Matlab introduction

```
ver                 % display information about Matlab and installed packages
```

```
-----------------------------------------------------------------------------------------------------
MATLAB Version: 23.2.0.2428915 (R2023b) Update 4
MATLAB License Number: 40973319
Operating System: Microsoft Windows 10 Education Version 10.0 (Build 19045)
Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------------
MATLAB                                                Version 23.2        (R2023b)
Deep Learning Toolbox                                 Version 23.2        (R2023b)
Econometrics Toolbox                                  Version 23.2        (R2023b)
Financial Toolbox                                     Version 23.2        (R2023b)
Global Optimization Toolbox                           Version 23.2        (R2023b)
MATLAB Compiler                                       Version 23.2        (R2023b)
MATLAB Compiler SDK                                   Version 23.2        (R2023b)
Optimization Toolbox                                  Version 23.2        (R2023b)
Parallel Computing Toolbox                            Version 23.2        (R2023b)
Statistics and Machine Learning Toolbox               Version 23.2        (R2023b)
Symbolic Math Toolbox                                 Version 23.2        (R2023b)
UCSD_GARCH Toolbox                                    Version 2.0
```

```
clc                 % clear the Command Window
clear               % remove all variables in the Workspace
3+4                 % Matlab Matlab saves the most recent calculatin in the variables
ans
```

```
ans = 7
```

```
k=5*6                % assign new variable k, k = 3
```

```
k = 30
```

Vector and Matrices

```
m = [1 2 3 4 5]       % m is a row vector
```

```
m = 1×5
     1     2     3     4     5
```

```
n = [1;2;3;4;5]       % n is a column vector
```

```
n = 5×1
     1
     2
     3
     4
     5
```

```
o = [1:5]             % m and o are the same
```

```
o = 1×5
     1     2     3     4     5
```

```
p = 0:3:9             % p is a column vector in which the element is from 0 to 9
with step wise is 3
```

```
p = 1×4
     0     3     6     9
```

```
p(3)                    % choosing the third element of vector p
```

```
ans = 6
```

```
p(1:3)                  % choosing the first three elements of column vector p
```

```
ans = 1×3
     0     3     6
```

```
p'                      % using ' to transpose vector (matrix) of p
```

```
ans = 4×1
     0
     3
     6
     9
```

```
randn(2,3)              % create 2x3 matrix in which elements are drawn from normal distribution
```

```
ans = 2×3
    0.5377   -2.2588    0.3188
    1.8339    0.8622   -1.3077
```

```
ones(2)                 % create 2x2 matrix in which all elements are 1
```

```
ans = 2×2
     1     1
     1     1
```

```
zeros(1,4)              % create 1x4 vector in which all elements are 0. This is the most common way to initialize variable
```

```
ans = 1×4
     0     0     0     0
```

```
q = [1 2 3 4;
    4 5 6 7]              % q is 2x3 matrix
```

```
q = 2×4
     1     2     3     4
     4     5     6     7
```

```
q(1,2)                  % select the entry in the first row and second column
```

```
ans = 2
```

```
q(2,:)                  % select the entries in the second row but all columns
```

```
ans = 1×4
     4     5     6     7
```

```
q(:,2:3)                % select the entries in all rows and in columns 2 to 3
```

```
ans = 2×2
     2     3
     5     6
```

```
q(:,end)                % choose elements in ALL rows and last column of matrix q
```

```
ans = 2×1
     4
     7
```

```
q(end,:)                % choose elements in the LAST row and ALL columns of matrix q
```

```
ans = 1×4
     4     5     6     7
```

```
q(3)                    % with one index, Matlab counts column-by-column
```

```
ans = 2
```

```
[nrows, ncols] = size(q);               % number of rows (2) and columns (4) of matrix
q
```

Matrices operation

```
A = [1 2; 3 4];
B = [5 6; 7 8];
c = 2
```

```
c = 2
```

```
A+c
```

```
ans = 2×2
     3     4
     5     6
```

```
A*c
```

```
ans = 2×2
     2     4
     6     8
```

```
A^2                     % A to the power
```

```
ans = 2×2
     7    10
    15    22
```

```
A.^2                    % element-wise power
```

```
ans = 2×2
     1     4
     9    16
```

```
eye(size(A,1))/A           % inverse of matrix A
```

```
ans = 2×2
   -2.0000    1.0000
    1.5000   -0.5000
```

```
inv(A)                    % inverse of matrix A (not recommended)
```

```
ans = 2×2
   -2.0000    1.0000
    1.5000   -0.5000
```

```
A\B                       % matrix division, alternative way: B/A or inv(A)*B
```

```
ans = 2×2
    -3    -4
     4     5
```

```
A*B                       % matrix multiplication
```

```
ans = 2×2
    19    22
    43    50
```

```
A.*B                      % element-wise multiplication
```

```
ans = 2×2
     5    12
    21    32
```

Loops and Conditionals

```
% ===== for loop
u = zeros(5,1);    % define u as 5x1 zeros matrix
for ii = 1:5
    u(ii,1) = ii+3;
end
disp(u)
```

```
     4
     5
     6
     7
     8
```

```
% ===== while
w = zeros(5,1);
jj = 1;
while jj<=5
    w(jj,1) = jj+1;
    jj = jj+1;
end
disp(w)
```

```
     2
     3
     4
     5
     6
```

```
% ===== if
```

```
v = 5
```

```
v = 5
```

```
if v>=0
    disp("v is non-negative")
else
    disp("v is negative")
end
```

```
v is non-negative
```

```

if v>0
    disp("v is positive")
elseif v == 0
    disp("v = zeros")
else
    disp("v is negative")
end
```

```
v is positive
```