Engineering Statistics
Professor Manjesh Hanawal
Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay
Lecture 29
Introduction to Python

(Refer Slide Time: 00:32)

Okay, hello everyone. Today we are going to look into some programming aspects of this engineering statistics. So, as we know or as we discussed so far, we are talking about various distribution, data generated from distributions and if we are given a data we want to see that what is the underlying distribution that is generating this data and all.

So, to do this we may need some tools to or mostly the software tools to analyze this data and for that we are going to talk about Python today which is one of the languages widely used in statistics. Okay, so, we are just talking about Python, there are other possibilities like one can use Matlab, one can use R and other languages. But here I am going to talk about Python.

(Refer Slide Time: 01:31)

So, to begin with, we will talk about how to get the things running for us. So, first I will talk about, very briefly about how to install and how to set up our environment. So, Python is very easily accessible, it is open sourced, you can just go to the python.org websites and just download the latest version.

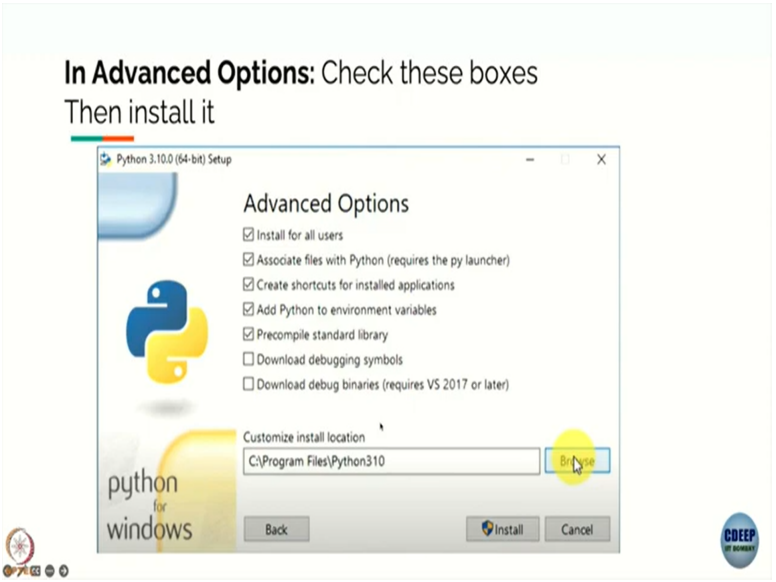
(Refer Slide Time: 01:49)

So, here when we try to do this, the version 3 dot 10 dot 6 was the latest version and you can choose whatever the latest stable version available, when you visit the site, you can download that and here we are talking about installation for Windows. But you can follow the instructions given on the Python web page to get it installed for if you are using the either Linux machine or Mac OS machine.

(Refer Slide Time: 02:21)

So, once you do this, you will just to follow the instruction and you will let check this options like you need to set this to be set up in certain paths and give access to other users use or like a standard, checkboxes you do this and go for customized installation.

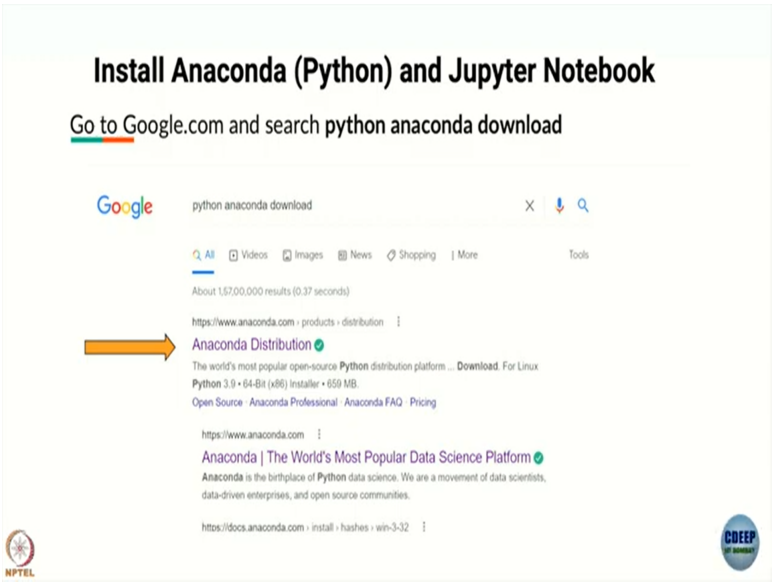
(Refer Slide Time: 02:44)



And then maybe the installer asked you to specify the location where you install it either choose a default one or if you want to put it as a separate location just specify your location and once you are done, this is done.

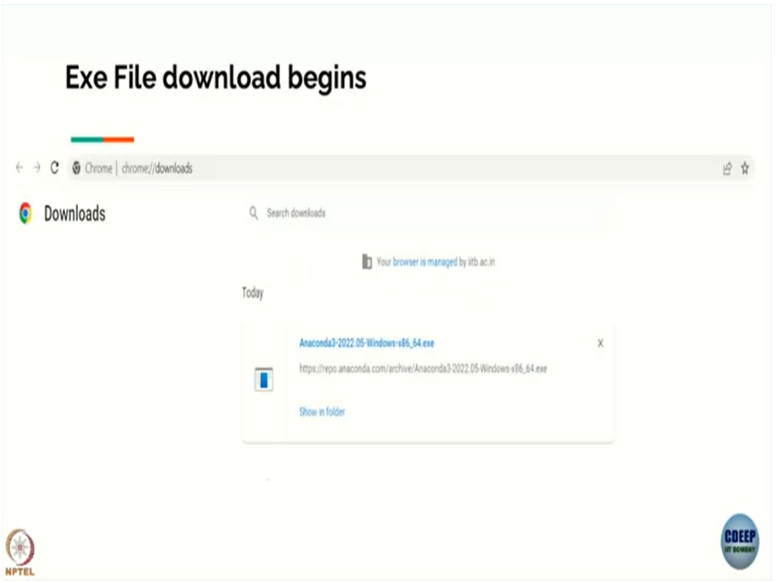
So, your Python gets installed. Now with this you have a Python, you may just launch your terminal and run your code there whatever your code okay but often we want execution the code to be more interactive and maybe if we have something like an IDE kind of environment is there then it is better. So, for that we will look for what options we have.

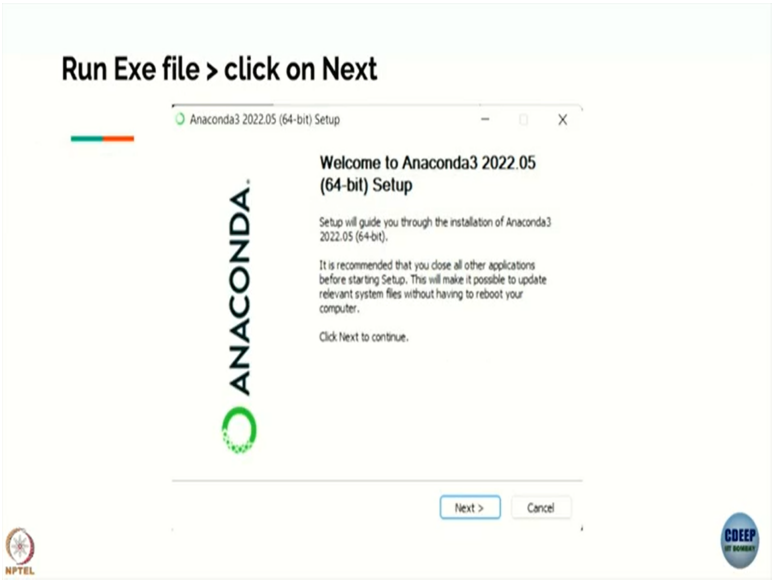
(Refer Slide Time: 03:27)



One of the popular option is Anaconda and Jupyter Notebook. So, Anaconda is basically a one of the open-source Python distribution platform. So, it supports maybe other languages also not and not necessarily Python. It is like a very comprehensive tool. You can use it.

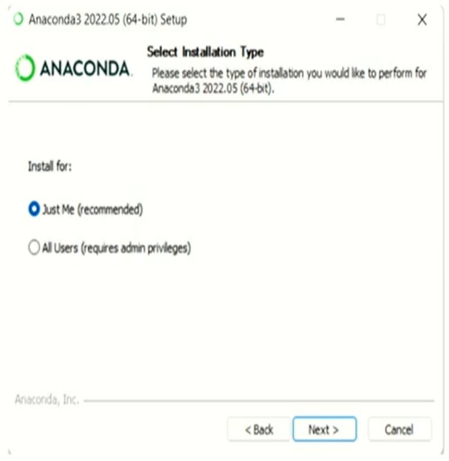
(Refer Slide Time: 03:57)



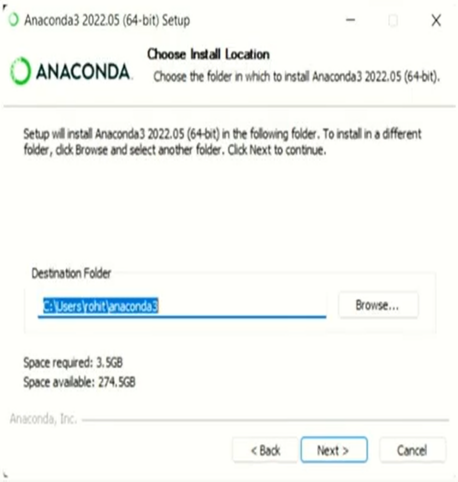


Click on I agree

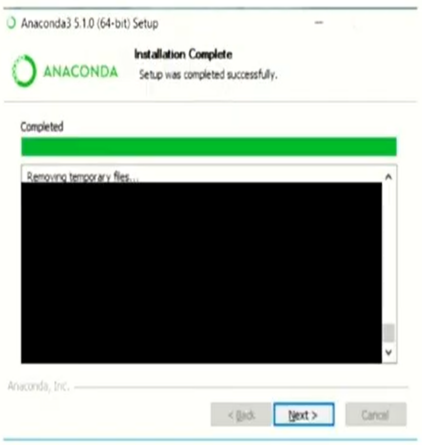
Click on Next



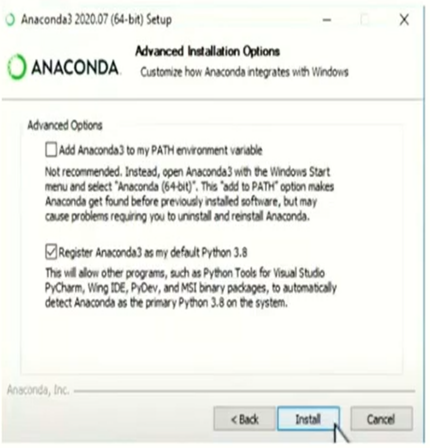
Let it be the default destination > Click Next



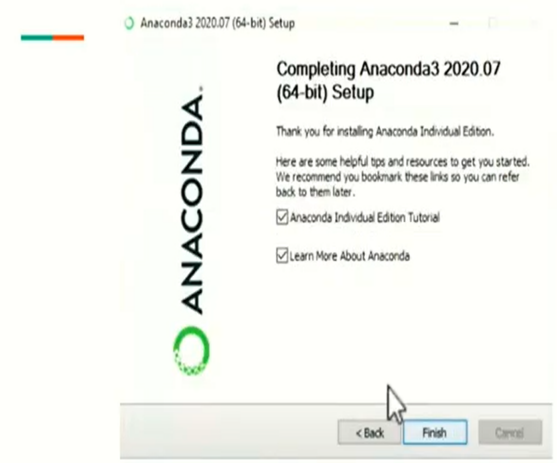
Click on Next



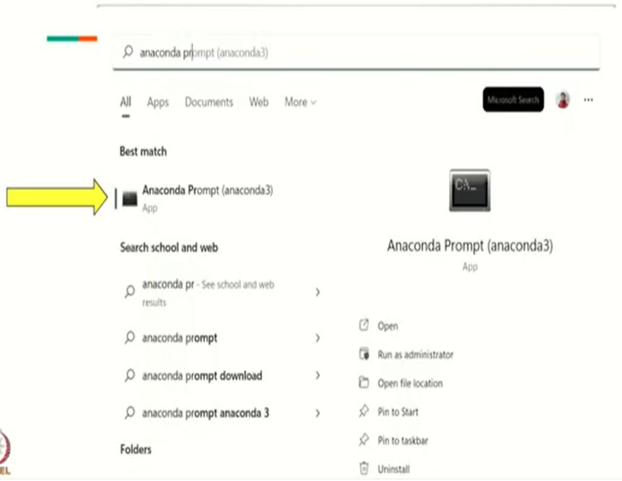
Click on Install

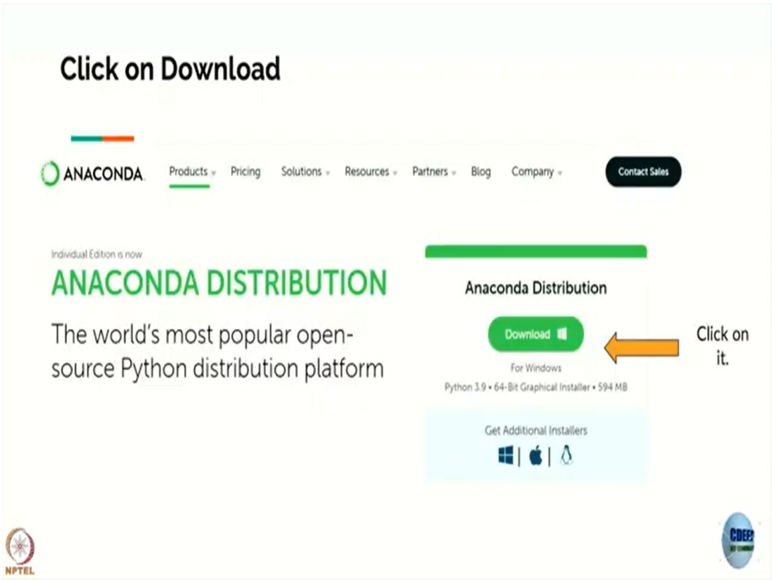


Click on Next > Next > Finish



GO to Search bar, type Anaconda Prompt

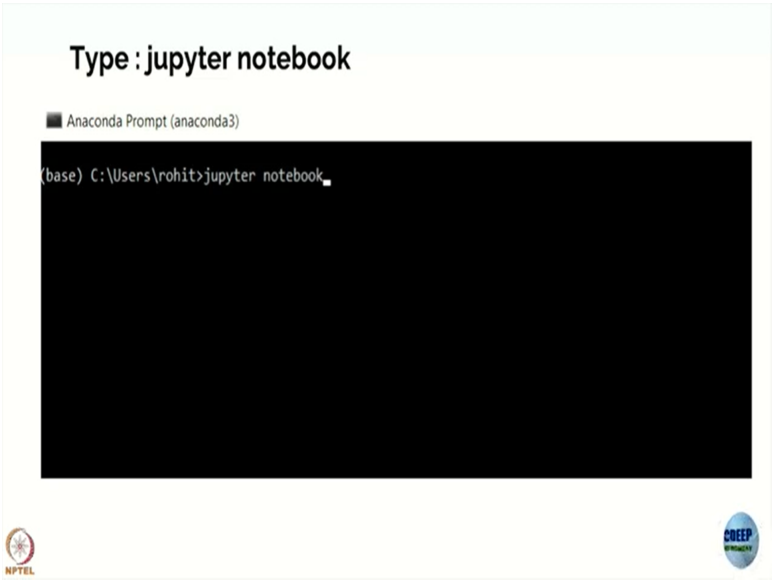


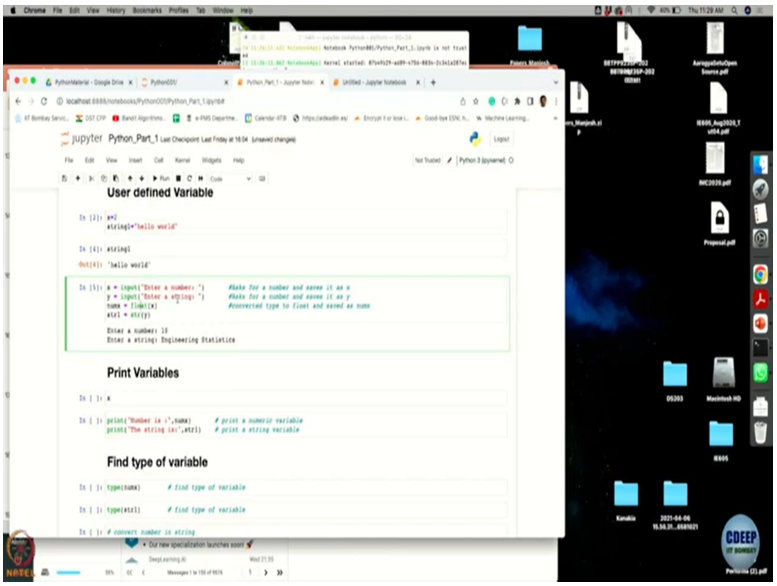


And when you use Anaconda, it comes with many ideas I will come to that but just let us quickly follow the steps here to get your Anaconda installed and here you can just go to the website of Anaconda, click the download and just follow the steps get to get your Anaconda installed.

Again, it will the installer will ask you to give you the path and it will you have to check certain options. Choose the options as necessary for your machine and check you need to also be careful whether your machine is a 64 bit, has a 64-bit processor or 32-bit processes accordingly you select and then you are ready.

(Refer Slide Time: 04:46)





As I said Anaconda is like a platform. It comes with various IDE tools like Eclipse, ninja, and I do not know maybe there are several others but one of the popular IDEs in it for running Python code is Jupyter Notebooks and that is what we will be using. That is what we will be discussing in this tutorial session.

So, now once you have the Anaconda successfully installed, what you need to do is you need to open your terminal. Let me see my terminal is over or maybe I will just come out of this presentation mode. Now, I will open my terminal here and in that I will just simply type Jupyter Notebook and this will open Jupyter Notebook. And here Jupyter is a web-based IDE. So, when you write this Jupyter Notebook, it will directly open inside the web browser.

Now, here, in whichever directories you want to write the code or you have already written the code, go to that directory. So here, the codes that are put are in this Python 001 folder. I am just going there. Okay and now I have a notebook already created Python 1, when you click, it will open this where I have already written some content which I am going to discuss. And if you want a new Python, sorry, a new notebook, you can just go here and open a new book here and do all your code executions here.

Okay, so let me go back to the ones I have already created. So, the first thing in any programming language is just to understand how to create variables and strings and Python gives very easy and intuitive ways of creating these variables and strings. Okay, so let us say if I want to create a variable x for which I have to assign a value to I have to just write x equals to true. And if I had to create a string of word hello word, I have to put it under this inverted commas and assign it to a variable which are denoted as string 1 here.

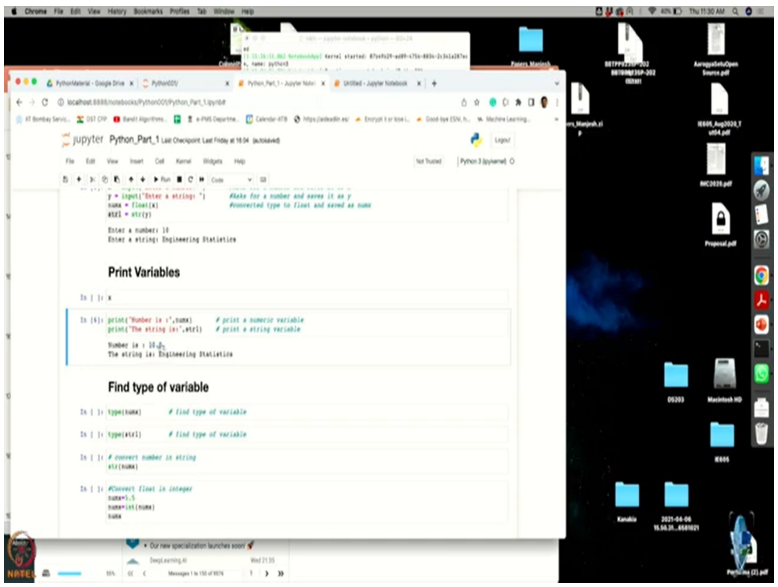
So, let us see how does this work. Now I am going to run this when to run either you can click here, or you can use some shortcuts, like I am using MacBook here, there my keyboard shortcut is to shift and enter, then it will run. And let us see if I want to insert another cell here, I have to just click on this plus symbol here and write whatever the code I want to so let us say I want to see what is that x I have assigned. And now I am going to simply execute by pressing shift and enter. So, it shows me the value that is assigned to the variable 2. And now if I want to see what is that is assigned to string one, again, I write string one here and write enter. And you will see I see that the string one has the string hello word.

And also, Python gives us a very easy way to accept inputs. Suppose like I want to instead of assigning value x directly here, I want user to give a value here so I can for that I can use users through this function input. Okay, let me execute this and let us see what happens. So, now see that it is asking me to enter a number let me enter number 10 and then for the next step, it is asking me to enter a string, let us say engineering statistics. Okay, and now I again press enter. Now these two outputs are shown.

Notice that now I have another return, I have basically converted this number x to float and store it under num x. So basically, if I want to store this numbers in a particular type, then I will just use that type and then store them. And here also like because like users here just have taken input user even though I asked to enter a number if let us say user has entered some characters or maybe he just entered a number. I want to store it under float, so I use it. And suppose when I asked him to enter a string so numbers are also some kinds of strings.

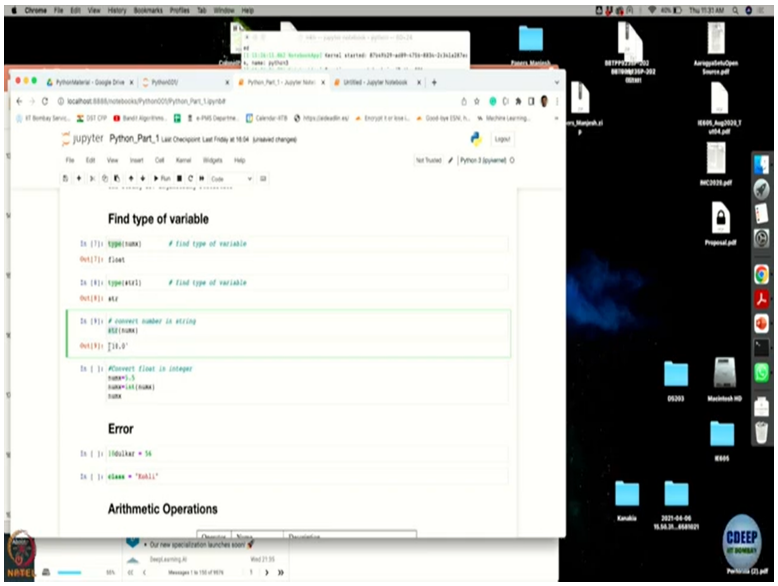
So, if users has entered the numbers as string instead of interpreting them as numbers, I want to interpret that string that is what I use this function string and then store in this variable STR one.

(Refer Slide Time: 10:17)



Okay, so now let us see what I will get if I print this num x and string one. So, notice that since I have stored it as float, it is taken stored the value I entered 10 as 10.0 and a string since I entered a proper string it I just stored it as a string.

(Refer Slide Time: 10:47)

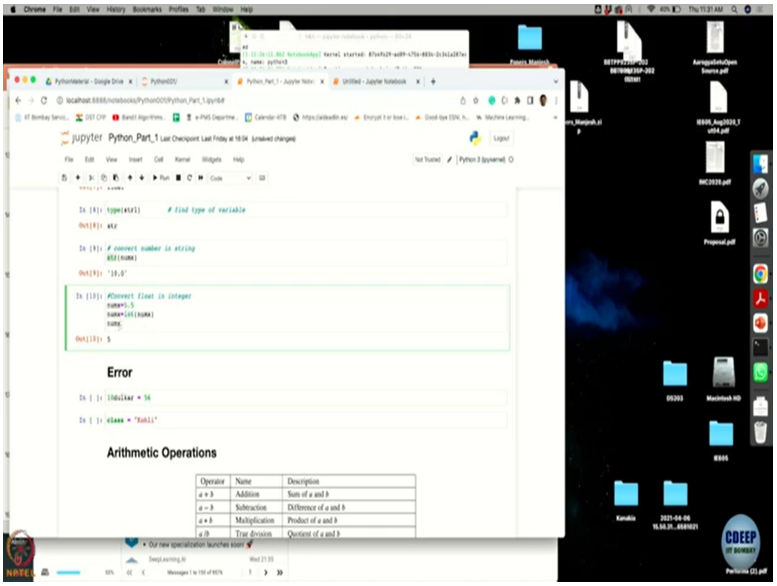


Okay, and now you can go back if you are just have a variable and you want to check what is this type, you can use the function type here to get what kind of variable it is and the for

string you will see that string one you see that it is just string 1 and here I can convert this number to string.

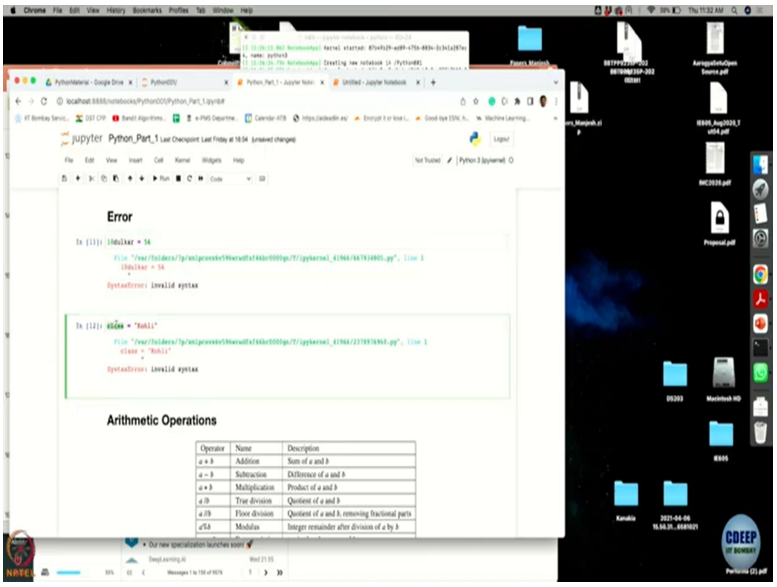
See that I use this function string on this number x now, it is showing me under this inverted commas that means when it is coming as inverted commas this is clear that even though 10 is an integer, but now it is treated as a string here okay.

(Refer Slide Time: 11:24)



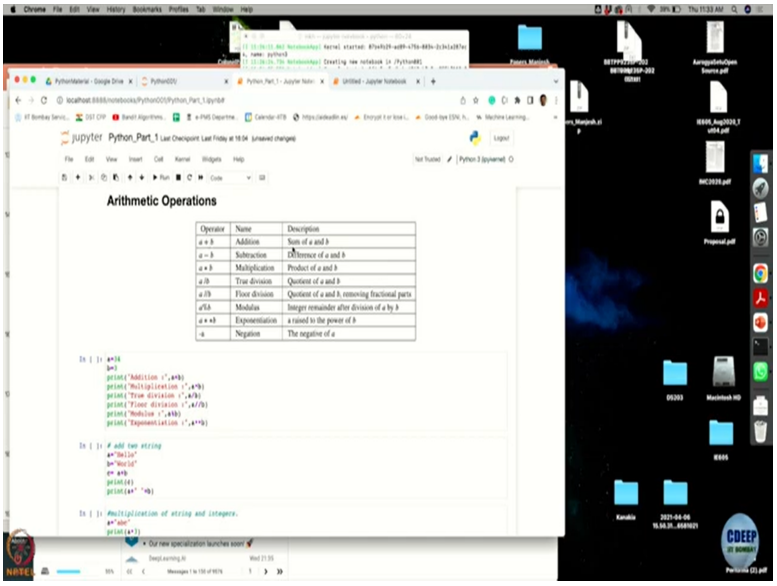
And now, we can do this conversions between various types. Suppose if I have a number which is like float or which has been given to 5.5 but I wanted to convert to integer, I will just apply this function integer and then when I see what it is, you will see that it is just shown as 5 and when we have to assign some variables, we have to be careful names to the variables we have to be careful.

(Refer Slide Time: 12:02)



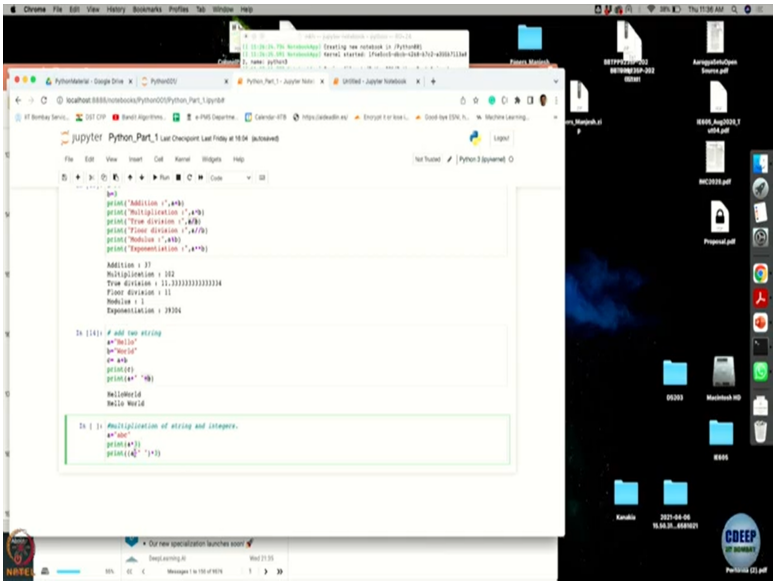
It is not that anything we can take as variable name, for example, this I want to define a variable called this Tendulkar but here 10 I have replaced by the integer 10. And I want to store in that variable value 56. If I do that, it gives me an error because this is not permitted. I cannot use numbers and characters in defining a variable. Similarly, if I want to save this name Kohli under the variable class, this is not accepted and Python will show me an error because this class is a kind of a reserved variable. Okay, so we have to keep these things in mind like I cannot mix up numbers and characters in defining variable. And similarly, I cannot use some reserved names for variables.

(Refer Slide Time: 13:03)



Python allows us to do various operations and the good thing is like we can do additions and all in a very intuitive way without the syntax it is so simple that we while writing this code, it is as if you are writing something on a paper like if you want to do this, add two numbers a and b it is simply we have to write it as a plus b and if you have to subtract two numbers, we have to simply write a minus b and if you have to multiply two numbers a and b it is simply a star b and if it is a division it is going to be written as a backslash b and it is division with the floor that where we ignore the fractional part then it is a two backslashes divided b and if you are only interested in the modulus then it is a percentage b and if you want to exponent a to b, you have to use a double stars b and if you have to negate the number it is simply minus.

(Refer Slide Time: 14:13)

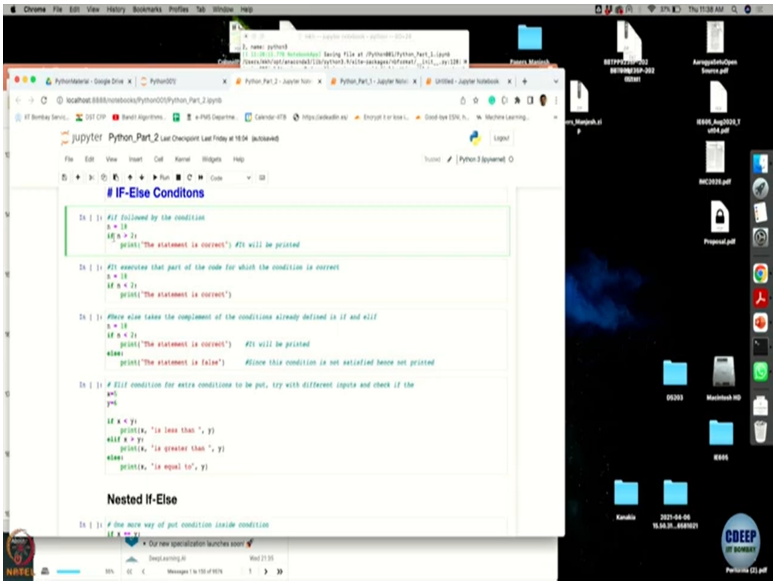


So here are some examples. Let us say I have this two values which I have stored in a and b here. a is 34 and b is 3 like here you see that like I can simply do the addition by doing this and I can print a value. So, this is as simple as that. So, notice that when I did a by v here that is the true division I got 11.333 like that. But when I do a backslash backslash b that is the data flow division, I only get it as 11 the integer part was ignored. Okay, I hope these things are clear and also, Python makes handling of the strings also pretty easy and very intuitive.

Suppose I have two strings, let us say one is two words Hello and another one is word and I want to do concatenation of them. It is like as if I am adding these two words or strings. So, I can simply say a equals to c, a plus b. And when I print it, you will just see that Hello word, C, but when he did a plus b, just concatenated them back-to-back. But if you want to add some space between the two words, one option for that is like you add space by using this plus symbols here.

So, what it first did is like when I use this, when I use this plus here, it is adding space to a and after that it is adding B to that string. So that is what you will get a word which looks like this Hello with a space and word. Multiplication is also pretty straightforward. Even if you want to use it on strings, suppose I have a string ABC, then if I want to appear the strings three times, all I need to do is take this A into 3 and if you want to add a space and then to multiply and add, multiply, and repeat them three times I have to add space, this part will adding space and then repeat it three times. So, this is how the output looks. So, these are like a basic, how to create variables and how to do some basic operations on numbers as well as strings.

(Refer Slide Time: 16:58)



Now if you want to do now, if you want to do coding, maybe we have to do a lot of repeat a task over numbers for that we need to check certain conditions or do some looping operations. So, let us look into the conditions how to check and some looping operations. First, let us look into the if else conditions, how it will work in Python. Suppose if you have a number n has certain number and if I want to check whether this n is greater than 2, all I need to do is write it like this, check whether n is greater than 2 and notice that after that, I have this colon here and then I am going to check and write this like print the statement is correct.

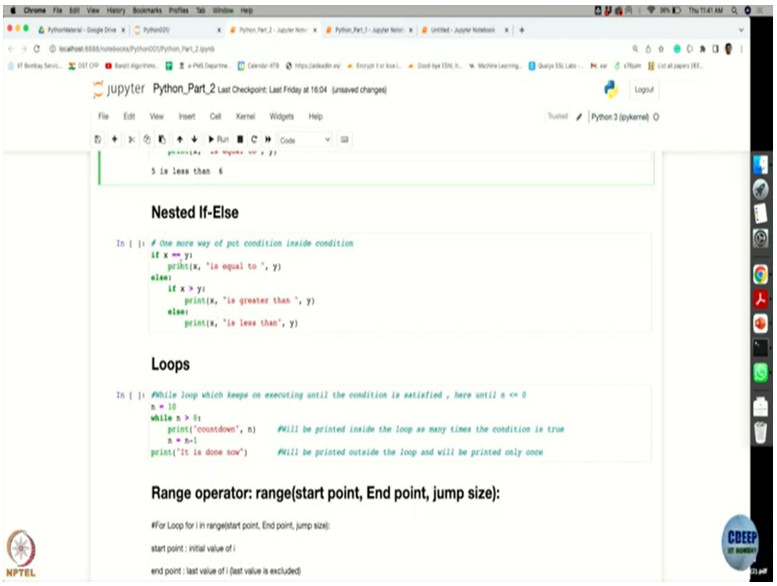
And after that, I do not need to write anything here. So, the one important thing is after I write n is greater than 2 complete that with this colon and then print the statement but you have to properly index it, align it, okay. And for example, if you after writing this, you do this, Python will automatically take where you should be writing the next condition here.

Okay, now let us write this now here. Since we have already put 18 in n, this is true and that is why the statement is coming out to be correct and similarly, if you want to check whether n is less than 2, I will do the same thing but now if you execute it see I am executing by compressing Ctrl sorry shift return, but nothing is executed because the condition is failing.

And now, if you want to check we want to now check both like which of them is correct, whether like the condition n is less than two holds or not, we can first check with n equals to less than 2 put a colon there then print statement the statement is correct if that is true else. So, when I say else, it is always going to be taken as a compliment of this condition. Whatever the condition you have written for the if statement and again you have to end it by a colon and then print the statement is false.

Okay, now this could be expanded. It is not that else can take any condition here else could also take a condition. But in that case, we have to write it as else if. So, this is same thing we do in other language like C, C++. So, if you want to check whether two numbers x and y, let us say you want to check the condition x less than equals to y, you check it and if that is true, you want a statement like x is less than y and if x is greater than y that also you want to be stated and if none of these two holds, then the only possibility is that both are equal and that you are going to write in the last case. Now, we can also do so let me execute this. Now, it is clear that since x is having value y 5 and y is value 6, 5 is less than 6 here is the correct value.

(Refer Slide Time: 20:54)



And now we can do the nesting of this under else conditions also we can put further if else. For example, if I want to check whether two numbers how they are first I can check whether

they are equal. If they are equal, I am going to print x is equal to y else there are two possibilities either x can be greater than y. So, I write this and write the one to print it x greater than y. Else if x is not greater than y, obviously it has to be less than y. I am going to still print like this.

So, notice that the things is like this has to be properly this lines has to be properly indented. You do not need to write like end every time like we do in C or C++ language. Here this end is automatically inferred by looking into that indentation. Now loops. So, we will stop here and we will continue discussion of loops in the next module.