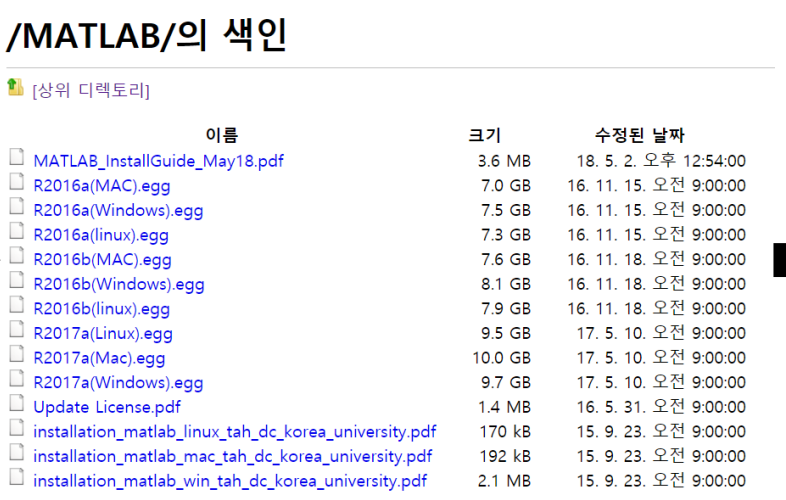
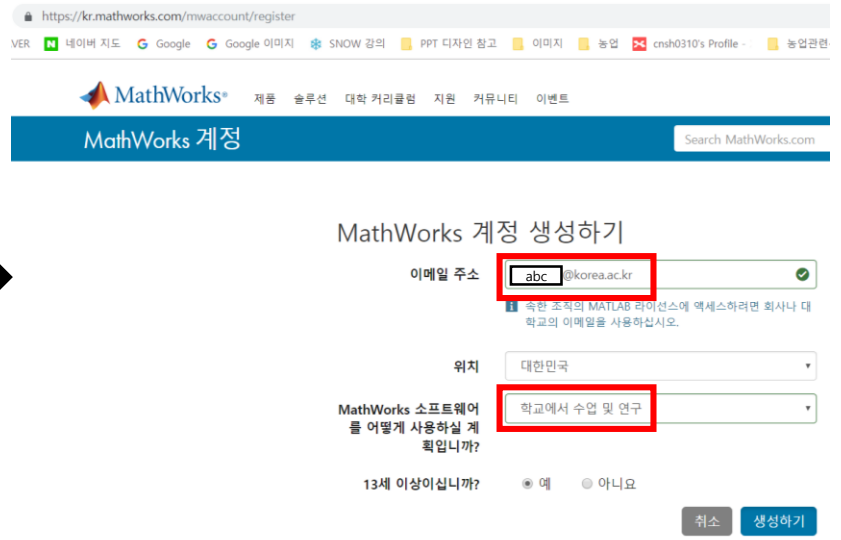
Programming-MATLAB

| Practice class course |

1. Install MATLAB

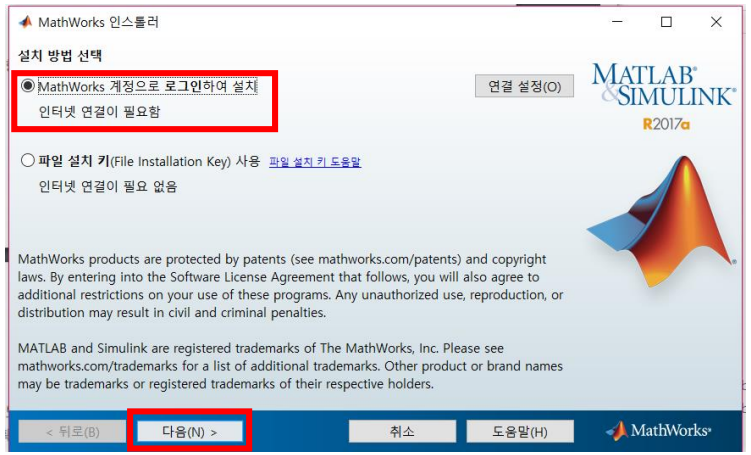
1. Enter the address ftp://ftp.korea.ac.kr in your web browser.
2. Click the directory MATLAB/
3. Choose appropriate R2017a.egg depending on your OS.
4. Download it.
5. During downloading, go to www.mathworks.com and create an account.
6. create an account [*your_ID*@korea.ac.kr]
7. Get a license key

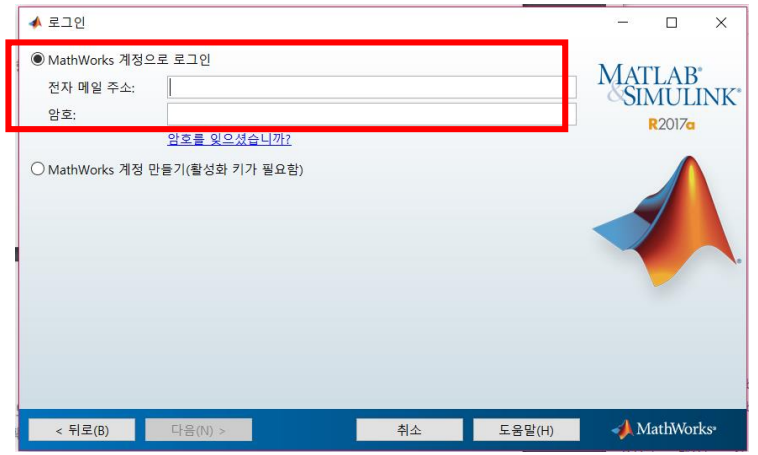


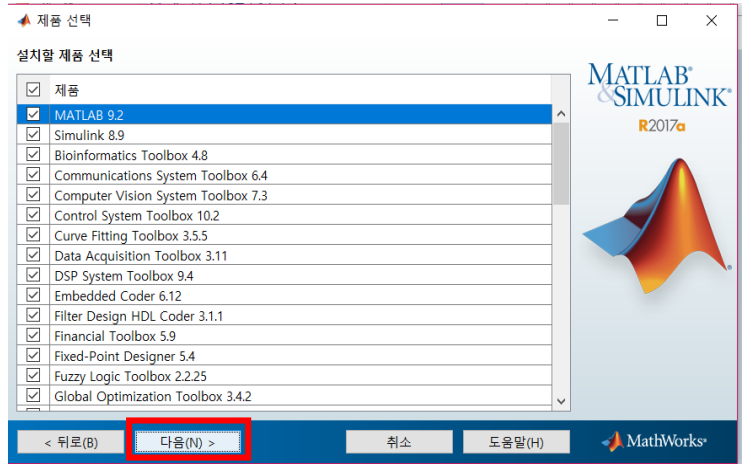


1. Install MATLAB

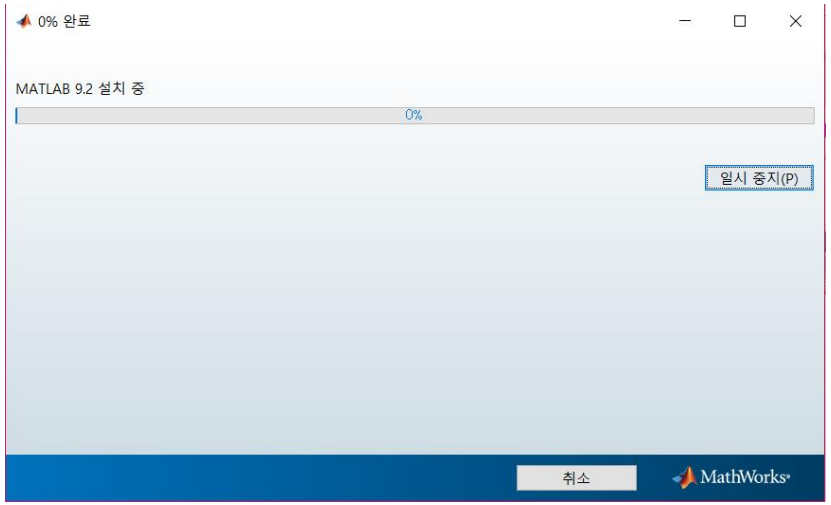
1. Unzip R2017a.egg and double click 'setup'
2. Choose 'install using the internet'
3. Fill in the blanks
4. Click next

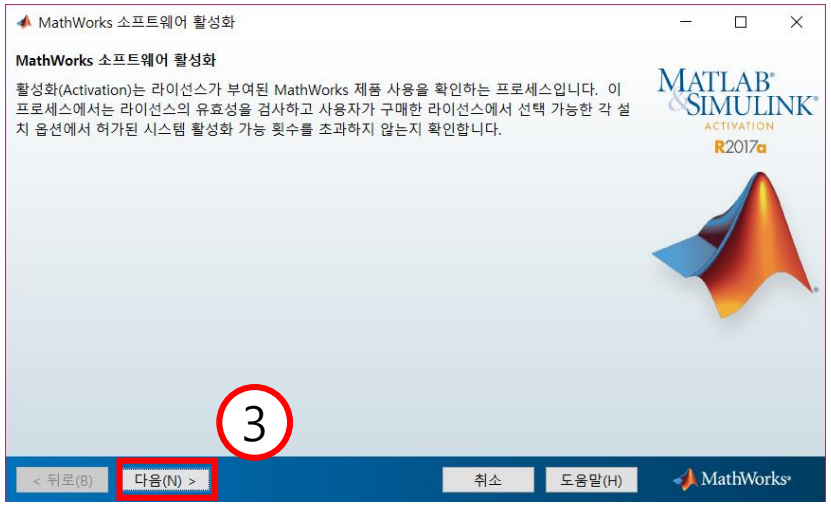


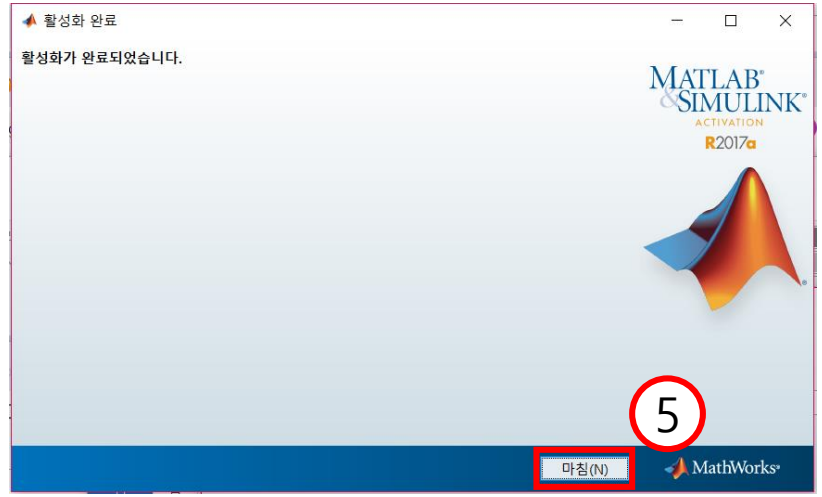




1. Install MATLAB

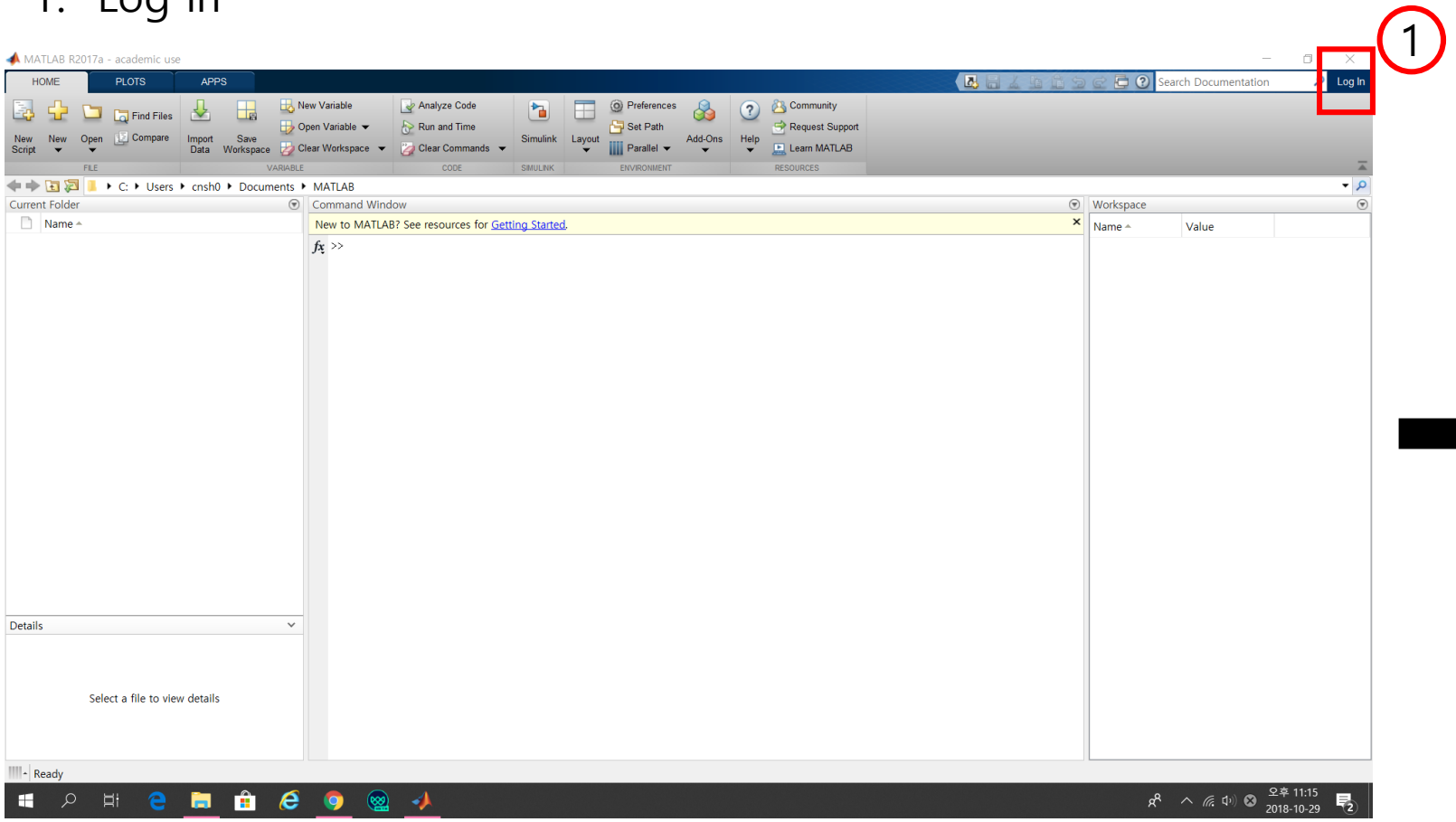


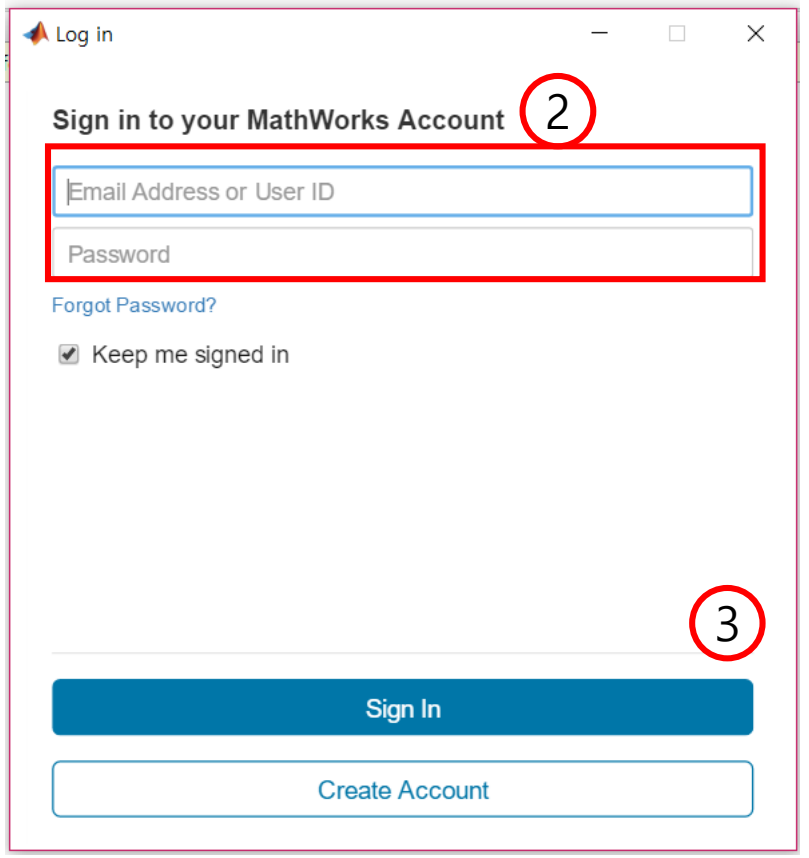




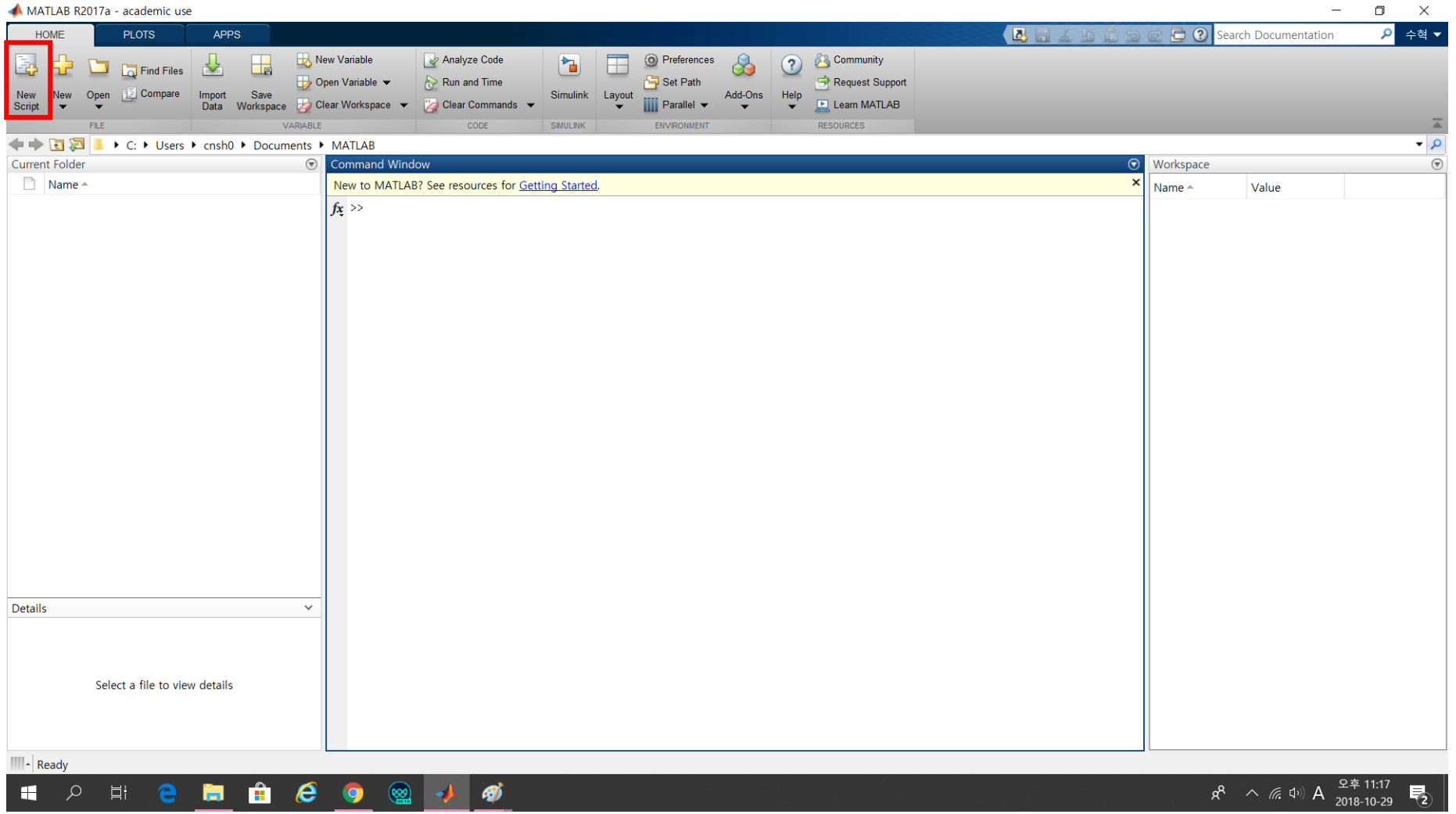
2. Make a new script file

1. Log in





1. Make a new script file



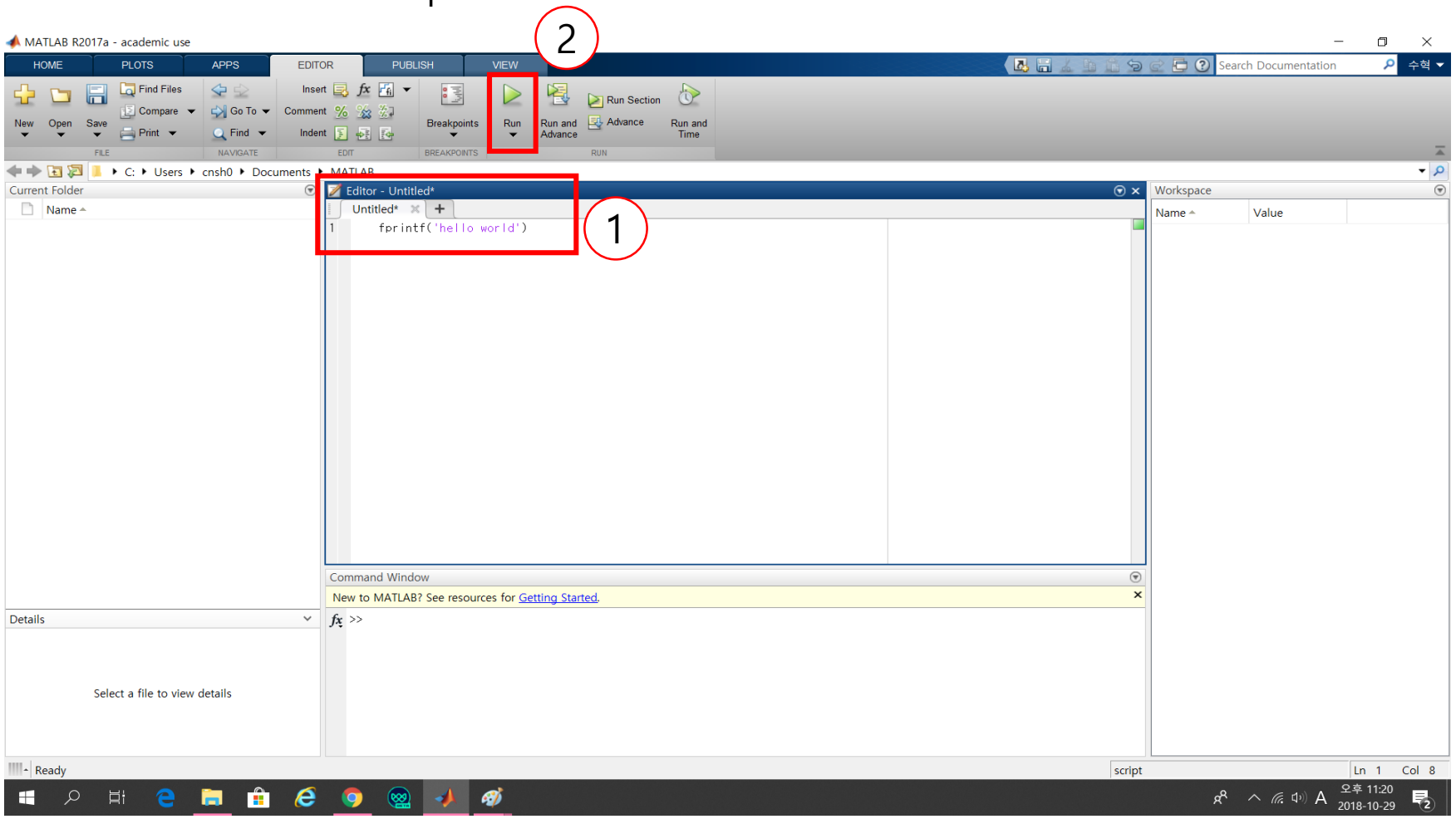
2. Make a new script file

1. Hello World

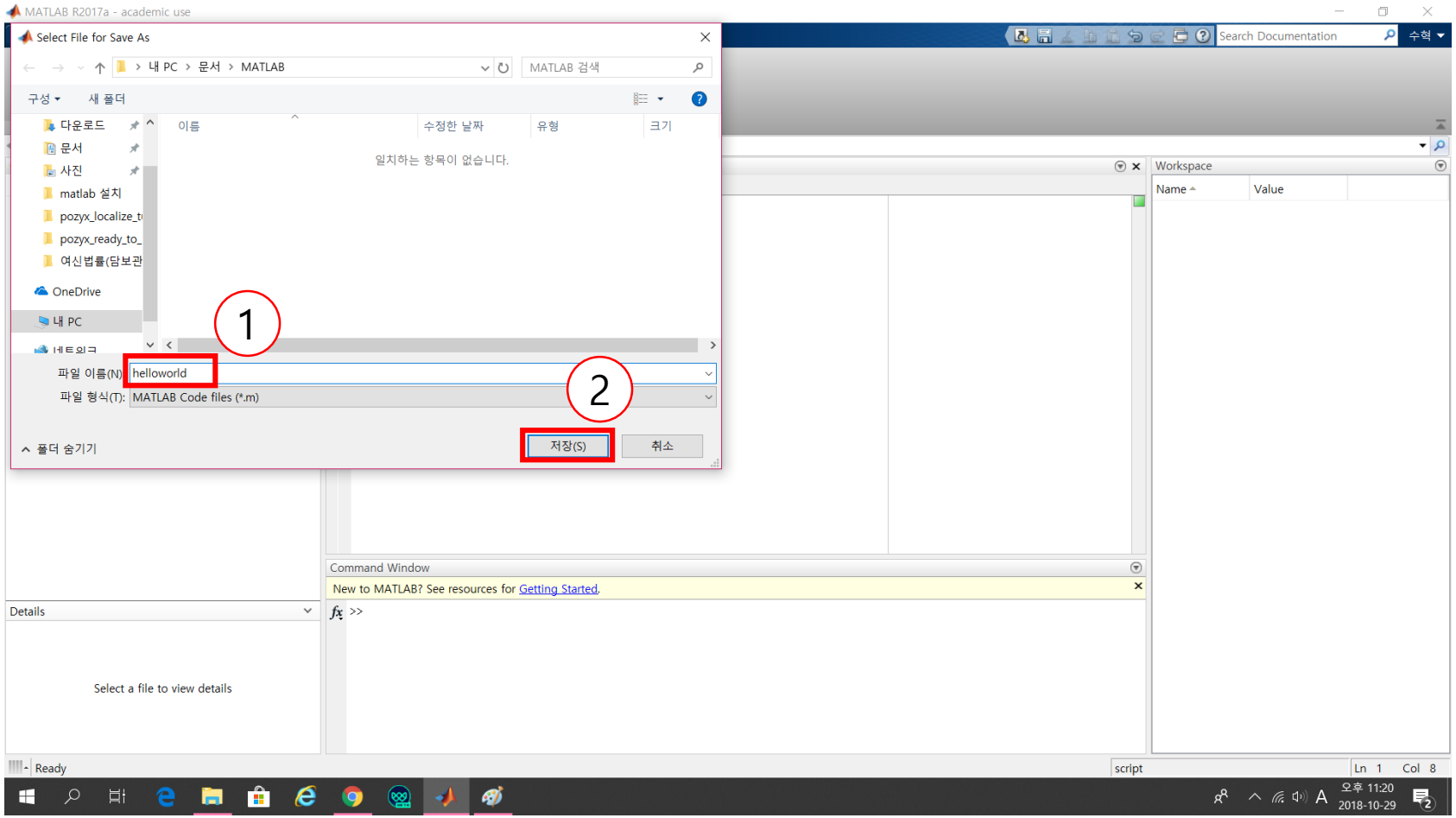
   fprintf('hello world')

   → Same function in C but

   → Matlab dose not require #include <stdio.h>

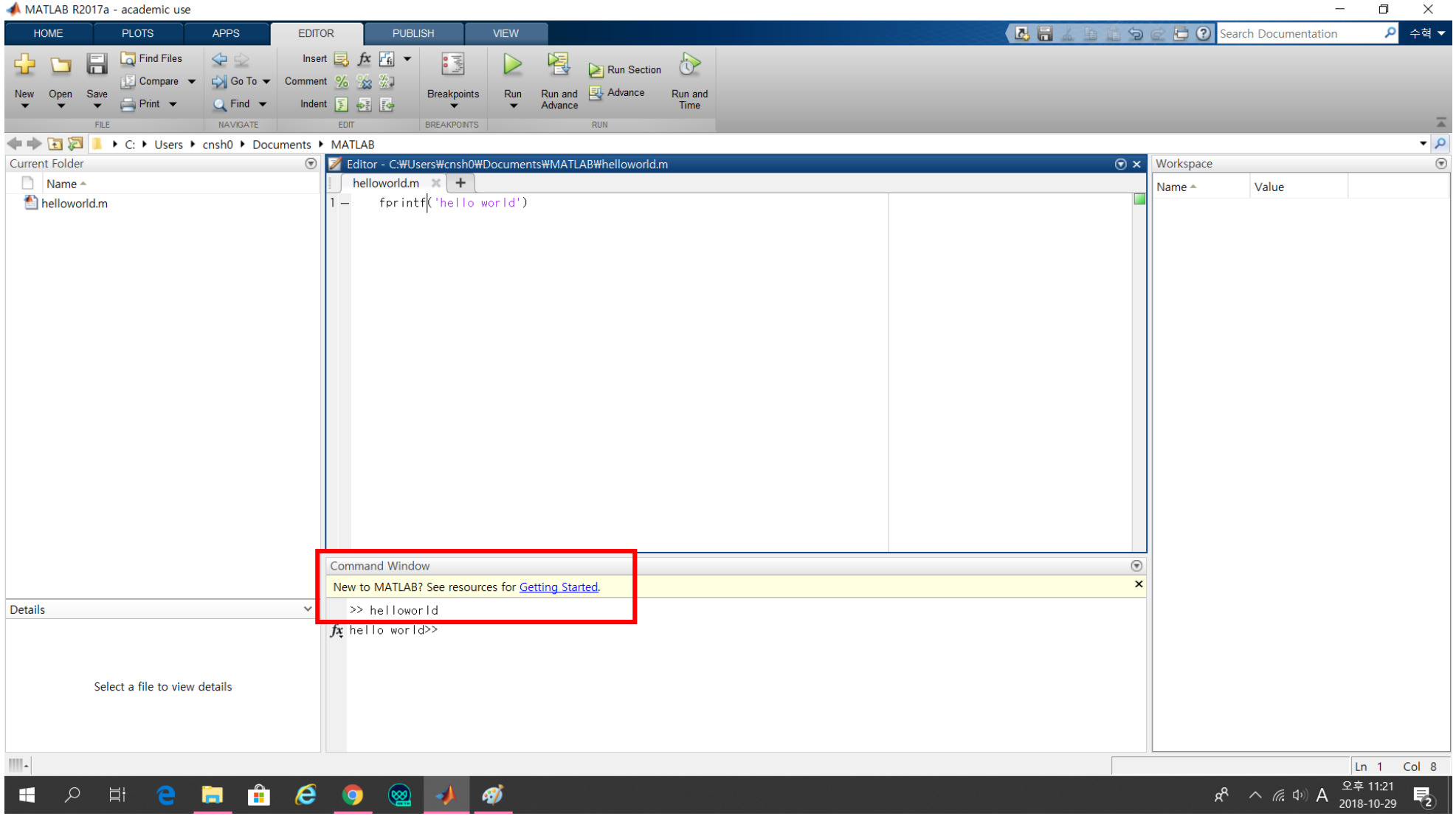


1. Save the script



2. Make a new script file

1. In the command window, you will get an output



❖ Built-in matlab function

The variable t represents time in seconds, and the dimensionless variable y represents the pressure difference acrross valve, normalized by a constant reference pressure. Plot this function for , $t \geq 0$

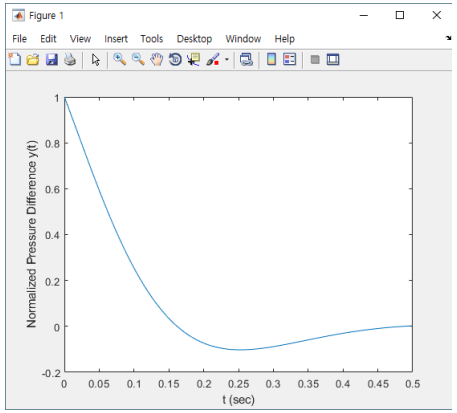
$$y(t) = e^{-8t}\sin(9.7t + \frac{\pi}{2})$$

Equation → source code using built-in functions

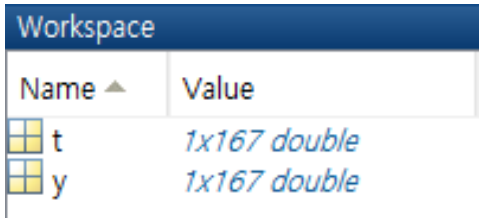
<source code>

```
Command Window
>> t = 0:0.003:0.5;
>> y = exp(-8*t).*sin(9.7*t+pi/2);
>> plot(t,y), xlabel('t (sec)'), ylabel('Normalized Pressure Difference y(t)')
>>
```

<result>



- The built-in matlab functions automatically operate on array arguments to produce an array result the same size as the array argument x.
- Use element-by-element operations(ex. .*, .^) if the arguments are arrays.
- exp(-8*t) and sin(9.7*t+pi/2) will be vectors(=array) the same size as t.



❖ Built-in matlab function

The maximum height *h* achieved by an object thrown with a speed *v* at an angle Θ to the horizontal, neglecting drag, is

$$h = \frac{v^2 sin^2\theta}{2g}$$

Create a table showing the maximum height for the following values of *v* and Θ :

v = 10, 12, 14, 16, 18, 20 m/s Θ = 50°, 60°, 70°, 80°

The rows in the table should correspond to the speed values, and the columns should corresponds to the angles.

<source code>

ex2_20181029.m

```
1 -    g = 9.8; v = 10:2:20;
2 -    theta = 50:10:80;
3 -    h = (v'.^2)*(sind(theta).^2)/(2*g);
4 -    table = [0, theta; v', h]
5
```

v' : transpose v
sind : 'sin' function in degree not radian

<result>

Command Window

```
table =

         0   50.0000   60.0000   70.0000   80.0000
   10.0000    2.9940    3.8265    4.5052    4.9482
   12.0000    4.3114    5.5102    6.4875    7.1254
   14.0000    5.8682    7.5000    8.8302    9.6985
   16.0000    7.6646    9.7959   11.5334   12.6674
   18.0000    9.7006   12.3980   14.5969   16.0322
   20.0000   11.9760   15.3061   18.0209   19.7928
```

- The arrays *v* and *theta* contain the given velocities and angles. The array *v* is 1X6 and the array *theta* is 1X4.
- *v'.^2* is a 6X1 array, and the term *sind(theta).^2* is 1X4 array. Then, *h* is a (6X1)(1X4) = (6X4) matrix
- The array *[0, theta]* is 1X5 and the array *[v', h]* is 6X5, so the matrix *table* is 7X5.

❖ Special matrices

Command	Description
eye(n)	Creates an n X n identity matrix.
eye(size(A))	Creates an identity matrix the same size as the matrix A.
ones(n)	Creates an n X n matrix of 1s.
ones(m, n)	Creates an m X n matrix of 1s.
ones(size(A))	Creates an array of 1s the same size as the array A.
zeros(n)	Creates an n X n matrix of 0s.
zeros(m, n)	Creates an m X n matrix of 0s.
zeros(size(A))	Creates an array of 0s the same size as the array A.

To create and plot the function

$$f(x) = \begin{cases} 10 & 0 \le x \le 2 \\ 0 & 0 < x < 5 \\ -3 & 5 \le x \le 7 \end{cases}$$

<source code>

Editor - C:₩Users₩user₩Documents₩MATLAB₩ex3_20

ex3_20181029.m

```
1 –    x1 = 0:0.01:2;
2 –    f1 = 10*ones(size(x1));
3 –    x2 = 2.01:0.01:4.99;
4 –    f2 = zeros(size(x2));
5 –    x3 = 5:0.01:7;
6 –    f3 = 03*ones(size(x3));
7 –    f= [f1, f2, f3];
8 –    x = [x1, x2, x3];
9 –    plot(x,f), xlabel('x'), ylabel('y')
```

<result>