

In this assignment you will learn about proper ways to use functions and create function files. In the last lab we created a basic script file.

Script files are different than function files, which can be called in the command window, or in another script file. This can sometimes be confusing since both are saved as `.m` files. There are two main ways in which we can define a function in MATLAB, either as an anonymous function, or as a function file.

Anonymous Functions

An **anonymous function** is a function that is not stored in a program file, but is associated with a variable whose data type is `function_handle`. Anonymous functions can accept inputs and return outputs, just as standard functions do. However, they can contain only a single executable statement.

For example, to define the function $f(x) = x^2 + 1$ we use the code `f = @(x) x.^2 + 1`. Where the variables are defined by the `@` symbol in front of the list of variables in parenthesis. You can also define functions of several variables: `f = @(x,y) x.^2 + y.^2`. It is also possible to define a vector of functions (we'll see this later.)

It is also possible to define a function composition as seen in Figure 1. Many ML functions accept function handles as inputs so that you can evaluate functions over a range of values.

Figure 1

```
f = @(x) x.^2 + 1;  
g = @(x) sin(x);  
y = @(x) f(g(x));
```

Function Files

What about more complicated functions? Or, suppose you need to use a function in several different script files or you want to use a particular function to test the output of another piece of code? Defining a function file can save you from defining the same function over and over again. The syntax for a function file is as follows.

Example Script File

```
function [ output_args ] = myFcn( input_args )  
% UNTITLED Summary of this function goes here  
% Detailed explanation goes here  
  
% your function goes here  
end
```

The easiest way to create a function file is to select the menu under “new” and select “Function”.

Which do I use?

Anonymous functions are helpful when you are using functions with a simple definition. For more complicated functions you should be using a function file. It is also worth noting that MATLAB will try to determine if you mean a vectorized function (i.e. one that takes vector inputs) or using the function element wise.

As a general rule, use an anonymous function if it's definition can fit on a single line (and is still easy to understand). For anything else, use a function file.

Branching: Making 'decisions' in your code

If Statements

The `if` statement is another important control structure which executes a 'decision' in your code. The basic format of an `if` statement is:

Example if statement

```
if (logical statement 1)
    (things to do 1)
elseif (logical statement 2)
    (things to do 2)
.
.
.
else
    (default things to do when ...
      above conditions fail)
end
```

Logical operators

Logical operators are used to construct logical statements. They are usually used in the execution of an `if` statement.

Matlab Symbol	Logical Operator
<code>==</code>	equal to
<code>~=</code>	not equal
<code>></code>	greater than
<code><</code>	less than
<code>>=</code>	greater than or equal to
<code><=</code>	less than or equal to
<code>&</code>	AND
<code> </code>	OR

Repeating Tasks: for and while loops

Some basic building blocks for repeating the same action in a program are `for` loops and `while` loops. These coding structures exist in MATLAB and many other programming languages. Fundamentally, these two structures operate in the same way; they repeat a block of code until they reach a condition that tells them to stop.

for loops

`for` loops use an indexing variable. In MATLAB this is a vector that is constructed using the colon notation. For each element in the index vector, MATLAB executes the block of code. Syntax for a `for` loop is:

Conditions can be set within the `for` loop to break out of the loop if needed.

Example for loop

```
for i:j:k
    (things to do 1)
.
.
.
end
```

while loops

`while` loops do not use an indexing variable. A `while` loop will repeat until some condition is met. Of course the danger in this type of loop is that it could repeat forever if the condition is never met. Often, an indexing variable is used to set some maximum number of iterations to prevent an infinite loop.

Example while loop

```
while i < maxIterations
    (things to do 1)
.
.
if (logical statement 1)
    break
end
```

Most of the time you will also include a conditional statement that dictates when the loop should stop. Note that unlike in a `for` loop, any indexing variable must be updated within the loop.

Lab Exercises

Before you begin be sure to download `lab03files.zip`, un-zip it, and place the files in your MATLAB folder. Enter `diary('lab03.txt')` and `beep off`

I. Anonymous Functions

1. Open and run the script file `lab03script.m`.
2. Define a new function $y = f(g(x))$ and re-run the script file. What happens? Enter the command `whos` to see which variables and functions are defined and what type they are.
3. Comment out the problem line(s) of code (Don't delete them!). Write a comment explaining the problem.
4. Correct the problem to calculate $f(g(x))$ properly on the line below.

II. Scripts vs. Function Files

1. Open and Run the script `lab03_factorialScript.m`. This script calculates the value $9!$.
2. Use the function `lab03_factorial` in your `lab03script.m` script file to calculate the values of $3!$, $5!$, $7!$. To determine what the inputs/outputs are, use `help lab03_factorial`.
3. Open and Run the script `lab03_calcFactorials.m`. This script demonstrates two common ways in which function files can be used within another script file.
4. Enter command `diary off`.

III. Basics of LaTeX

1. You will generate a LaTeX report for this lab using link provided in the assignment on WyoCourses. Add your script file `lab03script.m` and your text file `lab03.txt`.
2. In the last section of the report you will reproduce the following in LaTeX:

Consider the function

$$f(x) = \frac{x^2 - 1}{x + 1}$$

To simplify this function we can factor the numerator and cancel like terms

$$\begin{aligned} f(x) &= \frac{x^2 - 1}{x + 1} \\ &= \frac{(x - 1)(x + 1)}{x + 1} \\ &= x - 1 \end{aligned}$$

A general 3×3 matrix A has the form

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Submit your .pdf report file to the assignment in WyoCourses.