

게으른

매트랩

시작합니다.



닉네임: 게으른 ㅇㅇ

블로그: <https://lazymatlab.tistory.com/>

단톡방: 카카오톡에서 MATLAB KR 검색



• 강의의 지향점

- "프로그래밍 언어"로서의 매트랩
- 코드를 쪼개고 합쳐보며, 디테일과 전체를 동시에 바라볼 수 있는 시각을 훈련
- 매트랩에서 가장 중요한 두 가지: 자료형과 함수
- 이후는 각자의 분야에 필요한 툴박스 활용의 문제

• 다룰 것

- MATLAB + 약간의 Image Processing Toolbox
- 수치해석 조금 (이론보다 활용 위주로)

• 다루지 않을 것

- 매트랩 설치 과정 ([블로그 글](#) 참고)
- 제어, 로보틱스, 신경망, 시뮬링크 등

• 다룰지 말지 결정 못한 것

- GUI, OOP





- 각 창의 위치는 드래그로 바꿀 수 있음
- 실수로 창을 닫았다면?
 - 툴스트립 - 홈탭 - 레이아웃 - 디폴트
- 레이아웃은 저장 및 불러오기 가능



명령 창(Command Window)

- 명령 창 = 계산기
 - 한 줄씩 명령을 보냄
 - 프롬프트(>>)가 보이면 명령 입력 가능 상태
 - 위아래 화살표로 명령 기록 확인 및 재사용 가능
 - 간단한 계산, 테스트, 값 확인 용도로만 사용하는 것을 권장
 - 코드가 길어지면 에디터를 사용하는 것이 거의 필수
- 세미콜론의 역할은?
- 명령 기록 중 d로 시작하는 것만 보려면 어떻게 해야 할까?



현재 폴더(Current Folder)

- 경로지정 없이 파일에 바로 접근할 수 있는 폴더
- 현재 폴더가 아닌 폴더의 파일에 접근하려면?
 - 절대경로 작성 (↔ 상대경로: 현재 폴더 기준)
- 폴더 생성: `mkdir` (make directory)
- 폴더 이동: `cd` (change directory)
- 현재 폴더: `pwd` (path of working directory)
- 상위 폴더: `..` (현재 폴더는 `.`)



작업 공간(Workspace)

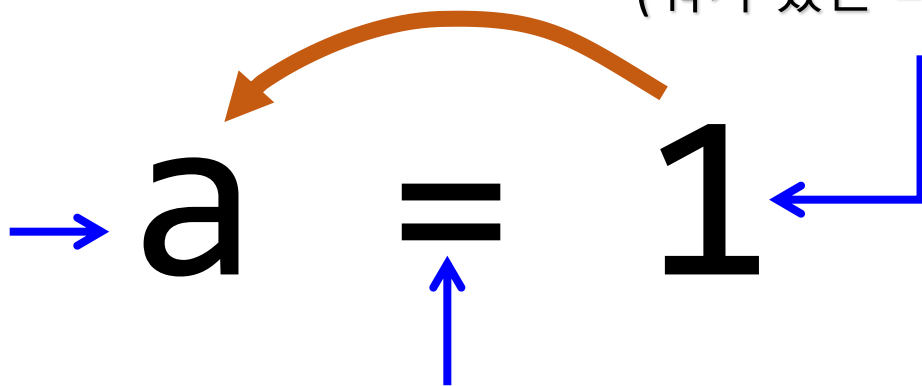
- 메모리 상에 올라와 있는 변수의 목록을 보여주는 공간
- 변수란?
 - 값을 저장 및 변경할 수 있고,
이름(변수명)으로 저장된 값을 읽어올 수 있는
프로그래밍의 기본 요소

미리 “정의”된

값, 표현식, 변수

(뭐가 됐든 미리 정의된 것)

변수
“이름”



대입연산자 (수학의 등호가 아님!)

변수를 “정의”한다.

ex. `a=a+1`

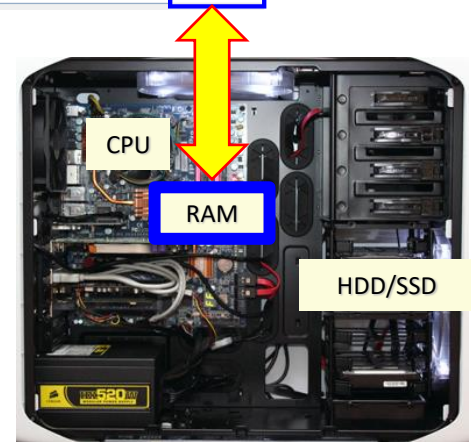
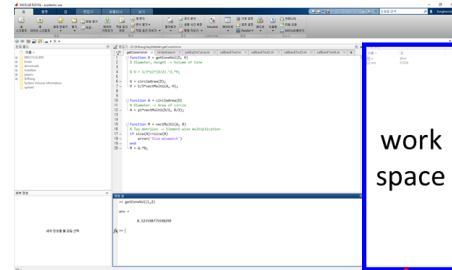
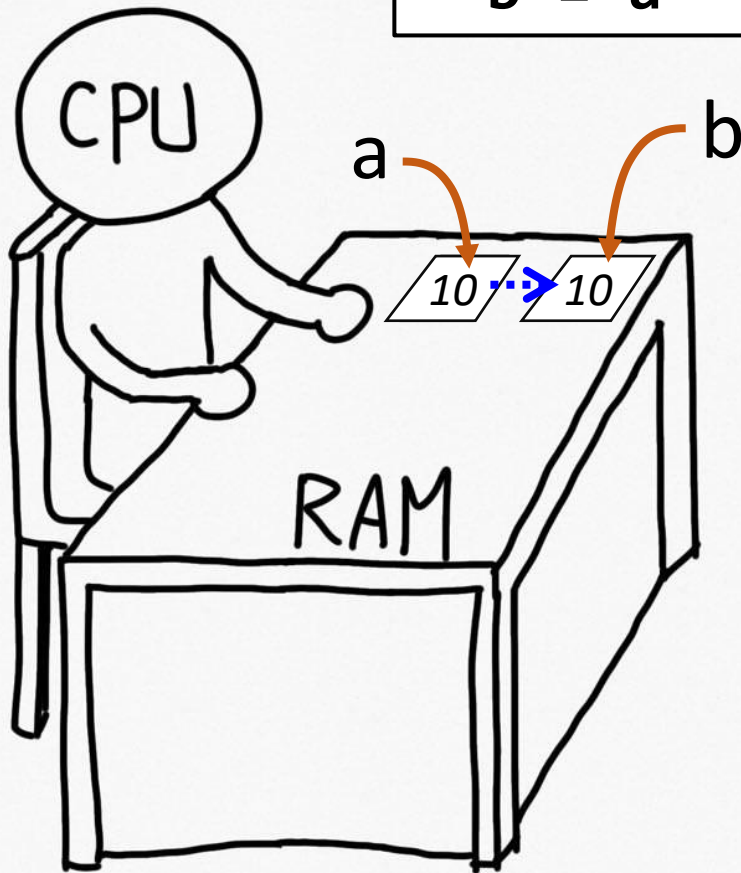


작업 공간(Workspace)



작업 공간(Workspace)

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



RAM (메모리)

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



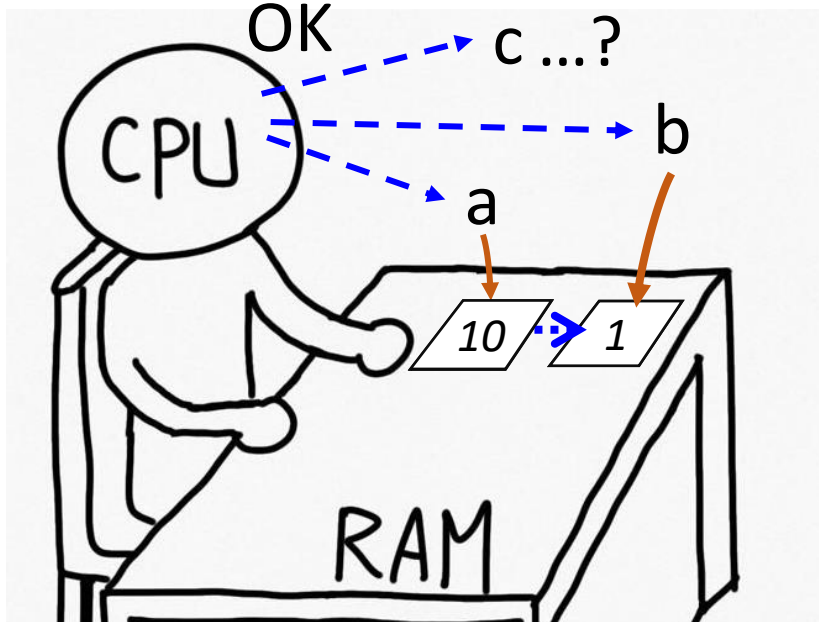
"정의되지 않은 변수입니다."

- 모든 것은 정의(definition)에서부터 시작된다.
- 정의한 적이 있는 변수명 → 이름으로 접근 가능
- 정의한 적이 없는 변수명 → "정의되지 않은 변수입니다."

```
>> a = 1;
>> b = a;
>> a = 10;
>> disp(a)
    10
>> disp(b)
     1
>> disp(c)
```

'c'은(는) 정의되지 않은 함수 또는 변수입니다.

??? → ERROR



b = c → ERROR
c = c+1 → ERROR

* 정의되지 않은 함수?
→ CPU에게 알려주지
않은 함수 (ex. dips(a))



변수 생성 실습

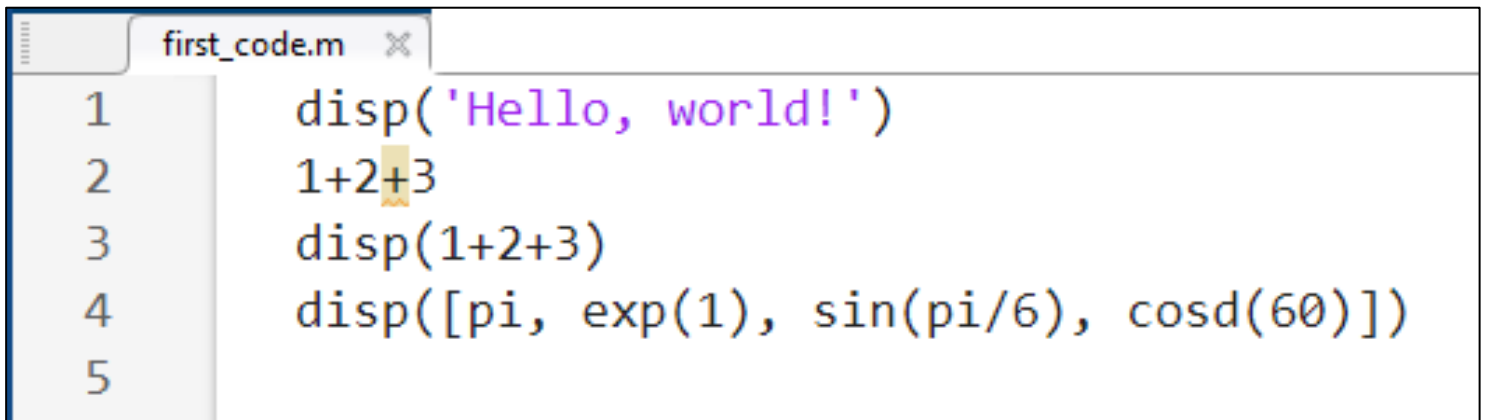
```
>> x = 10
x =
    10
>> x = 3*x - 12
x =
    18
>> y = 20;
>> x + y - 8
ans =
    30
>> z = x + y - 8;
```

- 명령+엔터 → 연산 결과를 출력
: 대입연산도 “연산”이다.
- 명령 끝에 세미콜론(;)
: 연산 결과를 출력하지 않음
- 변수명은 대소문자 구분함
: a와 A는 다른 변수



편집기(Editor)

- 일련의 명령을 작성하여 한번에 실행할 수 있음
 - 명령창은 매번 한줄 한줄 입력해주어야 함
- 편집기로 만든 m 파일은 현재 폴더 창에서 확인 가능
- 편집기를 통해 실행되는 명령들은 명령 창에 나타나지 않음



```
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)])
5
```

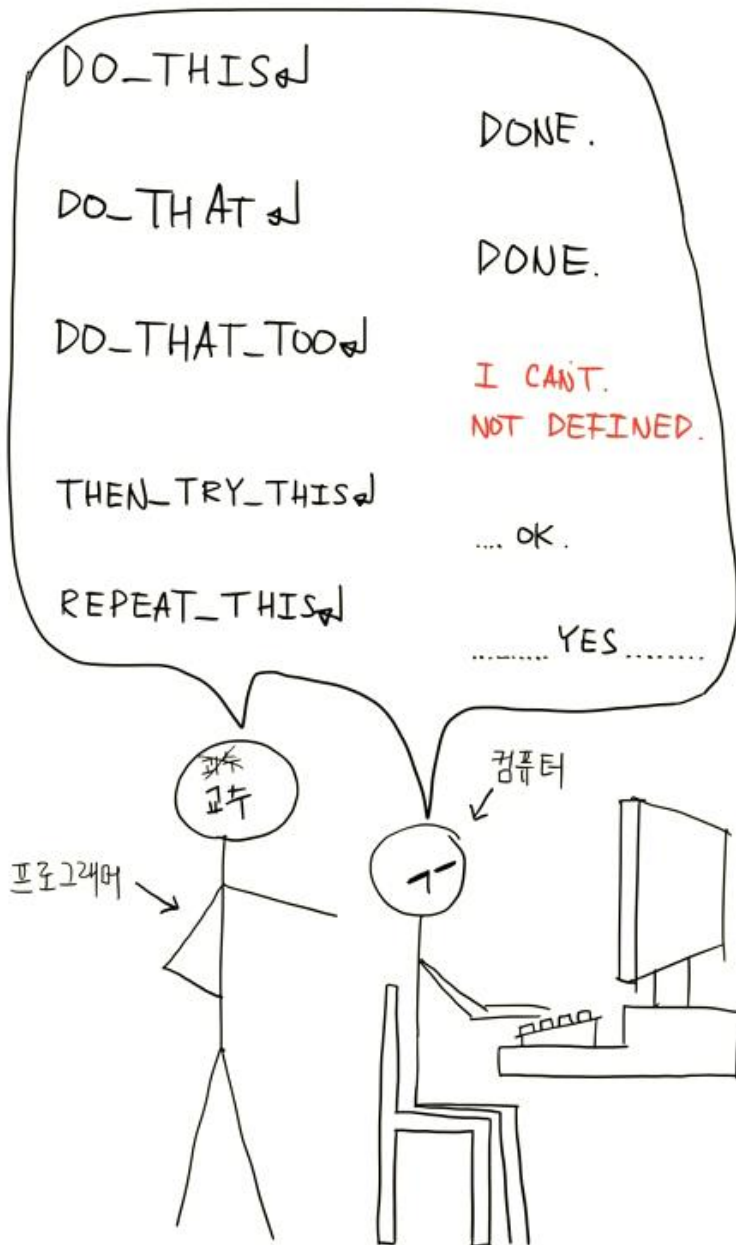
tip:

코드 일부만 선택하고 F9를 누르면
선택된 부분만 실행됨



명령 창 vs 편집기

Command window(명령창)

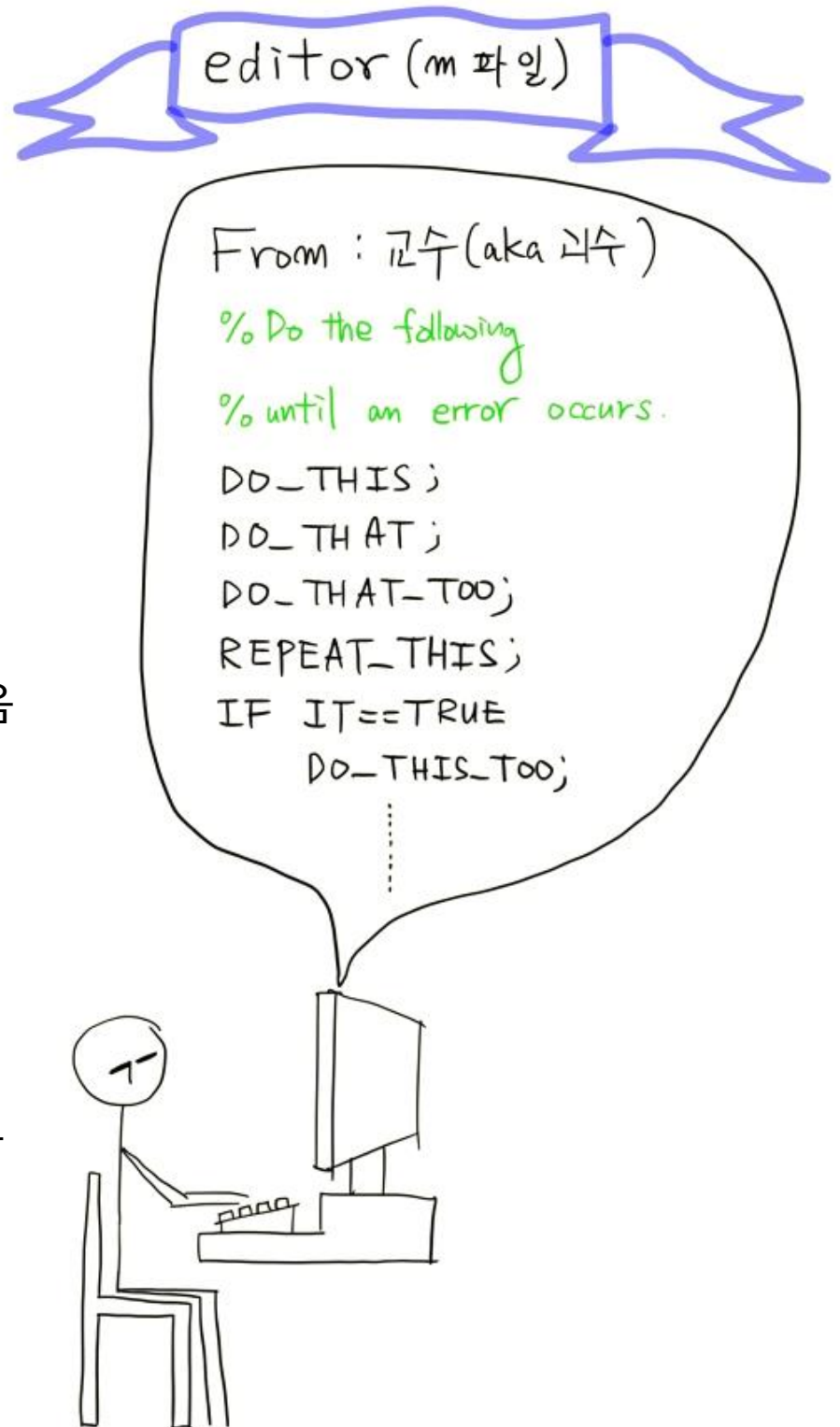


- 대화형
- 명령을 보낼 때마다 수행함
- 잠깐 무언가 확인할 때에만 편함
- 명령 창에 모든 기록이 남으므로 지저분해짐
- 함수를 정의할 수 없음
- 가급적 편집기에 작성하여 파일로 실행하자.



명령 창 vs 편집기

- 문서형
- 명령을 모두 써놓은 후
한번에 순서대로 실행
- 명령이 많아질수록
대화형에 비해
휘어어어얼씬 편함
- 실행해도 명령 창에
아무것도 출력되지 않음
(예외: fprintf, disp,
세미콜론 생략 등)
- 에러가 발생하면
바로 실행 중지
(즉, 에러 발생 전까지는
모두 실행)



도움말(help)

- 간단한 버전: help 함수명
 - 명령어 또는 함수의 기초적인 설명과 입출력을 보여줌

```
>> help disp
disp Display array.
    disp(X) displays array X without printing the array name or
    additional description information such as the size and class name.
    In all other ways it is the same as leaving the semicolon off an
    expression except that nothing is shown for empty arrays.

    If X is a string or character array, the text is displayed.

    See also formattedDisplayText, sprintf, num2str, format, details.

    Documentation for disp
    Other functions named disp
```

- Full 버전: doc 함수명
 - 명령어 또는 함수의 상세한 설명과 예제까지 보여줌



고맙습니다

