Introduction to Internet of Things
Prof. Sudip Misra
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 26
Introduction to Python Programming – I

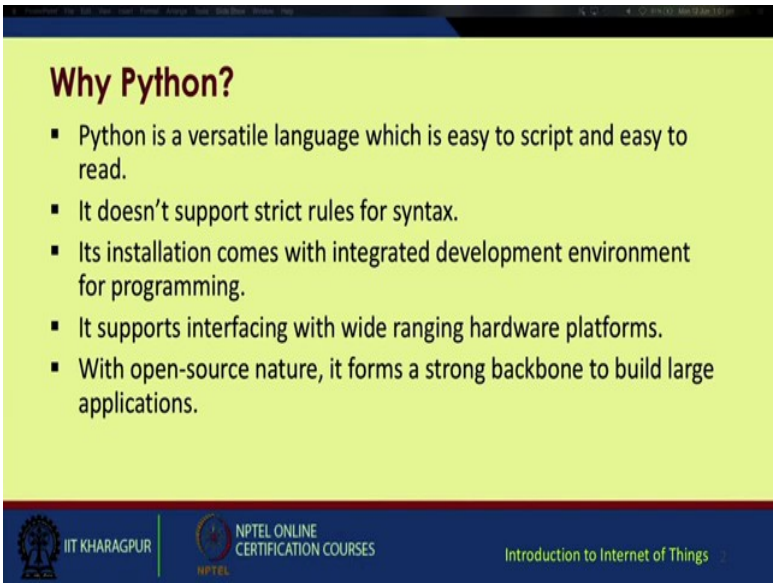
In this lecture we are going to introduce to you the language the python programming language. So, this python programming language lecture is divided into 2 parts. So, in both the parts again I will be assisted by your TA, Mister Anandarup Mukherjee. So, he is going to take you through the basics of the python programming language. So, before we start I just wanted to highlight upon that python is a very popular programming language at present. It is among other applications it is very much useful for embedded systems application development for example, IoT based application development python is very popular, there are several reasons for it one of the reasons is it is a lightweight programming language.

In the sense that from a programmer point of view first of all it is not very difficult to learn this language it is more like a scripting language it is of course, object oriented, but it is a scripting language and scripting like language, and you know it is very easy to learn this programming language. In the same way as matlab for instance is also very easy to learn python is also very easy to learn; and also you know python is supported by different embedded systems development platforms or IoT development platforms such as raspberry pi which you are going to learn in this course, but you know.

So, it supports different types of IoT devices and also you know you do not need to take help of complex libraries etcetera etcetera execution is faster and so, there were so many different advantages because of which python based programming is very important to learn particularly if you are interested in IoT based application development, and that is why you know in this course we are going to teach you a little bit of python programming to get you started and as before you know if you have your environment ready you can along with the lecture you can also code yourself. So, that the learning becomes better. So, that way it will become a enhance on experience for you. So, for all the lectures in module 2 of this course, you can do the same thing you can keep your respective environment ready. So, that you can do the do the you know the programming exercises while we teach you programming and enhance on exercises while we teach you the concepts.

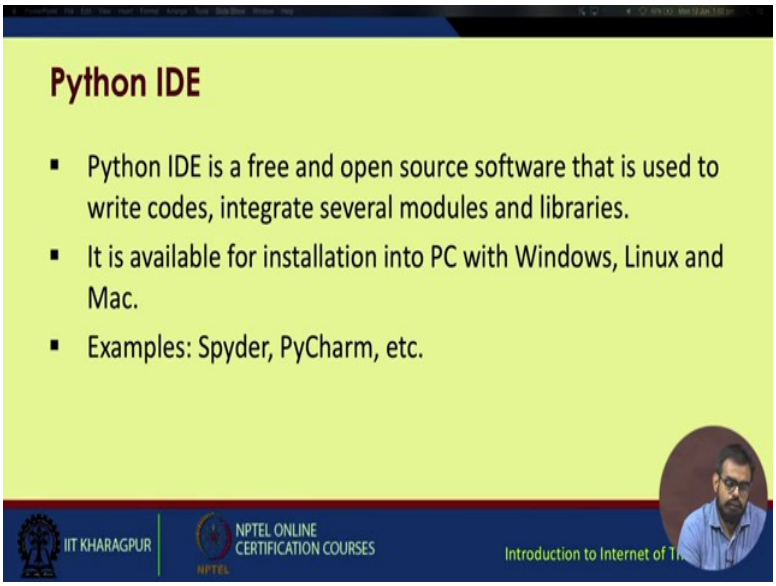
In this lecture I will be covering the basics of python programming. So, we will just talk about how to startup with the basic programming or scripting in python and the basic syntax and all those initial integrities.

(Refer Slide Time: 03:35)



So, first of all why python; now from my personal point of view python I have worked on matlab I have worked on C C++, but I find python to be a very versatile language the scripting is very easy it is very easy to write the code it is very easy to read the code and moreover it does not support strict rules for syntax. So, it is installation comes with an integrated IDE. So, the programming is actually very easy will come to that in the consecutive slides. So, and for IoT you must have seen in various online resources and lectures and courses people prefer python because it supports an interface with a wide range of hardware platforms and moreover since it is a open source platform. So, you have lots of libraries, lots of collaborative work, lots of examples available online on github on various repositories. So, it forms a strong backbone to build large applications.

(Refer Slide Time: 04:51)

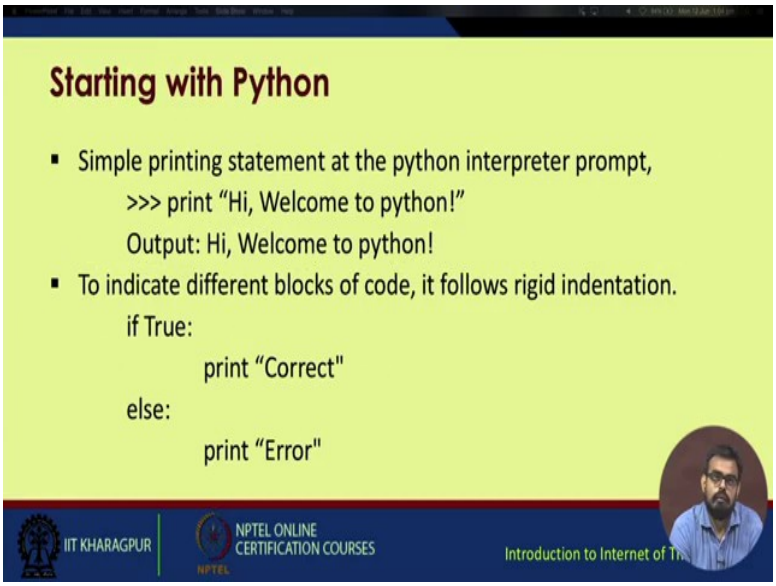


Now the python IDE is like arduino is a free and open source software. So, you can write various codes integrated integrate various modules libraries and so on. It can be easily integrated with windows, Linux and Mac machines. So, some examples of python IDE is are Spyder, PyCharm and so on.

So, my personal preference I as a personal preference I use Spyder to code my program I will show you a visual of Spyder. So, this is basically Spyder it is an editor for python as well as there is an output console over here, and for the python distribution I am using python 2.7 as you can see over here python 2.7, and I am using a python distribution using anacondas. Anaconda is a collection of various libraries and resources. So, I find it quite useful since lots and lots of library is very useful and commonly available as well as some uncommon libraries are also integrated with anaconda. So, that is also my personal preference. So, you can obviously, have normal idle ways systems. So, you have a IDE called idle idle and so on. So, this left hand side is the editor part this right hand side is the console part you can even write on the console. So, problem is you write one line and whenever you press enter it executes start line. So, that sometimes becomes bit problematic whereas, in the editor you write the whole code or the script and then collectively execute it.

So, for larger programs and systems this becomes much easier. Now to start off with python simple example is you write you want to print something some statement.

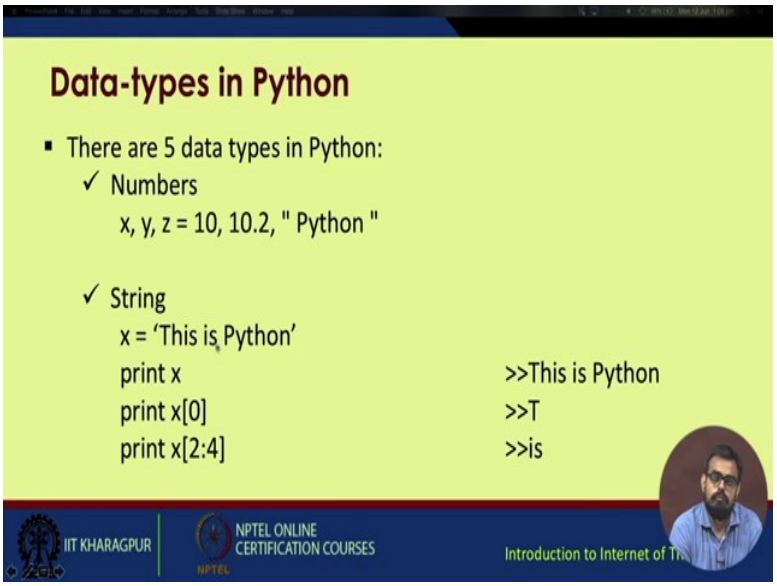
(Refer Slide Time: 07:23)



So, you just write print hi welcome to python or any other statement the output will be "Hi, welcome to python!" So, as you can see the syntax is pretty straight forward you do not need to call any libraries, you do not need any main function, you do not need other functions nothing. You just and remember whenever you encounter maybe online you encounter python codes or anything and you encounter these arrows basically means your code is being run on the console and otherwise it is for the editor you can interchange between the 2 no issues ok.

So, now to indicate different blocks of the code python however, follows a very rigid indentation policy right. So, suppose normal if else statement. So, if true then colon you have an indentation print correct else then again you go back else colon again indent print error. So, this indentation policy has to be followed whenever mainly whenever you enter into a loop. So, after one statement or this colon you have to give one tab space indentation.

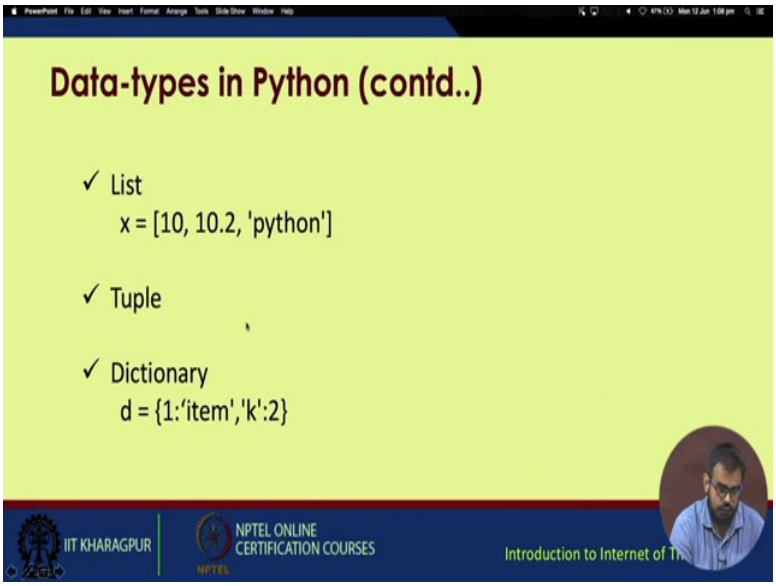
(Refer Slide Time: 09:00)



So, there are five data types in python, numbers you have x y z equal to you assigned 10, 10.2 then you write python. So, x will be assigned as 10, y will be assigned as 10.2 and z will be assigned as python. So, remember this x has been assigned with an integer value y has been assigned with a float floating type value and z has been assigned with a string right; again for the sake of toying around with how you manipulate strings. So, suppose you assign x equal to within quotes this is python; ok one more point single quotes and double quotes those actually do not matter too much you can use them interchangeably. So, over here you see the string python is within double quotes, over here it is within single quotes you can use either. So, string x equal to this is python. So, this string you are assigning to x now you print x.

So, your output will be this is python now you want to access, now this x is an array right. So, you want to access the zeroth element of x. So, you write x within bracket zero. So, this will give you the very first element that is the zeroth index element that is t, now suppose you want to access certain selected elements. So, you write x 2 colon 4; that means, select from index 2 up to index 4 and index 4 will be excluded. So, you have actually 2 and 3 right, so 0, 1, 2, 3. So, is right. So, is has been selected and this is the output of this third statement better still will take a look.

(Refer Slide Time: 11:12)



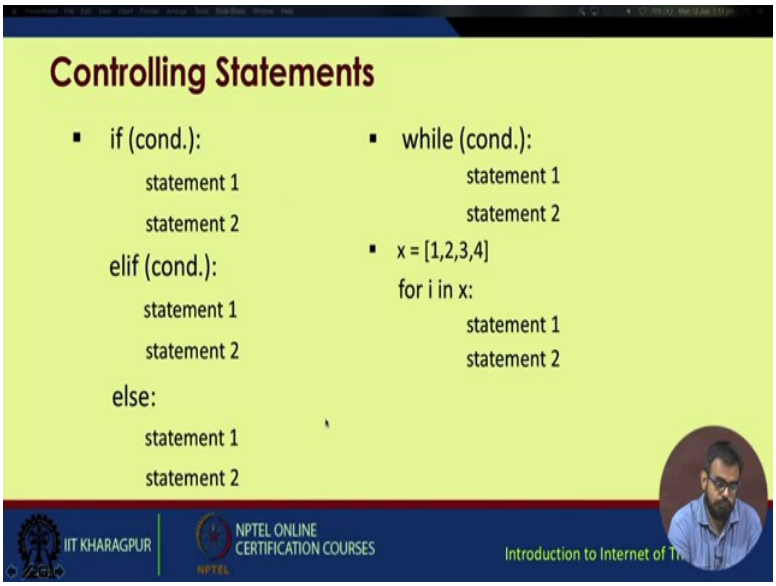
So, we will start off with a fresh console starting a new cornel. So, you can also write over here you write let us say print. So, I can execute this thing from the editor, see print this is a test message this was the line of the script and this in white is the output. So, this is a text message. Otherwise I can also write on the console itself maybe I right let us use single quotes I write hi there. So, you just get the output over here directly. So, as you can recall in C or C++ specially in C you had to call all those library functions using #include studio.h #include conio.h and so on and prior to printing you had to go to the main loop definition and all those things.

So, nothing is required for python you just start writing a script immediately. So, this was the first demo, now let us see let us assign x equal to let us assign string 2 x maybe this is a test right. So, this is a test this string has been assigned to x let us check. I just print x. So, perfect this is a test. Now I want to assign I want to select a specific element from the string let us say 0. So, x 0 it prints t which is the first character right now suppose I want to sell I want to print a specific range let us say 227 right. So, is space is right maybe we can change it to 629 right s space a right. So, it is fairly simple to understand.

Now, various other data types are you have a data type called list. So, list is an order sequence of items right you can see x equal to within the square brackets you have 10 comma 10.2 comma within quotes ‘python’. So, you can assign a integers a float as well as well as a string type to this various elements of the list, then next data type is called the tuple.

So, tuple is an ordered sequence of items which once created cannot be changed or modified. Next is a dictionary; So, dictionary is an unordered collection of key value pairs used to contain a huge amount of data for example, this key is one colon, the value is item then again k, the value is 2 and so on.

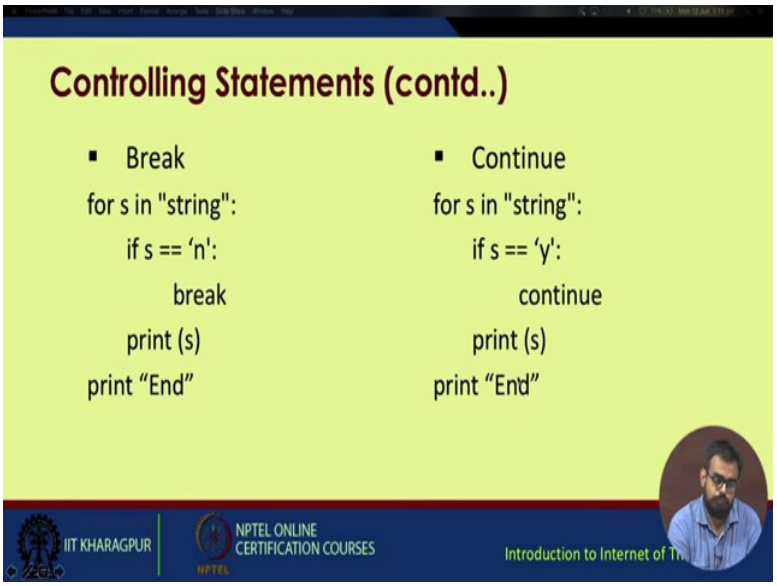
(Refer Slide Time: 16:03)



Now again similar to your Arduino programming or other languages, you have basic control statements. So, startup with if else statement. So, you have if then a condition then statement one, may be statement 2 you can have multiple statements, but remember if then the condition and then colon and since it follows a tabbing policy strict tabbing policy indentation policy. So, you need to have indentation while you enter a loop. So, statement 1 statement 2 have to be indented, then whenever you are using else if the syntax is elif, e l i f again a condition then again colon; then a post indentation you have 2 statements and finally, else which also has 2 statements post indentation

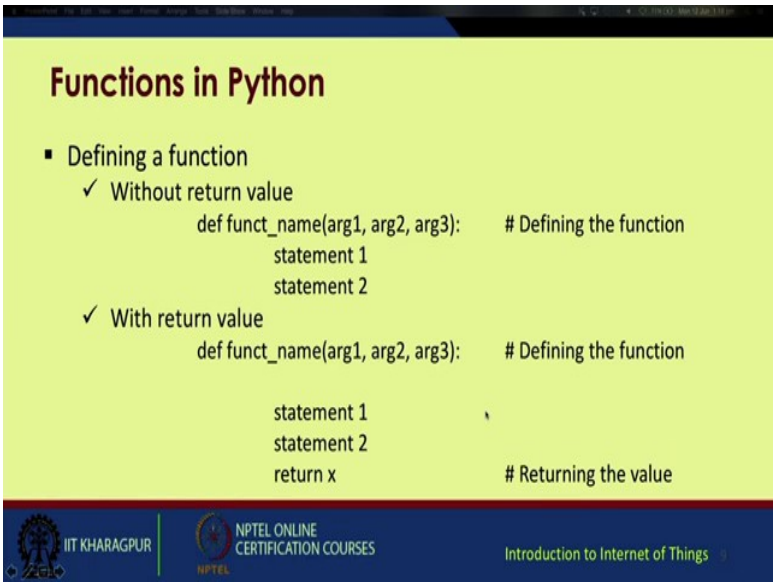
So, this is fairly simple to understand, another loop you consider is the while loop you have while condition colon after indentation you have statement one statement 2 that is it. Now for an simple example suppose x equal to 1 2 3 and 4 this is a list x is a list, now for i in x that is you iterate over the indexes one 0 1 2 3. So, for i in x you gave a statement you gave another statement.

(Refer Slide Time: 17:45)



So, you can modify it in various ways we will check it out later, then you have various other controlling statements like break. So, for s in string string may be anything. So, let us say this is the string then colon then indentation, if s equal to equal to n it compares if s is equal to n then it breaks it prints s and then prints end right and then continue. For s in string if s equal to y it continues then prints s and then end

(Refer Slide Time: 18:28)

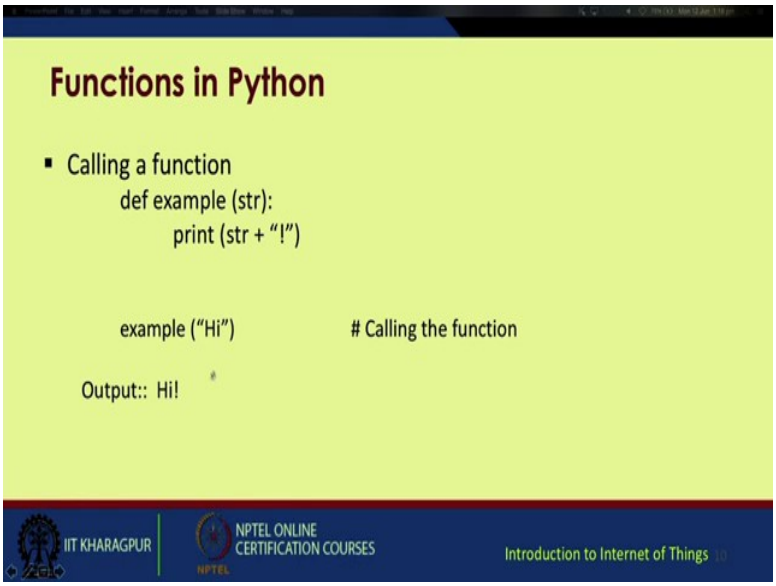


Now defining various functions in python you can for ease of your implementation for ease of your understanding whenever you are writing a very complicated function of very large

program, it is always advisable to modularize your code as in suppose your code includes checking for a prime number, checking for a factorial or returning the factorial value of a function and so on like you have 20 to 30 different such functions.

So, and you need to include this functions multiple times within the complete code. So, it is always advisable you define that function once only, and just call that function again and again. So, this would not only save you lots of confusion, but also will make your code easy to understand. So, this definition of a function it can be either without a return value. So, you write as def a function name of your choice, then various arguments of your choice you can have n number of arguments depending on the function and then a colon then again an indentation statement one statement 2 that is it, right maybe statement 2 is a print statement. So, you give arguments 1 2 and 3 statement one does some operations on these 3 arguments and statement 2 prints the result of the arguments another type is with a return value. So, as you can see these things are the same, in the end there is a return function. So, maybe statement 2 is x equal to some operation and eventually it returns the value of x.

(Refer Slide Time: 20:41)



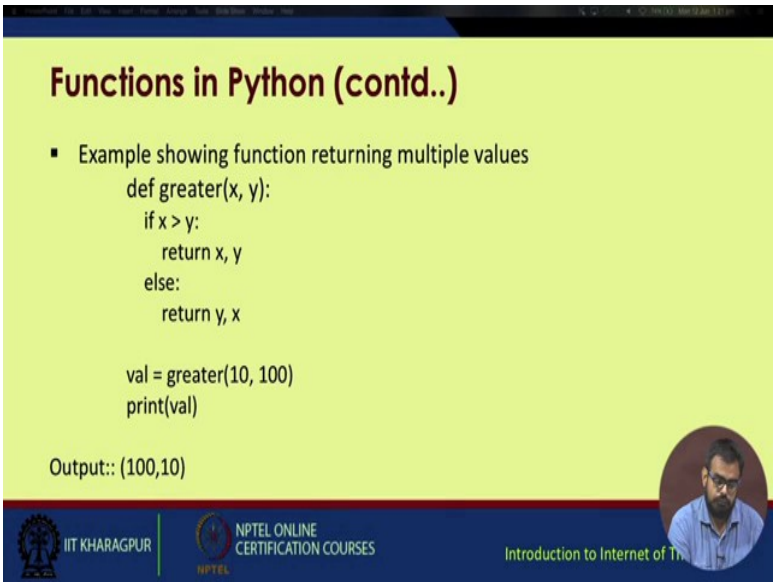
So, the function which calls this the point where this function has been called in the main code it will have x return to it right for example, whenever you are calling a function suppose def example("str"), then print str plus this not character. So, your example outside this function, you just call this function example hi and your output is hi it would be better.

If we do get to a little hands on let us say I define a function as capital IOT, give arguments as xy and z colon io statement as may be a equal to x plus y minus z right and it returns the value of x right. So, my function has been defined now outside this function maybe later on I just call this function IOT, it will require 3 arguments maybe I will right 5 4 3; right and since I will be expecting a return value I assign this function a variable sorry I assign this function to a variable. So, let the variable b b b equal to IoT 5 4 and 3 right let us see what happens. So, first we need to save this code now execute this code all at once ok.

So, this code has been executed, but I forgot to give a print statement let us print b now will execute this again, as you can see your arguments were 5 4 and 3. So, the first 2 will be added and the third will be subtracted from the result. So, your result will be 6 right. So, this is a result. So, this is pretty simple I guess.

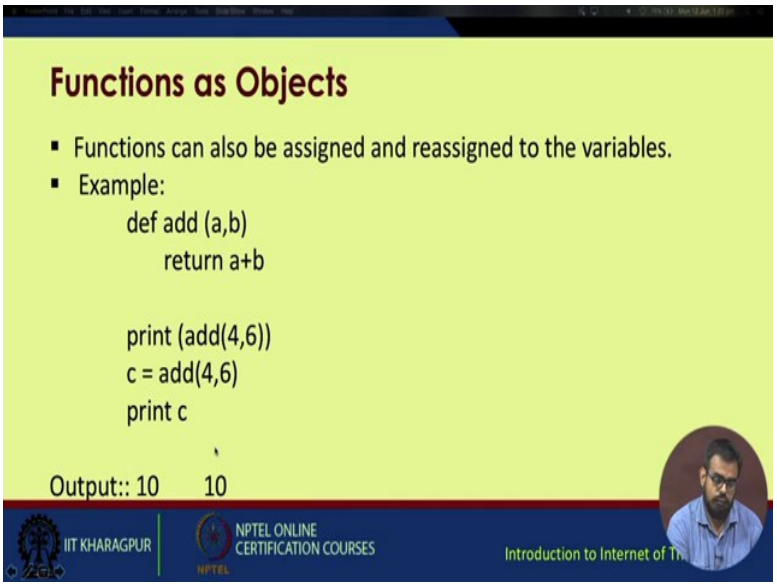
Now since defining a function has been covered. So, similarly you can define various other function suppose you take to arguments and define which one is greater or which one is lesser.

(Refer Slide Time: 24:17)



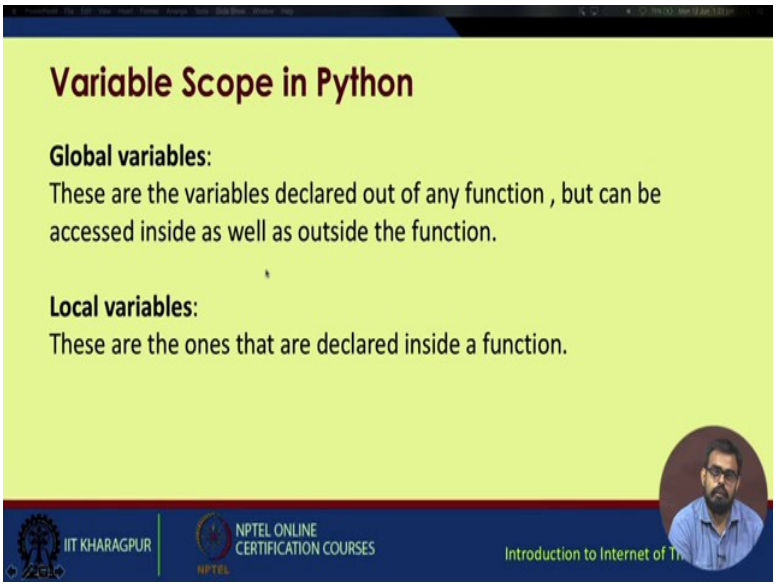
So, you check whether x is greater than y it will return x and y else it will return y and x. So, outside this function definition you assign you assign this function to a variable as greater 2 values as 10 and 100 print val. So, 100 is obviously, greater than 10. So, if this happens it will return this one y and x. So, your output is 100, 10.

(Refer Slide Time: 24:28)



So, it is pretty straight forward now functions as objects. So, whenever you are these using these functions these can be assigned and reassigned to various variables. For example, you write a function for addition you can directly do the operation at the return statement itself, now you print(add(4,6)) and again you assign add(4, 6) to c and then print c, so for both the output will be 10 right.

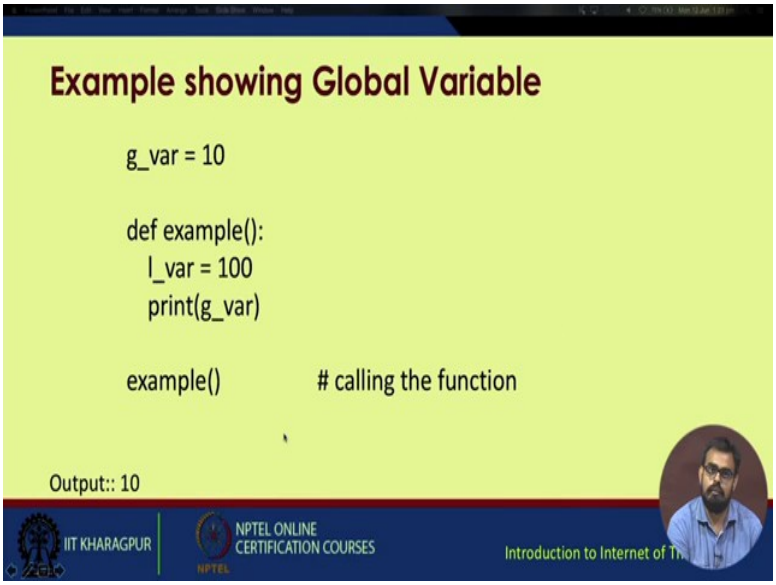
(Refer Slide Time: 25:24)



So, there are 2 types of variable scopes one is the global variable that is accessible those variables are accessible all across the your code, and these variables can be accessed outside

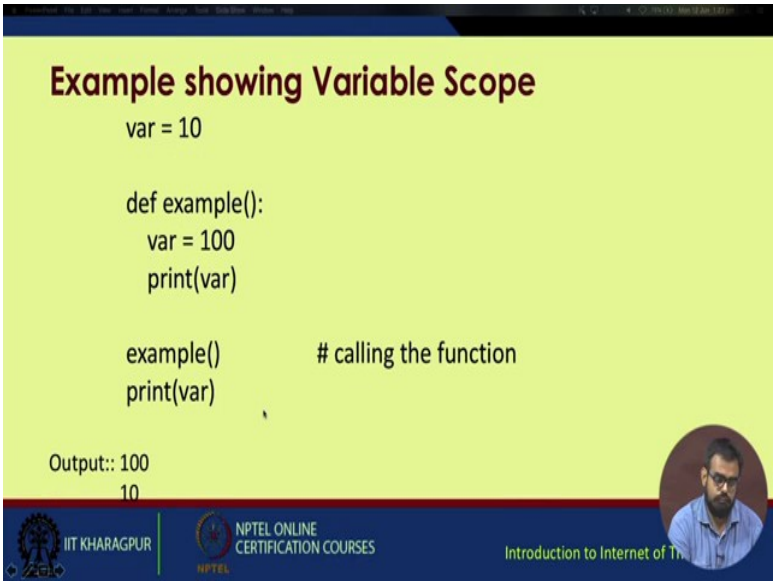
as well as inside a function and local variables these are the once which are only declared inside a function and cannot be accessed from outside.

(Refer Slide Time: 25:46)



So, for example, before a function suppose the definition of a function def example these are some operations you define a variable as g_var = 10. So, this and another variable as i_var = 100. So, within example you can call g_var, but outside example you cannot call i_var.

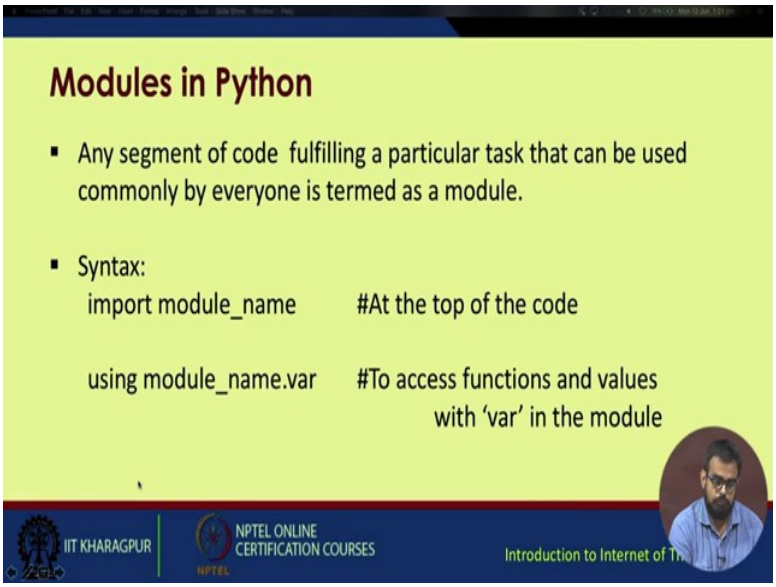
(Refer Slide Time: 26:15)



So, basic variation again you have a variable as 10 with an example variable equal to 100 you print this var and you call this example and again print var. So, what will happen is within

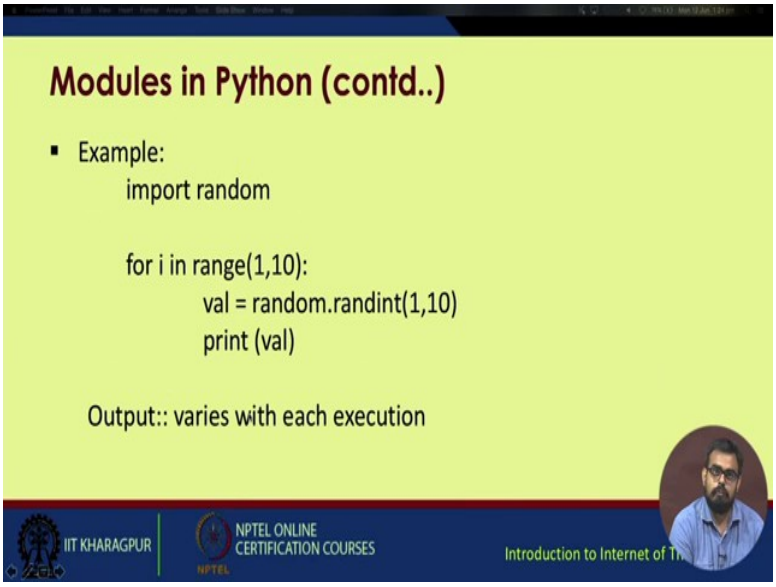
this example it prints var. So, this var will be 100 which is locally assigned right. So, initially it was the global variable is 10, but this overrides that value and reassign hundred to it, but outside this function this is not valid.

(Refer Slide Time: 26:59)



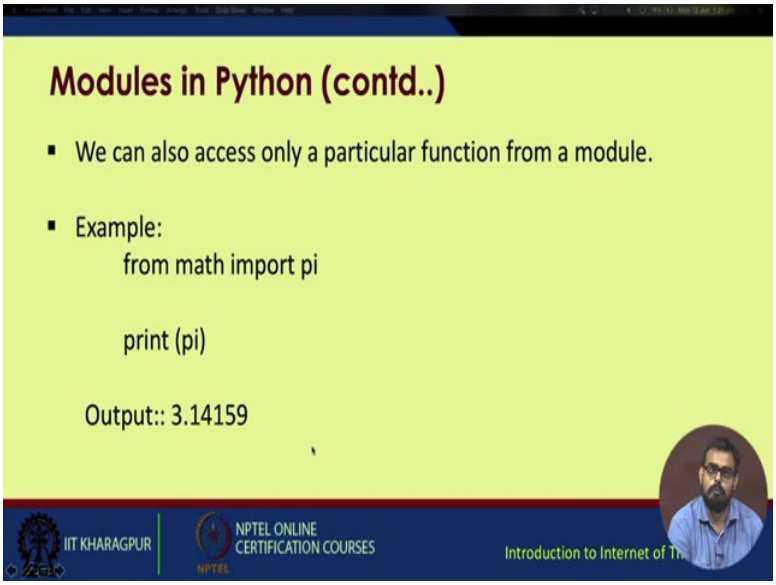
So, for the second print statement it will print the global variable 10. So, you have various modules in python. So, you import the module name now you can also call for the extension for example, you import random.

(Refer Slide Time: 27:11)



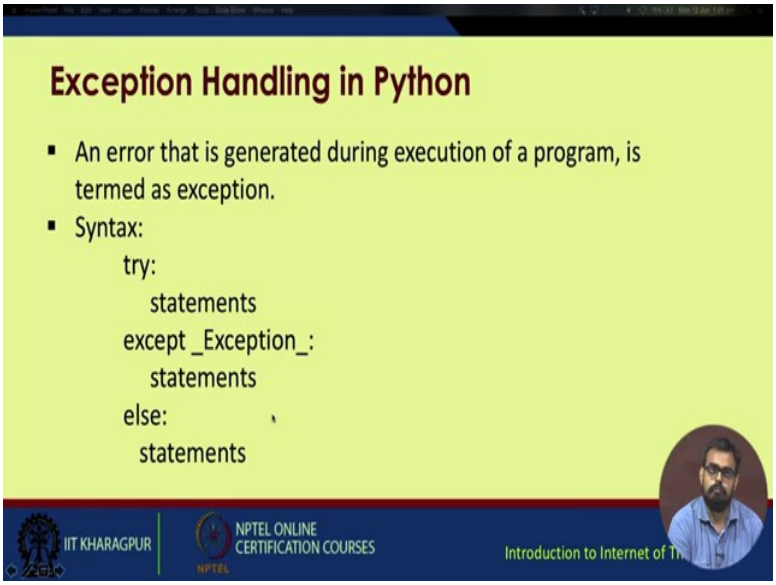
Now, the random has many such functions as random integer 1 to 10 and so on. So, for example, the script for I in range one to ten as when you have a list of 9 numbers 1, 2, 3, 4 up to 9 value is random.randint(1, 10) it will randomly generate numbers between one and 10 and print value. So, since this is random number generator it will the output will vary upon each execution. So, it is better you try this yourself.

(Refer Slide Time: 27:51)



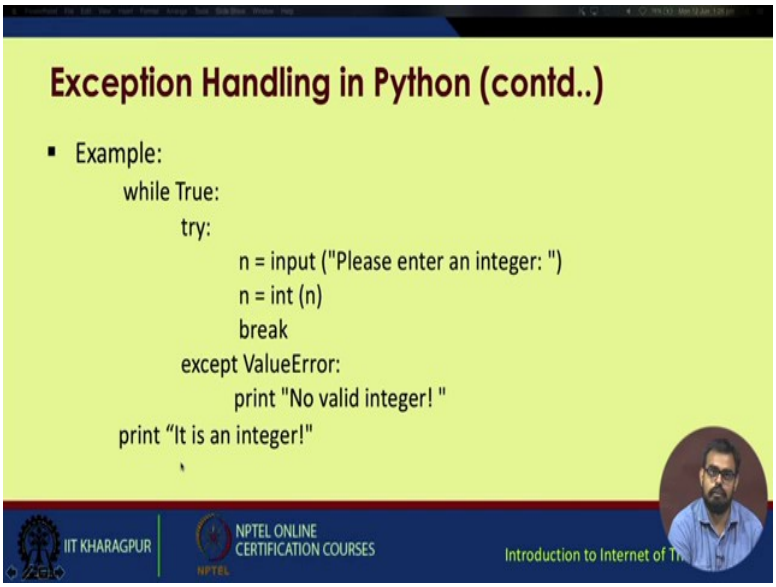
So, you can also try a particular function from within a module like from math import pi and then print pi it will just print the value of pi.

(Refer Slide Time: 28:06)



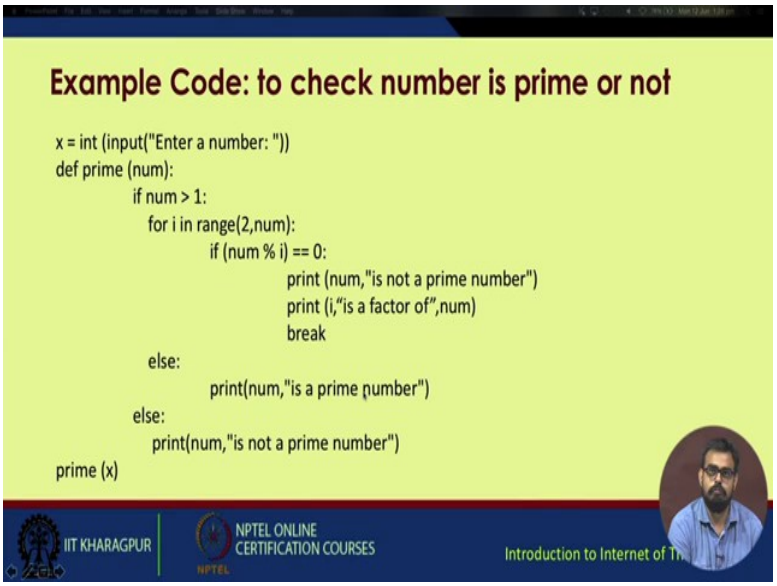
Now there are various exceptional handlers in python, these are mainly used for debugging or in case of errors in various complicated scripts they will give you the exceptions like try then statement except exception statements else statements one example is while true try.

(Refer Slide Time: 28:28)



So, within this type the function will execute this statements these statements will execute if it there is some error. So, it catches that error and accept the value error not a valid integer. So, whatever in number you are inputting it gets stored in n it is converted into integer and you break it and after that this print statement executes if you somehow erroneously input string number to it string or the character to it. So, it would not be converted to integer. So, it will print not a valid integer so, it is better you try this code also.

(Refer Slide Time: 29:12)



Another example code is to check whether a number is prime or not and so on. So, these kind of complication complications can be increased and you can have multiple nested loops you can have multiple functions, function within the function although it is not advisable, but still. So, this was it.

Thank you.