Matrices
Enter elements in a row using brackets, [], and either spaces or commas:

```matlab
>> row1 = [4 8 2]
row1 =
    4     8     2
```

```matlab
>> row2 = [4,8,2]
row2 =
    4     8     2
```

Enter elements in a column using brackets, [], and semicolons:

```matlab
>> column1 = [4;8;2]
column1 =
    4
    8
    2
```

Combine to form a matrix:

```matlab
>> mat1 = [3 4 5; 5 12 13; 7 24 25]
mat1 =
    3     4     5
    5    12    13
    7    24    25
```

Another option is to separate rows with “Enter” as you type in the matrix entries:

```matlab
>> mat2 = [1 2 3
4 5 6
7 8 9]
mat2 =
    1     2     3
    4     5     6
    7     8     9
```
Some special built-in matrices:

- **eye**: identity matrix (Ex: `eye(4)` is the $4 \times 4$ identity matrix)

- **ones**: matrix filled with ones (Ex: `ones(3,5)` is the $3 \times 5$ ones matrix)

- **zeros**: matrix filled with zeros

- **rand**: matrix filled with random numbers uniformly distributed in the interval $(0, 1)$

Try some of these on MATLAB to get a feel for entering matrices.

One piece of notation: to insert a comment after a command, use the `%` sign. MATLAB will read anything after the `%` as a comment, not a command to be executed. This notation appears below.

One more matrix you can create:

- **diag**: diagonal matrix

```
>> diag([2 1 5])
ans =
     2     0     0
     0     1     0
     0     0     5
```

```
>> diag(1:3)  % 1:3 means to count from 1 to 3, in units of 1
ans =
     1     0     0
     0     2     0
     0     0     3
```
The `diag` command can also extract diagonal elements from a matrix:

```matlab
>> Mat1=[10 9 8; 6 4 1; -2 -4 -7];
>> diag(Mat1)
ans =
    10
    4
   -7
```

Subscript notation with matrices (the equivalent of referring to, say, the $a_{32}$ element of a matrix):

```matlab
>> A = [1 2 3; 4 5 6; 7 8 9];
>> A(3,2)  % gives row 3, column 2 entry of A
ans =
    8
```

You can redefine matrix elements using subscript notation (for example, redefining the $a_{11}$ element to be 6):

```matlab
>> A(1,1)=6
A =
    6    2    3
    4    5    6
    7    8    9
```

Math operations with matrices:

```matlab
>> u = [1 2 3];
>> v = [7 5 9];
>> u + v  % addition is element-by-element
ans =
    8    7   12

>> u - v  % subtraction is element-by-element
ans =
   -6   -3   -6
>> C=diag(4:6); D=ones(3); % ones(3) = ones(3,3)
>> Mat2=C*D
Mat2 =
    4     4     4
    5     5     5
    6     6     6

>> Mat2' % ' gives the matrix transpose
ans =
    4     5     6
    4     5     6
    4     5     6

Concatenation

To combine matrices or vectors of appropriate sizes, you can simply concatenate them. For example, to join Mat2, above, with column1 from the first page of this document, and call it newmatrix, type

>> newmatrix = [Mat2 column1]
newmatrix =
    4     4     4     4
    5     5     5     8
    6     6     6     2

Or, to join row1 with Mat2, and call it newmatrix2, type

>> newmatrix2 = [row1; Mat2]
newmatrix2 =
    4     8     2
    4     4     4
    5     5     5
    6     6     6