https://www.youtube.com/watch?v=HW29067qVWk

we're going to be learning how to get started with using Jupiter notebooks now if you don't know what a Jupiter notebook is this is basically a way for us to run code interactively within a web browser alongside some visualizations and some markdown text to explain the process of what’s going on.

Now this Jupiter project evolved out of ipython so if you've ever used ipython or heard someone talking about that then that has now been rolled into this Jupiter project and there were some good reasons behind that that I won't go into too much detail about here.

Let me show you why this is useful and how powerful something like this can be.

So a lot of scientific institutions are using these notebooks in order to clearly explain exactly how they got the results and not only can these notebooks show us how they got the results but we can reproduce the results from within the notebooks themselves.

So how example i will show you one publication of the observatory that detected gravitational waves in the late 2015 they actually put out some of their research and notebook form so I have one of their notebooks pulled up here which walks you through how you can replicate some of their processing using their own data.

You can see that this isn't just Python code but they also have some markdown code which gets displayed as the instructions and also describes exactly what's going on here but you can see if we scroll down here we have a lot of markdown code that's describing what's going on we see some Python code and we see some charts here so the way that some of these notebooks are laid out it almost looks like a blog post or a textbook or something like that but these charts and everything that we see here these aren't just static visualizations like some PDF file or something like that these are visualizations that were produced right here in the browser by the code that we're looking at and we can even tweak this code and rerun these cells to get different results

So for example here we see this chart that was produced by this code here and I've actually come into this cell here then I can change something if I rerun that cell then you can see that this plot gets recreated right here within our browser interactively so you can see how that's a really cool concept of being able to interact with the data and the code in this way and you can see why notebooks like this would be becoming very popular so that's what we're going to do

We're going to learn how we can get Jupiter installed and how we can create our own notebooks get set up with different kernels what they see within these notebooks and also how to navigate around execute code and also some cool tips and tricks okay so let's go ahead and get started with installing Jupiter so I have the Jupiter documentation pulled up here on their installation page and you can see that they actually recommend installing Jupiter by using the Anaconda Python distribution

For starting a new notebook we just have to say Jupiter notebook in console and if I run that it's going to start a server and pop up with this dashboard here and this is on our localhost port 8 8 8 8 now that server that we started up in our terminal you have to leave that running that's called a notebook server and if we were to close that window and shut down that server then we wouldn't be able to access our notebooks within our localhost so you have to leave that running and you can see that currently in our dashboard here that we don't have much here if we had any files or folders in that test directory where we started that notebook server from then those would show up here but that directory was completely empty so we don't have anything here yet okay so let's go ahead and jump right into creating a new notebook and in order to create a new notebook you're going to have to have a kernel and the kernels are basically what programming language that you want to use so for python we could also have a separate kernel for python too and we'll go over how to set that up here in a little bit but for now I'm just going to go ahead and create a new notebook and this is going to say Python root and Python root on my machine is Python three and so I'm going to go ahead and select that okay so now we have a completely blank notebook here now if this is your first time using a Jupiter notebook then I would recommend coming up here and clicking on this help section here and let first let's go to this user interface tour now this tour will give a pop up on your screen that you can just use your arrow keys to scroll through and it will show you where certain things are so there it's showing us how to change the file name for the notebook and how some of these menu bars what the icons mean and things like that and also the mode indicators it will just kind of walk you through how everything is laid out but for now I'm going to go ahead and exit out of that and also let's go up here back to this help menu and now let's click on this keyboard shortcuts now depending if you're on Mac or Windows these modifier keys may be different for you but if you do scroll down here it shows you all the keyboard shortcuts that you can use within either command mode or edit mode and we'll talk about both of those in just a second so I'm going to go ahead and close this down but if you ever do want to see either of those again the user interface tour or the keyboard shortcuts they're right here in the help menu ok so the first thing that I'm going to do with our notebook is I'm just going to go up here and change this from untitled and I'm just going to call this testing Joop and I'll go ahead and hit okay there so now that renames our ipython notebook file okay so now let's actually look at adding some content to our notebook so notebooks have two different modes they have something called command mode and the other is edit mode so command mode allows us to perform actions like adding and deleting cells and you can put yourself in command mode by hitting the Escape key and the edit mode will allow us to edit ourselves and you can put yourself in edit mode by hitting the enter key and we can also just click into a cell to enter edit mode and you can see when I did that we get this green highlight around the cell so that's an indication that we're in edit mode and another indication is up here in the top right you see that we have this pencil icon here if I hover over that you can see that that says edit mode now if I hit the Escape key to go back to command mode then you can see that that pencil disappears and also our cell here turned blue but I did want to edit that cell so let's go ahead and just click in there again and now let's just add some simple code so I'm just going to do a simple print hello world okay so now that we have some code within our cell how do we execute this so if I just press ENTER you can see that I'm just adding more lines to my Python code here so if I want to execute this code then I can come up here to this cell menu and you can see that we have three different options here I can either run the cell run the cell and select below or run the cell and insert below so let's go ahead and walk through these so if I just run the cell then you can see that it executes the code that was in our cell and displays the output below here and the cell that we just ran is still selected now if I come up here and do a run cell and select below if I do that then you can see that execute the code in our cell and then it selected the cell below that one now if there isn't a cell below it which there wasn't then it will just insert a cell there so if I select this top cell again and rerun that then you can see that it didn't add another cell but it did execute that code and then selected that cell below that and lastly if I just choose our top cell here again and do this run and insert below you can see that even though there was an empty cell below here it's still expert it's still inserted this cell below the cell that we just executed so those are the differences between those three different options there and the shortcut keys for those at least on a Mac is ctrl enter to just execute the cell that you're currently in and stay there shift enter will execute the cell and move to the next one so that's executes the cell and select below and the option enter is to execute the cell and insert below so you can see that that inserted another cell below after I executed that content now for the rest of the video I'm going to be using those keyboard shortcuts just so that I can move a little bit more quickly here I'm not going to be coming up and clicking on the cell to run the cell every time so if you see code executing it's because I'm using those keyboard shortcuts now one thing about the Python code here is that this actually behaves kind of like an interactive prompt so I don't actually have to print this out here if I instead just came in here and did hello world just like we would if we started up Python in the terminal or something like that if I go ahead and rerun that then you can see that we get the same output okay so you may have noticed these incrementing numbers here beside the cell where it says in 13 and out 13 here now this is something that you have to get used to when working with these notebooks so these cells don't necessarily have to be executed from top to bottom like we're used to in regular scripts so these numbers show us the order in which these cells were executed so for example if I replace this top code here and I just say name is equal to John and I execute that now currently that doesn't give us any output but in this next cell I'm just going to say name and go ahead and run that can see that the output is John now that's what we'd expect so far but in this next cell I'm going to reset this name value here and I'm just going to say name is equal to Cori and I'm going to go ahead and run that and now if I come back here to this top cell and rerun this code here now you can see that that name variable printed out as having the value of Cori even though we didn't make that assignment until after that cell and it's because these numbers here determined the execution order not just how things are assigned from top to bottom now with that said you still want these notebooks to be readable and easy to understand so I'm still suggest putting things in the correct order from top to bottom but if you ever have a variable that doesn't have the value that you think it should have then you might want to check these numbers here on the side to make sure that your execution order isn't out of whack or anything like that now if you do get things mixed up and you just want to treat this like a normal script and run everything from top to bottom then we can come up here to the sell menu here and we can just click on run all and that will run all of the cells from top to bottom now if I rerun that then you can see that it prints out all of these numbers here are 1920 they're all in order from top to bottom and our name variable is what we would expect in a regular script and also you may have seen in that cell drop down here that we also have the options of being able to execute all of the cells above or below our current cell so if you have some cells that take a long time to compute and only want to re execute all of the cells below that point then you have the flexibility to do that okay so that's how you execute some basic code from within the cells so now let's look at how we can add some markup so markup is text that gets translated to HTML and we can use it within our notebooks to explain exactly what's going on and we saw a lot of that and the ligo example at the beginning of the video so let's say that I wanted to add some markdown to the top of this document here so first I'm going to select this top cell and now I'm going to add a new cell to the top like insert and then insert cell above and now with that top cell selected I'm going to come up here and click cell go down to cell type and I'm going to choose markdown now this isn't a markdown tutorial so I'm not going to go into detail about exactly what markdown text gets translated in HTML but I'm just going to drop in some sample markdown from a snippets file that I have pulled up here so I'm just going to grab all of this sample markdown here I'm just going to paste this into our top cell now you can see that while we're still in our edit mode here it kind of gives us a little sample of what we're going to get once we actually execute this cell but we still have our markdown here that hasn't been translated completely yet so I'm going to go ahead and just run that by hitting ctrl enter then when we run that you can see that our markdown got translated to HTML so we have a header here a header to a list and we have some italics and bold and things like that okay so that's how you can add markdown to a notebook now let's take a look at some other useful features that we can use within these notebooks so our current kernel here is using Python so any normal code that we add to our cells here will be interpreted as Python code but there are some special commands that we can use within our cells that won't just be seen as Python code so for example here I'm going to add another cell here below this so one of these special commands is if you put an exclamation point then it will interpret this command as a bash command so if I do exclamation point and do like a pip list and run that then you can see that the output that you get from that would be the same as the output if you were to run that command from both in the command line so that's kind of cool that you can run these bash commands from directly within the notebook now these notebooks also come with a bunch of built-in commands that they call magics so if you've looked at notebooks before then you may have seen these these are the ones that begin with either 1 % or 2% signs so the single % that means that the commands arguments will all come from that same line and those are called line magics and the two percent signs these mean that the entire cell will be used as that commands arguments and those are called cell magics so we can actually use one of these magic commands to list all of the other available commands and the command to list all those is called LS magic so if I put a percent sign and do LS magic and then run that cell so now you can see that lists out all of these magic commands that we can use and it splits these up into line magics here and cell magics here so let's go ahead and take a look at a couple of these so I'm going to go ahead and insert a cell below here so that we can still see all of these magics as we're working with these so now if I wanted to print out my current working directory then I could just use a single % here and do a PWD for print working directory if I run that then you can see that it shows the directory that we're currently in and you can see in this list of available line magics they also have an LS here so I can list the files and folders within that directory by doing an LS and if I run that you can see that currently we just have our single ipython notebook file now I can also pass normal commands to these just like I would within a terminal so if I do an LS - la then it'll do the long-form of those and you can see that there are more files and it lists out the users and permissions and things like that okay so those two commands that I just showed you are just similar to just being able to run bash commands within our notebook which we've already looked at but there are some of these magic commands that give us some functionality beyond just the bash commands so one of the more common magic commands that you'll see within notebooks is the matplotlib inline command and that is mat plot lib and then end line so what that does is it allows matplotlib charts to be displayed within our notebook so I'm actually not going to execute this yet and I'm going to execute your insert a cell below here and first I'm going to grab some sample code here from my snippets that will create a very simple map plot Lib chart so I'm just going to go ahead and copy that and paste that in here now this sample here is just some snippet code that I got directly from the matplotlib documentation so if I paste this in here and try to run this without first running this in line command so I'm going to execute this and try to show this plot and you can see that this is just sitting here spinning it's not doing anything and if we look over here this asterisk just means that it's still trying to run the cell to completion so I'm just going to kill this by coming up here to Colonel and I'm just going to do a restart and clear output and go ahead and hit YES on that so now I'm going to try to run this again but first I'm going to execute this matplotlib in line magic command first so if I run that and now you can see that it executed with that one there now I'm just going to go ahead and try to display this chart within our notebook so now you can see after we ran that setup command there that our plot shows up directly in the notebook now this is extremely useful so this is where we really start to see why these notebooks are becoming really popular because to be able to show plots and things like that as we're stepping through our code it really helps you explore your data and code in real time without needing to rerun the entire script okay so let me show you just a couple more of these magic commands that I think that you'll find really useful when you start using these notebooks so we can see that one of these sell magic commands here is this % % - % HTML and that will allow us to render HTML directly without using the markdown that we looked at before and this is a sell magic command which means that the entire sell will be rendered as HTML and not just the single line so just to show you an example this I'm going to go back to my snippets code here I'm just going to grab an iframe and this iframe is just going to pull up one of my other YouTube videos so if I go ahead and run this cell then you can see that we can actually embed this iframe directly in our notebook here and really you can use that to render any kind of HTML that you'd want so you can render images or links or anything like that directly within the notebook here so I'm not going to show an example of all of these commands but I do kind of want to highlight a couple of things that you can do so you can see that they also have a JavaScript command here so that would be everything that you wrote in that cell would get interpreted as JavaScript code you can see that they have a bash one here so that you can actually write out a series of bash commands and one so now there's also a useful magic command here that we can use to time our Python code and that is this time it command here so if I grab that I'm going to go ahead and insert another cell here at the bottom and I'm going to use this time at command now you may have used time it and your regular Python scripts before but this allows us to easily add that functionality into our notebooks with just this one short command so let's say that you wanted to time a list comprehension where you squared the numbers up to a thousand or something like that so I think I have that example yeah so I have that example here in my snippets so I'm just going to go ahead and grab that and paste it in and now if we run that cell then you can see it's actually going to run that time at command on our code there and then you can see that the output prints the output from that time at command and tells us the average time that it took to execute that so that can be really useful if you want to compare the execution time of different functions and things like that you can just add that time at command to the top of your cell and run those for multiple functions and get the output right below yourselves and like I said there's plenty more that you can do with those magic commands so if you do just run this LS magic command then you can see all the ones that are available for you here and you can just play around with any of these that you'd like now these Jupiter notebooks also have the ability to render certain things within the notebook without any kind of magic commands so for example we can also display a panda's data frame from directly within a notebook now I thought I had a snippet for this but apparently I don't sell just go ahead and write this out really quick so if we want to import pandas I'll do import pandas as PD and import numpy as MP and now I'm just going to create a data frame here with some random value so I'll do a panda's data frame I'm just going to put some random numbers in here so I'll do MP dot random dot Rand in and I will pass in ten and five there and then underneath here I'm just going to go ahead and print out that data frame so if I run that code then you can see that just by printing out this data frame here it actually outputs that directly within our notebook and a nice readable form and so if I was to do a DF head there and rerun that then you can see it does the same thing with just those first few rows now if you don't know what these pandas dataframes are that's not a big deal I just want to show you how you can display certain things within the notebooks that may be hard to visualize or display using a normal icon script okay so another thing that I wanted to go over as far as these notebooks goes is that you can export these notebooks in several different formats so if I wanted to export this notebook in HTML so that I could add it to my blog or something like that then I could just come up here to file that could come to download as and you can see that we have a lot of different options here we could download this as a Python file I want to choose HTML so I'll download that as an HTML and if I open up that file I just downloaded that you can see that we have an HTML version of this now these cells here I can't edit these since now it's HTML but I could post this in a blog or something like that and you can see that it's nice and readable has all of our code and all of the output now if you actually wanted to share this notebook file with somebody then what is actually in these notebook files so let me go ahead and open up one of these up in that's not the terminal window I want I'll go ahead and open up this blank one here so now if I do an LS here we can see that I have our ipython notebook file here so let's go ahead and see what this looks like if we open this up within a text editor so I just opened up our notebook file here within sublime text and we can see that it's really just a JSON file that has all of the information in here that we need so that's how these files are actually structured in the background okay so another thing that I wanted to mention about these notebooks is at the beginning of this video I mentioned that we could create multiple kernels using multiple versions of Python so if I do a new notebook here you can see that we have Python root which on my machine is Python 3 you can say I also have this Python 2 kernel here so how do you go about installing a Python 2 kernel now this is another instance where I think it's a little bit more convenient to go with the anaconda installation because with the Conda virtual environment setup all you have to do is install Jupiter within those virtual environments and then they will come up here as an option as a kernel when you create a new notebook so really this Python 2 7 here is just a Conda environment that came with anaconda if I do a Conda env list then you can see that that Python 2 7 is just in here in my anaconda environments and I go over how to create those anaconda environments and my other anaconda installation video now if you didn't go with the anaconda installation then you can install a Python 2 kernel if you go to their website then they have the pip installation instructions that you need to go through in order to install a Python 2 kernel ok and before I end the video here there is one last thing that I wanted to mention is that if you'd like some practice or to see how other people are using these notebooks then you can find a lot of examples online if you do a Google search for for Jupiter galleries one of the top results is this ipython github page now there are also newer Jupiter notebooks on here also but they had made this page before the name change so it still says ipython here but if you go to this page then there are a lot of examples of notebooks that you can actually download directly and play around with so for example I'll just scroll down here to introductory tutorials and after I clicked on that I'll go here to collection of notebooks and from here we see a lot of sample notebooks that we can play around with so they have one about cell magics and things like that so if I get come down and click on this example for a trapezoid rule you can see within this notebook here that they have some mathematical formulas they have some cell that are some cells with some code and they also have some visualizations down here at the bottom so if I wanted to I could actually come up here to the top right of this notebook and download this and so now I have that ipython notebook in my downloads folder so I'm just going to move that to my Jupiter demo I'm just going to say move from my downloads folder and that was called trapezoid rule and I'm just going to move it into the directory that I am currently within here so now if I do an LS then we can see that we have that notebook in our demo directory there so now I'm going to go back to our dashboard and reload this you can see now we have that trapezoid roll notebook so now I can click on that and I will just keep that as the Python 3 kernel and click OK and now you can see that we have that sample notebook that we can play around with on our own so now we can see all of the cells that we saw on the website and if we want to see how they did something so for example if I wanted to see how they created this formula then we can just click on this cell and go into edit mode and we can see exactly how they created that and if I rerun that sell it reruns that code and creates that formula again okay so I think that's going to do it for this video I hope this gave you all an idea for how you can begin using these Jupiter notebooks they're really becoming popular it's a great way to explore your data and your code and an interactive way it's great for displaying plots and charts and pandas dataframes and all kinds of different things and it's also something that's already being used in real research so for example we saw the like Oh notebook at the beginning of this video which showed some of their gravitational wave research and notebook form so it's definitely something that's nice to know how to use now if you do have any questions about what we covered here then feel free to ask in the comment section below and I'll do my best to answer those if you enjoy these tutorials I would like to support them then there are several ways you can do that these just ways to simply like the video and give it a thumbs up and also it's a huge help to share these videos with anyone who you think would find them useful and if you have the means you can contribute through patreon and there's a link to that page in the description section below be sure to subscribe for future videos and thank you all for watchingToday we're going to be learning how to get started with using Jupiter notebooks now if you don't know what a Jupiter notebook is this is basically a way for us to run code interactively within a web browser alongside some visualizations and some markdown text to explain the process of what's going on. Now this Jupiter project evolved out of ipython so if you've ever used ipython or heard someone talking about that then that has now been rolled into this Jupiter project and there were some good reasons behind that that I won't go into too much detail about here but they have an entire blog post on their website if anyone is interested so before we even get started let me show you why this is useful and how powerful something like this can be so a lot of scientific institutions are using these notebooks in order to clearly explain exactly how they got the results and not only can these notebooks show us how they got the results but we can reproduce the results from within the notebooks themselves so some of you may have heard of like oh that's the observatory that detected gravitational waves in the late 2015 they actually put out some of their research and notebook form so I have one of their notebooks pulled up here which walks you through how you can replicate some of their processing using their own data and you can see that this isn't just Python code but they also have some markdown code which gets displayed as the instructions and also describes exactly what's going on here but you can see if we scroll down here we have a lot of markdown code that's describing what's going on we see some Python code and we see some charts here so the way that some of these notebooks are laid out it almost looks like a blog post or a textbook or something like that but these charts and everything that we see here these aren't just static visualizations like some PDF file or something like that these are visualizations that were produced right here in the browser by the code that we're looking at and we can even tweak this code and rerun these cells to get different results so for example here we see this chart that was produced by this code here and I've actually come into this cell here then I can change this x-axis label instead of an S there I can change that to seconds and if I rerun that cell then you can see that this plot gets recreated right here within our browser interactively so you can see how that's a really cool concept of being able to interact with the data and the code in this way and you can see why notebooks like this would be becoming very popular so that's what we're going to do in this video we're going to learn how we can get Jupiter installed and how we can create our own notebooks get set up with different kernels what they see within these notebooks and also how to navigate around execute code and also some cool tips and tricks okay so let's go ahead and get started with installing Jupiter so I have the Jupiter documentation pulled up here on their installation page and you can see that they actually recommend installing Jupiter by using the Anaconda Python distribution and I have a separate video where I walk through the installation of anaconda and some of the benefits that you get from that and one of those benefits is that Jupiter comes bundled in with that installation so I'm going to use Jupiter for this walkthrough using the anaconda distribution but if you don't want to use anaconda then they also have the PIP installation instructions here as well so either method is up to you but I would recommend installing anaconda and that's what I'm going to use for this video so once you've run through the process of either installing anaconda or doing the PIP installs then now we can just pull up our terminals here and I'm going to kill that notebook server that was running the ligo notebooks and I'm going to CD into a blank directory here that is completely empty so now I'm starting from scratch in this directory if I do an LS dash L a on a Mac this is going to list out all the files and directories and you can see that it's completely empty so now within this demo directory to start a new notebook we just have to say Jupiter notebook and if I run that it's going to start a server and pop up with this dashboard here and this is on our localhost port 8 8 8 8 now that server that we started up in our terminal you have to leave that running that's called a notebook server and if we were to close that window and shut down that server then we wouldn't be able to access our notebooks within our localhost so you have to leave that running and you can see that currently in our dashboard here that we don't have much here if we had any files or folders in that test directory where we started that notebook server from then those would show up here but that directory was completely empty so we don't have anything here yet okay so let's go ahead and jump right into creating a new notebook and in order to create a new notebook you're going to have to have a kernel and the kernels are basically what programming language that you want to use so for python we could also have a separate kernel for python too and we'll go over how to set that up here in a little bit but for now I'm just going to go ahead and create a new notebook and this is going to say Python root and Python root on my machine is Python three and so I'm going to go ahead and select that okay so now we have a completely blank notebook here now if this is your first time using a Jupiter notebook then I would recommend coming up here and clicking on this help section here and let first let's go to this user interface tour now this tour will give a pop up on your screen that you can just use your arrow keys to scroll through and it will show you where certain things are so there it's showing us how to change the file name for the notebook and how some of these menu bars what the icons mean and things like that and also the mode indicators it will just kind of walk you through how everything is laid out but for now I'm going to go ahead and exit out of that and also let's go up here back to this help menu and now let's click on this keyboard shortcuts now depending if you're on Mac or Windows these modifier keys may be different for you but if you do scroll down here it shows you all the keyboard shortcuts that you can use within either command mode or edit mode and we'll talk about both of those in just a second so I'm going to go ahead and close this down but if you ever do want to see either of those again the user interface tour or the keyboard shortcuts they're right here in the help menu ok so the first thing that I'm going to do with our notebook is I'm just going to go up here and change this from untitled and I'm just going to call this testing Joop and I'll go ahead and hit okay there so now that renames our ipython notebook file okay so now let's actually look at adding some content to our notebook so notebooks have two different modes they have something called command mode and the other is edit mode so command mode allows us to perform actions like adding and deleting cells and you can put yourself in command mode by hitting the Escape key and the edit mode will allow us to edit ourselves and you can put yourself in edit mode by hitting the enter key and we can also just click into a cell to enter edit mode and you can see when I did that we get this green highlight around the cell so that's an indication that we're in edit mode and another indication is up here in the top right you see that we have this pencil icon here if I hover over that you can see that that says edit mode now if I hit the Escape key to go back to command mode then you can see that that pencil disappears and also our cell here turned blue but I did want to edit that cell so let's go ahead and just click in there again and now let's just add some simple code so I'm just going to do a simple print hello world okay so now that we have some code within our cell how do we execute this so if I just press ENTER you can see that I'm just adding more lines to my Python code here so if I want to execute this code then I can come up here to this cell menu and you can see that we have three different options here I can either run the cell run the cell and select below or run the cell and insert below so let's go ahead and walk through these so if I just run the cell then you can see that it executes the code that was in our cell and displays the output below here and the cell that we just ran is still selected now if I come up here and do a run cell and select below if I do that then you can see that execute the code in our cell and then it selected the cell below that one now if there isn't a cell below it which there wasn't then it will just insert a cell there so if I select this top cell again and rerun that then you can see that it didn't add another cell but it did execute that code and then selected that cell below that and lastly if I just choose our top cell here again and do this run and insert below you can see that even though there was an empty cell below here it's still expert it's still inserted this cell below the cell that we just executed so those are the differences between those three different options there and the shortcut keys for those at least on a Mac is ctrl enter to just execute the cell that you're currently in and stay there shift enter will execute the cell and move to the next one so that's executes the cell and select below and the option enter is to execute the cell and insert below so you can see that that inserted another cell below after I executed that content now for the rest of the video I'm going to be using those keyboard shortcuts just so that I can move a little bit more quickly here I'm not going to be coming up and clicking on the cell to run the cell every time so if you see code executing it's because I'm using those keyboard shortcuts now one thing about the Python code here is that this actually behaves kind of like an interactive prompt so I don't actually have to print this out here if I instead just came in here and did hello world just like we would if we started up Python in the terminal or something like that if I go ahead and rerun that then you can see that we get the same output okay so you may have noticed these incrementing numbers here beside the cell where it says in 13 and out 13 here now this is something that you have to get used to when working with these notebooks so these cells don't necessarily have to be executed from top to bottom like we're used to in regular scripts so these numbers show us the order in which these cells were executed so for example if I replace this top code here and I just say name is equal to John and I execute that now currently that doesn't give us any output but in this next cell I'm just going to say name and go ahead and run that can see that the output is John now that's what we'd expect so far but in this next cell I'm going to reset this name value here and I'm just going to say name is equal to Cori and I'm going to go ahead and run that and now if I come back here to this top cell and rerun this code here now you can see that that name variable printed out as having the value of Cori even though we didn't make that assignment until after that cell and it's because these numbers here determined the execution order not just how things are assigned from top to bottom now with that said you still want these notebooks to be readable and easy to understand so I'm still suggest putting things in the correct order from top to bottom but if you ever have a variable that doesn't have the value that you think it should have then you might want to check these numbers here on the side to make sure that your execution order isn't out of whack or anything like that now if you do get things mixed up and you just want to treat this like a normal script and run everything from top to bottom then we can come up here to the sell menu here and we can just click on run all and that will run all of the cells from top to bottom now if I rerun that then you can see that it prints out all of these numbers here are 1920 they're all in order from top to bottom and our name variable is what we would expect in a regular script and also you may have seen in that cell drop down here that we also have the options of being able to execute all of the cells above or below our current cell so if you have some cells that take a long time to compute and only want to re execute all of the cells below that point then you have the flexibility to do that okay so that's how you execute some basic code from within the cells so now let's look at how we can add some markup so markup is text that gets translated to HTML and we can use it within our notebooks to explain exactly what's going on and we saw a lot of that and the ligo example at the beginning of the video so let's say that I wanted to add some markdown to the top of this document here so first I'm going to select this top cell and now I'm going to add a new cell to the top like insert and then insert cell above and now with that top cell selected I'm going to come up here and click cell go down to cell type and I'm going to choose markdown now this isn't a markdown tutorial so I'm not going to go into detail about exactly what markdown text gets translated in HTML but I'm just going to drop in some sample markdown from a snippets file that I have pulled up here so I'm just going to grab all of this sample markdown here I'm just going to paste this into our top cell now you can see that while we're still in our edit mode here it kind of gives us a little sample of what we're going to get once we actually execute this cell but we still have our markdown here that hasn't been translated completely yet so I'm going to go ahead and just run that by hitting ctrl enter then when we run that you can see that our markdown got translated to HTML so we have a header here a header to a list and we have some italics and bold and things like that okay so that's how you can add markdown to a notebook now let's take a look at some other useful features that we can use within these notebooks so our current kernel here is using Python so any normal code that we add to our cells here will be interpreted as Python code but there are some special commands that we can use within our cells that won't just be seen as Python code so for example here I'm going to add another cell here below this so one of these special commands is if you put an exclamation point then it will interpret this command as a bash command so if I do exclamation point and do like a pip list and run that then you can see that the output that you get from that would be the same as the output if you were to run that command from both in the command line so that's kind of cool that you can run these bash commands from directly within the notebook now these notebooks also come with a bunch of built-in commands that they call magics so if you've looked at notebooks before then you may have seen these these are the ones that begin with either 1 % or 2% signs so the single % that means that the commands arguments will all come from that same line and those are called line magics and the two percent signs these mean that the entire cell will be used as that commands arguments and those are called cell magics so we can actually use one of these magic commands to list all of the other available commands and the command to list all those is called LS magic so if I put a percent sign and do LS magic and then run that cell so now you can see that lists out all of these magic commands that we can use and it splits these up into line magics here and cell magics here so let's go ahead and take a look at a couple of these so I'm going to go ahead and insert a cell below here so that we can still see all of these magics as we're working with these so now if I wanted to print out my current working directory then I could just use a single % here and do a PWD for print working directory if I run that then you can see that it shows the directory that we're currently in and you can see in this list of available line magics they also have an LS here so I can list the files and folders within that directory by doing an LS and if I run that you can see that currently we just have our single ipython notebook file now I can also pass normal commands to these just like I would within a terminal so if I do an LS - la then it'll do the long-form of those and you can see that there are more files and it lists out the users and permissions and things like that okay so those two commands that I just showed you are just similar to just being able to run bash commands within our notebook which we've already looked at but there are some of these magic commands that give us some functionality beyond just the bash commands so one of the more common magic commands that you'll see within notebooks is the matplotlib inline command and that is mat plot lib and then end line so what that does is it allows matplotlib charts to be displayed within our notebook so I'm actually not going to execute this yet and I'm going to execute your insert a cell below here and first I'm going to grab some sample code here from my snippets that will create a very simple map plot Lib chart so I'm just going to go ahead and copy that and paste that in here now this sample here is just some snippet code that I got directly from the matplotlib documentation so if I paste this in here and try to run this without first running this in line command so I'm going to execute this and try to show this plot and you can see that this is just sitting here spinning it's not doing anything and if we look over here this asterisk just means that it's still trying to run the cell to completion so I'm just going to kill this by coming up here to Colonel and I'm just going to do a restart and clear output and go ahead and hit YES on that so now I'm going to try to run this again but first I'm going to execute this matplotlib in line magic command first so if I run that and now you can see that it executed with that one there now I'm just going to go ahead and try to display this chart within our notebook so now you can see after we ran that setup command there that our plot shows up directly in the notebook now this is extremely useful so this is where we really start to see why these notebooks are becoming really popular because to be able to show plots and things like that as we're stepping through our code it really helps you explore your data and code in real time without needing to rerun the entire script okay so let me show you just a couple more of these magic commands that I think that you'll find really useful when you start using these notebooks so we can see that one of these sell magic commands here is this % % - % HTML and that will allow us to render HTML directly without using the markdown that we looked at before and this is a sell magic command which means that the entire sell will be rendered as HTML and not just the single line so just to show you an example this I'm going to go back to my snippets code here I'm just going to grab an iframe and this iframe is just going to pull up one of my other YouTube videos so if I go ahead and run this cell then you can see that we can actually embed this iframe directly in our notebook here and really you can use that to render any kind of HTML that you'd want so you can render images or links or anything like that directly within the notebook here so I'm not going to show an example of all of these commands but I do kind of want to highlight a couple of things that you can do so you can see that they also have a JavaScript command here so that would be everything that you wrote in that cell would get interpreted as JavaScript code you can see that they have a bash one here so that you can actually write out a series of bash commands and one so now there's also a useful magic command here that we can use to time our Python code and that is this time it command here so if I grab that I'm going to go ahead and insert another cell here at the bottom and I'm going to use this time at command now you may have used time it and your regular Python scripts before but this allows us to easily add that functionality into our notebooks with just this one short command so let's say that you wanted to time a list comprehension where you squared the numbers up to a thousand or something like that so I think I have that example yeah so I have that example here in my snippets so I'm just going to go ahead and grab that and paste it in and now if we run that cell then you can see it's actually going to run that time at command on our code there and then you can see that the output prints the output from that time at command and tells us the average time that it took to execute that so that can be really useful if you want to compare the execution time of different functions and things like that you can just add that time at command to the top of your cell and run those for multiple functions and get the output right below yourselves and like I said there's plenty more that you can do with those magic commands so if you do just run this LS magic command then you can see all the ones that are available for you here and you can just play around with any of these that you'd like now these Jupiter notebooks also have the ability to render certain things within the notebook without any kind of magic commands so for example we can also display a panda's data frame from directly within a notebook now I thought I had a snippet for this but apparently I don't sell just go ahead and write this out really quick so if we want to import pandas I'll do import pandas as PD and import numpy as MP and now I'm just going to create a data frame here with some random value so I'll do a panda's data frame I'm just going to put some random numbers in here so I'll do MP dot random dot Rand in and I will pass in ten and five there and then underneath here I'm just going to go ahead and print out that data frame so if I run that code then you can see that just by printing out this data frame here it actually outputs that directly within our notebook and a nice readable form and so if I was to do a DF head there and rerun that then you can see it does the same thing with just those first few rows now if you don't know what these pandas dataframes are that's not a big deal I just want to show you how you can display certain things within the notebooks that may be hard to visualize or display using a normal icon script okay so another thing that I wanted to go over as far as these notebooks goes is that you can export these notebooks in several different formats so if I wanted to export this notebook in HTML so that I could add it to my blog or something like that then I could just come up here to file that could come to download as and you can see that we have a lot of different options here we could download this as a Python file I want to choose HTML so I'll download that as an HTML and if I open up that file I just downloaded that you can see that we have an HTML version of this now these cells here I can't edit these since now it's HTML but I could post this in a blog or something like that and you can see that it's nice and readable has all of our code and all of the output now if you actually wanted to share this notebook file with somebody then what is actually in these notebook files so let me go ahead and open up one of these up in that's not the terminal window I want I'll go ahead and open up this blank one here so now if I do an LS here we can see that I have our ipython notebook file here so let's go ahead and see what this looks like if we open this up within a text editor so I just opened up our notebook file here within sublime text and we can see that it's really just a JSON file that has all of the information in here that we need so that's how these files are actually structured in the background okay so another thing that I wanted to mention about these notebooks is at the beginning of this video I mentioned that we could create multiple kernels using multiple versions of Python so if I do a new notebook here you can see that we have Python root which on my machine is Python 3 you can say I also have this Python 2 kernel here so how do you go about installing a Python 2 kernel now this is another instance where I think it's a little bit more convenient to go with the anaconda installation because with the Conda virtual environment setup all you have to do is install Jupiter within those virtual environments and then they will come up here as an option as a kernel when you create a new notebook so really this Python 2 7 here is just a Conda environment that came with anaconda if I do a Conda env list then you can see that that Python 2 7 is just in here in my anaconda environments and I go over how to create those anaconda environments and my other anaconda installation video now if you didn't go with the anaconda installation then you can install a Python 2 kernel if you go to their website then they have the pip installation instructions that you need to go through in order to install a Python 2 kernel ok and before I end the video here there is one last thing that I wanted to mention is that if you'd like some practice or to see how other people are using these notebooks then you can find a lot of examples online if you do a Google search for for Jupiter galleries one of the top results is this ipython github page now there are also newer Jupiter notebooks on here also but they had made this page before the name change so it still says ipython here but if you go to this page then there are a lot of examples of notebooks that you can actually download directly and play around with so for example I'll just scroll down here to introductory tutorials and after I clicked on that I'll go here to collection of notebooks and from here we see a lot of sample notebooks that we can play around with so they have one about cell magics and things like that so if I get come down and click on this example for a trapezoid rule you can see within this notebook here that they have some mathematical formulas they have some cell that are some cells with some code and they also have some visualizations down here at the bottom so if I wanted to I could actually come up here to the top right of this notebook and download this and so now I have that ipython notebook in my downloads folder so I'm just going to move that to my Jupiter demo I'm just going to say move from my downloads folder and that was called trapezoid rule and I'm just going to move it into the directory that I am currently within here so now if I do an LS then we can see that we have that notebook in our demo directory there so now I'm going to go back to our dashboard and reload this you can see now we have that trapezoid roll notebook so now I can click on that and I will just keep that as the Python 3 kernel and click OK and now you can see that we have that sample notebook that we can play around with on our own so now we can see all of the cells that we saw on the website and if we want to see how they did something so for example if I wanted to see how they created this formula then we can just click on this cell and go into edit mode and we can see exactly how they created that and if I rerun that sell it reruns that code and creates that formula again okay so I think that's going to do it for this video I hope this gave you all an idea for how you can begin using these Jupiter notebooks they're really becoming popular it's a great way to explore your data and your code and an interactive way it's great for displaying plots and charts and pandas dataframes and all kinds of different things and it's also something that's already being used in real research so for example we saw the like Oh notebook at the beginning of this video which showed some of their gravitational wave research and notebook form so it's definitely something that's nice to know how to use now if you do have any questions about what we covered here then feel free to ask in the comment section below and I'll do my best to answer those if you enjoy these tutorials I would like to support them then there are several ways you can do that these just ways to simply like the video and give it a thumbs up and also it's a huge help to share these videos with anyone who you think would find them useful and if you have the means you can contribute through patreon and there's a link to that page in the description section below be sure to subscribe for future videos and thank you all for watching