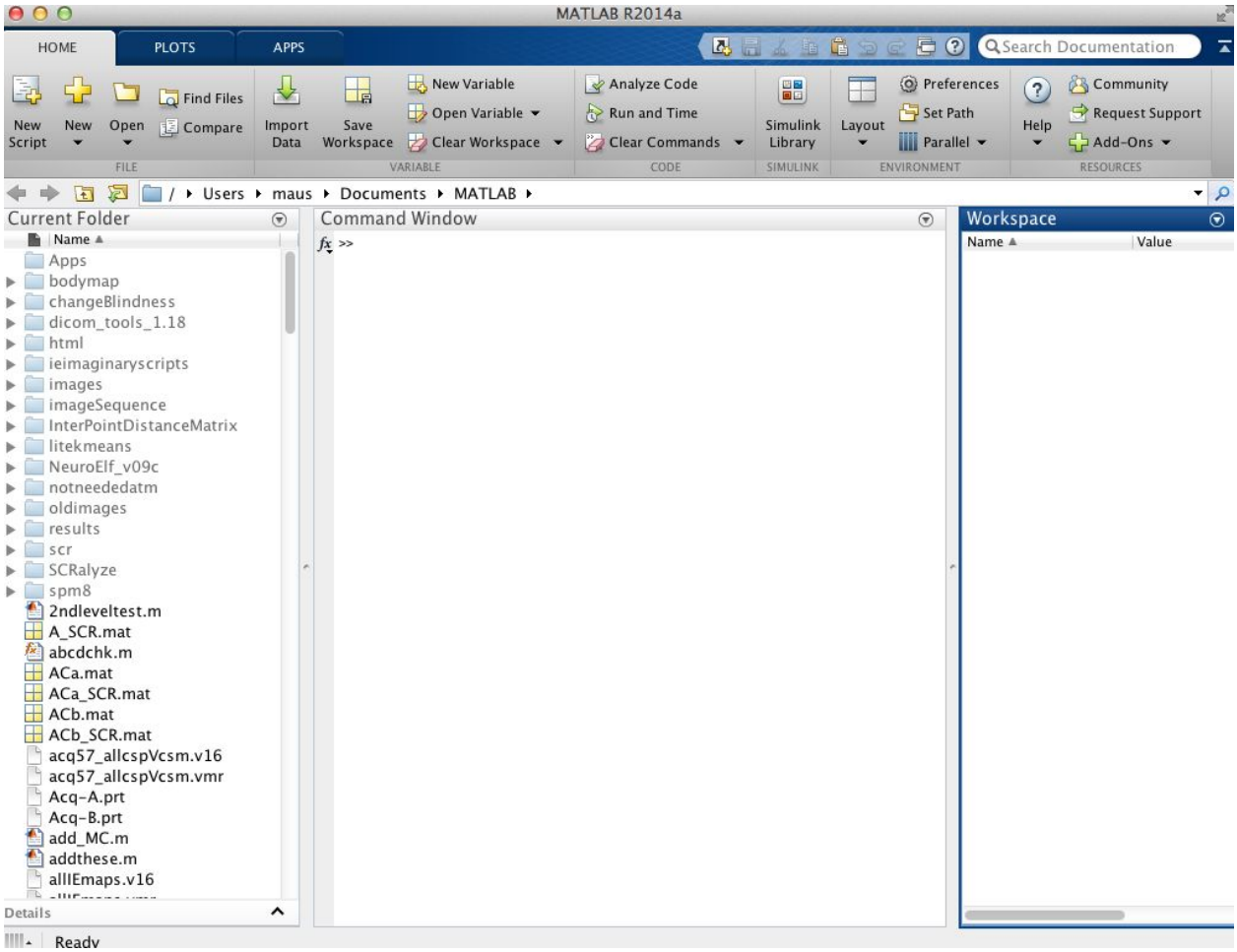
Beginner's Guide to Programming in MATLAB

by Marianne Reddan

MATLAB is a powerful coding environment, and you will soon see why. Also, all of our lab's in house software is in MATLAB, so you will not escape its grasp. I'm using a the codeacademy approach to teaching you basic programming concepts but inside of the MATLAB environment.

The Set Up



Command Window: where you will type and submit code. You can submit code directly in there, or you can open up the editor, save code, and run it there.

Workspace: All your current variables will be listed there.

Current Folder: Displays everything in the current directory. You will need to addpath to other directories which you might call code from.

Step 1:

In your command window type:

- edit MyFirstScript

When it asks if you'd like to create this, say yes. Now you are in the editor.
For now, let's play in the command line. Submit the code in these lessons directly into the command line prompt.

Lesson 1: Basic Concepts

how do i even...?

The number one most useful function to know is the **help** function (other then google). Have a question? Ask MATLAB before you ask anyone else (unless google is that someone else). If there is some function you want to use, but are not sure how to... say the ttest function... type:

- help ttest

and it will print out documentation on this function, with examples. We haven't discussed functions yet, but don't forget that you can always ask for help.

other useful help commands

- helpwin
- helpdesk
- helpbrowser
- lookfor

also available: www.mathworks.com
The matlab community is very helpful and there's lots of code sharing going on there. Googling your problem usually will take you to the very same problem and an answer on stackoverflow.

what is a variable?

A **variable** stores a piece of data, under a specific name.
for example, if you type:

- x = 100

You are saying: store the integer value 100 inside x. In MATLAB, this **datatype** is called **double**. More on this later, but pay attention to datatypes.
Note: MATLAB is case sensitive. X and x are not the same thing.

Challenge:
Assign the value 25 to a variable called age. How old are you? Correct the value of your variable age to be equal to your actual age. No you have assigned a value to a new variable, and then reassigned it. Notice how you do not save a copy of age's past values in the workspace.

---

what are Booleans?

Booleans are logicals. They can have only one of two values: ON or OFF, YES or NO, TRUE or FALSE. To have the variable a represent true, type:

- a=true

Lowercase t. Look at the workspace. See it turned your true into a 1? In MATLAB we typically use 1 or 0 for Booleans, but when you type it like this, instead of saying

- a=1

Your datatype is equal to **logical**. Not double.

Challenge:
Set the variable catsarenice to the logical value of false.
---
what is a string?
Computers don't like words, but we humans sure do. If we want one of our variables to be assigned a word, or any letter of the alphabet, we are creating a **string**. A **string** can contain letters, numbers, and symbols.
type:

- mystring='hello'

If you try to type that without the apostrophes, the computer will think you are trying to assign the value of hello to the value of mystring, but then find no existing variable called hello, and then get very upset and send you a big red error. To indicate strings, use the apostrophes.
what is a comment?
A **comment** is a line of text that MATLAB won't try to run as code. It's just for humans to read. It can be incredibly useful to not only other people trying to figure out what you did, but to your future self who will also try to figure out what the heck you did.
In MATLAB, we comment with **%**, as so:

- %this is a comment
- %but the next line is code
- cats={'Yoda','Monte','Lisa'};
- size(cats)

Protip: Develop good coding habits early. Document as you go. The future will thank you.
Now, if you want to write a really really long comment, so your future self has lots of reading material while your code crashes MATLAB, you don't have to start every line with %, instead you can do this:

- %{ this is the start of my comment
- it was a pretty good comment
- it had a lot to say
- and never confused anyone %}

This is called a **multiline comment**. Begin and end the long comment with **%{...%}**. When things are commented in the MATLAB editor they turn green! (unless you change your color settings).
but what does the semicolon do?
Placing a semicolon at the send of a line of code in MATLAB suppresses the output. This is good practice when writing scripts because writing things to the command window is messy and time consuming.

- string1='when i write code like this, it doesn't go into the command window, only the workspace';
- string2='when i write code like this, it goes into both'

how do I keep tabs on my workspace?
You have a few variables in that workspace now. Congrats! To see what's inside type:
For a list of all your existing variable names:

- who

For a print out of all the variables, their sizes, and datatypes:

- whos

For everything that is on your current path (so all the scripts and files you can access right now):

- what

For some comic relief when the going gets rough:

- why

Looking for a new start? Type:

- clear

With clear you can clear the whole workspace (which

- clear

or

- clear all

will do. Or you can specify select variables to clear:

- clear catsarenice

Looking to save this for another day? Type:

- save NAME

where NAME is the name of the file you will save your entire current workspace to. You can also save only certain variables by doing:

- save MyFirstWorkspace catsarenice x

Now if you clear all, you can load back up the variables catsarenice and x by typing:

- load MyFirstWorkspace

Protip: Tab complete! Save speed and be more accurate by typing less! When you start to type any name of a function or script on path or any variable in your workspace, hit tab, it will autocomplete with that word or it will give you suggestions (if multiple files start with same letters).

Lesson 2: MATLAB the Calculator

MATLAB is just a fat (linear algebra) calculator. Here is how you tell it to perform basic calculations:

Add

- four=2+2;

Subtract

- two=4-2;

Divide

- two=4/2;

Multiply

- eight=4*2;

Power

- nine=3^3;

order of operations rules hold, use parentheses when needed

e.g. (3+4)*7

***also*: powers, log, trig, exponentials:**

- log(x)
- exp(x) … e^x

- sin(x) etc.
- pi
- 2e3 = 2*10^3

VERY IMPORTANT: to denote **element-wise operation** when manipulating matrices put a . (period) in front of the operator

4 .* 5

We will get back to this

Accuracy

internally MATLAB uses 16 significant decimal digits, but doesn't always display these in an answer. You can control the display.

format long : for 14 - 15 digits

format short : for 4 digits

This is important when taking timestamps, etc

Matlab number range

-Inf …- x … 0 … x ... Inf

Challenge:

1. Add 2 numbers and subtract 1 in the definition of variable 'six' so that it may live up to its name.
2. Use exponents to set the variable cats to 8.

Modulo

Modulo returns the remainder from a division. So, if you tried it on 3 / 2, it would return 1, because 2 goes into 3 evenly once, with 1 left over. In matlab, this is accomplished with the function: **mod**

type:

- help mod

To see how to use this function. Because MATLAB is designed to work with matrices, that help function is telling you how to apply this to larger datasets (which is totally useful) but for now, let's keep it simple and try it with out original example, 3/2:

- mod(3,2)

ans=1

btw, ans is a variable

Type:

- ans

MATLAB returned you the last answer, right? You can use ans like a variable but it will be updated every time you do a new calculation. You can also assign the value of ans to a new variable so you can store it.

- 2+2
- four=ans;
- 5+5

in that example ans first equals 4, then equals 10, but the variable four is still 4.

Lesson 3: LOOPS!

Let's master your basic programming loops: for, while
Let's count to 10.

- for,i=1:10,i,end

This says, the first time we start this loop, i=1, the only instructions we give inside of the loop is to print the value of i in the command line. So it prints 1, first. The loop says, don't stop until i=10. So we print out 1 to 10.
We don't have to count by single integers.

- for,i=1:2:20,end

In this code, we counted from 1 to 20, by increments of 2, so we stop at the odd number 19.
Protip: Don't start a loop you can't finish. Essentially, be wary of creating infinite loops, where you have not specified to the computer when it should stop looping. If this happens, you can quit the code by typing: CNTRL+C (or COMMAND+C on mac).

Lesson 4: Paths

You need to tell matlab what folders in can look into on your computer. For example you want to add the CanLabRepository to your MATLAB path so it knows to use those functions.
You do this like:

- addpath(genpath('/Users/maus/Documents/MATLAB/Wager2008_EmotionReg_Sample data'))

Addpath adds filepath to a specified folder, but it won't add path to folders in that folder. Genpath takes care of that. Genpath iteratively adds path to every subfolder and subfolder's subfolders... etc within the mother folder your specified.