

COMMAND HISTORY RECORD FROM AUGUST 13

READ CHAPTERS 2 AND 3

MAKE SURE YOU UNDERSTAND EACH COMMAND AND RESULT BELOW...
ASK IF YOU DON'T

>> a = [1 2 3; 4 5 6]

a =

1 2 3
4 5 6

>> a = a'

a =

1 4
2 5
3 6

>> a = a(1,2)

a =

4

>> b = [1 2 3; 4 5 6];

>> c = b(2,:)

c =

4 5 6

>> c = b(1,2:3)

c =

2 3

>> c = b(1,1:2:3)

c =

```
1 3
```

```
>> c = b(1,:);  
>> d = [b; c]
```

```
d =
```

```
1 2 3  
4 5 6  
1 2 3
```

```
>> d(2,:) = [7 8 9]
```

```
d =
```

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

```
>> d(2,:) = []
```

```
d =
```

```
1 2 3  
1 2 3
```

```
>> % linspace = linearly spaced values (start, end, number of values)  
>> a = linspace(0,3,8)
```

```
a =
```

```
Columns 1 through 6
```

```
0 0.4286 0.8571 1.2857 1.7143 2.1429
```

```
Columns 7 through 8
```

```
2.5714 3.0000
```

```
>> % logspace = logarithmically spaced values (10^start, 10^end, number of values)  
>> a = logspace(0,3,8)
```

```
a =
```

1.0e+03 *

Columns 1 through 6

0.0010 0.0027 0.0072 0.0193 0.0518 0.1389

Columns 7 through 8

0.3728 1.0000

>> help logspace

logspace Logarithmically spaced vector.

logspace(X1, X2) generates a row vector of 50 logarithmically equally spaced points between decades 10^{X1} and 10^{X2} . If X2 is pi, then the points are between 10^{X1} and pi.

logspace(X1, X2, N) generates N points.

For N = 1, logspace returns 10^{X2} .

Class support for inputs X1,X2:

float: double, single

See also linspace, colon.

Reference page for logspace

>> % standard function rand for returning random numbers

>> % actually pseudorandom numbers because not perfectly random

>> % rand by itself returns value 0-1

>> a = rand

a =

0.9575

>> a = rand

a =

0.9649

>> a = rand

a =

0.1576

```
>> a = 5 * rand
```

a =

4.8530

```
>> rand(4)
```

ans =

| | | | |
|--------|--------|--------|--------|
| 0.9572 | 0.4218 | 0.6557 | 0.6787 |
| 0.4854 | 0.9157 | 0.0357 | 0.7577 |
| 0.8003 | 0.7922 | 0.8491 | 0.7431 |
| 0.1419 | 0.9595 | 0.9340 | 0.3922 |

```
>> randi
```

Error using randi

Not enough input arguments.

```
>> randi([1 50],1)
```

ans =

33

```
>> randi([1 50],1)
```

ans =

9

```
>> randi([1 50],3)
```

ans =

| | | |
|----|----|----|
| 36 | 3 | 35 |
| 2 | 5 | 16 |
| 14 | 42 | 48 |

```
>> % the standard function rng (random number generator) is used
```

```
>> % to set the "seed" and algorithm used to generate the
```

```
>> % pseudorandom number sequence
```

```
>> rng('default') % sets to what you get when your first start Matlab
>> rand
```

```
ans =
```

```
0.8147
```

```
>> rand
```

```
ans =
```

```
0.9058
```

```
>> rng('default') % sometimes you want the same sequence again to test code
>> rand
```

```
ans =
```

```
0.8147
```

```
>> rand
```

```
ans =
```

```
0.9058
```

```
>> rng(sum(clock),'twister') % use current time as seed, choose twister algorithm
>> rand
```

```
ans =
```

```
0.0853
```

```
>> rng(sum(clock),'twister') % value of clock changes every second, so get new seed
>> rand
```

```
ans =
```

```
0.6128
```

```
>> a = [1 2 3];
>> b = [4 5 6];
>> c = a ./ b
```

```
c =
```

```
0.2500 0.4000 0.5000
```

```
>> d = a .* b
```

```
d =
```

```
4 10 18
```

```
>> a = input('enter integer value')
```

```
enter integer value4
```

```
a =
```

```
4
```

```
>> a = input('enter integer value: ')
```

```
enter integer value: 4
```

```
a =
```

```
4
```

```
>> % do not use input in homework!
```

```
>> % does not work with Publish
```

```
>> % see ReactorLab.net > Resources > Matlab > CENG 15 - NANO 15
```

```
>> % for info on p_input
```

```
>> a = p_input('enter a: ')
```

```
enter a: 56
```

```
a =
```

```
56
```

```
>> % fprintf = file print formatted
```

```
>> % the default "file" is the screen, other files are disk files, etc.
```

```
>> fprintf('my variable a = %g OK? \n',a)
```

```
my variable a = 56 OK?
```

```
>> % search documentation for fprintf and see format specifications, e.g., %g here
```

```
>> % sprintf = string print formatted
```

```
>> % sprintf returns a string which you can assign to a variable
```

```
>> tt = sprintf('my variable a = %g OK? \n',a) %
```

```
tt =
```

```
my variable a = 56 OK?
```

```
>> % can use sprintf with plot titles to  
>> % display current input values used in plot  
>> a = 2;  
>> b = linspace(0,a*pi,100);  
>> plot(b,sin(b))  
>> ylabel('sin(b)')  
>> xlabel('b')  
>> tt = sprintf('sine from 0 to %g pi',a);  
>> title(tt)
```

```
>> % change value of a and replot  
>> a = 3;  
>> b = linspace(0,a*pi,100);  
>> plot(b,sin(b))  
>> tt = sprintf('sine from 0 to %g pi',a);  
>> title(tt)  
>>
```