MONASH University

MONASH
ENGINEERING
ENG1060

VARIABLE SCOPE

Presented by Tony Vo

Slides by Tony Vo

GROUP OF EIGHT AUSTRALIA

VARIABLE SCOPE

MONASH University

- Variable scope governs where variables live
  - MATLAB does this using two workspaces
    - Base workspace
    - Function workspace

- Where is the variable g after you call this function?

```
function F = FindWeight(m)
g = 9.81;
F = m*g;
```

- Remember: functions are designed to hide information (black boxes)
  - Command window doesn't "see" inside of functions

VARIABLE SCOPE

- Variables are automatically stored in two separate workspaces
  - Variables created using the command window or m-files are stored inside the base workspace
  - Variables created inside a function are stored inside the function workspace

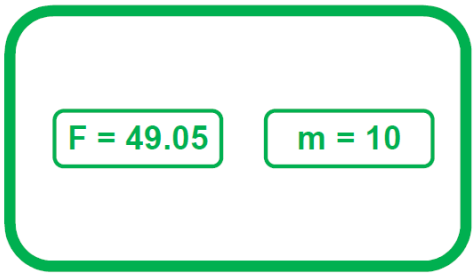
- Variables in the two workspaces do not "see each other"
  - Data that is input to function files get reassigned variable names that are understood by the function
  - Therefore, variables in the two workspaces can have the same variable name

VARIABLE SCOPE EXAMPLE

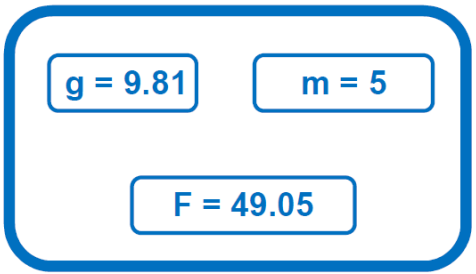
- Two "m" variables live in separate workspaces and do not affect each other

```
>> m = 10
m =
    10
>> F = FindWeight(5)
F =
   49.0500
```

```
function F = FindWeight(m)
g = 9.81;
F = m*g;
```



Base Workspace
(after running commands)



Function Workspace
(right after `F=m*g` command)

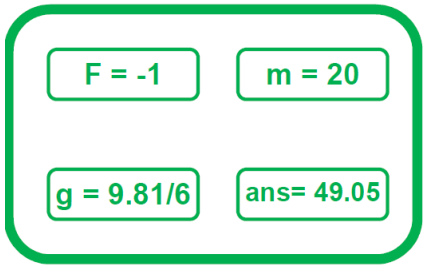
VARIABLE SCOPE EXAMPLE

- M-files use the same base workspace as the command window

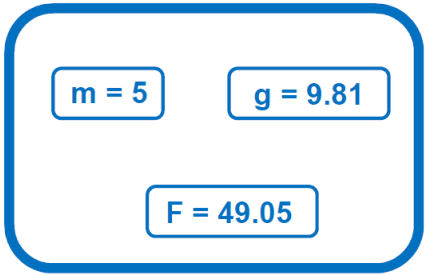
```
clear all; close all; clc;

F = -1;
m = 20;
g = 9.81/6;
FindWeight(5);
```

```
function F = FindWeight(m)
g = 9.81;
F = m*g;
```



Base Workspace
(after running commands)



Function Workspace
(right after `F=m*g` command)

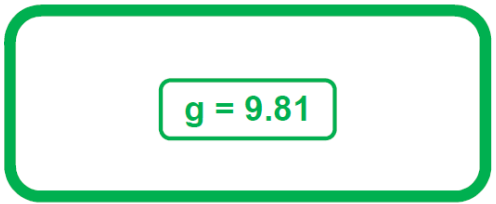
VARIABLE SCOPE EXAMPLE

- Removing the line g = 9.81 in the FindWeight() function will cause an error even if a "g" variable is declared in the base workspace

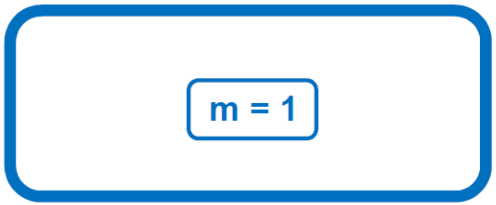
```
g =
    9.8100
>> F = FindWeight(1)
Undefined function or variable 'g'.

Error in FindWeight (line 3)
F = m*g;
```

```
function F = FindWeight(m)
% g = 9.81;
F = m*g;
```



Base Workspace
(after running commands)



Function Workspace
(right after `F=m*g` command)

VARIABLE WORKSPACES

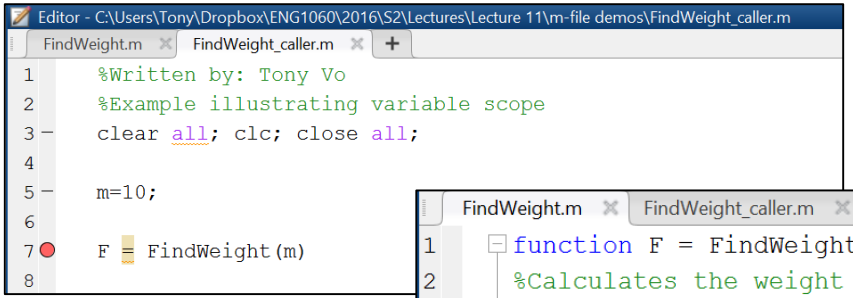
- MATLAB handles the two separate workspaces for you automatically
  - You do not have to worry about using different names for variables in function files and those in m-files

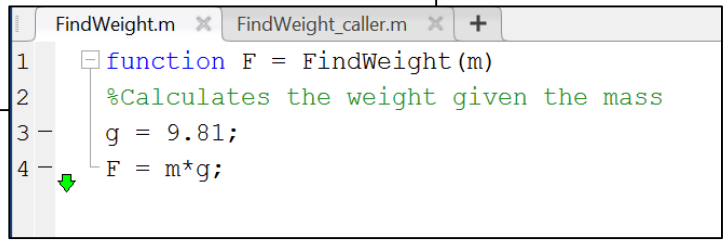
- However, follow proper variable naming rules and avoid using names that will override built-in functions and variables
  - pi, i, j, inf, sin, cos, tan, mod, sign, etc…

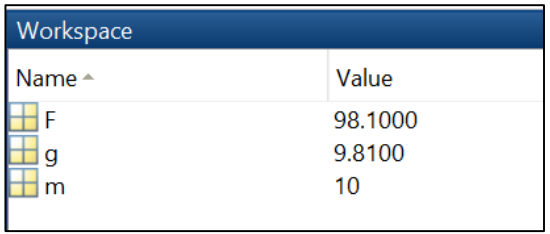
VIEWING THE FUNCTION WORKSPACE

- MATLAB displays the base workspace by default

- It is possible to view the function workspace through the debugger tool
  - Create a breakpoint at the function call line and "step in"







GLOBAL VARIABLES

- There are variables that can exist and be shared in both workspaces
  - These are known as global variables

    Syntax: global <variable name>

    <variable name> = <value>

```
% Global variable example
% Distance fallen due to gravity
clear all; close all; clc;

global G; % Global acceleration due to gravity
G = 9.81;
t = 10;
d = distance(t);
fprintf('Dist. fallen in %.2f seconds: %.2fm\n', t, d);
```

```
function d = distance(t)
global G;
d = 1/2 * G * t.^2;
```

- Note that G is not a function input

EXIST

- You can check for existing variables, built-in functions etc. using the "exist" command

0	`name` does not exist.
1	`name` is a variable in the workspace.
2	One of the following is true: ▪ `name` exists on your MATLAB® search path as a file with extension `.m`. ▪ `name` is the name of an ordinary file on your MATLAB search path. ▪ `name` is the full pathname to any file.
3	`name` exists as a MEX-file on your MATLAB search path.
4	`name` exists as a Simulink® model or library file on your MATLAB search path.
5	`name` is a built-in MATLAB function.
6	`name` is a P-file on your MATLAB search path.
7	`name` is a folder.
8	`name` is a class. (`exist` returns 0 for Java classes if you start MATLAB with the `-nojvm` option.)

```
>> m=999
m =
   999

>> exist m
ans =
     1

>> exist tan
ans =
     5
```

SUMMARY

- Variable scope: base and function workspaces
- Reassignment of variable names in the function workspace
- Global variables

- When would using global variables be considered a bad programming practice?