

MATLAB 프로그래밍 및 실습

1강. MATLAB 소개

참고한 곳들

<https://www.mathpark.com/524>

<https://bit.ly/2YoK9Qs>

<https://namu.wiki/w/%EB%A1%9C%EA%B0%80%EB%A6%AC%EB%93%AC>

<https://suhak.tistory.com/250>

<https://bit.ly/34mL9bP>

<https://www.exploringbinary.com/why-0-point-1-does-not-exist-in-floating-point/>

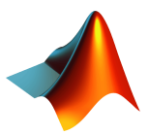
<https://www.slideshare.net/insideHPC/evolution-of-matlab>

<https://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html>

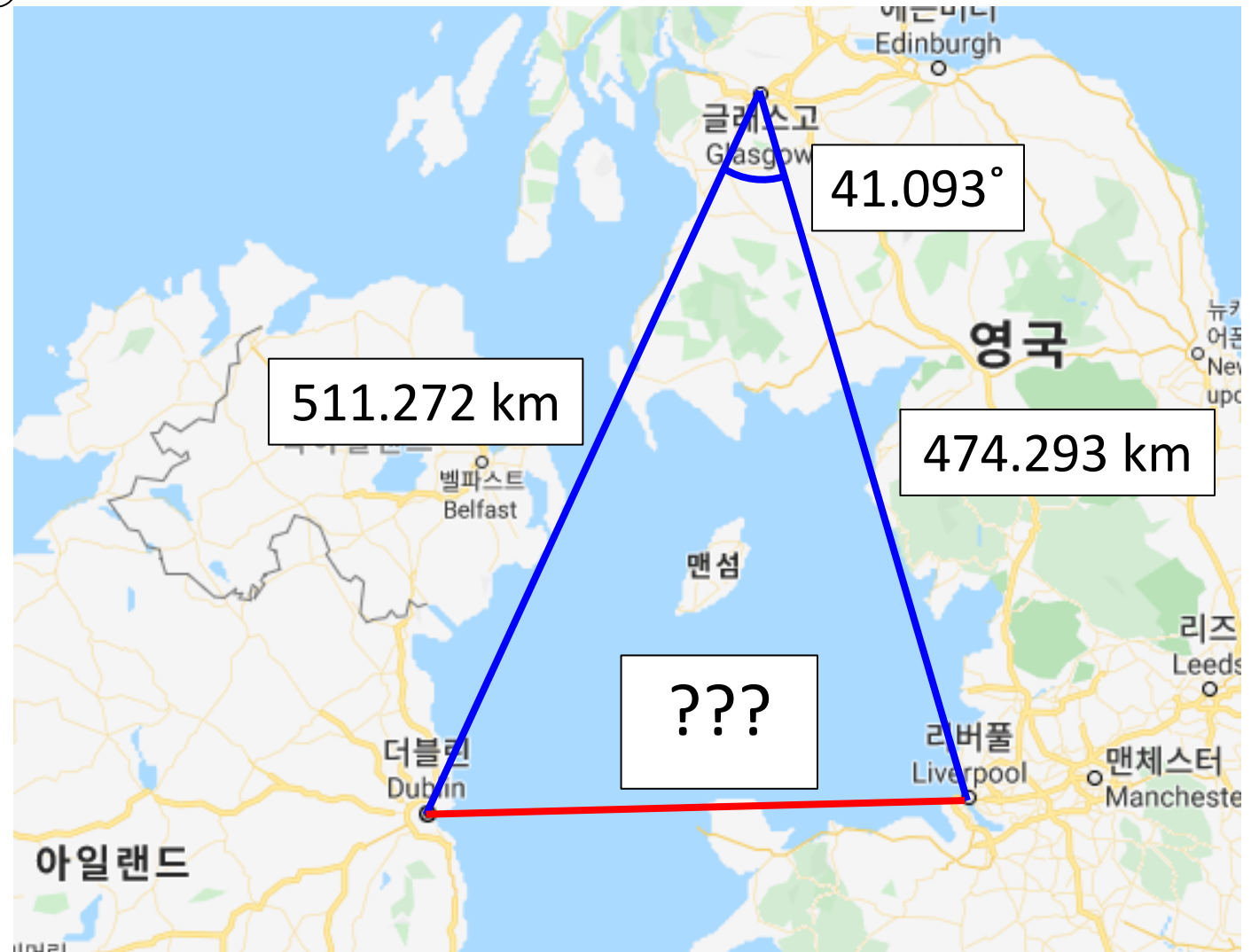
<http://www.netlib.org/utk/people/JackDongarra/SLIDES/sc2011-tutorial.pdf>

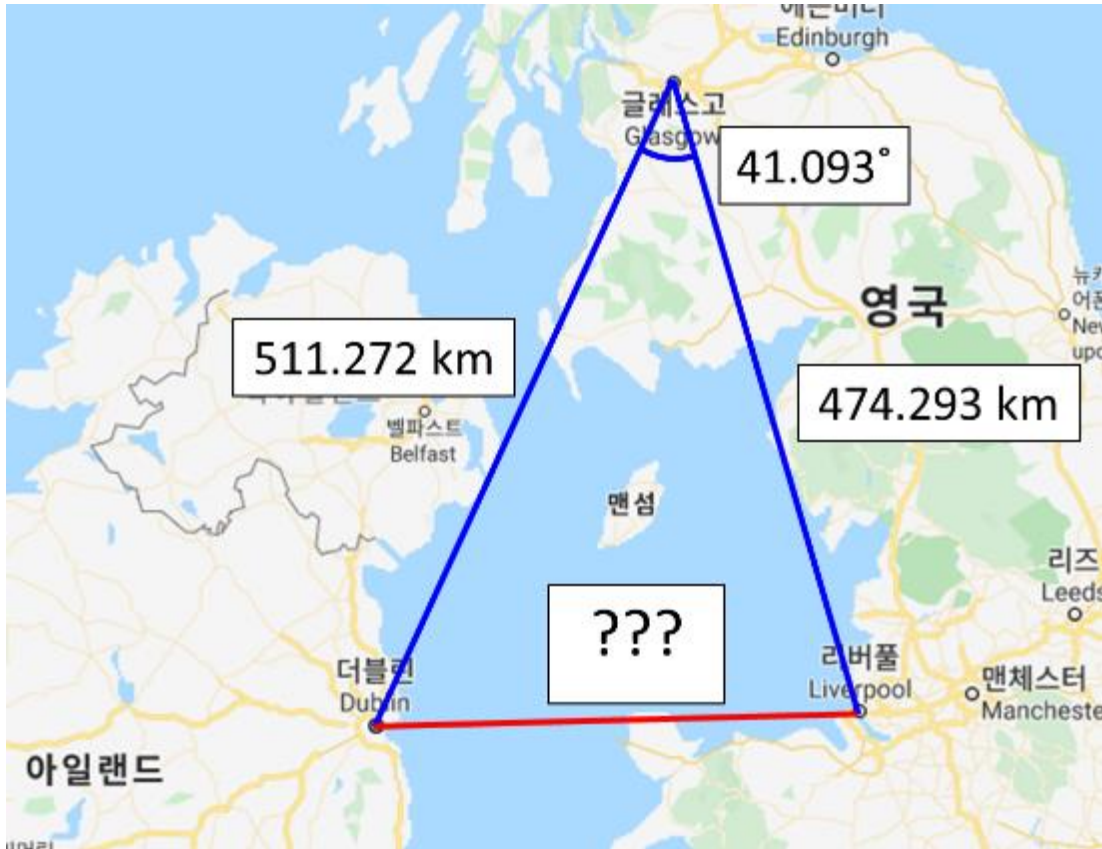
<https://cacm.acm.org/blogs/blog-cacm/166115-why-scientists-and-engineers-must-learn-programming/fulltext>

<https://kr.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm>

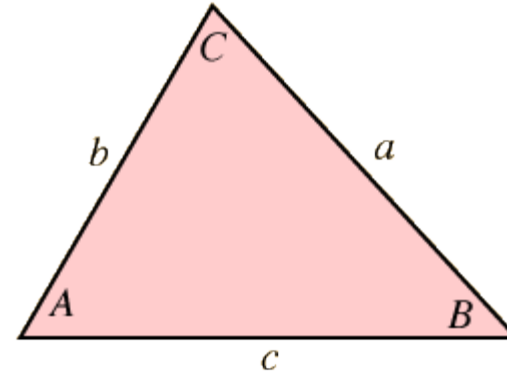


로그_만든_셀_폰다.jpg





$$c^2 = a^2 + b^2 - 2ab \cos C$$



$$\begin{array}{r} 511.272 \\ \times 511.272 \\ \hline 1022.544 \\ 35789.94 \\ 102254.4 \\ 511272 \\ 511272 \\ 2558356 \\ \hline 261598058.884 \end{array}$$

$$c = \sqrt{511.272^2 + 474.293^2 - 2 \times 511.272 \times 474.293 \times \cos(41.093^\circ)}$$

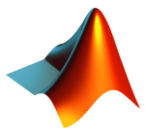




그림 1

P의 속도는 PB 거리에 비례 (점점 느려짐)
P의 초기속도는 10^7

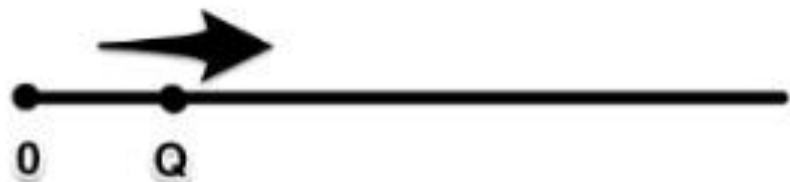
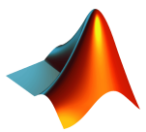


그림 2

Q는 등속도 운동 (속도 = 10^7)

$$OQ = NapLog (BP)$$

$$NapLog \left(\frac{N_1 \times N_2}{10^7} \right) = NapLog(N_1) + NapLog(N_2)$$



$$\log(AB) = \log(A) + \log(B)$$

<https://bit.ly/2YoK9Qs>

<https://namu.wiki/w/%EB%A1%9C%EA%B0%80%EB%A6%AC%EB%93%AC>

LOGARITHMS, BASE 10

$\log_{10}x$ or $\lg x$

x	0	1	2		1	2	3	4	5	6	7	8	9
10	.0000	0043	0086										
11	.0414	0453	0492										
12	.0792	0828	0864										
13	.1139	1173	1206										
14	.1461	1492	1523										
20	.3010	3032	3054										
21	.3222	3243	3263										
22	.3424	3444	3464										
23	.3617	3636	3655	3674	3692	3711	3729	3747	3766	3784			
24	.3802	3820	3838	3856	3874	3892	3909	3927	3945	3962			
25	.3979	3997	4014	4031	4048	4065	4082	4099	4116	4133			
26	.4150	4166	4183	4200	4216	4232	4249	4265	4281	4298			
27	.4314	4330	4346	4362	4378	4393	4409	4425	4440	4456			
28	.4472	4487	4502	4518	4533	4548	4564	4579	4594	4609			
29	.4624	4639	4654	4669	4683	4698	4713	4728	4742	4757			
49	6902	6911	6920	6928	6937	6946	6955	6964	6972	6981			
50	6990	6998	7007	7016	7024	7033	7042	7050	7059	7067			
51	7076	7084	7093	7101	7110	7118	7126	7135	7143	7152			
52	7160	7168	7177	7185	7193	7202	7210	7218	7226	7235			
53	7243	7251	7259	7267	7275	7284	7292	7300	7308	7316			

Handwritten calculation showing a large red 'X' over a multiplication problem: 511.272×511.272 . The result 261598658.884 is written below the 'X'.

$$511.272^2 = ?$$

$$\log_{10} 511.272 \cong 2.7087$$

$$\log_{10} 511.272^2 \cong 2.7087 \times 2 = 5.4174$$

$$10^{5.4172} \cong 261500$$

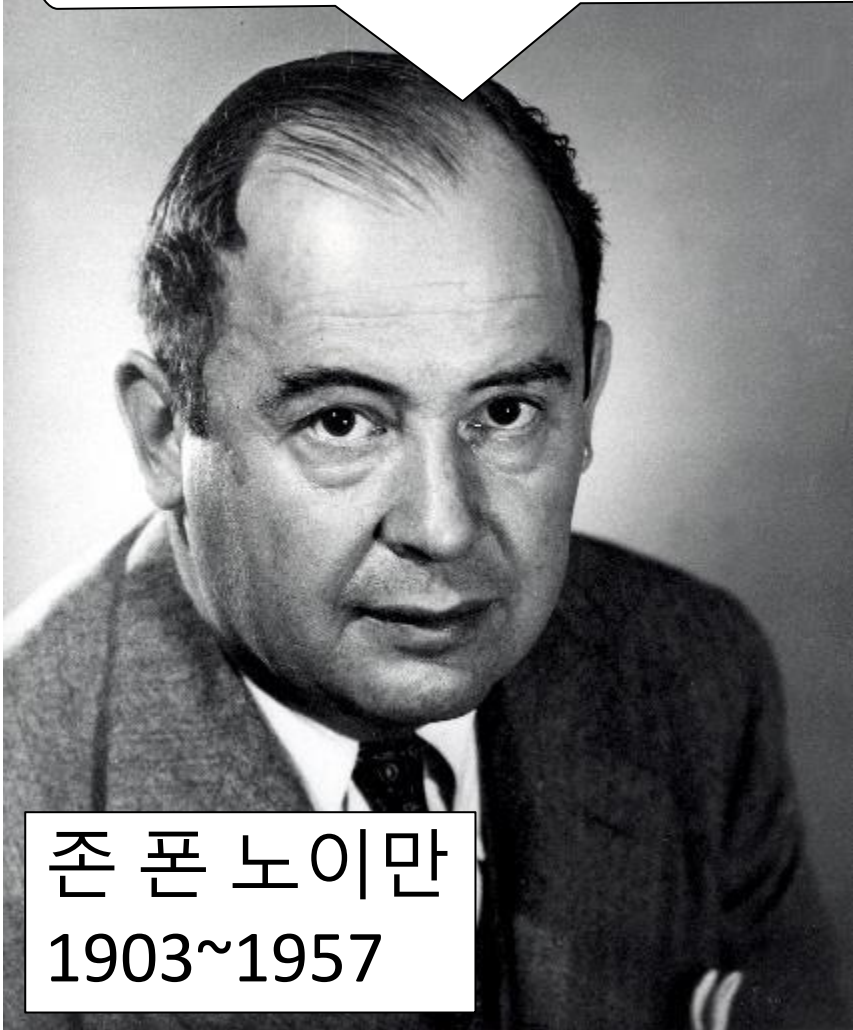


실제 계산결과:

$$511.272^2 = 261399.057984$$

“로그의 발명으로 천문학자들의 수명이 두 배가 되었다.”

“(너네 몸값보다) 비싼 장비를 가지고 그런 잔머리나
굴리려 하다니 한심하기 그지없다.”



존 폰 노이만
1903~1957

```
0010000000010000
1001000000000000
0001000000010000
0100000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
```

Assembly Language

```
mov ecx, ebx
mov esp, edx
mov edx, r9d
mov rax, rdx
```

Programmer

Assembler + Linker

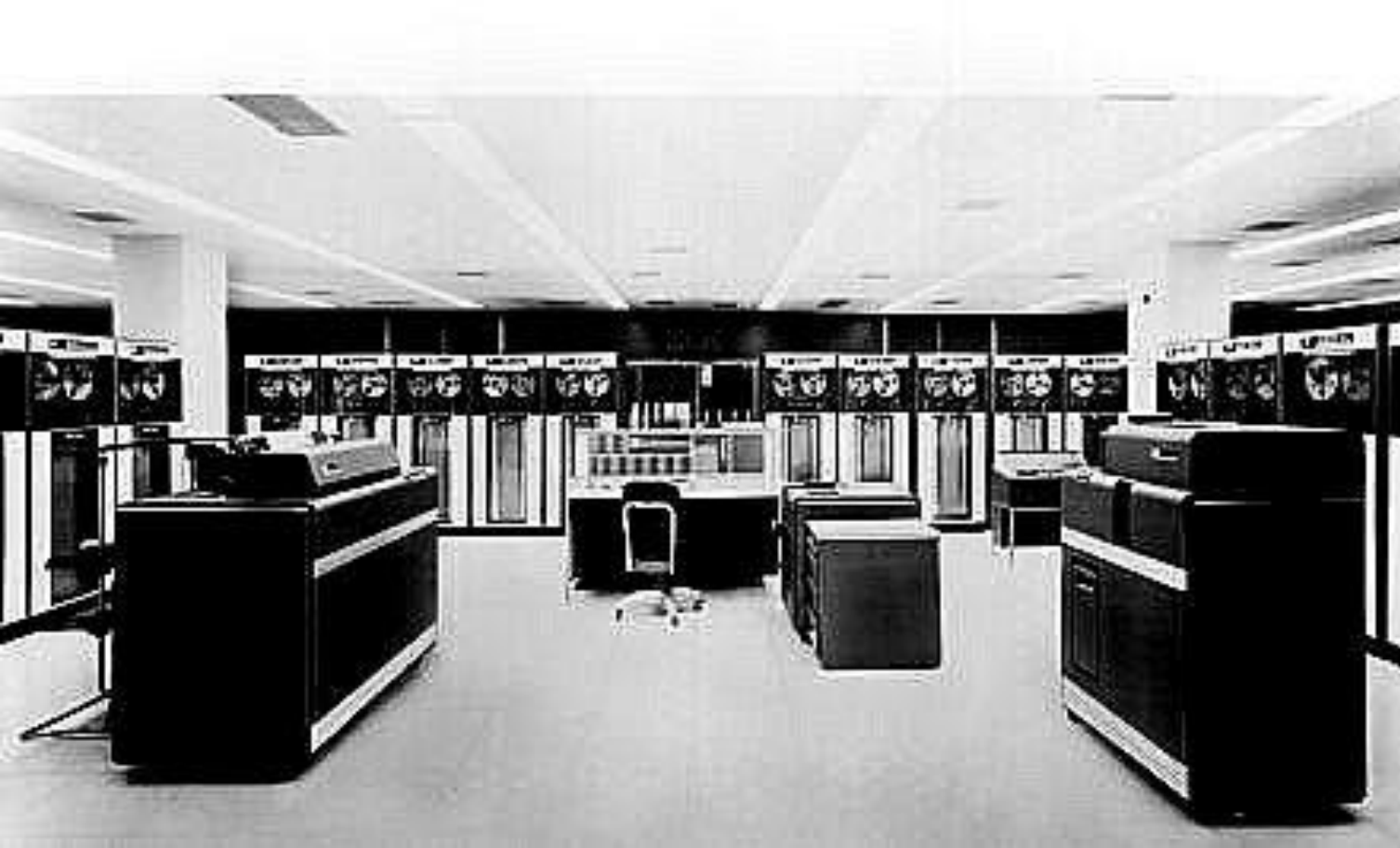
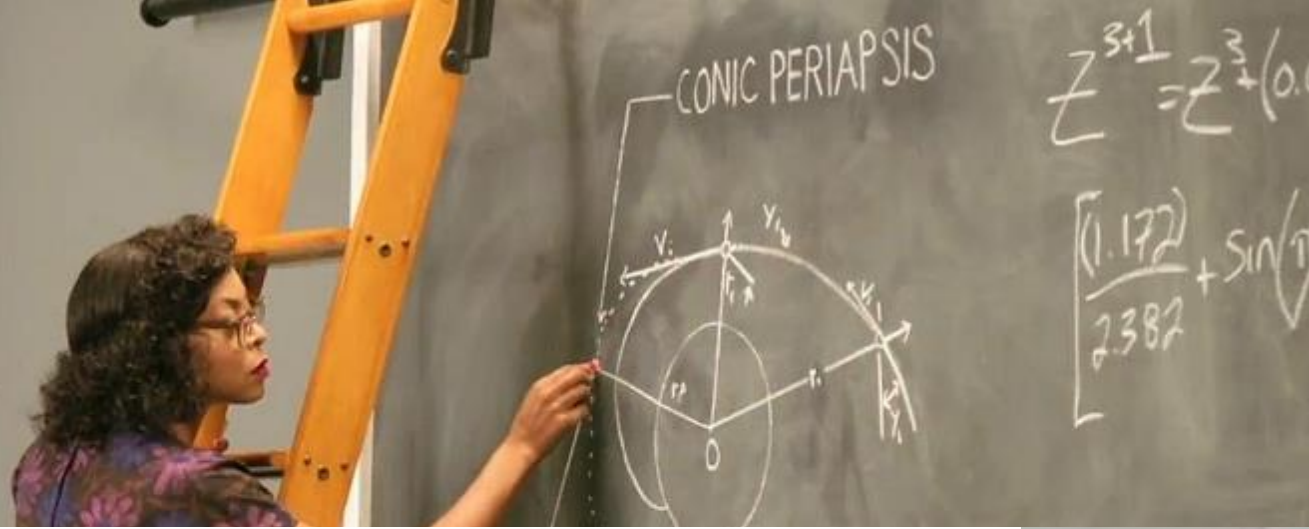
Machine Language

```
100101011001
010011111011
111010101101
01010101010
```

Processor

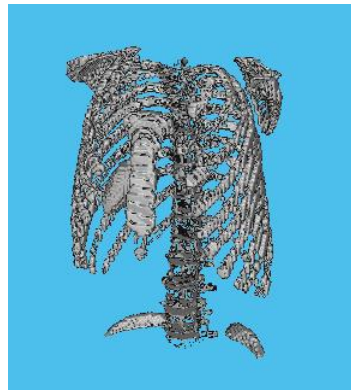


Computer Punch Cards

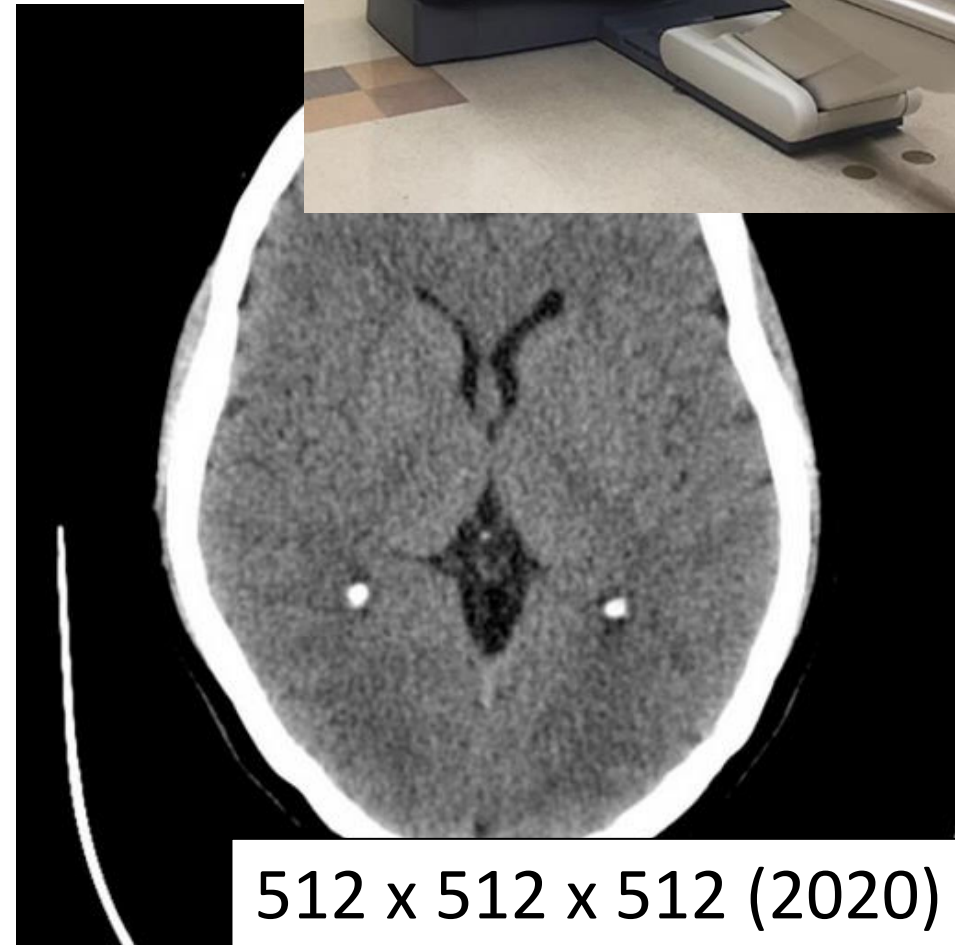




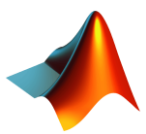
고드프리 하운스필드
(1919~2004)



128 x 128 ('71)



512 x 512 x 512 (2020)



< M A T L A B >
Version of 01/10/81

HELP is available

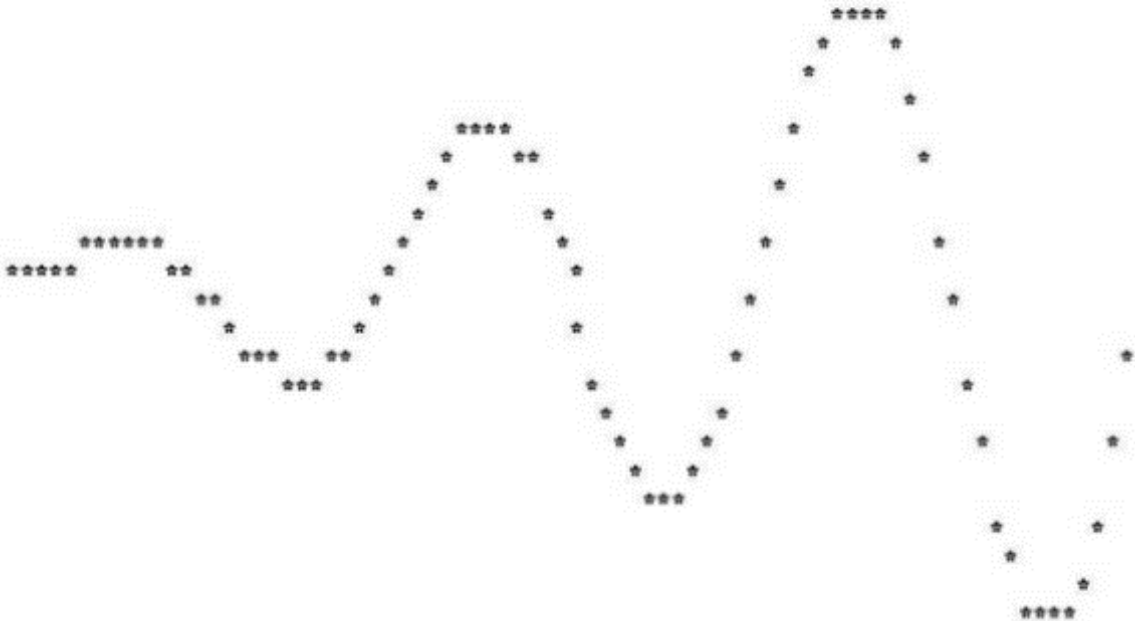
<>help

Type HELP followed by
INTRO (to get started)
NEWS (recent revisions)

ABS	ANS	ATAN	BASE	CHAR	CHOL	CHOP	CLEA	COND	CONJ	COS
DET	DIAG	DIAR	DISP	EDIT	EIG	ELSE	END	EPS	EXEC	EXIT
EXP	EYE	FILE	FLOP	FLPS	FOR	FUN	HESS	HILB	IF	IMAG
INV	KRON	LINE	LOAD	LOG	LONG	LU	MACR	MAGI	NORM	ONES
ORTH	PINV	PLOT	POLY	PRIN	PROD	QR	RAND	RANK	RCON	RAT
REAL	RETU	RREF	ROOT	ROUN	SAVE	SCHU	SHOR	SEMI	SIN	SIZE
SQRT	STOP	SUM	SVD	TRIL	TRIU	USER	WHAT	WHIL	WHO	WHY

< > () = . , ; \ / ' + - * :

<>plot(x,x.*sin(3*x))



FORTRAN



EISPACK, LINPACK



MATLAB
(MATrix LABoratory)

```
<> n = 10;  
<> J = 1:n;  
<> J = J(ones(n,1),:);  
<> I = J';  
<> E = ones(n,n);  
<> A = E./(I-J+E/2);  
<> long  
<> S = svd(A)
```

S =

```
3.141592653589681  
3.141592653566663  
3.141592651393167  
3.141592527498736  
3.141587781570560  
3.141459305862260  
3.138952480600910  
3.104107683136387  
2.786915482404130  
1.300969070029703
```

<>



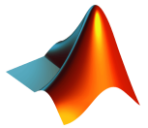
왜 공학에서 프로그래밍을 배워야 할까?

- 로그
- 어셈블리어
- 맷랩

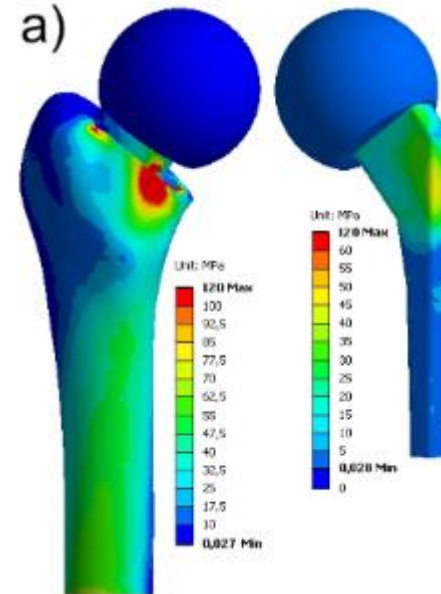
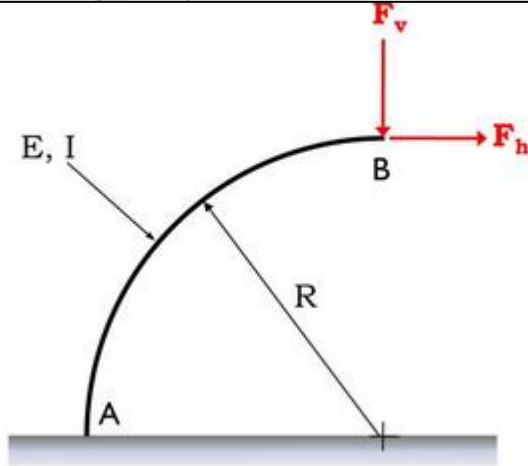
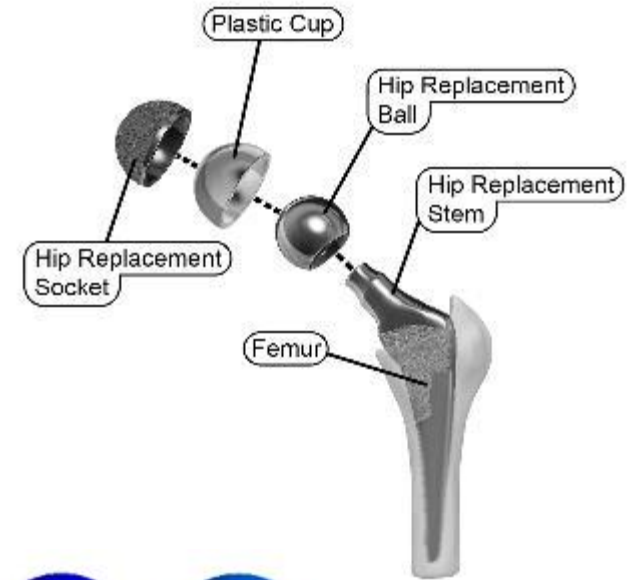
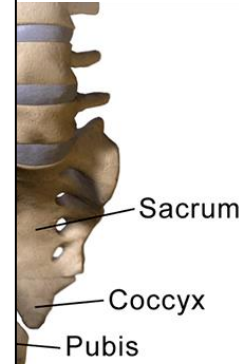
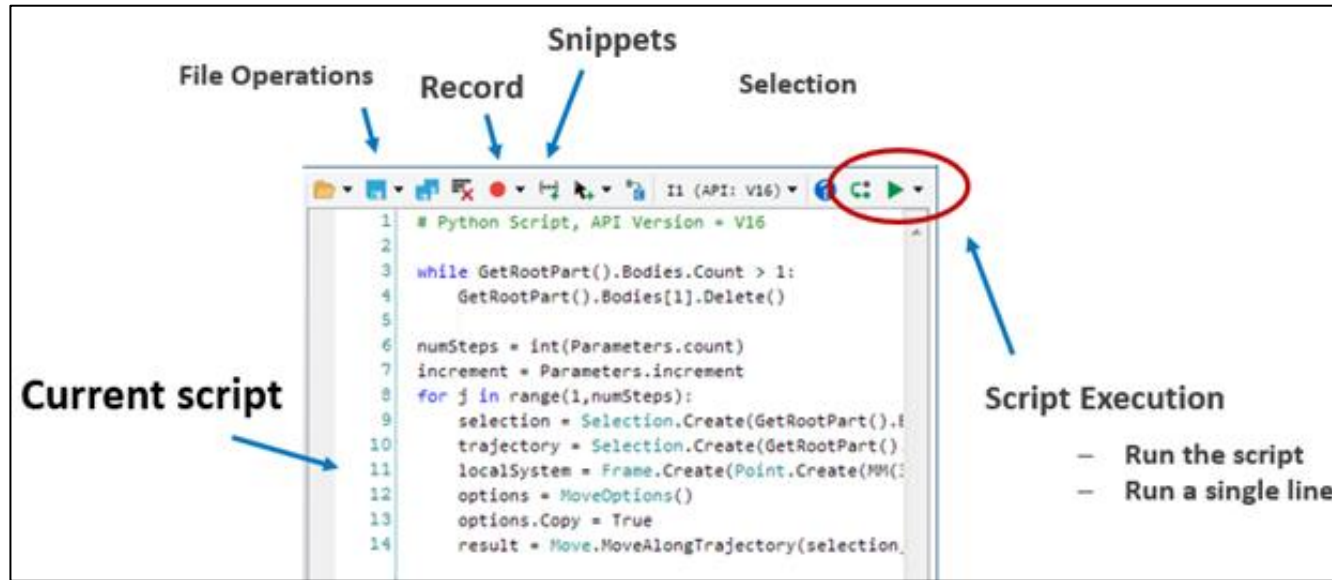


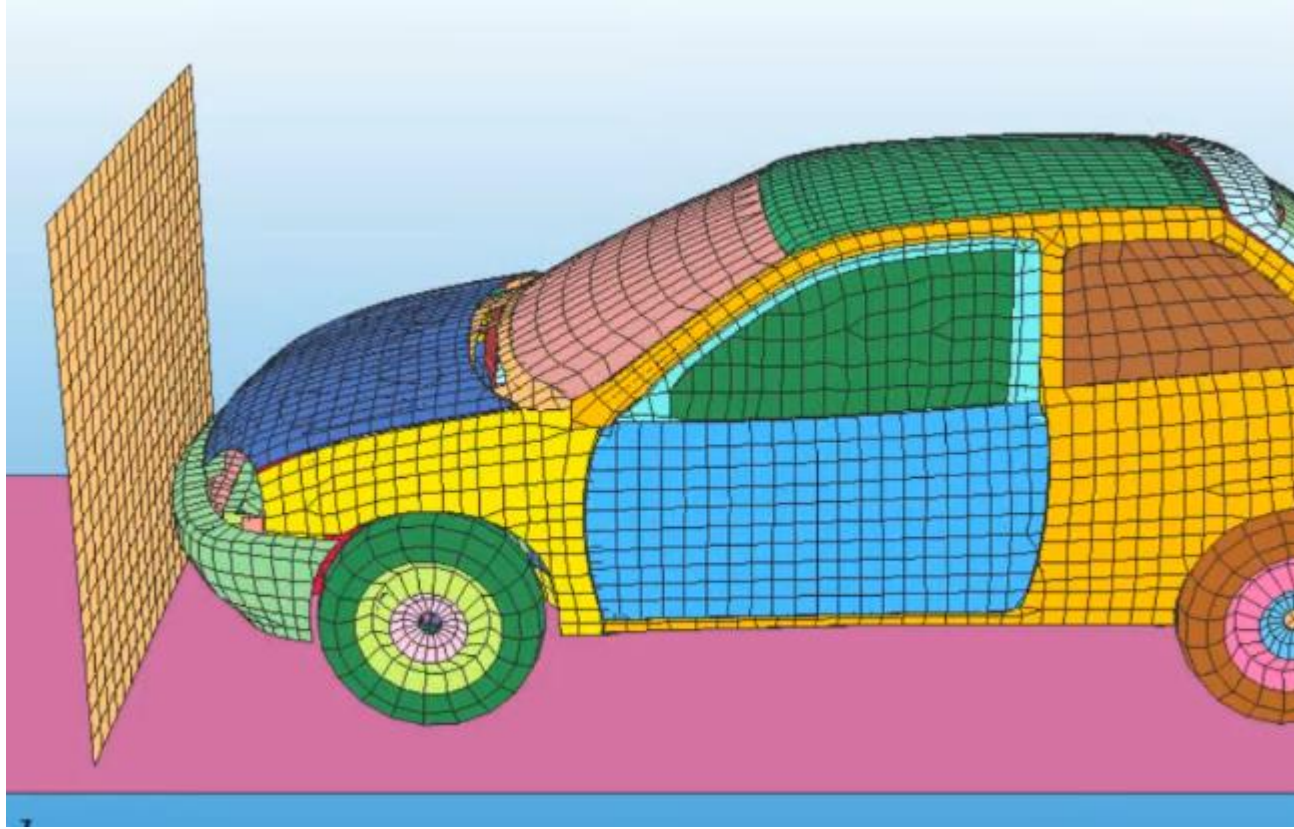
계산기 개발의 역사

- 같은 작업을 훨씬 빠르게
- 같은 데이터도 여러 시각으로
- 전문 프로그래머와 소통 가능

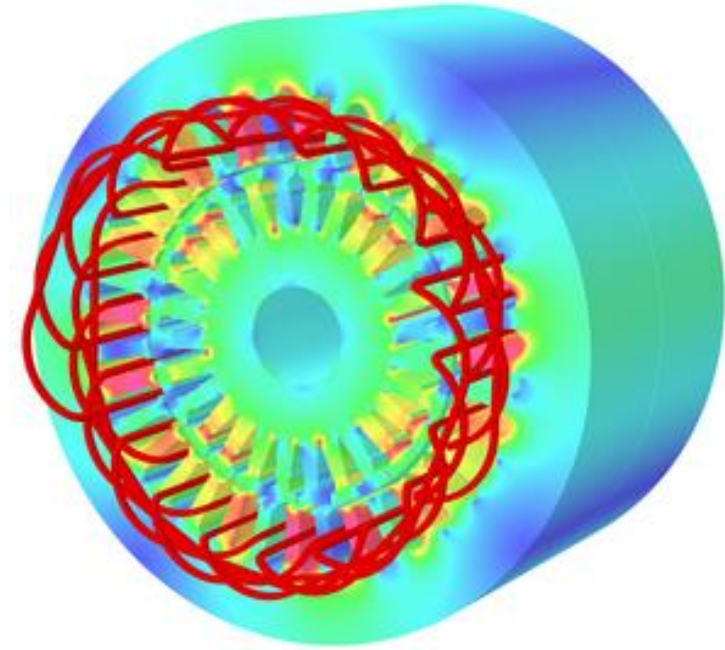


왜 공학에서 프로그래밍을 배워야 하는가?





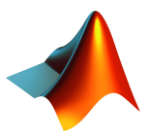
<https://www.youtube.com/watch?v=a7LzGsemQ4E>

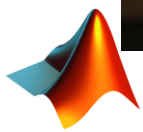


```

/                               Open input file for reading
$ open 1 points.dat read
/                               Read the first line into user variable #np
$ read 1 #np
/                               Open output file for writing
$ open 2 fields.dat write
/                               Define a string format to space the output
$ form 1 string string=' '
/                               Define a floating point format
$ form 2 expo 0
/                               Assign format numbers for the output
$ assi 2 1 2 1 2 1 2
/                               Start a loop from 1 to #np
$ do #i 1 #np
/                               Read #x and #y from input file
$ read 1 #x #y
/                               Evaluate fields at #x #y
poin x1=#x y1=#y
/                               Write coordinates and flux density to output file

```







(a)



(b)



(d)



그럼 왜 매트랩인가?



아파트

리눅스

C



오피스텔

윈도우

Python

- 고급 언어로 갈수록...
 - 알아서 해주는 게 많음
 - 이미 갖춰진 것이 많음
 - 프로그래밍을 처음 시작하기에 좋은 언어 (한 언어를 깊게)



고오급호텔

OSX

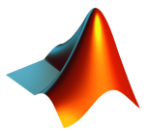
MATLAB

Tiobe index

- 언어를 사용하는 엔지니어 수
- 언어 관련 강좌 및 홈페이지 수
- 검색엔진에서의 검색 횟수

MATLAB: 20위권 내 꾸준히 유지 중

Aug 2020	Aug 2019	Change	Programming Language	Ratings
1	2	⬆	C	16.98%
2	1	⬇	Java	14.43%
3	3		Python	9.69%
4	4		C++	6.84%
5	5		C#	4.68%
6	6		Visual Basic	4.66%
7	7		JavaScript	2.87%
8	20	⬆	R	2.79%
9	8	⬇	PHP	2.24%
10	10		SQL	1.46%
11	17	⬆	Go	1.43%
12	18	⬆	Swift	1.42%
13	19	⬆	Perl	1.11%
14	15	⬆	Assembly language	1.04%
15	11	⬇	Ruby	1.03%
16	12	⬇	MATLAB	0.86%
17	16	⬇	Classic Visual Basic	0.82%
18	13	⬇	Groovy	0.77%
19	9	⬇	Objective-C	0.76%
20	28	⬆	Rust	0.74%



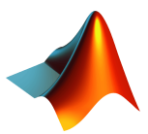
매트랩의 장점

- 쉬운 문법 및 디버깅, 빠른 개발속도 → 과학, 공학용 계산에 특화된 언어
- 행렬연산, 시각화에 강점 (매트랩에서는 모든 것이 행렬) 예제: boxBlur
- 방대한 양의 툴박스
- 프로그래밍 언어 + 애플리케이션
- 중앙집권형 언어
 - Mathworks에서 모든 것을 관리, 성능 유지가 잘 됨 (↔오픈소스)
 - Mathworks에서 자체 제작하는 고퀄의 도움말과 예제 + 한글화
 - 단일 IDE

드론에 카메라 장착 하여 시각 비행알고리즘 테스트
→ 모터제어, 통신, 영상처리, 위치/자세 제어
→ 모든 라이브러리를 MATLAB에서 자체 제공

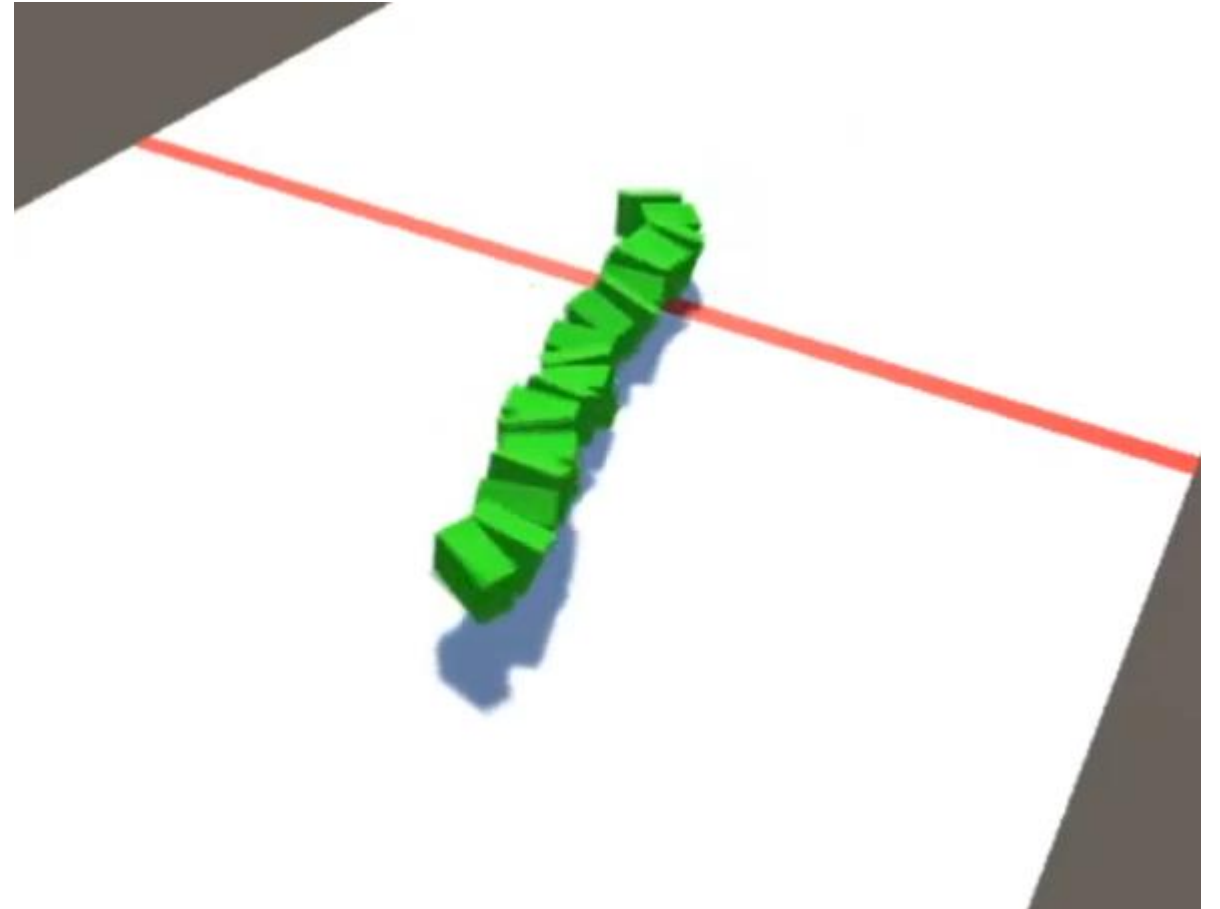
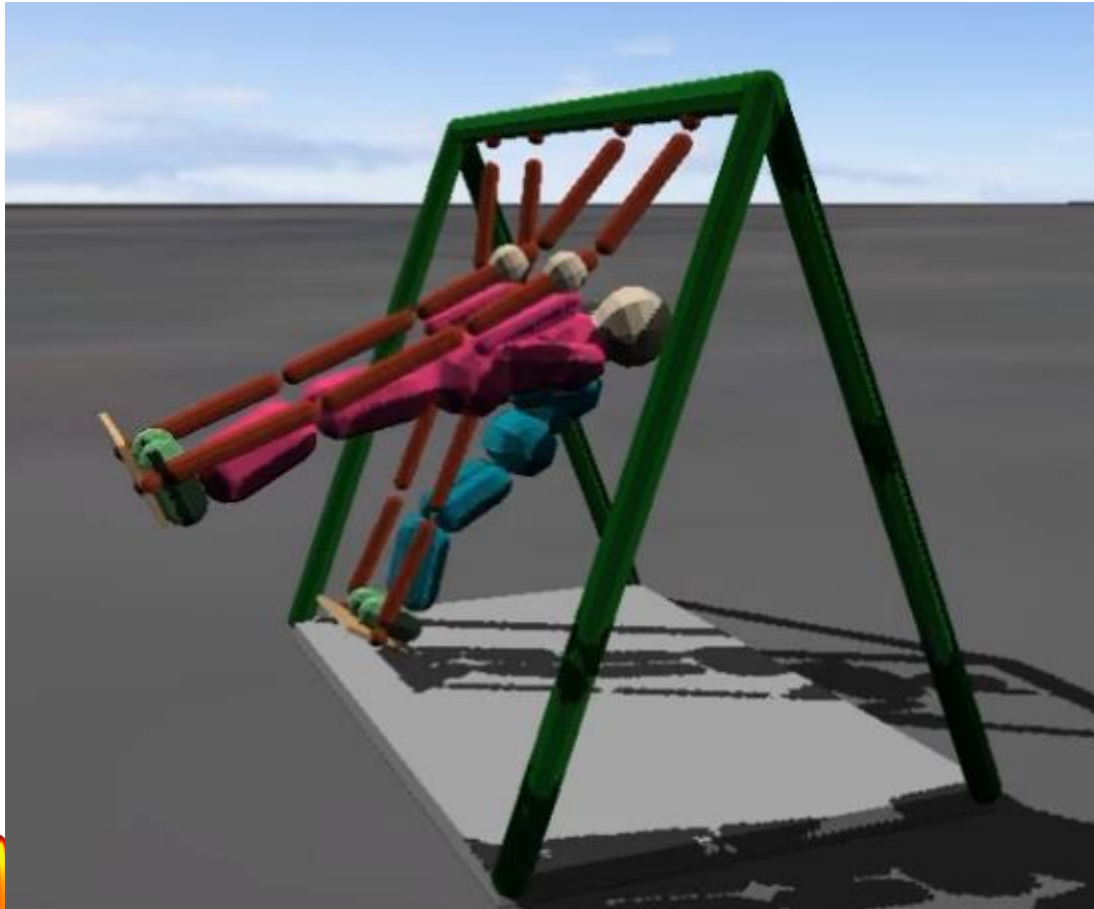
단점

- 비쌈 → 학교 및 연구소에서 주로 사용
- 느림 (스크립트 언어) → 자체 최적화 및 JIT compiler로 커버
- 노근본 언어? → 최신 트렌드를 꾸준히 따라가는 중

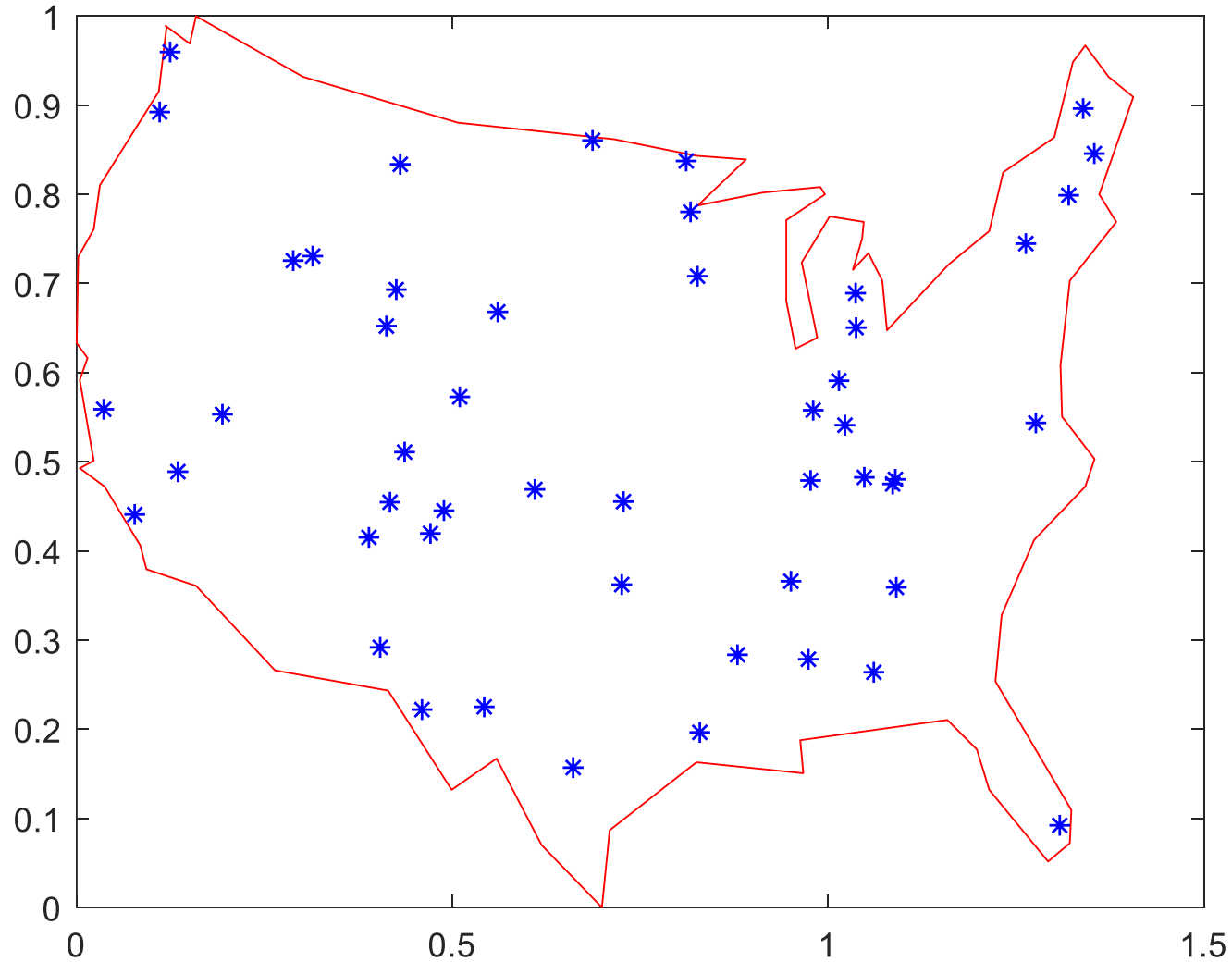


매트랩 활용 사례 - 머신 러닝

- 최적화된 그네 타기 https://www.youtube.com/watch?v=Yr_nRnqeDp0
- 머신 러닝 지령이 <https://youtu.be/mkvVeFWKZ7k>



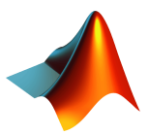
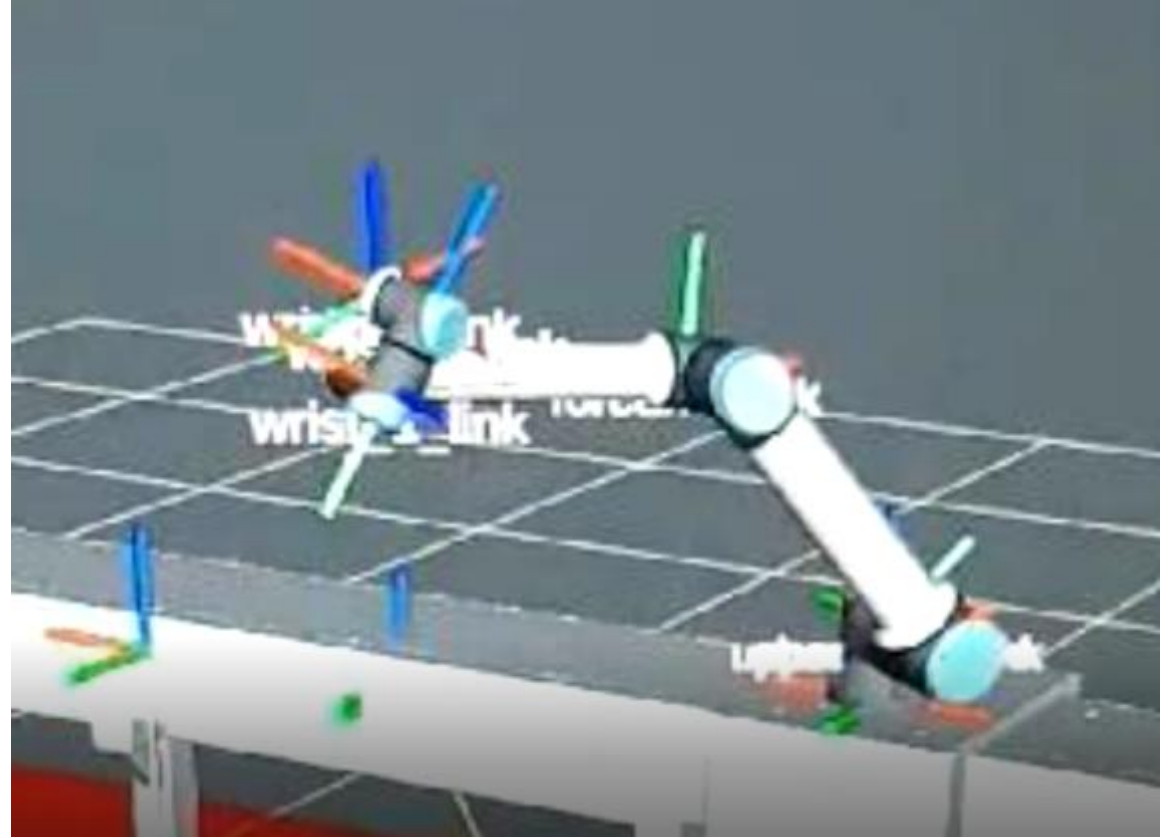
매트랩 활용 사례 – TSP (GA)



Source code:

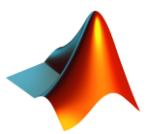
<https://kr.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm>

매트랩 활용 사례 – Robotics



매트랩 개발 환경 소개 / 프로그래밍 기본 지식

- 각 창의 의미, 용도, 사용 예
- 변수란? “a = 1”을 실행했을 때 컴퓨터 내부에서 일어나는 일은?
- 명령 창 (command window) vs 편집기 (editor)
- 매트랩 = 프로그래밍 언어 + 애플리케이션
- 컴파일러 vs 인터프리터
- 자료형이란? 왜 필요한가?



툴스트립 (toolstrip)

현재 폴더 주소

현재 폴더
(current folder)

편집기
(editor)

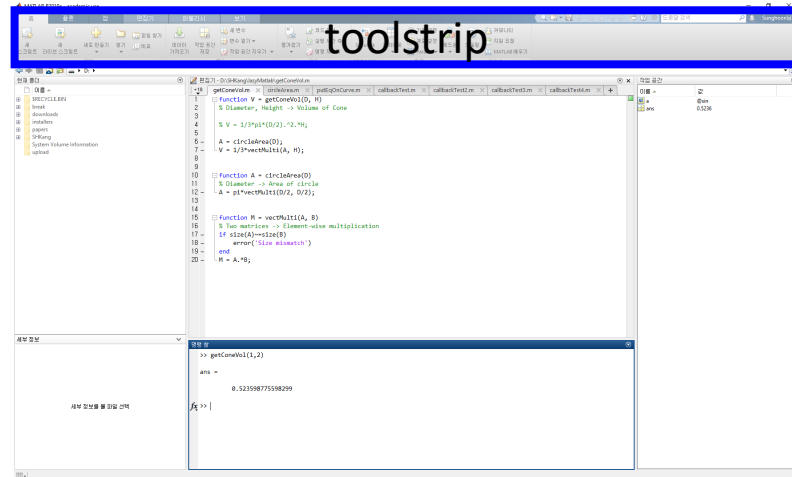
작업공간
(workspace)

tip. 각 창의 위치는 드래그로 바꿀 수 있음
실수로 창을 닫았다면? → 툴스트립 - 레이아웃 - 디폴트

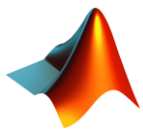
명령 창
(command window)

툴스트립 (toolstrip)

- 각종 설정, 빠른 작업 버튼 등
- 처음 배울 때는 주로 홈 탭과 편집기 탭을 사용

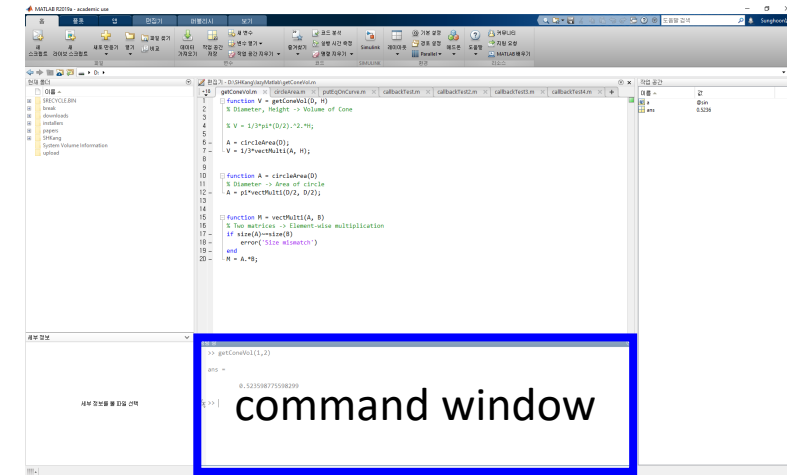


tip: 자주 쓰는 기능들은 이왕이면 단축키를 쓰자.



명령 창 (command window)

- 한 줄씩 명령을 보낼 수 있는 창 (=계산기)
- 프롬프트(>>)가 보이면 명령 입력 가능 상태
- 위아래 화살표로 명령 기록 확인 및 재사용 가능
- 실습



명령 창

```
>> disp('Hello, world!')
```

```
Hello, world!
```

```
>> 1+2+3
```

```
ans =
```

```
6
```

```
>> [pi, exp(1), 0, 10, 100]
```

```
ans =
```

```
3.1416
```

```
2.7183
```

```
0
```

```
10
```

```
100
```

tip:

d를 쓰고 위 화살표를 누르면
→ d로 시작하는 명령 기록 확인 가능

세미콜론(;)의 역할

- 세미콜론이 뒤에 붙으면
연산 결과를 화면에 출력하지 않음

현재 폴더 (current folder)

- 경로지정 없이 파일에 바로 접근할 수 있는 폴더

- 실습

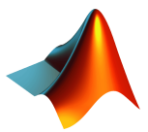
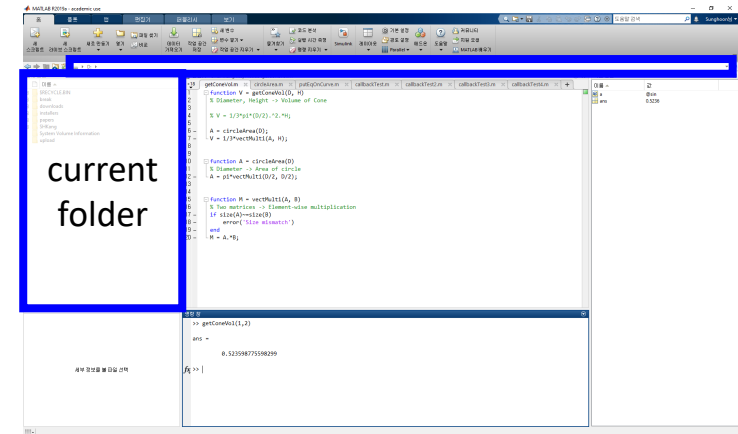
- 강의에 사용할 폴더를 하나 만들고, 현재 폴더를 해당 폴더로 변경
- 해당 폴더에 아무 이미지나 하나 추가 (ex. cake.jpg)
- 명령 창에서 cd와 dir 입력하여 현재 폴더 확인
- 명령 창에 다음과 같이 입력

```
명령 창
>> imshow(imread('cake.jpg'))
fx >>
```

- 현재 폴더가 아닌 폴더의 파일 접근? → 절대경로 작성 필요

```
명령 창
>> imshow(imread('D:\downloads\cat.png'))
```

tip: 경로에 쓰는 구분자는 /와 \ 모두 가능



작업 공간 (workspace)

- 메모리 상에 올라와 있는 변수의 목록을 보여주는 공간
- 변수란?
 - 값을 저장하고, 저장된 값을 바꿀 수 있으며, 이름으로 값을 읽어올 수 있는 프로그래밍의 기본 요소

실습

- 명령 창에서 변수를 선언하고 작업공간 확인

work
space



a = 1 (O)
b = a (O)
b = a+1 (O)
b = c (X)

변수 이름 → a = 1 ← 미리 “정의”된 값, 표현식, 변수
(값이든 변수든 미리 정의된 것)

대입연산자 (수학의 등호가 아님!)

변수를 “정의”한다.

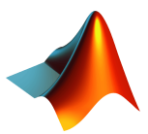
ex. a=a+1



CPU

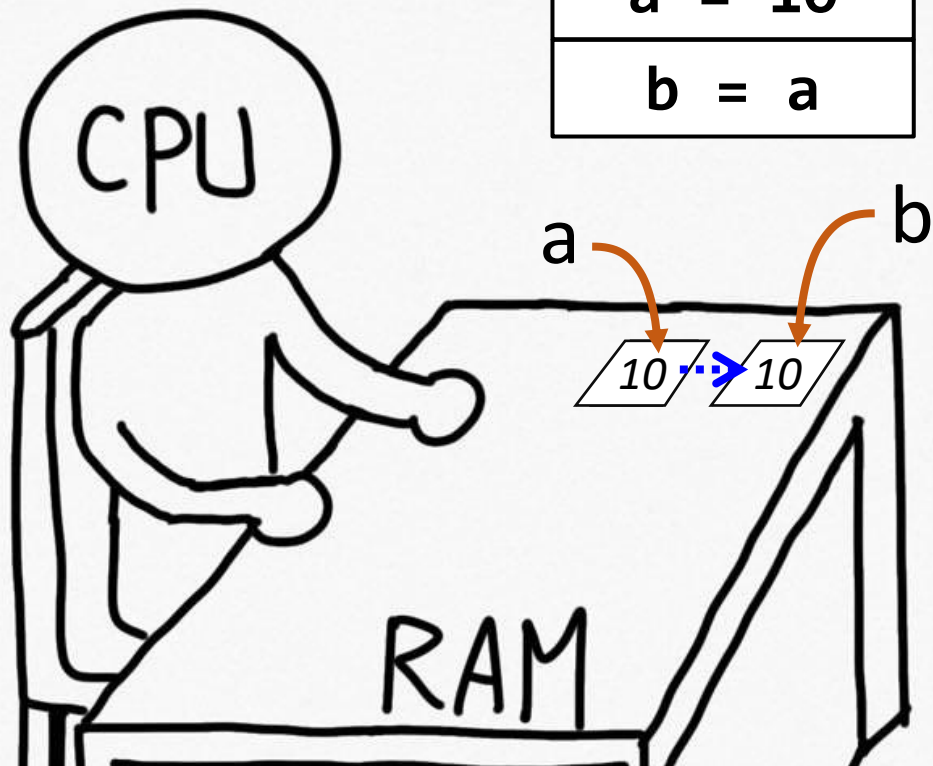
RAM

HDD/SSD



변수명
= 메모리에 붙은 이름

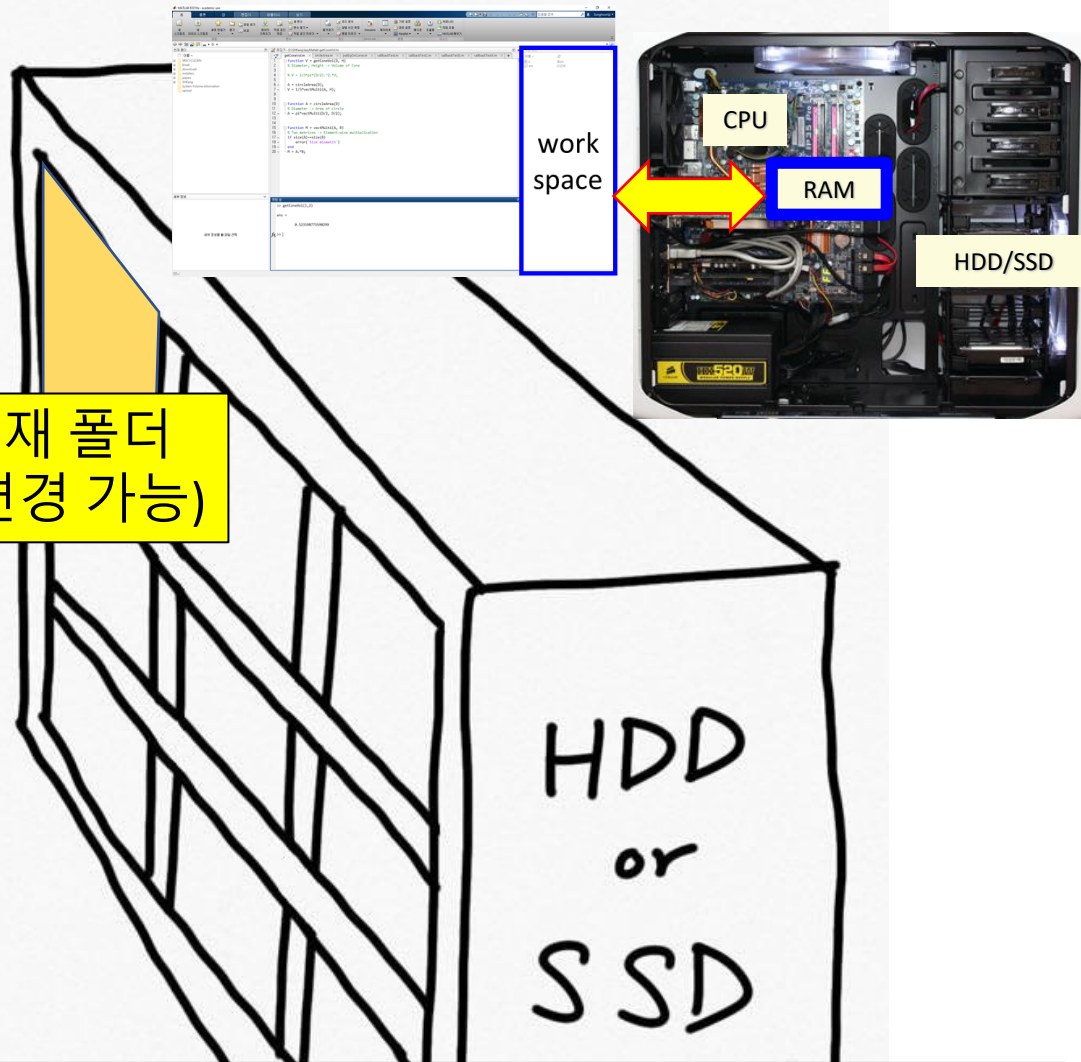
$a = 1$
$a = 10$
$b = a$



RAM (메모리)

- CPU가 현재 작업 중인 것들을 올려두는 공간
- CPU가 빠르게 접근 가능
- 컴퓨터의 **RAM** = 매트랩의 “**작업 공간**”
- CPU가 현재 작업 중인 것 = 매트랩의 “**변수**”
- 컴퓨터 재부팅 → RAM 초기화
- 매트랩 종료 → 작업공간 비움

현재 폴더
(변경 가능)



HDD/SSD (저장장치)

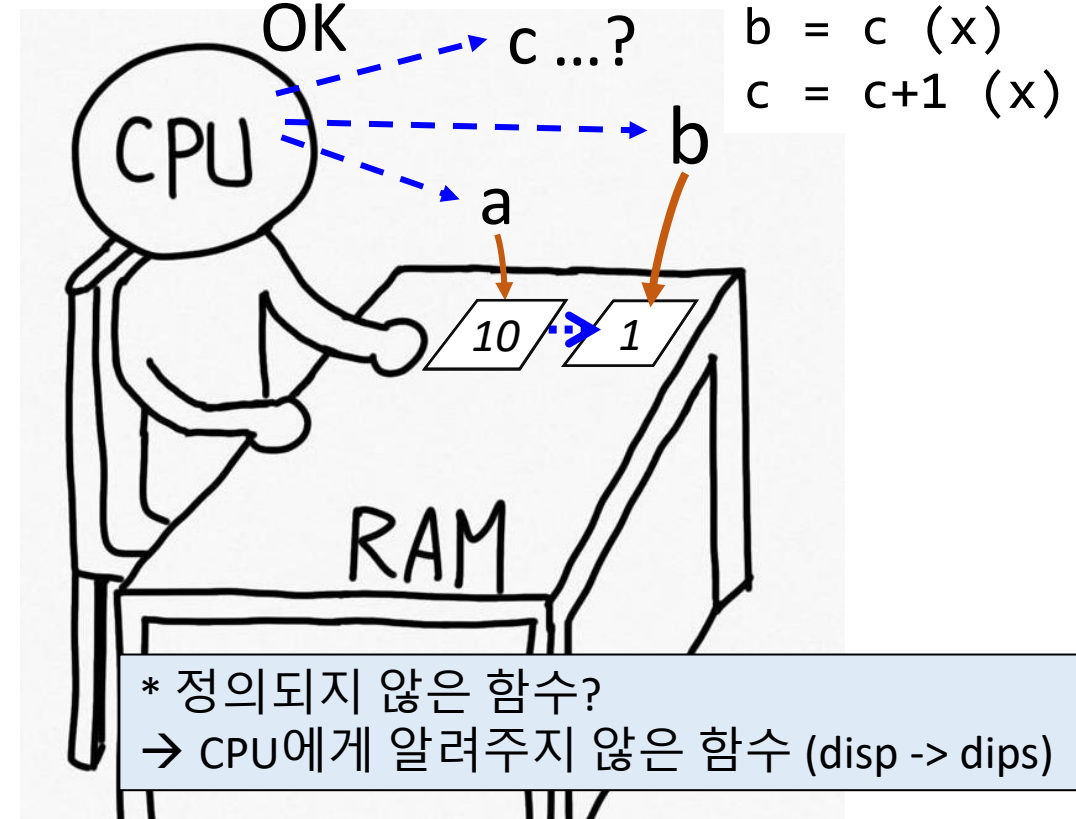
- 데이터의 장기간 저장을 위한 공간
- CPU가 접근하기 위해서는 다리가 필요
- RAM에 비해 느리지만 넓음
- 저장장치의 특정 폴더 = 매트랩의 “**현재 폴더**”

“정의되지 않은 함수 또는 변수입니다.”

- 모든 것은 정의에서부터 시작된다.
- 정의한 적이 있는 변수명 → 이름으로 접근 가능
- 정의한 적이 없는 변수명 → “정의되지 않은 변수입니다.”

<code>a = 1</code>
<code>b = a</code>
<code>a = 10</code>
<code>disp(a)</code>
<code>disp(b)</code>
<code>disp(c)</code>

??? → **ERROR**



※ 에러 메시지를 잘 읽자.

변수 - 실습

* 명령+엔터 → 연산 결과를 출력
: 대입연산도 “연산”이다.

* 명령 끝에 세미콜론(;) : 결과 출력 X

* 변수명은 대소문자 구분함
: a와 A는 다른 변수

```
>> x=15
```

The number 15 is assigned to the variable x.

```
x =  
    15
```

MATLAB displays the variable name and its assigned value.

```
>> x=3*x-12
```

A new value is assigned to x. The new value is 3 times the previous value of x minus 12.

```
x =  
    33
```

```
>>
```

```
>> a=12
```

Assign 12 to a.

```
a =  
    12
```

```
>> B=4
```

Assign 4 to B.

```
B =  
     4
```

```
>> C=(a-B)+40-a/B*10
```

Assign the value of the expression on the right-hand side to the variable C.

```
C =  
    18
```

```
>> a=12;
```

The variables a, B, and C are defined but are not displayed, since a semicolon is typed at the end of each statement.

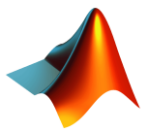
```
>> B=4;
```

```
>> C=(a-B)+40-a/B*10;
```

```
>> C
```

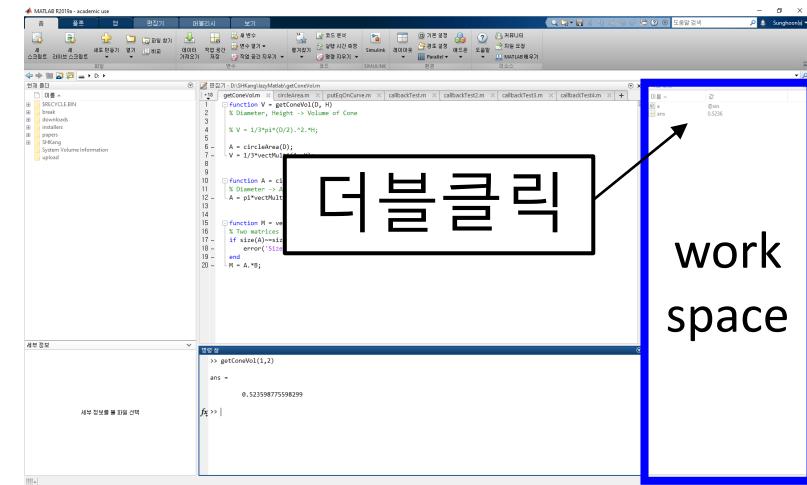
The value of the variable C is displayed by typing the name of the variable.

```
C =  
    18
```



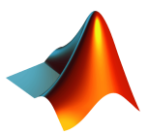
변수 창 (variable editor)

- 변수의 전체 내용을 확인할 수 있음
- 작업 공간의 변수를 더블클릭
- 실습



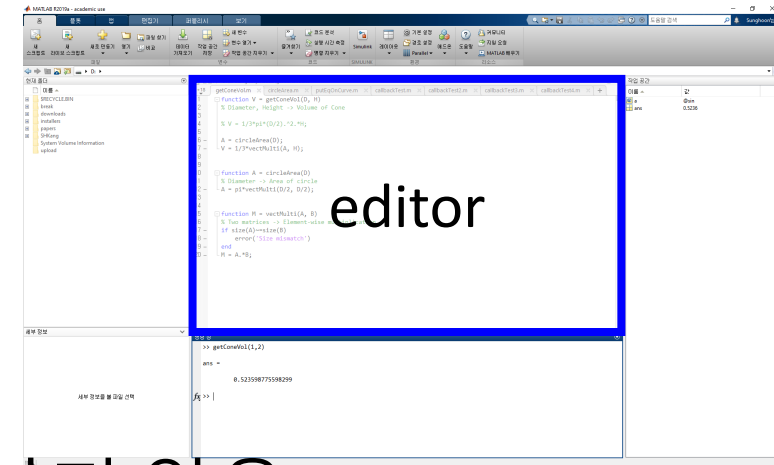
```
>> a = 1;  
>> b = rand(5);  
>> c = 1:2:10;  
fx >>
```

변수 - b						
a x b x c						
5x5 double						
	1	2	3	4	5	
1	0.5109	0.5926	0.8033	0.0074	0.0354	
2	0.1019	0.5486	0.5192	0.7500	0.3945	
3	0.2174	0.5588	0.7131	0.9130	0.7640	
4	0.6928	0.0535	0.0628	0.7414	0.6074	
5	0.3737	0.7154	0.0368	0.7855	0.1995	
6						



편집기 (editor)

- 일련의 명령을 작성하여 한번에 실행할 수 있음
 - 명령창은 매번 한줄 한줄 입력해주어야 함
- 편집기를 통해 실행되는 명령들은 명령 창에 나타나지 않음
- 실습
 - 아래와 같이 작성 → 파일 저장 (스크립트 파일 또는 .m 파일)
 - 현재 폴더 창에 파일이 생성된 것을 확인할 것
 - 실행방법: 편집기가 선택된 상태에서 F5 (또는 툴스트립의 “실행” 버튼)



```
+19 first_code.m x
1 - disp('Hello, world!')
2 - 1+2+3
3 - disp(1+2+3)
4 - disp([pi, exp(1), sin(pi/6), cosd(60)])
```

tip:
코드 일부만 선택하고 F9를 누르면
선택된 부분만 실행됨

도움말 (help)

매트랩의 장점: 고퀄의 도움말
→ 모르면 찾아보자.

- 간단한 버전: 명령 창에 아래와 같이 입력
 - 명령어 또는 함수의 기초적인 설명과 입출력을 보여줌

```
>> help disp
disp - Display value of variable

This MATLAB function displays the value of variable X without printing the
variable name.

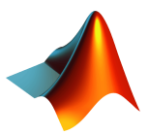
disp(X)

참고 항목 format, fprintf, int2str, num2str, sprintf

disp에 대한 함수 도움말 페이지
이름이 disp인 다른 함수
```

- Full 버전: 명령 창에 아래와 같이 입력
 - 명령어 또는 함수의 상세한 설명과 예제까지 보여줌

```
>> doc disp
>>
```

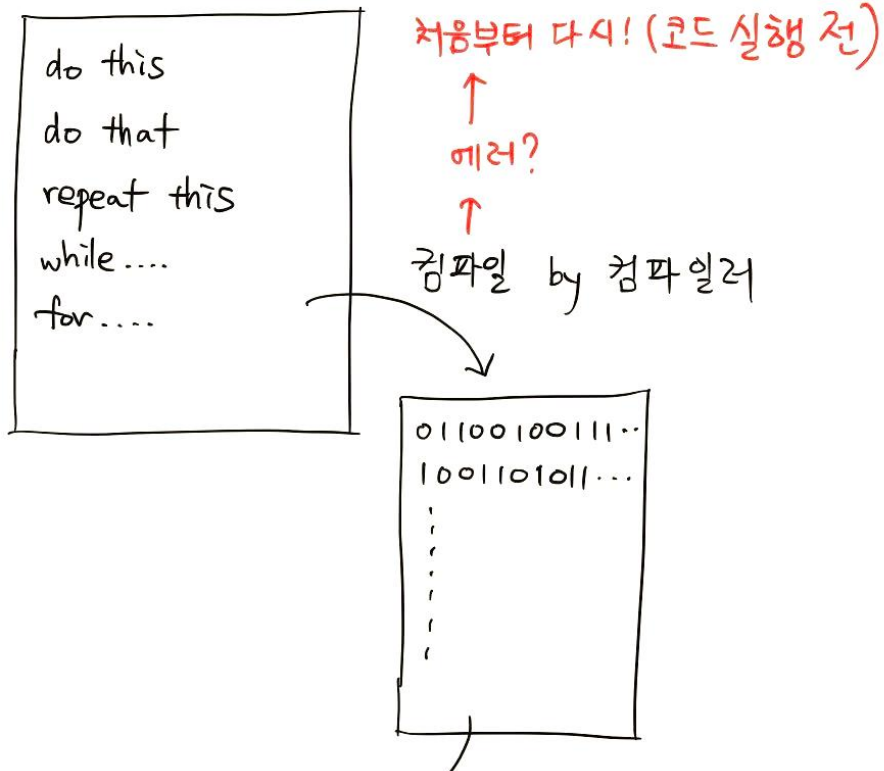


매트랩 = 애플리케이션?



컴파일러 vs 인터프리터

컴파일 언어 (C/C++/C#, Java, ...)

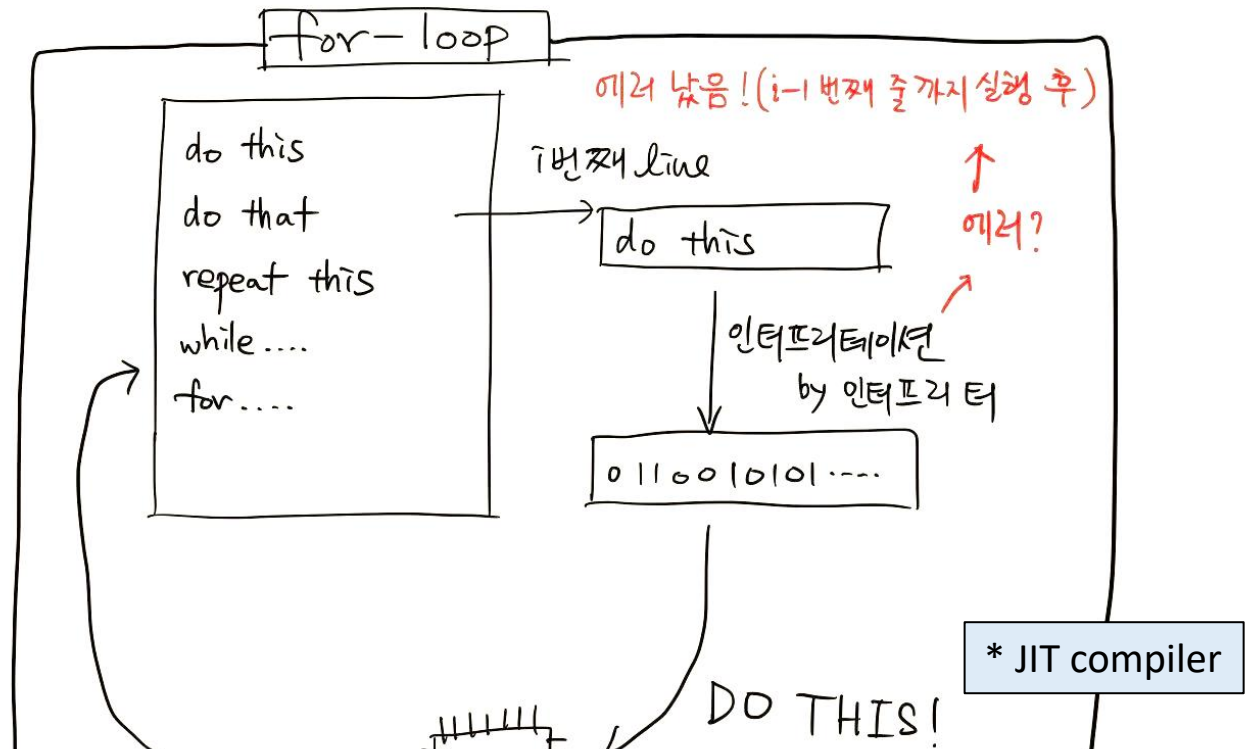


- 한꺼번에 번역 (번역기 = 컴파일러)
 - : 오류 없으면 실행
 - : 오류 있으면 실행하지 않고 멈춤
- 일반적으로 문법이 어렵고 실행속도가 빠름

특히 매트랩은...

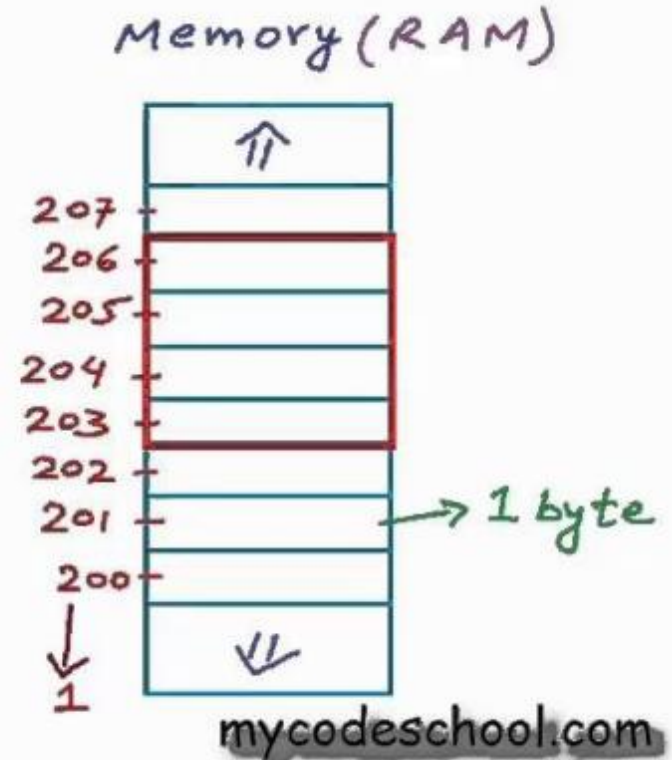
- 명령 창에서 진행상태를 언제든지 확인 가능
- 디버깅이 매우 편리

스크립트 언어 (파이썬, R, JS, 매트랩, ...)



- 한 줄씩 번역 후 실행 (번역기 = 인터프리터)
 - : 오류 발생 전까지는 번역 후 실행까지 함
 - : 오류 발생 시 멈춤
- 일반적으로 문법이 쉽고 빠른 구현이 가능하나 실행속도가 느림

-



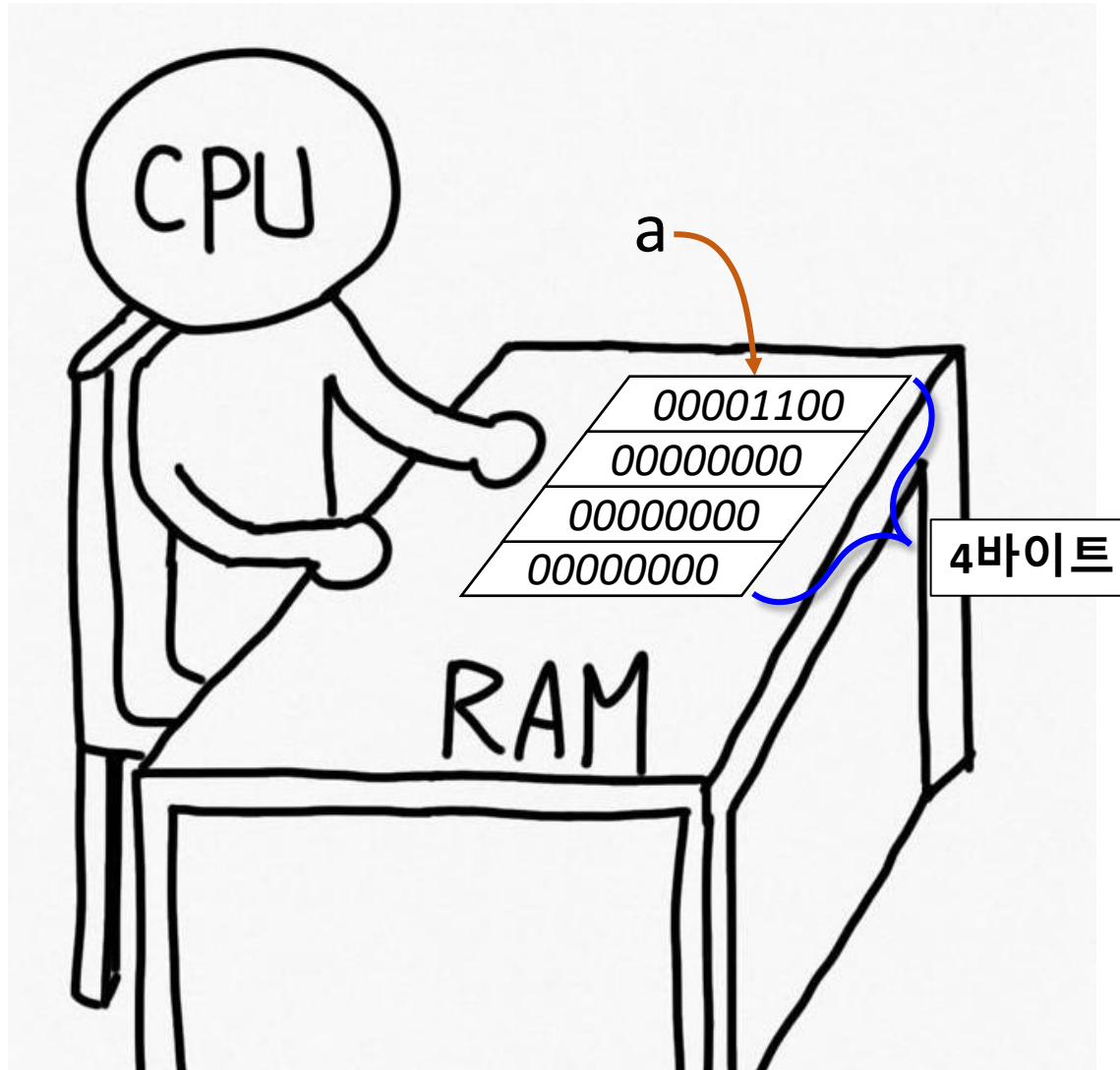
8바이트(64비트) 정수의 표현범위
 $\rightarrow -2^{63} \sim 2^{63} - 1$

a = 12 (4바이트 정수)

00000000 00000000 00000000 00001100

→ 사람이 이해하는 언어

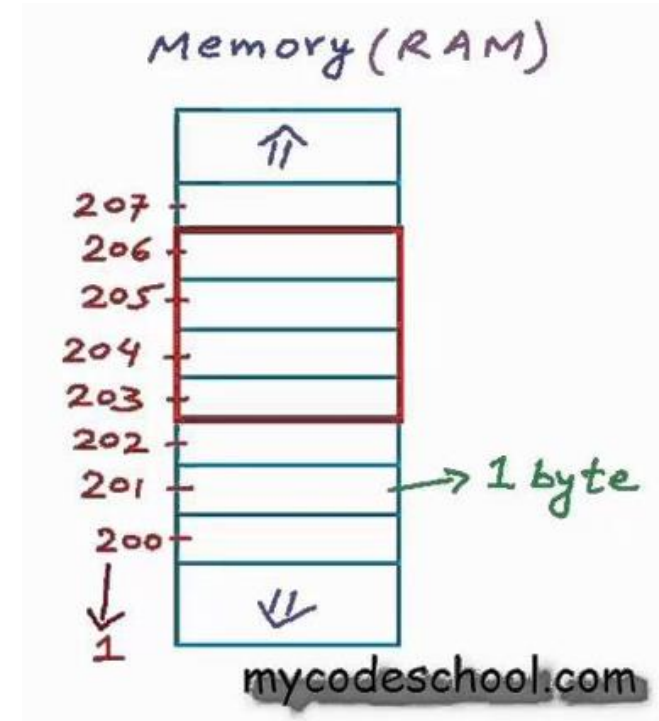
→ 컴퓨터가 이해하는 언어 (기계어)



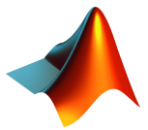
자료형 – 문자가 실제로 저장되는 형태

- ‘Hello’

- H → 72 → 01001000
- e → 101 → 01100101
- l → 108 → 01101100
- l → 108 → 01101100
- o → 111 → 01101111



- ‘Hello’ → 0100100001100101011011000110110001101111



자료형 – 자료형이 필요한 이유

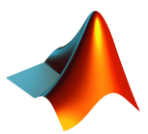
- 흑백 이미지

- 각 픽셀이 1바이트(8비트, 0-255)의 정수 값을 가지고 있음
- 필요한 건 1바이트인데 강제로 4바이트나 8바이트를 쓰게 하면?



```
>> img = imread('cameraman.tif');  
>> imshow(img)
```

img				
256x256 uint8				
	1	2	3	4
1	156	159	158	155
2	160	154	157	158
3	156	159	158	155
4	160	154	157	158
5	156	153	155	159
6	155	155	155	157
7	156	153	157	156

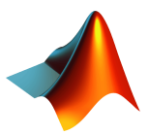


- | | exponent | fraction |
|--------|-----------------|-----------------|
| single | 8 bits | 23 bits |
| double | 11 bits | 52 bits |



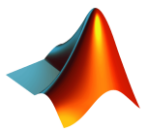
매트랩 사용되는 자료형

- 소수 – double, single
 - 자료형을 지정하지 않은 모든 숫자 변수는 **double**
- 정수 – int8, int16, int32, int64, uint8, uint16, uint32, uint64
- 문자 – char, string
- 유용한 명령어
 - whos: 작업 공간 상의 모든 변수 목록 확인 가능 (크기, 자료크기, 자료형 등)
 - class: 변수의 자료형을 확인
- 실습



강의 요약

- 계산 툴의 발달 과정에서 매트랩이 개발되었다.
 - 매트랩은 다른 언어에 비해 “알아서 해주는 게” 많은 편한 언어이다.
- 명령 창에서는 명령을 한 줄씩 실행할 수 있다.
- 편집기에서는 일련의 명령들을 한번에 실행할 수 있다.
- 모든 것은 정의에서 시작한다.
 - 변수를 정의한다 = 메모리에 값을 저장하고 이름(변수명)을 붙인다
 - 미리 정의된 적이 없는 변수를 사용하려 하면? → 에러 발생
- 자료형
 - 상황에 따라 필요한 자료형이 다를 수 있다.
 - 정수형, 소수형, 문자형 등이 있다.
 - 매트랩의 기본 자료형은 double이다.



강의 로드맵

기초 문법

행렬 다루기

- 행렬 만들기
- 인덱싱, 행렬 연산

시각화

- plot, plot3
- mesh, surf

조건문과 반복문

- if, elseif, else
- for, while

함수

- 사용자 정의 함수 기초

기타

- 파일입출력 (이미지, 텍스트, 엑셀 파일 다루기)
- 프로그래밍 스킬
- cell, struct 자료형

고급 문법

시각화 고급

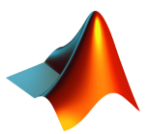
- 그래픽 객체
- gca, gcf, axes, set, get
- 애니메이션

함수 고급

- 로컬함수, 익명함수, 함수핸들
- varargin, nargin

수치해석 활용

- 선형보간
- 회귀분석
- 수치미분
- 수치적분
- 방정식의 해
- 미분방정식의 해
- 테일러 전개
- ...
- ...



Q&A

