Day 2

- (Concept Question)
- Getting help from Matlab
- Scripts (m-files)
- Assignment statements
- Printing results to a file
- (Exercises)

Getting help from Matlab

Two methods:

- help fprintf
- doc fprintf

(Sometimes too detailed when you are just starting out– will be more useful as you gain knowledge!)
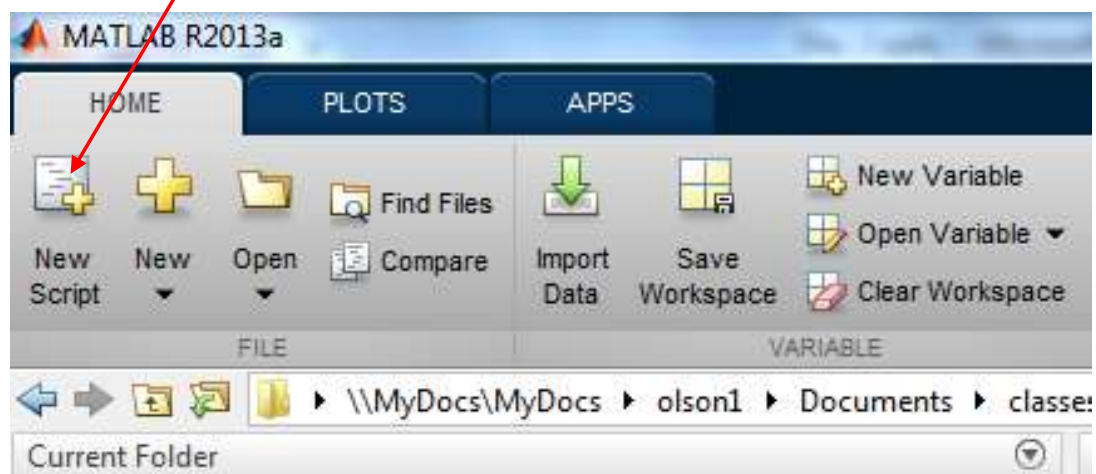
Scripts (m-files)

- Typing everything in the command window is a lot of trouble.
- We can put the commands in a "script" or "m-file" instead. Allows us to
  - edit the commands
  - save the commands
  - run them all at once
  - print out the list of commands on paper
- "Script"="m-file"=PROGRAM!

Scripts (m-files)
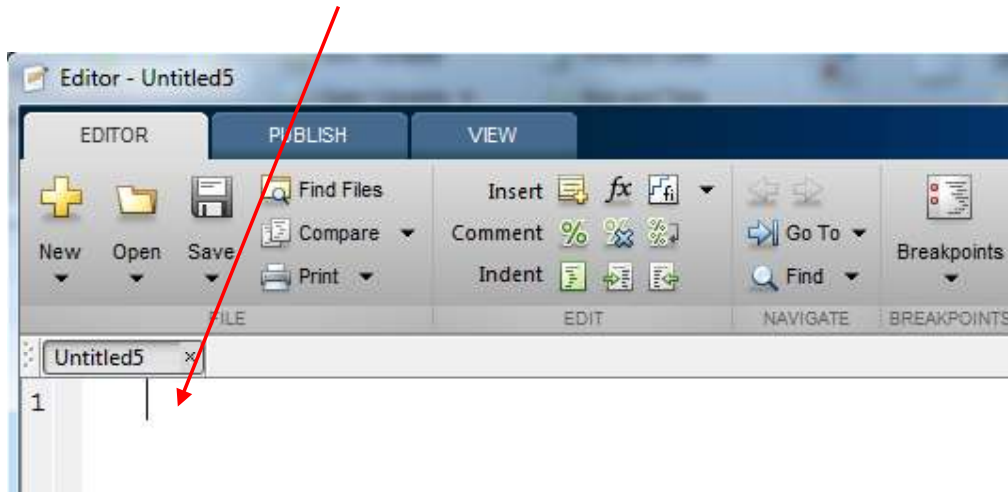
Click here to create a script:

Scripts (m-files)

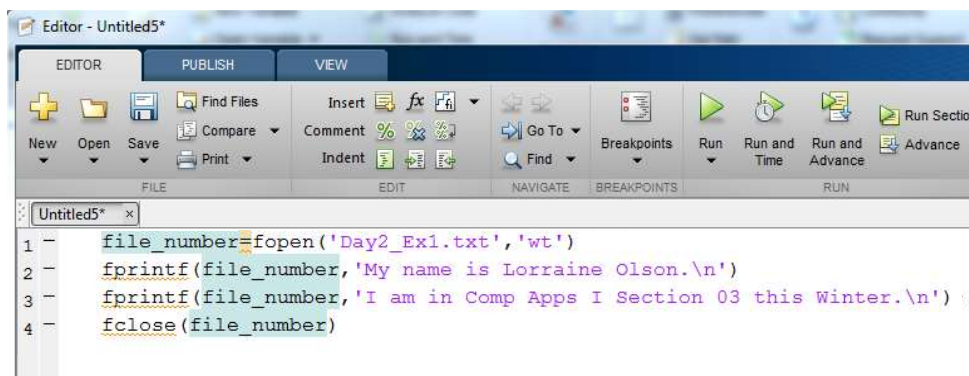Brings up a new (editor) window. You type where the cursor is:

Scripts (m-files)

Now we can type the lines from last night's homework and edit them until they are correct:

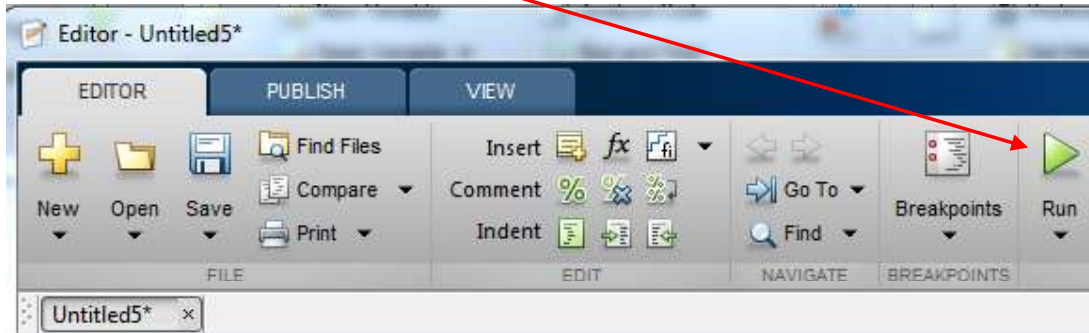Notice that nothing happens yet– you're just typing.

Scripts (m-files)
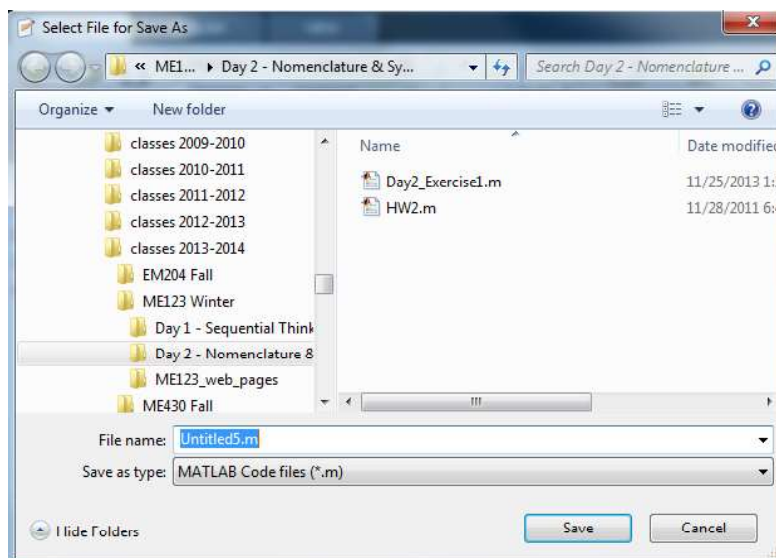
Click 'run' when you have the lines correct:

Scripts (m-files)

Matlab asks you to pick a name for the file:

Don't name it
Untitled5.m. Pick
a good name. (See
next slide)

Scripts (m-files)

Good file names

MATLAB names must [start with a letter] and contain only [letters], [numbers], or [underscores]

Good	Bad
Day2_Excercise1	My First Program
Day2Excercise1	My-First-Program
My_First_Program	1stProgram
MyFirst_Program	First#$%$#Program
My_1st_Program	sin
	cos

No spaces!

No hyphens!

No predefined functions!

Pick a good name for your m-file. Matlab will add the '.m' to the name.

Scripts (m-files)

After you name the file, it tries to run

- If all correct →
  - File name and all of the "ans = " stuff prints in the command window
  - Text will appear in the Day2_Ex1.txt file. (That's where you told Matlab to put the text.)

Scripts (m-files)

- If mistakes in script →
  - Matlab "dings" at you.
  - Red words appear in the command window
  - READ THE WORDS. Tells you what line number has the mistake.
  - Fix mistakes in script.
  - Click run again.
  - Repeat until all mistakes are eliminated.
  - You may need to clear the command window to see the new mistakes clearly (next slide).

Scripts (m-files)

To clear the command window:

right-click in the command window and choose "clear command window"

or

`>> clc` (type in command window)

or

insert 'clc' as the first line in your script (this will clear the window with every run)

Assignment statements

Assignment statements allow us to:
- Make useful calculations
- Save the results for printing

Simplest assignment statement

```
>> a=3

a =

        3
```

Assignment statements

More complicated assignment statements– put in a script:

'a' gets set to 10

```
a=10
b=5
c=2
a=b*2+c*b
```

Notice that we must define b and c before we use them– like a calculator!

Use a '*' for multiplication

'a' gets set to 20

Assignment Statements

Assignments with functions:

```
a=cosd(30)
b=cos(pi/3)
```

cos**d** for arguments in degrees

cos for arguments in radians

Other common functions:

```
a=sind(60)
b=sqrt(2)
```

```
x=5
c=exp(x)
d=x^2
```

$c = e^x$

$d = x^2$

Notice that we must define x **before** we use it– like a calculator!

Assignment Statements
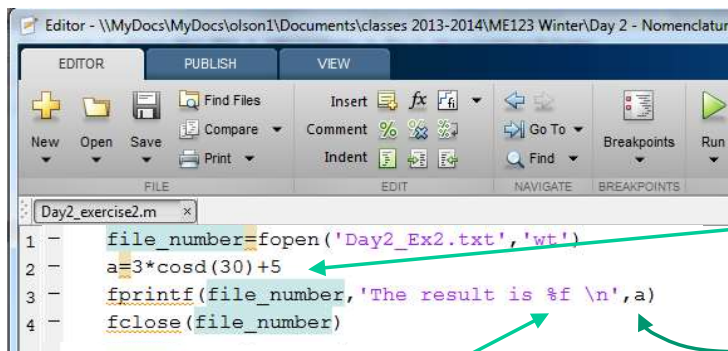
Good variable names are similar to good file names

Bad	Why?
V launch	No spaces!
sin	sin is already a function
1st_variable	can't start with a number
pi	already equal to 3.14…
v-launch	no hyphens
Day2_exercise2	Never name a variable the same as the file name!
v(t)	'(' not allowed Matlab creates an array!

Printing results to a file

```
Editor - \\MyDocs\MyDocs\olson1\Documents\classes 2013-2014\ME123 Winter\Day 2 - Nomenclature
```

```matlab
1   file_number=fopen('Day2_Ex2.txt','wt')
2   a=3*cosd(30)+5
3   fprintf(file_number,'The result is %f \n',a)
4   fclose(file_number)
```

Make the calculation

variable to print

%f is a placeholder that shows where the variable goes in the text we are printing

Printing results to a file

Printing two variables in one fprintf statement

Make both calculations

```matlab
file_number=fopen('Day2_Ex2.txt','wt')
a=3*cosd(30)+5
b=a^2
fprintf(file_number,'The result for a is %f and the result for b is %f \n',a,b)
fclose(file_number)
```

Two %f placeholders

variables to print, in the order in which they should print