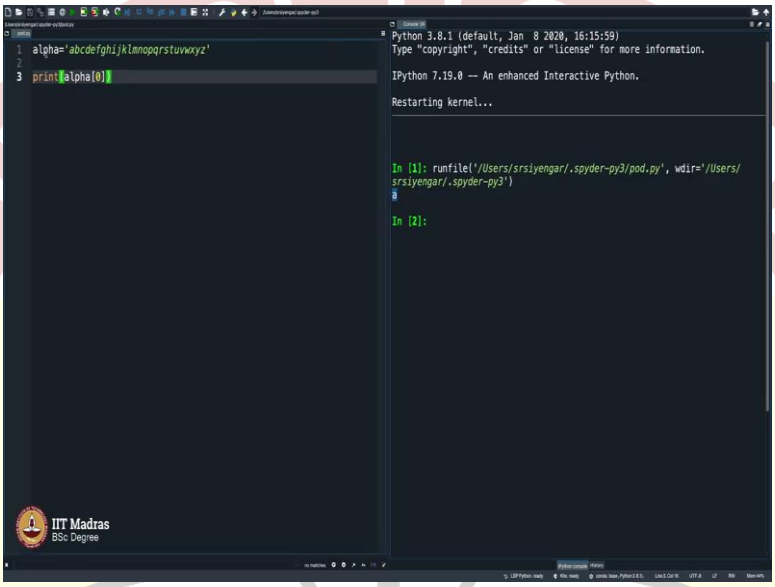
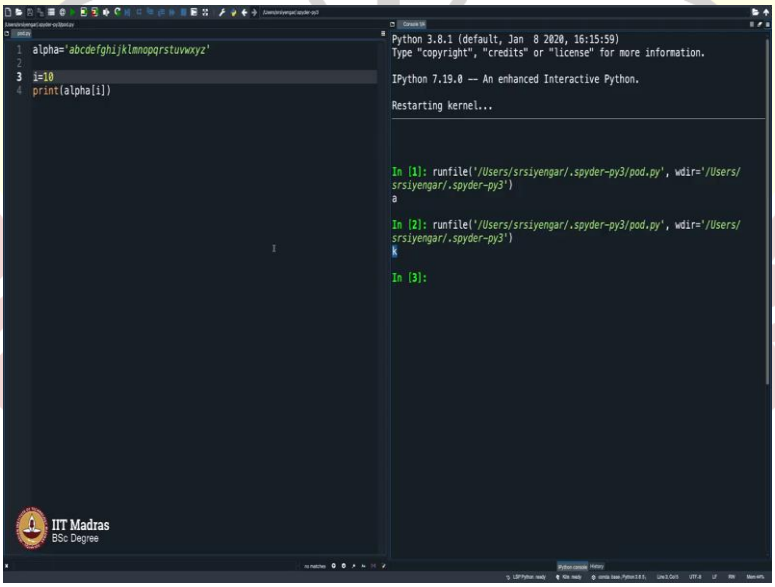
IIT Madras

ONLINE DEGREE

Programming in Python
Professor Sudarshan Iyengar
Department of Computer Science and Engineering
Indian Institute of Technology Ropar
Omkar Joshi
Course Instructor
Indian Institute of Technology Madras Online Degree Program
An Interesting Cipher: More on Strings

(Refer Slide Time: 0:15)

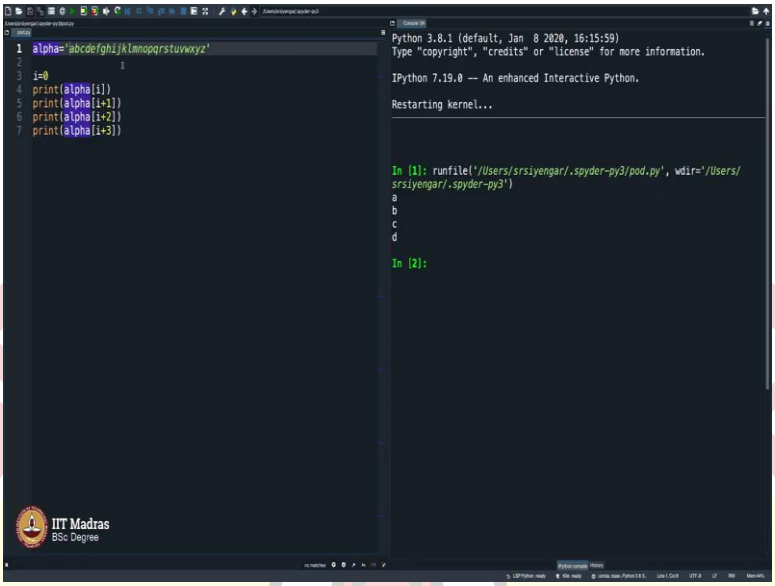


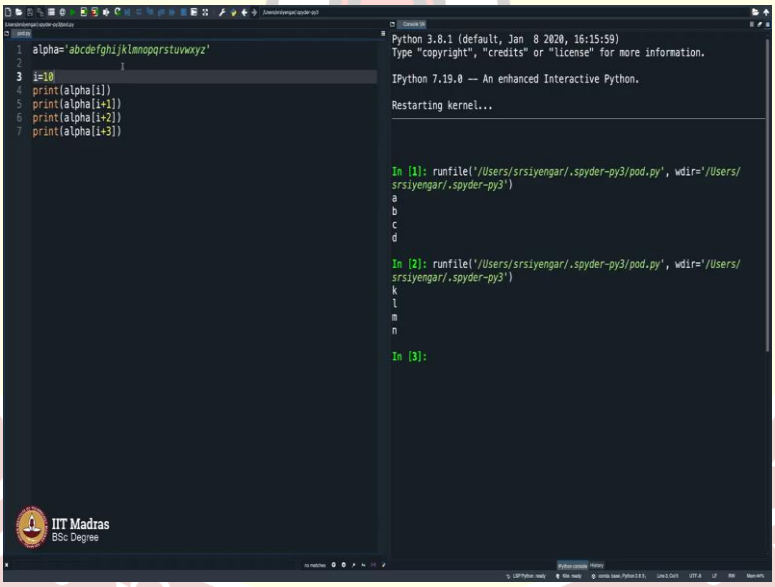


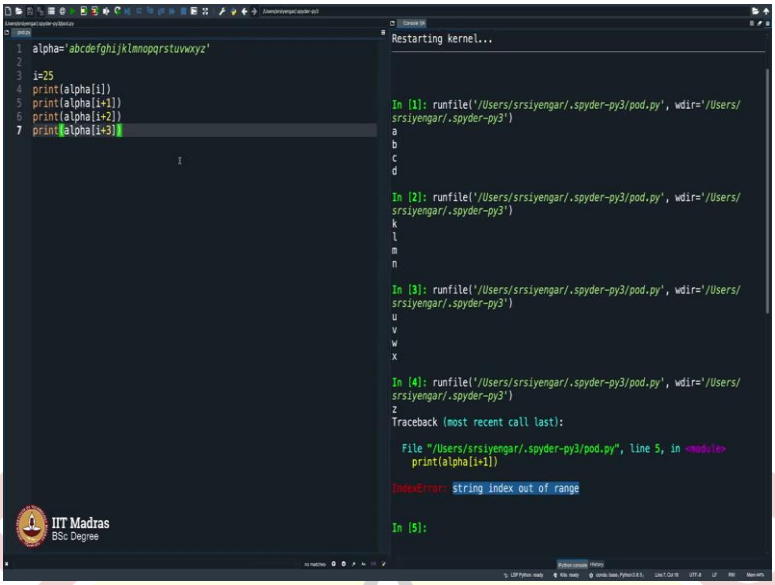
We allow them to play around with strings and we are going to learn something nice. So let us start with our alphabet set and we call them alpha and alpha will be all the alphabets; abcdefghijklmnopqrstuvwxy and z. So after this when I say print alpha of 10, what will it do? Let us start with 0 maybe, it will print the first letter in the alphabet set here, alpha of 0 will

be the first letter here; alpha of let us say i and I will say i equals 10 it will display the 11th alphabet k. 10 is, if 0 is a, 1 is b, 10 will be your k. Clear. So far so good.

(Refer Slide Time: 1:19)



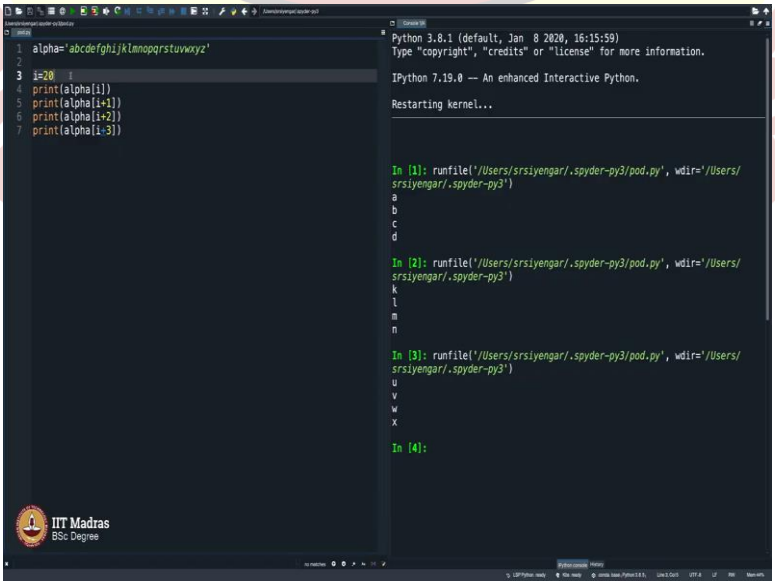




Now just in case I go on like this and I display the 10th alphabet rather ith alphabet and then display the alphabet after i, which is i plus 1, as simple as that. And then I say alpha i plus 2, then print alpha i plus 3, so what does this do? Let us see, if I start with i equals 0, it gives me a b c d, it is very obvious what this is doing, I do not have to explain it to you. By looking at the code you can tell me what exactly is happening.

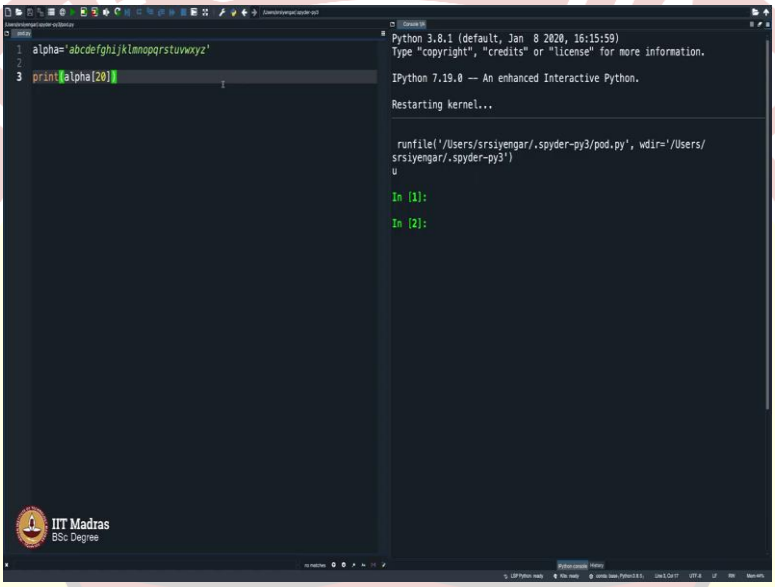
I declare the variable alpha which is a string into z, all letters in alphabetical order in lower case as you can see all 26 letters are here abcdefghijklmnopqrstuvwxy and z. And I say i equals to 0 and I display alpha of i, alpha of i plus 1, alpha of i plus 2, alpha of i plus 3. If I were to do i equal 10, it will start from k as you can see and then go to l m n.

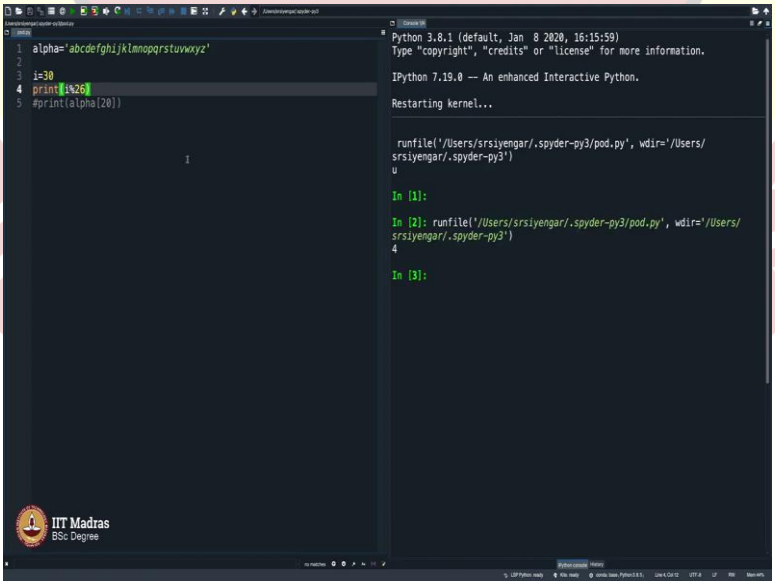
(Refer Slide Time: 2:31)

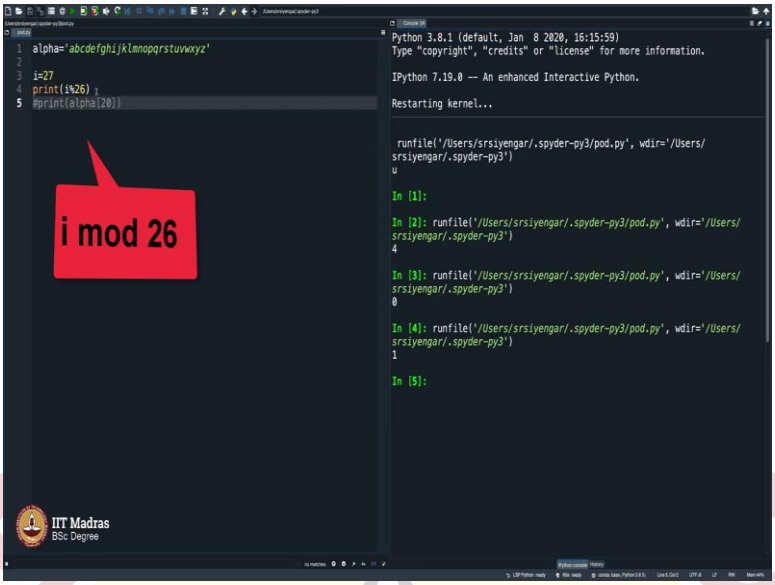


If I go to 20, it starts from u v w and x. The best part now is if I say 25, it will start with z of course, alpha of 25 is z, well, you do not guessed it right, alpha of 26 will throw an error. Precisely what we see here; string index out of range. Your alpha is only 26 letters long, which means you can go up to alpha of 0 to alpha of 25, you cannot go beyond that, while i equals to 25, i plus 25 becomes this becomes 26, alpha of 26 becomes outside the range, precisely what my, what python is telling me. String index is out of range, well no problem, there is a way to sort this out. So what do we do?

(Refer Slide Time: 3:32)

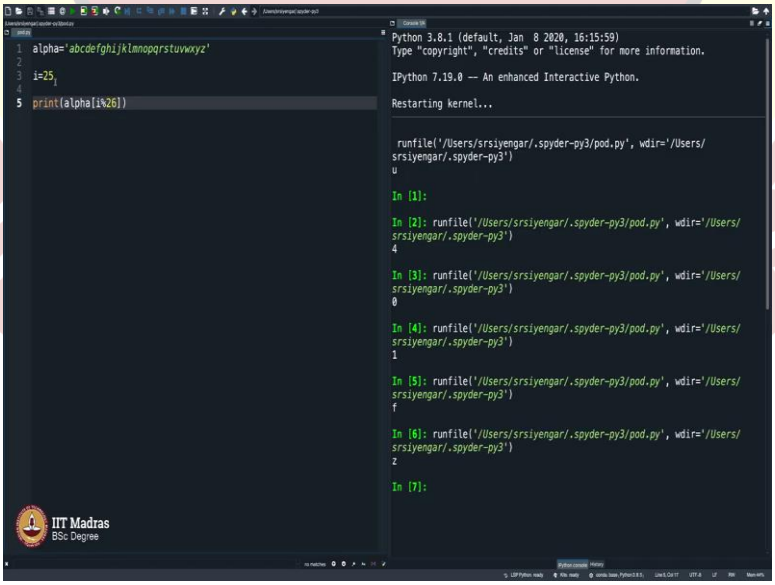


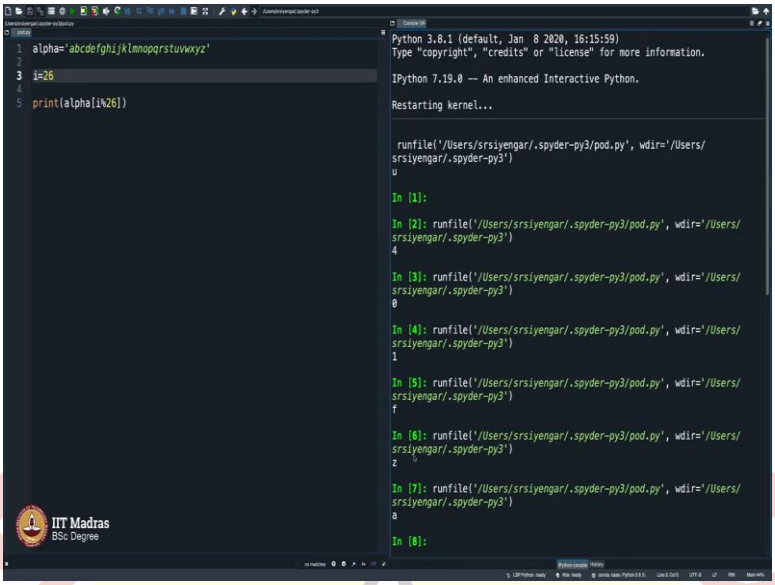




Observe. Let me delete this for the time being. And then say alpha of 20, I will print this, just this nothing else, just this. It shows (()) (3:47). Let me clear the screen. It simply shows the 21st alphabet letter which is u, as you can see it is u here. But then, if I say i equals 30 and then if I just print, let me just comment this, if I just print i modulo 26, this is modulo operator; it gives you the remainder when you divide the number by 26. So 30 by 26, the remainder happens to be 4. Observe, if i equals 26, i mod 26 will be 0. If i equals 27, i mod 27 will be 1 as it is displayed.

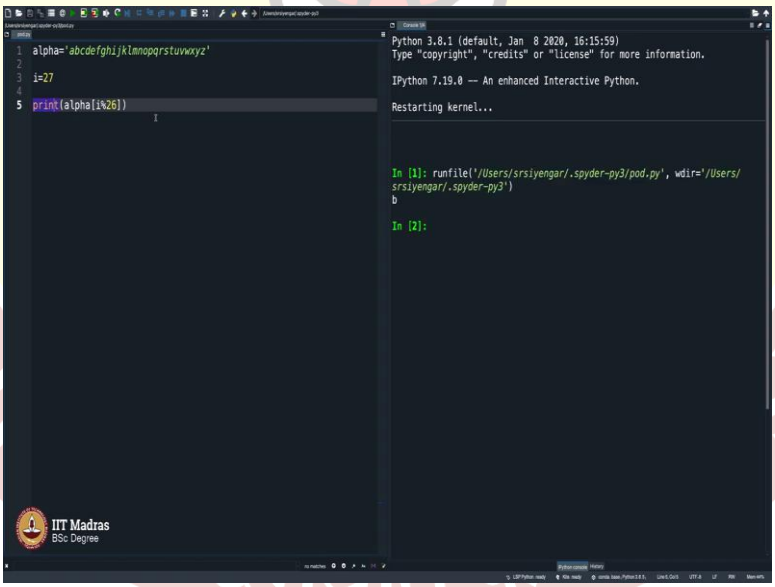
(Refer Slide Time: 4:43)

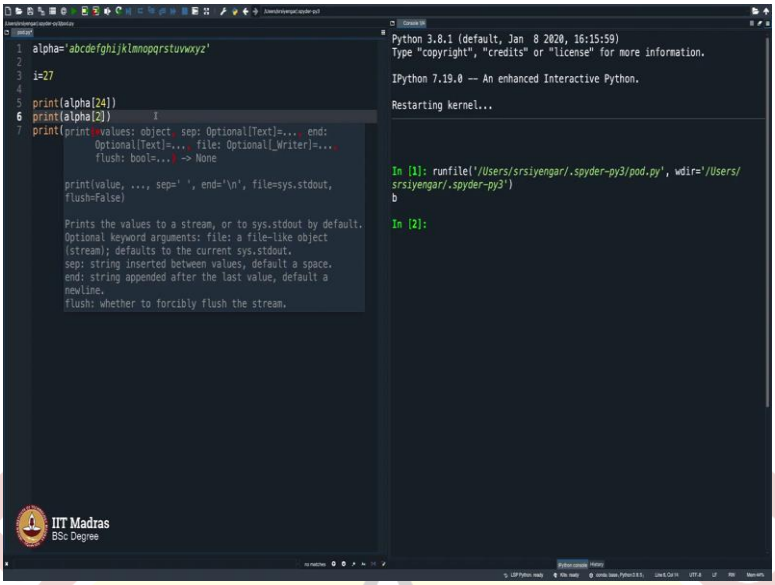


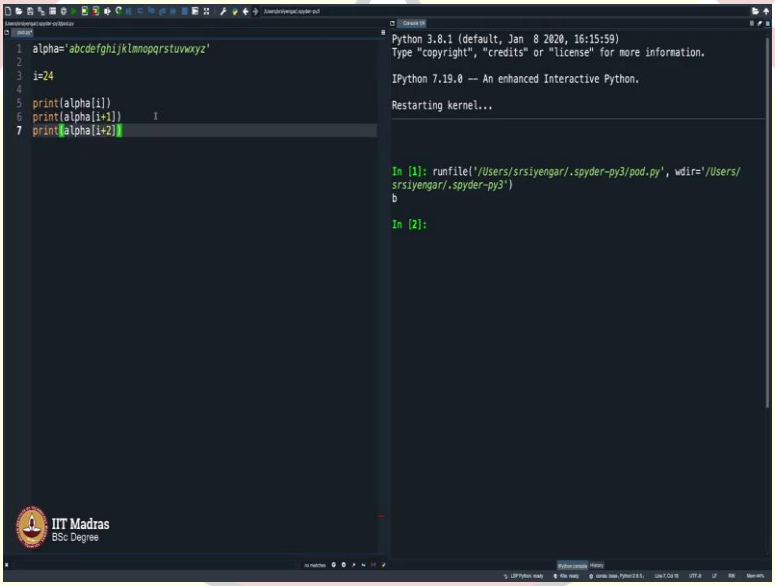


Now you know where I am getting, what I will now do is I will print alpha of i modulo 20. And then when I say i equals 25, it displays the 25th letter, I am sorry it should be 26 here, i mod 26, the 25th i equals 25, means the last alphabet z here but the moment I say i equals 26, it goes to 0, because 26 modulo 26 is 0 and alpha of 0 will be a. Correct?

(Refer Slide Time: 5:20)



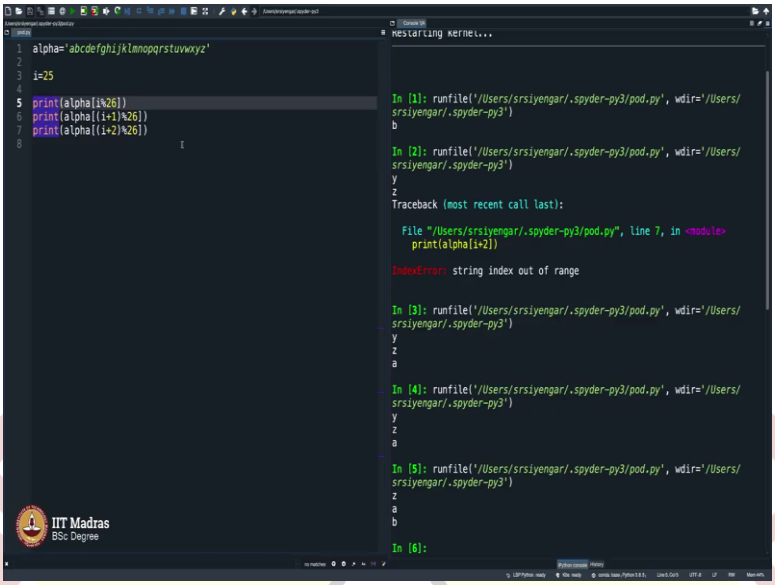




Let us see what happens when I say i equals 27, 27 modulo 26 is 1, so alpha of 1 should be b, so on and so forth. So do you see what is happening? If I were to type alpha of 24, what is that 25th alphabet, which is y, correct. Then I say alpha of 25, good way to do this is call this 24 and keep calling this i, I mean they are all one and the same, I am trying to do it in a way, that it is easy on the mind.

It is good to use variables as much as possible. So alpha of i, alpha of 24, alpha of 25, alpha of 26. These should be y z and a, correct? Think about it. Yeah I get y z and there is an error. Why the error? Because I did not do the modulo 26.

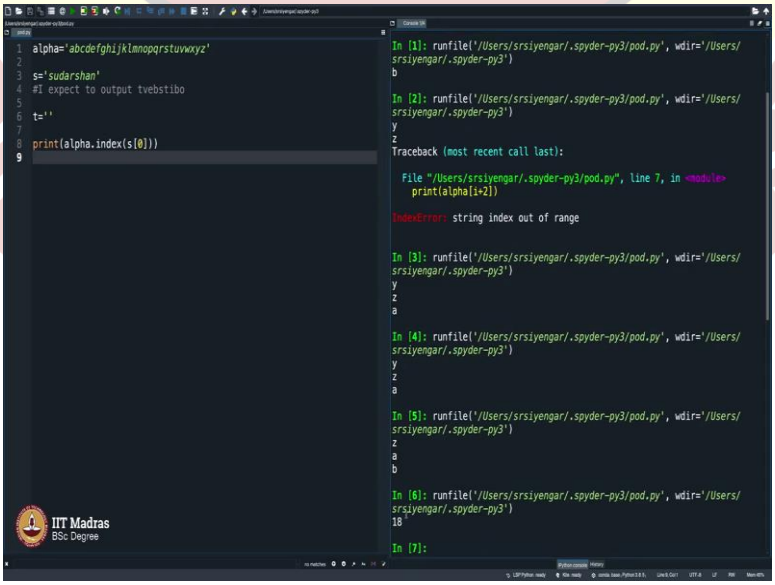
(Refer Slide Time: 6:18)



When I do modulo 26, it will not go beyond 25. If it is beyond 25, it becomes 0, 26 is 0, 27 is 1, 28 is 2 and so on, i modulo 26. I hope this is clear to you all. Pretty simple, I am in fact going very slowly assuming that for some people modulo operator maybe new. So y, z and a, if I well to make i equals 25, you see 25 mod 26 is 25, it is z alpha of 25 and a, b.

You see it is getting rotated, it goes from abcdefghijklmnopqrstuvwxyz abcdefgh, after z it again starts with a. So this has a very interesting application which I will tell you in a minute now. Yeah, so far so good.

(Refer Slide Time: 7:17)



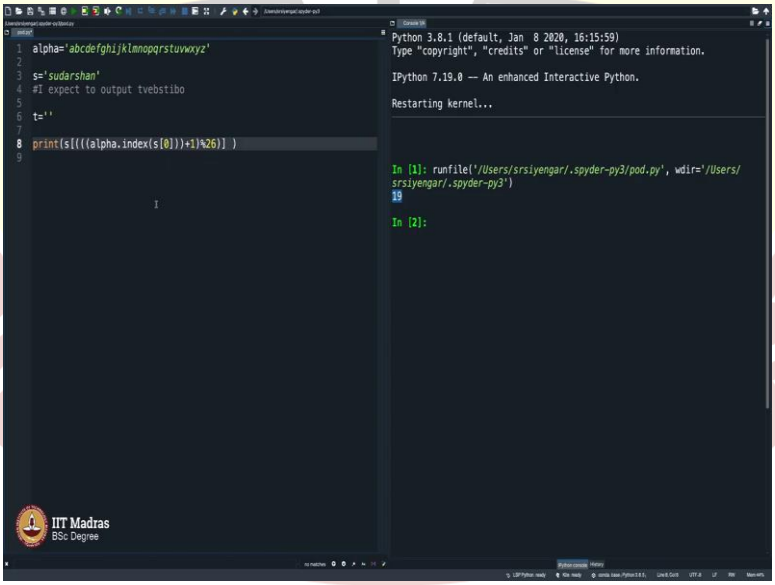
Now what I will do is I will take my name, you can take your name. So I will say s equals sudarshan, what I will expect to see as an output is I want to print tvebstibo, you know what

this is? This is simply, let me comment this; I expect to output tvebstibo, what is that? This is exactly one letter shifting of this string; s u d a r s h a n, if you shift that by one letter, you will get this and whatever you want to type here, India, Chennai, your name, your friend's name and any English word, it should shift it by one letter.

How do I do that? It is very simple. Let us try it out, as you know I am going to use this string function called index which we have discussed recently. So I believe you know of it, if not please go ahead and revise your previous videos and understand what index does. It is basically self-explanatory, you need not go through it, but in case you are in doubt, you can go through it. So let us start, I would say t will be my new string where I will be creating one letter shift of this string that I have here. How do I go about it?

I first say, I will check s of 0, what is it and what is its index in alpha. So s of 0, let us see what this does, patiently you must observe s of 0 is s, the first letter s. And what is its index, what is s's index in this alpha, I mean which alphabet is s? If a is 0, s is what? Let us check, 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18; this is 18. Let us check that. Print alpha of index of s of 0 has to be 18. Yes. It says 18 here as you can see. Good.

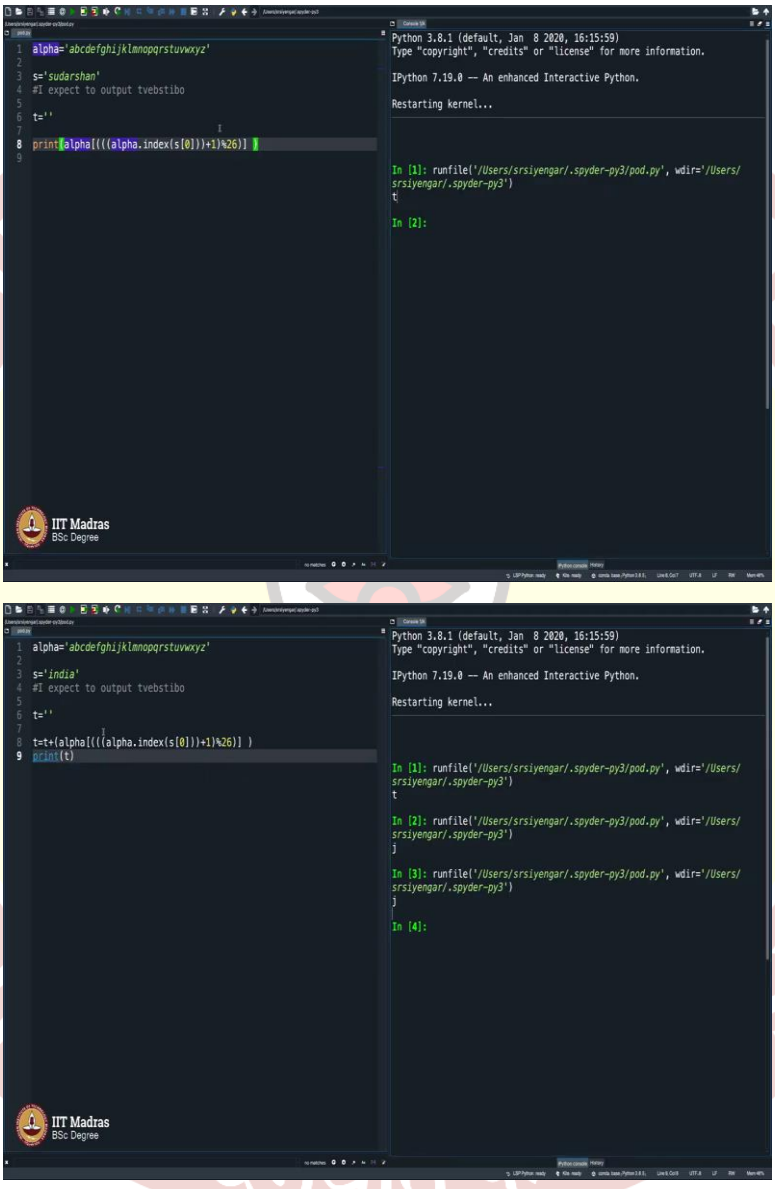
(Refer Slide Time: 9:50)



So I am just clearing the screen. So fine alpha of index of s of 0 is 18 and what I want to do now is I do not want s of 0, I want 1 shift of it. But then when I shift by 1, I explained a problem that you will encounter, I am just trying to close the bracket that I have opened; I hope I am right, may be one bracket needs to be closed. So now this will print the index of s and plus 1 to it.

And what is the alphabet, I basically want the alphabet. So for that I may have to say this is a number, so s of so much and I want to print this, maybe I have used one extra parenthesis, good, more the parentheses, less the confusion. So this should print; index is out of range, what is happening?

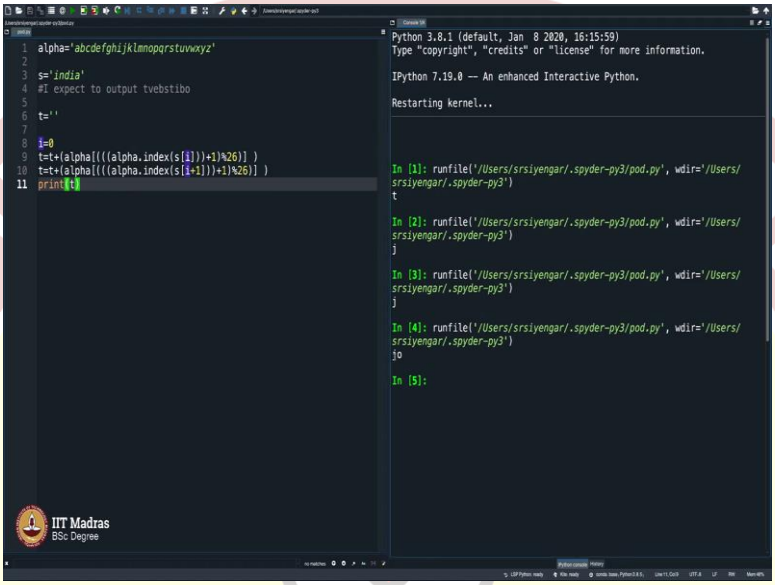
(Refer Slide Time: 11:01)



So there is a small mistake in the programming, I should not display s, I should say alpha. Why? That is because I take the first letter of s, very simple, just look at this, convert this to plain English. I look at the first letter of s, look at its index in the alphabet. What is its number? Whatever is s of 0, which is s, what is its number in the alphabet set? I add 1 to it and then display that, by display that I mean what is the, which alphabet is that when you add 1, when you shift by 1 unit, what alphabet do you get?

Whatever it is, you will see that here. So it is t, I executed this, it is t. If instead of sudarshan I want to say, let us say India, i becomes j here. That is the first letter and what I want to do is I want to, I do not want to print this, I want to append this to t. So what I will do is I will say t equals t plus, remember, plus of a string appends that to it. If we say print t, it will show me j. Correct? So far so good.

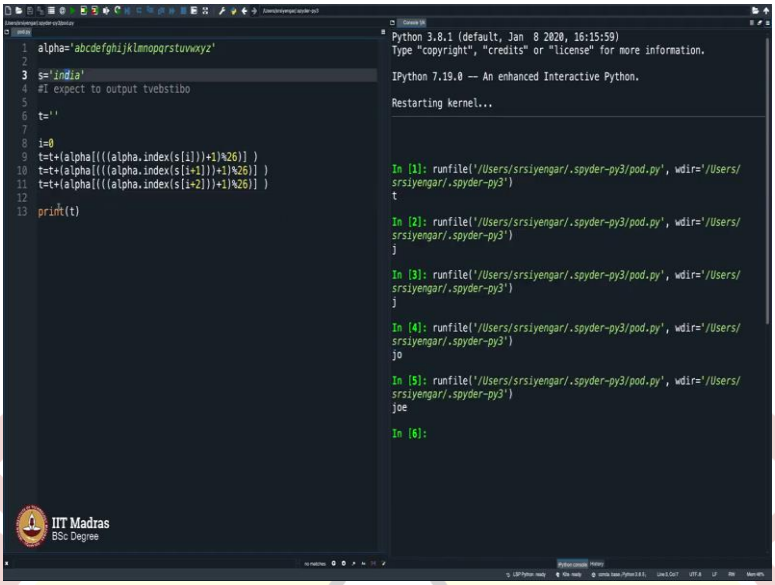
(Refer Slide Time: 12:18)



So next let us say I will try to take the next letter which is s of 2, so for that a good idea would be to say i equal 0 here and then make this my i. And then now do this and append the next letter in the string s. Add 1 to it and then take modulo 26 and then append that to t. Things are getting a little complicated, so let me try to explain this slowly to you, the string t will be equal to the string t append that with the next letter of the i plus 1th letter which is n.

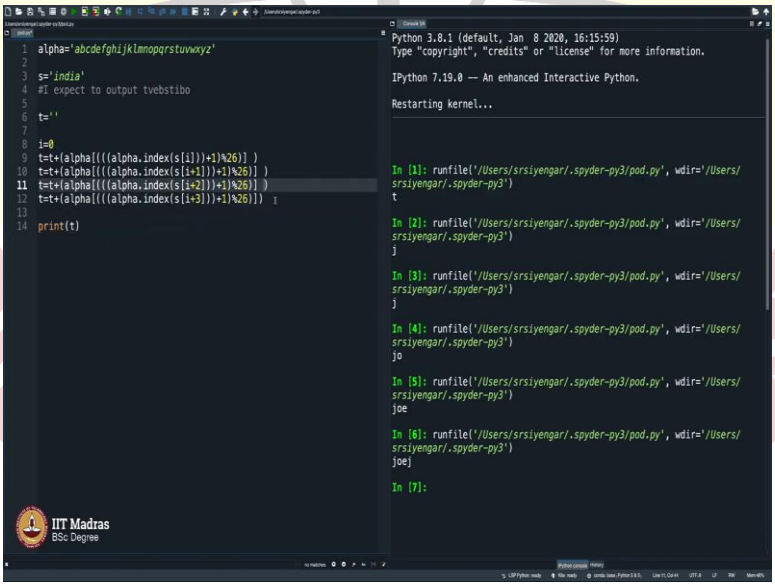
So in fact before going any further, it is probably a wise idea to print and see what is t. So what do you expect t to be, i becomes j and gets appended to t, n should become o and should get appended to t. Let us see if that is happening. Perfect. i is becoming j and n is becoming o, so far so good.

(Refer Slide Time: 13:29)



Now let us see what is next that needs to be done, I will copy paste this to make it easy. So I say t is equals to the existing t, concatenate that with the, take the index of s of i plus 2 it should be the next letter. And shifted by 1, take modulo 26 because in case it goes beyond z, then it should rotate as I explained. And then print it; j o e, as expected, d becomes e.

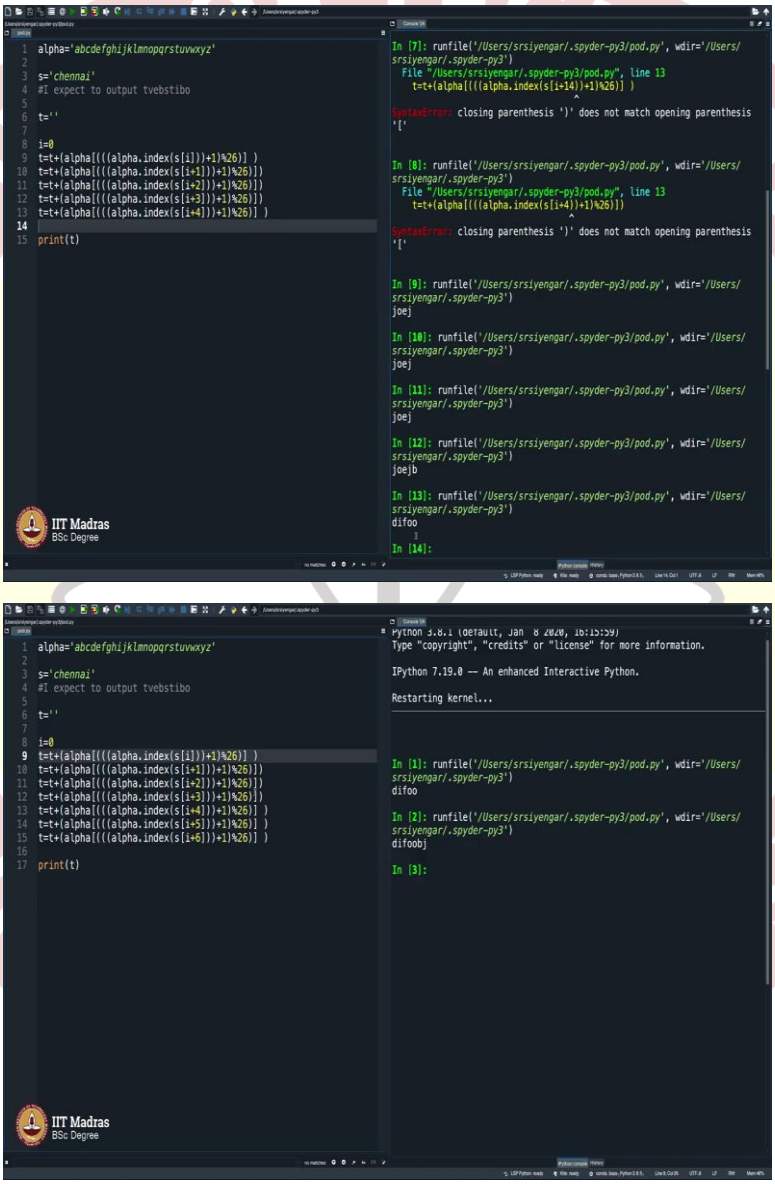
(Refer Slide Time: 14:06)



Now let me do it for i, I will go slowly once again, I am sorry I am doing repetitive again in the interest of people who are seeing this for the first time. Please note that programming can get complicated but if you make it easy on your mind by going slowly, staying patient, using a pen and paper and trying to write a diagram of what is happening, things will be very easy for you.

So t will be whatever t was, concatenate with the alphabet which is the shifting of s of the 0 1 2 3, the 4th letter. Then close the bracket, close the bracket, and then add 1 to it, because it is shifting and do modulo 26 to it just so that you do not mess up when it is the last letter. So in d indi becomes so much, very good and this is not looking neat, let me remove the spaces here, so it will look uniform, perfect. So as you can see, I get j o e j b, India becomes j o e j b, perfect.

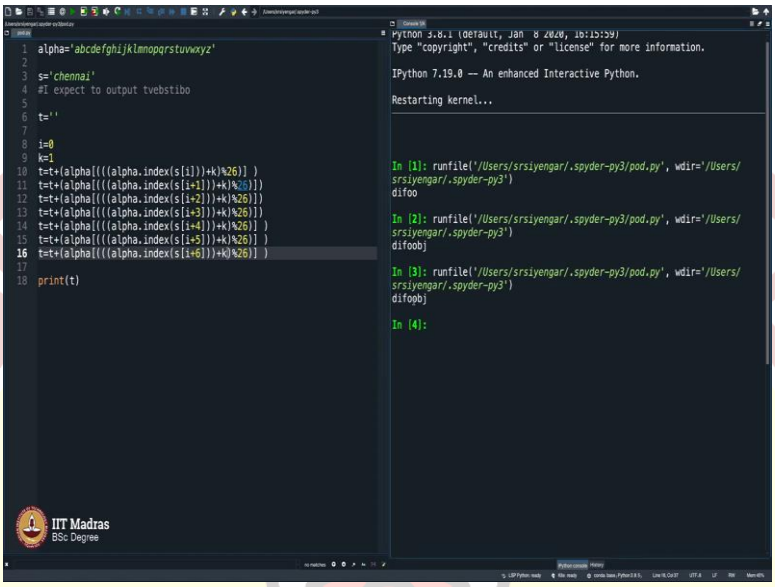
(Refer Slide Time: 15:32)



So as you can observe, I go up to the last letter here, if it were to be one letter more, let us say I take Chennai instead of India which is 1 2 3 4 5 6 7, seven letters. And I have to type this seven letters, otherwise what will happen, let me just execute this and see what happens. Chennai is getting shifted here d i f o o. Let me clear this screen and then execute it.

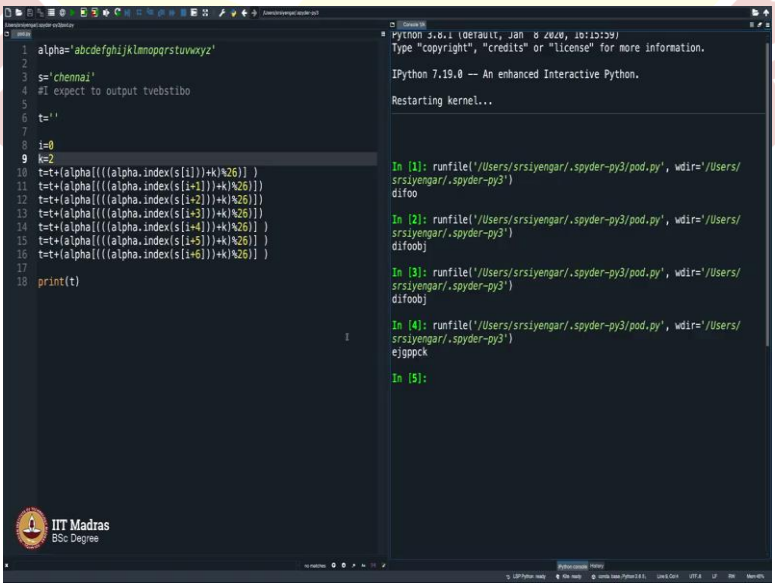
Chennai is becoming, c is becoming d, execute this, c is becoming d; h is becoming i; e is becoming f; n is becoming o; n is again becoming o; a i (()) (16:10) to do. So let me copy paste it and then do it two more times, let us say i plus 5, i plus 6, the entire Chennai right now is shifted by one letter.

(Refer Slide Time: 16:37)



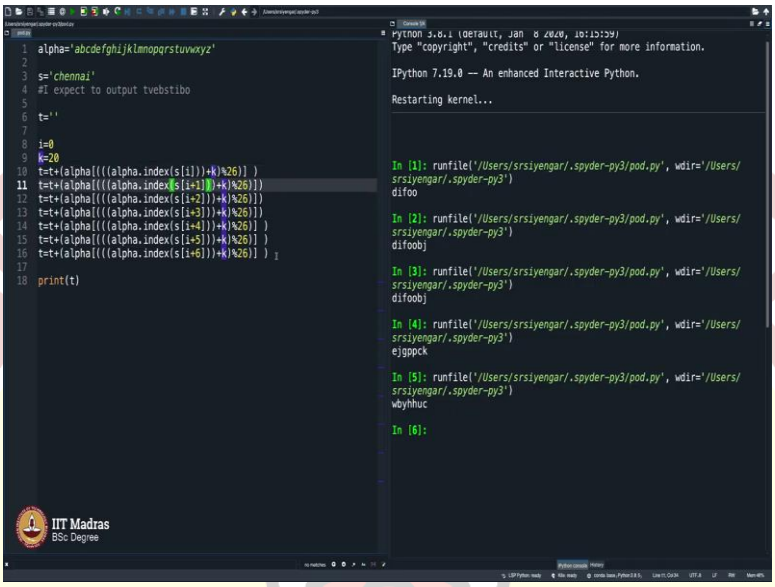
Now what if you want to shift to b by two letters. Then instead of 1, I just need to put 2. Let me make that a variable here. Now you see an application of variables, I say k equals 1 and make this 1 k, k, k. Why would I do that? You will see that in a minute, I only explained that if I want this k to be different, I do not want to shift once, I want to shift twice, let me execute it. Perfect, it is the same answer.

(Refer Slide Time: 17:06)



But if I want to shift it twice, then I may want to put k equals 2, when you put k equals 2, look Chennai is getting shifted twice k equals 2 and c is becoming d, e; h is becoming i, j; and e becomes g; n becomes an o, p; n becomes again an o, p; a becomes c and i becomes k. Very simple.

(Refer Slide Time: 17:30)

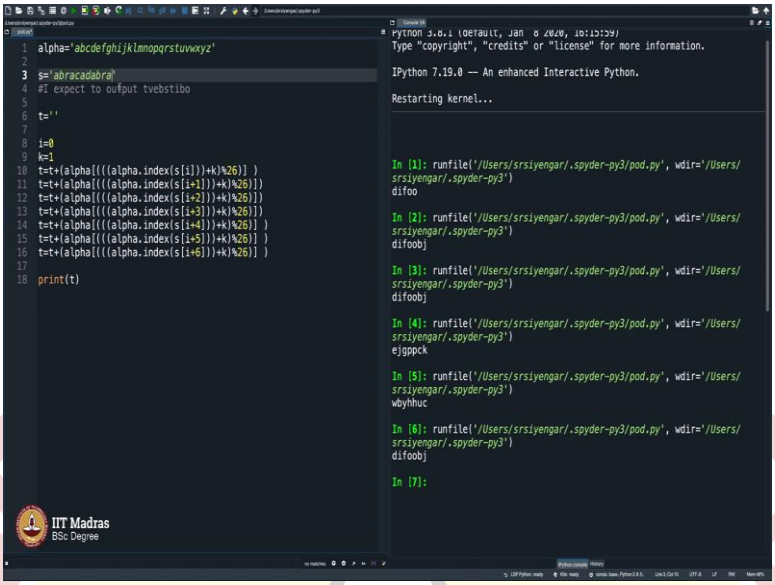


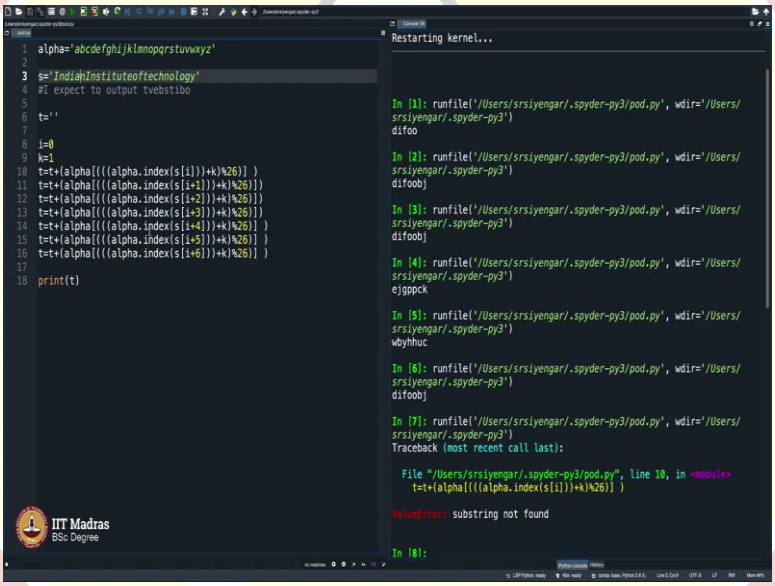
If you want to shift even more, let us say by 20 units, whatever that is. So c becomes w; h becomes b because it goes beyond z, as you can it is getting circled and it is coming back. Very good. So as you can see, points to note here is that point number 1, programming can get complex, you are seeing that it is getting complex here, you may have to think through it. In fact, as I was compiling this, I ended up getting confused a little, but then when I wrote it down, it was pretty clear to me.

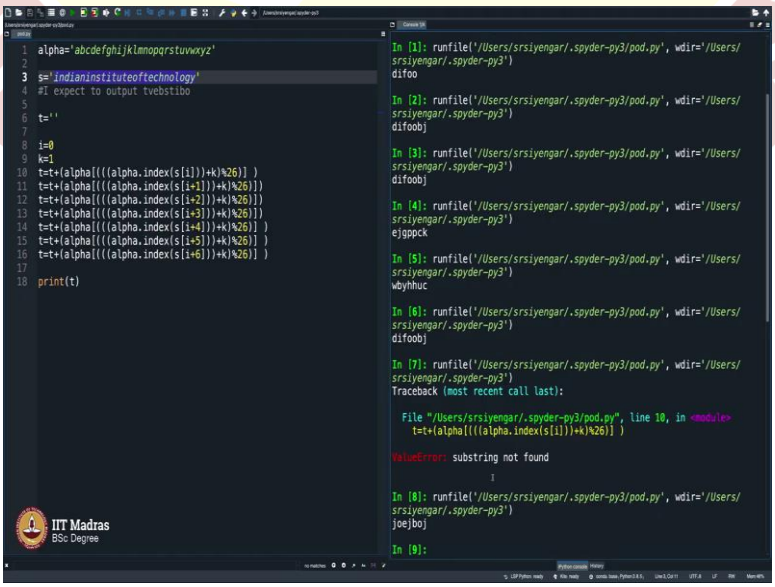
It is just a matter of practice, you will get used to it, also please note that errors are very common in programming. As in when you type, it is never flawless, as in when you type, you will face errors and you must fix them on the fly. So it was complex here, so point to notice, things can get complex.

Point number 2, the moment you type whatever you type here, you must type it so many number of times here. Point number 3 is you can keep this k as a variable instead of putting it 1 1 1 1 here in k or 2 2 2 2 instead you can simply put k equals whatever number you want and execute it, you will get a shift. This is another quick, importance of why use variables. Perfect.

(Refer Slide Time: 18:59)

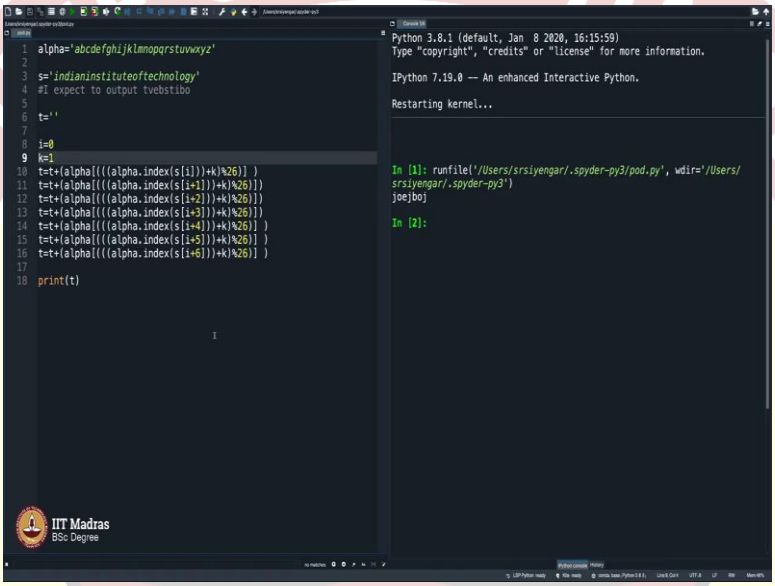






Try coding this, you probably of wondering is there any way we can avoid writing so many lines here, if it is a very long word like abracadabra, it is a very long word or let us say we have IIT, indianinstituteoftechnology, if you want to convert this, this will only convert up to this point, it will not go beyond that. So, it is already throwing error here saying that the substring is not found. That is because we have you know capital I here, see such small mistakes one has to be very careful about. So capital I here created a problem and now it is going good.

(Refer Slide Time: 19:51)

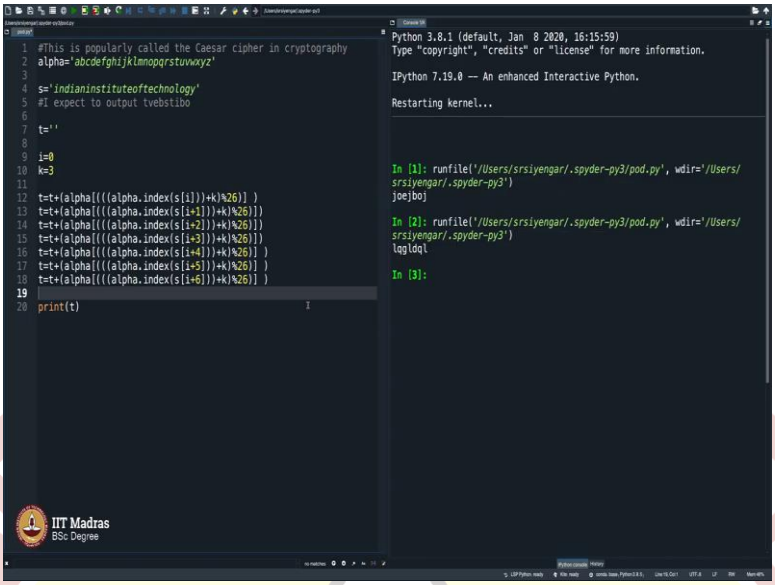


Clear screen and let me show it to you what is this is doing is simply taking the first 1 2 3 4 5 6 7, seven letters of this and then shifting it by one Indian becomes j o e j b, and you should type this so many times whatever is the number here, I mean how much, I will be give this 1 2 3 4 5 6 7 8 9 10, roughly maybe 25 approximately. So you should type this 25 times, is there any way we can automate this?

Yes, you can. We can do this in just a one liner, we are coming there, the next will be an introduction to how to make repetitive things easily some sort of codable, by writing a piece of code. So, what I will then do is instead of going this one by one, one letter at a time, what I will do is, I will tell the computer go through these letters one at a time, until you hit a dead end, there is a way to do that. And we will be discussing that in the next week.

But as if now, you be, this is just a warm up exercise for you all to understand that we can meddle around with strings like this. In fact, what we tried doing just now is popularly called the Caesar Cipher.

(Refer Slide Time: 21:09)



When k equals 3, which is three shifts. This code let me just write that down in the comment. Just about you can take a look at it; this is popularly called the Caesar Cipher in cryptography. If you want someone to not read your messages, then you can shift it by a particular k and then use that k as a key and tell them that unless you shift back by k units, you will not know what the text is.

That is precisely what we did just now, we just did one way, the other way is to shift it back. You make a minus k here and you will get back the letter. So you can try that out all by yourself and that is it for, I would say in this week this is slightly complicated an example, but then do not worry, we have to introduce complexity somewhere, we introduced it here. Things will get easier with time and in fact you will be thrilled to see, that this very same code can be done, by the fact end of the third week you will be able to write this code in just a couple of lines.