

## Predefined MATLAB Functions

Arithmetic expressions often required computations after than addition, subtraction, multiplication, division, and exponentiation. For example many expressions required use of logarithms, exponentials and trigonometric functions. MATLAB includes a built-in functions library of these useful functions. For example, if we want to compute the square root of x and store the result in b, we can use following commands:

```
b=sqrt(x)
```

This statement is valid if the x is scalar or matrix. if x is a matrix, the function will be applied element by element to the values in the matrix.

### 1-abs(x)

**Compute the absolute value of x, |x|**

```
>> x=-1
```

```
x =
```

```
-1
```

```
>> abs(x)
```

```
ans =
```

```
1
```

```
>> abs(-20)
```

```
ans =
```

```
20
```

```
>> abs(4)
```

```
ans =
```

```
4
```

```
>> y=[2 -1 0 -6]
```

```
y =
```

```
2 -1 0 -6
```

```
>> abs(y)
```

```
ans =
```

```
2 1 0 6
```

### 2-sqrt

**Compute the square root of x**

```
>> x=4
```

```
x =
```

```
4
```

```
>> sqrt(x)
```

```
ans =
```

```
2
```

```
>> sqrt(85)
```

```
ans =
```

```
9.2195
```

```
>> y=[3 1 -4]
```

```
y =
```

```
3 1 -4
```

```
>> sqrt(y)
```

```
ans =
```

```
1.7321 1.0000 0 + 2.0000i
```

### **3-round (x)**

**Round x to the nearest integer.**

```
>> x=3.1
```

```
x =
```

```
3.1000
```

```
>> round(x)
```

```
ans =
```

```
3
```

```
>> x=5.5
```

```
x =
```

```
5.5000
```

```
>> round(x)
```

```
ans =
```

```
6
```

```
>> y=[0 4.900 3.100 6.7]
```

```
y =  
    0  4.9000  3.1000  6.7000  
>> round(y)
```

```
ans =  
  
    0    5    3    7
```

```
>> r=round(5.7)  
r =  
    6
```

#### **4-fix(x)**

**Rounds (or truncates) x to the nearest integer toward zero.**

```
>> fix(3.6)  
ans =  
    3
```

```
>> x=2.1  
x =  
    2.1000
```

```
>> fix(x)  
ans =  
    2
```

```
>> y=[0 4.9 3.1 6.7]  
y =  
    0  4.9000  3.1000  6.7000
```

```
>> fix(y)  
ans =  
    0    4    3    6
```

#### **5-floor(x)**

**Rounds to nearest integer floor-infinity.**

```
y=[5.8 -2.4 -8.5 0.2]
```

```
y =  
    5.8000 -2.4000 -8.5000  0.2000
```

```
>> floor(y)  
ans =
```

```
    5   -3   -9    0
```

### 6-ceil(x)

**Rounds x to the nearest integer toward + infinity**

```
>> ceil(3.6)
```

```
ans =  
    4
```

```
>> ceil(-1.9)  
ans =  
   -1
```

```
>> ceil(-0.6)  
ans =  
    0
```

```
>> ceil(6.7)  
ans =  
    7
```

```
>> y=[5.8 -2.4 -8.5 0.2]
```

```
y =  
    5.8000 -2.4000 -8.5000  0.2000
```

```
>> ceil(y)  
ans =
```

```
    6   -2   -8    1
```

### 7-rem (x,y)

**Compute the remainder of x/y.**

```
>> rem(4,2)  
ans =
```

```

0
>> rem(10,3)
ans =
    1
>> rem(2,4)
ans =
    2

>> rem(8,5)
ans =
    3

>> y=[2 5 1 0]
y =
     2     5     1     0

>> rem(y,2)
ans =
     0     1     1     0

```

### 8-exp(x)

Compute the value of  $e^x$ , where  $e$  the base of natural logarithms approximately 2.718282

We can represent the function  $2 \cdot e^{-1}$  MATLAB

```

>> 2*exp(-1)
ans =
    0.7358

```

### 9- log(x)

Computes the  $\ln(x)$ , the natural logarithm of  $x$  to the base  $e$ .

```

>> log(3)
ans =
    1.0986

```

```

>> x=-1
x =
    -1

```

```
>> log(x)
ans =
    0 + 3.1416i
```

### **10-log10(10)**

```
>> log10(10)
ans =
    1
```

```
>> log10(-10)
ans =
    1.0    + 1.3644i
```

```
>> log10(2)
ans =
    0.3010
```

Also we can use the logarithm for base 2

```
>> log2(2)
ans =
    1
```

```
>> log2(10)
ans =
    3.3219
```

Example: if we want to represent this equation in MATLAB  $y = 3 * \sqrt{x}$

```
>> x=3;
>> y=3*sqrt(x)
y =
    5.1962
```