

Math 107 Lecture 6

User-defined Functions and Local Variables

by Dr. Kurianski

on September 16, 2024

» Announcements and Objectives

Announcements

- * Skill Check 2 is NEXT Wednesday (9/25, 60 mins then lecture)
- * Pre-Notes due before start of next lecture
- * Assignments Due Friday (9/20):
 - * HW3 Handwritten Questions
 - * HW3 Coding Problems
 - * HW3 MATLAB File Upload
- * Office Hours Update: All of my office hours are now offered in hybrid format.

Objectives

- * Create and call user-defined functions in MATLAB
- * Practice using local variables

User-defined Functions

» What is a user-defined function?

- * A block of code that performs a specific tasks
- * Saved as its own m-file
- * Can help streamline your work by breaking down tasks into smaller pieces
- * Helps when troubleshooting code errors

» Function Structure

```
function [output1,output2] = functionname(input1,input2)
```

```
% Summary of what the function does.
```

```
% Inputs:
```

```
% Outputs:
```

```
Body of the function, i.e., the code that performs the  
desired task
```

```
end
```

» Function Structure

Function environment

```
function [output1,output2] = functionname(input1,input2)
```

```
% Summary of what the function does.
```

```
% Inputs:
```

```
% Outputs:
```

Body of the function, i.e., the code that performs the desired task

```
end
```

The function environment tells MATLAB that the file you are creating is a function rather than a script.

» Function Structure

Function name

```
function [output1,output2] = functionname(input1,input2)
```

```
% Summary of what the function does.
```

```
% Inputs:
```

```
% Outputs:
```

```
Body of the function, i.e., the code that performs the  
desired task
```

```
end
```

The function name must be the same as the file name that you save it under.

» Function Structure

Inputs and outputs

```
function [output1,output2] = functionname(input1,input2)
```

```
% Summary of what the function does.
```

```
% Inputs:
```

```
% Outputs:
```

```
Body of the function, i.e., the code that performs the  
desired task
```

```
end
```

The input arguments are the values (or variables) that you will give to the function so it can perform the desired task. The output arguments are what the function returns once the task is complete.

» **Function Structure****Function description**

```
function [output1,output2] = functionname(input1,input2)
```

```
% Summary of what the function does.
```

```
% Inputs:
```

```
% Outputs:
```

Body of the function, i.e., the code that performs the desired task

```
end
```

The description is a set of comments that describe the function's purpose, its inputs, and its outputs. This will display when you run "help functionname".

» **Function Structure**

Function body

```
function [output1,output2] = functionname(input1,input2)
```

```
% Summary of what the function does.
```

```
% Inputs:
```

```
% Outputs:
```

```
Body of the function, i.e., the code that performs the  
desired task
```

```
end
```

The body of the function is where you will write the code that performs the desired task. This is where you will use the input arguments in computations and assigns values to the output arguments.

» Practice

Practice running the functions TriangleArea.m and Rectangle.m

Important! Function files must be visible in your Current Folder for you to call them from the Command Window. When calling a function from a script file, the function file must be in the same folder as the script file.

» Local variables

Local variables are variables that exist only inside the function.

- * This can be confusing, because they can have the same names as variables that you use outside the function.
- * They are only accessed by MATLAB to perform the computations of the function.
- * They cannot be accessed outside of the function.

» Example

```
function [c] = demofun(a,b)
```

```
% Subtracts b from a
```

```
% Inputs: a, b
```

```
% Outputs: c
```

```
c = a - b;
```

```
end
```

» Local variables practice

Consider the function exLocalVars.m

```
function [x,y,z] = exLocalVars(x)
y = x^2;
z = x+y;
x = x+1;
end
```

What is the output of the following lines of code?

```
x = 2;
y = 4;
z = 6;
[x,y,z] = exLocalVars(x);
```

» Local variables practice

Consider the function exLocalVars.m

```
function [x,y,z] = exLocalVars(x)
y = x^2;
z = x+y;
x = x+1;
end
```

What is the output of the following lines of code?

```
x = 2;
y = 4;
z = 6;
[y,z,x] = exLocalVars(y);
```

» How do I create a function?

Option 1: Create a new script and type the function from scratch

Option 2: Select New > Function and use MATLAB's template

» Additional Notes

- * Function names must follow the same conventions as for variables.
- * When you save the function, the file must have the same name as the function name (except for the .m).
- * Run function from the command window by typing the function name, giving input values, and assigning output variables.
 - * This is called “calling” a function. You can also call a function from a separate m-file.

» Activity 1 - projectileMotion

A projectile's height is given by

$$h(t) = -4.9t^2 + 147t + 500$$

and its velocity is given by

$$v(t) = -9.8t + 147$$

where t is in seconds and height is in meters.

1. Create a function called `projectileMotion` with input of time and outputs of height and velocity.
2. Call the function to evaluate it from 0 to 30 seconds in increments of 0.5 seconds. (Hint: In the command window, create a vector `t` that represents time and use that as your input.)
3. From the command window, plot the height as a function of time and velocity versus time on the same figure.

» Activity 2 - conversionFun

Create a function called `conversionFun` that has one input (`dollars`) and one output (`ConversionTable`). The conversion table should contain four columns, with (`dollars`) in the first column and the conversions into Yen, Euros, and British Pounds in the next three columns. The conversion rates (at the time of writing) are given below:

- * 1 USD = 142.84 Yen
- * 1 USD = 0.9 Euro
- * 1 USD = 0.76 British Pound (Pound sterling)

Call the function from the command window with input (`dollars`) as a vector that ranges from 0 to 10 in increments of 2.