

Record of work in Matlab during class lecture, August 14, 2017, with some added comments

These are examples of material in Chapter 2 - vectors and matrices

**Instructions entered in command window are in bold**

*Results echoed to command window are in italic*

**a = 0:2:10** % create a row vector using colons, start:increment:end

*a =*

*0 2 4 6 8 10*

**b = 0:1:6** % colons are used to specify ranges

*b =*

*0 1 2 3 4 5 6*

**c = 0:6** % default increment is 1

*c =*

*0 1 2 3 4 5 6*

**d = c'** % ' is transpose operator, switches rows and columns

*d =*

*0  
1  
2  
3  
4  
5  
6*

**e = d'**

*e =*

*0 1 2 3 4 5 6*

**a = [0 1 2 3]** % create array using values inside square brackets

*a =*

*0 1 2 3*

**a = [0,1,2 3]** % commas and spaces can separate elements in rows

*a =*

*0 1 2 3*

**b = [0; 1; 2]** % semicolons separate rows

*b =*

*0  
1  
2*

```
% new lines also can be used to separate rows during entry
```

```
b = [0  
1  
2  
3]
```


```
b =
```

```
0  
1  
2  
3
```

```
b = [0:3 4; 6:10]
```

```
b =
```

```
0    1    2    3    4  
6    7    8    9   10
```



```
b(1,3) = 9 % note parenthesis ( ), reference indexes row 1, column 3 of array b
```

```
b =
```

```
0    1    9    3    4  
6    7    8    9   10
```



```
b(:,3) % reference all rows, column 3, a bare colon means entire range
```

```
ans =
```

```
9  
8
```

```
length(b) % length returns length of longest array dimension, here columns
```

```
ans =
```

```
5
```

```
b = b' % transpose b
```

```
b =
```

```
0    6  
1    7  
9    8  
3    9  
4   10
```

```
length(b) % now the longest dimension is the number of rows
```

```
ans =
```

```
5
```

```
numel(b) % numel returns total number of elements in an array
```

```
ans =
```

```
10
```

```
size(b) % size returns number of elements of each dimension in an array
```

```
ans =
```

```
5     2
```

```
[r c] = size(b) % here assign result to variables so we can use later
```

```
r =
```

```
5
```

```
c =
```

```
2
```

```
b(2:4,2) % reference row index 2 to 4, column index 2 of array b
```

```
ans =
```

```
7
```

```
8
```

```
9
```

```
b(2,2)
```

```
ans =
```

```
7
```

```
b(2,2) = 99
```

```
b =
```

```
0     6
```

```
1    99
```

```
9     8
```

```
3     9
```

```
4    10
```

```
b(7) % sequential index = 7 of array b, count down cols starting at col 1
```

```
ans =
```

```
99
```

```
b(1)
```

```
ans =
```

```
0
```

```
b(2)
```

```
ans =
```

```
1
```

```
a = [1 2 3 ; 4 5 6; 7 8 9];
```

```
a =
```

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

```
a(2,3) = 99 % assign value of 99 to element in row 2, column 3 of array a
```

```
a =
```

```
    1    2    3  
    4    5   99
```

```
% a(2,3) = [ ]; % bad, throws error
```

```
a(2,:) = [ ] % assign a "null vector" (empty vector) to row 2, deleting it
```

```
a =
```

```
    1    2    3  
    7    8    9
```

```
a(:,3) = [ ] % assign a "null vector" column 3
```

```
a =
```

```
    1    2  
    7    8
```

```
% help rand % comment out here, use to get instructions to use functions
```

```
rand(2) % puts random (really pseudo random) numbers in range 0 to 1 into 2x2 array
```

```
ans =
```

```
    0.0855    0.8010  
    0.2625    0.0292
```

```
rand(2,3) % puts random numbers into 2x3 array
```

```
ans =
```

```
    0.9289    0.4886    0.2373  
    0.7303    0.5785    0.4588
```

```
randi([2 9],2) % puts random INTEGERS in range 2 to 9 into 2x2 array
```

```
ans =
```

```
    9    6  
    6    3
```

```
randi([2 9],2,3) puts random integers in range 2 to 9 into 2x3 array
```

```
ans =
```

```
    5    7    4  
    6    5    9
```

```
a = [1 2 3];
b = [4 5 6];
a .* b % dot-multiply does element-by-element array multiplication
```

```
ans =
     4     10     18
```

```
a ./ b % dot-divide does element-by-element array division
```

```
ans =
    0.2500    0.4000    0.5000
```

NOTE: dot is optional when multiplying an array by a constant (scalar) and when dividing an array by a constant, e.g.,

```
5 * a
```

```
ans =
     5     10     15
```

```
5 .* a
```

```
ans =
     5     10     15
```

```
a / 5
```

```
ans =
    0.2000    0.4000    0.6000
```

```
a ./ 5
```

```
ans =
    0.2000    0.4000    0.6000
```

NOTE: you should also learn about creating vectors with the standard function `linspace` (linearly spaced values), e.g.,

```
a = linspace(0,2,15) % (1st value, last value, number of values)
```

```
a =
```

```
Columns 1 through 9
     0     0.1429     0.2857     0.4286     0.5714     0.7143     0.8571     1.0000
1.1429
```

```
Columns 10 through 15
```

```
    1.2857    1.4286    1.5714    1.7143    1.8571    2.0000
```