Intro to Functions

Function Definition

functionname.m

```
function [outputarguments] = functionname(inputarguments)
% Comment describing "what" the function does.
% Author: Ahmet Sacan, 2011
%{
Detailed comments about "how" the function works.
Examples of how to call this function.
%}

%body of the function can be separated into sections.
%% Section 1
statement1;
statement2;
...

%% Section 2
statement20;
statement21;
...

% some of the statements must assign values to
% output arguments.
```

header

body

Examples

```
 function shout ( aword )
disp(aword);
```

```
 function addandprint ( x, y )
disp ( x + y );
```

```
 function a = add2 ( x, y )
a = x + y;
```

```
 function a = add3 ( x, y, z)
a = x + y + z;
```

```
 function [a, m] = addandmultiply ( x, y, z)
a = x + y;
m = x * y;
```

Functions vs. Scripts

- Anything you write on the Command window can be included in a script, which is a text file with ".m" extension. Scripts share the same Workspace as the Command window.
- Functions (can optionally) accept input arguments and (can optionally) return output values. Functions have their own Workspace.

Comments, code-blocks, multiple lines

```
% this is a single line of comment
%{
this is a
multiline comment
%}
%% this is a code-block
1+2 ...
*3+4
```

Recommended Development Cycle for Scripts/Functions

- Write the steps of your solution as comments, in English comments
  - Make sure you are able to go through these steps by hand (manually) and get the answers you want.
- Start in Command Prompt
  - Give initial values to input variables.
  - Add calculations step by step. Note the "overwriting problem".
- Copy working pieces to a "script" as you go.
  - You can copy the initializations also, but remember to comment them out in the final function.
- When script is completed, add a function header to convert it to a function.