Matlab Level 5: Loops

Many times you may want to run a series of commands. To do this, you will need to use the ‘for’ command.

>> help for

The ‘for’ loop requires that you specify how many times the loop should repeat. When doing this, you create a new variable within the ‘for’ function. Good programming skills include using unique variables for the ‘for’ loop.

Create a loop that repeats 5 times. Within the loop, add 1 to a variable for each loop. Start by clearing the workspace.

>> clear
>> a = 1;
>> for n = 1:5
>> a = a + 1
>> end

You can also place loops within loops. This can be a useful programming skill. Create two loops, let the inner loop repeat 5 times, the outer loop 2 times. Add 1 to a variable each time the inner loop repeats, reset the variable each time the outer loop repeats.

>> a = 1;
>> for n1 = 1:2
>> for n2 = 1:5
>> a = a + 1;
>> end %this is the end the inner loop
>> a = 1;
>> end %this ends the outer loop

Note: the ‘%’ indicates a comment. MATLAB does not process anything written after the ‘%’ symbol.

You can also use the ‘while’ function for loops. This loop will continue to repeat while a particular statement is true.

>> help while
>> a = 0;
>> while a < 10
a = a + 1
end
Try these problems.

1. Write commands that prints character arrays from ‘s1c1l1’ to ‘s3c5t3’ the screen (this will take three loops).
   Hint: look up ‘int2str’ function.

2. Using loops, write commands that result in the following array:

   \[
   \begin{array}{ccc}
   1 & 2 & 3 \\
   \text{Total} =& 2 & 3 & 4 & 7 \\
   & 3 & 4 & 5 & 8 \\
   \end{array}
   \]

   Hint: The outer loop repeats three times and controls row values. The inner loop repeats four times and controls column number.

3. Create two vectors:
   \[
   A = [1 2 3 4 5 6 7 8 9 0]; \\
   B = [2 2 3 3 5 5 6 6 9 0];
   \]

   Using loops, compare A and B position by position and identify the position number that A and B are equal.
   Save the position numbers in a variable called ‘equalvalue’. The final value of equalvalue should be: [2 3 5 9 10].

4. Use the ‘rand’ function to calculate the following
   a. The number of random numbers between 0 and 1 that are needed to add up to 20 (or more).
   b. The number of random numbers observed until a number between 0.2 and 0.5 is observed.

5. Use the ‘input’ function and generate a loop in which the user is asked to enter a speed in mph and the program returns the speed in m/s. The loop should continue until the user enters the value ‘0’. Use the ‘fprintf(1,’message’) to print specific messages to the user (e.g., ‘enter 0 to exit’, ‘m/s’).