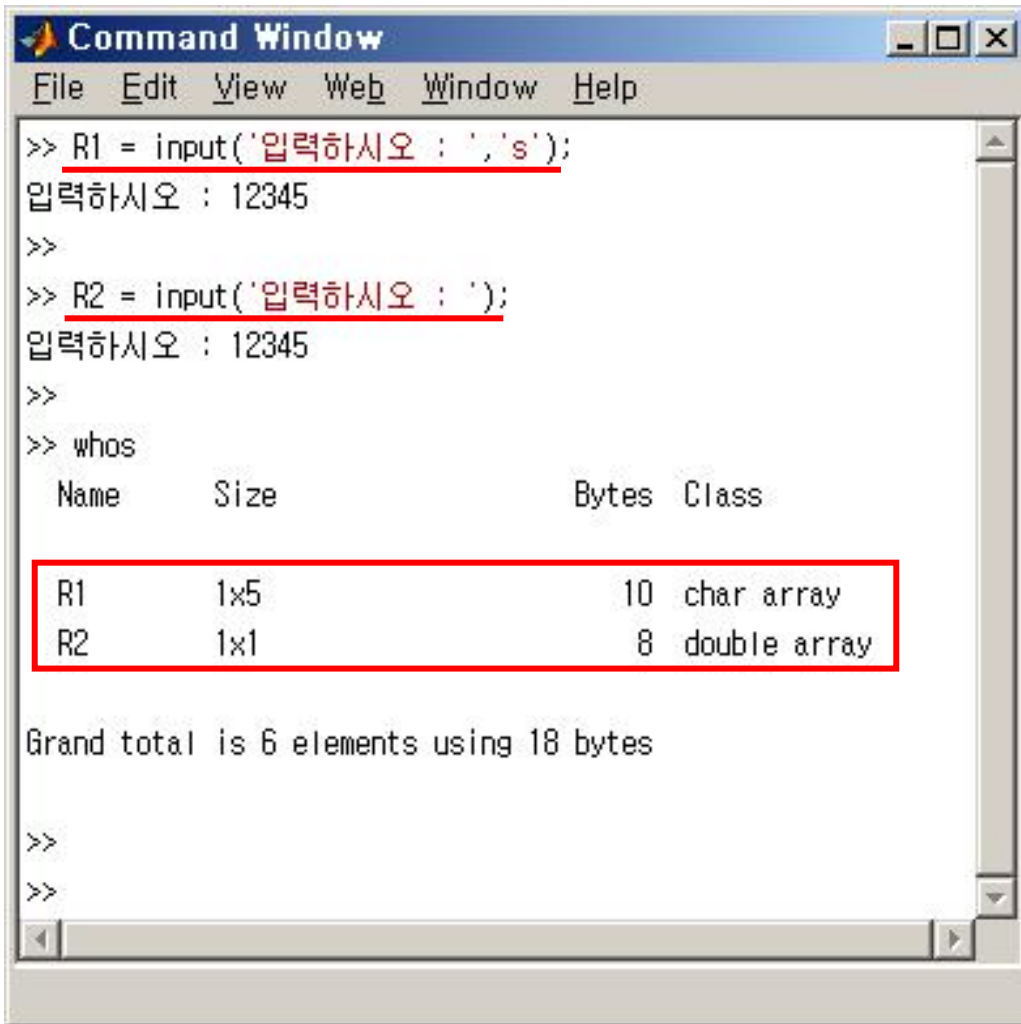
M-file 작성시 유용한 함수들

- input 함수
- menu 함수
- dialog형 함수
 - questdlg, msgbox, helpdlg, errordlg, warndlg etc.

input 함수

- `R = input('prompt','s');`
 - command window에 'prompt'가 출력되고 문자열을 입력할 때까지 기다린다
 - 입력된 문자열은 R에 반환된다.
- `R = input('prompt');`
 - 수식이나 변수의 이름을 입력할때까지 기다린다.
 - 입력된 것이 수식인 경우 workspace에 있는 변수들에 기반을 두고 계산하여 결과를 R에 반환한다.

input 함수 사용 예



```
>> R1 = input('입력하시오 : ','s');
입력하시오 : 12345
>>
>> R2 = input('입력하시오 : ');
입력하시오 : 12345
>>
>> whos
```

Name	Size	Bytes	Class
R1	1x5	10	char array
R2	1x1	8	double array

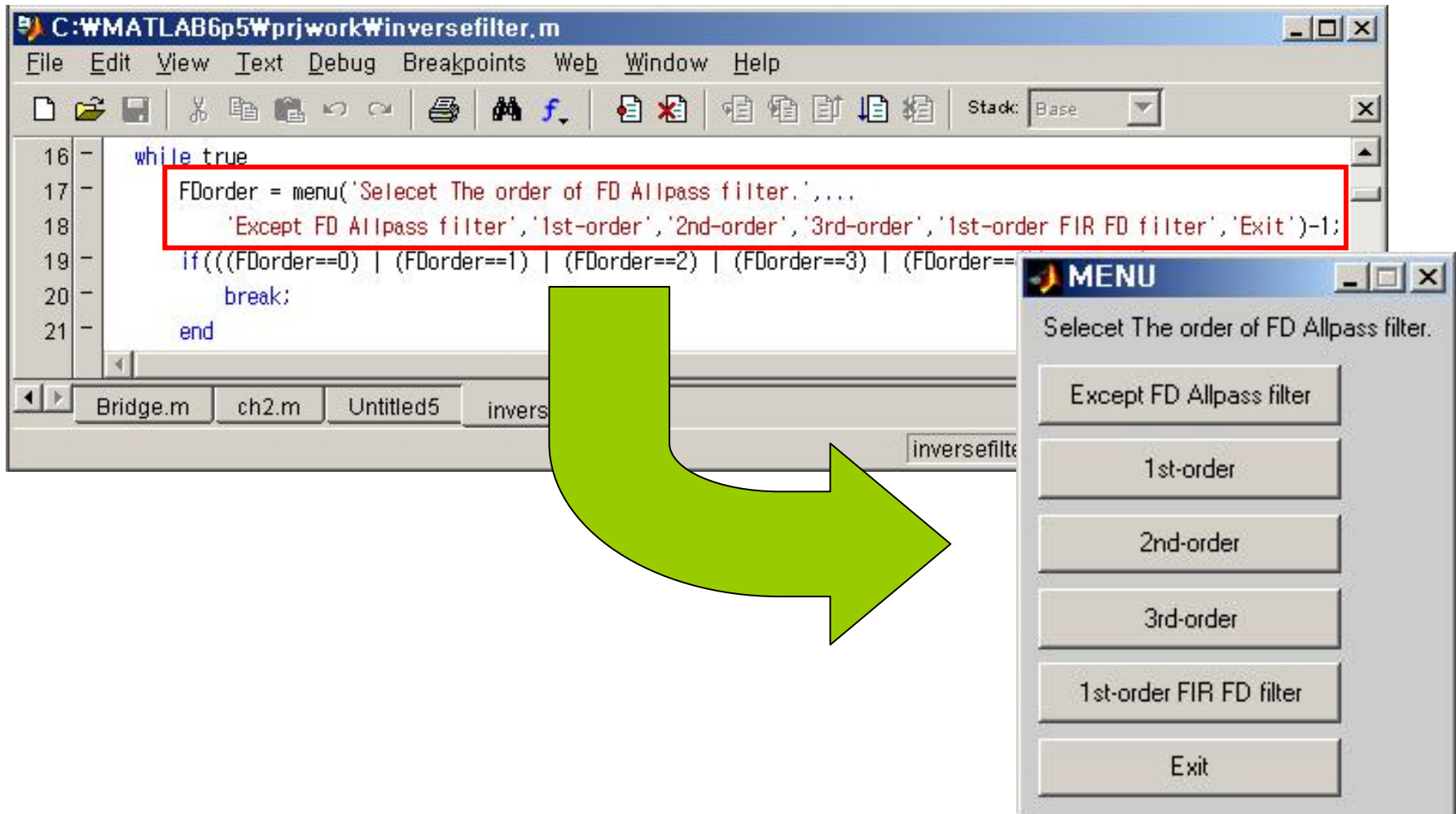
```
Grand total is 6 elements using 18 bytes
>>
>>
```

R1과 R2에 동일하게 '12345'라는 값을 입력하였지만, R1은 char형으로, R2는 double형으로 저장되었다.

menu 함수

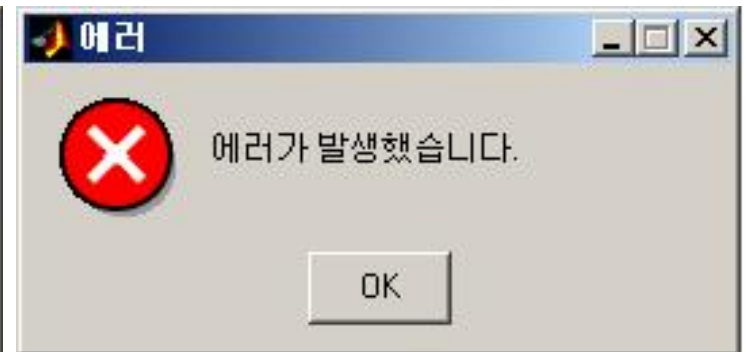
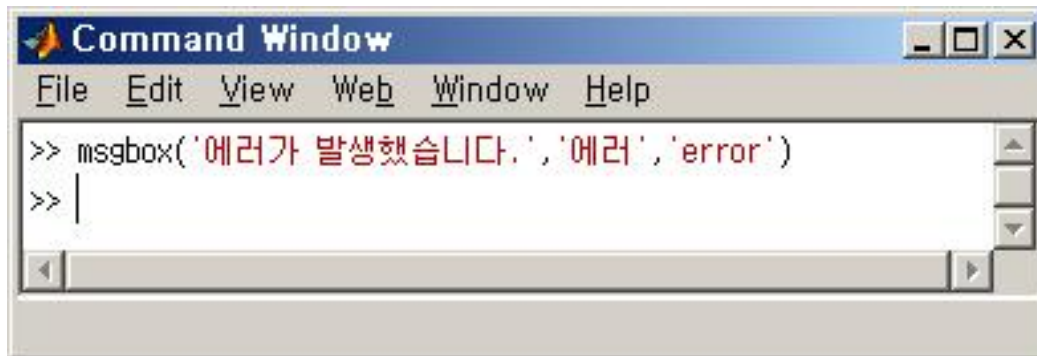
- `R = menu('menu title','s1','s2',...,'sN');`
 - 주어진 'menu title'을 가진 menu window를 생성
 - `s1,...sN` 항목은 “push”버튼의 label로 나타남
 - 임의의 “push”버튼을 클릭하면 `s1`부터 `1,2,...,N`의 값을 `R`에 반환한다.

menu 함수 사용 예



dialog형 함수

- msgbox(message, title, 'icon')
 - title을 박스의 이름으로하고 message를 표시하는 박스를 만든다.
 - message박스 안에 표현할 아이콘 모양을 결정
 - 'icon' 종류 : 'error', 'help', 'warn'



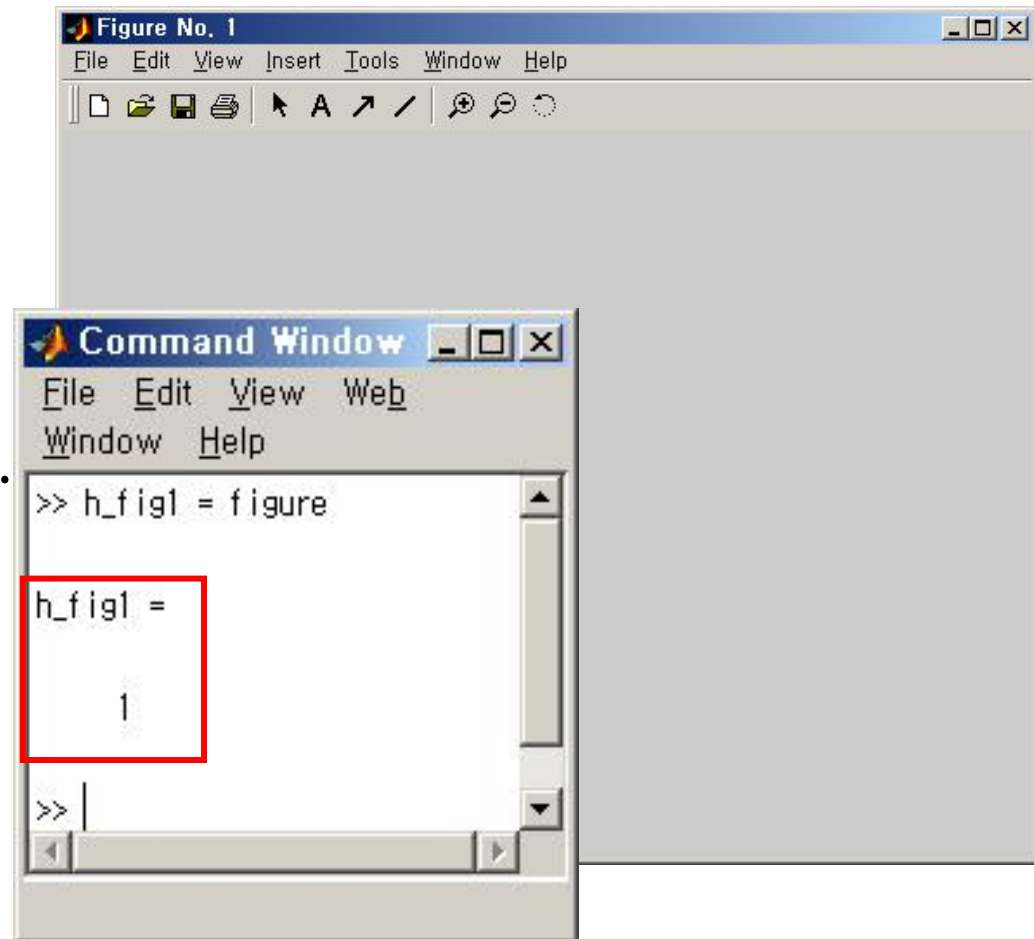
Matlab Graphics

- Handle Graphics system
 - 그래프들의 체계
- Graphics Objects 9가지
 - figure object, line object, axes object,
 - patch object, surface object, text object,
 - image object, uicontrol object, uimenu object

Matlab Graphics(cont'd)

- handle : Matlab 그래프 함수는 객체들을 제어 할 수 있도록 새로 만들어지는 객체마다 고유의 숫자를 반환한다. 이들 고유한 숫자를 handle이라고 한다.

Handle값이 “1”이다.



자주쓰는 그래픽 관련 함수

- 2차원 그래프
 - plot, subplot etc.
- 3차원 그래프
 - plot3, mesh, contour, surf etc.
- 그래픽 개체 라벨링
 - title, xlabel, legend, text, gtext etc.



Matlab을 이용한 신호처리 예제

- 요구조건 1
 - 녹음된 소리의 샘플링 주파수를 알아내고, FFT분석을 통해 기본 주파수 및 주파수 특성을 분석하여라.
- 요구조건 2
 - 위 신호의 STFT 그래프를 보여라.
- 요구조건 3
 - 신호를 입력받고 주파수 특성을 보이는 제어창을 GUI를 이용하여 구현하여라.

요구조건 1 해결 포인트

- `[X, Fs] = wavread('filename.ext');`
 - wave file을 읽어들이어 PCM데이터(X)와 샘플링 주파수(Fs)를 저장한다.
- `fftdata = fft(X,N);`
 - PCM데이터를 N-point DFT를 이용하여 변환한다.
- 기본주파수 확인
 - N이 2의 배수라면 `length(fftdata)/2` 샘플이 Nyquist 주파수($F_s/2$)를 가리킨다.
 - 샘플간의 주파수 간격을 통해 기본주파수를 확인

요구조건 2 해결 포인트

- `win = window(@hamming,ws);`
 - 기본 주파수를 통해 윈도우 사이즈(ws)를 결정하고, 목적에 맞는 윈도우(@hamming)를 결정한다.
- while을 이용한 looping
 - `length(X)/ws`의 횟수만큼 윈도우된 신호 fft반복
- `[xx,yy] = meshgrid(X_fq,[0:size(sX,1)]);`
`mesh(xx, yy, sX);`
 - 반복된 fft의 결과물을 mesh함수를 통해 그림

요구조건 3 해결 포인트

- guide : GUI프로그래밍
 - callback function정의
- axes object – 창 생성
 - `axes(handles.axesObjectTagName); %창 활성화`
`plot(fftdata);`
`guidata(hObject, handles); %handle값 저장`
- uicontrol object – edit, button
 - `handles.file = str2double(get(hObject, 'String'));`
 - `set(handles.editTag, 'String', handles.file);`