

Functions 3:

By the end of this class you should be able to:

- Create and use a function with multiple outputs
- Call a sub-function within a script or function
- Describe the difference between **global** and **local** variables

Please download: [makesquare2.m](#)

From the class website:

www.engr.ipfw.edu/~moor\121

Note: Exam 1 date moved to February 20th

Solve this problem individually from scratch

We wish to create a function that will convert temperatures in Fahrenheit to Centigrade.

The relation between the temperature in degrees F and the temperature in degrees C is:

$$T \text{ } ^\circ\text{C} = \frac{5}{9}(T \text{ } ^\circ\text{F} - 32)$$

- Code this function (with minimal comments)
- Test it for for 43 F
- Copy function and test to word and print out

User Defined Functions (Two Output Variables)

Defining Function

```
function [A, B] = Name(input)
% general help comments
A = f1(inputs)
B = f2(inputs)
```

Calling Functions

```
>> x = ...
>> y = ...
>> [A, B] = name(x, y ...)
>> on going calculations with A & B
```

Two Outputs Example

- Consider if I want the cylinder function to return both Surface Area and Volume
- Function Definition line:
`function [SA, V]=cyl(r,h)`
- Function call
`>> [s, vol]=cyl(radius, height)`
- Develop this function (handout, pg 1)
- Complete solution on web (pg 1 & 2)

Download functions

www.engr.ipfw.edu/~moor → click on 121

Download ("Save Target as ...")

→ cyl.m

→ makesquare2.m

Save to active directory

output options

example

- Examine setup of cyl
`>> help cyl`
→ notice it has two outputs
- Simple call without target variables
`>> cyl(1.3,2)`

only one output is shown, the default variable ans provides a place for only one output
- Call with two target variables
`>> [s, v]=cyl(1.3,2)`

Two variables are shown, you have provided two places for output to be stored

Vectors in Functions

cyl is set up to handle vector inputs

```

One Vector
>> r = [1, 1.3]; h = 2;
>> cyl(r, h)
or one vector typed into the call statement
>> cyl([1, 1.3], 2)
Two Vectors (lengths must match)
>> r = [r, r];
>> h = [2, 2]; h = [h, h/2];
>> cyl(r, h)
Two Vectors (what happens when lengths do not match)
>> h = h(1:2)
>> cyl(r, h)
Open function in editor - see placement of dot operators to make function work with vectors
    
```

Local & Global variables (an example using cyl)

Function → Local Vbls

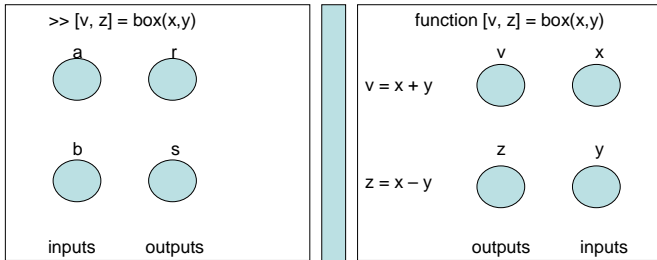
Script → Global Vbls

- | | |
|--|---|
| <ul style="list-style-type: none"> • type >> clear • look at the workspace (it will be empty) • run function >> cyl(1.3,2) • look at workspace again - all you will see is ans. The function variables are local to the function and do not end up in the main workspace | <ul style="list-style-type: none"> • type >> clear • look at workspace (again it will be empty) • open function, comment out the function line (put a % in front of it), and save the result. • type >> r = 1.3; h=2; • run as a script >> cyl • look at workspace again - all variables appear. The script variable are global and do appear in the main workspace. |
|--|---|

Function Box Example

Outside the box (function)

Inside the box (function)



Two sides separate

- Variable names separate on each side
- position is what is important

Global vs Local

Global Variables

Local Variables

- | | |
|--|--|
| <ul style="list-style-type: none"> • Available to all calculations • Used in scripts • Can be declared in functions (generally to be avoided) | <ul style="list-style-type: none"> • Available only in part of the program • Used in functions • Can be reused in separate parts of program with out conflict |
|--|--|

makesquare2.m a new music function
use to learn more about the usefulness of functions

- type help makesquare2 notice the response
- Based on help use makesquare2 to create a note vector at the command line.
>> n = makesquare2(440,20000, 0.8, 0.5);
- notice: can use it even though we do not know how it works
- clear workspace (i.e. type "clear")
- rerun makesquare2, examine workspace
- only command line variables appear
- different from a script

Follow the leader:
Develop & Run the following script

```

% program testnotes
% S. Scott Moor
% this file is a trial of using the makesquare2 function

sf=20000;

a4half = makesquare2(440,sf,1,.5);
a5quart = makesquare2(880,sf,1,.25);

tone=[a4half,a5quart];

sound(tone,sf)
    
```

Notice: we are using a script to call a function (a sub function) this function allows us to create notes of different frequency, amplitude and duration.



Practice: Five-Sided Field (handout)

1. Problem statement:
Find the perimeter of a field given its width and area. The field is five sided, made up of a rectangle and a right triangle
2. Inputs:
Area, Width of rectangle
3. Output:
Length of rectangle, perimeter