

Topics: iteration using **for**, vectorized code, user defined function

Reading (ML): Sec 4.2–4.5 (exclude 4.2.2), revisit Sec 2.8.2 (exclude matrix operations), Sec 5.0, 5.1.

General form of the for Loop

```
for index = expression
    Statements to execute
    Also called loop body
end
```

Expression usually takes the form of a vector. E.g., **1:n**.

Two patterns for doing something n times

<pre>for i = 1:n % do something % ... end</pre>	<pre>i = 1; while i<=n % do something % ... i = i + 1; end</pre>
---	---

Example 1: Average

Write a program that prompts the user for 10 numbers and then print the average. Use a **for** loop and store all user input in a vector.

Vectorized Code

MATLAB can perform operations on *more than one value* (variable) at a time. Program segments that show this feature are said to be *vectorized*.

E.g., let **a** and **b** be vectors of equal length. One can add these two vectors such that $c(i) = a(i) + b(i)$ for all indices i . In most programming languages (including Java), one must perform this add operation on each element of the vector individually. In MATLAB, the add operation can be performed on the entire vector at the same time.

Code with a loop	Vectorized code
<pre>n = length(a); c = zeros(1,n); for i = 1:n c(i) = a(i) + b(i); end</pre>	<pre>c = a + b;</pre>

Mathematical and logical operations that can be used in vectorized code include

`+` `-` `.*` `./` `.^` `==` `>` `<` `~=` `&` `|`

Example: Are they prime?

Write a program that saves in a vector all the prime numbers in the range of $[2, n]$, $n > 1$.

Script file **savePrime.m**:

```
% Save prime numbers in [2,n] to vector prime

n = input('Enter number: ');
prime = 2; % vector to store prime #s
i = 3;     % next number to be checked
while (i<=n)
    % check number i, save if prime

    % go to next number
    i = i+1;
end
prime
```

User-Defined Function

- Can easily “reuse” code
- Functions can be independently tested
- Upon invocation, each function has its own memory space inaccessible by other functions or the command window space—variables in a function can be “seen” only inside the function
- Values stored in variables are not preserved between function calls.
- Arguments are “passed by value”