

Class 3

```
v = [12 13 14]
```

```
x = find( v>12 & v<=14 )  
2 3
```

```
v(x) = v(x) + 2 % only values in vector v which were >2 & <=4 are increased by 2  
2 5 6
```

```
v = v(x)+2 % here we create new vector v which is subset of previous vector v  
7 8
```

```
v2 = [3 5 9 8 5]  
3 5 9 8 5
```

```
x=find(v2==5)
```

```
v2(x)=0 % put 0 only in places where before was 5
```

```
3 0 9 8 0
```

```
x = find( v>2 | v<=4 ) % OR
```

```
1 2
```

```
v = [2 3 4]  
x = find( v>2 | v<=4 )  
1 2 3
```

Exercise: in v= [1 2 3 4 5 6 7 8 9] find((v>3 & v< 6) | v > 8)

```
clear all
```

```
Arr = rand( 2, 3 ) % create array with random numbers between 0-1
```

```
0.8147 0.1270 0.6324  
0.9058 0.9134 0.0975
```

```
Arr = rand( 2, 3 ) % now it will generate new set of random numbers
```

```
0.2785 0.9575 0.1576  
0.5469 0.9649 0.9706
```

```
Arr = rand( 2, 3 ) * 10 % random number between 0-10
```

```
7.9221 6.5574 8.4913  
9.5949 0.3571 9.3399
```

Exercise: make array with random numbers [-2:6]

```
rand( 2, 3 ) *8 - 2
```

% To set the seed for random number generator

```
rng(1)
```

```
rand(2, 3)
```

```
rand(2, 3)
```

```
round( 2.2) % round number to nearest integer
```

```
2
```

```
round( 2.8)
```

```
3
```

```
ceil(3.6)
```

```
floor(3.6)
```

```
round( Arr )
```

```
8 7 8
```

```
10 0 9
```

```
Arr = rand( 2, 3 ) - 0.5 % random numbers between -0.5 +0.5
```

```
0.1787 0.2431 0.1555
```

```
0.2577 -0.1078 -0.3288
```

```
Arr = randi([-10 10], 2, 3) - 0.5 % generate random integer between -10 +10
```

```
3.5000 -5.5000 -8.5000
```

```
-10.5000 -10.5000 6.5000
```

```
round = 3 % this changes function to variable
```

```
round(Arr)
```

Subscript indices must either be real positive integers or logicals.

```
clear round % restore original function of round
```

```
round( 3.3)
```

```
3
```

```
rand(2,3)*5 -2 % random numbers between -2+3
```

```
1.8276 -1.0656 0.2279
```

```
1.9760 0.4488 1.2316
```

```
ones(2,3)
```

```
1 1 1
```

```
1 1 1
```

```
zeros(2,2)
```

```
0 0
0 0
1
```

```
Exercise: make array with 10s
```

```
ones(2,3)*10
```

```
10 10 10
10 10 10
```

```
zeros(2,2) +10
```

```
10 10
10 10
```

```
A = nan(5, 10)
```

```
A = inf(5, 10)
```

```
A = false(5, 10)
```

```
A = true(5, 10)
```

```
A = 1:100;
```

```
B = reshape(A, 10, 10)
```

```
B = reshape(A, 2, 50)
```

```
B = reshape(A, 25, 4)
```

```
B = reshape(A, 25, 3)
```

```
A = 1:100;
```

```
B = reshape(A, 10, 10)'
```

```
C = B(:)
```

```
B = reshape(B, 100, 1)
```

```
A = [1 2 3];
```

```
B = repmat(A, 2, 3)
```

```
% Exercise 1: Create vector 1x9 in which the elements alternate between 1 and 0
```

```
A = zeros(1, 9);
```

```
A(1:2:end) = 1
```

```
% Exercise: Create a 10 by 10 matrix in which the elements alternate between 1 and 0
```

```
repmat([1 0; 0 1], 5, 5)
```

```
v = [2 3 4]
```

```
sum(v)
```

```
9
```

```
mean( v )
```

```
3
```

```
Arr = ones(2,3)
```

```
1  1  1
```

```
1  1  1
```

```
sum( Arr ) % summation is along columns
```

```
2  2  2
```

```
mean( Arr )
```

```
1  1  1
```

```
mean( Arr' )
```

```
1  1
```

```
sum( sum( Arr ) ) % sum of all elements of Arr
```

```
6
```

```
sum( ans ) % be careful to use variable ans because it change value after each computation
```

```
6
```

```
sum( Arr, 2 ) % summation across rows
```

```
3
```

```
3
```

```
% Aggregating Functions
```

```
Arr = [2 3 4 5];
```

```
mean(Arr)
```

```
sum(Arr)
```

```
std(Arr)
```

```
min(Arr)
```

```
max(Arr)
```

```
% To compute the sum of all elements
```

```
sum(Arr(:))
```

```
min(Arr)
```

```
min(Arr, [], 2)
```

```
Arr = [5 7 2 3 4 5];
```

```
[value, index] = min(Arr)
```

```
=====
```

```
Matlab has own text editor and files with code have extension .m
```

```
=====
```

```
Homework
```