



IIT Madras

ONLINE DEGREE

(Refer Slide Time: 00:16)

```

1 '''This is a piece of code to check if a given
2 list has 0 in it or not. If it has "zero" in it, we
3 return true(1), otherwise we return false(0)
4 '''
5
6 def check0(L):
7     #if the list is empty, return False
8     if (len(L)==0):
9         return 0
10    if (L[0]==0):
11        #if the first element is zero then return 1
12        return 1
13    else:
14        return check0(L[1:len(L)])
15    #the above code simply outsources.
16
17 ans=(check0([1,2,4,5,10,8,7,3]))
18
19 print(ans)
20
21 '''Please note, this is not an efficient code, we
22 will see this later'''

```


```

In [7]: run recursion.py
1

In [8]: run recursion.py
0

In [9]:

```



IIT Madras
BSc Degree

"recursion.py" 22L, 5368 written

[0] 0:Vim*

"Sudarshans-iMac.local" 21:14 20-Jun-21

So, a small request, before writing the code, I would like to tell you all that I am using a different editor. I am not using Spyder anymore. If you remember, we started off with a web-based editor and then we moved to integrated development environment IDE, which was Spyder, and now I have moved to a simple editor. This is a simple vim editor, also called, this is called the Vim editor. So, it is there on Unix and on Mac. I like using this.

And on the right side, I have the IPython, simple IPython, interactive Python terminal. The reason why I have moved to this is that it is lightweight. It does not, it is a very simple editor and a simple IPython terminal. It is also easy for me to record this. And for the viewers also, the background and the foreground combinations are nice on their eyes. So, I have moved to this.

So, do not worry, you need not move to. You can continue to use Spyder or any other programming environment that Python programming enrollment that you like. Now that people are matured, I thought I will make the shift. You can also try moving to your own choice of editor, although I would not recommend it unless you are completely familiar with it.

So, I will now come this side and write my code. I will write my code here, print hello world and I will execute that this side recursion, recursion dot Py. That is the name of the file. This is how we will be doing right from now onwards. It should be very confusing because we are in our advanced weeks of the course. We should introduce slightly uneasy things with time so that you are familiar with everything. So, without further ado, let me get into the program now.

What were we supposed to do? I was supposed to write a code to check, this is a code, this is a piece of code to check if a given list has 0 in it or not. If it has 0 in it, we return true, which is 1, otherwise we return false which is 0. We will only return these numbers. So, what I do is I will close this comment and start writing my code. So, what is this define check 0? I will take a list as input and then check if the list is empty or not.

First, I will see if the list is empty, if length of L equals 0, I will simply return 0, false. I did not see zero because the list itself is empty. If the list is empty, return false, 0 means false. And if it is not, in case it is not empty, then what I do is I check if the list L, the first element if it is 0 or not. If it is 0, if the first element is 0, then return true. That is what I am going to do now.

I will say return true, because I found element 0. This is about check 0, you see. I found the element 0. But what if this is not true? What if L of 0 is not 0? Else, we will go through the code once again just so that it is easy for you people. It is not confusing. By this time I believe you should be familiar with what is happening here. It does not require explanation. So, else, what I will do is I will return the outsourced part of this boom recursion comes.

L of 1 to len of L, what do I mean by this? Let me come this side and tell you, if you were to create a list, let us say L equals 0, 1, 2, 3, 4, L of 1 to len of L will simply give you from 1 to 4. This is the things that you should get familiarized with by this time. So, that is what I am doing. I am omitting the first element and trying to feed the rest of the list here. And that is it.

So, let me just see if this works. I will say print check 0 of 1, 2, 0, 4, 5, 10, 8, 7, and maybe 3, and so on. The bracket is not close, I will close the bracket. And there I am I will execute this. You see, I am just printing the, or if it is easy for you people. So, why am I assuming that you all are very good at finishing the code in one line. So, let us do it in two lines. It is easy on the minds and then I will print the answer.

So, I will come here and then run this code, run recursion dot py. There is a problem. Check 0, what is the problem? Answer equals so and so, function object is not subscriptable. So, I made a mistake here. I should put a bracket here. Oh my god. So, this should fix the problem. So, it is now, let me clear the screen here. All these pop ups I will remove from next video onwards, please bear with me. So, if I say run recursion, it shows me 1 just because there is 0 here. If there were to be no 0 here, I would expect a false here. It is says false.

So, what is happening here? Let me look at the code. You are checking the presence of 0 in the list L. And I will see if the, first of all, the list is empty or not. If it is empty, I will return false. There is 0, there is no 0 there simply because the list itself is empty. The list is not empty, I will just see the first element, just the way I told you, I will just wash one vessel and leave the rest. That is precisely what I am doing here.

So, I will just check the first element. If it is 0, I return 1. The moment you return 1 everything ends here. The Python ensures that you come out of this function, it does not execute any further. If this is not true, assuming it is not true, else, which means this is not true, first element was not zero, then pass the responsibility to the function with a smaller list.

The above code simply outsources, I hope there is a word called outsourcers, if not, you know what I mean, outsources it to a smaller list for the function. It gives a smaller list. This is the smaller list. Outsources this to a smaller list is coming to the next line. So, what I will do is I will, you know this so simply outsources, as simple as that.

And the moment I say this, the function takes this and then checks whether 0 is there or not. If it is there, it says, yes. If it is not there, it says no. A simple program, take a look at it, think about it and one word of advice to some advanced programmers. I am sure you have a problem with this code. Please do not correct me here. There are reasons why we should make mistakes.

For beginners, you would not know what mistake I have made. But for people who know programming well, they will say there are two mistakes here. I will address them later. It is important for us to simply write A, B, C, D, E, F, G, H up to Z not worrying about our handwriting, and later on, we should fix the handwriting. First, we must learn the alphabets.

So, I am introducing recursion right now. I will tell you what is actually not so nice about this code. This code works, but it is not a good way to, please note, this is not an efficient code. We will see this later, suspense. But all you need to know is this particular function works recursively simulate exactly your, what does it simulate, it simulates exactly the vessel question or let us say factorial question.

You take a list, you check whether it has 0 or not. The idea is simple. Just take the first element and if it is not 0, check the rest. That is all we are saying this simple English statement is converted to code. I will not go in this detail. From next video onwards, we will jump into programming and you may want to record, see the recording twice or thrice for it to make sense. I am sorry. I might have taken more time than what I am supposed to take at this time.

I am going to assume, in other words, I am going to assume more maturity from your people. So, let us go ahead and solve a couple more questions in recursion.