Subject: Introduction to Computing

Topic: Arithmetic Operations

Teacher: Dr. Ajeet Kumar

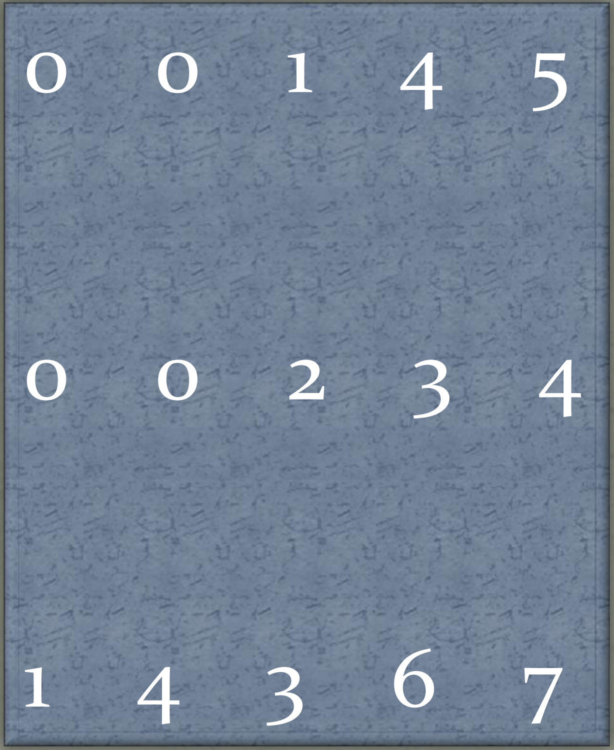
Array

- *Any set of numbers arranged in the rectangular pattern is called an Array*

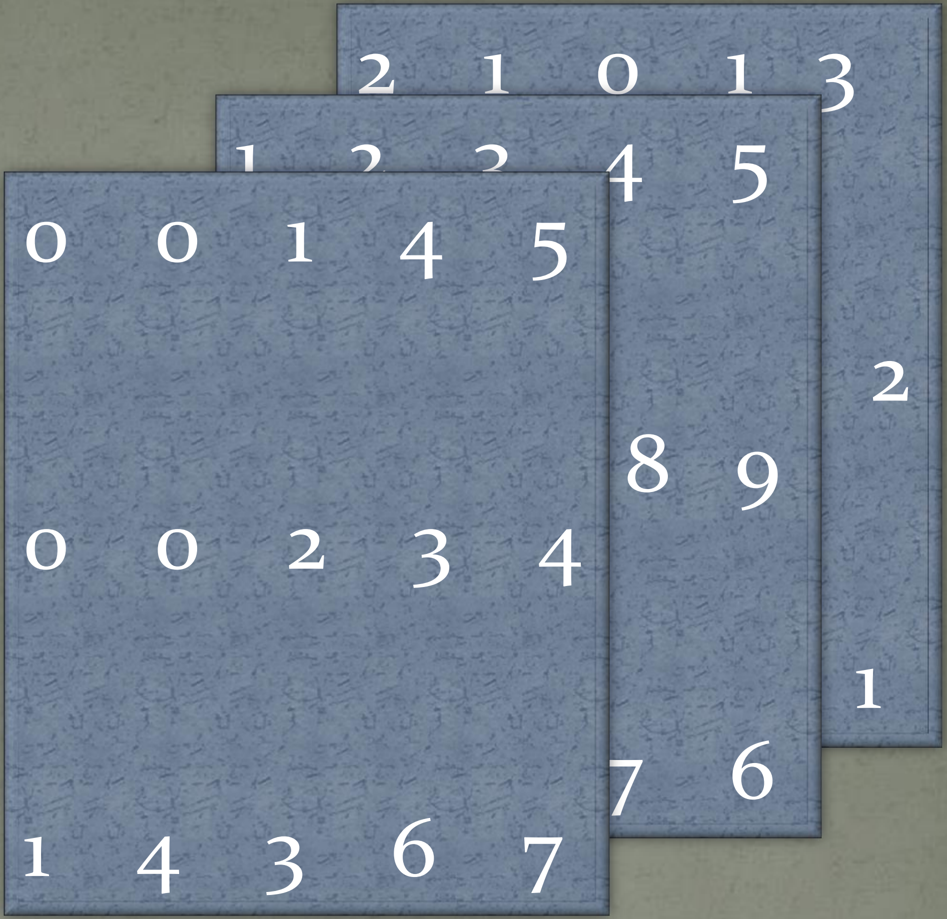
- *Array is a very useful concept to store more than a data with same variable name and used widely in programming*

0	0	1	4	5
0	0	2	3	4
1	4	3	6	7

Array



2-D array



3-D array

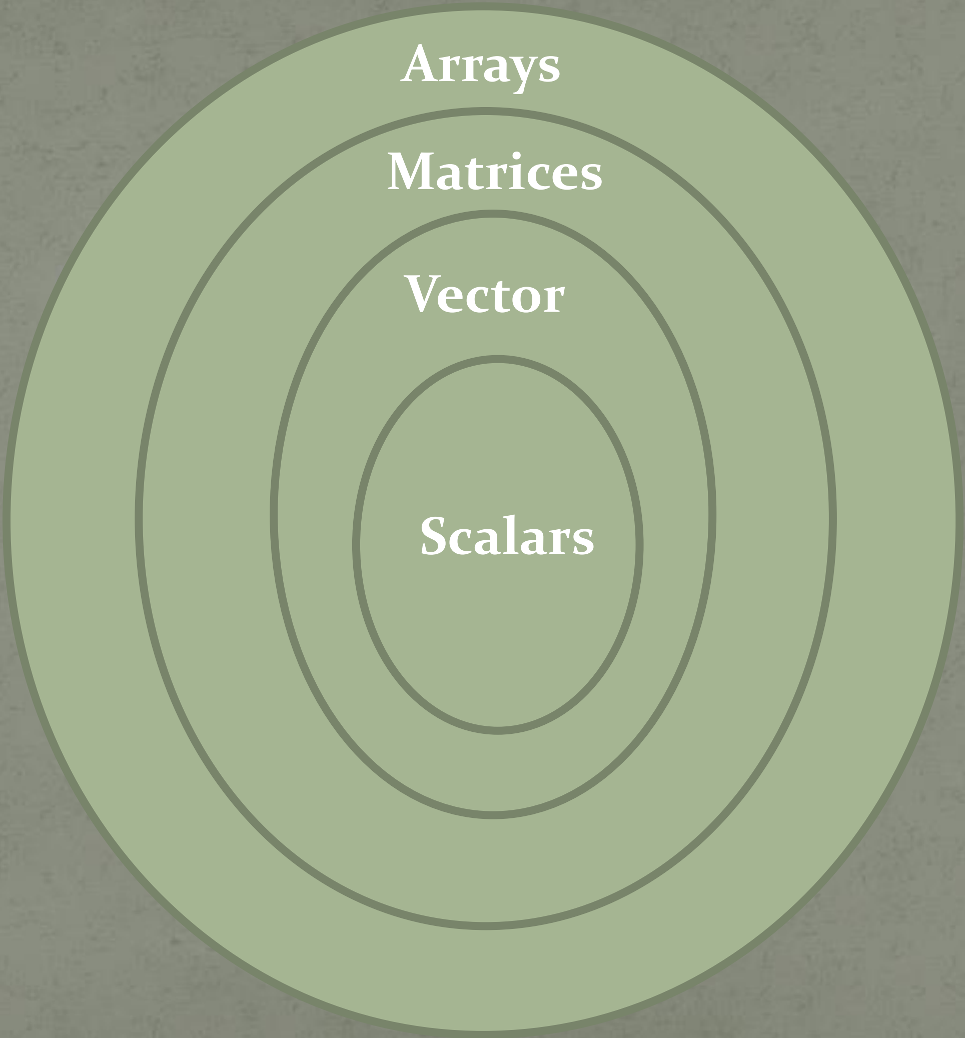
Array

- *Higher-dimensional arrays are not so common*
- *Very frequently used arrays are*

  - *2-D array also called Matrix*
  - *1-D array also called as vector (row/column)*

"Biggest advantage of Matlab is the way it deals with the 2-D array's i.e. Matrices"

Array



Arithmetic Operation

In Matlab there are two classified arithmetic operations

- ***Matrix Operation***
- ***Array Operation***

- ***Matlab defines two different operators to perform these operations which is unusual.***
- ***Operators used to perform Matrix Operations are called Matrix Operator***
- ***Operators used to perform Array Operations are called Array Operator***

Arithmetic Operation

Matrix Operation (Follow rules of usual matrix operation)

Arithmetic Operation	*Matix Operator*
Addition	*A+B*
Subtraction	*A-B*
Multiplication	*A*B*
Division	*A/B*

Arithmetic Operation

Array Operation (Element-by-Element Operation)

Arithmetic Operation	***Array Operator***
Addition	*A+B*
Subtraction	*A-B*
Multiplication	*A.*B*
Right Division	*A./B*
Left Division	*A.\B*
Power	*A.^n*

Arithmetic Operation

Array Operation (Element-by-Element Operation)

Multiplication

$A.*B = a_{ij}*b_{ij}$

```
>> A=[ 1    2;
       3    4];
>>   B =[5    6;
       7    8 ];
>> C=A.*B
        [1*5=5    2*6= 12;
         3*7=21    4*8= 32]
```

Arithmetic Operation

Array Operation (Element-by-Element Operation)

Right Division

$A./B = a_{ij}/b_{ij}$

```
>> A=[ 1    2;
       3    4];
>>  B =[2    4;
       6    8 ];
>> C=A./B
      [1/2=0.5    2/4=0.5;
       3/6=0.5    4/8=0.5]
```

Arithmetic Operation

Array Operation (Element-by-Element Operation)

Left Division

$A.\backslash B = b_{ij}/a_{ij}$

```
>> A=[ 1    2;
       3    4];
>>   B =[2    4;
        6     8];
>> C=A.\B
        [2/1=2    4/2= 2;
         6/3=2    8/4= 2]
```

Arithmetic Operation

Array Operation (Element-by-Element Operation)

Power

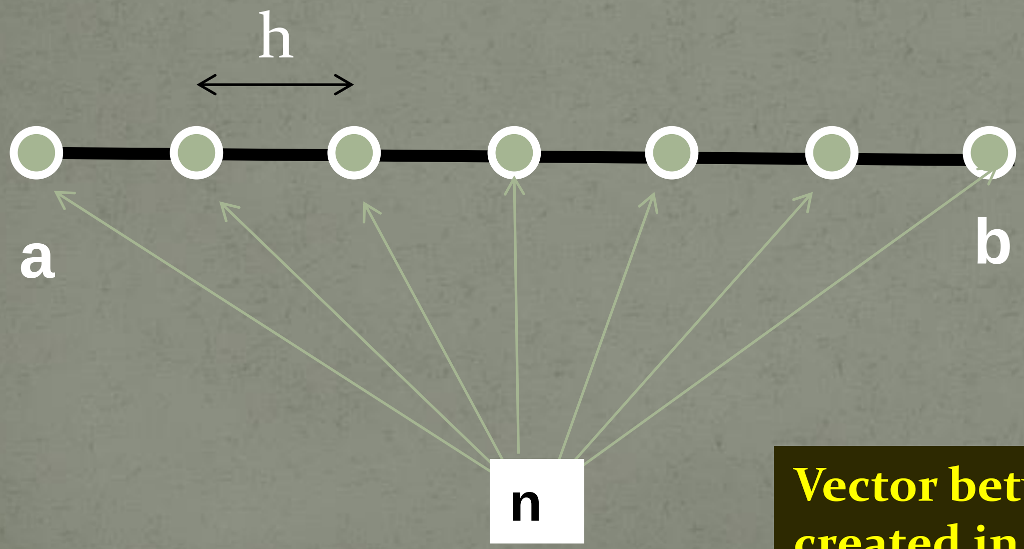
$A.\hat{}n = a_{ij}\hat{}n$

```
>> A=[ 1    2;
       2    4];
>> A.^2 =[1^2=1    2^2=4;
          2^2=4    4^2=16 ];
```

Creating Vectors

Vectors: 1-D array is called vector.

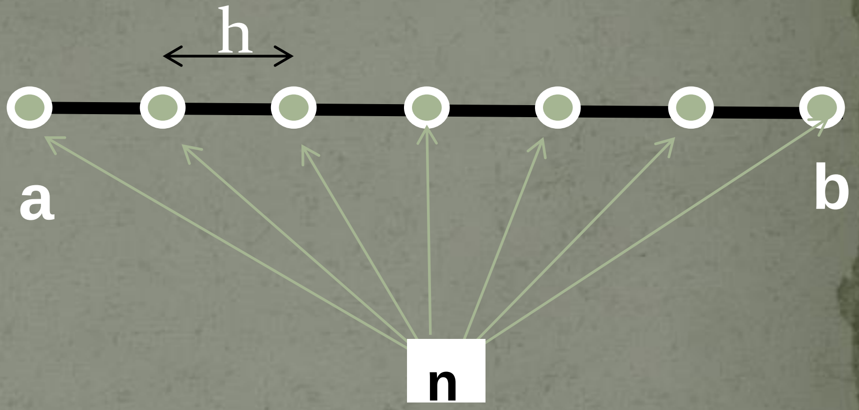
a= starting point
b = end point
n = no. of data
h = step size (interval)



Vector between two numbers can be created in two ways one by changing "*h*" and another by changing "*n*"

Creating Vectors

1. **Fixed spacing (h)**



Syntax

```
>> v=a:h:b
```

a= starting point

h= increment value

b= maximum possible value of the last element of vector "v"

Last element stored in the vector not necessarily same as "b". It could be "b" or smaller than "b"

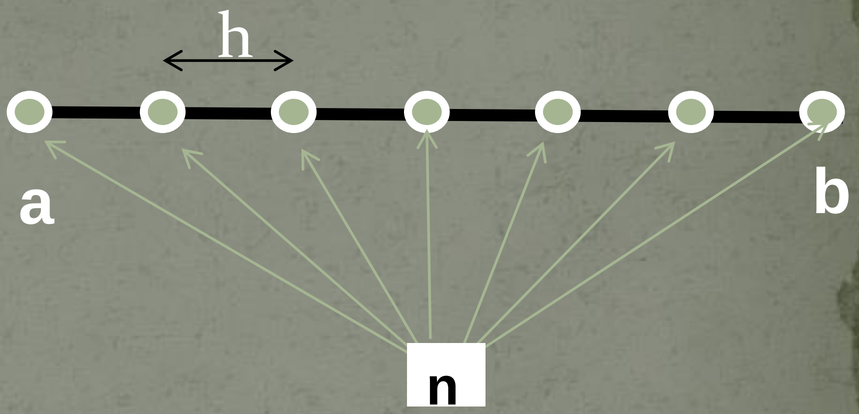
Creating Vectors

```
>> v=1:1:5
v =
    1    2    3    4    5
>> v=1:3:5
v =
    1    4
>> v=1:6                          (default increment is 1)
v =
    1    2    3    4    5    6
```

Generated vector is a row vector

Creating Vectors

2. Fixed no. of points (n)



Syntax

```
>> v=linspace(a,b,n)
```

a= starting point

b= last point

n= total no. of elements of vector

h=(b-a)/(n-1)

Creating Vectors

```
>> v=linspace(1,9,5)
v =

   1    3    5    7    9
>> v=linspace(1,5,4)
v =

   1.0000    2.3333    3.6667    5.0000
```

h=(9-1)/(5-1)=2

h=(5-1)/(4-1)=1.3333

Default value of n =100 taken by Matlab