Module 0 – Course Tools Module

Video Transcript

Video 0.5 (03:02) – Python Environments

So, there are many different python environments. It depends on where you are, and what team you're working with, as towhich ones you might use. We're going to start off using Jupyter notebook, and for that we're going to need to first load python on our machine, and then we'll load up the Jupyter notebooks. We'll also be moving then later to using VS code, which is a full-blown integrated development environment, an IDE. It supports almost all the languages that are available today. And it also integrates with GitHub. So, GitHub is a software management system, the code management system thatis used by almost all developers now.

It also integrates with docker, and also with GitHub actions, which are quite popular now with devops pipelines, and we'll talk about those later in the course. Now, Google Colab is an alternative, in that it's an online environment so you need to be on the web basically to access it, but it's already loaded python, and all the libraries that are needed, as well as its own version of Jupyter notebook. And so, we'll be looking at that later when we get to machine learning. If you can't use Jupyternotebook, and python on your own machine, Colab is no alternative. But we highly recommend that you start off on your own machine using Jupyter notebook, and python so that you get used to controlling these environments.

We'll also be talking about GitHub and using GitHub later. GitHub has its own version of Jupyter notebooks called code spaces, which is in beta at the moment, but it's a full stack environment that co developers use for managing and deployingtheir code. So, we'll look at GitHub actions, and some of the devops pipelines in machine learning, but that will be in later in these courses. Now, AWS and Microsoft have their own version of Jupyter notebooks called AWS Sagemaker, and that's a great environment if you're operating on AWS.

Similarly, Microsoft Azure have their version of notebooks, and that integrates really well with their machine learning environment. So, we'll be looking at those later. Now, let's start loading up python, and getting our Jupyter notebooks working. Okay, bye for now.

Video 0.6 (04:34) – Python Installation - Mac

If you have already installed python on your machine, and it's configured correctly, you can go ahead, and skip this video. For the rest of us, let's go ahead and install python and do a small Hello World application, simply to verify that everything was installed. We'll go ahead, and start off by navigating to python.org, and downloading the latest installer. So, you can seethere that's a pretty fast download. I'm going to go ahead, and open this up, and find there. And I'll go ahead, and launch the installer. As you can see here, the the screens that we're going to move through are the typical ones.

So, as you can see there, Dan has gone ahead and installed. Now, let's go ahead and move to visual studio. But before we do that, let's go ahead and close some of these Windows. We're going to go ahead and close the installer, and move the installer to the trash since we don't need it anymore. We're going to go ahead, and close this as well. Now, let's go

ahead, and launch visual studio. And we're going to go ahead and create a folder here on the desktop. We'll go ahead and call it sample. And we're going to go ahead and drag and drop that on to visual studio, like so. And then inside of it, we're going togo ahead and add a file, and this is going to be our hello world file. Hello with the extension of pie. Now, we're going to go ahead, and next install the extension that visual studio has for python. And so I'm going to go ahead, and enter here python.And the very first one that we get is the one that has been developed by Microsoft, as you can see there, it is widely popularover 35 million people have installed it. So, we're going to go ahead and install it.

You might get a prompt for the developer tools, however, we don't need that at the moment. So, we're going to go ahead, and simply close that Window. Now, inside of here, I'm going to go ahead, and write the smallest python program possible. I'm simply going to enter print, then I'll enter Hello World. And that's all we're going to do. I'm going to go ahead, and save it. And next we're going to go ahead, and open up the terminal. You can do it by navigating through the toolbar here, and selecting new terminal, or using the shortcut.

I'm going to go ahead, and simply open it through the menu. And as you can see there, we can see the styling that we did previously when we configured visual studio. We can see the clear difference between the editor, and the terminal Window. And I'm going to go ahead, and enter a command simply to see if python was installed correctly. There is a default version that the Mac operating system has, which is python two, that's a very old version of python. So, we differentiate the one wejust installed as python three. That's the way it is installed. And we can confirm that by entering version, and as you can seethere, we get the version that we just installed. Now, I'm going to go ahead, and list the files here in the directory.

These are the files inside of the sample folder. If we go ahead, and open up sample, we can see there that we have Hello, and you can see hello there as well. So, next I'm going to go ahead, and run that file, that is hello. And as you can see there, that executed without any errors. And so we have installed python onto our machine. We have open, we have created a simple application inside of visual studio, and a very minimal hello world. And we can see that that executed. We can confirmthat the version that we installed is now accessible. And so if you've made it this far, then you have properly configured yourmachine to run python programs. And if you have not, then go ahead and walk through the steps, and make sure that you can do what we just demonstrated.

Video 0.7 (04:56) – Python Installation - Windows

If you have already installed Python in your machine and everything is working properly, you can go ahead and skip this video. For the rest of us, let's go ahead and navigate to Python.org and, and download the latest installer. As you can see here, I am doing precisely that and I'm going to go ahead and give it permissions to keep it. And then going to go ahead andrun the installer. And this is a pretty important point here. Remember to add Python to your path because when you open up a terminal window, then you will be able to access without having to do any additional steps.

So, in this case we're going to go ahead and install, we'll select the default location. We'll give it permission to do so. Well, then go ahead and allow, we'll go ahead and disable the path limit as well. And at this point, as you can see there our set upis complete. I'm going to go ahead and close the browser and I'll go ahead and open up Visual Studio. So, make sure that opened. And as you can see there, we have our default window. I'm going to go ahead and remove that splash window andthen going to go ahead and create here a new folder.

And I'm going to go ahead and call this folder sample, simply to indicate a sample that I'm going to write. I'll drag and drop this onto the editor and inside of it, I'm going to go ahead and add a file. This file is going to be the single file that we're going to be writing in this application and I'll call it hello.py for Python. And inside of it, I'm going to go ahead and write a minimal instruction. But before I do that, as you can see there, it is asking me to install a recommended extension.

And that recommended extension is the extension for Python. And I'm going to go ahead and select it manually here. And as you can see there, this one was written by Microsoft, it's a very popular extension, as you can see there are over 35 million people who have installed it. And so I'm going to go ahead and install it now. So, now that that has installed, I'm going to go ahead and close this pop up. We do not need to work with that configuration at the moment. So, I'm simply going to close the windows here, go back to the listing of the files and I'm simply going to go ahead and enter my minimal instruction here, which is 'Hello World' and that's all we're going to write.

I'm going to go ahead and save it. And now, I can go ahead and open a terminal here by doing it through the menu or I can go ahead and use the shortcut, in this case, I will simply use the menu. I'm going to go ahead and make my window a little bit bigger to give myself a little bit more room. And note that my terminal opens up in desktop sample and if we go ahead and open this, we can see there that instead of, inside of that folder, there's the Hello file and if I list my files by entering directory, I can go ahead and see them listed there.

So, at this point I'm going to go ahead and before I run the program, I'm going to go ahead and check the Python is in fact been installed and it's accessible at the command line. So, I'm going to go ahead and enter here Python version. Let's see if that works. And as you can see there, that is now accessible through the command line which means I can now run my program by entering python and then the name of the file which is hello.py, let me go ahead and enter it by hand here instead of using autocomplete.

And as you can see there, we get the 'Hello World' back. So, if you've followed everything up until this point and you have the same results, that means your configuration is good, you can access it through the terminal, you can write a program, you have added the extension. If not, go ahead and repeat the steps that have been done in this lesson so that you can have your configuration to be able to run Python programs in your environment.

Video 0.8 (07:47) – Jupyter Notebook

So, once we've installed Python, we're going to use an app called Jupyter Notebook. So, we go to Jupyter.org and click on 'Install' and you'll see the instructions here. I suggest that you use pip to install. And the latest version is called JupyterLab. Now JupyterLab, we can run as either a Jupyter Notebook or as a JupyterLab Notebook, which has some different behavior. We're just going to run it as an ordinary Jupyter Notebook once we've installed it. So, first do pip install. So, what you'll need to do is you'll need to go to a terminal window.

Let me see if I've got one here and we need to do pip install jupyterlab. So, once you've done that and I've already done that, obviously, then you'll have that installed. And as I say, you need to make sure obviously you have got Python installed first. So, then we can just bring up our Jupyterlab Notebook by typing jupyter notebook. And you'll see here, it will redirect you. I've already got some notebooks, but we can start a new notebook, so we'll start and you need to make sure they have different kernels. So, kernel is just a different version, if you like, of Python, so we can have Python3, which you should have, the latest version of Python.

And this will bring up an untitled notebook, we can title the notebook, I'll just call it test. So, now it's called test and then we can start typing in commands. So, when the window is like this and it's labeled as code, we can type in ordinary Python code. So, for example, we could type, c = 7 + 9, for example, and now we can execute that by doing shift return, and so that will execute it. But if we want to see the result, we can re-execute it and now it will show us the result. And there are lots of other commands and we'll show you some of the shortcuts. For example, let's suppose I want to make a cell.

So, these are called cells, and you'll see, they'll have a green bar across them, that let's suppose I want to put a cell

above. I can hit 'Escape' and then 'A' and now it puts a cell above it. Now, I've got the line numbers turned on here. If they're not on, I suggest you toggle the line number bar so that, you see mine now have disappeared, it's quite nice to have them on. Now, at the moment, this is a code cell, but I can make it a markup cell or a markdown cell. And what that allows me is, and we'll discuss markdown later.

But this is an example. It allows us to describe what these cells are doing. Now when I execute markdown, you see it gives this kind of text and we can even put images into markdown, we'll come back and discuss those. Now the next things that we should look at is this row here, that most of the time this will be sufficient. So, for example, here this is save and checkpoint. So, we can just save and checkpoint all our work. This inserts cell below. So, here notice we've got a number of cells but if we click this we'll get yet another cell.

To cut a cell, let's take that last one and we can just delete it. To insert cells, we can insert above or we can insert below. We can run cells, so for example, if I'm in this, we could run it again, gives the same answer obviously. But let's put six, say, and if I could run it be 13. We can stop things running, interrupt the kernel. And if we want to restart the kernel, for example, if we get into an infinite loop, we can go up here and we can restart and run all again or just restart the kernel or we can even shut everything down.

At the moment we've only got a Python kernel but it's possible to get other kernels. So, cells can have different modes, at the moment, this cell is in code mode, we mentioned markdown mode. If for example you get by chance into this mode, it will appear that you can't execute your cell. You can try and run it but it won't run and doesn't give you any messages. So, make sure that when you get to this cell, if it's not working, make sure it's a code that you can execute. Okay, there's help and for example, keyboard shortcuts.

Most of the things that we do along here can also be done in the keyboard shortcuts. But I found that just using this row is probably sufficient for most of them. But if you want to look at the keyboard shortcuts, they're here. As you will see, you can enter the command mode with escape and various commands. So, you can take a look at those. And over time, you'll see me using more and more of those. So, that's the help and also the reference. So, this is Python reference which we'll be using. Here's NumPy and Matplotlib and pandas which we'll be using and we may SciPy as well later on in the courses.

This shows you that we're running the Python3 kernel, I've only got that one. If for example, it's not trusted yet, you may need to click and trust the kernel. Okay, so, that's a brief introduction to Jupyter Notebook. We're just finishing off with, let's do the 'Hello World'. 'Hello World' and we can display it, there we are. That's our 'Hello World'. Okay, so, that's JupyterNotebooks, bye for now.

Video 2.1 (08:36) – Introduction to NumPy

So, let's take a look at importing the NumPy library. So, NumPy documentation is available at NumPy.org. Let me show you. So, here if you go to NumPy.org/doc/stable, it gives you instructions for setting up, quickstart tutorial, etc. Now, let's install NumPy library. Now, the first thing we need to do is we're going to import NumPy as np, but we first need to load up the NumPy library. So, we need to go to our terminal window, and we need to do a pip install NumPy. And you'll see that the latest is 1.19 at the time that I'm making this. It may be, there may be other versions if you're watching this later.

Okay. So, now we can install, now that we've done our pip install, we can install our NumPy library. Install NumPy as np. Now, NumPy is excellent for handling data, so it's widely used in Data Science. Let's take a look at one of its most powerful features and that's NumPy arrays. So, we can construct an array a=np.array and we can specify for example an array 1, 2, 3, 4. Now, NumPy arrays are different to most arrays and it's certainly different to python lists because we can only have one data type in the array. We can't mix data types. We can't for example have strings mixed with integers, mixed with floats.

So, that's one way of developing a NumPy array. So, here we have it. And if we want to print it out, we could for example dofor i in a, we then do print i and this would print out all the numbers 1, 2, 3, 4. So, we can loop over these NumPy arrays. Now, there are other ways of constructing arrays in NumPy. So, for example, we could have arange, and that would give usan array with 0-9. It gives us 10 integers, 0-9. Now, we can specify the limits. We could go, for example, from 10 to 20 in steps of 2, and now we'd have 10, 12, 14, 16, 18.

So, that's one way of constructing NumPy arrays and there are several more that we can take advantage of. For example, we can have a=zeros and that will give us an array of 10 with everything being zero. Now, notice these are floating points now, or we could have ones, and now we get an array of ones. Now, the beautiful thing about NumPy arrays is that we canoperate on the whole array at once. So, for example, we could say b=a+1. Now, is an array, it's a NumPy array and python issmart enough to recognize that and we'll now print out b. Remember it was all ones. Now, it's added 1 to every element ofthat array. So, this is one of the most powerful things about NumPy is we can operate on whole arrays. We can operate on two dimensional arrays; three dimensional arrays and we'll see how to do that. So, we can generate randomly from a choice,and in this we could put, for example, let's say 0 and 1.

So, we can specify where we're choosing from and we can specify, so this could be say a coin toss where it's either 0 or 1 and we can specify how many we want to generate. So, now this is the result of our coin toss. Now, it's purely random, so eventually there should be the same number of heads as tails if it's a fair coin and it is in this case. So, that's one way of doing choice. Another way of doing choice for example, we could use the arange. So, let's go and choose from say, 10 to 20 in 2s. So, now it'll give us 10, 12, 14 and let's choice 20 of them again. And so, you see here, 16, 12, 18, 14, etc. So, this gives us a lot of opportunities for generating different kinds of data with underlying statistics. Here this is random and uniform with general, we're choosing from this bag if you like, we're putting our hand in and picking out numbers between 10 and 20, remember not including 20. So, this is from this to this but not including it in steps of 2. Okay. So, that's, we've seen zeros, ones, choice. Let's go and generate, I'll choose some x values.

So, let me generate that from np.arange. So, let's go from say 0-10, but I'm going to go in 0.1 steps now. So, x is going to have a lot of numbers right up to 9.9, okay? Now, let's generate a sin wave. So, now I do np.sin of x and that will give me allthe y values. Let me just print them out to show you. You can see them here and you can see that I've got quite a lot. Now,let's plot those out. One of the nice things is the plotting is quite easy, matplotlib.

Remember to do pip install if you haven't got it already and we're going to use pyplot as plt, and now we can plot this out and we can just have y against x, and we get a beautiful sine wave. So, that's NumPy. We've generated quite a few examplesthere. We've done arange, and we've seen start and the step size we can alter. We can alter how big the steps are, they needn't be integers. They can be floating point numbers and we can plot these out quite easily. Okay. So, that's an introduction to NumPy and we'll delve more into some of these generations in the next video. Okay. Bye for now.

Video 0.9 (01:35) – MySQL Installation - Introduction

In this section, you're going to be doing a full database installation. That is, you're actually going to touch the configuration, you're going to install directly onto your operating system. You're not going to have something pre-configured. And the reason for that is to give you exposure to the type of issues that you might run into. Its very different buying or purchasing access to something that has been pre-configured then having to install it from the ground up and address all of the issues that might come up. So, you're going to be doing that full installation much as if you were starting say a data science groupor a data engineering group from start.

So, this will give you a good experience as to how to replicate fully the data science or data engineering environment whenit comes to setting up the database server. Now, more than that, you're going to be installing MySQL server. And although there are a great variety of database servers in the market, it is still number 1. When it comes to the developer

service, thelargest ones, it's still out there at the very first stop with quite a bit of margin. Now, as you go on you will look at more databasis but to get started, this is a great one to install.

It will give you an opportunity to go through the full installation. It has a good set of tools considering it's the number one used database platform. So, it's a good one to start with. So, now it is your turn and let's go ahead and jump into database.

Video 0.10 (04:04) – MySQL Installation - Mac

Let's install MySQL. We'll start off by navigating to the website and once we arrive there we will go ahead and select the downloads option. Well scroll down the page. Once that comes up we're going to select the community downloads. And from here we're going to select the community server. From these choices we're going to select the DMG. You can see therethe very first option. So, I'm going to go ahead and click on it then say no thanks, just start my download. Now, that we havedownloaded the file, let's go ahead and click on it and that will launch the installer.

Once that comes up, let's go ahead and double click on the package, allow it to run, continue, continue once again. We agree, let's go ahead and install and then you're going to be prompted for a password. This is the password that you use to log into your machine. We're going to go ahead and install the software. Now, we're going to go ahead and add a password.This is the password that you will use to log into your database server. Now, we're going to go ahead and click on 'Finish,' enter our machine's password one more time.

And as you can see there we have completed the installation of our server. Now, let's go ahead and close, move it to the trash. Now, we're going to go ahead and go back and install the graphical tool that we will use. This is the graphical user interface to interact with our database server. So, we're going to go ahead and back up one page here. One more, and as you can see there, MySQL Workbench is one of the options. This is the tool that we will use and now because of a bug that exists in the current version, we're going to go ahead and use the previous version, that is, instead of 8.0.23, we're going touse 8.0.22. And so we're going to go ahead and download once again the DMG. Once that it's available, we're going to go ahead and run the installer. Now, we can go ahead and minimize our browser. And as you can see there, we can now drag and drop into applications MySQL Workbench. Now, that that has installed, we can go ahead and navigate to applications, look for icon and you can see it there, MySQL Workbench. We're going to go ahead and double click on it. And as you can see here we're being challenged.

We're being told that we cannot run it because it might be malicious software. We're going to go ahead and say okay, then we're going to go ahead and open up our system preferences, navigate to security and privacy, and click on the general tab and we're going to go ahead and allow this program to run. We're going to open anyway, then select open and as you can see here at this point we have installed SQL server and we have installed the graphical interface as well which is the Workbench. You can see here the server that we installed and you can see here the graphical interface. So, at this point we're done with the installation.

Video 0.11 (05:03) – MySQL Installation - Windows

Let's install MySQL. We'll start off by navigating to MySQL.com. Once we're there, we're gonna go ahead and select downloads, scroll down the page, and select the community downloads. Once we're there, we're gonna go ahead and selectthe community server. Now, we're gonna go ahead and select from all those options, the installer for windows. And once that page comes up, as you can see there are two options. We're gonna select the big one, that one has built into it configuration, which makes the installation simpler, and so we're gonna go ahead and select that bigger file, the 400 mega file and I'm gonna go ahead and select download then I will say no thanks, just start my download and as you can see here we get a warning but I'm gonna go ahead and say keep anyway, and as you can see that file now is in the

process of completing and once that does, I'm gonna ahead and double click on the file and wait for the installer to launch. We'll go ahead and say yes. We'll say yes once again. And as you can see we now have the installer in front of us. As you can see there are a lot of options and most of the complexity in installing this pretty straightforward database has to do with the options that are presented by the installer.

There are a great deal of choices but for the most part, you can simply go with the default as you, as we will see. So, I'm gonna go ahead here and click on next. I'm going to go ahead and see here the requirements that the installer thinks I'm missing, but since I'm not going to be using those, I'll go ahead and ignore the warning. And as you can see here, these are the products that are going to be installed. The most important ones there to point out are the Server and the Workbench,which will be our graphical utility that we will use to interface the database. So, I'm gonna go ahead and click on execute and this will take some time, perhaps five minutes or seven.

Now, as you can see there, we have completed the installation, I'm gonna go ahead and click on next. And now we're goingto step into the configuration. Now, as you will see some of the naming of the buttons is not ideal, it makes you feel at times as you as if you finished. But as you can see there will be additional steps. So, this is what I mentioned at the beginning aboutit being a little bit confusing but let's ignore that for now, as I mentioned for the most part, we can go with the default options.

So, let's move through that installation. We're gonna go ahead and take the defaults. Take the default, go ahead and enter a a good password. Once you have done so, go ahead and click on 'Next,' take the defaults and as you can see here, we're going to apply that configuration and we're going to press on execute. We're now going to go ahead and click on 'Finish' and as you can see here we're still in the configuration. So, we will go ahead and select next, choose the defaults, go ahead and click on 'Next' again. And now we're going to go ahead and test that our password is correct and that everything has been installed properly.

So, I'm going to go ahead and enter my password then click on check and as you can see we were successful and our connection succeeded. Then going to go ahead and continue this configuration as you can see here, and it looks like we are finished, no. There are a few more options and I think the last one here is simply what to launch once we have completed. The Workbench is the tool, as I mentioned at the beginning, that allows us to use a graphical user interface to interact withour database. So, we're gonna to go ahead and leave that one selected and unselect the Shell. So, let's go ahead and finish and as you can see there were finally finally through the the installer and now we have the tool, the Workbench tool and wecan see here our local instance has been successfully installed.

Video 0.12 (06:12) – MySQL Workbench Interface Overview

Now that we have installed the server and the Workbench. Let's go ahead and move into the tool and make some commentsabout its functionality and what it contains, we'll start off here by looking at the landing page once you run the Workbench, and as you can see here we have the connection to our local server, the one we installed. And this connection is made automatically for us with the installer. Now let's go ahead and remove it simply to point out that we can and also to note how you would connect to other databases that you might have on your machine as well.

So, I'm going to go ahead and delete it. I'll go ahead and add the connection. I'll call it simply DB server. And as you can seethere, it is connecting to the local machine and the default port. So, let's go ahead and say, 'Okay', and as you can see there we have our connection back. I'm going to go ahead and select it and I'm going to go ahead and enter the password that you chose. Well, that I chose but you will enter the one that you chose during the installations. So, let's go ahead and do that.

And as you can see here, we have our working environment. Over on the left-hand side, you have administrative tasks

and list of things that you would do. Then you have a pane there for schemas and schemas is the name that is used more formallyfor databases. However, most everyone calls a database or a schema a database. And so that's the name we will be using. You can see here, this is the default installation, that is, I have not added anything to the server yet. If you made an installation on Windows, things will look a little bit different depending on the installer you use, you should see more or thesame.

The default full installation on Windows has a few sample databases, the one on Mac does not, so you might see somethingdifferent. Down here you have some information about the object that you are selecting. You can see in those panes, someinformation about the session as well. Now, in the middle pane you can see here or in the middle panel, you can see here the the console like equivalent. This is really a place where you can write your own scripts, you can write your queries, you can choose to save them or you can choose to open up some that you already have and we will give you some of those as we go on.

You can see here that you can add additional scripts, you can open ones that you already have, some help. You can see here,new databases, new tables, new stored procedures, functions, and so on as well as managing your connections to the database. From here on the upper right hand side you can control what panes you're looking at, you can take a look here atthe log at the bottom. This turns out to be very helpful for debugging when you are writing some of the queries or statementsthat you will make into that middle pane, that middle panel. As well, you can have here help or some of the documentation,and this can be quite useful if you wanted to remind yourself how update or some of the syntax works.

And there's some additional features there for snippets. Now having covered that. Let's go ahead and run here a very simplestatement. Some of the simplest things or one of the simplest things you can do, we want to take a look at here at what databases I have, and as I start to type that, you might note there that my font is very small. So, one of the things that you'dwant to do to get a comfortable size is to set that for yourself. There's a lot of things that you can configure and set to yourpreferences.

One of the ones I usually do here on the editor is to set upper case and this is simply a style that we will be using for the keywords within SQL. And we're going to go ahead and make one more change, and you can see here once again, that this is the line. And I'm going to go ahead and make one more change for bigger fonts. At this point I will simply enter 24 for theeditor and for the result grids, I will enter 15. And as you can see there, the font looks pretty much the same.

However, if we create a new one, you can see there the number one is nice and big. So, at this point we're going to go aheadand enter a very, very simple statement. I'm simply going to enter 'SHO' and you can see there that I'm getting the autocomplete, and if I tap over, you can see that is upper case, and so we're going to go ahead and enter 'DATABASES' and you can see there that that statement is being completed. So, I'm going to go ahead and run and you can see here that the lightning bolt allows you to run.

This will run the entire set of code that you have there in the pane. This one allows you to run a single line and you can go through the options there. Many of them are similar to what you have in an IDE, an Integrated Development Environment.And so on this point let's go ahead and run. And as you can see here, we have even more options, you can hide the the log in this case, right, to simplify here what you're looking at. And you can see here something that is pretty clean, you could even in fact remove or hide the one on the left hand side.

So, I'm getting a listing here of databases, these databases are the ones that are default onto the installation. They are administrative databases, not ones that we will be working with, but you can see here the configuration in the setup of theWorkbench. Once we make installations, we will come back with more details. So, for now, that is a first look at the Workbench.

Video 6.12 (03:43) – Installing Visual Studio Code on a Mac

In our next section, we will write some code. And because of it, we will need a code editor. If you have one that you're happywith already, you can go ahead and skip this section. For those of you who don't, let's go ahead and install Visual Studio Code. We'll start off by doing a search, and we'll go ahead and follow here, the first link, code.visualstudio.com, we're goingto go ahead and download the file. We'll then go ahead and wait for that file to download.

We'll then open up the folder, and we will extract Visual Studio Code from the zip. Once that opens up, we're going to go ahead and copy it over to our applications. And we do that simply, so it's available as one of the applications in the machine. Now, we're going to go ahead and open this up, and we'll get a warning here initially, or you might get a warning. See there,we did get it. And we'll go ahead and open up anyway; this is something we trust.

Now, we're gonna go ahead and make some customizations, the very first one being the one where we will change the theme. And as you can see here, the shortcut command is Shift, Command, and p. I'm gonna go ahead and enter that. And then I'm gonna go ahead and change the theme. In this case, I'm gonna go ahead and go here with Visual Studio Light. And I'm doing this simply because this is something that is very visible when we write the code. You of course, choose whateverit is that you prefer.

Next, we're going to go ahead and set our font size. And this is especially important for the lesson we're going to be going through to make sure we have a nice big font. I'm going to go ahead and set it at 25. There it's been set. Now soon, in the second part where we install the package that we're going to be using. We will use the terminal as well integrate it into this editor. So, we're gonna go ahead and also set the font size of the terminal. And so we're gonna look here for terminal integrated, integrated font size. Now, as you can see here, we have 12, and that's pretty small. So, we'll go ahead and make this 25 as well. Now, there's one more thing we want to add, and that is a separate color for the terminal. Otherwise, it tends to blend into the rest of the editor, and we want to make sure that we see that distinction.

So, because of it, I'm gonna go ahead and look for the Workbench. And I'm gonna go ahead and look here for Settings. I'm gonna go ahead and select it. And when that comes up, as you can see there, I get color Customizations. Now, we're gonna go ahead and enter here a couple of colors. This gives a dark background and a greenish, as you can see there, color for the font and the text that we enter. And so these are ones that I think draw a pretty distinct contrast. Choose any colors you like. So, I'm gonna go ahead and save this. Go ahead and close. And I'll close this one as well. And so there we have it. That'sthe basic configuration. We made the installation, we set the theme, the colors, then we set the font size for both the editor,and also the terminal. So, now it's your turn, make sure you can do these same steps.

Video 6.13 (03:52) – Installing Visual Studio Code on Windows

In our next section, we're going to be writing some code. And because of it, we will need a code editor. If you already have one that you're happy with, you can go ahead and skip this section. For the rest of us, let's go ahead and install visual studiocode. I'm going to start off by doing a search, and as you can see here, the first hit that I get is what I need, which is code.visualstudio.com. I'm going to navigate to that address and then I'm going to download the installer. I'll go ahead and wait, and as you can see here, I get a warning.

I'm going to go ahead and keep the file. I'm then going to go ahead and launch the installer. As you can see here, I get the usual type of prompts, where do I want to install it? I'll go with the defaults for now. And then I'm going to go ahead and add a few of these options which will make it more useful when it comes to the usability that we can get from, or the utilitythat we can get from this tool. Add some integration into the file system as you can see here as well as an icon on the desktop.

So, once I do that, I'm going to go ahead and complete my installer. And as you can see here, we have finished doing the installation. I'm then going to go ahead and launch visual studio code here, and I'm going to go ahead and make a few modifications. The very first one that I'm going to do is that, I'm going to go ahead and install or change the theme. And I'mdoing that mainly because it's more readable when it comes to recording videos. If we go with the clear theme, so I'm goingto change the colors, and this will show you as well how you can change them.

I'm going to go ahead and select here light visual studio, and then going to go ahead and change the font. And changing thefont is also for the same purposes. It's also a good thing to do when you set out to get to know the editor. I'm going to go ahead and select here size 25 which is a nice big size for the purposes of creating videos. We go ahead and hit 'Return' thereto make sure that that has taken effect. Now, in the next section, we're going to be using the terminal within the editor, andsimply to draw some distinction between them, I'm going to go ahead and customize that a little bit more. And so, initially,I'm simply going to set the same font here, because 14 is pretty small. But then I'm going to go ahead and do a search for the Workbench here, and let me go ahead and make sure that I'm entering it instead of what I had before. And once that comes up, you can see here that I have Workbench Color Customizations, and as you can see here, once I opened that file,I get a few options. I'm simply going to enter a couple of options here for the terminal as mentioned, and I'm going to go ahead and enter those, and as you can see there, that's the common terminal type of styling.

A dark background, font that is, that tone of green, but you can choose whatever you like, whatever you feel comfortable with. So, I'm going to go ahead and finish by saving that file. Note that, we have the big font already even in settings here, and I'm going to simply close these tabs here and close the browser. So, if you've made it this far, you've done the installation.If not, go ahead and finish up those additional steps.

Video 6.14 (04:03) – Installing a Driver on a Mac

Now, that you have installed your code editor in python, it's time to connect programmatically to the database. In order todo that, we're going to need to add something called a driver. Now, the function of that driver is to encapsulate all of the communication that needs to be done to be able to connect to this package. Each package that you connect to, each serverthat you connect to, you can imagine has different rules, and all of that can be wrapped up into a driver. So, as you can seethere, a driver is a python package that allows us to make that communication.

Now, there is a very specific note here noting what not to install. And the reason for that is, that if you do a search for a driver to communicate with MySQL, you'll see that there are a number of them. In our case, we will be using mysql-connector-python, as you can see there at the command line for pip install and then the rest of that name. However, note that I'm making a very specific statement not to install the similarly named mysql-connector. If you happen to install that bymistake, make sure you fully remove it before you install mysql-connector-python.

So, with that in mind, let's go ahead and move to our operating environment and add that package. As you can see here, I have an open Window without any project being loaded on visual studio code and I'm gonna go ahead and create a new folder on the desktop. I will once again call it sample, this is an empty folder and I'm gonna go ahead and drag it onto the editor. So, there's nothing in it. Now I'm gonna go ahead and open up the terminal. I'm gonna go ahead and select the option here and here is where I will be making that installation.

So I'm going to go ahead and paste that command. So, as you can see there is pip install mysql-connector-python. So, let's go ahead and install that. Now, in our environment as mentioned previously on the Mac, this is the 3 versioning of the tools.So, so let me go ahead and re-enter that with pip3 and as you can see there that is now being installed. So, as you can see there, we have made that installation I'm gonna go ahead and add a file. And this is going to be a very

simple.

Well, let's call it check since all we're doing is checking the driver at this point. So, I'm going to go ahead and call it check.py.I'm gonna go ahead and cancel this since we don't need it for now. And I'm gonna go ahead here and simply enter import mysql. connector. So, that's all we're going to go ahead and do. I'm gonna go ahead and save that file. I'm gonna go ahead here, enlist my files in my terminal and as you can see there. I can see the file, check.py. If I open up my Window, you can see it here inside of the sample directory as well.

So, now I'm going to go ahead and try to run that and I will enter checked dot py. And as you can see there that ran now, itseems like we haven't done that much, but we have we have in fact added that driver, we have been able to include it in a program and we have verified that that loads correctly. So for now if you've made it this far, you have configured your environment to be able to use this driver and it is working correctly.

Video 6.15 (03:44) – Installing a Driver on Windows

Now as you can see here, I have an empty visual studio, I'm going to go ahead and create once again a new folder on the desktop. I will call it once again sample, and then I'm going to go ahead and drag and drop this onto the editor. And now I'mgoing to go ahead and close and open my terminal. And what I'm going to do next is that I'm going to go ahead and install dot package. Now will do that using the pip command, which allows us to install a package. So, I'm going to go ahead and hit return, and as you can see there that is being installed and that has completed.

Now at this point, I'm going to go ahead and add an empty file. I'm going to go ahead and call that file check, because all we're doing is we're going to be checking that this package has been loaded. And all we're going to do inside of the file, is that we're going to import the package that we just install.

So , I'm going to go ahead and save that. Then I'm going to go ahead and take a look at my files here to make sure that thatis in the directory and as you can see there it is, I'm going to go ahead and open it, and as you can see there, we see the filein the file explorer as well. So, once we've done that, now let's go ahead and run that application that simply has one line, and this is going to be check.py and as you can see there that ran. So, it might seem like we did not do anything, but we didquite a bit.

We added that package, that driver that's necessary to be able to communicate with the database. Then we were able to import it into our application, and all of that ran without any errors. So, we've made we've made a good step forward, and this is a good stopping point, make sure you can do the same.

Video 6.19 (05:26) – Installing MySQL Shell for Mac

There are multiple ways to connect to the database server. Originally, we connected using MySQL Workbench, a very friendlyGraphical User Interface, but at other times you will need to use command line tools, and one of them is MySQL Shell as you can see here. Now conceptually, you're doing the same thing, you're reaching across the wire and you're connecting and interacting with the database server. However, in this case, you will have to do it from the command line. So, let's go ahead and install that MySQL shell and connect to the database. We'll start off by making the installation.

We're going to go ahead and navigate to MySQL downloads. We'll scroll to the bottom of the page and follow the line for community downloads. Then we'll select the 'Shell'. We'll go ahead and 'Download' that file. We'll say, 'No thanks, just startmy download.' Once that file is downloaded, we're going to go ahead and load it. And in a moment, it should appear on thedesktop, and you can see it there. We also have the installer here. Going to go ahead and double click on that. And as you can see there, we get a warning, that is we need to give this permissions.

And you can navigate here to 'System Preferences.' Open up the 'Security' section, and then go ahead and give it permission. We're going to go ahead and say here, 'Open it anyway' this is something we trust. And so we're going to go ahead and startgoing through the installer. I'm going to go ahead and 'Continue' 'Continue' 'Agree.' And we're going to go ahead and install. We need to enter the password for the machine to be able to give it permission to make this installation. And as you can see there we have completed. I'm going to go ahead and move the installer to 'Trash'. I'm going go ahead and close this, andclose the browser as well.

Now I'm going to go ahead and open up a terminal window. And now that it's open, I'm going to go ahead and enter the name of that program, and that is 'mysqlsh' for shell. And as you can see there we are the command prompt and we're goingto perform a few operations here. We're going to start off by switching, and as you can see here actually before we do that, let's go ahead and list the commands. These are the type of options that you have in case you have some questions about how to perform some of the most commonly used commands.

You can see here some of the documentation for that. And so the first thing that we're going to do is we're going to switch to sql. And you can see here what that command is and you can see here it allows us to write SQL commands So, we're goingto go ahead and start off with that. We're going to go ahead and enter '\sql'. And as you can see there our prompt has changed from being the default being JavaScript to now being SQL. So, I'm going to go ahead then now and connect to my server, and the command is the following.

Again, you can take a look at those options and help for more detail on your own. And I'm going to go ahead and enter the the parameters here. I'm going to connect using root and then the server is going to be localhost, and the port is the default port, which is 3306. And once they do that I am prompted for my password, and I'm being prompted there to save my password. I'm not going to do that. And as you can see there, we have made the connection. Now, once we make the connection and we are here on the shell and we have enabled the mode to be SQL, then we can add enter the same commands that we did when we were working on the Workbench.

So, we can do something such as 'show databases'. And we're going to go ahead and finish that to commit that command with a semi-colon. And as you can see there we get a listing of the databases that we have. So, we have the education database, we have a few others that are part of the default installation as well. So, as you can see this is pretty closely related to the experience that you had when you were using the Workbench except now we're working on a terminal window. It takes a little bit of getting used to at first, but once you do you'll see that it can be at times a much faster way to do things. And at other times as well, this will be the only way with which you can connect. And so it's good to be able to understand both and know how to operate in them. So, now it's your turn, go ahead and install the shell and run your first command. You can also do some queries, but this is enough to show you here the basic functionality. Now it's your turn.

Video 6.20 (04:28) – Installing MySQL Shell for Windows

There are many ways to connect to our database server. Originally, we connected using MySQL Workbench, a very friendly Graphical User Interface. Other ways with which you can connect other clients or command line clients. One of them is MySQL Shell. And this is what we're going to be using next. Let's go ahead, and move to the machine. And let's go ahead, and take a look at how we can run MySQL Shell. If you did the full installation of the database, you already have it, if not, you need to download those extra element from mysql.com. So, in this case I have already installed it, and you can search for it by entering mysql shell.

You can see there that the option appears for me right away. I'm going ahead, and select it. Once that comes up, I'm going ahead, and make my window bigger, because this is pretty small. And you can do that by going through properties. I'm going ahead, and select fonts, and I will choose a pretty big font here, 28, and I'm going ahead, and use Lucida Console, and let's see if this fits within the window. It's going to complain that it cannot save the changes, and that's fine.

Let's go ahead, and move it into the window. We barely fit, but for what we want to do this is fine.

So, as you can see here, we get some instructions on how to get help on the commands that we can enter, as well as to, how to quit. But I'm going to start off here by listing the help. And as you can see there we get a nice list of the type of commands that we can enter. One of the first ones that we're going to enter is switching to sql mode, as you can see here. So, I'm going ahead, and enter that. Now, I'm going ahead, and enter '\sql'. And as you can see there the prompt has changed.

You can see there that, I am switching to SQL mode, and that to be able to commit my SQL statements, I need to end with a semicolon. Now, once we've done that, the next thing that we need to do, is we need to connect to the server. This could be local or it could be remote. In this case it's local. I'm going ahead, and then after enter \c enter the parameters for the connection. And that is route is my user @localhost, that is my local machine. Then I'm going to go ahead, and use the default port which is 3306.

Once I do that, I'm going be prompted for my password, as you can see here, and as you can see there I have made a mistake, and that authentication failed. So, I'm going ahead, and try it one more time. As you can see there, I have gone a step further, is asking me to save my password. I'm going with the default, and you can see there that I've gone ahead, and made a connection. Now, at this point it's no different than being on the Workbench console, when you're executing a script. So, I'm going ahead, and enter a similar command.

I'm going ahead, and enter here simply show databases. To see the databases, the schemas that I have on the server, and so I'm now going to go ahead, and enter return and as you can see there, I have the education database. One of the databases that we have been using to illustrate some examples, and then I have a number of others that were either installed by default or are part of the administrative databases that hold some configuration information. So, at this point as you can see it's no different than being on the console.

At times you will not have Workbench available, or a graphical environment available, and you would have to connect through one of the command line tools. The Shell is a pretty good one. So, now it's your turn, go ahead, and make sure you have the program installed. Make sure you can run it. You can change the modes. You can connect the data to the database, and that you can enter some of the commands that you have already entered in the Workbench environment.

Video 10.5 (00:51) – Postman Installation — Mac

Okay, so let's install Postman. So, you go to postman.com downloads and it has downloads for Windows and Mac. And we click on 'download' and we're going to get a zip file. And you'll unzip the zip file and go through the installation. And when we finished, we should have it in our applications folder. So, if we look at Postman, here, then we can double click and bring up the app.

Video 10.6 (01:48) – Docker Installation

The easiest way to share code, for example, an application is to create a docker image out of it. The docker image then can be moved to docker Hub and can easily be downloaded, and then you can run it in a container on your own machine. Now, if you go to docs.docker.com/get-docker/, you'll find instructions for loading docker on the Mac, on Windows, and on Linux. Now on Windows, you need to be careful because there are certain system requirements that are needed, especially if you've got Windows home, you may not be able to run docker. So, check that out, and the instructions are all here for that.

So, load docker up, and then we'll be using docker to actually create images and to run docker containers with those

imagesin them. Here's a cheat sheet. So, if you go here, I have downloaded this because it will be very useful as you can see here. This shows you how to run an image. This shows you how to pull images down, how to create images, etc. I suggest you download this cheat sheet and we'll be using docker containers later in the course. Okay, bye for now.

Video 11.3 (17:12) – Flask Server Registration and Login

We've already seen how to set up a simple web server using flask. And that web server had two routes; it had the default route, this slash here. So, localhost:5000 / would take us here. Or the book's route. So, localhost:5000/books would take us here, and it would fire the functions that were on those routes. So, in the flask code, there are functions that are under the decorator, and it fires those. Now, what this means is that from this browser, we can control which function is fired on this web server. This is actually distributed computing.

This was a major challenge for many years in the 1990s of how we could do cross-machine computations, how we could fire functions on another machine and have them return a result to us. Now, Sir Tim Berners-Lee solved this in a very neat way by using this HTTP request and response. So, we can make a request on some route which will fire a function and return to us some result, an HTTP response. That's pretty amazing. And we'll come back to that because it's at the basis of most of thecomputing, the web computing that's going on today. But for now, we're going to make our web server a little bit more realistic.

One of the things that we'd like to do is when a user hits us, we'd like to have control over what routes they can visit or overwhat functions they can fire. So, we're going to force them to log in. Now we don't want them to log in every time they visit us. So, what we're going to do is when they issue an HTTP request, we're going to check if they have registered or not. So, if they haven't logged in, we're going to send them to this register page, we're going to return to them registered on HTML. Andwhat that is a form that will ask them their username and their password. So, when they submit that form, the action on theform is to go to /login. So, once they click 'submit' that form will be passed to /login and /login then will pick out their user name and password and check whether that matches the usernames and passwords that they allowed to log in. So, we're going to control the user now, and only if they've got the right username and password, will we let them see our books, for example. Now, we're going to use session cookies, sometimes called session IDs, to track the users. We're not going to ask them to log in every time. We're going to put a cookie on their machine.

So, once they've logged in, we're going to send back a response that has a cookie in it that will be then placed on that browser.And every time that browser then makes a request, it'll pass that cookie in the request back to us. We're going to check thatcookie, and if it's valid, then we're going to let them do what they're allowed to do on our machine. So, this is the security that we need when we're talking to remote browsers, and we don't necessarily know who they are. So, we have to validate them in some way. Okay, let's go and take a look at the code. So, first, let's create a login or register html page. So, we need to have a form where the username and password can be inputted. And then, when we submit, we want the action to go to
/login to that route. Again, we're going to extend the underlying index.html form, and we're going to insert this in the block content. So here, this is going to replace on the index file. It's going to replace this block content down here.

Now, we've got three routes, and it's time we put in a navigation bar. So, here I'm putting in a nav element, and I'm putting in an href, and if we click on this, we'll go to /login, and if we click on this, we'll go to /books. So, now page will look a little different. Let's take a quick look. What we'll have is we'll have register up here and list books here. So, this will be our navigation bar and will make it much prettier later on. But for now, we just want that functionality. And so, when we click onregister, we'll come to this form, and I've set up a user called 'testuser' so that, when we type that in and submit it, we'll be able to recognize that user name and password.

Let's go back to the code. So, now we've got this navigation bar that will be, remember this is the base template

index.html. So, all the others inherit this. So, this little navigation bar will appear on books and on register. Okay, so now, let's go to the app, and we need to start putting in modifications here. So, one of the things that we're going to need to do is we're going tocreate a session ID cookie. And to do that, we're going to need to store a secret key. And the way we do that is in define secret key, and this is arbitrary. If this was a production system, we'd put a much longer key in there and one that couldn't be guessed. But for now, I'm just putting secret key. Now let's go and take a look. We need to input a user. So, hereI'm going to input, and I'm going to make a list for a number of users. I'm only going to put one in at the moment, and I'm going to put that in as a dictionary in this list. So, the outer brackets of the list. And here I've got the dictionary, and I'm goingto have user name, and that's going to be testuser and password testuser. So, I only have one user.

Now I need to be able to check the user is going to register. They're going to input their user name and password. And I needto check if it's amongst the users that I recognize. So, I need some code to check that. So, here I've got to check user if username and password is given to me. I'm going to loop over all the users in user. So, I'm going to check if username in user and if password is in user. Then I'm going to return true. Otherwise, I'll return false. So, this is a simple check that I know theusers, username, and password.

We already had the default route. Now I want to put in a login route. So here, I'm going to have /login and remember this is where I come when I submit that form, and that form will be posted. So, I need to support the post method, and I'm going tosupport the get method as well. But that's just so that, I can redirect somebody they should be posting me their username and password. I can check the method. Flask provides a request function. And part of that is the method, and I can check thatit's a post. Now, let's take a look at what we need to do here. Assuming that it's a post, I'm going to pick up from the form. So, from the request.form I'm going to pick up the username and the password. So, this is the way that we get the contents of a form that's passed to us. So, now I want to check if that username and password exist. If they do, I'm going to come insidethis, if loop, and now what I'm going to do is I'm going to set something called session. I'm going to set, and you can use any key here. I want to use username because I'm going to store the username in there. We could store an object in there. For example, I could have the username, and I could have their role. Later on, we may want that because not everyone should have access to everything. But for now, I'm just going to set up session with username. Now, the user is logged in. I'm going to send them back to the index.html template and I'm going to pass in their username. If they don't pass the check for a username and password, we're going to send them back to register. And now, if it's not a POST method, it must be a GET method because these are the only two, we support. And if they do anything else, if they send us an update or a delete, thenwe'll just reject it at this level.

This decorator will reject any other HTTP request. But here, if it's a GET, we send them back to register because they haven'tfilled in the form. So, that's login and I'm providing a logout where we'll just destroy that session cookie. And as usual we've got the books and here, we're just printing out the books. So, let's 'Save' that. And one more thing to notice is that if they hitthe default route, I'm passing them to the register template. So, let's run this code. It's already running. So, let's go now to here and we hit localhost:5000. Now, you see our navigation bar, it's not a very pretty one, but it's a functional one. And we can now put in a user; testuser and testuser, and we can 'Submit' it. Let me put in a breakpoint before we submit it so, we can see what's happening. So, I'm going to go in here and in login, I'm going to put a breakpoint. Now, let's step. Let's 'Submit' and we immediately come to login. Now, let's step through. So, the request method is a 'POST.' Now, we're going to try and get the username and password. Username, 'testuser,' password, 'testuser.'

Now, it should check out. Let's go through. We'll see, we're looping over users and users and we're returning 'True' because there's only one user at the moment and we pick them out and it passes our test. So, now we step in here. Now, we set the session cookie. We're setting username in the session cookie as our username; the one that we got from the form. Now, we're going to go back to render the index template. And so, we go back to our form, and you'll see 'Welcome testuser' and you'll see our nav bar here where now we can list books. So, this part is coming from index.html, this is coming from books.html. And we still got the register one, and if I click on that, it'll take me back around again here. So,

we may want to since we're already have registered, we may want to check that if they come to register again that we don't want them to actually register. So, we may not want to put it. If they already have a session cookie, maybe we don't have them re-registeragain. Let's take a look at these session cookies. So, I want to go back to list books here.

Here we are list books. Now, let me go and bring up the id in Chrome. So, I'm going to do 'Control' option I on the Mac, 'Control Shift I' on the PC. And let me now, I'm going to Let me scrunch down this one so, we can see what's happening here.So, now let me 'Replay' this. We'll see there's a message, it's hit books. Let's take a look. Now, I'm going to expand this so wecan see the messages. So, in books, we can take a look at the headers. But I'm interested in the cookies.

And here you see our session cookie. This is the cookie we set, and it looks, it's like quite long and it looks like it's encoded. And we can take a look at it later to see its structure. We'll take a look at cookies in some detail to describe them. But we'll see here that with the books, it was a GET. And our response, while it had the cookie already in there, we saw that. And thenthe response message was the books. So, every time now that the browser hits the web server, the browser passes the cookie; it's stored in the browser.

So, the browser restoring now our cookie and passing it back to us each time, it makes a request to the web server. And we'lltalk later about exactly why we're doing this. It's because we need our web server to scale. And by doing this, we allow it to not have to keep track of the conversation with the user. So, it means that our web server can handle potentially hundreds of thousands of users. So, that's how we place a cookie. We'll see that there are more secure methods than this one. But thisone is why a lot of companies are placing cookies on your browser because they can then track you.

Okay, bye for now.

Video 11.10 (09:20) – OpenSSL: PKI Keys, Encryption, and Signing

So, let's take a look at how to use OpenSSL. The first thing we need to do is to do a brew install. Okay, so, let's take a look at OpenSSL version just to check that we've got it. And we're running 2.8.3. So, let's take a look at how we use keys. So, we're going to generate some OpenSSL keys, and we're going to show first how to sign a document and then we'll look at how to encrypt a document and also sign it. Okay, so, first we need to generate a key. And to do this, we're going to use OpenSSL and what I need to do is to first generate the key.

So, let's generate a key. So, we use OpenSSL genrsa. We're going to use genrsa key of 512 bits, and we're going to output it to a file called myprivate.pem. So, this is going to use a standard called pem. We'll see later that we may have keys in using a different standard, but this one generates a key used called myprivate.pem. Let me just cat that show you. So, that's what it looks like. Begin rsa private key and then end rsa private key.

Okay, so, that's our private key. Now, what we need to do, let me clear, we need to generate from that public key. We're going to use our private key to output our public key. So, let's take a look now. We should have public key. So, here's our public key and you can see it's typically shorter than the private key. So, now we have our two keys. Let's sign a document. So, the document we're going to sign I've already got, is a file called hello.txt. So, it just says 'Hello World.' So, we're just goingto sign that file.

Notice this one is not encrypted. And basically, what we're going to do, if we go back to our diagram. So, what we're going todo now is take this file, and we're going to create a hash code that needs to be padded. But then, we're going to use our private key to encrypt that hash code. And so, that's called our signature. Now, the only way to decrypt is with our public key,which is widely shared. So, what we're going to do then, is we're going to share the file with another person, and we're goingto share our signature, and they've already got our public key, which is widely distributed.

So, they apply the public key to the signature, decrypt it, they get the padded hash, they then get the unpadded hash. Now, because I sent them the file, they can take the hash of that, and they can compare it with the hash they just decrypted. And if those two match, it means that I must have signed with my private key because my public key generates or decrypts and get the same hash. And so, you know that the file came from me. Okay, so, that's what we're going to do now. So, let's go back to our terminal.

So, now we're going to use OpenSSL and notice we're taking the dgst, and this is going to be sha1 digest. And we're going to sign with my private key, and we're going output to sha1 sign, and we're going to sign the file, hello.txt. Okay, so, let's do that. And if I cat sha1.sign that's our signature. So, now we send our signature and the hello.text file to for example a friend. Theycan access my public signature. So, they've got all three of those. Now, what do they do? Well, they want to verify that it came from me.

So, they now want to verify using my public key and the signature that it was applied to this document hello.txt. So, they have those three pieces of information. Now, they're going to do exactly what I said and see if it came from me. And they verify that it does. They used my public key to decrypt. So, just to remind you this is what they did. So, we signed with our private key, which basically signed the hash of the document, encrypted the hash and that's our signature. So, we sent the documentseparately and we sent this hash and our signature basically, and they used our public key to decrypt it.

They took the hash of the document and compared it with the hash that they got by decryption. And if those hashes matches,they know the file came from me. So, now let's see how this was a file sent in the open 'Hello World.' But let's suppose now we want to encrypt that. So, what we're going to do now is encrypt it. And so, our file now will be this file that's input it, willbe an encrypted file, but we're just going to sign it in the same way. Okay so, now that allows us to send hidden text to our friend. Okay, so, we'll go back, and we'll encrypt the document. Okay, so, let's take a look now at encrypting the hello.txt.

So, now we need to encrypt with the public key of my friend. So, I've generated my friend's keys, and here's their public key. Okay, and now what I need to do is I'm going to use their public key here to encrypt the hello.txt and I want the output put into encrypted hello. So, let's do that. Okay, and let's look at encrypted hello. Okay, so, to decrypt that encrypted message, we need to use the private key. This is my friend's private key. So, this is my friend doing it. And they've got the encrypted text, and now they can decrypt.

So, let's do that. And they're going to put it in my decrypted, and we've retrieved 'Hello World.' So, that's how we encrypt a message. Now, what we need to do, we actually would sign, we'd go through that signature process on this encrypted message. So, instead of feeding the plain text 'Hello World' and we would now sign this encrypted hello. So, that's how signature's work. There are some limitations in the size of file that we can encrypt this way. But essentially, that's how you encrypt and decrypt documents and how you sign them. Okay, so, bye for now.

Video 14.1 (04:05) – Java Containers

Setting up a Java development environment can be quite a pain. Now, luckily for us, we can use a container to do precisely that. And as you can see here, we're going to be doing the following. We'll start off by creating an interactive container for Java development. As you can see there, we're going to be using the image called 'openjdk', and we're going to be using version 11. And we're going to open up the container into the bash. Now, inside of it, we're going to run a couple of commands to update the packages, as you can see there at the very first line.

And then the second one to install an editor. This is something that you'll frequently want to do when you have a very trimmed container that doesn't even have an editor so that you can do a little bit of testing. In this case, we're simply going to set up this to illustrate how to set up our development environment. And later, we will write code into it. So, let's

go ahead and paste that instruction into the command line. Let's go ahead and run it. And my machine right now is a clean machine. I don't have any container.

So, it's going to go ahead and download them all. And as you can see there, that was pretty fast. Also note that we are now at the command prompt of the container. So, you can see up here, this is my machine, this is the host, this is where I ran the instruction. And over here, I am sitting, as 'root', inside of the container. So, the very first thing that I'm going to do is... Not that. But I'm going to go ahead and paste that instruction. This is simply going to update the packages. And it looks like I included there the previous line.

But you can see there now that the packages are updating. And now, I'm going to go ahead and add the editor. Now, as you can see there, that was installed. If we list our files, you can see there that we are at the root of the file system. I'm going to go ahead and move into 'home'. And as you can see there, 'home' is empty. I'm simply going to demonstrate here, creating a file with our editor. This is a commonly found editor in most Linux distributions. So, I'm going to go ahead and create here a 'hello.txt'.

And as you can see there, I am now inside of the editor. All I will say is 'Hello World!' in here. Then I'm going to go ahead and save my file. And you can see there that the commands, where at the bottom of the file, I'm going to go ahead and save it. Show you that that file exists and that those are the contents. And if we were to go back to the file, you can see there the menu at the bottom of the screen that would allow you to perform some of the most frequently used commands.

So, I'm going to go ahead and exit at this point. Now, I'm going to go ahead and show you the container. And as you can see here, I have opened up the dashboard on Docker. And you can see there that my Java container is up and running. I can have the usual commands to stop it or restart it. And that now it is prepared to do Java development. So, simply to overview, we created a container. The container was created with the given command. The name that we gave it was 'javahi'. That's why you saw that in the dashboard. And the rest of it is simply some of the options that we created the container with. And in this case, that last one is to log in or to jump in at the bash once we get that container up and running.

Video 14.7 (09:57) – Spring Boot: A Web Developing Application for Java

Programming languages are pretty compact. So, when it comes to creating an application, there's more that's needed and because of it communities create application frameworks and the Java domain one of the most popular one of those widely used is Spring Boot. As you can see here, Spring is the application framework and Spring Boot goes even farther. In fact, there's a website that they have created that makes it very easy for you to create an application. And so, that's what we're going to do, we're going to start off by navigating to that website, creating that application, downloading those files onto our host machine.

We'll then create a container, that container will have everything that you need to run these applications. We will add to it an editor and also expose support as you can see. We're then going to upload that code that we have downloaded onto our host machine and then we're going to make some configurations to do our Hello World and then we're going to go ahead and run it. So, let's get started by navigating to that website and creating our application. And as you can see here on my desktop, I have opened up a browser next to and I have a directory file system. This is pointing to a folder on my desktop, you can see desktop sample and the path and then below it, I have the same except that I am at a terminal and so you can see there the same path as well.

So, I'm going to go ahead and navigate to that address, and you can see here that I have start that spring that I own that is the address that I'm navigating to and we're going to go ahead and select or leave the default as you can see there. It is using the project Maven. It is using the language of Java, the version of Spring Boot that we're using the Project Metadata.

We're going to leave all of that as default and we're simply going to add one dependency to be able to create a web server and we're going to create within it our Hello World. So, that's all we're going to do; we're then going to go ahead and generate this application and as you can see that file, that zip has been downloaded onto my machine, the host machine. Now, I'm going to go ahead and drag and drop that zip onto my sample directory as you can see there. And so, I have them here at my command line, I can list the files as well. And so, having done so we're going to go ahead and move on to our next step which is to create the container. Remember, we're going to expose the port and add an editor and as you can see here is the command line to be able to do that. I'm going to go ahead and run that container interactively that is going to open up to a prompt and you can see there that I'm using Maven, which is required for this environment. This project is from the Apache Software Foundation, if you want to learn more about it. You can see there that I am exposing the port 8080 and that I'm naming that container javamaven.

Now, below it are just the lines to be able to add an editor once I create the container. I've opened my window full size and now I'm going to go ahead and 'Paste' that command onto it. And as you can see there that moves pretty quickly through the installation and now, we have our command prompt. If I list my files, you can see there that I am at the root of that new container that I've just created. I'm now going to go ahead and install the editor and now that that is installed, we have a container configured with the right development environment and an editor for us to use. So, our next step is to upload the code and we will do so from the host. As you can see here, once again I'm at the desktop inside of the sample folder and we're going to be copying the files that are inside of demo and we're going to be copying those files to the home directory. I've chosen that one because it's empty. So, now I'm going to go ahead and enter the command and as you can see there that is docker copy demo pointing to the files inside of the demo directory here locally on the host machine and then I'm going to copy that to the container. The container is called javamaven and inside of it to the home directory.

So, once I hit that and 'Enter' it, then if we list our files here inside of the container, we can see the same files that we have on the host. Once we have moved their code into the container, then the next step that we want to do is that we want to modify it slightly that application that we just uploaded and we're going to do so by adding this code. This is a template application as you might imagine, the one that we just downloaded is just, it trimmed template to get you going on web applications and all we're going to do is that we're going to replace the code that we will see in a file in a moment with the following.

This one is simply doing the Hello World, as you can see there at the bottom it takes two parameters and if no parameter is given it simply returns Hello World. Otherwise, it returns Hello and the name that was provided. So, let's go ahead and move to the container and inside of it we're going to navigate to the source; we start off by the source directory. The path that we're going to navigate through is painful and this has to do with how this platform does directories, how Java does name spaces and as you can see there, I am going deep, deep into this code; I'm still going through it. Let's see. I got into it. Yes, you can see there that I am at the demo application and now I'm going to 'Run' my editor and edit that code; as you can see there, I have the default code and we're going to replace that with what we saw on the slide. So, I'm going to go ahead and remove this code and replace it as so. Now, I'm going to go ahead and 'Save' this file, confirm my changes and as we can see there, we have the code with that method there at the end that takes those parameters.

So, our last step now is to try it. As you can see there, we're going to access that from our host machine will be opening up a browser that will hit that container in the port that it opened, the 8080 port that was opened at the very beginning and that is the same port that is running inside of the container within that application. But before we do that, we need to run the application and you can see here what the command is. Let's 'Paste' the command and as that installs, I'm going to go ahead and open up a browser.

So, oh looks like I have an error. Yes, and the reason for that is that I am not at the root of the application, so let's go ahead and move into home. It's looking for the dependencies, is looking for the file that defines the application. You can see there that pom is the one that does that if we were to look through it and I will leave that up to you; you could see the definition

of the application. So, because it couldn't find it, it got stuck. So, I'm going go ahead and 'Run' it now and the application is now up and running and listening, I'm going to go ahead and navigate to the path; so that would be a local host. We're going to go ahead and enter Hello and once we do, once we do you can see there that we get Hello World. There were two options, or two parameters provided there as you saw in that method inside of the application. If you provided no parameter accompanying just the path that we defined, then you got the default Hello World. However, if we pass in a value and in this case the value that we're going to pass in is going to be Peter then we should get Peter and 'Yes' we do.

Conceptually, we did quite a bit so, let's review. We started out by using some scaffolding the one that is provided for us by that website. Start.spring.io and we created there our application. We downloaded that onto our host machine, then we created a container. We copy that code into the container then we added an editor. We edited the code; we created our Hello World. And then we ran that application in the container and accessed that externally. Now, there's a lot going on here from an architecture standpoint. There are several packages being used, their ports being mapped, there's a container being used in type of a host machine. So, go through this application in detail. Make sure you understand all of those abstractions and what the architecture is. Definitely make that code run for you. Make some edits, make some changes, and make sure that what you think is taking place is in fact happening. So, now it's your turn.

Video 14.9 (01:15) – Introduction to Debezium

We previously wrote a change data capture application using Python. We used the package MySQL replication. Well, as it turns out a lot of the work that is being done on data platforms is written in Java. And because of it, no surprise, the biggest packages for change data capture are written in Java as well. As you can see here, I have listed three of them with the last one being one that has been proposed by Netflix. However, by far the most widely used, the one that has been deployed the most, the one that has been built into some of the biggest applications in the market today, it's Debezium, the open-source project of Debezium.

So, we're going to be creating an application, a spring boot application that is going to implement this package. We're going to break that up, that application into three steps. The very first one is going to be one where we create a network. The second one is one where we create the database. And the third one is going to be the spring boot application that uses Debezium.

Video 16.2 (03:58) – Mapbox: A Web Client Library

Next, let's take a look at the front end at the small web client that we have provided you. This is using a library, a great free library called 'mapbox' and it's supported by the chrome browser. Now before I go any further, I should say that if you do not change anything on this client, this will still work. But I'm going to provide you with the background and walk through the code, in case you want to make any changes or you want to do any extensions to what we have provided you. So, let's take a look at that code now.

As you can see here, I have the 'index.html' file. This is the file that will be served if you had the root of the application by Flask. The top of the file brings in the mapping libraries from mapboxes, it establishes some style. And then, the rest of this file defines the application. You can see here that you will need a token. At this point, I have simply written in there, 'YourToken' but you can get your actual token from this address. I had mine before, but I updated it with 'YourToken' simply to illustrate that you will need your own.

This is a great one to have, as mentioned it's free, and you can use these maps on any application you like. Now, most of the application involves querying the server, that is asking for updated information from the server and then, updating the location of the busses on the screen. Now, you can see here that there are a couple of global variables. The first one is mapped, the second one is markers which will hold all of the busses, locations and the visual marker on the screen. We

then create a map and this is aided by that library. And then, we initialize the application by adding markers.

The very first thing that happens is that we get the bus locations. This calls the location on the server '/location' and brings back that 'json' data, then that is parsed and we return an object to this 'adMarkers' function. The very first thing that we do after that is that then, we loop through all of the busses that have been sent. If we find the marker for it, we simply update its location as you can see here, 'moveMarker.' Otherwise, we add a new marker to the screen. Now below this, there are all the functions to support that and you can read through those on your own.

Now, let's fire up the server, bring up the web client and take a look at that and see what it looks like. So, let's fire up the server. As you can see there, that's up and running. Let's open up the browser and the address is 'localhost 3000', and that's 3000. As you can see there, this is monstrous. Let's go ahead and make the size on this a little bit more reasonable. Go ahead and move in here. And if we took a look at it for a little while, you see the move, you can see there that they just move.

The other thing to note as well is that they're not following precisely route one because this is simply doing a linear mapping from one end of that bounding box to the other. But it's a pretty good one here to simply understand what's taking place, you're still working with the simulated server. If you wait for them to get all the way to the end and come back, you'll see them change color as well. So, this is a good one to get understanding on and then, once you do, you can go forward.

Video 17.4 (03:25) – Installing NiFi in a Container

Let's create the containers for our system. As you can see here, I'm starting with a blank Slate, I have no containers running and I'm going to be doing the installations one-by-one. The very first step is to create a network, this is going to be a network for the two additional containers that we're going to be creating. The first one will be a database server MySQL-my sequel and then the second one is going to be NiFi. So, let's create that network now. I will give the network the name of 'netable' for network able.

So, let's go ahead and create that, as you can see there, we have created the network. Next, let's create the container for NiFi. As you can see here from the command, I'm going to be calling it 'nifiable'. I'm going to be running it on ports 8080, mapping onto the host on 8080 as well. I have a note there saying ‘Watch out for poor conflicts. This is a frequently used port.’ So, if you're, if you're already using it, you're going to get a conflict the collision there. The network I am referencing, the network that we just created network able, ‘netable’. I'm going to run this detached, and you can see there the image that I'm going to be using. So, let's go ahead and create the container. So, as you saw there the image for NiFi is quite large, so if you're in a slow connection, it's going to take a while. Now, that we have installed or created the container, we can go ahead and navigate to the following which will be localhost, then the port is 8080, then nifi, and we can take a look at the user interface. Now, I've done this simply to confirm that the installation was successful and that we can reach it through this port.

Now, we'll go back and install the database. As mentioned, we're going to be using MySQL and as you can see there, I'm going to be using in the command line. I'm going to be using the usual ports 3306. I'm going to be calling it mysqlable. I'm also going to be passing in a password as you can see, I will be adding it to the network. It'll run detached and you can see there the version that we're going to be creating. So, let's go ahead and paste that command and you can see there the same parameters that I had on the slide.

I'm gonna go ahead and return and install MySQL. So, at this point we have created a network, we then added to that network NiFi, then we added to that network MySQL. Now, let's go ahead and take a look at the dashboard and confirm that those containers are running. So, as you can see here, we have both of those containers, we have NiFi and MySQL, up

and running. You can see the ports that they're in, and that they're nice and green and healthy.

Video 17.6 (04:15) – Installing a Driver in a Container

In order for NiFi to be able to access MySQL, it will need a driver. So, next we're going to do precisely that we're going to go ahead and download the drivers from 'dev.mysql.com', and we're going to install those directly into the container. But let's go ahead and get started by downloading those drivers. We'll go ahead and navigate to the following URL, you can see there 'Downloads Connector/J'. Once we are on the page, we're going to select the 'Platform Independent' choice. And from that, we're going to go ahead and 'Download' the zip file.

Once you've done that, go ahead and unzip that file and move into the directory in a terminal window. Once you do, you should see the following, you can see here the directory that I'm in, and the file we're looking for is this one right here. Now, next we're going to go ahead and take that driver and upload it into the container. Simply to be explicit here, we're going to take it from the host. We downloaded it onto our machine and then we're going to push it into the NiFi container. We can open a terminal window inside of the container by going through the dashboard, then mousing over here you can see the options and this one is going to open up one precisely that the terminal. So, as I do that, you can see there that has opened up a new window, inside of that container. Now, that I am inside I'm going to go ahead and move inside of 'opt/nifi'. And as you can see there if I looked at my path, I am precisely at that location. And then I'm going to go ahead and make a directory and I'll call this directory 'drivers' and move into that directory.

Confirm again that I am in the right place. And the reason I'm getting some errors is because these containers tend to be pretty trim and a lot of times the functionality that you're used to having within the host terminal, is missing here. But for what we're going to do, we can function or operate at this very trim level without an issue. Now, as you can see, the window that I'm on is the window for the container. And as you can see here right next to it, I have another one and that one is the one on the host.

So, we're going to go ahead and move back to the host that is, to our machine, outside of our container. This is where we have our directory that we downloaded. And then we're going to go ahead and upload from there. So, the command that we're going to use is the following. It is simply indicating there that we're going to copy. You can see here the very first part of that command copy MySQL connector, the name of that file. Then it is going to move that into NiFi, that's the name of our container, right? And then the path that is going to move it into.

And that's the path that we just created. So, I'm going to go ahead and hit return on this. As you can see there that command has gone through without an error. And now if we list our files, you can see there that we have that file inside of the container. So, before you go forward, make sure that you understand what's taking place here, we're working both on the host and inside of the container, we're then taking something from the host and uploading it into the container. We will use it later, but make sure you understand those distinctions and that you can upload your file, to a location that you can later find in reference.

Video 18.1 (08:18) – Overview of Hadoop: A Powerful and Extensible Platform

Hadoop is one of those projects that changes the course of computing, especially when it comes to the data side of computing. The early days, previous to the web were days in, where when it came to data there was a lot less complexity. A lot of the organizations in the enterprises had data that was pretty well structured in its generation and so storage fit neatly into rows and columns, mostly as you can see. And so, you had a world where you had a lot less velocity, variety, and volume. Now after the world goes through the revolution of the web, you certainly have a tremendous amount of variety.

The two-way channel that is the web, allowed anyone to be able to produce data and certainly to intake data as well. So,

that is an explosion. You have much more variety, you have velocity. The amount of data that is being produced is certainly overwhelming and you have a tremendous amount of volume as well. So, all of that doesn't fit neatly anymore into that rows and column, neat picture of the past, and we really enter the era of big data that has very different needs. Now a company that was looking at this at the time was Google.

They were faced with the challenge of being able to parse the information that existed in the web, certainly a great deal of variety, of velocity, and volume. And so, the machines, and the systems, and the platforms that existed to be able to address this need simply didn't stretch to that. And so, they needed to come up with a different way of doing this and they wanted to leverage commodity machines, that they could simply throw all of this data into and from that a programming model, that allowed you to be able to process those great deal big volumes of data.

And so, they introduced a number of innovations into the market and especially, in this case, I want to emphasize Google File System, the fabric that stores all of that information. MapReduce, the programming model, that allows you to process all of that data. And they added additional layers that are not as relevant to what we will be discussing here today when it comes to Hadoop. So, Hadoop tries to take those ideas. The software that Google had built was proprietary, but they shared their innovations and their ideas in carefully detailed publications.

One for the Google File System, another one for MapReduce, they did another one for Bigtable, another layer. And these were then used, and they inspired a new generation of software engineers and data scientists and data engineers to build this new platform based on those ideas and that idea is called Hadoop. And as you can see here, the very first layer of it was the distributed file system, which is the equivalent to Google's File System GFS, in this case, called, HDFS to be able to store those volumes of data. And as you can see here the fabric is responsible for taking those inputs of data, breaking them down into blocks, and mapping them onto machines.

As you can see here, there is that, varied or rich, or unstructured data that then gets mapped onto those machines. And you can see they are the block size that the default there being 128 MB and how that would be broken down. Now, it’s worth noting that this was also or part of this platform, was also the redundancy that was needed, to be able to afford those failures in that commodity hardware. As you can see there, the computing fabric was handling this and in fact, one of the number that became well known was that there would be three copies made in case one of those failed.

And it turned out to be a good one, and one that has held for a long time. Now the next layer was called the exact same thing as Google's, which was MapReduce and that is that model to be able to carry out computation. And this is one that existed even or the idea of it, existed even prior to Google. That you have a list of data, onto which you're going to map functions, and then the output of those functions of each of the items of that data are going to be reduced onto a result.

And so, here is a practical example of how you might do that and in this case this one specific to Hadoop. And you can see there that input file from the famous book, 'It was the best of times, it was the worst of times', and so on and how that could be a split there onto that list, then you map the functions onto it, and in this case, we're counting the frequency of words, and you can see there the values of the output of the mapping of the functions. Then we would do a sort and finally, a reduction, where we simply add up all the counts of those words and you can see there the final result.

Now, this turns out to be the foundation upon which a great deal more is built. But even at this level, you can see here that you have a tremendously scalable resource, that the world really didn't have and it was free, it was open-source, and it was shared openly. The last layer, there is one that I didn't mention the YARN, Yet Another Resource Negotiator. You can think of that as the management for this platform. So, what is it more concisely? It's a scalable platform for distributed storage and distributed processing. And the next part of the definition there is the important one that it is made for very large datasets really this has no limitations.

You can simply throw hardware at it as your needs increase and this scales horizontally, meaning that you simply need to

add more machines, which is a great thing. Now the other thing to notice well is that this was a big shift and how things had been done up until that point. The platforms that organizations had, were very focused at a few specific use cases. The data warehouses at the time certainly were that. And here you have a very different model, one that allows you to act upon tremendously large data sets. And as you can see there in the second block, increased computing power, fault tolerance, flexibility in data management.

That is, you could come up with any type of application that you wanted, beyond the cookie-cutter solutions that existed in the data warehousing days at those times. And you had now a tremendous amount of resources that you could pick off-the-shelf, that they don't require you to buy a license that were open-source, and we see that shift that takes place in the industry. Nowadays, that is commonplace, that the biggest companies in the world, companies like Facebook, Amazon, many others are powered by open-source but in those days, you started to see one of the first chefs.

You have this tremendous new platform that is robust, that is reliable, that is being used heavily in industry. And so, as you can see here, that becomes part of the story when it comes to big data needs and a platform to address them. You can still see here that even though this is been extended to capture a lot of the packages that are now new and that have emerged since then, you still have that footprint, you have HDFS, you have MapReduce, you have YARN, and then you have a host of other ones that are using those building blocks. But through it, you now have data storage, data processing, access and management. All of it in that new open-source paradigm. All of it addressing the multitude of needs when it comes to big data and big data needs.

Video 19.5 (07:34) – Using PySpark to *Query* Flight Data

Now that we have a Spark system up and running that we have some data, let's go ahead and do some analysis of that data using SQL, one of the languages that is supported by Spark. We'll start off by creating a Spark session and I'll give you a walkthrough through the steps that we're going to be doing and then later we'll come back and write the code in detail. The very first thing that we'll do is that we'll read the data and we'll infer some of the schema, we'll create a table as well, we'll look for the flights that are greater than 1000 miles as an example. We'll then look at some of the delays. This is one example. Here is another one where we're trying to see how long of a delay we have. So as mentioned, this is just a quick overview.

Let's go back and walk through that code one by one and we'll start off by creating a session. So, as you can see here, I am at the console on the Spark shell, the Python Spark shell I'm going to go ahead and clear my screen here and I'll get started here by creating a session, as you can see there. Well, at this point I'm simply importing the package. Next, I'm going to go ahead and create that session, I'm going to name it, DepartureDelays. Next, I'm going to go ahead and create a variable that will hold the path to our data, the departuredelays.csv file, the file that has over a million rows. Next, we're going to go ahead and read the data and as you can see here there's a few points worth noting one that the format we're telling it to expect CSV, another option there is that we are inferring a schema that is, take a look at the file make your best guess and as you can see there that will be good enough for what we're trying to do.

This is a pretty simple file structure and so let's go ahead and execute that. And as you can see there that takes a moment, we have quite a bit of data in that file, but you can see there that now has gone through. Now, we're going to then based on that, we're going to go ahead and read that into a view. As you can see here, you can think of it as a table. So, now that we have a session that we've loaded the data and that we have a table, we can go ahead and start to write some queries, and as you can see there, that is the SQL syntax that you would expect. We're selecting and we're specifying there the columns that we're looking for the distance, origin, destination from the delays table. And if we go back, you can see here where we specified that table, that view, we gave it a name. And then, where the distance is greater than 1000, we're then going to order by, we're going to do an aggregation and you can see they're by distance and descending order. We're only going to show 10 there.

So, let's go ahead and hit return. And as you can see there we get a result nicely formatted onto the console. So, it's worth noting here what took place. We uploaded a CSV file, we then infer the schema from it that is, we do not have to go through the usual exercise of being able of defining tables and creating data types and so on. And then, we were able to write a query in sequel against that data. So, you can see that there that's pretty speedy, pretty fast to be able to do this. Now next let's say we were interested in flights that had delays of over two hours between a couple of well-known airports, that is the San Francisco airport and the Chicago airport. So, let's go ahead and enter another query as you can see there we're selecting for date, delay, origin and destination from the delays table where delays are greater than 120 and the origin is San Francisco and the destination is Chicago. We're going to order once again and let's go ahead and return. So, as you can see there we get the delays, we get the origin and destination, and we get the dates as well. So, once again pretty handy functionality great combination there of SQL, python and the platform of Spark.

Now, one more in this case we want to get a sense of the types of delays that we have and I'm going to had and pasted the query onto the command there as onto the console. You can see there that we're selecting once again the delay, the origin and the destination. But in this case, we're looking at a case statement and we are going to report on very long delays, long delays, short delays, tolerable delays and so on to be able to get a sense of what this data looks like. Now, initially I'll run it quickly just as is there and you can see there the data that is coming back, but let's go ahead and update that so that we can see more of the data. Now, as we do that you can see here that we start to get tolerable, we get some early, some short and some long. So, you can start to get there more of a sense of the data. It's easier to parse certainly visually and you can see there the type of statement that we wrote.

Now, to illustrate the flexibility, we have been working on the traditional sequel model, but to illustrate that, we can do the same with a more traditional use of a dataframe. Let's go ahead and import that package and then write the very first query that we did but do it using the dataframe. And as you can see there we can select and we can specify there the columns. And then, you can do the filtering, as you can see there the where clause and then the order by. And if we had returned, you can see there that we get the same output that we originally got when we did the SQL query, which I'll repeat here again just so we can have it next to each other. Well, I wasn't quite able to get next to each other, but I think you see there the comparison of both. So, now it's your turn you can flex your SQL muscles, go ahead and dig into this data. This is a fun dataset. You can post many more questions and you can explore here within the shell what you can do.

Video 22.3 (09:28) – Introduction to DASK and Running Parallel Operations

So, if you remember last time, we were doing reinforcement machine learning and we were trying to capture the geometry of a problem, and we managed to get the layout here in a graph form and we decided that we'd flag whether an action was possible or not by a 1. If it was possible, we'd have a 1, if it's not possible then 0. And what we mean by an action is moving from one room to another. So, for example, moving from room 1 to 5 is possible, moving from room 1 to 2 is not possible. So, we started out with state 0.

So, down here and the only possible move is to 4. So, we put a 1 corresponding to that action of moving from 0 to 4. Now you needed to complete being in state 1, what were the possible actions while you could go to 3 or 5. So, this was the table that you needed to complete. This is what I asked you to do. So, this is the solution. And this is what we call our reward matrix. Now, we're going to add a wrinkle to this that we're going to add for our goal 100 units as a reward.

So, we're going to update our reward so, that if you move from to 5 you get a reward of a 100. If you move from 4 to 5 you get a 100. And we also we add this in that if we stay at 5 move from 5 to 5, it's a 100. Now, it'll be clear why we need this in as we move on. Now what we're going to do is we're going to introduce another matrix; we're going to call it a quality of move matrix or Q-matrix for short. And this is going to specify these are going to be our breadcrumbs.

So, these numbers are going to be high in some areas and low in other areas. So, we know what direction we need to move

in. So, let's take a look at this move, the quality of moving from 1 to 5. So, it means we're dealing with this row here, 1 to 5. Now, this is the formula we're going to apply, is that the quality of being in state S and moving to A depends on the reward of moving from S to A. So, moving from 1 to 5, we have a reward here of 100. So, this is going to be at least 100.

Now we choose Gamma as being 0.5, I'm choosing 0.5 so that it makes the calculations easier. And now we look at where we moved to the Q in the row of where we've moved to. So, we've moved to row 5 or state 5. So, we look along this row and see if there are any Q-values and we choose the largest. Now, at the beginning, there aren't any here. So, we're just left with that Q now will get updated to be 100, and this quantity is 0 on the right, so this becomes 100. Let's move on now.

So, the algorithm we're going to apply is we're going to choose the state at random, see what actions are possible, and calculate the Q-value for one of those actions. We've seen Q for 1 to 5, let's now take a look at Q for 1 to 3. So, here we are in 1 and here's 3. So, we know we've got a reward of 1. Now, it means that we've moved from 1 to 3, now we need to look in row 3 to see if there's any other Q that contributes. We're looking at this term here, this Qs_next.

S_next is 3, and so we look at this row and there's no value there, so we're just going to be left with that reward of 1. So, now our Q gets updated in row 1 and we've got a 1 here and 100 here. Let's go and see another action 3 to 1. So, here's 3 to 1, we know we've got a reward of 1 there. Now let's see we have to look in row 1 now to see if we've got any Q-values and we do we've already got a 1 here and we've got 100 and we take the maximum 1, so we take the 100 multiplied by Gamma, which is 0.5, so that's 50, and the 1 from our reward in our pocket if you like makes this 51. So, now we've got our Q-value here of 51. Let's take another one, moving from 1 to 3. So, here we are in 2 to 3. So, we've got the reward of 1. Now we need to look in row 3 to see what we've got there. Well, we've got a 51 there. So, we take half of that because that's the maximum in that row, we take half of it 25, I'm rounding down and then add that 1.

So, we should get 26. You're beginning to see the pattern, feeling comfortable with this? Let's do another one, we'll do 1 to 3 again. Now, this had a value of 1 already in there, but things have changed since then. So, now we need to take a look at the reward, moving from 1 to 3, so that was a reward of 1. Now we look in row 3 and we see that the maximum Q in there is 51. So, we're going to get 25 when we multiply by the Gamma the half and adding this gets updated from one to 26. We'll just do a couple more 5 to 1. So, here we are we're in row 5 and we're going to move to row 1. Now, 5 to 1 has a reward of 1. So, this term the R of the SA is 1 and now we need to look in row 1 to see what the maximum reward is, and it's 100. So, we're going to get 50 plus that 1, 51. I'm just going to do one more. This is the last one I promise you. Now, 4 to 5. So, well there's a reward of 100. So, this is already a 100 of going from 4 to 5. Now we need to look in row 5 to see the maximum Q and it's 51. So, we take half of that and add it to this. And this time I rounded up. So, that gives 126. I just wanted to show you that, even though we start with 100 here as our maximum reward this can increase because other numbers can contribute to it. So, we do this until this Q-matrix is static and has converged and I ran it for 100 iterations, and it converge to this. So, now our robot can find its way around.

Let's start the robot in row 2. So, the robot starts in row 2. It looks along here for actions that it can take, and it looks for the maximum quality and it sees a 52 here. So, it takes that. So, that means it's going to go to 3. This action is moved from two to state 3. So, now we're in state 3, we're here. Now we're in this row and we look for the maximum action. In this case we've got two that are the same. So, we could go to 1 or we could go to 4, and you could see that, those are equally possible and will get us to the goal. So, let's suppose we go to 4, let's go to 4 first. So, we look along here, and we see we've got two of 52's but we've got the 200 here moving from 4 to 5, that's the best quality move. And so, we get to our goal. So, that's how Q-matrix once we've got it will help our robot find the goal no matter where you put it now, it's got a recipe for getting to the goal. Let's see what it means in terms of code.

So, we're going to choose a random state, we're going to find a list of possible actions, choose an action at random, scan that row of Qs and choose the maximum Q. Now update the Q-value using our formula our Bellman's equation. So, adding in whatever reward plus Gamma times the maximum Q in that row. So, last time we ended up with our Q-matrix here and

we solved 4 to 5. Now I want you to update this Q matrix as follows. I want you to find the solution of 4 to 3 and of 0 to 4. So, you need to find the Q-values for both of these. Now make sure that you understand exactly what's going on. Because this is one of the most important things in reinforcement learning, understanding how we can create a map for the robot so it can find its way around. Okay, good luck. Bye for now.

Video 24.13 (06:00) – Installing Kafka in a Docker Container

In this section, we're going to take a look at Kafka, one of the most successful applications for handling streaming data at scale. We're going to use an image from Confluent, that is one of the supporters of Kafka, and we're going to deploy it into a Docker container. In fact, it's going to be deployed into multiple containers because Kafka actually has quite a number of components that need independence and are distributed across multiple machines usually, we're going to put them in multiple containers.

Kafka allows us to store streams of data in topics, and those topics are stored in multiple partitions within Kafka that ensure that if one machine goes down then we have our data still stored in other partitions. So, Kafka is meant as an application to operate at very large scale, however, it can also operate on your own machine. The goal in a lot of companies is to be able to handle this streaming data and do analytics in real-time. We've talked about the necessity of this in modern companies and here we're going to look at the components of Kafka.

Our implementation is going to be a little simpler than the one I showed there. What we're going to do is we're going to spin up Confluent's Kafka REST image and that's going to be put into nine containers and we'll look at how we need to update Docker because we need to increase the storage. And Confluent's version of Kafka has a REST API, and that's what these connections are supposed to indicate. We're going to construct a web server that is going to be both a producer of data, streaming data, and a consumer. So, we're going to stand up the server and then through a web page we're going to kick off streaming data into a topic and then we're going to consume that topic in another part of that web server.

So, let's take a look now at how we implement this. So, to do that, we're going to use a docker compose file, let me show you that docker compose file, so here it is, it builds several other applications, zookeeper is one, it builds a broker and we'll see that it builds many other apps that we're going to use. The most important one is the REST interface that we're going to communicate with, so this rest-proxy will handle that API for us. Let's go and build this, so all we need to do is to execute this file. Now, before we execute this, and this is important, we need to bring up our Docker desktop interface.

So, here I've got my Docker desktop Docker, there are no applications running, but what I need to do is I need to click on this button and go to 'Resources' and increase the resources, they will be down at a memory of 2GB, we need at least six. I'm going to set mine to eight. And if we scroll to the bottom of the page, we can apply this and restart, and it's critical that we do this because the containers will be unstable if we don't. Once you've done that, you can go back to the general interface, and we see we have no applications running. So, now we're going to bring up our terminal window, and we've got our Docker compose file in here, and we're going to do docker-compose up, and we'll see it's starting to build the containers.

So, here in Docker desktop, we see kafka-docker and if I click on that, it's already spun up the zookeeper, the broker, schema-registry, and you'll see these are running on various ports. It's a little lightly typed here, so you can't see it too well. But we're going to be communicating with Kafka via it's rest interface, and we'll be able to make contact with it. So, let me continue spinning it up. so you see here, it's doing a lot of work. Here is the sequel server, and my machine is beginning to thresh a little, and the fan is spinning up because it's doing a lot of work building these containers.
