

PROFESSOR: This is Dr. MATLAB, lecture 3. If you're using MATLAB on the command line for enough time, you realize that you really want to save a bunch of commands in a file and then execute all those commands in one go. This helps you so that if you have typos, you can correct the typos without introducing new ones. It also allows you to create large pieces of code that call each other and really have complicated projects.

To see how this works, we need an example. What we'll do is generate the Fibonacci sequence. So in the Fibonacci sequence, the first two terms are 1. And then each term is the sum of the previous two, like that.

Notice that this will create a new term in `f` that will be the sum of the very last one plus the second-to-last one. Here it created 2, the sum of 1 plus 1. If we run that command again, it will generate 3, and then 5, and then 8, and then 13, et cetera.

So now I want to take these three important lines-- this one, this one, and this one-- and create a file that will do all of this and put this line inside a loop so that it does that, say, 20 times. To do that, I open the History browser. I want it to be docked. And it's docked.

And here I find my commands that I issued here. I'm going to select all of them, the three commands, right click, and copy. So now I've copied these three commands. Over here, I can open a new script. The first thing I want to do is get rid of this bar because it uses up a lot of important space. So now I have a bit more space, and I can close this command history because I have copied those commands. And now I'm going to paste.

After I've pasted this, of course, I need to clean up a bit. So I'm going to add semi-colons at the end of these commands because I don't need to see the output. And I'm going to add a for loop that will calculate the first 20 to the next 20 terms. So I'm going to have 22 in total-- two from here, and then another 20 from here. At the end, add a Tab character just for alignment to make it easy to read.

And now at the end, I will just display `f` like that. Because everything else is suppressed. I need to save this. I need to select a place to save it. The default is in Documents, Matlab. I want to save it in a different place to demonstrate what the path is.

So to do this, I open a new folder. I will call it Scripts, for example. So now this script is under

Matlab. And then in Scripts, I will call this file Fibbo. It already is an M file, which is the extension that you need to have. Fibbo stands for Fibonacci. And I save it.

And now I go to my command line and I try to run it, and it doesn't run. And the reason it doesn't run is that Fibbo is sitting inside a subdirectory of Matlab. And my current directory, found by typing CD, is Matlab. So I'm sitting in Matlab. Fibbo is sitting in Scripts, which is inside Matlab.

We can see this by looking at the current folder. So here's the current folder. Under Scripts, there's Fibbo, but we are not there. We're up here in Matlab.

So there's two ways we can solve this. We can either change directory to Scripts-- for example, by double clicking here. Now our current folder is Scripts. Now if we go here and call if Fibbo, it works, and it calculates the 22 terms of the Fibonacci sequence.

We could have also done this from the command line. Here I'm going to go back to MATLAB. And I can say change directory scripts, Tab completion. And now I'm in Scripts, and I can call Fibbo again.

Another thing I could do-- going back to MATLAB-- is to add scripts to the path. What is the path? Let's see. So if I type path, I get a list of directories, a long list of directories that Matlab uses in order to look for the commands that I issue. All the commands that you issue will have to be in one of these directories for MATLAB to look at them, or in the current directory.

So you see that the Matlab directory, the Documents Matlab directory, is always in the path. That's a default. But you can add to the path whatever you want. So for example, I can add add path scripts. Now that I've done that, let's look at the path again.

Scripts is in there. And if I try to run Fibbo, then it works even though I'm here. The way I've written it, it will look in Scripts as a relative directory and not as an absolute directory. If I want an absolute directory, I would have had to say add path slash Users slash Documents slash Matlab slash Scripts. This is a different path because this is the exact one. Let's look at the path now. Here we have the absolute and the relative scripts.

At any case, if you want to see whether Matlab can see a file, you can type which Fibbo, and then MATLAB tells you, if it sees it, which file it actually is going to. So if you have the same file name in different directories, both of which are in your path, the first one that Matlab encounters will be the one that MATLAB calls.

Let's do one more thing. So let's see what happens when we actually hide something. This can be potentially dangerous. Let's open a new file. And we'll say-- it doesn't do much. Let's say it says x equals 1. And we're going to save this file, but we're going to be a bit insidious about it, and we're going to call it Sin. And we're saving it in the Matlab directory. So we're saving a file called Sin in the Matlab directory.

And now if I do sin of 3, I get a very strange error. Notice that it's a function. But if I would have called it sine, I would have gotten the 1, because that would have been the script. So you have to be careful with things like this. And I would recommend not having such a file in your current folder.

You can also execute a file from the graphical user interface by clicking this button. If that file is not in the path and not in the current folder, Matlab will ask you whether you want to change folders or add it to the path. And you can answer whatever makes sense to you.

For a function, you need to add the keyword function in the beginning of your file. Function f to c. So let's write a function that accepts Fahrenheit and converts it to Celsius. So return value, name of the function, input. And now we want to say something like b equals n minus 32 times $5/9$.

Now v becomes this number. And that is the conversion between Fahrenheit and Celsius. So this value v will be returned to whoever calls this function.

We need to save it. MATLAB already suggests f to c. And we can put it inside Scripts next to Fibbo. Now we can write it as a function f to c. We need to give it an input. If we don't give it an input, we get an error.

So if we type 32, we get 0. If we type 101, we get this as a function. You can either end your function in the file or not. This mostly affects the behavior when you have sub-functions. And will talk about this in the future.