• Array Manipulation

```
EDU >> A = [1 2 3; 4 5 6; 7 8 9]

A =
     1     2     3
     4     5     6
     7     8     9

EDU >> A(3,3) = 0 % set element in 3rd row, 3rd column
% to zero.

A =
     1     2     3
     4     5     6
     7     8     0

EDU >> A(2,6) = 1 % set element in 2nd row, 6th
% column to one.

A =
     1     2     3     0     0     0
     4     5     6     0     0     1
     7     8     0     0     0     0

EDU >> A(:,4) = 4
EDU >> A = [1 2 3; 4 5 6; 7 8 9]; restore original data
EDU >> B = A(3:-1:1, 1:3) % creates a matrix B
% by taking the rows of A in reverse order.
```

اي اخذ الصف الثالث من A وجعلها الصف الاول من B
ومن ثم اخذ الصف الثاني من A وجعلها الصف الثاني من B
وهكذا ..

```
B =
     7     8     9
     4     5     6
     1     2     3

EDU >> B = A(3:-1:1, :) % The final single colon
% means take all columns, does the same as the preceding
example

B =
     7     8     9
     4     5     6
     1     2     3
```

Array Manipulation   أ.ح.ان 2

```
EDU>> C = [A B(:,[1 3])] % Creates C by appending
% إضافة all rows in the first and third columns of B to
% The right of A.

C =   1   2   3   7   9
      4   5   6   4   6
      7   8   9   1   3

EDU >> B = A(1:2,2:3) % Creates B by extracting
% استخراج the first two rows and last two columns of A.

B =   2   3
      5   6

EDU>> C = [1 3]
>> B = A(C,C) % Uses C to index the matrix A rather than
% specifying them directly.
B =   1   3
      7   9

EDU>> B = A(:) % كون B وذلك بأخذ أعمدة A واحداً بعد الآخر وجعلها
  من stretching بشكل عمود واحد، وبالتسلسل.

B =
  1
  4
  7
  2
  5
  8
  3
  6
  9

EDU>> B = B.'
EDU>> B = A
EDU>> B(:,2)=[] % redefines B by throwing away
% all rows in the second column of original B. When you
% set something equal to empty matrix [], it gets
% deleted.

B =   1   3
      4   6
      7   9
```

EDU>> B=B.'

EDU>> B(2,:)=[] % Throws out the second row of B.

B =

1 4 7

EDU>> A(2,:)=B % replaces the second row of A with B.

A =

1	2	3
1	4	7
7	8	9

EDU>> B=A(:,[2 2 2 2]) % duplicating كون B بمضاعفة كل الصفوف في العمود الثاني من A اربع مرات. (لاحظ A لا 3 صفوف).

B =

2	2	2	2
4	4	4	4
8	8	8	8

EDU>> A % show A again.

A =

1	2	3
1	4	7
7	8	9

EDU>> A(2,2)=[]

??? Indexed empty matrix assignment is not allowed

تبين هذه الرسالة ان الـ Matlab لا يقبل بجعل عناصر او اجزاء من الصفوف او الاعمدة فارغة إلا اذا كانت تشمل كل الصفوف او الاعمدة.

EDU>> B=A(4,:)

??? Index exceeds matrix dimensions.

الـ Matlab لا يقبل ذلك لان A لا اربعة صفوف.

EDU>> B(1:2,:)=A

يجب ان يكون عدد الاعمدة من A و B متساوي

???

EDU>> B(3:4,:)=A(2:3,:)

منح العمود الثاني والثالث من المصفوفة A من نفس حجم ساحة المصفوفة B وبما ان الصف الثاني الى الرابع من المصفوفة B غير موجود. وكذلك بما ان الصف الثاني للمصفوفة B غير مشغول لذا فهو يملأ بالاصفار.

B =

1	4	7
0	0	0
1	4	7
7	8	9

Array Manipulation تابع ٤

```
EDU>> G(1:6) = A(:,2:3) % Creates a row vector
% G by extracting all rows in the second and third
% columns of A.

   G =
       2   4   8   3   7   9
```

```
EDU>> A(2,:) = 0 %
```

املأ عناصر الصف الثاني لكل الأعمدة من المصفوفة A بالأصفار

```
   A =  1   2   3
        0   0   0
        7   8   9
```

```
EDU>> A(2,:) = [0 0 0]
```

مماثل للمثال السابق

```
   A =  1   2   3
        0   0   0
        7   8   9
```

In Matlab the index counts elements down the columns, starting with the first. For example.

```
EDU>> D = [1 2 3 4;5 6 7 8; 9 10 11 12] % New data

   D =  1    2    3    4
        5    6    7    8
        9   10   11   12

EDU>> D(2) % second element
   ans =
        5

EDU>> D(5)
   ans =
        6

EDU>> D(end)
   ans =
        12

EDU>> D(4:7)
   ans =
        2   6   10   3
```

- Array Size:

```
EDU >> A=[1 2 3 4; 5 6 7 8];
EDU >> S=size(A)
        S =
            2    4
```

نلاحظ انه 2 تشير الى عدد الصفوف و 4 الى عدد الاعمدة.

```
EDU >> [r, c] = size(A) % r عدد الصفوف  c عدد الاعمدة
        r =
            2
        c =
            4
EDU >>  r = size(A, 1) % number of rows
        r =
            2
EDU >>  c = size(A, 2) % number of columns
        c =
            4
EDU >> length(A) % يعطي اي رقم اكبر هو عدد الصفوف او الاعمدة
        ans =
            4
EDU >>  B = pi: 0.01: 2*pi;
EDU >>  size(B) % shows that B is a row vector
        ans =
            1    315
EDU >>  length(B)
        ans =
            315
EDU >> size([]) % zero size.
        ans =
            0    0
```

Practice: تطبيقات

يتم تنفيذها على الحاسبة والطالبة كل تتبع مع الشرح.

1. If x and y are vectors, then x(y) is a vector [x(y(1)), x(y(2)), ---, x(y(n))], where n = length(y).

```
>> x = [3 -1 4 2 7 2 3 5 4];
>> y = 1:2:9; x(y)
>> y = [5 5 5]; x(y)
>> y = [1 5 1 1 7]; x(y)
```

2. Create a row (or column) vector of n uniformly spaced elements:

```
>> a = -1; b = 1; n = 10;
>> x1 = a:2/(n-1):b          % a row vector
>> y1 = (a:2/(n-1):b)'       % a column vector
>> x2 = linspace(a,b,n)      % a row vector
>> y2 = linspace(a,b,n)'     % a column vector
```

3. Shift K (K should be positive) elements of a vector:

```
>> x = [3 -1 4 2 7 2 3 5 4];
>> x([end 1:end-1]) % shift (right) (or down for columns) 1 element.

>> K = 5;
>> x([end-K+1:end 1:end-K]) % shift right (or down for columns) K elements.
>> x([2:end 1]) % shift left (or up for columns) 1 element.
>> x([K+1:end 1:K]) % shift left (or up for columns) K elements.
```

Practice sheet

4. Initialize a vector with a constant:

```
>> n = 1000;
>> x = 5* ones(n,1)        % 1st solution
>> Z= Zeros (n,1); Z(:)=5  % 2nd solution - should be faster
```

أعد التطبيق بجعل n=10000 ولاحظ سرعة الحل.

5. Create an n x n matrix of 3's.

```
>> n = 1000;
>> A= 3* ones (n,n)
```

6. Create a matrix consisting of a row vector duplicated m times:

```
>> m = 10;
>> X = 1: 5;
>> A = ones (m,1) * X        % 1st solution
>> B = X( ones(m,1),:)       % 2nd solution - should be faster
```

7. Given a vector x, Create a vector y in which each element is replicated n times:

```
>> n=5; x=[2 1 4];
>> y= x(ones(1,n),:); y=y(:)'
>> x=[2 1 4]';
>> y = x(:,ones(1,n))'; y=y(:)
```

8. Reverse a vector:

```
>> x=[3 -1 4 2 7 2 3 5 4];
>> n= length(x)
>> x(n:-1:1)      % 1st solution
>> fliplr(x)      % 2nd solution
```

ابحث عن فكرة عمل الدالة fliplr من خلال أمثلة
أو من خلال help.