

## **Radio Astronomy**

**Prof. Abhirup Datta**

**Department of Astronomy, Astrophysics, and Space Engineering**

**Indian Institute of Technology Indore**

**Lec-24**

**Python Crash Course**

Hello everyone, I am Harsha Avinash Tanti and I am currently a Research Scholar at the Department of Astronomy, Astrophysics, and Space Engineering and also I am your Course TA for this Radio astronomy course so, welcome again. So in this tutorial lecture what we will do we will learn something about python. Which is a programming language

and what we will do in this lecture is we will introduce python language as a computational and visualization tool not as a programming language but like a computational and visualisation tool. Okay! So to begin with what we can learn here is How to use python as basic calculator then we will also learn how to use python as a visualisation tool which we will use this for plotting. Plotting means physical representation of any data. Okay, and we will also learn different data types. So, Data types just refers to kind of number system okay and also symbols like we assign character or string or something like that to a variable okay. So, if you have a string 'Hello' like what I am writing here it will be stored somewhere in some variable which we can use to display on the screen. So, this is what we will do in this particular session okay. So, let's not waste our time and begin with the crash course also I will show you typically whatever we write in the python code okay. It is nothing but a representation in terms of kinds of equations and commands. Also for this we will be using a free Jupyter notebook by Google - Google Colab. So, Google colab we can use to understand how to use python and there we can write line by line and what are our results. Okay! So let's begin.

So, what will we learn, again here what I will say about what will we learn, that are different data types, so data types are basically what we said numbers strings okay, then printing on screen, then list - there is something called list in python its just like grocery list we create is called list in python, we will discuss all these things what are all these things okay. So numbers as like we have learned in our elementary schools also in class 11 and 12 upto our higher educations there are different kind of number okay like complex numbers, then there is decimal number, Real numbers also we have integers. So, these are different kind of numbers you have okay. The same thing you have in python okay. There are certain limitation if approach it from a computer science manner but we will use this as just a simple computational and visualisation tool okay. So, How to utilize python effectively we will learn here. Okay, so lets first here the numbers - what are numbers? So, 1, one is a kind of number. Now, how do we add number here okay, so to use this as a simple calculator we can do  $1+1$  okay then we have to just click here run. I already ran this but

for demonstration purpose what I can do is to run this and by running this it will connect to the server here you can see this one so when you open the google colab it allots you a certain amount of memory as well as a certain amount of RAM now by this what you can do you can get into the allotted computational device, a free computational device where you can run the python program okay. So, if I change it to 5 or say 5.8 so it will show the decimal value 6.8 similarly we can multiply divide and we can do power operation also. In power operation we generally annotate it as (for those who do programming in other languages) is like  $^$  but in python `**` (double asterisks) means you are doing power operation. Say, 2 to the power 4 is what 16 if you run this you will get 16 okay. So, just to verify it, you can do it with 6. So you will get 64 and also you can do decimal power also. So 2 to the power 0.5 is 1.414.

And then there is a modulo operand. Modulo means to verify whether the number is perfect division or not or whether it keeps a remainder. Say here if we divide 4 by 2, then you will get 0 remainder. If we divide by 3, then we will get remainder 1. So this operation, the percentage operation is called modulus operation which shows that a remainder is, what is the remainder? Okay. Say if I do 1, 1 it will say 0. Okay. If I say 5, 4 because 5 cannot divide 4. So 5 is larger than 4. So entire 4 is the remainder part. Okay.

So now coming to the means, community when other properties like it will follow the bad mass rule. Okay. So, that means badmas rule, brackets, addition, multiplication and all those things. Okay. That whatever rule we learnt in our higher education, that rule it will follow. Okay. Now, coming to an interesting part, variable. So in linear algebra as well as the arithmetic, what we do is we assign variables to different quantities. So here also we can do that. Say we can define a variable as name of the variable and then equal to what its value is like here. I can say x equals to say 6 and y equals to 7. Okay. Say 7 point, some value. Okay. Now, you will run it, x and y has been assigned with the certain value, the given values. So with this, if I say z equals to this and just to see the value what is stored in z, we will, we can just run z and we can run this code, we will get 13.957 Okay. So this is how you do general simple arithmetic. Now coming to the other data type which is called strings, you can denote string, have strings in single quotes as well as double quotes. Double quotes provide with a leverage where you can use single quotes in the text. Okay. Now similarly with the strings also, you can define a variable and through the variable you can, using variable you can see what is in the variable.

Okay. Now there is something called print command. Print command what it does, it displays what is inside the variable onto the screen. This and, but the question will come here. This line also shows thing, correct? See x, if I run this we will get hello, right? Say let us do this one. Okay. So hello. Okay. If we run this we are getting hello and if we run this also we are getting hello. Okay. But the difference is here in this particular line it is showing what is inside the variable, what is stored inside the variable. Whereas print command is saying that to show whatever inside in the variable on the display. This is only for the compiler or interpreter to understand. This there is something called compiler interpreter. Interpreter what it does is it executes only single line of code. So if there is a variable it will show what is inside the variable.

Okay. Now whereas print it will display onto the screen if you run this code like some kind of

script just humor me for a second means there is something called script where you can write entire thing and run it on a single go. You do not want to run step by step just like what we are doing right now. Then in that case what will happen, the print command becomes necessary to display the variables on the screen. Okay. So that is where the print command is used.

Okay. But do not take tension about this print command but that is just for introduction. We will just use this as a mathematical tool as well as visualization tool. I am repeating it again. Okay. So just this is to refresh that how to use Python and if you are interested and you can dig more deep about it using a computer science way and computer engineers approach to understand what the language is and all those things.

Okay. Now similarly you can give multiple variables and we can print in different formats and all. Okay. Now this is what this says is you have, you can, how you can print two variables in a single line. So you have to define one in curly braces and two here curly braces and then you have to format what is one and two here.

Okay. Instead of one and two you can use a, b whatever variables like that. Okay. And it will print like this. Okay. In fact if you do not do anything, you do not do anything just in dot format it gives the name at the specified means order, in the specified order it will print the same.

Okay. Now coming to something called a list. Okay. This is something important. Just like in, we make some kind of to do list, grocery list something like that there is something called list.

Okay. We saw what a variable is. Now there is something if you want to store multiple things in a single variable, then we can use list. So list is defined as shown here within square braces. Say if you have a square bracket, then this is list. So here something is called list of list.

So you can have a list inside that, say this 1, 2 is again a list. So this is list of list. And similar to the value definition in defining a value of a variable, we can define a variable for list also. Okay. So for example, if you have my list here.

Okay. Now this is abc is our list and that is in my list. Now if you want to append another item to the list, what command we should use? We should use dot append. So mylist.append, what it will do? It will append d letter into my list and if we print this, we will get abcd.

Okay. Now and then there is something called list indexing. Okay. Something called list indexing. So indexing is what? Say if we have, for example, if we have the list abcd here, indexing means how we will print only a.

I want only a from the list. Okay. So just like a matrix, it is defined as i,j. Okay. Any matrix can be defined as i,j. So here this one, the first one is 0.

Then this is 1, 2, 3. Okay. It starts from 0. So if I say list of 0, it will print a, 1, b and if I want to print from 1 onwards to the end, what I will do? I will do 1 colon that is all. But remember

we have to do it in a square bracket.

Okay. So it will print abcd. Okay. Similarly if I want to print from starting to 1, it will, so I will do it this. But one thing to notice here, if we say 1 colon, it will print till the end. Okay. And if we say colon 1, that means before 1, print whatever it is before 1.

Okay. If I say generalized way, so in generalized way what I can say is list say 1, if I want to print anything from 0, it will, 0 to n, so it will have, it will print 10, n minus 1, 0 to n minus 1 location. Okay. So this is how it is. Okay. Now what you can have, now this few example shows how you can have list inside list.

Okay. So these are means, this particular notebook will be shared with you. Okay. Now you can go through in detail. I will skip few concepts and this, not skip, I will go a bit faster through that. You can just go through it and you can experiment and see what can you do and you can get it, what you can do.

Okay. So simply say list of 0, we can change, we can change means like the specifying the location also. So if I change it, 0 change to mu, you can see mu here. Okay. Similarly, you can have a nested list. Nested list means 1 inside another or list of lists, something like that.

Okay. So 1, 2, 3, then inside this there is another list, 4, 5, then this. So how you will see this. So if we put 3, then 0, 1, 2 and 3, the 4 to 5 is, this is the entire list is in the third position.

Okay. So it will show the entire list. So it is kind of you can say a two-dimension or the three-dimension thing. Say if I print 3, 2, so it is saying from the third element of the list, print the second element of the list, means list of list. So second what is 0, again 0, 1 and 2, that is target. Okay. From the, this is also a list because it is under square bracket.

So you can write list of list of list. Okay. So similarly you can have multiple nested lists. It will help for different reasons, means when you go too deep into data analysis and all. So it might help sometimes.

Okay. Now going to something called as dictionary. There is another type, data type which is called dictionary where you have, you can assign a key to an item. Okay. Just like you have a dictionary where you can search by letter, then you can go by alphabetical order and you can find out where the word is and what is its meaning. Okay.

So similarly, similar to that, you have a key here. That key can hold an item, that item can be a single number or a list or something called, there is another data type called tuple, that can also be there and also you can assign another dictionary to that. Okay. That is also there. So and with dictionary what you can see, d for example, if I run this, it will store this dictionary value into this.

Okay. Now if I want to see what is inside the key 1, what I will do, I will run this, it is saying

inside key 1 is item. Now say for example, you forgot what are the keys in the dictionary. So what you have to do? You have to do just type keys enter.

Okay. So keys, bracket open close. So it will show what are the keys inside the dictionary. So key 1, key 2, there are only two keys. Okay. You can assign any different name to the keys also.

Say I will say a, a and then same thing and then same thing. So now it became a. Right? Now, now other than this there is Boolean operation. Boolean operation is basically true or false operations. So true, true represented by the word true, t-r-e, false and false.

Okay. Now how you can check that what is true and what is false, else you can do logical operations on that. So for example, I will say 1 is greater than 2. This is a false, right? So false. So these are Boolean operation where if you do any logical operations that will also, that will result in Boolean answers. Sometimes it is necessary to use Boolean operations where you have only limited means, either something can be true or something can be false.

That then and then only you should use Boolean operations. Okay. Now there is something called tuple. So tuple is a data type which is not changeable, which is unchangeable. Okay. So what happens in a tuple, when you assign it, it will not, it cannot be changed.

So tuple is mainly used to assign constants. Like say if you have something called, say you have read box constant, so it will not change, right? Or Boltzmann constant. So you can say e minus 20. So I write it as e minus 20, that means 10 to the power minus 20, it is also called main text format. So if I have this, okay, then if I run this, so now if I by mistake in somewhere in my, when I am doing some kind of computation, I am trying to change this to any other number, it will throw an error.

Okay. So tuples are unchangeable data type. So it once assigned, it will remain same. So it is very useful to use tuple to define constant values.

Okay. So then comes something called sets. Okay. So sets, just like our mathematical set operations, we also have sets here, where we can use, where we can use the curly braces to denote sets. Okay. So this is how you can define set and you can define it as a variable also. Now as I said, we can do logical operations also here. So greater than 1, less than 1, so true, false, whatever it is, equal to operation, greater than equal to or less than equal to operation.

And okay, so you can also compare strings over here. Okay. You can also compare strings, like say hi equal to equal to bi. What is it? Is it equal? I do not think so it is equal. So false. Say even if I say hi equals to capital H i, it will say false. Why? Because it is small h, it is comparing the, it is case sensitive also.

So when only hi is equals to hi, then only it will show true. So let us say so c2. So also other logic comparison operations, then logical operations also and, or, not, those all things can be done. Like say you have kind of two conditions. So say for example temperature should be

above this and also the, also the some other parameter say some coefficient should be below this.

So you can use and operation. So say we can say 1 instead of 1, we can take it as 5 because it is already false. So you can see whether it is now should be coming true. So true. So this is how you can do logical operations also and you can do multiple logical operations also.

So here you can say this, this and and, we should have 5 less than say 7. So true. And if I change this 5 to 4, then it will become false. So because or is choosing between best of two, whereas and, then we have done and, then that means kind of logical multiplication, logical multiplication. Logical multiplication, if both are true, then and then only it will be true, otherwise it will not be true.

Now coming to something called if and else statement. If and else statement is nothing but a decision making thing. Say if this happens, then do this, else do this. So you can do simple comparison. Say if I am comparing two numbers, so instead of this what I will do, I will have say two numbers A equal to, let us assign 67, B equals to say 78 and let us have third variable also.

We will see how to use the nested if also. C equals to you say 54. Now what we have to do, we have to print the largest of 4, largest of 3. Let us first take the two thing, two number A and B. If A is greater than B, then print A like A is greater. Now then what you can say, else print B is greater.

Now since we are using if statement, I am giving print command. If I do not give print command, it would not display any output. So this is why print is important. So say B is greater. Now if I want to compare C also, so what I have to write, we have to write I if and then again we will have a comparison that A is greater than C also, then A is greater than B, then A is greater than B and inside this we have to see if A is also greater than C, then and then only print A is greater. Now otherwise what you have to do, I if, you need to check for the second thing which is say B is greater than C.

Since A is not greater than, if A is greater than B, then it will check for A is greater than C, then A will become greater. Otherwise you have to check for is B is greater than C. If B is greater than C, what will happen, print B is greater. So that means the first condition A is greater than B failed.

If we check for B is greater than C, then B will be greater. Otherwise if both condition fails, what you have to do is print C is greater. So this is how you use equal statements. So we got the answer B is greater and you can see here this is true. So if I reduce the number of B say 9 and if I take it as 10 and run again, then you will see C is greater.

So what is happening here, here I did a mistake. So if you see here, in if we have A is greater than B, if we see A 10 B 9. So this is how you do dry run also by mistake I did this. So here what you do A greater than B, you check for A is greater than C, it is false.

So it ended the if here. But here what you can do, here you can put another statement here. So if this is true, you can say if A is greater than B false, then check for B is greater than C. If that is also false, then you can see check for then C is greater. Let us see whether this works out or not.

So C is greater. Now let us again go to something called B is greater. So it is not printing anything. So what is happening here, A is greater than B, it is false. But there should be a statement suggesting comparison between B and C also. So here we should have similar else if comparison between B and C where to print B is greater than C is greater.

So let us run this. Now let me see more data. Let us see. Now we do not have any loopholes in this if else statement. So this is all the basics. Now there is something called loops.

Loops is like a sigma operation. And it is a iterative operation. You can go back in a loop, means like you started something and you can print, you can go in a line and a circle. While completing that circle, you can finish certain jobs. You can imagine like that. So if you have a sequence of numbers like 1, 2, 3, 4, 5, 6, let us take one more number. Now for statement, you can, for statement, there is, how you can use for statement is for some variable, means like you are assigning iterative variable.

This is something called iterative variable. That in sequence means the set operation you remember means like sets, in sets what you say if this item in this then print this. Now for is saying this item is in the sequence, this sequence means this, this and then print item. So what it will do, it will take, it will start from the first part, 1, 2, 3, 4, 5, 6, it will print like that. Now the same thing if you can print, instead of printing the item, you can print anything.

Now also there is another way to define loop is using something called range. Instead of just showing kind of say, showing what are the inside list, what you can do, you can print, use the kind of numbers or the range of the variable to show what is in the, to show something or you can reiterate that loop or that amount of time. So I will say I, I will say in range of 0 to say 6. So print, instead of item, I can print I.

If you run it, you can see it will go up to, oh sorry, I misspelled range. So it will go to 5. So 0 to 6 means 6 times it will iterate from 0. So n minus 1. The last variable will be n minus 1. Now the one thing we can try out is say we have already a sequence.

What I will do, I will iterate it to what? I colon. So we will see a, it is kind of a beautiful pattern here. So this way you can play with it and also you can just say colon I and plus 1 I will give. So you will have this kind of pattern because it will go to colon to n that means n, 0 to n minus 1 it will print. So this is how you use for loop and there is something called while loop. While loop is iterative loop which loop means which kind of loops around or means runs whatever inside the loop until the condition is true.

So say you have, you want to do a Newton-Raphson approximation or some kind of approximation of a function. So you want to find a root of a function. So you want to be correct

up to two decimal places. So what you will do, you can put a condition here where say  $x$  is equals to say you know the near about answer up to two decimal say 4.35. If it is there, then and only it will exit the loop or else it will be or it is more precise then it will exit the loop.

So that means when this thing is means when the while loop is false then and then only it will exit. So if I have while equals to 1 and where I is less than 5, so it will go up to 4, 1, 2, 3, 4, 4 iteration. So similarly what I said about range, when we define range it will give a value means it should iterate from 0 to 5. Then also you can see list range, range defines the iterative numbers. So that way you can define range.

Then what you can, there is something next thing on the agenda is list comprehension. How you can have, you can understand list a bit more. So if you have a list say here  $x$ , now I want to append into a list means I have few values or few values which is being generated from some other thing or some other loop I want to or I have some list of items. I want to append it into a new list. I want to create a new list. So what one can do, one can have an empty, define an empty list and using append command you can append the  $x$  list to this and you can have the another list.

Say here what is happening, whatever the items in  $x$  are getting iterated one by one, it is doing a power operation. Say if I have 5 here and run, so it is order of 5. So you can make a geometric position, Fibonacci series, whatever you want using loop. So and this is what is called shorthand writing for list appending using for loop. So how it is done? The shorthand writing is like say the end product will be a list wherein so the, you should have, you should write the operation or assignment, okay.

Then the for, then the iterative variable or you can have the element. Then you can look in or you can use in range also of whatever value you want to iterate, okay. So this is how you can have a shorthand notation for the assignment of variables using for loop. So now coming to something called function. So say functions are very useful tool when you are doing kind of redundant operations like say you want to calculate the power, noise power, for example noise power given the value of noise temperature of the system and also the bandwidth.

So I will say here noise power PWR. So what I will give? I will give the temperature  $T$ , frequency,  $FREQ$  and then what I will do? I will print the power in dBm equals to, I will have this, then dot format and inside this what I will do?  $T$  and this is not frequency, it will be bandwidth  $BW$ , okay. So if I have  $T$  times  $BW$  times  $1.38 \text{ E minus } 23$ , then not dBm, sorry, we will give in watts. So for dBm we will see what we can do afterwards, okay.

So we will have the flux constant and everything, so bandwidth, okay. So this will print this. So if you define this, now what I have to do? I have to just call this function by noise power. I will assign 400 Kelvin of temperature with 6 MHz of bandwidth and press enter. So this is our power in watts, okay. So this is how you can define a function and then call again and again and see what is And then you can also have something called return type function.

Return type function means you are returning the value to another variable. For example, I am returning return, there is a keyword called return. What you can return is the calculated power in watts. So instead of just printing, you can assign power, a variable to it and if you see here, you will have power in watts, then you can say PWR.

So none. This is coming none because I did not compile this. So do not forget, if you change the function, then again also you have to compile this. Now it will show the value, okay. So if we do this also, this will show the value. So I am doing this on the fly and you will find some mistakes in this and that is how you learn.

Means if you do mistakes, then see what is happening actually. There might be some error in your logic or in your assignment, okay. Now this is how you do and there is also something called lambda expressions where you directly just return the value multiplied like this, okay, where you assign for some trivial tasks, okay. Now lambda function like var is equals to var into 2. So it will say, it will define a function which will map this thing, okay.

Now here, if you see map and filter, actually in this what you can do, use, you can map say times 2. Times 2 means you can map this function, say lambda in a lambda expression. What you have to do, you have to say define this function. This is something called multiply by 2 function and you define a lambda function, okay, where you can define lambda function is in single line statement, you can define anything, okay. Now you have a sequence, you can map this sequence to the function and the following list will have everything multiplied by 2, okay, when you map it. Means you say that take this list and do element wise multiplication by developing, means you give, instead of having a for loop and giving means running a for loop like we did earlier and providing each element to the function what you can do is something called map.

Map is means assign every element and do the operation of the function, okay. And similarly you can map a lambda function also with that and you can do different types of other operations. But these things actually is not needed. This is just an extra item which might be helpful for you but okay, so the things which we need is till here, here means what are the different kind of data types. So this is something called string operations which you can go through. It is very simple where you can means sends everything to lower case, upper case and you can split the names, okay, then assign splitted and two items and all means.

You can have dictionary and all those things whatever you can do with that and this is something called pop and push in the list. Pop means it will remove the top item from the list and sorry bottom item from the list, bottom top item, okay, top item means 0, 1, 2, okay, 2 is the top most thing. So it will pop out the top most and then you can insert there, okay, using push command also. So now this is something which is important to us which is called numpy. Numpy is a library, Python library which allows you to have array of variable.

Array is nothing but a list of variables which can store a single data type, okay. If it has numbers, it can store only numbers, no other list, okay, no other data type. So you can have, you can define a list by this and you can convert it into an array using np.array my list, okay or you

can have list of list.

So arrays are kind of symmetric. You cannot have, not symmetric means it has a particular order. Say it is 3 cross 3. If I remove this, let us say if I run this, you have an array and something called shape command.

So it will shape, okay. So it is not taking any shape command. So there is another thing np.shape, okay. So it is saying np not defined, okay. So because I have not run this, so library is not imported. So in MATLAB, sorry, Google Colab, all libraries are almost there, so 3 cross 3.

So 3 cross 3 matrix it is. Say if I do this now, it will throw an error, okay. So it threw an error. What it did, it only took to the first value and stored it. So if I make it an array, try to make it an array, it became something array of list, okay. So each element of the array is list.

You cannot treat it as an array. So the beautiful thing about array is you can use this as a matrix, okay. And matrix solves most of the problems in our linear algebra. So that is very powerful tool. So here what you can, there are different operations. You can define an empty array as 0, 0's array, means it will be filled with 0's.

So if you have 3, so 3 linear, 5 cross 5, then you can also have 3 dimensional arrays also. And there is something, the basic command something called lean space where you can define linear, linearly, linearly spaced variables, okay. Say lean space 0, 10 start and end and you should have number of, how much number of elements should be there.

So start and number of elements, so 1, 2, 3, okay. 0, 1, 2, that means total of 3. Now 50, so it will be, it will define 50 number of variables. Then there is something called I which gives identity matrix which is useful again for matrix operations. Now if you want to generate random variables what you can use, there is something, some library called random, np.

random and ran. And 2 means it will give you 2 random variables and 5 cross 5, that will give. So it will give random variables between 0 to 1, okay. Now if you want to have between 0 to 100, you can go ahead and multiply it by 100, no problem, okay. You will get between 0 to 100. Now there is something called randn also where it has both positive and negative values. So minus 1 to 1 is the range of that.

So, okay and something called randint which will only give integer random variables and then there is something called reshape, okay. Now reshape is say if you have a long say 25 column matrix, say 25 column matrix means only 1 row and 25 columns.

So I have, I will say arr equals to np.random.randn say 25. So arr will become, see the arr will be single, okay. Now dot reshape, I will have 5, 5, okay. See. It will reshape it into this.

Now let us again, I will just do this and separate this code out. This code out I will separate it and paste it here and here I will try to see arr, okay. Now here see how it broke down the array.

So it took first 5 elements as the first row, next 5 as the second row, then after that next 5 as the third row like that. So you need to always be careful how you distribute it, okay.

So that is a way you have to do. Now what we can see, say it is not 25, say I have 36, 36 which can be, so 6, 6, 6, 6, okay. Now remember to run this one first, okay.

So here if I do 6, 6, it will create it 6, 6. Now but 36 can be represented in another way. So I can say that as 4, 9 is also 36, correct. See, 4, 9. Now let us say we do it 6, 6 and again now let us see, let us print arr but it is showing one dimensional array. You reshaped it but you did not save it into the array. So what we have to do arr equals to this. So now it will reshape and store in arr and then you print arr, you will get this and if you again try to reshape this into 4, 9, it will reshape it again by same thing.

Means the first 9 elements will become the first like that and it will, first of all it will stack and linearize it, then it will reshape, okay. You need to always be careful how you are reshaping it, okay. Now there is something called min and max argument which is used to find maximum and minimum numbers in an array, okay. And there is something called shape which was actually failing earlier.

I do not know why but you can, yeah, that is because I did not use this. So now let us see if I, see this process. So shape because I did not run the np library, so it was not showing shape.

So np library. Now there is something called dtype also. This you can see what is the data type inside the array. So this will be float, okay, dtype. Here array has been reassigned and on but what we are using float 64, okay. So generally the data type is float 64 if you want to specify into another data type, it can be done.

So now there is something called indexing. Indexing is in a similar manner as we see in list. So that is there, okay. Then using brackets, notations, you can do there. Then also broadcasting. There is something called broadcasting, say if you want to have from 0 to n values same as the something, so you can assign it like that, 0 start value and 5 is the end location, start location, end location in the array, okay.

And that will be changed to 5, changed to 100 as shown here. So if you do this, see, it changed to 100. So why this? Because this is taking number of rows, okay. I have not specified columns. So every row, row 0 to 5, 5 means not 5, n minus 1.

After colon, it will go till n minus 1, whatever n written here, n minus 1. That is to be taken care of always. You should keep in mind. Then this is also called slice of an array where you have 0 to n minus 1 number displayed, okay. Now you can also assign everything the same. Also there is negative indexing, something called negative indexing, okay.

Say I have, I will rearrange the array. So say if I want to show the last value, not the different one, so I will slice, I will tell show minus 1, okay. So here what you can see, 10. So how the

indexing is assigned in an array or also in a list is the first element is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. So 10 elements. So 11 elements, so 0 to 10, okay.

But it has a inverse indexing also where it can be written as minus 1, minus 2, minus 3, minus 4, minus 5, minus 6, minus 7, minus 8, minus 9, minus 10 and minus 11. So minus indexing.

That is also kind of spicy. Say if you want to go, say I will write minus 11, minus 1. Why this? N minus 1, okay. So you have minus 1, minus 1, that is minus 2. So if I say here 0, let us see what happens. Nothing, because 0 is 0, okay. If I put it a colon, it will show all the things, okay, because it is considering minus 10 to the end, okay. So end is towards the 0, okay, negative indexing.

So coming to the two dimension array, you have 2D array and in that just like list, you can use to display the element like this, say, okay. If I press 1, it will show row. So format is row and column. So here I can see second rows. I want to see third element, which is 2.

So we will get here some value. So also instead of this, you can use comma notation, comma notation where this one is row and this one is column, okay. And also you can do slicing in array also in row and column location, okay. So coming, this is what is called matrix, means you have n number of rows, n number of columns, matrix and defining matrix in for loop, okay. Now this is the kind of summary of something called indexing or how the data structure will be look like, okay. So if you have a matrix or defined array like this, so data type is what is called the whatever the value storing data type, it can be eight byte integer or something else.

The shape will be the shape of this, how much by how much. So for row by column, four rows, three columns, then stripes are 24, eight, that is means number of, how much bytes it is consuming or something like that. So one segment is eight bytes per element if it is an integer. That is a different thing when you go into, approach it into a public science manner, then it is okay. So now there is something called vectorization which is our part of interest.

So what you can do, you can add array to get the result, this you can do kind of simple multiplication to get the variable. Now there is something called broadcasting, you have the single array and you multiply it by this and you will get 2D array, okay. So you can do reduction also, okay. Now there is various things you can do.

Also there are several functions inside this numpy library, something called, we can use mean operations, okay. Now let us move on to numpy operations. So you can arrange, you can divide it by, just dividing means you are doing element wise division, okay. Element wise division means one, the first location will be divided by the first location of the other matrix, like that, okay. Now the double star is power as we discussed.

Now coming to universal array functions, okay, you can find out square root of each element inside an array, okay. And then you can find the exponential to the each element is what, okay. We will say you have an array of, let us define an array here. This is an, let us do this. So array,

we will define an array and if we do to the power, we will get this.

Say if you do the exponential, what we will have is  $e$  to the power 0 which is 1,  $e$  to the power 1 which is 2.71, like this, okay. Now if you want to find the maximum value, we have to do `np.max`, `sin`, if you want to see the `sin` values of each element, we have to do the `sin`, `log`, this you have to do the `log`. So there is something we defined, we defined a function, power calculation if you want to do something, let us define it again.

So if you want to calculate power in dBm, so PWR dBm, say we took 400 as our, 400 kilometers of temperature. If I correct the difficulty,  $6e$  plus 6, 6 megahertz of bandwidth and  $1.38 e$  minus 23, our constant, okay. Now I want to find out in dBm, that means what I have to do, I have to do 10 into, I have to use `log 10`, `np`.

`log of 10`, okay. Now this done and then minus, sorry, plus, this will be in dBw watts, plus 30, we will have power in dBm. Now to display PWR and dBm in node book, not in other thing, so 1.4, okay. So then coming to another type of data type or this is a different library called Pandas, which is used to define something called data frames. Data frames is advanced version of matrix or it is not matrix, but it is a kind of table, okay, where you can have different data types also for a matrix, okay. Say this requires a bit more of understanding and if you are more towards handling different, want to handle different kind of data types, this you should learn, okay.

But just for general representation, what we can do, we can just go through it, okay. So here what we will have labels, my list and we have some kind of array, we will create a dictionary and using series, we can create Pandas series, okay. And then also we can assign different labels to it, different, instead of array 1, 2, 3 index, we can define A, B, C index and also this can be taken as this, means like different kind of notation to write it and if you say only we provide the arrays to `pd` series, it will just create a normal Pandas series. So there are many other operations what you can do, what you can do, it is very useful when you want to assign some kind of category and create a sorted data set with a key and all and means when you want to means have different kind of, say you want to have a grade sheet of multiple students. So what you can have, you can have a key as a roll number which we call as index, okay.

Index will be unique for everyone, the roll number will be unique for everyone, then you can have name, their mark, their total marks, average marks, everything you can do. So this is very helpful for handling different kind of data. Now, there is something called data frame, earlier this was all example of Pandas series, Pandas series means data series. So series will have only one column whereas data frame can have multiple columns, so like this, so you can assign index and columns like differently, so and you can see each selection, means if you want to see, means data frame as a means one column only, what are the values in one column, you can just print that in multiple columns, you can print that at, okay. So there is another way to denote which is dot the column value, that way you can also denote, then you can also see the data type, what data type it is, so if you want to see the, what is inside it and its data type, so you can index it like normal array also, this is w of 0, so `df` is not defined, it is saying because I have not ran this thing, so the first thing to solve the error is read the error, okay. So what you can do, you can read here `rand n`

is not defined, so I have to run this one, so rand n can be defined and then I have to do this, read the error carefully, then you will be able to do, you will be able to resolve it quickly, so now if I run this, so float 64, so the data type of the 0th location is float 64, now you can do addition of two columns and all those things, you can create a new column by, by like this and so on, so simple means if you go through this notebook, you can find all the basic operations, how to do that, how to select rows, means how to select rows, so for selecting rows, you can do something, use an inbuilt function called loc and you can specify row a, b, c, d, if you have changed the row name, otherwise it will be 0, 1, 2, 3, okay.

Now collection of series, so you can always do different kind of operation, logical operations, say if you want to find whatever the value greater than 0 is, so you can just display that and find out what are the different values which is greater than 0 in w, so it is saying in w, whatever greater than 0, show that all the values, also all the values where w is greater than 0, okay, so it is showing like that, these are all means kind of selection of gimmicks, if you are more into selections of new particular type of thing, this is just kind of extra bit, okay. So this is all, then there are multiple types of things you can learn by simply going through it and also there is one thing which I would like to go through before concluding this, is yeah, this one. So here what is shown is how to, how you can load data into a data frame, so you can read from file also, okay.

So I would not run this because it needs to connect to drive, okay. Now you will be given some example files and all where you can run this and see. So you can read it using, if you have a CSV file, you can read that, at also you can save as a CSV file, then you can also read from excel, okay, using read excel command, also you can read from, you have to install through libraries if you end up using Anaconda which is also Anaconda software which is used for Jupyter manual too. So you can read from HTML file also where you can say read anything from, read a table from the particular website link, okay. So it read, it read, gone through many things and you can also say that only read a table which matches this thing, something, some literal, okay, there.

So that also you can do, okay. So with this, the first part I would like to conclude for the Python course, okay. So we will start next with data visualization, okay. So let us come to the next part where we will talk about data visualization. So what we will do here that we will try to visualize the array or the values in a graphical manner, okay, or in a pictorial representation. This is important in order to what we will be doing in the next session and that how to see what are the, how to graphically observe Fourier transform, what it means by different kind of aperture distributions.

You already saw in the lecture that how it looks graphically. However, we will see first hand how it looks, okay, using programming and you can do it by yourself. Also if you remember correctly, we on one of the tutorial we said that we will do electromagnetic simulations or kind of visualization using Python also. That also we will try to do, okay. So basically anyone if you, I can, if I say in the simplistic words or boil it down to a basic level what simulation is. This simulation is nothing but representation of your system or dynamic system using codes, okay.

So using certain algorithms or certain set of formulas you say how the system is mutating over

time or over space or whatever, okay. This is what simulation is. In fact if you try to plot  $y$  is equals to  $mx$  plus  $c$ , in a sense it is, you can say that it is a simulation you are doing for a straight line, okay, where you mutate the  $x$  coordinate to observe what is happening in the  $y$  coordinate, okay. So this is very, I think in a, it is like a rudimentary way or a rudimentary way we can say we are simulating something, okay. Simulating means providing a stimulus and observing what is happening, okay. So there is particularly different libraries using which you can visualize.

There is something called Matplotlib, C bond, built in Pantas function is there, Plotlib, Guffling and Jor unful plotting is also there. So we will discuss about Matplotlib and this file is shared and you can go through C bond and do it. So you have to first import Pantas library as `plt`, `plt` is a kind of a like, like something called object for the library by which you can access the library functions, okay, which I did not tell it earlier that `import num p as then np`, all those things are written. But this is something called object which is a kind of a go between, between the actual functions inside those library, okay. So you can give it any name, instead of `plt` you can write `p`, `a`, `b`, `c`, `d` but `plt` is kind of unsaid rule to write `matplotlibs dot pyplot as`, okay.

Now this statement says that, this statement is for our Jupyter node loops where it will show the inline result directly without saying `plt dot show`. This means if you want to use matplotlib you have to define what the plot function will be and then you have to use `plt dot show` to show that plot function, what is generated. So if you use this then it will directly show the plot instead of putting `plt dot show`, okay. So we will also define `import num p`, okay, and this is something called, we will provide a `linspace` variable.

If you see  $x$ ,  $x$  is like this and  $y$  is defined as  $x$  to the power 2, double star means to the power, okay. So if you plot this, we will plot in that, okay. So `plt dot plot`, `plt` has a function called `plot`, okay, where you have to define  $x$ ,  $y$  and  $r$  is the, stands for color red. So you can change it to say, `b`, it will come blue, okay. Now if it is say, `k` is black, `k` is for black, okay, then again `r`, `r` is like this. Now why this is happening? Because in somewhere during the end of the program, we have written here `plt dot show`, if you move this then also it should plot, okay, see, it shows, okay.

Now if you see this, see, now you can use subplot, okay. There is something called subplot. If you have a figure window, it will assign row and column to the each section of the plot, okay. So, you have to define the subplot like `1, 2, 1` means the subplot will have 1 row, 2 column and you should, you have to place the first plot in first location. So it will define it like `1`, then `2, 3, 4, 5, 6`, it have multiple rows. So first one, you have to do this, second one then  $y, x$ , so this thing.

And `gr minus minus` means show in dotted line, `star minus` means this. So you can use different means combinations, say if I use `o minus`, so like this, it will come, okay, `o minus` and all those things. So this shows that how to set  $x$  label,  $y$  label, this is command set  $x$  axis,  $y$  label. So this is different kind of representation, different kinds of representation to plot, okay. But the same means this can be used here also. So directly you can plot like this.

So this till here it is important to understand till here this portion because we will be using till this portion only to plot also this also, this other kind of, another kind of representation to plot,

okay. So now moving on, we can have plot in plot by means defining the axis and the, means like the box, say main box is 0.1, 0.1 this and the inner box you can define something else and then if you see, it will have box in box. So you will see this is the large, enlargement is happening just because somewhere in the end of the codes I have increased the means value of the, increased the font size to a certain extent because of which this is happening, okay.

So just to, I will, what I will do, I will restart it and start it again, okay. So this won't happen, okay. So somewhere in the down the code you will see that I have changed the font size due to which this kind of shitty thing is happening, okay. So you can create an empty figures also like this and on that you can plot, okay, like this. So using calling that you can plot, okay. Now if I go further you can go through it, means it's all just specific means kind of notation, specific kind of notations here. So you can just follow through and it's just easy means the thing, it's easy to understand like say here you can create means, you said you already created two empty plots, okay, empty plots by this, okay, and using for iteration we try to access each of the sub plots, okay, and plot in that.

This is how you do in iterative manner, okay. So now there is something called you can set the figure size, expect ratios of these things say like this. So figure size 12, 3 that means you will have means, let me just run this, okay.

So this will only create a means figure size. So now here if we have a subplot, it will have like this. So figure size it says 12, 3. So that means you have 12 pixel, figure size of 1200 plus 400. So 400 is in the width and 1200 is like this, okay, length and width, okay. So here you can save the fig file using say fig command.

So in fig and this you have already created and then fig.save you can use to save. However, if you are not using this kind of notation and you are using direct using plt, then you can say plt.save fig, okay. So that way also you can save the figure file. Now setting the title is like access.

set title, this and that. Now here also you can add separate titles coming to the unlabeled part, right, this one. So if you want to plot multiple things in a single plot, you do not have to again assign xy, xy kind of variable. If you want to plot in a multiple way, the general command is plt.plot x1, y1, x2, y2. This is the general command. Instead of that what you can have, you can have the separate plots and one after the another it will plot one over the other, okay.

So instead of plt you can just define the access and or figure separately and on that you can plot, okay. Now there is different locations which you can give to the legend. Here legend is this label you can, there is a keyword called label using which you can define anything. Here like here I can say square, now here I can say cube. So if I run this, so here square and cube, okay, square and cube.

Now this is a kind of a keyword which defines the label and it will, if you invoke legend keyword, it will show. Now you can change the location of the, location where the legend occurs.

So this is the standard places where you can say. So  $\log_0$  if I say  $\log_4$ , it came here, lower right corner, okay. Now let's see the different types of formatting to the lines, means where you can change the color and show it better to the viewers to, so that they will be able to differentiate. Say for example you have, you want to write a research article where you have to do multiple plots, but some might take a black and white printout or have a black and white version of it where they can't, they won't be able to differentiate with colors.

So it's important to show them with different means line formats. So dash line, dot dash line, this different kind of, say you can use dot dash, again dot you can use and then you can plot it. See dot dash dot, that can be used. Now also you can give a hex value for the color code if necessary. If you don't have enough number of colors available, full value, full name of the color you can give and you can assign markers and line width, different line width and line style also, line style, LS means line style, okay, line width and line styles like this shown here and which will result to these things, okay. Now if you have multiple sub plots and you want to have custom access, custom clicks access and also the titles, here you can use setting title as set title and invoking the, which number of access you are trying to set the title for and also you can change the limit, limit of the x-axis and y-axis using `xlim` and `ylim` command. So if you are using this kind of notation where you have `fig` and access and assign a subplot to it, okay.

So here access represents the, this accesses where you are plotting and whereas `fig` represent the entirety of the figure which you can see with later, okay. Now in this case you will need to point out the access on which you want to plot but if you are using direct command like `plt`, you can, for `plt`, if you are using `plt` there is command called `plt`.

`xlim`. You don't need to have set `xlim` or `ylim`, okay. You can directly use `plt.xlim`, okay. `xlim` or `xlabel` if you want to label, okay. So there is another type of plot which is called scatter plot. Scatter plot they will only plot in points where it is, it won't join the line, okay. Then histogram plot is there, okay. And there is also walks plot where you can see bunch of data and the error bar also and the mean also, okay, which is very useful for statistical studies, okay.

And there is other thing called range scaling and different beautification add on to the plot here on on, here on. So this is how you plot it. So this is the basics you required for what we will do in the next session and of course you can, this is very detailed, detailed notebook which you can go through by yourself and see what are the different things and of course you can get back to us on discussion forum as well as you can write back to me by if you have any query, okay. So with this thank you and this is my email id you can always write to me and your queries