B2 • Discussion #2 • 2021-02-05

Make sure that you are in the correct discussion section!!!
(this affects your attendance grade)

Announcements

- make sure you get all 3 exam dates from the syllabus on your calendars!
  - ↳ first exam is 2 weeks from today from 4^{30}–6 pm EST for everyone
  - ↳ note the mistake on the syllabus that says it's due at 6:15 pm ... it's due @ 6 pm!
- Reminder about questions ... look @ the discussion forum first to see whether the question has already been answered
  - ↳ if not, use the discussion forum on BB, do not email Prof. Attaway
  - ↳ (you can email us though)
- MATLAB Grader problems are now being released after lab, at 12^{15} pm EST on Mon/Wed
- TA Open Hours are posted
- Homework #1 is posted (it's a fun one!)
  - ↳ you must work in groups of 2 or 3
  - ↳ individual assignments will not be accepted and will receive a zero.
- Quiz #1 at 10^{45} am EST today, at the end of discussion!

Review Last Week's Practice Quiz

`4 * 2 - 6 / 3`

step 1: 4 * 2 → 8

step 2: 6 / 3 → 2

step 3: 8 − 2 => 6 (this is stored in ans)

`2 : 3 : 10`

produces a vector from 2 to 10 in steps of 3

2 5 8 ← notice that 9 and 10 are not included because they fall within that step value of 3

Review of Material

Editor – /Users/leahgaeta/Desktop/myscript.m

myscript.m | myfunction.m

```
% This is a script, this is explains what the program does

var = input('Please input a number: ');

vec = input('Thanks! Now please enter an array: ');

var_changed = myfunction(var);
vec_changed = myfunction(vec);

fprintf('Your new number is %.2f\n', var_changed)
fprintf('Here''s the new array:\n')
disp(vec_changed)
```

← this is a script. it is a .m file

Editor – /Users/leahgaeta/Desktop/myfunction.m

myscript.m | myfunction.m

```
function out = myfunction(inarg)
% This is the comment that tells us what the function does
% This function works for scalars and vectors

out = inarg.^2;
end
```

← this is a function. it is also a .m file

things to notice:

- the script calls the function (both files must be in the same working path, here, they're both in Leah's desktop)
- the function filename must be the same as the function name
- the block comments:
  - for the script, the block comment comes first
  - for the function, the block comment comes after the function header

let's at the function again:

Editor – /Users/leahgaeta/Desktop/myfunction.m

myscript.m | myfunction.m

```
function out = myfunction(inarg)   ← function header
% This is the comment that tells us what the function does   } ← block comment
% This function works for scalars and vectors

out = inarg.^2;   ← function body
end
```

↑ reserved word "end" which must be used to end a function

one more thing: notice what happens when you type help and then the filename without the extension

Command Window

```
>> help myscript
  This is a script, this is explains what the program does

>> help myfunction
  This is the comment that tells us what the function does
  This function works for scalars and vectors

fx >>
```

we ♥ the comment block! this is why it's so important to add comment blocks to your scripts and functions; if someone is trying to make sense of your program then they know what to do!

plotting

```
plot(x, y, formatting)
title()
xlabel()
ylabel()
axis([xmin xmax ymin ymax])
```

↑ this is a vector

for ex.

```
x = linspace(0, 2*pi, 50);
y = sin(x);

plot(x, y, 'bo')
title('Sine Plot')
xlabel('x axis')
ylabel('y axis')
axis([0, 2*pi, -2, 2])
```

} this creates a sine plot of blue circles

formatters that you need to know:

symbols:		colors:	
.	(dot)	r	(red)
-	(dash)	b	(blue)
+	(plus sign)	g	(green)
o	(circle)	k	(black)
*	(star)		

... if you're curious about more:
m (magenta)
c (cyan)
y (yellow)

... for fun, look up how to make your own!

let's say you want two plots on one figure window:

```
x = linspace (0, 2*pi, 50);
y1 = sin(x);
y2 = cos(x);

figure
plot (x, y1, 'bo')
title('Two Plots!')
xlabel('x values')
ylabel('y values')
axis([0, 2*pi, -2, 2])
hold on
plot(x, y2, 'r*')
legend('sin(x)', 'cos(x)')
hold off
```

} this puts the sine and cosine graphs on one plot! you need to use hold on and hold off, and you should always include a legend ... MATLAB knows that the first argument to legend is the first plot, the second argument the second plot, etc.)