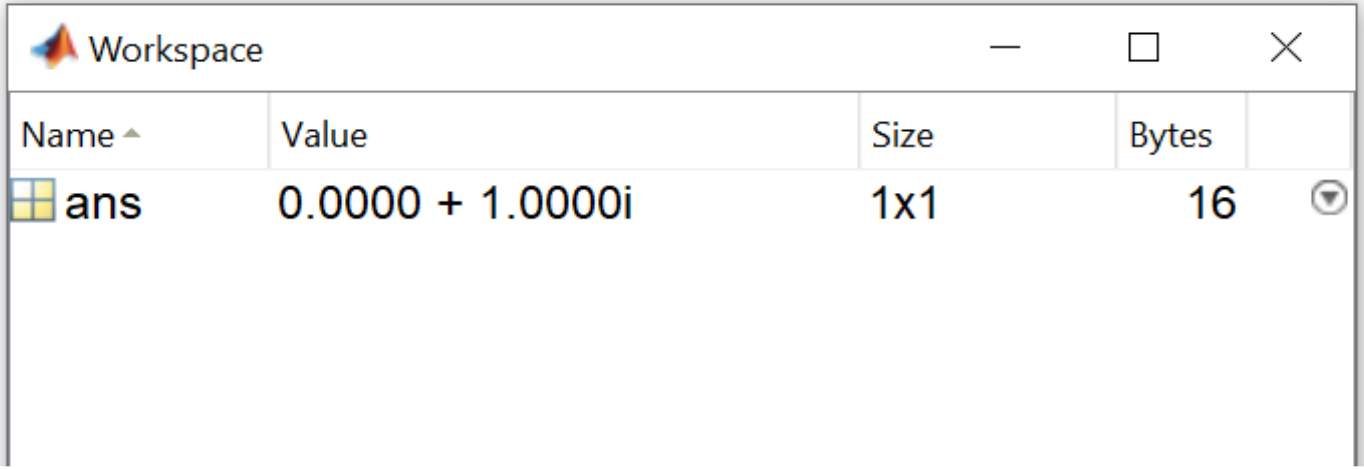
Functions /Operators

1. Parentheses ()
2. Transpose (.'), power (.^), complex conjugate transpose ('), matrix power (^)
3. unary minus (-), logical negation (~)
4. Multiplication (.*), right division (./), left division (.\),
   matrix multiplication (*), matrix right division (/),
   matrix left division (\)
1. Addition (+), subtraction (-)
2. Colon operator (:)
3. Less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), equal to (==), not equal to (~=)
4. Element-wise AND (&)
5. Element-wise OR (|)

Defined variables

```
>> 1/0
ans =
   Inf
>> 0/0
ans =
   NaN
>> pi
ans =
    3.1416
>> log(0)
ans =
  -Inf
>> i
ans =
   0.0000 + 1.0000i
>> j
ans =
   0.0000 + 1.0000i
>>
```



Numeric Types

MATLAB represents floating-point numbers in either double-precision or single-precision format. The default is double precision.

name	**byte**	**min**	**max**
Double	8	-1.79769e+308	1.79769e+308
Single	4	-3.4×10^{38}	3.4×10^{38}

Numeric Types

name	**byte**	**Min**	**max**
uint8	1	0	2^8-1
uint16	2	0	2^{16}-1
uint32	4	0	2^{32}-1
uint64	8	0	2^{64}-1
int8	1	-2^7	2^7-1
int16	2	-2^{15}	2^{15}-1
int32	4	-2^{31}	2^{31}-1
int64	8	-2^{63}	2^{63}-1

Elementary Functions

Elementary Functions
Trigonometric functions

1) sin , returns the sine of argument in radians,

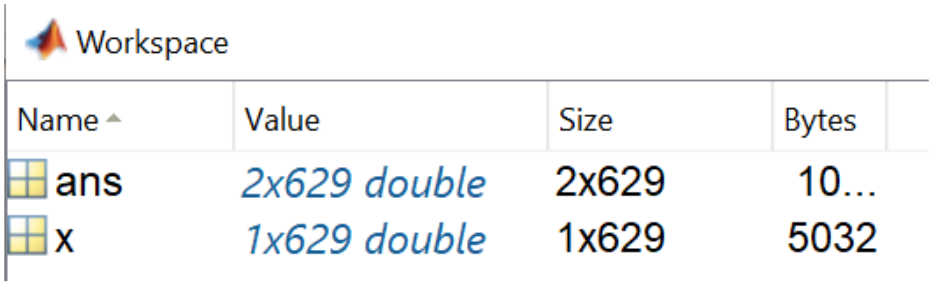
   Y=sin(X),

   returns real values in the interval [-1, 1].

- Example

>>x = -pi : 0.01 : pi;

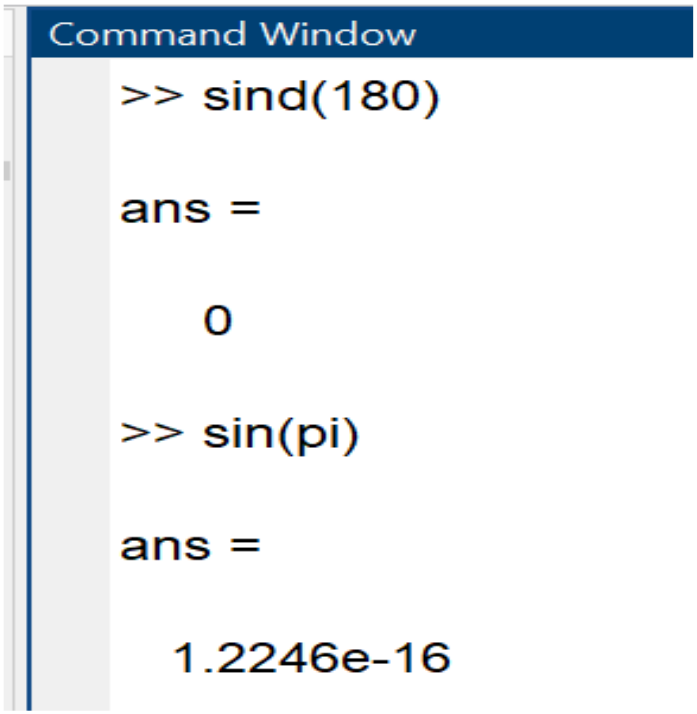
>>[x ; sin(x)]



Elementary Functions
Trigonometric functions

2) sind , returns the sine of argument in degrees,

Example :



Elementary Functions
Trigonometric functions

3) asind , returns the Inverse sine ($\sin^{-1}$) of the elements of X in degrees.

real values of X are in the interval [-1, 1],

asind(X) returns values in

the interval [-90, 90]

Examples:

```
Command Window
>> asind(1)

ans =

   90

>> sind(asind([2 3]))

ans =

   2    3
```

Elementary Functions
Trigonometric functions

4) same for cos, tan, cot

acos, cosd, acosd, ….

5) deg2rad , Convert angles from degrees to radians.

```
>>R = deg2rad(90)
    R = 1.5708
```

6) rad2deg , Convert angles from radians to degrees.

```
>>D = rad2deg(pi)
    D = 180
```

Elementary Functions
exponential functions

- exp , returns the exponential e^x for each element in array X.

Example

Calculate the exponential of 1, which is Euler's number, e.

```
>>exp(1)
    ans = 2.7183
```

Elementary Functions
exponential functions

- log returns the natural logarithm ln(x) of each element in array X.

```
>> y=log(exp(1))
    ans =     1
```

- reallog returns the natural logarithm ln(x) of each element in array X .

  Array X must contain only nonnegative real numbers.

- The size of the result is the same as the size of X.

Elementary Functions
exponential functions

- Log10 returns the common logarithm of each element in array X.

For real values of X in the interval (0, Inf), log10 returns real values in the interval (-Inf ,Inf).

>>log10(10)

ans = 1

The result is 1 since $10^1 = 10$

>>log10(100)

ans = 2

The result is 2 since $10^2 = 100$

>>log10(0)

ans = -Inf

The result is -Inf since $10^{-\infty} = 0$

>>log10(0)

ans = -Inf

The result is -Inf since $10^{-\infty} = 0$

Elementary Functions
exponential functions

- Realsqrt(X) , returns the square root of each element of array X.

  X numbers should be greater than or equal to zero, instead realsqrt returns error messages.

- sqrt , returns the square root of each element of the array X.

  For the elements of X that are negative or complex, sqrt(X) produces complex results.

- The size of the answer is the same as the size of X.

Elementary Functions
exponential functions

- nthroot , returns the real n^{th} root of the elements of X.
- Examples :

```
>> nthroot(8,3)
        ans=2
>>N = [5 3 -1];
>>Y = nthroot(-8,N)
        Y =
          -1.5157   -2.0000   -0.1250
```

Using nthroot to calculate several real roots of -8 : The result is a vector of the same size as N.

- Both X and N must be real scalars or arrays of the same size. If an element in X is negative, then the corresponding element in N must be an odd integer.

Elementary Functions
Complex functions

- z = complex(a,b) Construct complex output data, z, of real and imaginary parts, using two real inputs,

- The complex function provides a useful substitute for expressions, such as a + 1i*b or a + 1j*b, when :
  - b is all zeros
  - i and j are already used

- Examples

  >>z = complex(3)

  Z= 3.0000 + 0.0000i

  >>i=2, Z=3+i, Z=complex (3,1)

Elementary Functions
Complex functions

>>Z= complex(3,5)

- abs (Z) - Absolute value and complex magnitude
- angle (Z) - Phase angle.
- conj(Z) - Complex conjugate.
- imag (Z) - Complex imaginary part.
- real(Z) - Complex real part.

```
ans =
    5.8310
ans =
    1.0304
ans =
   3.0000 - 5.0000i
ans =
     5
ans =
     3
```

Elementary Functions
Rounding and remainder functions

- fix , rounds each element of X to the nearest integer toward zero. This operation effectively truncates the numbers in X to integers by removing the decimal portion of each number

  >>fix(3.5) → ans=3

- round , rounds each element of X to the nearest integer.

  >>round(3.5)→ ans=4

Elementary Functions
Rounding and remainder functions

- floor - Round towards minus infinity.
- ceil - Round towards plus infinity.

```
>> ceil(3.1)
ans =
     4
>> floor(3.9)
ans =
     3
```

Elementary Functions
Rounding and remainder functions

- **r = rem(a,b)** returns the remainder after division of a by b, where a is the dividend and b is the divisor.

  This function is often called the remainder operation, which can be expressed as r = a - b.*fix(a./b).

  r = rem(27,5) → r=2

- **w = mod(a,m)** returns the remainder after division of a by m, where a is the dividend and m is the divisor.

  This function is often called the modulo operation, which can be expressed as w = a - m.*floor(a./m).

  w = mod(27,-5) → w = -3

  W=mod(-27,5) →W = 3

Elementary Functions
Discrete math functions

- Discrete math functions perform operations on integers (…, -2, -1, 0, 1, 2, …), or return discrete output as integers.
- You can use these functions to factor large numbers, calculate factorials, find permutations and combinations, or find greatest common denominators.

Elementary Functions
Discrete math functions

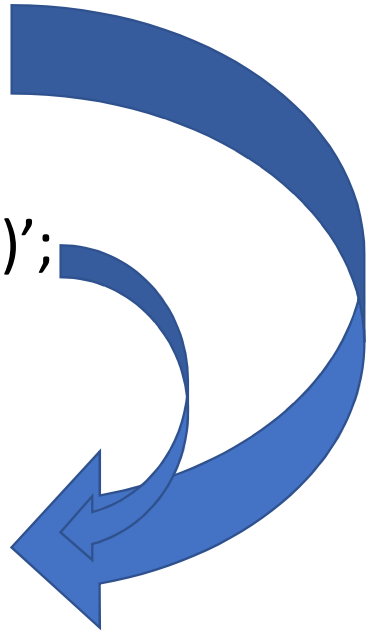
factor	Prime factors
factorial	Factorial of input
gcd	Greatest common divisor
isprime	Determine which array elements are prime
lcm	Least common multiple
nchoosek	Binomial coefficient or all combinations
perms	All possible permutations
matchpairs	Solve linear assignment problem
primes	Prime numbers less than or equal to input value
rat	Rational fraction approximation
rats	Rational output

Elementary Functions
Statistics (sum)

```
>>X=0:10:100;
>>sum(X)

>>X=(0:10:100)';
>>sum(X)

ans=
 550
```



```
>>M=rand(3,5)
    0.14    0.79    0.04    0.68    0.39
    0.42    0.96    0.85    0.76    0.66
    0.92    0.66    0.93    0.74    0.17
>>sum(M)
    1.48    2.41    1.82    2.18    1.22
>>sum(M,1)

>>sum(M,2)
    2.04
    3.64
    3.42
```

Elementary Functions Statistics

- sum (M)
- sum (M,1)
- sum(M,2)

- mean(X)
- mean (X,1)
- mean(X,2)

- prod (X)
- prod (X,1)
- prod(X,2)

M=rand(3,5)

0.14	0.79	0.04	0.68	0.39
0.42	0.96	0.85	0.76	0.66
0.92	0.66	0.93	0.74	0.17

- sum(sum(M))
- all(prod(prod(M))==prod(M(:)))
- mean(mean(M))==mean(M(:))

- ans =1 (logical)

Elementary Functions Statistics

X =randi(10,1,5)%X=rand(5,1)

X = 8.00 1.00 3.00 1.00 1.00

M =randi(10,5,5)

M = 9.00 5.00 5.00 3.00 5.00
7.00 4.00 5.00 7.00 10.00
4.00 8.00 7.00 7.00 4.00
10.00 8.00 8.00 2.00 6.00
1.00 2.00 8.00 2.00 3.00

Same for max

min (X)	**ans= 1**
min(M) %by column	ans = 1.00 2.00 5.00 2.00 3.00
~~min(M,1)~~ ~~min(M,2)~~	
[val,pos]=min(M) val = 1.00 2.00 5.00 2.00 3.00 pos = 5.00 5.00 1.00 4.00 5.00 Pos = the row index that corresponds to the minimum value of M on each column. (according to the operating dimension)	

Elementary Functions
Statistics

- variance

V = var(A)

returns the variance of the elements of A :

If A is a vector of observations, the variance is a scalar.

If A is a matrix, V is a row vector containing the variances corresponding to each column.

<u>Mathematically:</u>

$$V = \frac{1}{N-1}\sum_{i=1}^{N}|A_i - \mu|^2 \qquad \mu = \frac{1}{N}\sum_{i=1}^{N}A_i.$$

Elementary Functions
Statistics

- Standard deviation

S = std(A)

returns the standard deviation of the elements of A.

If A is a vector of observations, then the standard deviation is a scalar.

If A is a matrix then S is a row vector containing the standard deviations corresponding to each column.

Mathematically:

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|A_i - \mu|^2}, \qquad \mu = \frac{1}{N}\sum_{i=1}^{N}A_i.$$

Elementary Functions
Statistics

- **R= sort(X)**
- For X a vector, it sorts the elements of X in ascending order.
- for matrices, it sorts each column of X in ascending order.

Example
m=[28 3 5 -10 0 6 4 3];

```
>> sort(m)
ans =
-10 0 3 3 5 6 28
```

```
>> sort(m,'descend')
ans =
28 6 5 3 3 0 -10
```

```
>>[n ind]=sort(m)
n =
-10 0 3 3 5 6 28
ind =
4 5 2 7 3 6 1
```

```
A = [3 6 5; 7 -2 4; 1 0 -9]
A=
   3    6    5
   7   -2    4
   1    0   -9
```

```
>>sort(A)
  ans =
       1.00   -2.00   -9.00
       3.00       0    4.00
       7.00    6.00    5.00
```

```
[n,ind]=sort(A)
n =
       1.00   -2.00   -9.00
       3.00       0    4.00
       7.00    6.00    5.00

ind =

       3.00    2.00    3.00
       1.00    3.00    2.00
       2.00    1.00    1.00
```

Elementary Functions Algebria/shape

- reshape (M,row?,column?)

  Example

  A=1:10

  reshape(A,5,2)

  reshape(A,[5,2])

ans =

1.00	6.00
2.00	7.00
3.00	8.00
4.00	9.00
5.00	10.00

- Repmat (M,rows, columns)

  Example

  M=1:4

  A=M’

  repmat(M,3,2)

  ans =

1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00
1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00
1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00

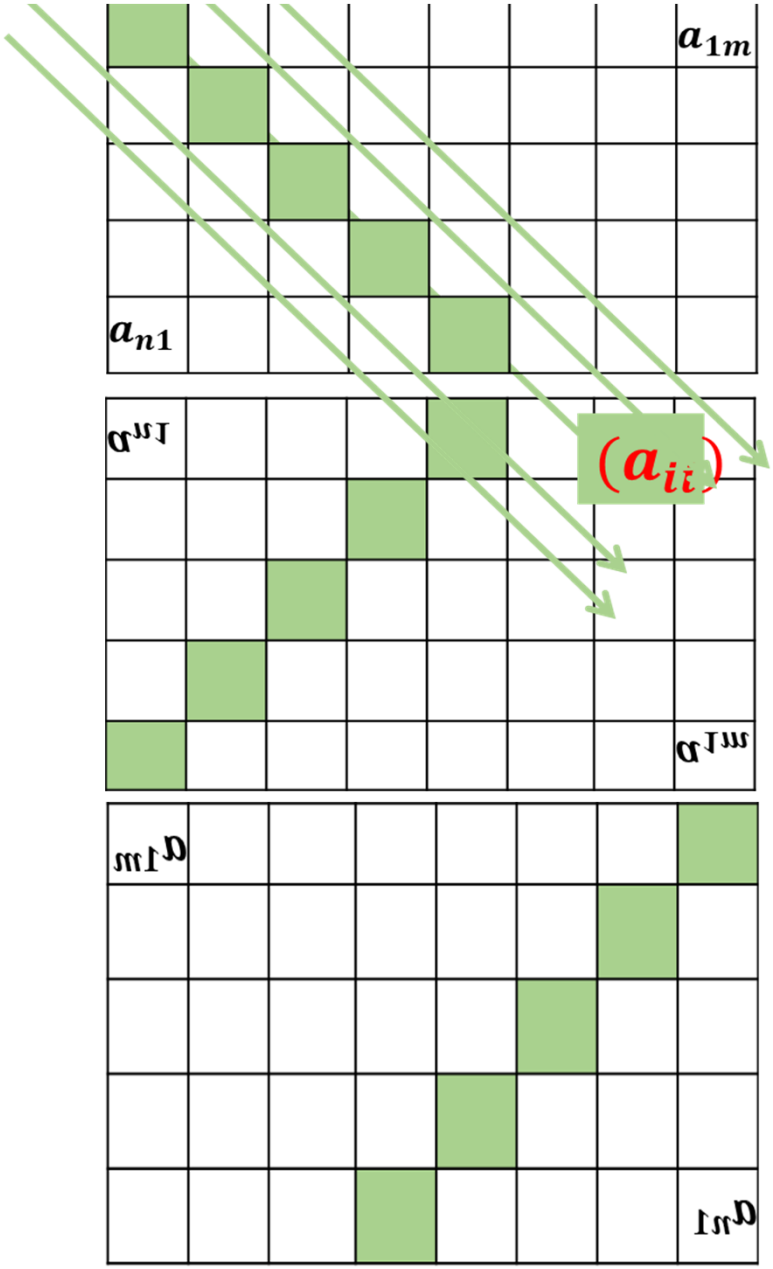
Repmat(A,3,2)

ans =

1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00
1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00
1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00

Elementary Functions Algebria/shape

- diag(A)
- diag(A,k) %k>0 ou k<0
- Flipud
- fliplr



Elementary Functions Algebria/shape

- sub2ind (size, row, column)
- ind2sub (size, ind)

Example

A=rand(3,2)

```
A =   0.75   0.70
      0.26   0.89
      0.51   0.96
```

I=Find(A>0.5)

```
I =   1.00
      3.00
      4.00
      5.00
      6.00
```

[i,j]=ind2sub(size(A),I)

```
i =
      1.00
      3.00
      1.00
      2.00
      3.00
j =
      1.00
      1.00
      2.00
      2.00
      2.00
```

sub2ind(size(A),i,j)

```
ans =
      1.00
      3.00
      4.00
      5.00
      6.00
```

[i,j]=find(A>0.5)

```
i =
      1.00
      3.00
      1.00
      2.00
      3.00
j =
      1.00
      1.00
      2.00
      2.00
      2.00
```

Format command

- MATLAB always attempts to display integers (whole numbers) exactly. However, if the integer is too large, it is displayed in scientific notation with five significant digits,

Example:

first enter 123456789 at the command prompt, and then 1234567890.

```
>> 123456789

ans =

   123456789

>> 1234567890

ans =

   1.2346e+09
```

Format command

- Numbers with decimal parts are displayed with four significant digits. This is what is called the default format, i.e. what normally happens.
- However, you can change from the default with variations on the format command, as follows.
- Also, All output from subsequent display statements will be in

examples

>> format short e % scientific notation (floating point form)

>> format long e %for more accurate output.A scientific notation, but with 15 significant digits

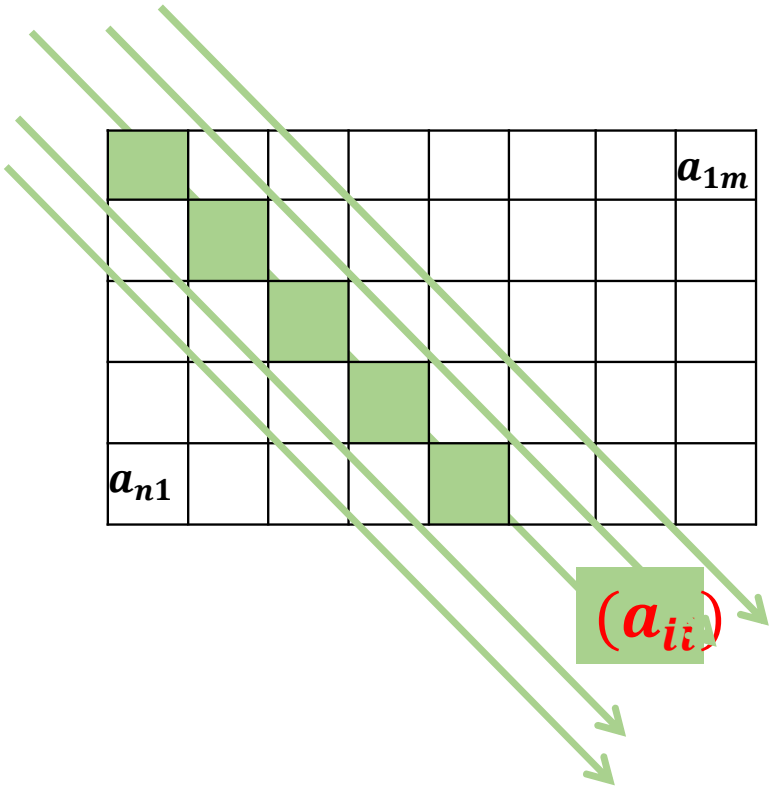
Format command

```
>> format short e
>> 1/7
ans =
  1.4286e-01

>> format long e
>> 1/7
ans =
   1.428571428571428e-01

>> format short
>> 1/7
ans =
```

- Gardé en cas ou



Format command for me

Numbers with decimal parts are displayed with four significant digits. If the value x is in the range $0{:}001 < x \leq 1000$ it is displayed in fixed point form, otherwise scientific (floating point) notation is used, in which case the mantissa is between 1and 9.9999, e.g. 1000.1 is displayed as 1.0001e+003. Check this by entering following numbers at the prompt (on separate lines): 0.0011, 0.0009, 1/3, 5/3, 2999/3, 3001/3

Elementary Functions
Algebria/shape

- [row,col] = ind2sub(sz,ind) returns the arrays row and col containing the equivalent row and column subscripts corresponding to the linear indices ind for a matrix of size sz. Here sz is a vector with two elements, where sz(1) specifies the number of rows and sz(2) specifies the number of columns.