Matlab Basics: Day 1
EDGE 2019

You should have the "Command Window" open in front of you.

1. **How to declare and manipulate vectors and matrices.**

   To declare a variable to equal a number just set it equal to that number. So to set the variable "x" equal to 47 type:

   ```
   x = 47
   ```

   followed by "enter". You should see:

   ```
   x =

       47
   ```

   which means that Matlab thinks that x is now equal to 47. (From now on, it is implied that you type "enter" after a given command.) If you type a semi-colon after the command before typing "enter", you won't see any output, but x will still equal 47. Try it:

   ```
   x=4747;
   ```

   Now see what Matlab thinks x is:

   ```
   x
   ```

   and you should see

   ```
   x =

           4747
   ```

   To declare a matrix, put the numbers in square brackets, and separate rows by semi-colons. You may separate entries by commas for legibility, but that isn't necessary.

   ```
   A = [1 2 3; 4 5 6; 7 8 9];
   ```

   You should see that A is a 3×3 matrix. Type

```
  A
```

to get

```
A =

     1     2     3
     4     5     6
     7     8     9
```

Multiplication is with an asterisk:

```
  47 * 47

ans =

        2209
```

or to multiply matrices:

```
 A * A

ans =

    30    36    42
    66    81    96
   102   126   150
```

Suppose you have two vectors:

```
x = [1 2 3];
y = [4 5 6];
```

and you want to multiply each element of x by the corresponding element in y. You can do "component-by-component" multiplication using the '.*' command:

```
 x.*y

ans =

     4    10    18
```

The same goes for division or exponentiation, which is done with a carat ^ :

```
 y.^x

ans =

     4    25   216
```

You can also invert matrices, find their determinants and eigenvalues:

```
>> A = [4 2; 1 3]

A =

     4     2
     1     3

>> inv(A)

ans =

    0.3000   -0.2000
   -0.1000    0.4000

>> det(A)

ans =

    10

>> eig(A)

ans =

     5
     2
```

To get the $n \times n$ identity matrix, use eye(n):

```
>>eye(3)

ans =

    1     0     0
    0     1     0
    0     0     1
```

Similarly, zeroes(n) and ones(n) will give you, respectively, an $n \times n$ matrix of zeroes and ones.