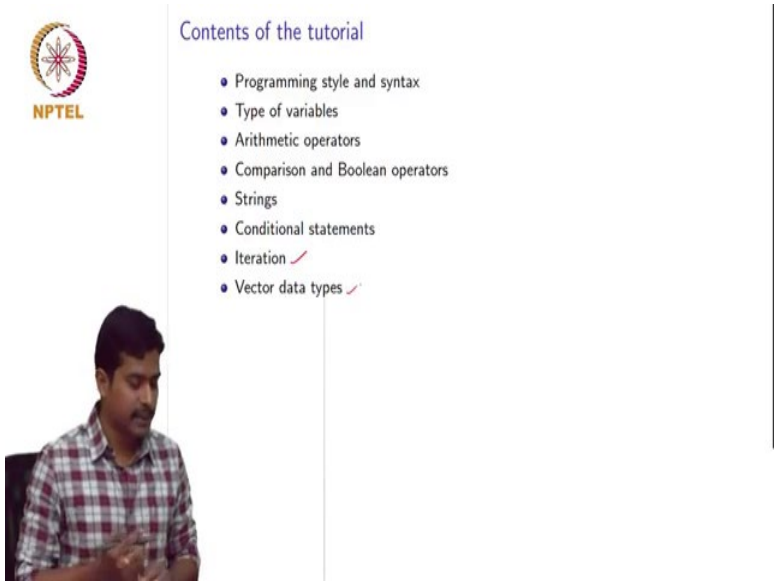
Optical Engineering
Prof. Shanti Bhattacharya
Raghu Dharmavarapu
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 33
Introduction to Python

Good morning, my name is Raghu. I am a student of Shanti Bhattacharya. Today I will give you a tutorial on python programming language. So, we are teaching python here, so that you can model some of the optical elements and the optical phenomenon that you have studied in this course; for example, a lens or a Gaussian beam. So, this is the tutorial.

(Refer Slide Time: 00:41)



So, all of you have already installed python, you have not? Maybe if you have not maybe you can sit with somebody who has python installed, so that. So, the idea is so, you do when I do it here do it along with me. So, I request all of you to install python before watching this video and try the python commands along with me.

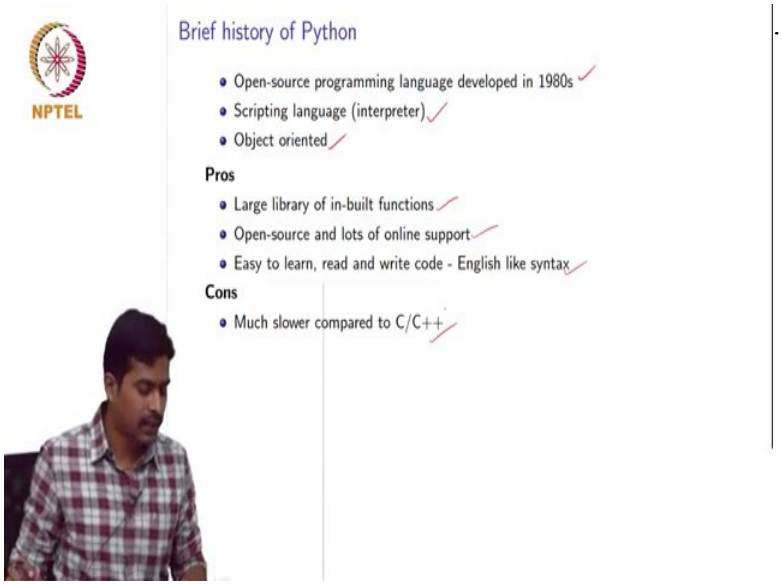
So python is a very easy programming language and it is very popular nowadays. So, if you see machine learning applications or scientific computations python is very widely used and python is very easy to install sorry, python is very easy to learn also. So, if you already know some other programming language like C or C plus plus, you can easily learn python. Because,

if you take any programming language, the basic building blocks of the programming language are the same ok.

So, these are the important topics that I will be covering in today's tutorial. So, basic programming style of python and whatever different types of variables like integers, floats and Booleans and what are the arithmetic operations on these variables like adding two numbers divide division multiplication. And we need some comparison operations to see if a number is greater than or less than a given number and strings are to give some sentences or something like that.

And conditional statements to see if a given condition is true or false. And finally, the most important thing is iteration because if you have some computations in your program and if you want to do it many times like 100 times instead of writing the same code 100 times, you can use an iterative approach like for loop or while loop to do it n number of times and finally, we will see a vector data types which is like array to store similar types of data types ok.

(Refer Slide Time: 02:56)



So, now let us go into some brief history of python. So, python was originally invented in the 1980s and it was first used for word or text processing kind of applications. But, later on because python is very easy to learn and it has a very powerful in built library people have adopted it by python in different applications. So, now, python is used in many applications ranging from academics and machine learning and a lot of applications.

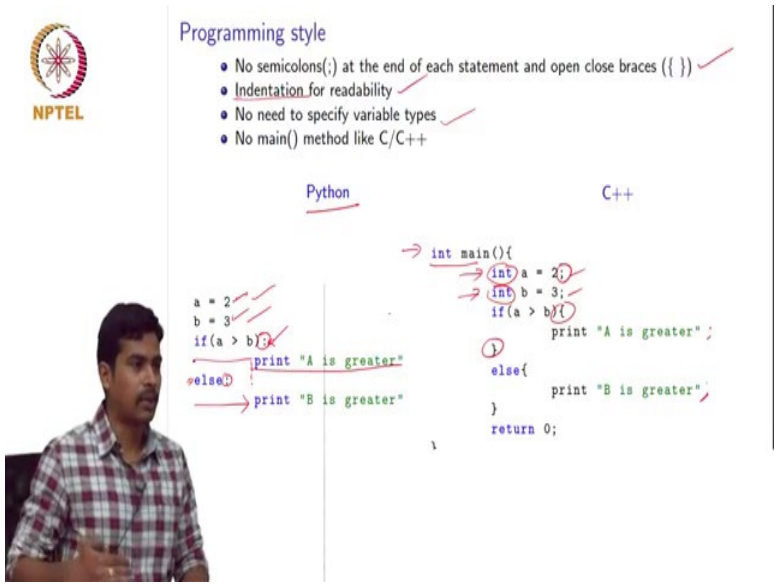
And python is a scripting language and it is an interpreter based language not like C or C plus plus, so, which is a compiler language. So, what is the difference here is so, in C language, if you write a program and if you compile it, the whole program gets compiled and then after that you execute the program. But in the case of python; so, if you write some 20 line program. So, the interpreter will take one line and then go to the next line. For example, if you have some error in some tenth line or some. So, the interpreter goes to the tenth line if there is an error it stops there and it does not execute the next lines.

But in compiler it will compute the whole program even if there is an error somewhere in between of the program and it is an object oriented programming language. And we are not covering this particular aspect of python in this lecture because it is beyond the scope of this course ok. And these are the advantages of python because it has a very big library, open source library, most of the functions are already written by some other people you can just directly import those libraries here.

And it is open source if you have some doubt you can just post your question on stack overflow or some other website you can get a solution within a few minutes. And it is very easy if you can it is The syntax of python is very similar to English language. So, when you are writing a program you actually feel like you are typing English sentences.

The biggest disadvantage of python is it is very slow compared to C language. Because C is like a very lower level language python is a very higher level language ok. So, this is a brief overview of python.

(Refer Slide Time: 05:05)



So, now let us go into how a python program is written; so, what is the style and syntax that we use to write a python program. So, when you see a python program, the first difference that you encounter is the semicolons that you use in normal programming like C or C plus plus. So, where you put a semicolon after the end of each of these statements, sorry, there was a mistake. So, they should have been a semicolon here as well. But if you see a python program, there are no semicolons after each of these lines ok. So, you do not have to end your statement with a semicolon. So, that is the first difference to any other programming language.

And the second difference is; so, if you see C program and if you are writing an if statement or a for loop, so, whatever code that gets executed in a if statement should be enclosed in open and closed braces. So, that your program knows this is the content I have to execute in this block.

But in python we do not use these curly braces instead of that what we do is we use this colon operation, colon symbol. So, after the colon symbol if you enter a if you press enter button, so, the program does not start from this point it shifts to some 5 spaces to this location. So, the python interpreter will know, so, which part is included in the shift statement by looking at this extra space. So, whatever lines you write after if, has this amount of spacing, they will all be considered as part of the if statement ok. That is another significant difference between other programming languages and python ok.

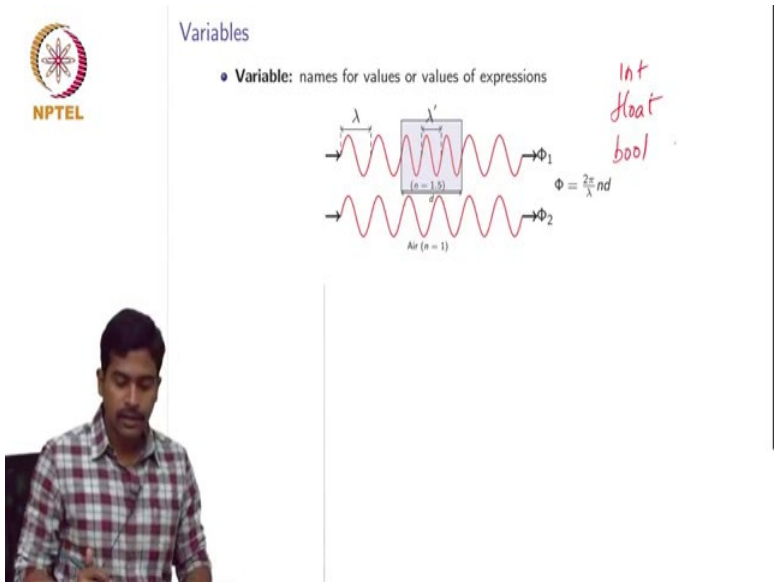
And yeah this is what I meant before the space. This is called indentation ok. This extra space is called indentation. So, if you see here after the if statement we have used the colon and then after that we gave this indentation. So, this particular statement comes under the if block. And next we have moved back again to this point and we have written the else statement and again a colon operation and we have given the space, so that the interpreter will know. So, this line is part of the else block ok.

So, this is the second major difference between C and python. And if you see another third one. So, in C language, we have defined two variables a and b and we have declared them as integers by using the key word int ok. But in python if you see these two statements we have not specified whether it is an integer or float or you know different the type of the variable. So, what happens is when you write an equals to 2, the python interpreter will itself take care of this assigning the type to the variable ok.

So, a is by default declared as integer because we are given it to so, this simplifies the python programming very much. So, you do not have to worry about variable types or anything you just write like English sentences. So, this is the third difference. And in C program so, your execution of the program starts in the main method right, you all know it. So, in order for your program to be executed you should have a main function, so this is a main function.

But in python you do not have to write any main function, you can just keep writing several statements and they all get executed one after another ok. So, these are the differences between C and python and this is the typical style of writing a python program ok.

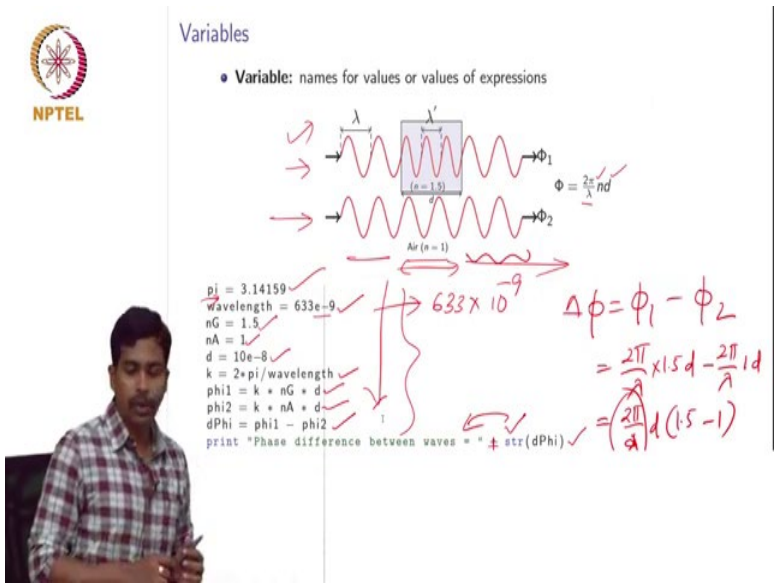
(Refer Slide Time: 08:36)



So, now let us move onto different types of variables in python. So, in any programming language you should have a variable type to store a particular data value. For example, you can store an integer, you can store a float value, float values I mean; which have decimal parts. Integers do not have decimal parts, but floats have decimals parts.

So, you need variables to store these values and in python we have 3 types of variables which are integers; ints, floats and boolean variables. So, boolean means true or false, it has only 2 values. So, to demonstrate how do we use variables in python?

(Refer Slide Time: 09:19)



So, I want to just introduce to some simple program. So, do not worry about what I have done in this program, I will explain. So, what we are trying to model in this example is what happens when light travels in a medium ok. So, we all know these are two light waves travelling in this direction and the upper light is travelling in a medium with the refractive index of 1.5.

So, we can say this is a glass material on the other hand the lower material is travelling in an air medium. So, and the refractive index is 1 ok,so, because of this refractive index difference. So, the output light is going to have a phase difference ok. So, that phase extra phase accumulated by this light because of travelling in this medium can be given by this equation ok.

So, this is $\frac{2\pi}{\lambda}$ which is also called the wave number and n is the refractive index and the d is the thickness of this material. And if you see in this region and in this region, there is no difference in phase between the upper and lower rise light waves, because they are travelling in air in those regions.

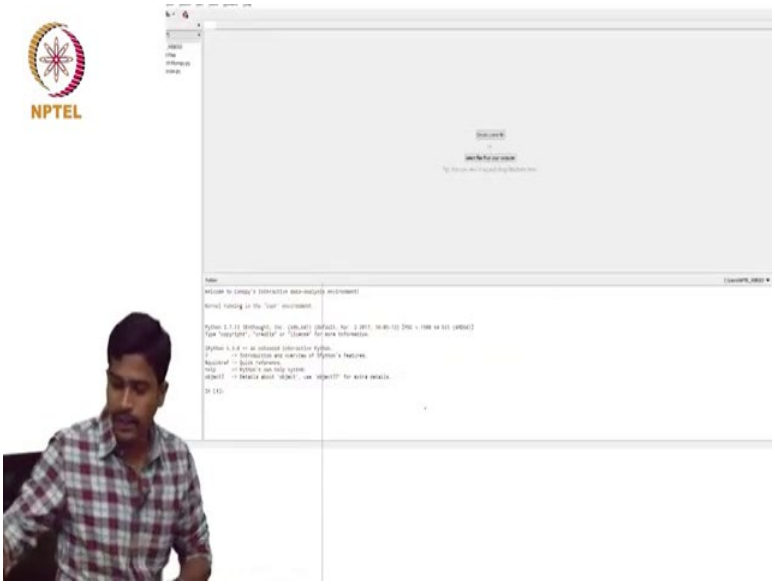
But if you see in this region; the upper one is going in glass and the lower one is going in air. So, there is a phase difference. So, how do we calculate this phase difference? So, $\Delta\phi$is given by $\phi_1 - \phi_2$. So, where$\phi_1 = 2\pi/\lambda * 1.5 * d - \phi_2 * 2\pi/\lambda * d = 1$ here and distance is the same ok. So, the overall phase difference is given by $2\pi/\lambda$ sorry, $\lambda\pi 1.5 - 1$ This is the phase difference between these two waves.

So, this is the same thing I tried to write a python program. So, here pi value is assigned to a variable called pi and we have not said it is an integer or float here, because the python interpreter will itself take care whether it is an integer or afloat. So, by default pi is going to be a float type of variable ok. And the wavelength of light we are choosing as 633 nanometers here, nanometers instead of writing $633 * 10^{-9}$. In python, we say $\pi\pi - 9$e , which means $633 * 10^{-9}$ ok.

And I just chose some randomly the wavelength 633 nanometers we can choose any of any other wavelength and μ_g is the refractive index of glass and nA is the refractive index of air and d is the distance travelled. I just took it some $10 * 10^{-8}$10 meters and k the wavenumber I have calculated using $2\pi/\lambda$ formula, this one and ϕ_1 is the phase difference and this is the equation and ϕ_2 for this equation.

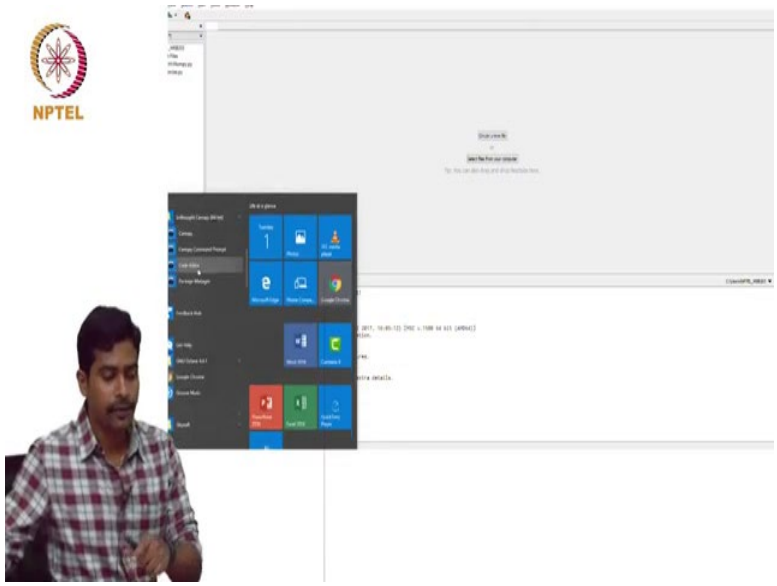
So, delta ϕ is given by $\phi_1 - \square_2$ Finally, we are printing the phase difference between the waves given by so and so number. And another thing you have to notice here is this statement. So, $\square\square$is a float type of number ok. So, in order to print it on the screen, we need to convert it into a string. So, that is what we are doing here. So, this is a key word str. So, if you write str of some number it will convert into your string. It will add this to the previous thing using this plus operator and it will print on the screen ok. String is a series of characters. It is like a sentence English sentence ok.

(Refer Slide Time: 12:48)



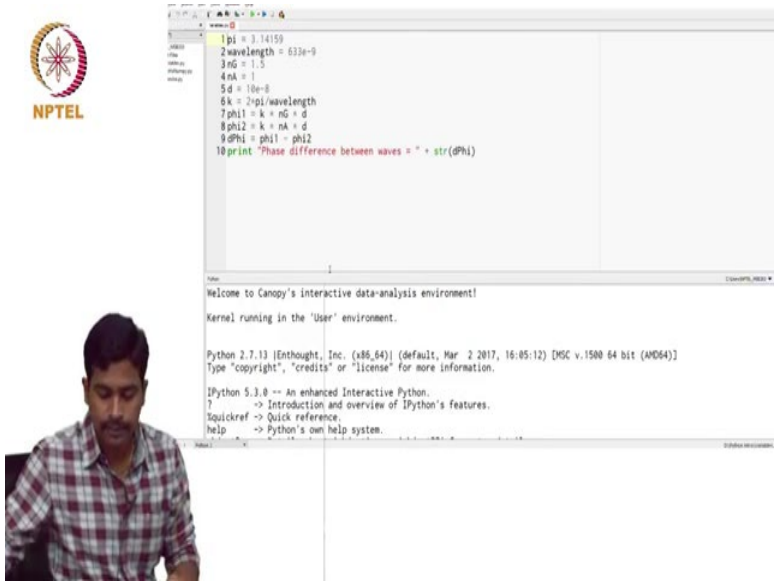
So, now I will just show quickly how to run this program. So, this is the software that I ask you to install which is called canopy.

(Refer Slide Time: 12:59)



So, to open this you just go to programs enthought and code editor. If you click this, you will see this screen. So, I already opened it, so it does not take a lot of time and I have already stored this ok.

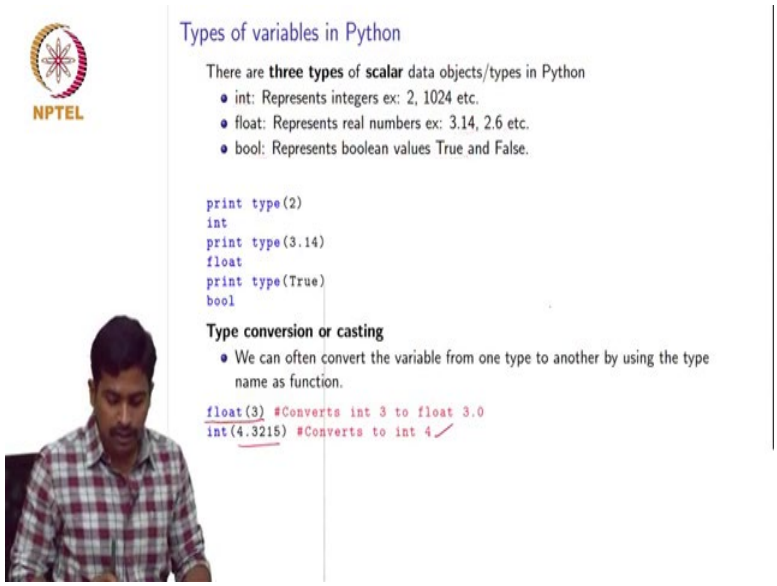
(Refer Slide Time: 13:21)



This is the program I will increase the font size. Can you see it now or you want it still bigger? Is it ok?

So, I will explain how to use the software in a few minutes, but for now just see the program running. So, this is the program and this is the button you have to click to run the program. So, now, here it is asking I mean; it is just giving the phase difference between the 2 waves is 0.49 radiance ok. So, now, we have seen some example programs.

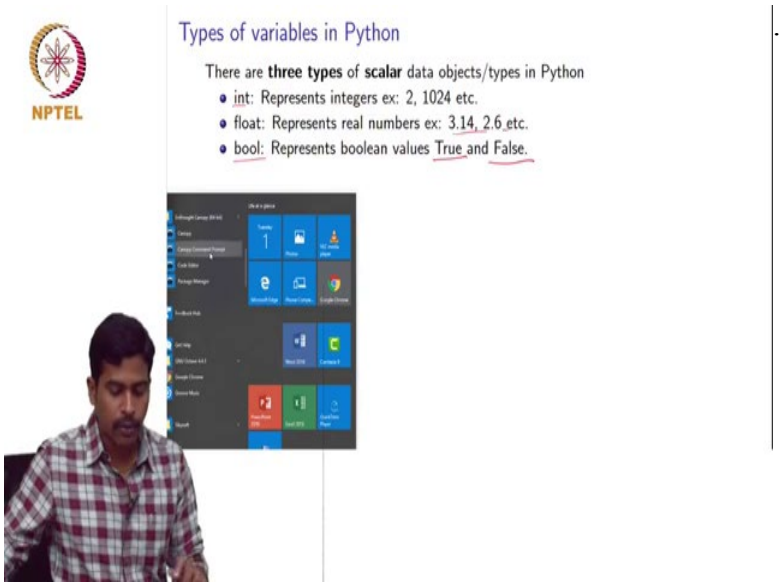
(Refer Slide Time: 14:08)



So, let us go deeper about what are variables in python. So, as I have said before in python there are three types of variables; the integers, floats and boolean type of variables.

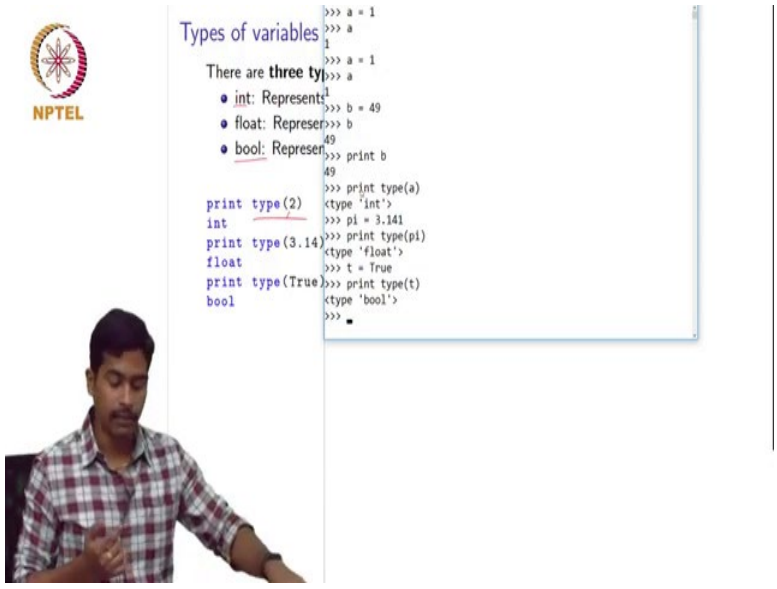
So, integers you know they are all real integers. So, they do not have fractional parts and in integers there are 8 bit integers, 16 bit integers. So, you do not have to worry about it, the size number of the more number of bits, you have in integer the bigger number you can represent. And similarly float you can present decimal numbers and boolean it has only true or false values ok. So, now, I want you guys to open the python shell. So, you can try some of these commands by yourself.

(Refer Slide Time: 14:52)



So, to do this, I hope you already installed the software. Just go to the same enthought, here you just open this canopy command prompt ok.

(Refer Slide Time: 15:05)



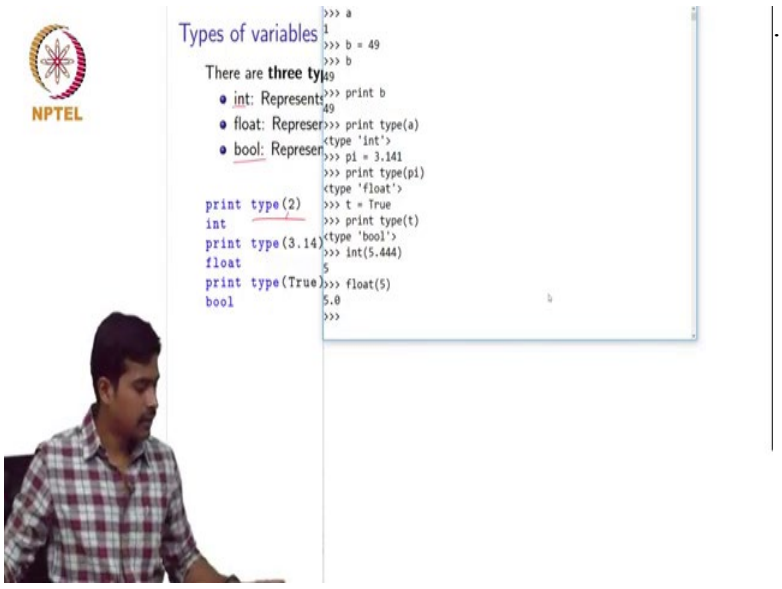
So, I have already opened it and this is how the command prompt looks ok. So, this is called the python shell. So, in python shell what you can do is, instead of writing a big program you can just give commands one at a time ok. So, for example, I am showing you, we can say an equals to 1 and the value of a is stored in the python in python in memory ok.

So when you just in the next line if you say a and it prints out the value of a as 1. Similarly, you can say b equals to some 49 and if you say b it will give the value of b back. You can also try print print b, it will print the value of b ok. So, now, I have , so now, we have seen there are three types of variables and there is a command in python to know what is the type of variable ok. So, for example, we know two is an integer variable, but we want to know what is the type of variable. So, just go to this python shell you can say print type of a.

So, since a is a integer type it will print the type of a as int. And we will see another example where we see$\pi = 3.141$and now we will ask the python to say what is the type. So, it gives the type to be float ok. So, these are the simple commands that we can use to know the type of a variable. And similarly, we will see what happens with a boolean variable. So, I will just write it True and. So, t is a boolean type of variable.

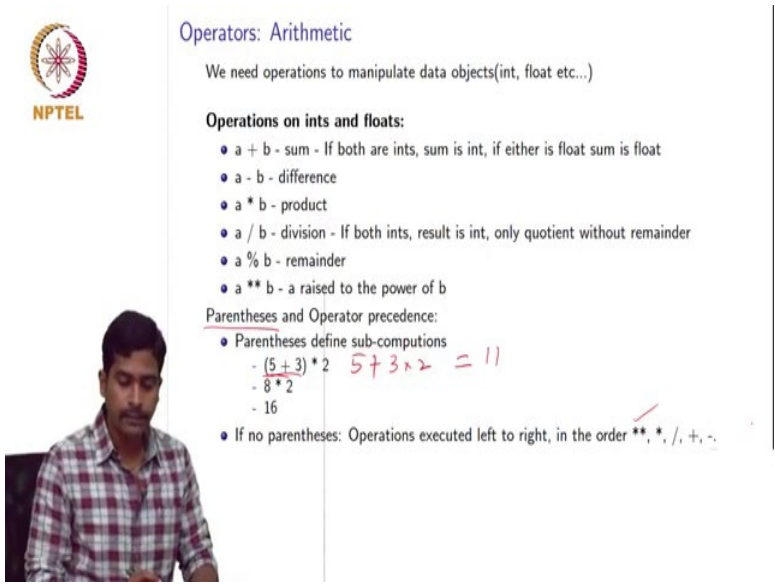
So, in a for example, in your program sometimes what happens is. So, you have a variable that is an integer or float ok. But you want to convert from integer to float or float to integer. So, in python there is a way of doing it. So, if you have a number 3, which is an integer if you want to convert into a float. So, you can use this command float and you pass the number that you want to convert into float. So, now, here 3 will be converted into 3.0 ok. And similarly, you can convert this 4.3215 which is a float number into integer by using this int of command and it will return the value 4.

(Refer Slide Time: 17:33)



So, we can just quickly try those commands here. So, it gives us the value 5 back, you can give 5 so it becomes 5.0 ok. All of you are trying or you have not installed? I highly recommend it because you guys if you do when you do it faster you can learn python very fast ok. This is called type conversion or type casting in python to change one type to another type.

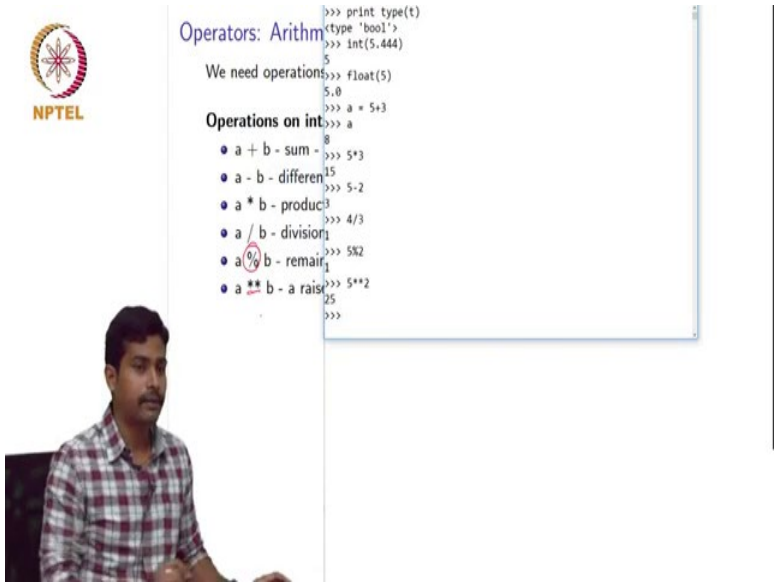
(Refer Slide Time: 18:06)



And we have seen variables and now we will go onto what are the type of operations that we can do on these variables.

So, the first type of operation is called arithmetic operations. So, by name itself it says arithmetic means; addition, multiplication, division, subtraction and so on. So, these are the symbols used to do these operations in python. So, plus is addition, minus is subtraction and hash we use this for multiplication and division symbols we use for division.

(Refer Slide Time: 18:39)



So, we can quickly try these things in python shell. So, we can see an equal to 5 plus 3 and we just say a it gives the value similarly or else we can just type 5 *3 is 15 and 5 - 2, 3 and so on yeah. So, if you had noticed here 4/3 = 1.333.

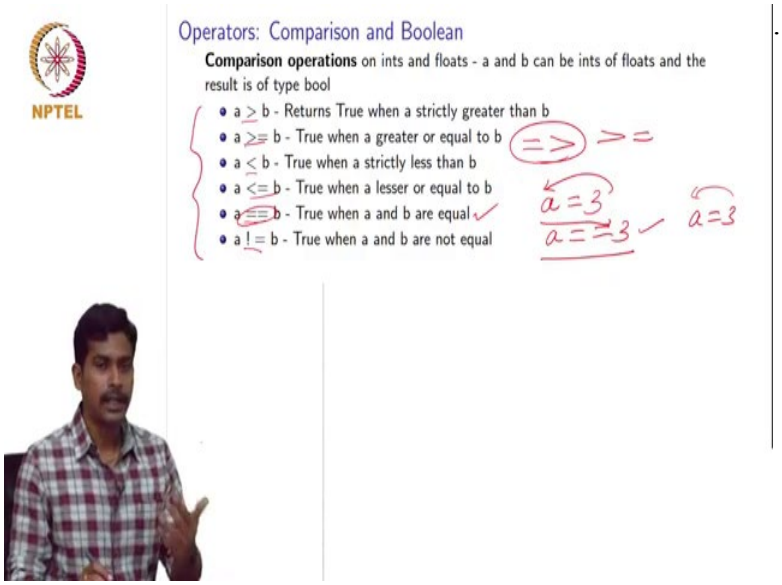
But here both of these variables are of integer type. So, the result of this computation will also be integer. So, this is another thing you should always keep in mind when you are dealing with integers. If you are just dividing 2 integers, you should always take care to at least convert one of them into a float. So, that the result will be a float otherwise you will lose some mistake that will happen in your program and you do not realize, you will get some error in your output.

So, these are the basic arithmetic operations and there are other operations for example, this percentile operation which is a reminder operation. So, if you want to know a reminder when you divide a number with another number, you can use this percentile symbol and if you want to calculate power of a number you can use this double hash track ok. Single hash track is for multiplication and if you use two hash tracks it is power. So, let us quickly try those things.

So, 5 percentile 2 is 1 and 5 power 2 is 25 ok. So, now, we have seen these yeah. So, another important thing that you have to keep in mind when using these arithmetic operation operations is where you keep your parenthesis. For example, if you have a line of programs. So, that has so many arithmetic operations like additions, multiplications and divisions and you have to keep your parentheses properly instead of writing 5 plus 3 in brackets if you write 5 plus 3 into 2. So, what happens is the result will be can you just tell it is going to be 11 instead of 16 ok.

So, you have to keep in mind where to put your brackets correctly. So, that you will get the desired result and in a python; operations are executed from left to right in this order. These are the precedence of operations. So, first is the power operation then multiplication,and division, addition and subtraction ok.

(Refer Slide Time: 21:08)



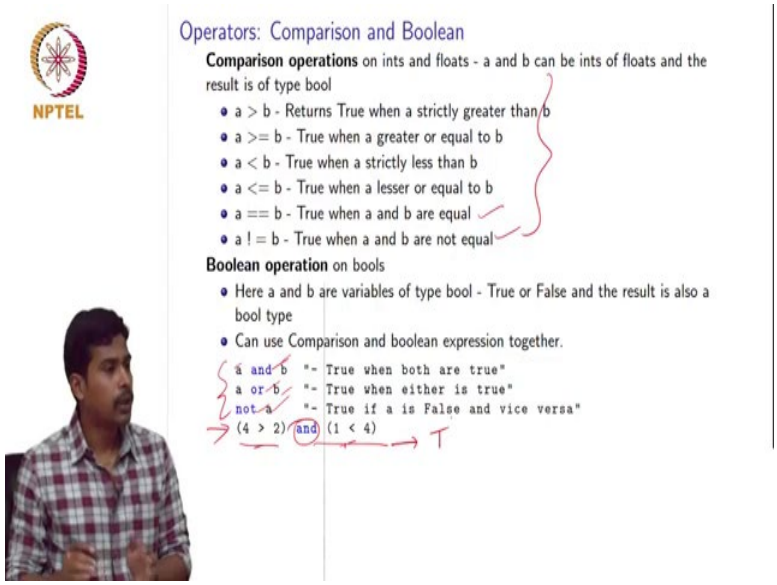
So, now, we have seen different kinds of arithmetic operations that can be done on both integers and floats; obviously, we cannot do addition using boolean variables. So, now, let us see what we can do with these booleans or how do we do comparison between two numbers ok. So, in python we have 4 types of comparison, actually 6 types of comparison; one is greater than, greater than or equal to, less than and less than or equal to. There are two more comparisons to check if two numbers are exactly equal or not equal.

So, to do this we use these symbols greater than, greater than or equals to and the order should be the same. You cannot use equal and greater, this is illegal. You have to use greater than equals to be ok. Similarly, less than less than or equal to and double equals to compare two numbers and not equals to to see if they are not equal and the many people make this mistake. So, for example, if they say an equal to 3. So, what happens is the value of three is assigned to the variable a, but by mistake for example, if you want to see if it is equal to 3, you have to use this statement.

Beginners in python or any programming language tend to make this mistake when they want to compare; they just say an equal to 3. So, this is wrong, because here you are assigning the

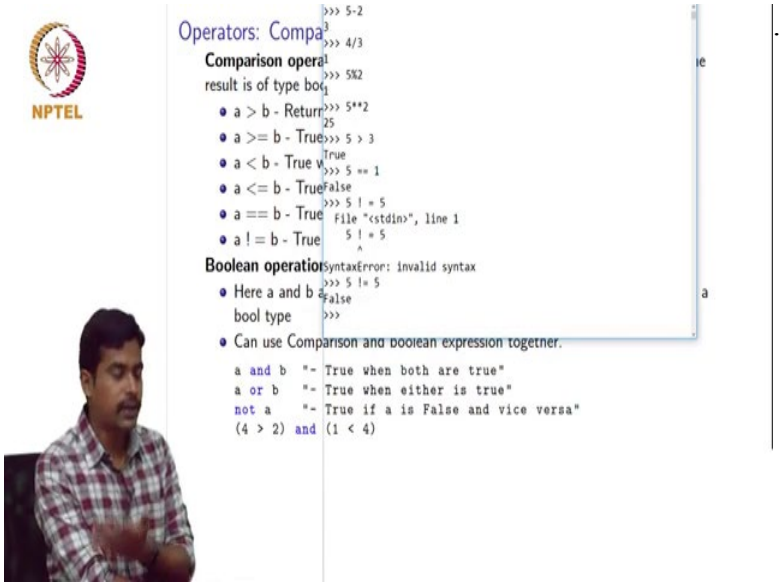
value to a and you are not comparing. So, this is one thing that beginners should keep in mind and not equate to give so, yeah. So, the output of a comparison variable is always a boolean type of variable ok. So, when you are comparing two things. So, outcome can be either true or false it cannot be integer or float ok. So, for all these comparison operations, the output can be either true or false.

(Refer Slide Time: 22:58)



So, if you, we will just try out some of these new comparison operations in the python shell.

(Refer Slide Time: 23:08)

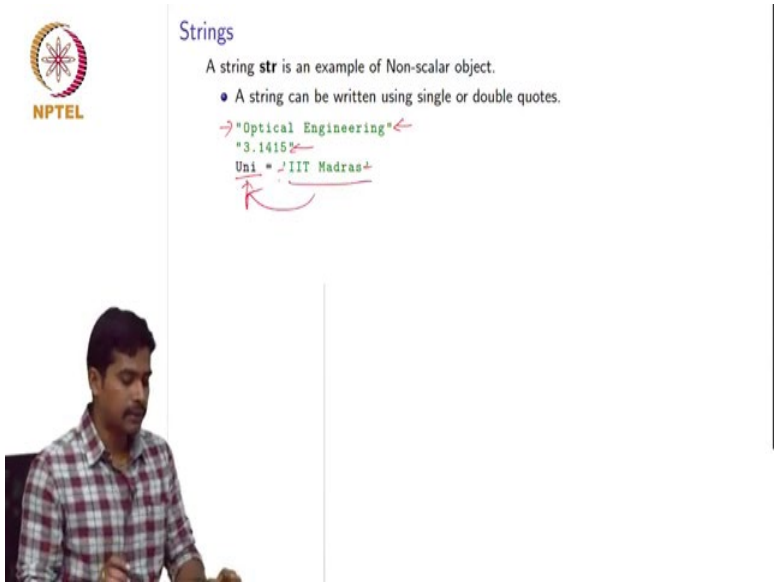


So, we can say 5> 3 which is true and we got the outcome true we can say 5=1 which is false, we can also say 5 not equals to 5 yeah there is a syntax error, because there should not be any space in between the not equals to symbol and equals to there should be next to each other. So, since 5 not equals 5 is an incorrect one. So, we are getting the value false ok.

And we have seen comparison operations can be performed on integers or floats ok. You cannot compare true or false with each other. We can compare using equals to, but not the other greater than and less than. So, now, we will see what are the different kinds of operations that we can use on boolean type of variables. So, like these are the three allowed boolean operations in python. So, which are and, or and not. So, we all know what is an and if both a and b are true then only the result of the operation will be true in other cases it is going to be false.

Similarly, for a or b it is only false when both are false and in all the other 3 cases it is going to be true. Not a is going to flip I mean; if it is true it is going to make it false and if it is false it makes it true ok. And also we can use these comparison operations in conjunction with these boolean operations. For example, if you see this statement. So, 4 greater than 2 will give us a boolean output which is true and 1 less than 4 is also true. So, the outcome will be true and we have applied this and boolean operation with between these two boolean variables ok. So, the result will be true because true and true is true.

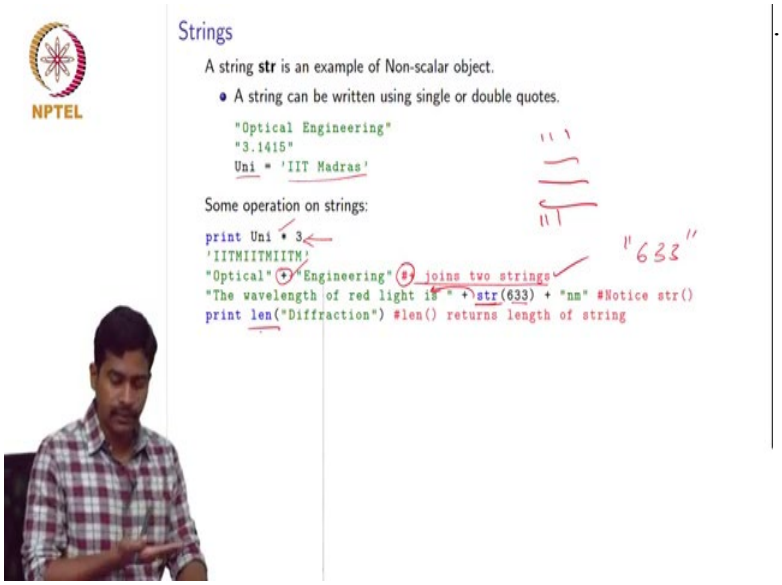
(Refer Slide Time: 24:57)



So now let us see what strings are in python. So, string is nothing but a sentence or group of characters that we write in python. For example, in a python the string always should be

enclosed in double quotes or you can also use single quotes. So, both are treated as strings, there is no difference. For example, here Optical Engineering is a string and you can also declare numbers as strings. For example, here 3.1415 is enclosed in braces which means this is a string and not a float value ok. And you can also assign variables to string for example, here Uni is a variable and the value string is assigned to this variable ok.

(Refer Slide Time: 25:40)



And let us see some of the operations that we can do on strings. So, this is one operation that can be done on string. So, here what happens is as we have declared earlier. So, the variable Uni is assigned with a string called IIT Madras. When I say multiply it with 3, what happens is, the string is repeated three times and it makes a bigger string. So, that is one operation and the other operation plus plus means concatenation operation. So, what this does is it will take two strings and then it will join these strings together ok.

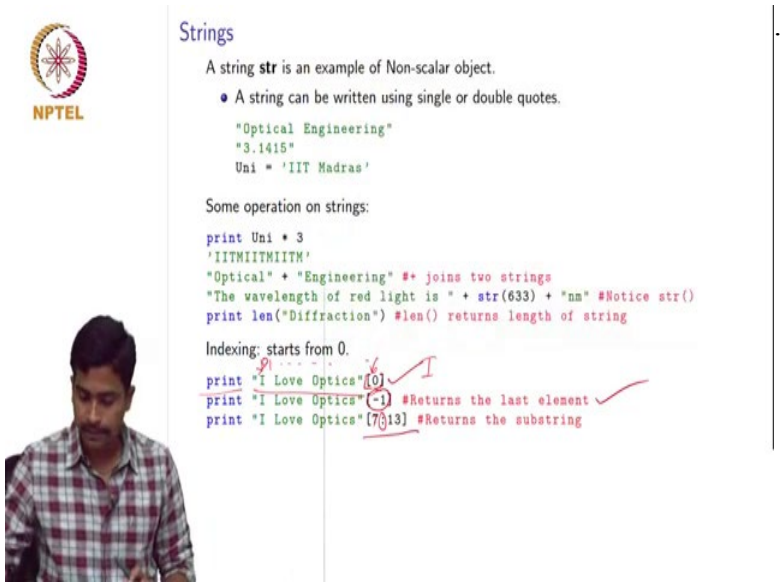
So, here another thing I want you to notice. So, this hash symbol. So, in a python program if you want to write a comment. So, which should not be executed in your program, you can use this hash symbol and you can write anything after that that will not be executed in the program and using comments is very useful because when you write a program and if you look at the program after a few months. So, you may forget what you have done using your variables or your statements.

So, if you use these comments it will be easy to find out what your program is doing. Another way of writing comments in python is using three quotation symbols and you can write your

state comments here and then close again with 3 single quotes. So, this is another way of writing comments; this is also called block comment. So, yeah so, we have seen the multiplication and this plus operations using in the context of strings.

And now there is another operation this was also seen in the first slide. So, which is a string keyword. So, what this does is; it takes the number 633, it converts it into a string 633 and it joins this to the previous string and it makes a single string ok. And in python you can also find out the length of a string for example, if you want to know how many number of how many characters are there in a string you can use this length command which is a in build command to know the number of lines in a string and most of people may be working with MATLAB.

(Refer Slide Time: 27:43)



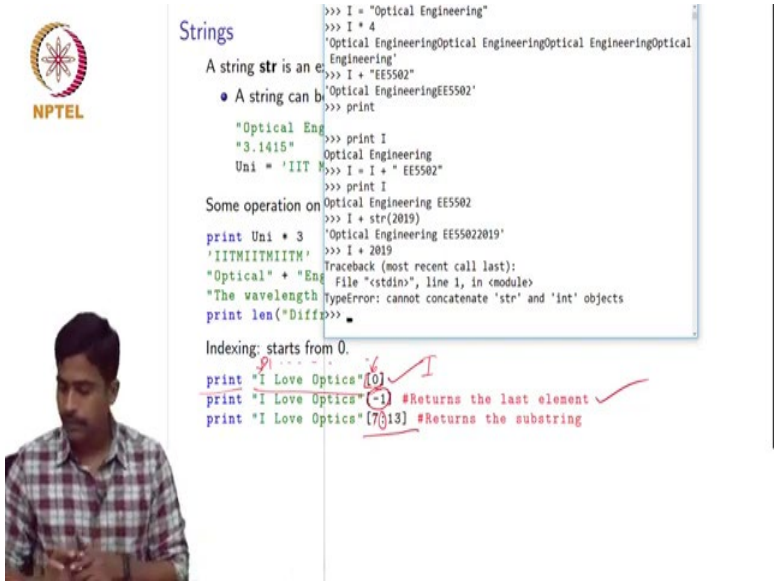
So, I want to emphasize here the key difference between python and MATLAB here is in python. So, the indexing starts from 0. So, what do I mean by indexing is so, the counting I mean; how do we count the numbers of a string. So, the counting starts from 0, if you know MATLAB before. So, the difference is in MATLAB the counting starts from 1 ok. So, if you are switching from MATLAB to python be careful to remember this point this is very important.

So, for example, if you want to print a particular character in a string like tenth character or first character. So, we can use this square braces and inside the square graces we can give what is the index of the character that we want to print out. So, we can just say print and we can give some string. So, this will give the letter I at the output.

Similarly, another important thing in python is you can if you do not know the length of a string for example, you do not know if there are 40 characters or 100 characters, but if you want to know what is the last character. You can use this minus 1. So, minus 1 means the last character even if you do not know the length if you just say minus 1, it will print the last character in the string ok

You can also group I mean you can also index groups of characters like seventh to thirteenth or first to fifth character using this command. Here you use the starting index and the new semi colon and the last index. So, here seventh to thirteenth characters will be printed at the output.

(Refer Slide Time: 29:31)

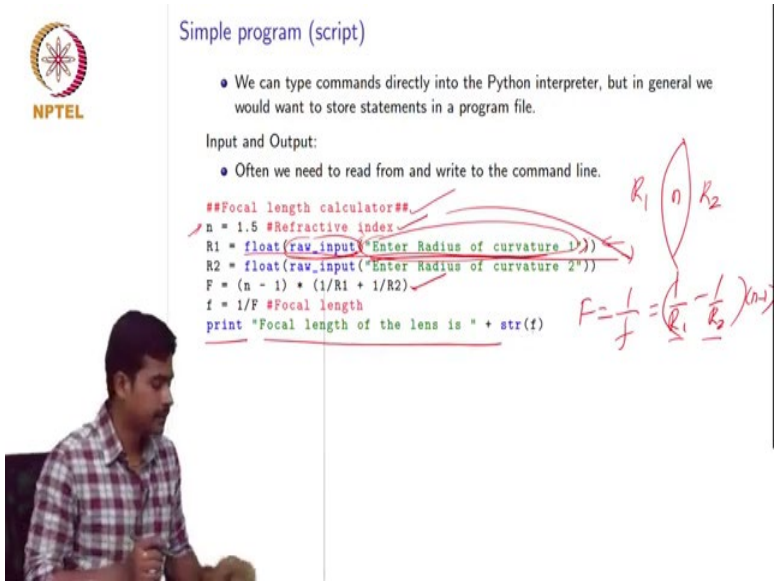


So, now let us try some of these in the python shell ok. So, it is a variable that is storing the value of a string that is optical engineering. So, now let us see what happens when I say. So, now, the optical integrating string has been made into a bigger string repeating itself four times. And now let us see I plus so, the output is Optical Engineering EE 5502. So, even though we have done this operation. This is not going to affect the value of the original string. So, if you print sorry so, I am still going to be an optical engineer and we have not changed the value of I, but if you do this ok.

So, now if you tried to print. So, now, the value of the string is modified ok. And we have seen the concatenation and multiplication. Now we will see if we can convert numbers to strings and then add to I ok. So, now, it has added a 2019 character to this, but without converting if

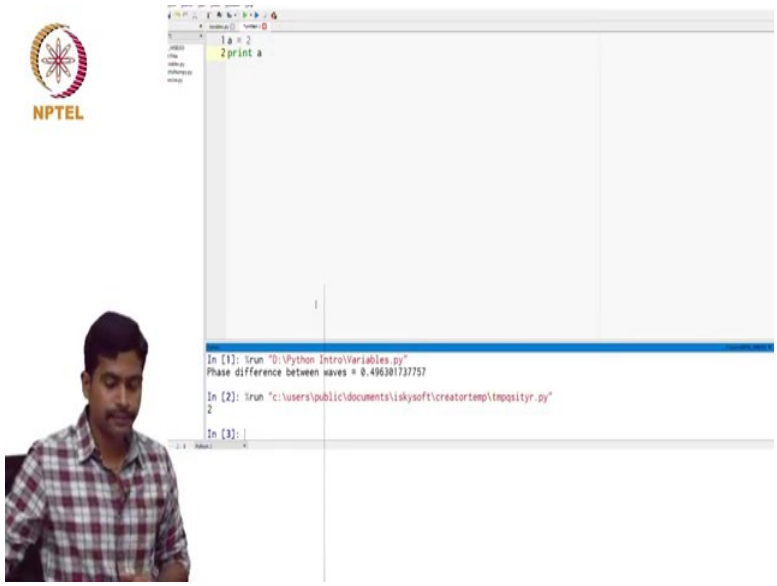
you tried to do this if. So, it will give an error because, integer and string are different types of data types and you cannot merge them together ok.

(Refer Slide Time: 31:18)



So, now we have seen different types of operations on strings. Now let us try to write some simple programs ok. So far what we have been doing is we are opening this python shell and then we are typing different commands arithmetic or comparing whatever commands we are typing in the shell. This is not a very good way of writing bigger programs. For example, if you want to write a big simulation program or if you want to create some application. So, you cannot open this command and write your application because it gets executed after each statement. So, instead what you can do is you can open a text file.

(Refer Slide Time: 31:59)



For example, so, you can open this editor and you can open a new file and you can write all of your program in this file and then you can execute the whole program one at once instead of executing one after another ok.

So, in order to open this program; I hope you are already installed, go to enthought and you can open this code editor. So, when you open this code editor it will open like this. So, here you can start writing your programs for example, I will write a simple program. So, what happens here is when I execute it. So, all the statements in this file will be executed one after another ok. So, now, let us go back and see the simple program that we want to write.

So, before going to this program, I want to introduce another important thing. So, in some of our applications or you know some of our programs. So, we want to take input from the user or we want to add some numbers into our program. So, that the program will work based on the input that we give ok. So, far we have not seen how to give data to python.

So, in python; so, there is this command that we use to enter value into the program ok. So, the main command is raw underscore input. So, raw underscore input takes your data from the console and then if you know if for example, you can give whatever the. So, in raw underscore input whatever the value enters it will be by default a string type ok. So, as I have said before you cannot do arithmetic operations with strings. So, you have to convert it into an appropriate type of data type whether it can be an integer or float ok.

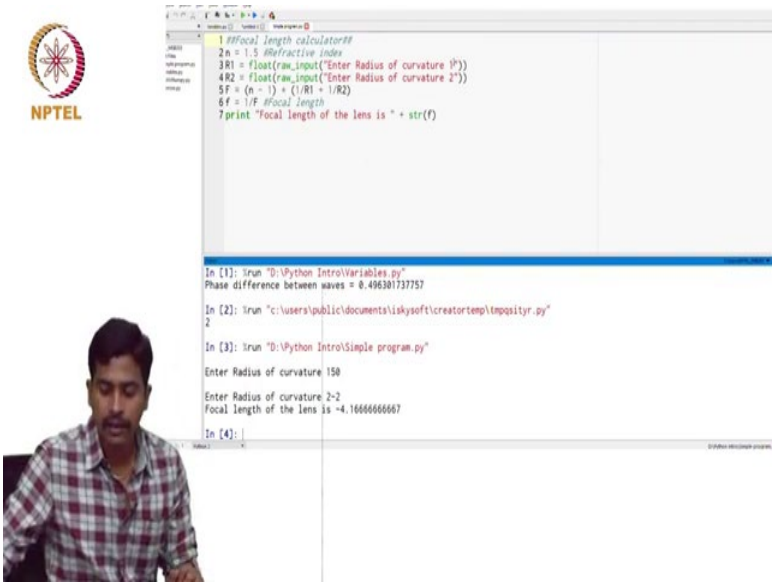
So, in this example what we are going to do is to calculate the focal length of a lens ok. So, the focal length of the lens is given by in order to know the focal length of the lens you should know it is the radius of curvature and the material of the lens ok. So, for example, if this is a lens and this is $\square_1$ this is $\square_2$ and if n is the refractive index, the focal length is given by $\frac{1}{\square} = \left(\frac{1}{\square_1} - \frac{1}{\square_2}\right)\square - 1$ ok.

So, now we are writing a program to calculate a focal length. So, given the values of radius of curvature 1 and radius of curvature 2. So, in the first line, we have declared the refractive index n to be 1.5 and this value we are keeping it constant and we are not taking it from the console, but we want to take the radius of curvature from the console. So, we all know the radius of curvature is a float value. So, this raw input whatever we receive from raw input will be converted into float ok.

So, you can also in the raw input command, you can give a string. So, what happens is when this executes. So, this statement will be prompted on the screen. So, that you will know what is the expected value from you. So, if you do not include this string, it will just be a blank space. So, you do not know what you are supposed to enter ok.

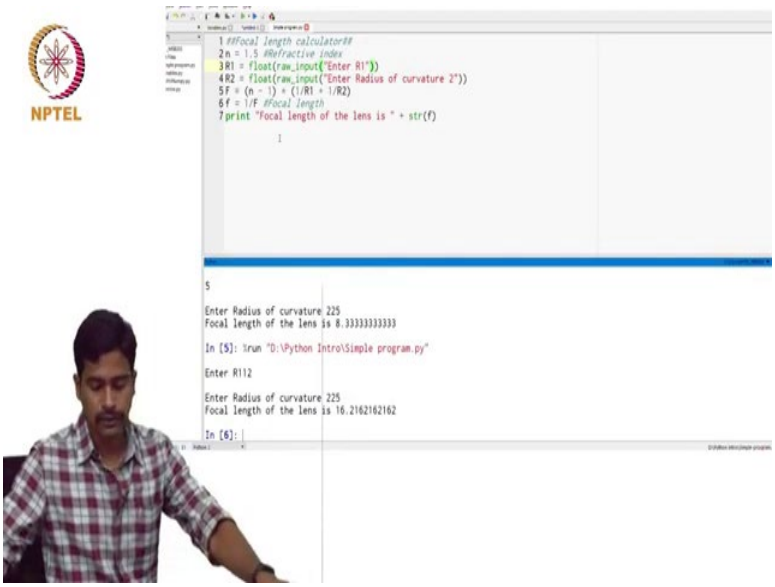
So, to understand this program and this is the formula that we use to calculate the focal length and capital F is this and in order to know the focal length we just have to calculate 1/F. And finally, we want to print the focal length of the lens so and so, value.

(Refer Slide Time: 35:47)



So, now let us go back. I have already written this ok. So, now, let us try to run this. So, here at the bottom if you see it is asking for the radius of curvature let us give some value and it is asking for the radius of curvature for the second surface. So, now, it has calculated the focal length and it is printing the value of focal length ok. So, now let us try to modify this program a bit.

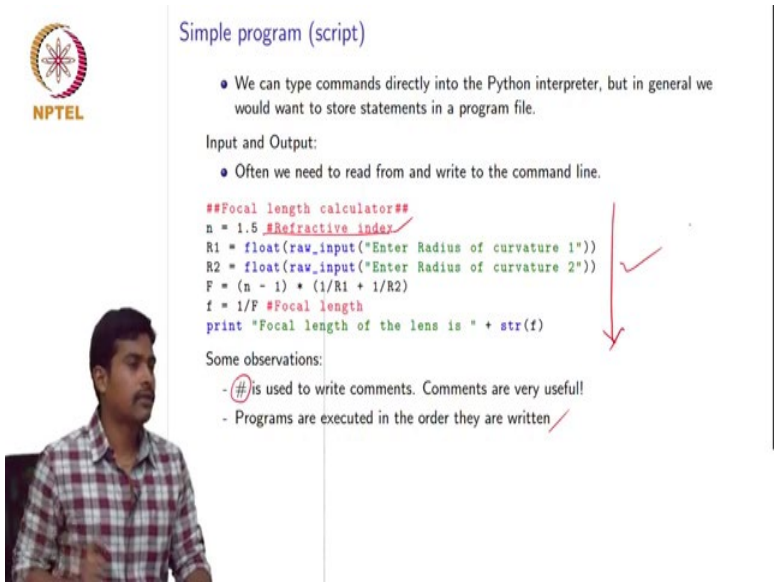
(Refer Slide Time: 36:14)



So, what if I remove this line here ok. I just gave raw input and then this raw input is being converted into float. So, if I run this program and it is not asking to enter the radius of curvature of the lens, but it is just blinking it is waiting for us to give a number so, if I say 5. So, now, it is going to the next line where we have given the string and it displays. So, what is the radius of curvature of the R2. So, this is more informative. I mean you will know what you are supposed to do ok.

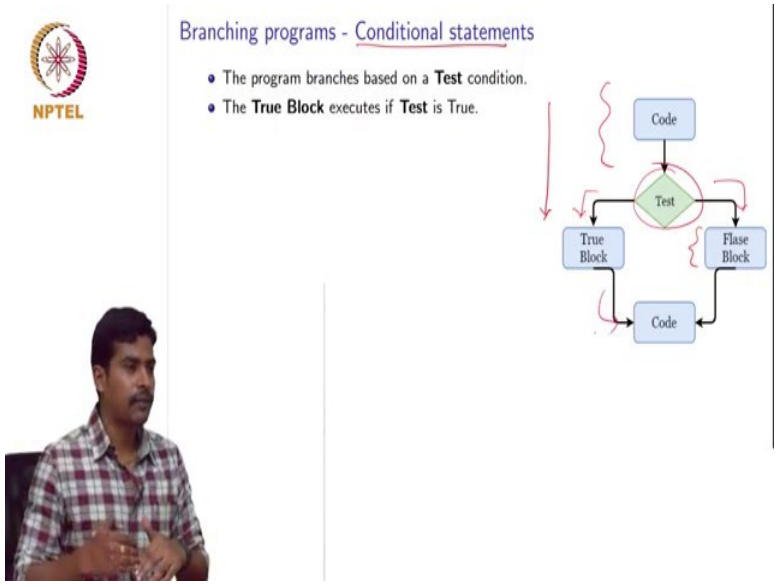
So, now let us give some other value. So, now, I have again calculated the focal length and the value is 8.33. So, you can write any kind of string inside this and now we will run it. So, it will just say enter R1. So, this is the focal length yeah.

(Refer Slide Time: 37:12)



So, I hope by you understanding, you all understood how we write a simple program and how it runs in python. And these are the observations that I want you guys to make at this point. So, as I have said before, this hash symbol is used to write comments. So, this is very useful when you are looking at your programs after a few months or few weeks. You will know what is your program each line of your program doing and another thing is so, the program is executing in the order that we have written ok. So, it is not executing the first line and then it is jumping to the third line it is not doing all those things, it is just executing one line after another line ok. So, this is good in most of our programs.

(Refer Slide Time: 37:56)

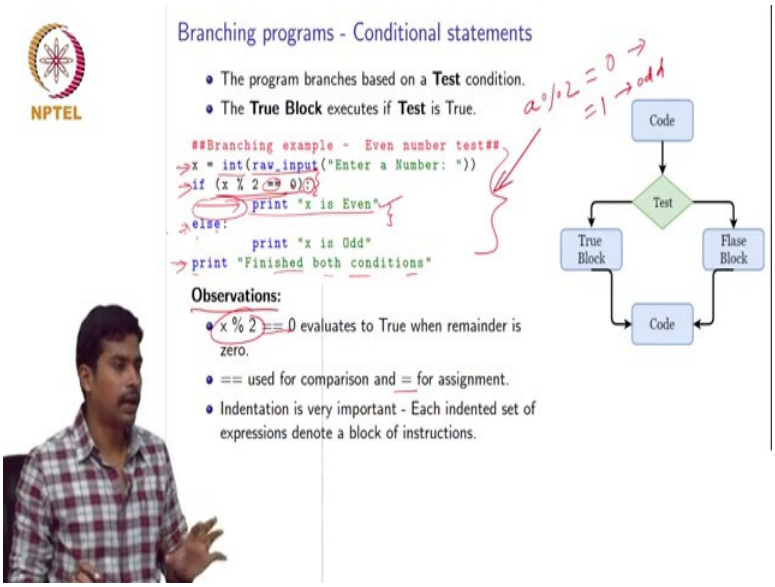


But what if you want to execute a particular part of your program when a particular condition is met ok. So, for example, you have two radii of curvature values. So, you want to use one formula if you all know there is a thick lens and thin lens right. So, for example, if you want to write a program and you want to, you have two different formulas; obviously, we have two different formulas to calculate focal length for a thick lens and a thin lens. So, when you ask the thickness of the lens, based on the thickness of the lens you want to use either this formula or that formula right.

So, in order to do that kind of branching in your program; so, we need something called conditional statements or branching statements ok. So, we all know what conditional statements from other programming languages the if and else statements are ok. So, this is the way these programs execute. So, you have a bunch of program lines and after a particular line you arrive at a point where you want to decide either I want to run this part of code or I want to run some other part of the code.

So, you write a test case if the test is true. So, you will write this, you will execute this bunch of code and if the test is fall you will run another bunch of lines and finally, you will go to the next part of your program ok.

(Refer Slide Time: 39:20)



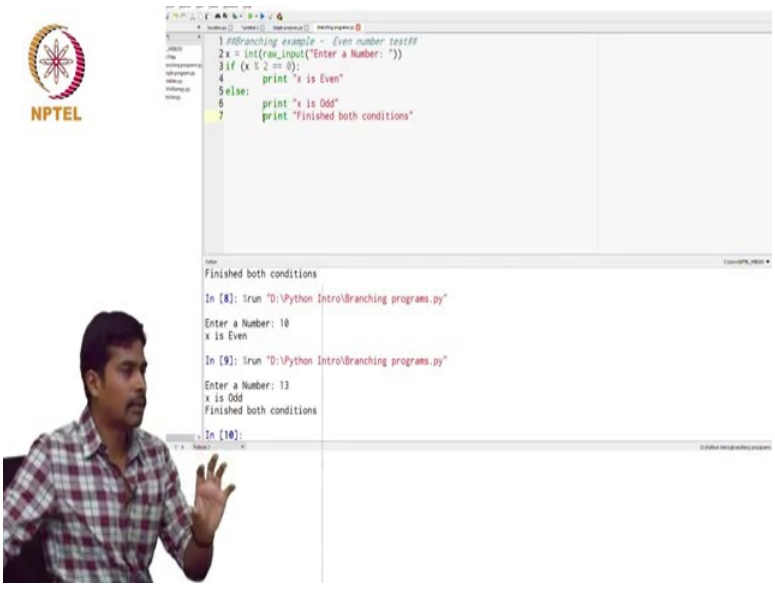
So, in order to understand this, I want you to see this program. So, here what we are doing is we are asking the user to enter a number and we want to check if it is an even number or an odd number. So, in order to check if a given number is even or not. So, the simplest test is to

do percentile with 2 and if the percentile operation is 0, it is an even number and if the percentile is 1 it is an odd number ok.

So, the same thing we have converted to writing a simple python program. So, here what we are doing is we are using raw input commands to take the value from the user and we want to convert it into an integer here rather than a float like before and here we are using an if statement. And in this if statement, so we are calculating the percentile and the output we are comparing with 0 and if it is true as I have said before there are no open and close braces in python. So, we have to use this semi colon. And after semicolon the moment you enter the python editor itself will give this extra spacing and here you have to write whatever commands you what to write.

So, here we want to say it is an even number and we use this command and if this condition is not true. So, the program will come to the else statement and in the else statement we want to say x is an odd number. And finally, we after these two are executed the program comes here after one of these gets executed the program comes here and we say I finished I finished checking both of the conditions. So, now, let us go back to our python.

(Refer Slide Time: 41:00)



So, this is the same program I have shown before. So, if I run this, so, if I give for example, 25. So, 25 is an Odd number what happened here. So, it calculated 25 percentile to since it is not equals to 0, it does not go into this branch and it goes to the else branch and in the else branch it prints this x is an Odd number and finishes both conditions ok. So, similarly if we give an

even number. So, it says x is an Even number and it finished both of the conditions ok. So, another thing I want you to notice in this is.

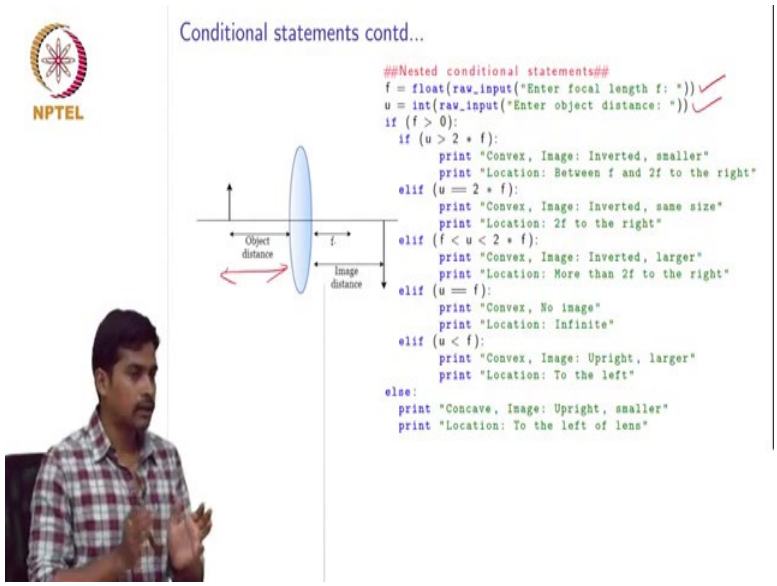
So, after you write the statement if you press enter. So, the cursor is still at this point, it has not moved to the starting point. So, instead of writing else here if you write else. So, it would not work ok. So, if you in order to write the else statement you have to come all the way to the starting and then you have to write otherwise if you write the statement here itself. So, what happens is for example, if you say equal to 2 or something 3 or something. So, the program will execute in this block for example, if you want to execute the statement irrespective of what happens. So, I mean you cannot write it here because it should be followed by something else.

So, you have to move to the starting ok. Or else I will demonstrate what happens if you are ok. So, now, we have seen, if you do not write it here for example, if you write this statement inside an else statement if you execute. So, we gave an even number x is an even number and it has not printed this particular statement because it is inside the else block.

Similarly, if you give an odd number, since this is an odd number and the program is going inside the else block and it is printing both of these statements ok. So, you always should keep in mind where your next statement is going. Whether you want to include it in that block or you do not want to include it in that block, ok.

yeah So, these are the observations that I want you guys to make at this point. So, the percentile operation is used to calculate the reminder and this double equal is used for comparison and single equals to is used for assignment and indentation is another important thing when you are writing if or else or for kind of statements ok.

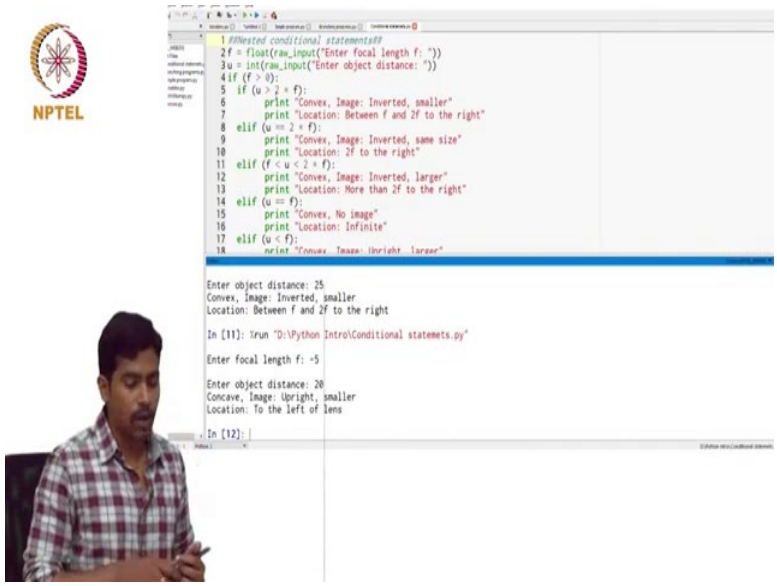
(Refer Slide Time: 43:50)



So, now that we have seen conditional statements; so, let us try to write another example program. So, in this example, we are trying to model what happens when an object lies at a distance x from your lens, so ok. So, we all know based on the object lens object distance from the lens the image distance is going to depend right.

So, this is the program I want you guys to open. So, the user has to ender the focal lengths of the lens and the user has to enter the object distance, which is this distance. So, based on these two values so, it is going to say if the image is going to be a real image or a virtual image or if the image is going to be inverted or upright image and about the position of the image whether it is going to be at the infinity or it is going to lie within f or 0 to 2f. So, it is a simple program that uses the conditional statements.

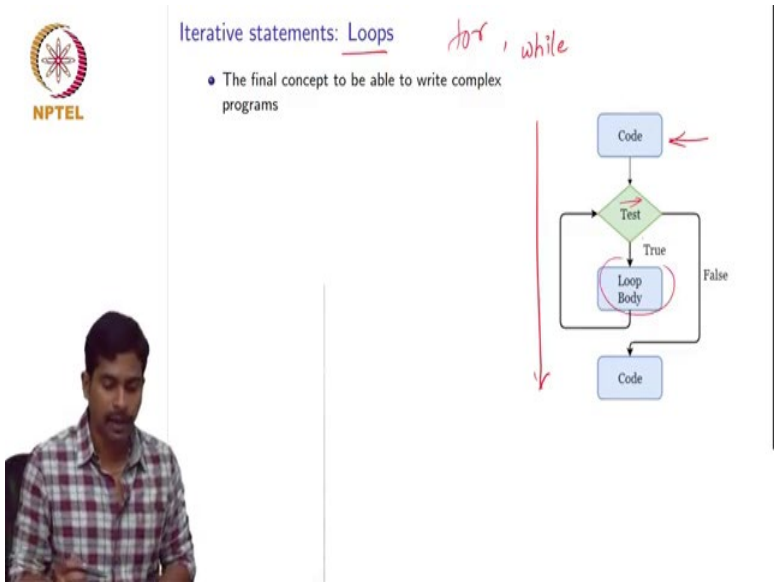
(Refer Slide Time: 44:54)



So, I will quickly run this program; so, we will just give some focal length 10 and object distance 25. So, with these parameters, the lens is convex because the focal length is positive. We know the focal length if the focal length is negative it is a concave lens right. So, because here focal length is positive it is a convex and image is going to be inverted and small it is not going to be magnified it is a smaller image and location is going to be between f and 2f to the right of the lens. So, that is the information.

So, the program is giving you based on the value of focal length and let us try to give some other values maybe minus 5 and 20. So, here the lens is a concave and the image is going to be to the left of the lens and it is upright and it is again a diminished not magnified image ok.

(Refer Slide Time: 45:47)

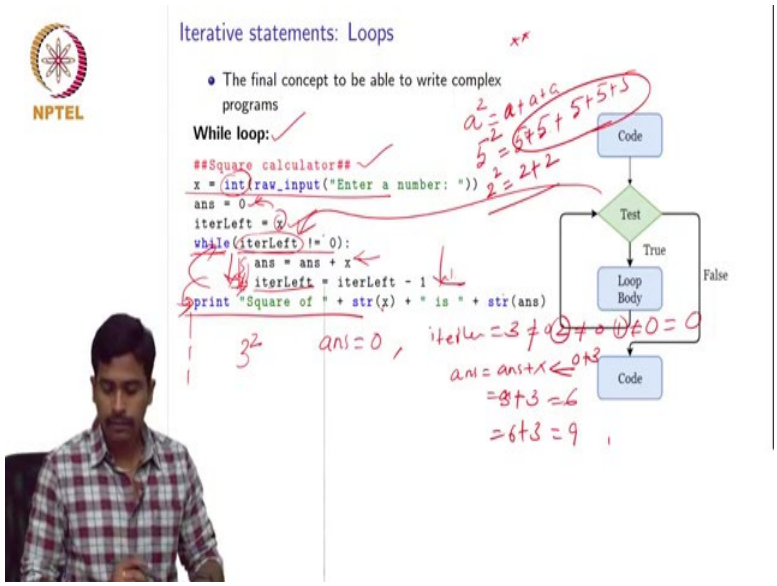


So, now let us go to iterative statements. So, as I said before if you have a block of code that you want to run multiple times instead of copying the same code. So, many times you can use these iterative or looping statements.

So, in python then we have two different types of loop statements one is the for loop and the other one is while loop. So, this is how the iterative statements in python works. So, you have a block of code and after the block of code you have another block of code that you want to execute multiple times. So, for example, if you want to run it 10 times you have to check.

So, how many, what is the iteration number? So, if you want to exactly run 10 numbers of times it should keep track of the iteration right whether it is the first iteration or second iteration. So, for that there is a test to see what is the iteration number based on that if it is true it will go go go to the block of code and it will execute the statements and again it will come back and it checks the condition. If the condition is false it will exit the block of code and then it will move on to the next part?

(Refer Slide Time: 47:01)



So, here I have shown an example of a while loop. So, while loop what it does is so, the key word key word to use a while loop is a while and then this is the condition as I have said we have a test condition, so, this is the test condition. Based on the test condition, this is the block of code that I want to execute ok. So, if the condition is false, it will stop executing this block and then the program will start executing from this point ok. So, this is how a while loop works.

So, in this program I want to calculate the square of a number. For example, if you give a number a. So, I want to calculate a square right. So, how I did this in this program is. So, for example, a square, so a square can be given by so, adding a, a number of times right. Or for example, if I want to calculate 5 square, 5 square can be calculated by 5 + 5 plus adding 5 5 times ok. Similarly you can calculate 2 squares by adding 2 times right. So, this is the algorithm I used to calculate a square of a number.

So, as I have said before you can also use this double star operation. So, just to demonstrate how to do a while loop I have written this program. So, here initially we want to take the number that we want to calculate square. So, I use this command and I convert it into an integer and we have declared a variable answer to be 0 initially. And finally, whatever is the answer we are going to store in this, but initially we have initialized it to 0.

And the number of iterations as we are seen here, if you want to calculate 5 squares. So, it has to add 5, 5 times. Similarly, if you want to calculate 10 squares you have to add 10 times 10

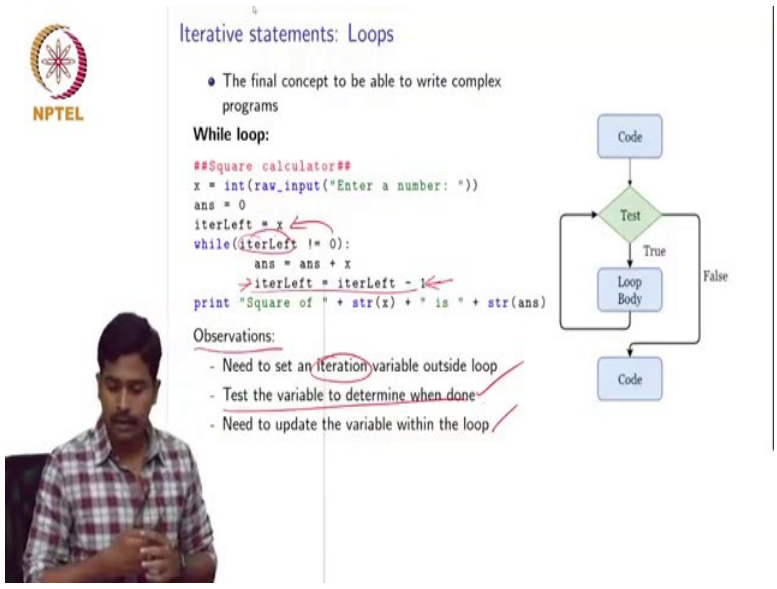
numbers to itself ok. So, we have the number of iterations here is going to be the number itself and we have written a while loop.

So, the condition here is so, the number of iterations left equals if it is not equals to 0 and we go back and then we add the answer to the number itself ok. And then we decrement the number of iterations by 1. So, now, I will explain in the case of 3 squares. So, initially we said answer equals to 0 and number of equals to 3. During the first iterations; so, it will compare 3 to 0 and since 3 is not equal to 0, it will enter the while loop ok. So, inside the while loop it says answer equals to answer plus x. So, the answer is assigned a value of 0 plus 3 ok.

Now, the answer is 3 and in the next statement we are reducing this value to 2 ok. So, again we come back to the while loop and we check if this value is equal to 0 or not. Since 2 is not equal to 0. So, it again comes back into the while loop again we add the answer 3 plus 3, because earlier answer was 3. So, now, it becomes 6 and we are decrementing the value to 1 here using this statement.

And again we go back to the condition we check 1 is not equals to 0, we again go back in the while loop and this time we add 6 plus 3 so, going to 9. And we decrement the value of iter equals to 0. We again come back to this point and we check 0 not equals to 0 which is false. So, now, we do not go into this loop we exit this and we directly come to this statement here we print square of we converted x to string so and so number is so and so. So this is how a while loop works.

(Refer Slide Time: 50:55)

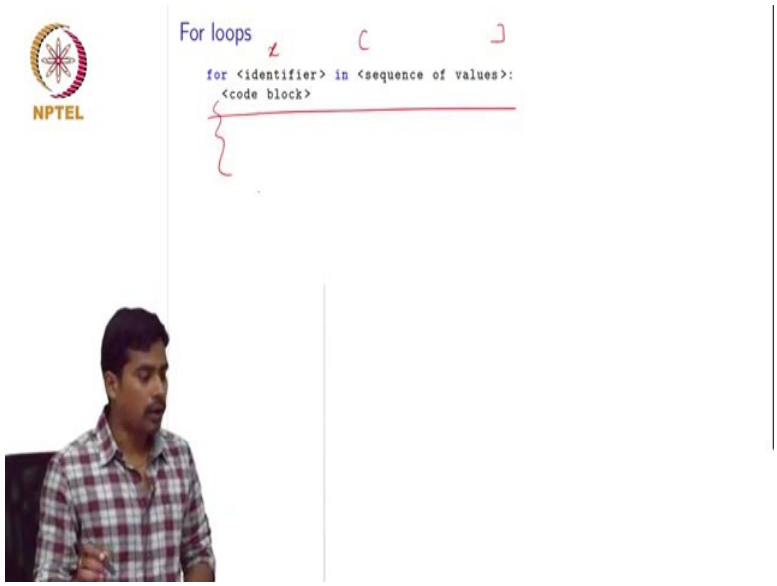


And these are the observations I want you guys to make at this point. So, for an iteration to work you always have to set the iteration number of iterations out outside the while loop ok. You cannot set this variable inside and cannot say the number. So, in order for this to work it always should already have been declared before the while loop.

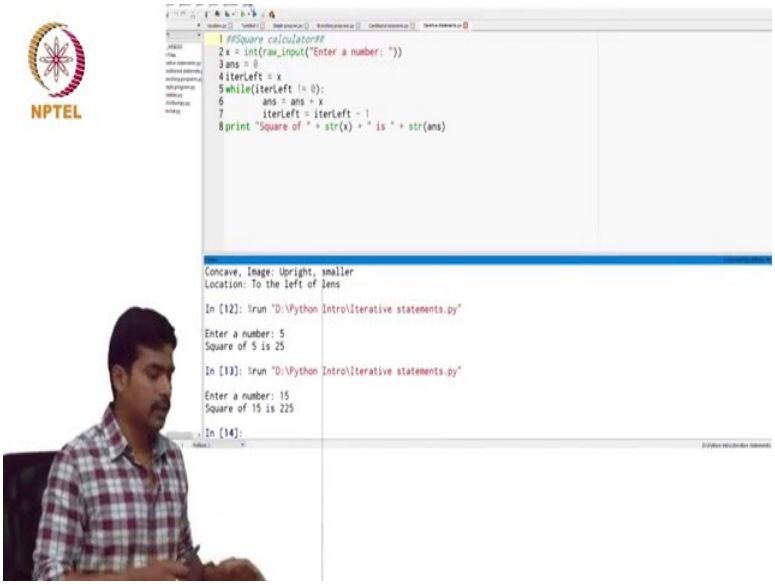
And another thing is that the test variable should be decremented. So, here this is the test variable, we are testing the while loop based on this variable and this should be decremented inside the while loop for example, if you do not decrement here what happens is. So, whenever you check the condition it is always going to be three right. So, 3 does not equals to 0 this is always true. So, what happens is this while the loop goes into an infinite loop. So, in order to avoid that you always have to make sure.

So, your teste variable will fail at some point of time. So, if it does not fail the program is going to execute an infinite number of times. So, that is a second observation and yeah ok. So, these are the observations I want you to make in the case of while loop and the next ok. This is the program that I showed on the slide.

(Refer Slide Time: 52:01)



(Refer Slide Time: 52:15)

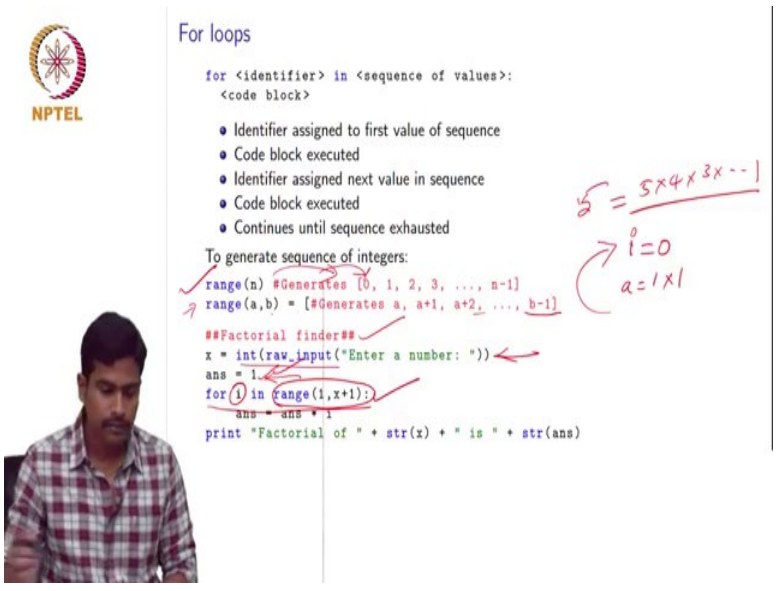


So,. So, let us give 5 and the square of 5 is 25, let us run it again and give some other number. So, 225 is the square of 5 oh sorry, 15 yeah. So, now, we have seen a one type of iterative statement which is a while yeah.

Student: (Refer Time: 52:39).

Yeah Please. So, the next type of iterative statement is a for loop. So, for loop for loop is a little different from while and this is the syntax for using a for loop ok. So, for loop followed by some variable name and they should be in a series of sequences and a block of code followed by that.

(Refer Slide Time: 53:00)

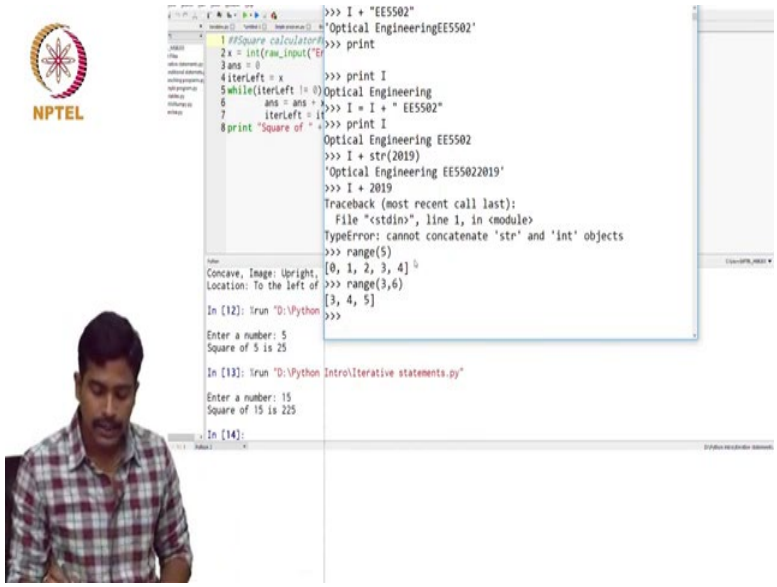


So, what happens here is. So, to see this I will show some examples. So, here the example that we are seeing is to find the factorial. So, factorial of a number 5 is given by 5 into 4 into 3 into and so on until 1. So, this is how you calculate a factorial.

So, again we have used this raw input command to take a number from the user and initialize the answer to 1 ok. And finally, we store our final result again in this variable, but we initially declare it has 1. So, this is the syntax for loop. So, in the loop we are declaring an identifier also a dummy variable here the dummy variable is I and we write another thing here is range of 1 comma x plus 1. So, what this range function does is if you say range of n it will generate a list of values from 0 to n minus 1 ok.

Similarly, if you say range of a comma b. So, it will generate a series of integers a to b minus 1 ok. So, to verify this let us just go back to our python shell again.

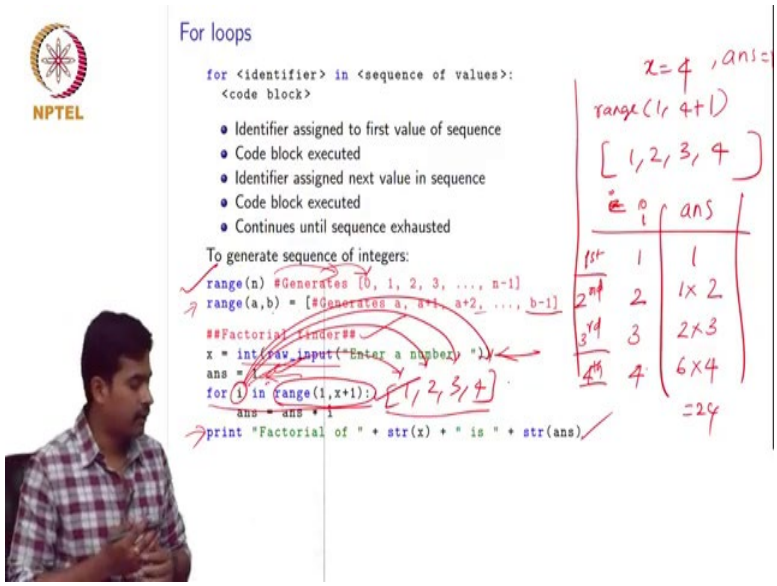
(Refer Slide Time: 54:19)



So, this is an inbuilt command range of 5 so, the output is going to be 0 to 4, not 5 3 comma 6. So, it is going to be 3, 4, 5 again not 6. So, you also have to remember. So, it does not go all the way to the b it goes only to b minus 1 ok. So, in this for statement; in each of these iterations what happens is the value of I is assigned to one of these numbers sequentially ok.

So, initially when the for loop is started. So, the I value is equal to 0 and answer equals to 1 multiplied by 1 because the value of answer is already 1 and in the next iteration. So, again here sorry ok.

(Refer Slide Time: 55:23)

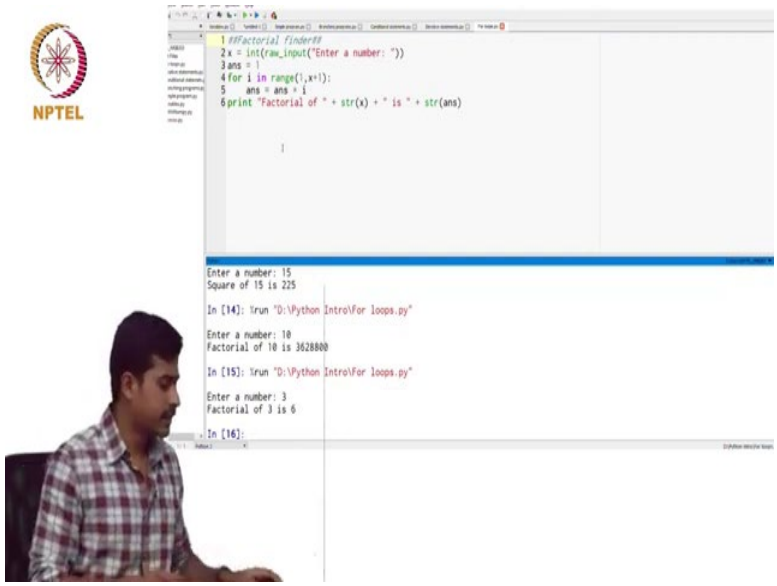


So, I will explain this program by using an example, where we calculate the factorial of a number 4 ok. So, here x is equal to four and the answer we have initialized to 1. So, again as I said before the range command 1 comma 4 plus 1 will give 4. So, these are the values written by this command.

So, when the loop is initially started i value on first iteration i value and answer value you will see in this table on the first iteration i is 1, answer is 1 right. So, after the first iteration the loop go goes back again. So, initially, i is assigned 1 after the first iteration i is assigned 2 here. So, now I equals to 2 and the answer is, answer multiplied by i answer multiplied by oh sorry it should have been i. So, it is 1* 2 and again the control goes here and this time I assigned a value 3 on the third iteration i is three and the result is 2 * 3 and on the fourth and final iteration. So, the value of I is assigned 4 and 6 into 4 24.

So, after the fourth iteration, there are no more numbers in this series of numbers. So, the loop exits at this point and then it prints the statement factorial of a given number so and so ok. So, this is the difference between a while loop and a for loop. So, now, let us open this program.

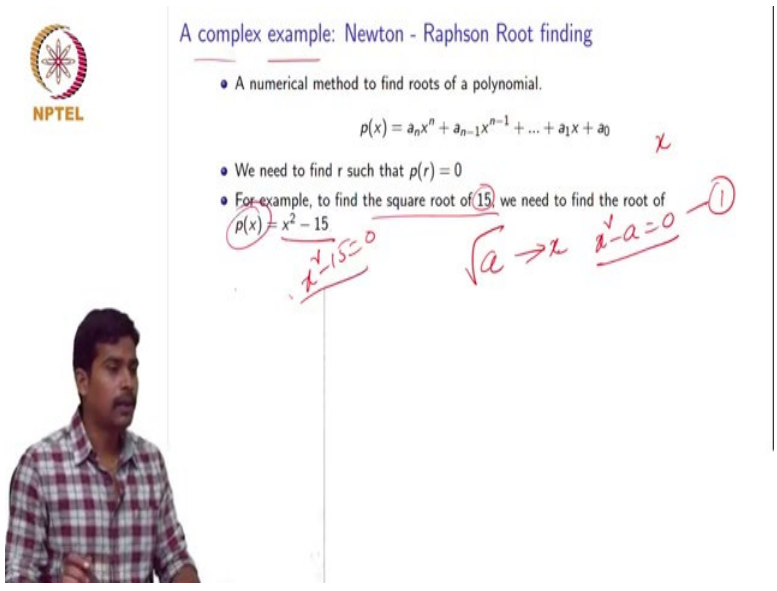
(Refer Slide Time: 57:25)



So, this is the same program. So, here let us give 10 and the factorial is. So, this big number I will give 3 factorials of 3 is 6.

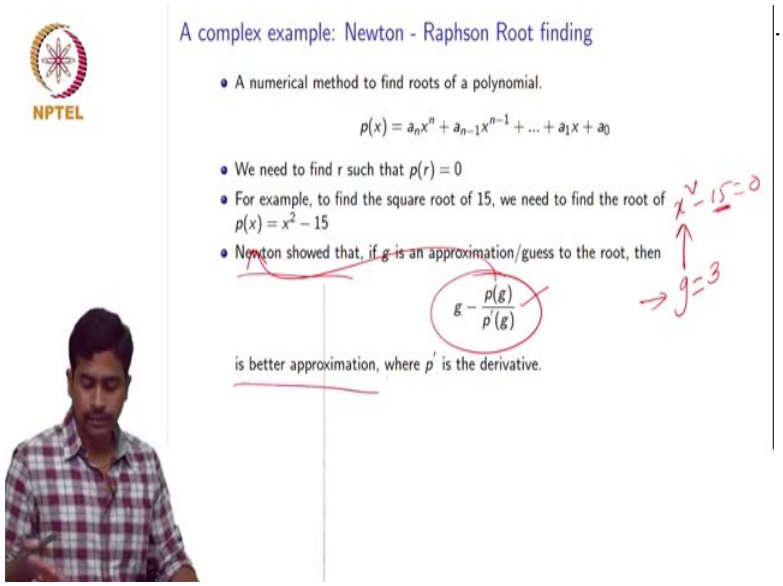
Now, we have seen all the basic building blocks required to write any python program. So, they are types of variables: arithmetic operations, conditional operations, boolean operations and comparison like if you want to see if a number is bigger or smaller which is greater than less than and finally, the iterative statements the while loop and for loop. So, with all these tools you can right now any program that you want um.

(Refer Slide Time: 58:06)



So, I have shown some examples here. So, which is calculating the root of a given number. So, this program utilizes all the basic building blocks that you have learned so far, starting from different variables to the iterative statements. So, to understand the problem, let me explain what we are going to do here ok. So, we all know what Newton Raphson method is right.

(Refer Slide Time: 59:43)



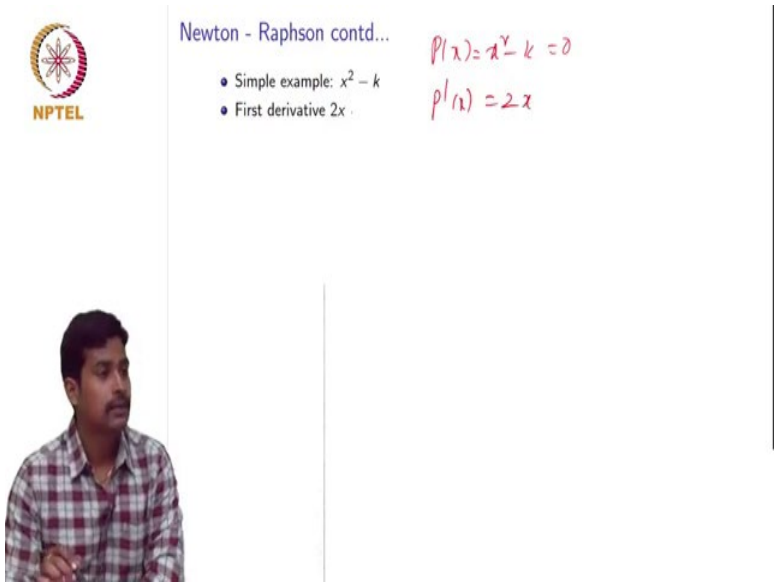
So, according to Newton Raphson method. So, if you have a polynomial p of x of this form and if you want to find a solution for this polynomial solution is r such that p of r equals 0. For example, let us take a polynomial or for example, in this example we wanted to calculate the root of a number x. So, in order to find the root of the number so, we so, let us say sorry. So, for example, if you want to calculate the root of a number a right and let us assume the root is the x ok. So, if the square root of a is x then x square minus a equals to is going to be 0 right. So, we are using this principle to calculate the square root of a number.

So, for example, if you want to calculate the square root of a number 15. So, we have to solve this polynomial $\square^2 - 15 = 0$ ok. So, this is our polynomial and according to Newton Raphsons method. So, for this x square minus 15 equals 0 let us say we have a guess for x ok. So, since 15 is closest to 16. So, the root can be 4 or root cannot be 4, it will be closer to 4 or it will be closer to 3.

So, initially we have to make a guess what could be the solution for this equation we can say 3 is the closest approximation to the actual solution and according to Newton Raphson method if this is your guess this is going to be much better guess than what you already guessed ok.

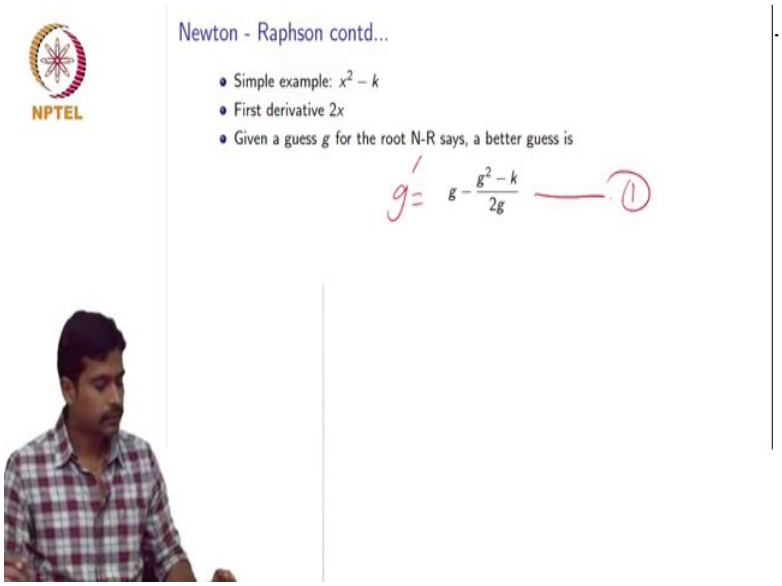
So, the modified guess is given by g minus p of g by substituting this guess in your actual polynomial divided by substituting again this guess in the derivative of a polynomial right. So, this is the basis of Newton Raphson method and let us try to use this approach to calculate a square root of a given number ok.

(Refer Slide Time: 60:47)



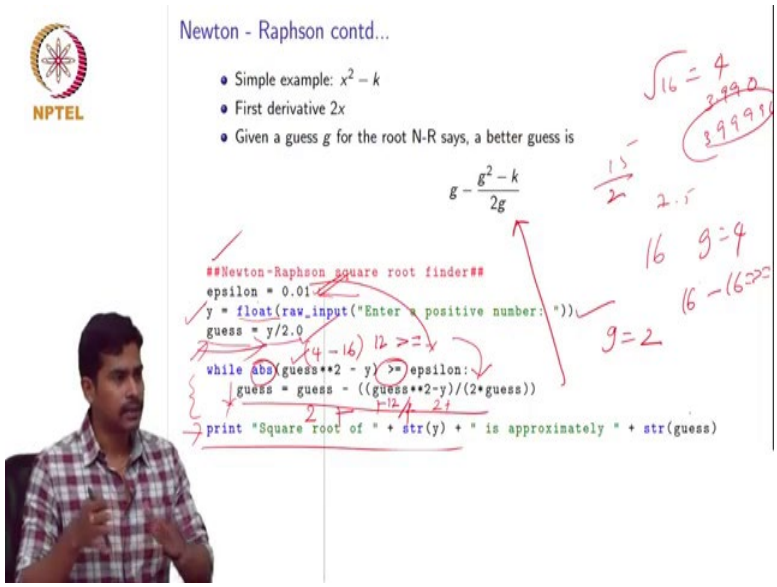
So, as we have seen before, let us say k is your given number and you want to calculate at the square root of this polynomial. So, for this we have to use this equation $□^2 - □ = 0$. This is our $□(□)$ and the derivative of p of x p dash of equals to 2 x right.

(Refer Slide Time: 61:08)



So, now according to Newton Raphson method. So, you if you have a guess. So, you can have a better guess by substituting your new guess or we let us say g dash equals to g minus g square minus k divided by so on, so, this equation.

(Refer Slide Time: 61:25)



So, this is the program that computes our square root and I will explain what each of these lines of this program is doing. So, let us say epsilon is a number. So, what epsilon does is? How close you want to find the root to the actual root ok.

So, if the square root of 16 is 4, how close do you want to find? How accurately do you want to find 4? Whether you want it if you are maybe you or are you satisfied if you get 9 9 0 or are you satisfied if you get 9 9 9 0 ok. So, how accurately we want to find the root is given by this epsilon ok. And this statement as you all know we are taking input numbers from the console.

And we can also find square roots for a float type of number. So, this can be either integer or it can be a float value ok. And so, how do we get a reasonable guess ok. So, what is a good starting point to guess what could be the root? So, a good point to start the guess is to start if you want to find the root of number 15, you can start by 15 by 2 ok. In most of the cases; so, half of the number is a good starting point to calculate the square root.

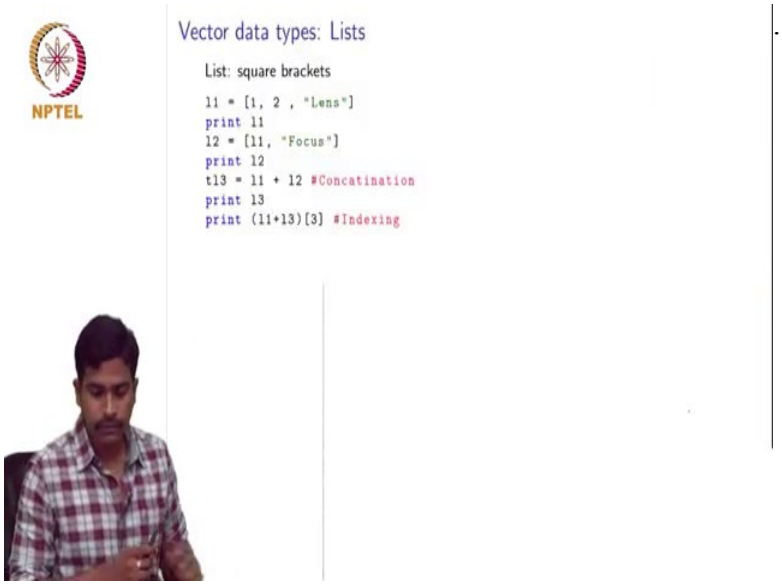
So, in this case we have said the guess is half of the actual given number and here we are using the while loop to calculate the square root of the number ok. So, in this while loop what is happening is. So, for example, guess is our actual root in this example let us assume. So, you

actually guessed the solution right away for example, if you want to find the square root of 16 and you guess it is 4. So, guess square which is 4 square minus actual number 16 is greater than a or equals to your epsilon which is true and it would not execute this block of statements and your result is already given ok.

On the other hand, if you have guessed the square root to be 2 ok. So, now what happens? 2 square 4 minus 16 which is 12 and the here absolute command calculates it converts negative to positive numbers, it takes only positive numbers. So, 12 is greater than equals to your epsilon so, which is true. So, it goes into this while loop and your guess is updated by this equation ok. So, your previous guess was 2 right.

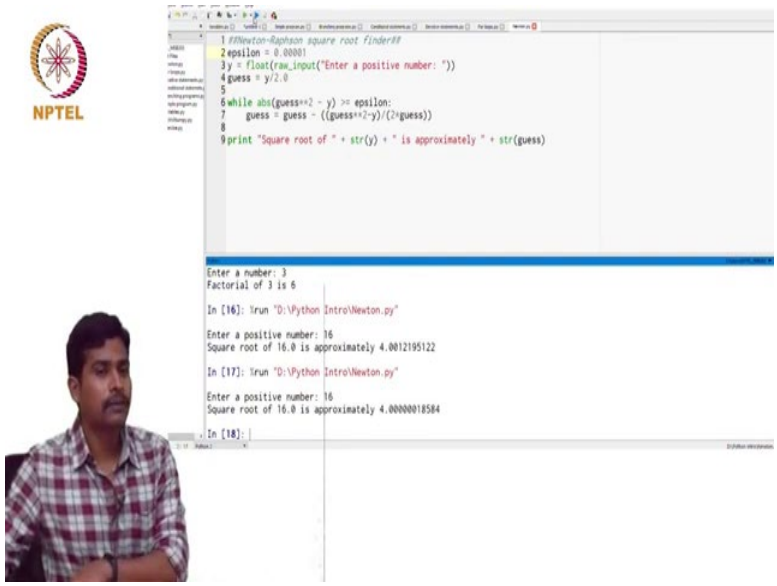
So, now what you are doing is replacing your previous guess to minus guess square which is minus 12/ 4. So, which is going to be some other numbers ok. So, we come back here. We use the new guess to calculate this check and then we keep updating the value of guess and until finally, this condition fails which means we have approached very close to our solution right.

(Refer Slide Time: 64:39)



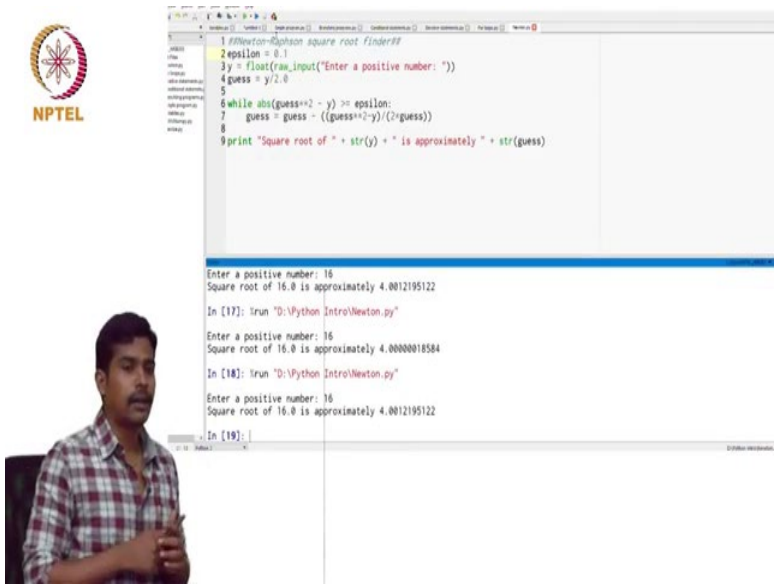
And after that we stopped the iteration and then finally, we print this is the approximate yeah solution yeah ok. Now let us just open the program ok.

(Refer Slide Time: 64:45)



So, this is the same program and let me run it. So, we will give 16, $\sqrt{16}$ is approximately 4.00012. So, if I increase the epsilon I mean if I make it even more closer. So, if I give the same 16, it is going to be even more accurate ok. So, you can change your accuracy just by changing the value of epsilon.

(Refer Slide Time: 65:13)



I hope all of you understood. So, whatever is explained so far, so we have seen whatever is necessary to write a simple program in python. So, in the next class we will see a vector data type and plotting and num pi library of python. So, that is it.