

© 1984-2017 The MathWorks, Inc. Protected by U.S and international patents. See mathworks.com/patents. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.



R2017b

**Education Session 01** 

# **MATLAB**

2018년 5월 8일 @ 대기과학과 학부전산실 | 서승환





sets

• 변수를 선언하는 방법

- 변수의 종류
- 배열의 구조
- 배열을 선언하는 방법

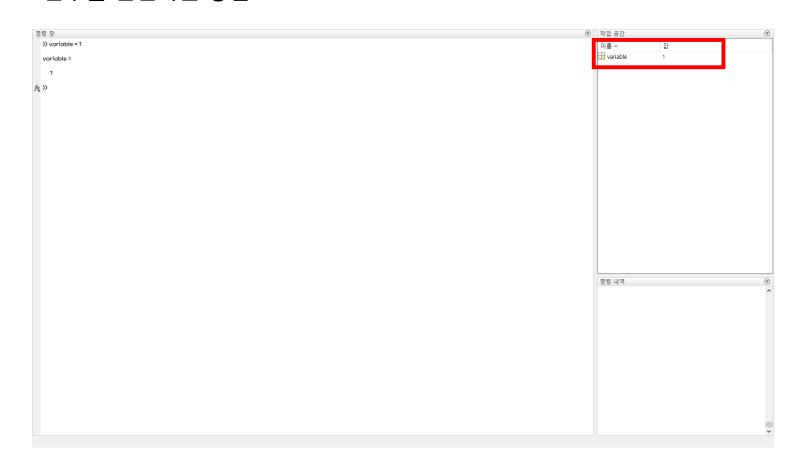


- 사칙연산과 제곱
- 배열 간 사칙연산
- 합, 평균, 표준편차
- 여러 가지 함수의 응용



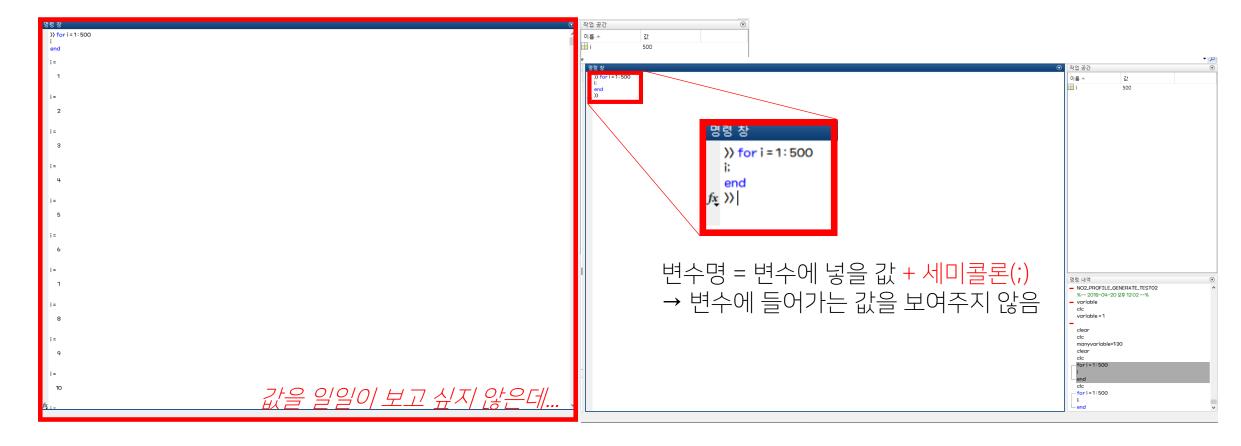
- 조건문의 기본 구조
- 반복문의 구조
- 조건문과 반복문의 활용

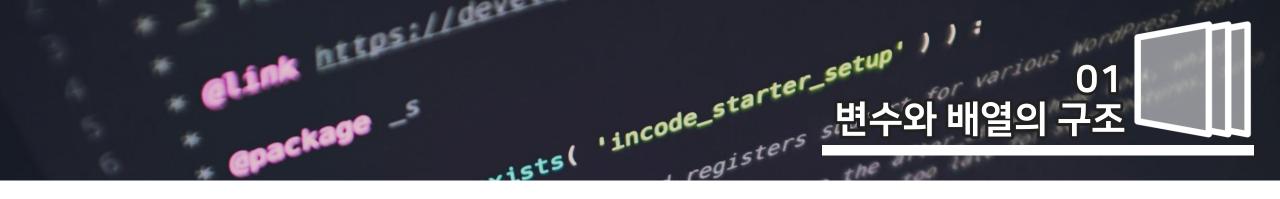
#### • 변수를 선언하는 방법



변수명 = 변수에 넣을 값 (대소문자 구별함)

• 변수를 선언하는 방법





• 변수의 종류

double/single : 실수형 변수

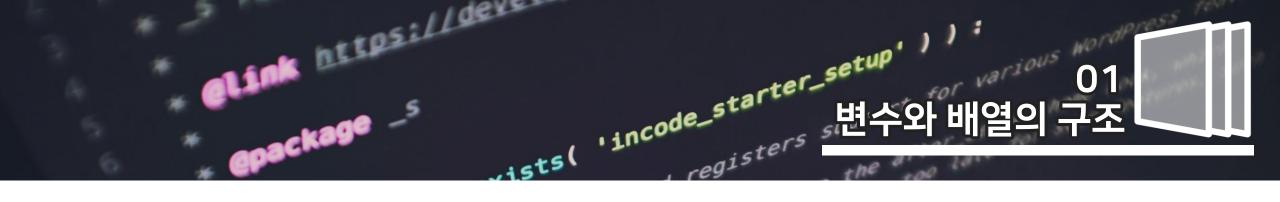
character(char)/string : 문자형 변수

datetime : 날짜형 변수

complex double : 복소수형 변수

.

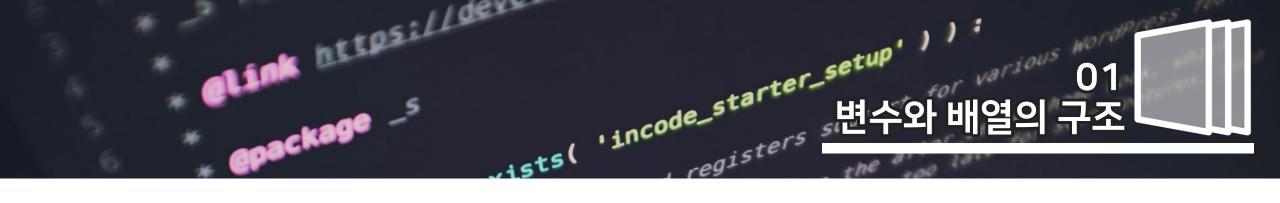
하지만 지금은 double, character/string 정도만 알면 된다





이론적으로는 무한한 차원까지 가능(실제로는 메모리 문제로 거의 불가능)

우리는 2차원 배열만 살펴볼 것(보통 1~3차원 배열을 많이 씀)



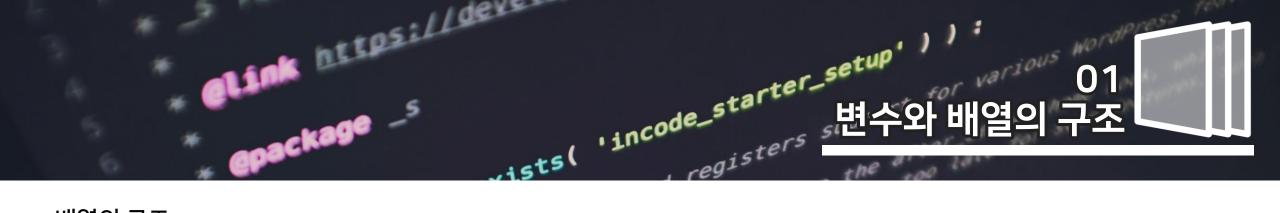
예제 코드 MATLAB\_EX\_01.m 실행

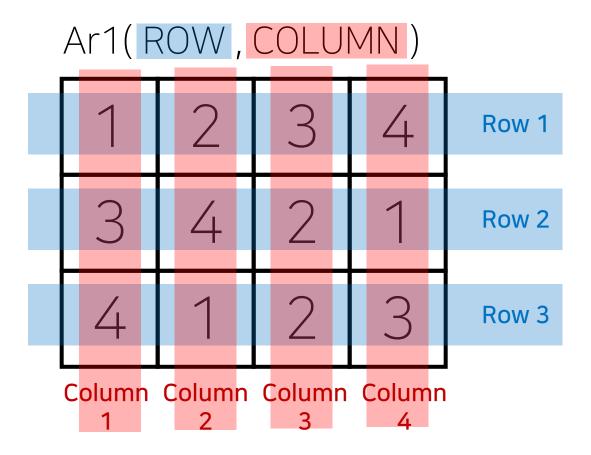
명령 창

 $f_{x}$  >> MATLAB\_EX\_01

Ar1:3 X 4 배열

1	2	3	4
3	4	2	\
4	1	2	

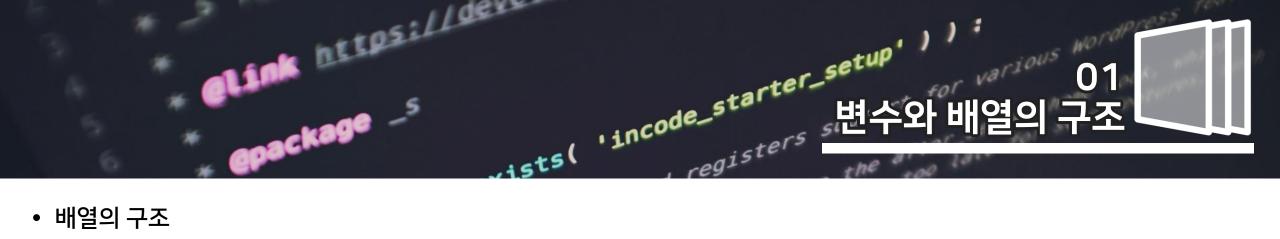




$$[Q1] Ar1(2, 1) = ?$$

$$[Q2] Ar1(1, 4) = ?$$

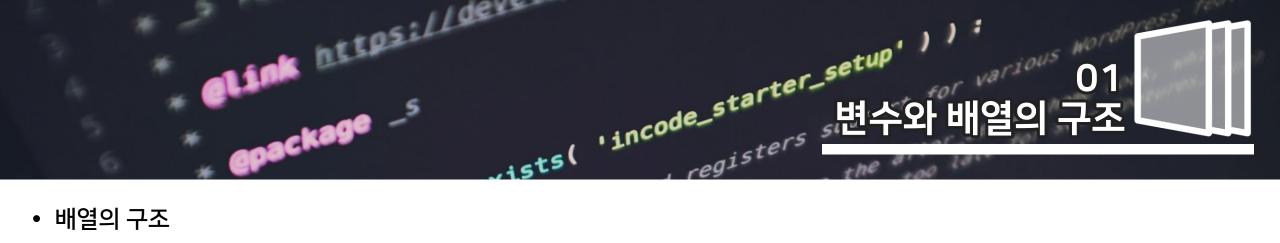
Ar1(4, 2)를 치면 뭐가 나올까?



범위 지정은 콜론(:)으로 한다

1	2	3	4
3	4	2	1
4	1	2	3

[Q] Ar1(2:3,1:3) = ?

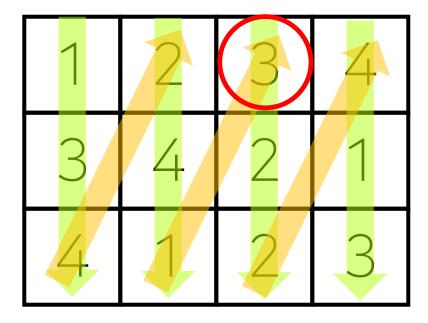


콜론(:)만 있으면 전체 범위가 지정된다

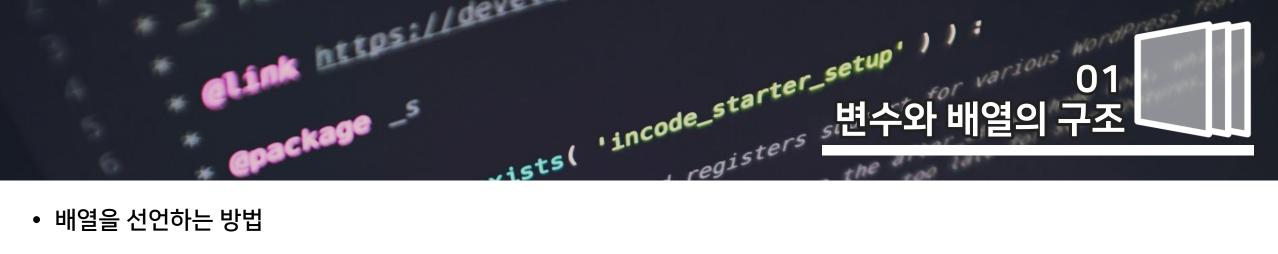
1	2	(A)	4
3	4	2	1
4	1	2	3

$$[Q] Ar1(3,:) = ?$$

한 줄짜리 배열로도 생각할 수 있다



$$[Q] Ar1(7) = ?$$

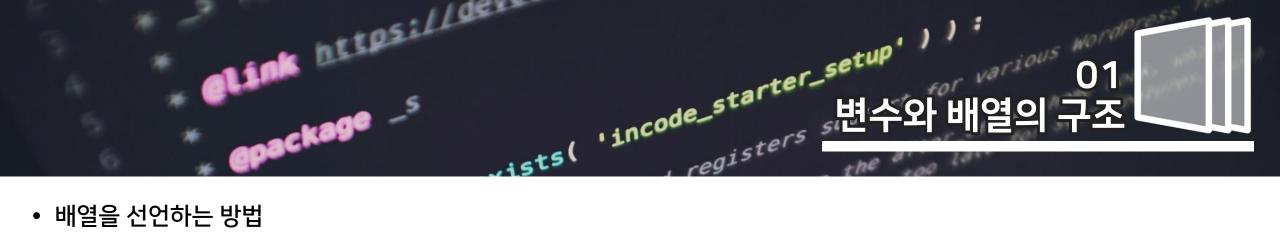


ar1 = [1, 2, 3, 4];

ar2 = [1; 3; 4];

1	2	3	4
3	4	2	1
4	1	2	3

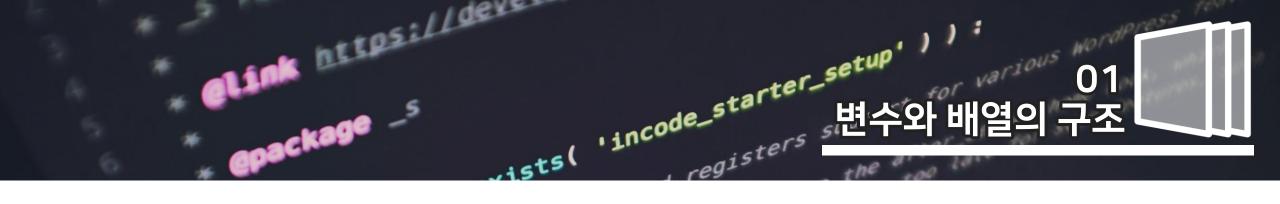
ar3 = [1, 2, 3, 4; 3, 4, 2, 1; 4, 1, 2, 3];



배열의 끝을 나타내는 명령어 : end

1	2	$\sim$	4
3	4	2	()
4	1	2	3

[Q] Ar1(2, end) = ?



예제 코드 MATLAB\_EX\_02.m 실행

배열의 크기를 확인하는 명령어 : length(1차원), size(n차원)

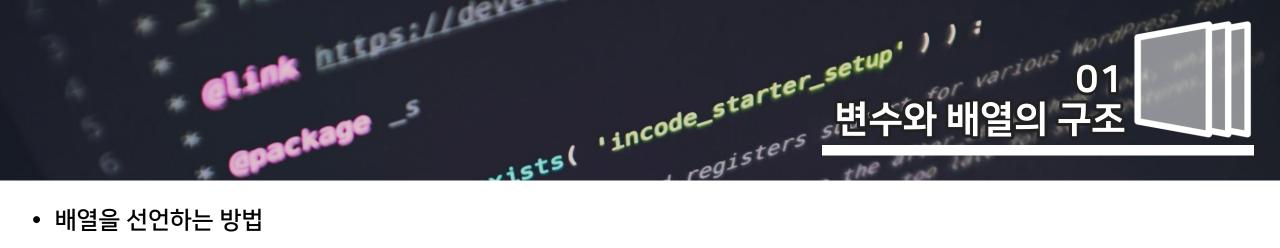
> length : 배열의 행과 열 중 더 큰 숫자를 출력하는 명령어

> size : 배열의 각 차원별 크기를 <u>배열로</u> 출력하는 명령어

[x, y, z] = size(Ar3)

특정 차원의 크기는 보고 싶지 않다면 변수 대신 물결표 활용

 $(ex) [x, \sim, z] = size(Ar3)$ 

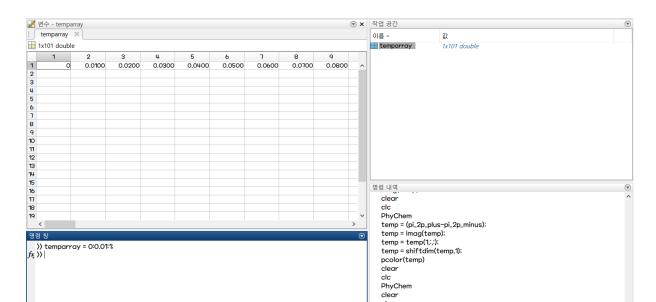


등차수열의 배열을 선언하는 방법

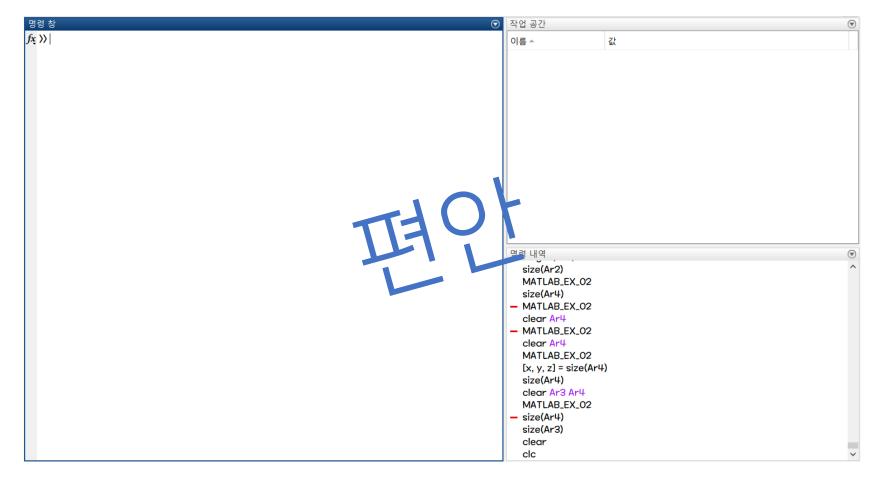
배열\_이름 = 시작\_숫자:간격:끝\_숫자

※ 등차수열이 되어야 하므로 단조증가/감소해야 함

(ex) temparray = 0:0.01:1

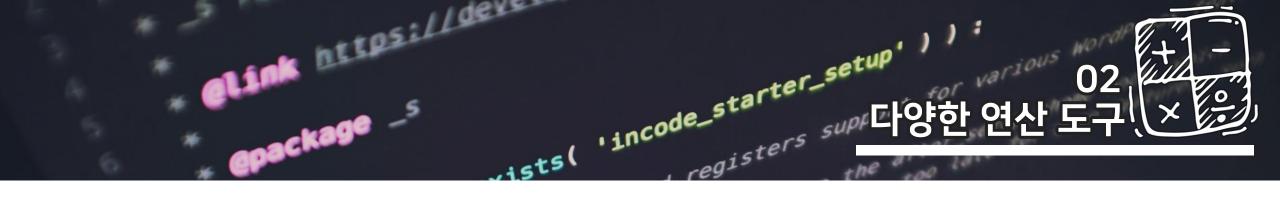


• 명령 창/작업 공간 정리



clear : 작업 공간을 초기화하는 명령어

clc: 명령 창을 초기화하는 명령어



# • 사칙연산과 제곱

덧셈, 뺄셈은 단순히 +, -를 이용

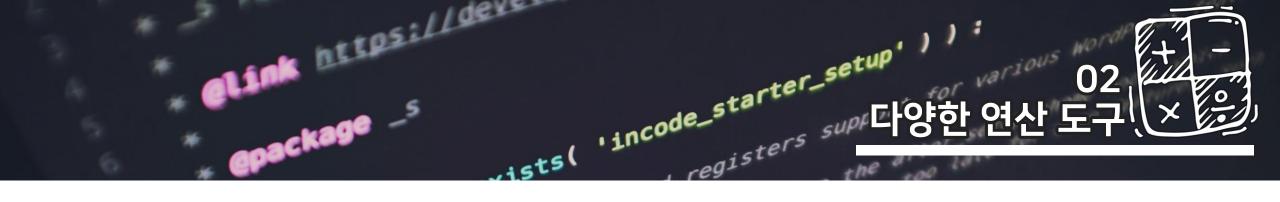
$$a = 3 + 2;$$
  $b = 2.3 - 4.7;$ 

변수끼리의 곱셈, 나눗셈 역시 단순히 \*, /를 이용

$$c = a * b;$$
  $d = c / 3;$ 

거듭제곱은 ^ 키를 이용

$$e = c^3$$
;



# • 배열 간 사칙연산

예제 코드 MATLAB\_EX\_03.m 실행

덧셈, 뺄셈은 똑같이 +, -를 이용

42 12

[Q1] 5 + Ar4 = ?

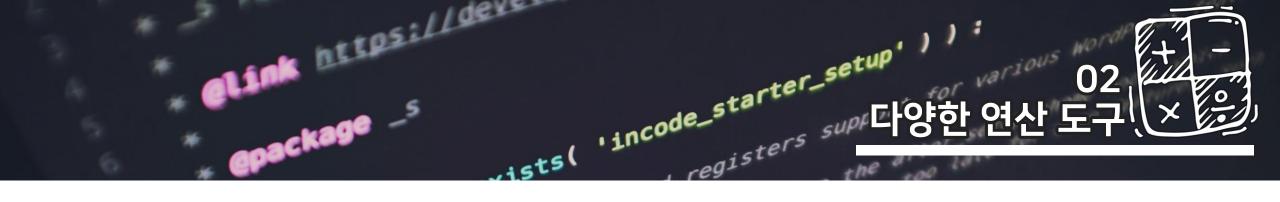
\*, /, ^를 이용한 곱셈, 나눗셈, 거듭제곱은 <u>행렬곱</u>으로 처리됨

a <sub>1</sub>	a <sub>2</sub>	b <sub>1</sub>	b <sub>2</sub>		
a <sub>3</sub>	a <sub>4</sub>	b <sub>3</sub>	b <sub>4</sub>		

a<sub>1</sub>b<sub>1</sub>+a<sub>2</sub>b<sub>3</sub>

[Q2] Ar4 \* Ar5 = ?

 $[Q3] Ar6^2 = ?$ 

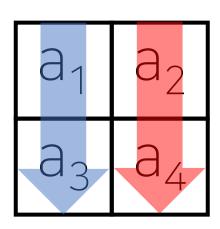


• 배열 간 사칙연산

평범한 곱셈/나눗셈/거듭제곱을 하고 싶다면 연산자(\*, /, ^) 앞에 점(.)을 찍어주면 된다

sum : 합을 구하는 명령어

※ 별도의 추가 명령이 없을 경우 열마다의 합을 구함



$$sum(Ar5) = [a_1 + a_3, a_2 + a_4]$$



행마다의 합을 구하고 싶으면?

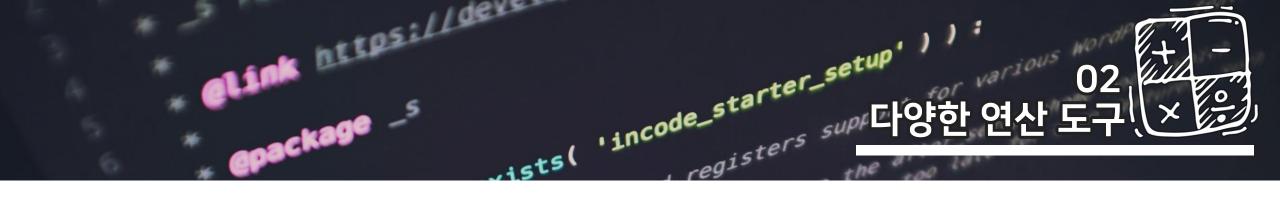
sum(Ar5, 2)

배열의 차원을 의미

1 : 열마다의 합(Default) 2 : 행마다의 합(2차원 이상) 3 : 깊이마다의 합(3차원 이상)

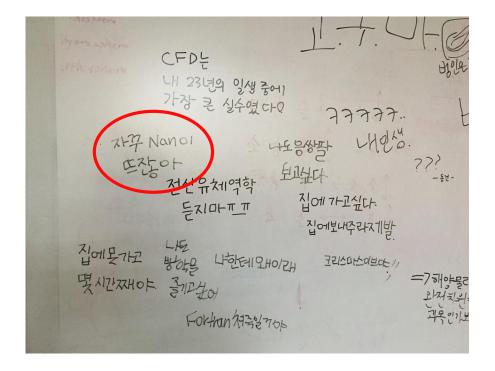
1차원 배열에서는 전체 합을 계산

→ 2차원 배열에서 전체 합을 구하기 위해서는 sum(sum(Ar5))와 같이 하면 됨

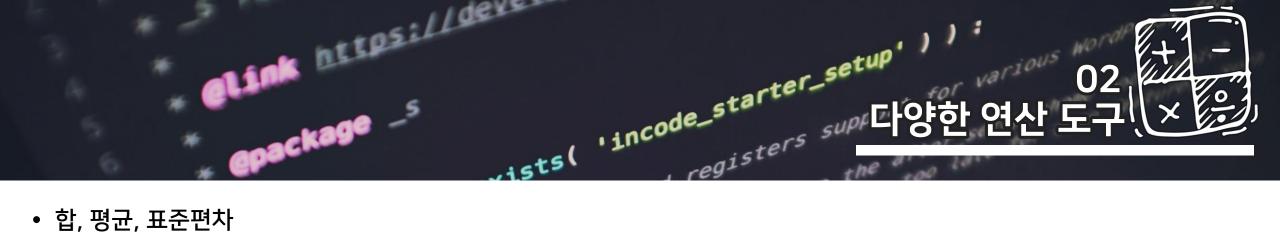


NaN(Not a Number): 숫자가 아닌 값(데이터형은 숫자)

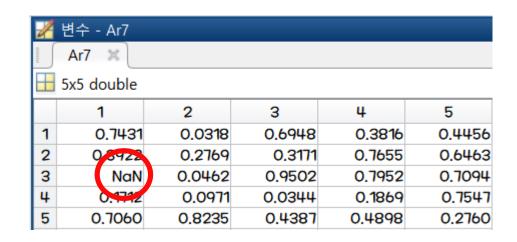
→ MATLAB을 포함한 다양한 프로그래밍 언어에서 <u>계산을 꼬이게 만드는 주범</u>



전세 함니		ž.	전투 - data		⊎ x
data	× tempa	array × DAI	x ans	×	
139536	0x6 double	#			
	1	2	3	4	5
1930	111121	7.3458e+05	0.0040	0.0300	0. ^
1931	111121	7.3458e+05	0.0040	0.0350	0.
1932	111121	7.3458e+05	NaN	NaN	
1933	111121	7.3458e+05	NaN	NaN	
1934	111121	7.3458e+05	NaN	NaN	
1935	111121	7.3458e+05	NaN	NaN	
1936	111121	7.3458e+05	NaN	NaN	
1937	111121	7.3458e+05	NaN	NaN	
1938	111121	7.3458e+05	NaN	NaN	
1939	111121	7.3458e+05	NaN	NaN	
1940	111121	7.3458e+05	NaN	NaN	
1941	111121	7.3458e+05	NaN	NaN	
1942	111121	7.3458e+05	NaN	NaN	
1943	111121	7.3458e+05	NaN	NaN	
1944	111121	734585	NaN	NaN	
1945	111121	7.3459e+05	NaN	NaN	
1946	111121	7.3459e+05	NaN	NaN	
1947	111121	7.3459e+05	NaN	NaN	
1948	111121	7.3459e+05	NaN	NaN	
1949	111121	7.3459e+05	NaN	NaN	
1950	111121	7.3459e+05	NaN	NaN	
1951	111121	7.3459e+05	NaN	NaN	
1952	111121	7.3459e+05	NaN	NaN	
1953	111121	7.3459e+05	NaN	NaN	
1954	111121	7.3 <mark>45</mark> 9e+05	NaN	NaN	
1955	111121	7.34 <mark>59e+</mark> 05	NaN	NaN	
1956	111121	7.3459e+05	NaN	NaN	
1957	111121	7.3459e+05	NaN	NaN	
1958	111121	7.3459e+05	NaN	NaN	
1959	111121	7.3459e+05	0.0050	0.0430	0.
1960	111121	7.3459e+05	0.0040	0.0460	0.
1961	111121	7.3459e+05	0.0040	0.0420	0.
1962	111121	7,3459e+05	0.0040	0.0320	0.
1963	111121	7.3459e+05	0.0040	0.0250	0.
1064	*****	7 24504 . 05	0.0040	0.0360	0 4



예제 코드 MATLAB\_EX\_04.m 실행



$$[Q1] sum(Ar7) = ?$$

>> sum(Ar7)

ans =

2.4353 2.6189 2.8320 [Q2] sum(sum(Ar7)) = ?

>> sum(sum(Ar7))

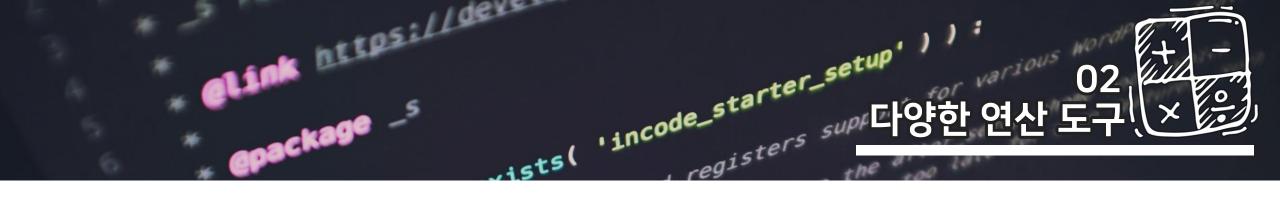
ans =

NaN

어떻게 해야 NaN을 무시하고 계산할 수 있을까?

a sum(Ar7,'omitnan')

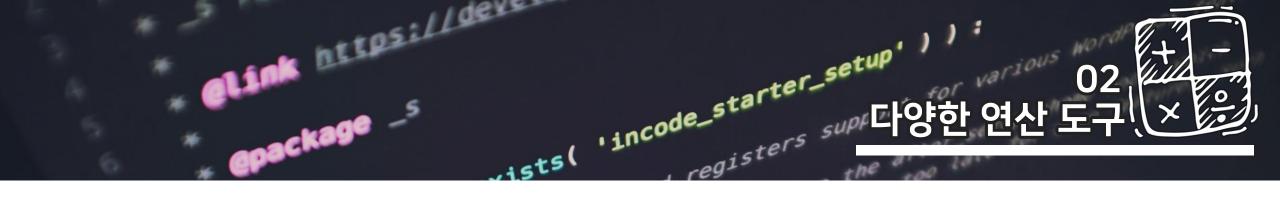
(b) nansum(Ar7)



mean : 평균을 구하는 명령어

std: 표준편차를 구하는 명령어

사용 방법은 sum과 같음



• 여러 가지 함수의 응용

[Q] 0부터 0.01 간격으로 1까지 정의역 x의 배열을 만들고, y = x<sup>2</sup>을 만족하는 배열 y를 만들어봅시다.

$$x = 0:0.01:1;$$
  
 $y = x.^2;$ 

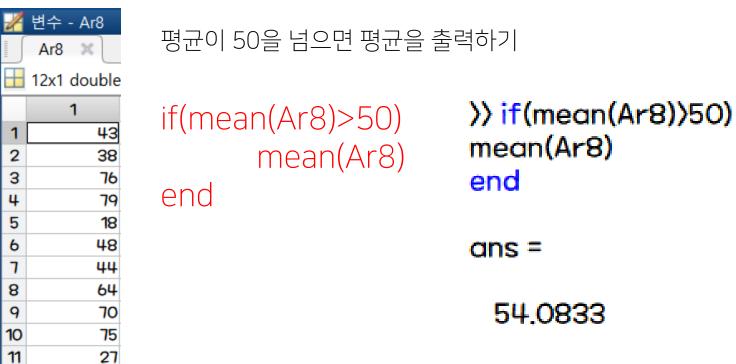
조건문: 특정 조건을 만족할 때만 명령이 실행되게 하는 문법

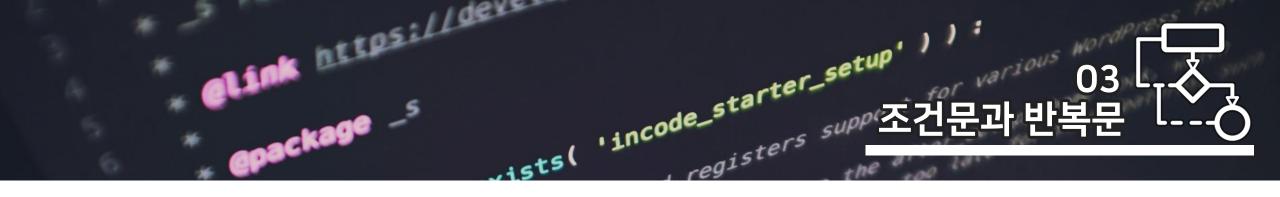
<기본 구조>

if(*true인지 확인할 조건*) *명령* end

# 예제 코드 MATLAB\_EX\_05.m 실행

67





비교 연산자 : 두 변수의 크기나 일치 여부 등을 비교하는 연산자

A가 B보다 크다: A > B

A가 B보다 크거나 같다 : A >= B

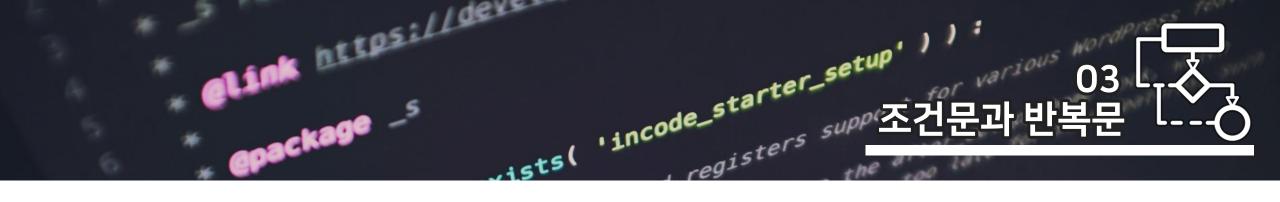
A가 B보다 작다 : A < B

A가 B보다 작거나 같다 : A <= B

A와 B가 같다 : A == B

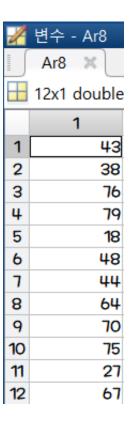
A와 B가 다르다 : A ~= B

이 외에도 데이터가 NaN인지를 확인하는 isnan, 논리값이 true인지를 확인하는 istrue 등의 비교연산자 함수도 있음

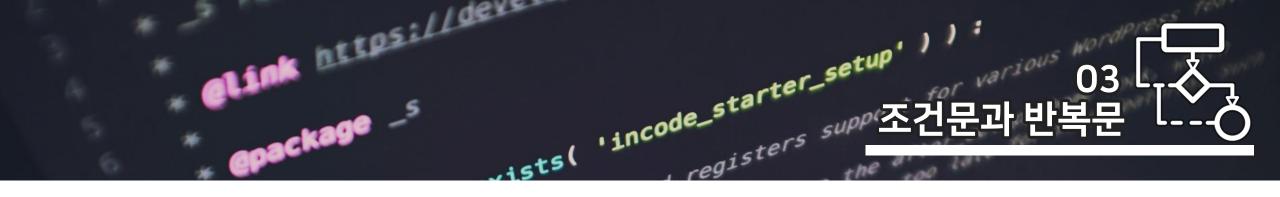


조건을 만족할 때와 만족하지 않을때의 명령을 다르게 주고 싶을 때는 else를 활용

if(*true인지 확인할 조건*) *명령 1* else *명령 2* end



평균이 60을 넘으면 평균을, 넘지 못하면 표준편차를 출력하기



여러 조건으로 분류를 하고 싶을 때는 elseif를 활용

```
if(true인지 확인할 조건 1)
명령 1
elseif(true인지 확인할 조건 2)
명령 2
elseif(true인지 확인할 조건 3)
명령 3
```

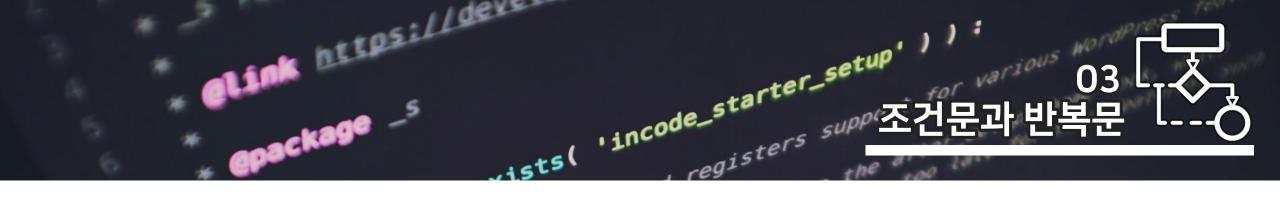
.

end

1	변수	- Ar8
	Ar8	×
	12x1	double
		1
1		43
2		38
3		76
4		79
5		18
6		48
7		44
8		64
9		70
10		75
11		27
12		67

평균이 60을 넘으면 평균을, 55를 넘으면 표준편차를, 그렇지 않으면 전체의 합계를 출력하기

```
>> if(mean(Ar8)>60)
if(mean(Ar8)>60)
                              mean(Ar8)
       mean(Ar8)
                        elseif(mean(Ar8))55)
elseif(mean(Ar8)>55)
                              std(Ar8)
                        else
       std(Ar8)
                              sum(Ar8)
else
                        end
       sum(Ar8)
                        ans =
end
                         649
```



한 번에 여러 조건을 만족하거나, 여러 조건 중 하나만 만족하면 명령이 실행되고 싶게 할 때는 논리 연산자를 활용

여러 조건을 모두 만족해야 할 때(and의 역할): &&

여러 조건 중 하나만 만족하면 될 때(or의 역할): ||

Shift + ₩ 두 번

# 두 조건을 모두 만족해야 명령이 실행되게 하기

if(*true인지 확인할 조건 1 && true인지 확인할 조건 2)* 명령

end

# 두 조건 중 하나만 만족하면 명령이 실행되게 하기

if(*true인지 확인할 조건 1 || true인지 확인할 조건 2)* 명령

end