MATLAB

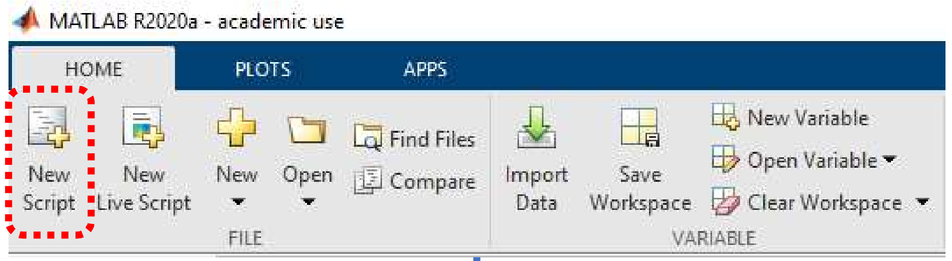
Programming

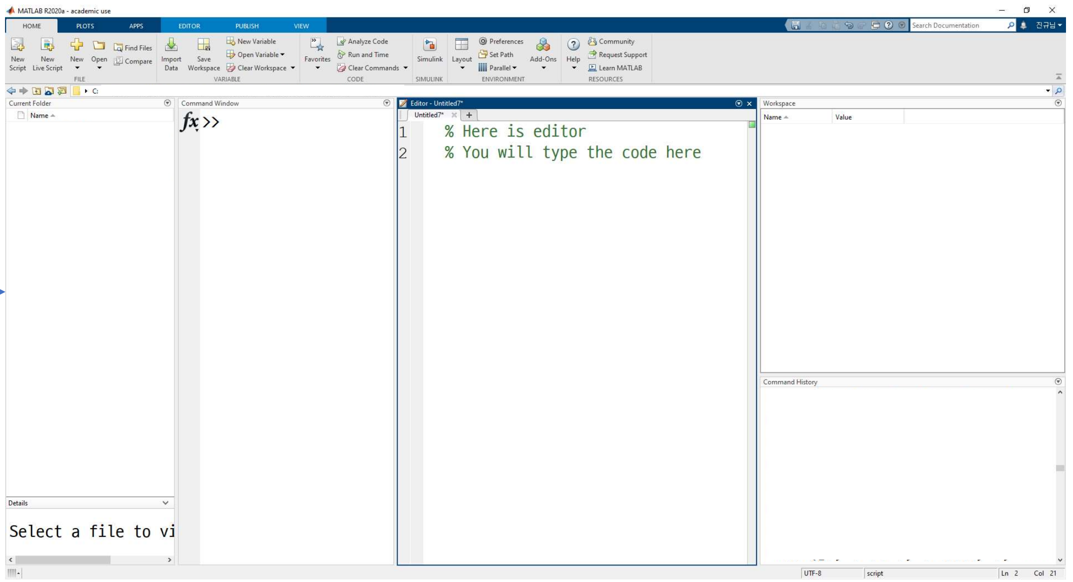
2020-09-25

JINKYU YU (hortensia@kaist.ac.kr)

MATLAB programming

From now on, all MATLAB code will be typed in editor.





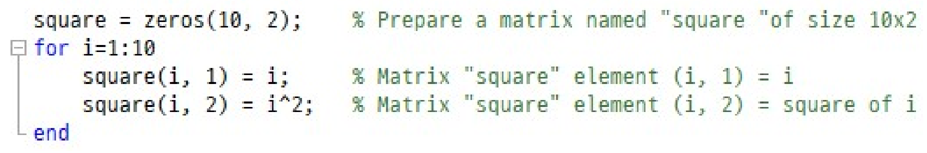
```
% Here is editor
% You will type the code here
```

Refer the [MATLAB basic manual.pdf] which was uploaded at week2.

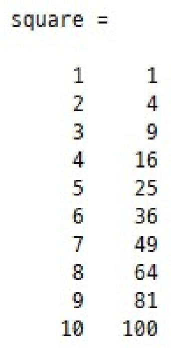
MATLAB programming : “for” loop
- Repeat specified number of times

MATLAB programming “for” loop repeat specified number of times.

For examples, compute the square for 1 to 10.

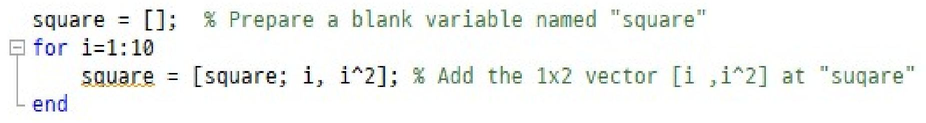


```
square = zeros(10, 2);    % Prepare a matrix named "square "of size 10x2
for i=1:10
    square(i, 1) = i;     % Matrix "square" element (i, 1) = i
    square(i, 2) = i^2;   % Matrix "square" element (i, 2) = square of i
end
```



```
square =

     1     1
     2     4
     3     9
     4    16
     5    25
     6    36
     7    49
     8    64
     9    81
    10   100
```

If you don’t know the variable, the you can add the data at variable.



```
square = [];  % Prepare a blank variable named "square"
for i=1:10
    square = [square; i, i^2]; % Add the 1x2 vector [i ,i^2] at "suqare"
end
```

Here, the orange underline warns that the size of the variable “square” changes.

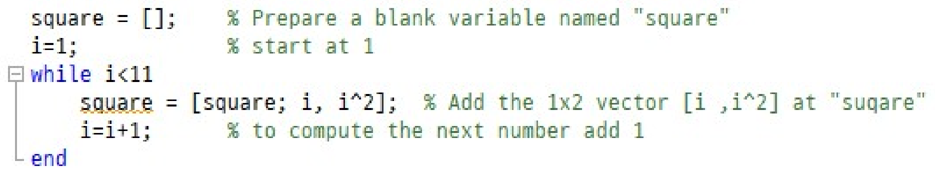
If you know the size of variable, it is recommended to use the first code.

MATLAB programming : “while” loop

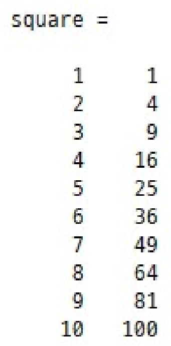
- Repeat when condition is true.

MATLAB programming “while” loop repeats when condition is true.
“for” loop gives the information about the repeat number so you can use first code.
But usually “while” loop do not know about the repeat number.

So, code to compute the square for 1 to 10 with “while” loop is like below.



```
square = [];    % Prepare a blank variable named "square"
i=1;            % start at 1
while i<11
    square = [square; i, i^2];  % Add the 1x2 vector [i ,i^2] at "suqare"
    i=i+1;      % to compute the next number add 1
end
```



```
square =

     1     1
     2     4
     3     9
     4    16
     5    25
     6    36
     7    49
     8    64
     9    81
    10   100
```

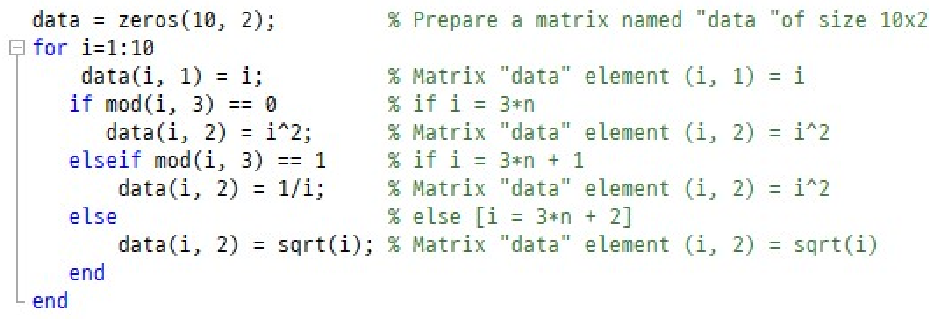
As “while” loop progresses, variable “i” grows 1,2,3, … 9,10,11.
The final value of variable “i” is 11. So condition 11<11 is false. And stop the loop.

MATLAB programming : “if” loop

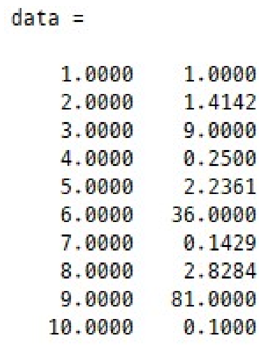
- Execute statements if condition is true.

MATLAB programming “if” execute statements if condition is true.

For examples, let’s compute the values $\begin{cases} i^2, & i \equiv 0 \ \ (mod\ 3) \\ \frac{1}{i} & i \equiv 1 \ \ (mod\ 3) \\ \sqrt{i} & i \equiv 2 \ \ (mod\ 3) \end{cases}$ for 1 to 10.



```
data = zeros(10, 2);         % Prepare a matrix named "data "of size 10x2
for i=1:10
    data(i, 1) = i;          % Matrix "data" element (i, 1) = i
   if mod(i, 3) == 0         % if i = 3*n
      data(i, 2) = i^2;      % Matrix "data" element (i, 2) = i^2
   elseif mod(i, 3) == 1     % if i = 3*n + 1
       data(i, 2) = 1/i;     % Matrix "data" element (i, 2) = i^2
   else                      % else [i = 3*n + 2]
       data(i, 2) = sqrt(i); % Matrix "data" element (i, 2) = sqrt(i)
   end
end
```



```
data =

    1.0000    1.0000
    2.0000    1.4142
    3.0000    9.0000
    4.0000    0.2500
    5.0000    2.2361
    6.0000   36.0000
    7.0000    0.1429
    8.0000    2.8284
    9.0000   81.0000
   10.0000    0.1000
```

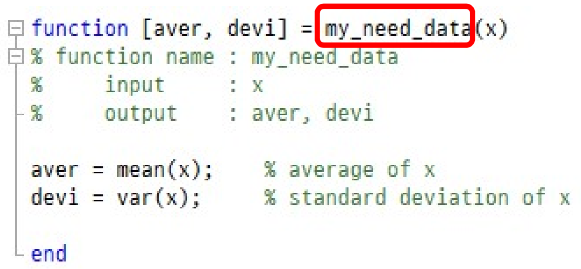
Here “mod(a, b)” gives the remainder part of a÷b.

“for”, “while”, ”if” loop must be finished with “end”.

MATLAB programming : "function"
- Declare the user's function

You can functionalize parts that are frequently used repeatedly.

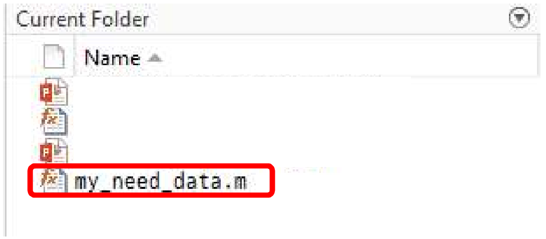
Example) For a given data "x", you need to compute the average and standard deviation of "x".



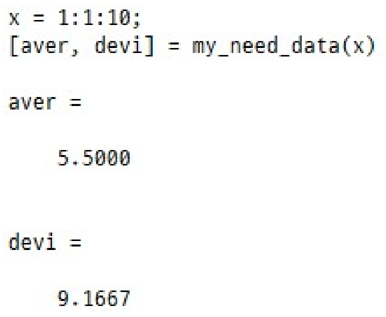
```
function [aver, devi] = my_need_data(x)
% function name : my_need_data
%     input     : x
%     output    : aver, devi

aver = mean(x);    % average of x
devi = var(x);     % standard deviation of x

end
```

When saving a file, make sure that the file name is the same as the function name.

<save the file>



<usage example>

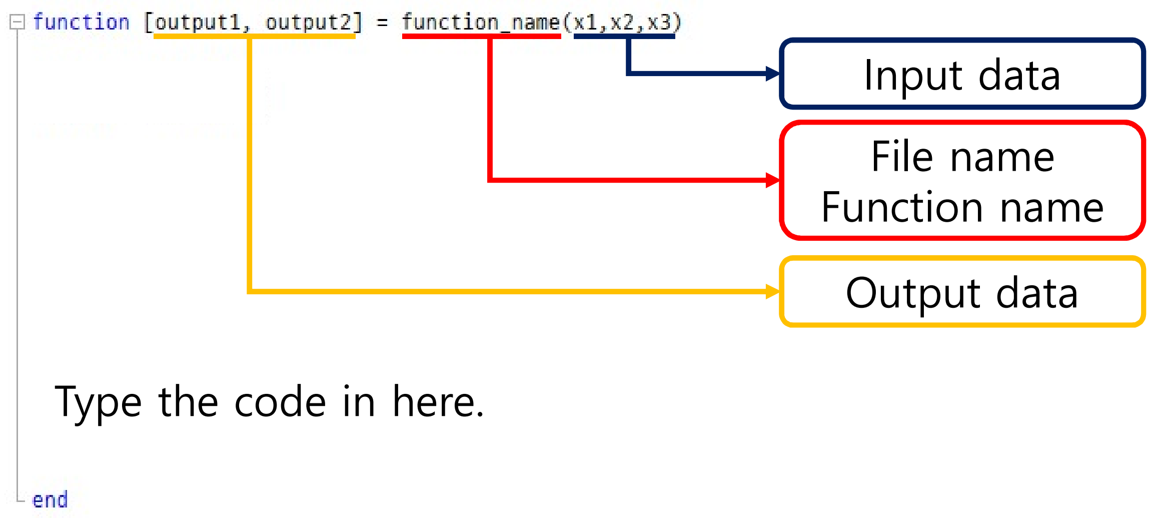


```
x = 1:1:10;
[aver, devi] = my_need_data(x)

aver =

    5.5000


devi =

    9.1667
```

MATLAB programming : "function"

- Declare the user's function

The basic format is like below.



```
function [output1, output2] = function_name(x1,x2,x3)

  Type the code in here.

end
```

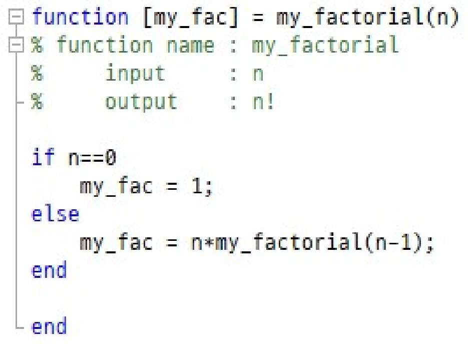
When saving a file, make sure that the file name is the same as the function name.

MATLAB programming : “function”

- Declare the user’s function

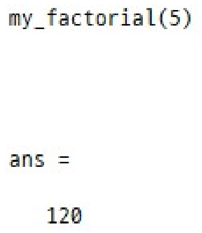
You can make a recursive function.
i.e. When you define a function, you can use that function in definition.
Like a recurrence relation ($ex.\ f_{n+1} = f_n + f_{n-1}$)

For example, we will make a function that compute the factorial $n! = \begin{cases} 1 & n = 0 \\ n \times (n-1)! & n \in N \end{cases}.$



```
function [my_fac] = my_factorial(n)
% function name : my_factorial
%     input     : n
%     output    : n!

if n==0
    my_fac = 1;
else
    my_fac = n*my_factorial(n-1);
end

end
```

Usage example.



```
my_factorial(5)


ans =

   120
```