



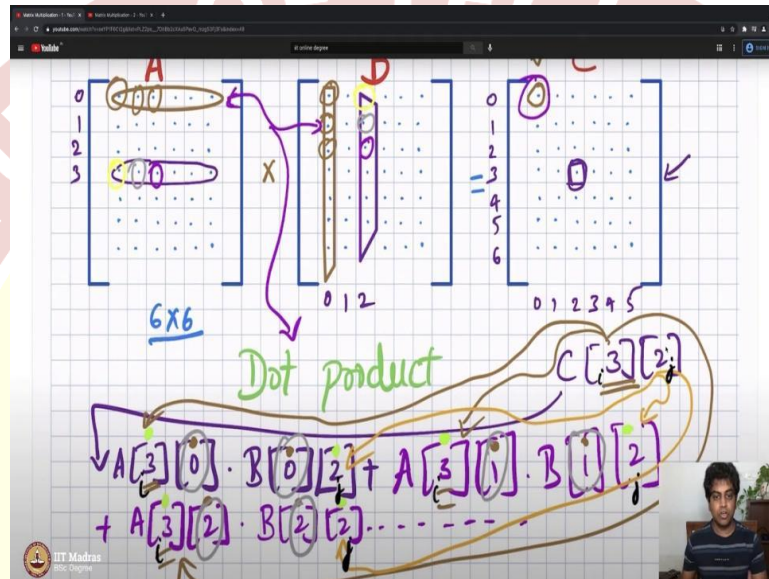
# IIT Madras

ONLINE DEGREE

**Programming in Python**  
**Professor. Sudarshan Iyengar**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Ropar**  
**Matrix Multiplication Using Functions**

Let us take a look at the matrix multiplication program that we discussed the last time. Let me open the video and then try to see what is it that we discussed here.

(Refer Slide Time: 00:27)



So, as I see, we took two matrices, as you can see A and B, and then tried multiplying it and we got C. So, I was trying to explain here that any entry here, let us say C 11 is basically the dot product, what is called the dot product of two vectors. You see the first row and first column gives you the first entry in C, the C 11.

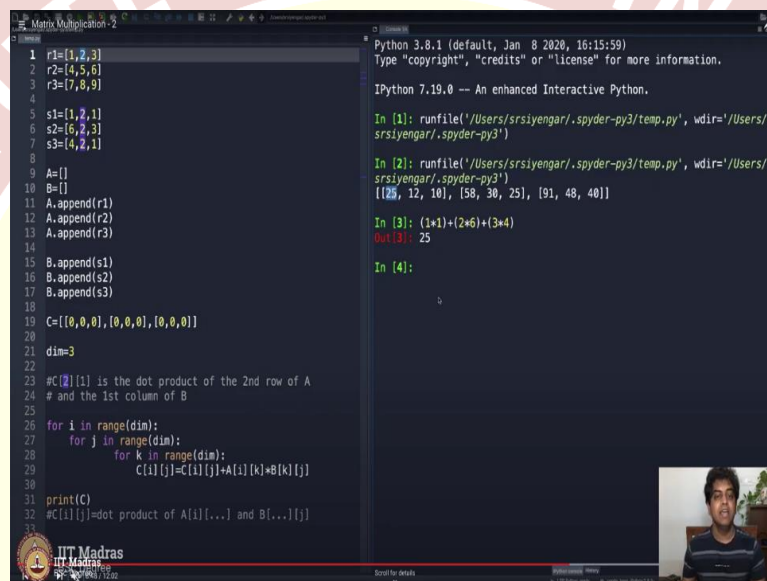
We went ahead, we showed how we do the dot product, and then finally, we get the answer. I have just opened the video and I am trying to see, how we explain matrix multiplication, I would like to redo the code right now, of course, using functions. So, we are trying to illustrate here that the ith row and the jth column results in Cij here. Cij, C ij simply is ith row off A ith throw third row remains the same, but the columns change. First column, second column, third column, fourth column, fifth column.

Of course, the first one is called the zeroth column. So, 30, 31, 32 and so on, here we have second row, I am sorry, second column, and then the rows keep changing. Second column, first row, which is 02 entry here, and 12 entry here, and then 22, 32, 42 and so on. So, when

you multiply these two things, what you get is the third row of A and the second column of B results in the entry C 32. Correct that is what we explained.

I will quickly go ahead just fast forwarding and seeing what I explained basically the dot product as you are able to see, and that is pretty much it, very good. So we understood this part, the last time if you did not, the mistake probably is mine, because I probably did not explained it properly, so let us go ahead and then see what was the program that we tried writing the last time.

(Refer Slide Time: 02:34)



The screenshot shows a Jupyter Notebook interface with two main panels. The left panel displays a Python script for matrix multiplication. It defines three rows of matrix A (r1=[1,2,3], r2=[4,5,6], r3=[7,8,9]) and three columns of matrix B (s1=[1,2,1], s2=[6,2,3], s3=[4,2,1]). It then initializes a 3x3 matrix C with all zeros. A nested loop calculates the dot product of the 2nd row of A and the 1st column of B, storing the result in C[2][1]. The output of this calculation is 25. The right panel shows the execution of the code, with the output of the dot product calculation displayed as 25. A small video feed of the presenter is visible in the bottom right corner of the notebook interface.

```
1 r1=[1,2,3]
2 r2=[4,5,6]
3 r3=[7,8,9]
4
5 s1=[1,2,1]
6 s2=[6,2,3]
7 s3=[4,2,1]
8
9 A=[]
10 B=[]
11 A.append(r1)
12 A.append(r2)
13 A.append(r3)
14
15 B.append(s1)
16 B.append(s2)
17 B.append(s3)
18
19 C=[[0,0,0],[0,0,0],[0,0,0]]
20
21 dim=3
22
23 #C[2][1] is the dot product of the 2nd row of A
24 # and the 1st column of B
25
26 for i in range(dim):
27     for j in range(dim):
28         for k in range(dim):
29             C[i][j]=C[i][j]+A[i][k]*B[k][j]
30
31 print(C)
32 #C[i][j]=dot product of A[i][...] and B[...][j]
33
```

Python 3.8.1 (default, Jan 8 2020, 16:15:59)  
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')

In [2]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')

[25, 12, 10], [58, 30, 25], [91, 48, 40]]

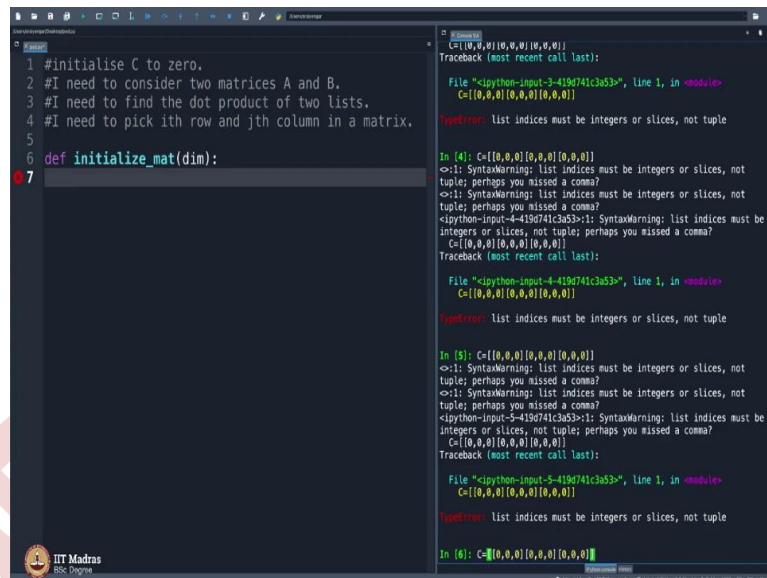
In [3]: (1\*1)+(2\*6)+(3\*4)  
Out[3]: 25

In [4]:

So, as I remember, we took this matrix A and B, and then we initialized C to all zeros, and then all that we did was, take the dot product of the ith row of A, as you can see here and the jth column of B, and we assign that to C of ij. And that is pretty much it. How did we do it? It was a very complicated process with some for i in range dimension j in range dimensions C of ij is dot product of A and B of j, I wrote that in English and then tried converting that to a function, you see.

So, I remember, it was quite a long session of thinking, and programming, so I get this and that is pretty much it, 3 for loops, 1 within the other and I print C, and C comes here, and we even verified whether it is right or wrong. And we got 25, which was the first entry in C. Perfect. So far, so good.

(Refer Slide Time: 03:36)



```
1 #initialise C to zero.
2 #I need to consider two matrices A and B.
3 #I need to find the dot product of two lists.
4 #I need to pick ith row and jth column in a matrix.
5
6 def initialize_mat(dim):
7
```

Traceback (most recent call last):

File "<ipython-input-3-419d741c3a53>", line 1, in <module>  
C=[[0,0,0],[0,0,0],[0,0,0]]

TypeError: list indices must be integers or slices, not tuple

In [4]: C=[[0,0,0],[0,0,0],[0,0,0]]

<~! SyntaxWarning: list indices must be integers or slices, not tuple; perhaps you missed a comma?>

<~! SyntaxWarning: list indices must be integers or slices, not tuple; perhaps you missed a comma?>

<ipython-input-4-419d741c3a53>:1: SyntaxWarning: list indices must be integers or slices, not tuple; perhaps you missed a comma?  
C=[[0,0,0],[0,0,0],[0,0,0]]

Traceback (most recent call last):

File "<ipython-input-4-419d741c3a53>", line 1, in <module>  
C=[[0,0,0],[0,0,0],[0,0,0]]

TypeError: list indices must be integers or slices, not tuple

In [5]: C=[[0,0,0],[0,0,0],[0,0,0]]

<~! SyntaxWarning: list indices must be integers or slices, not tuple; perhaps you missed a comma?>

<~! SyntaxWarning: list indices must be integers or slices, not tuple; perhaps you missed a comma?>

<ipython-input-5-419d741c3a53>:1: SyntaxWarning: list indices must be integers or slices, not tuple; perhaps you missed a comma?  
C=[[0,0,0],[0,0,0],[0,0,0]]

Traceback (most recent call last):

File "<ipython-input-5-419d741c3a53>", line 1, in <module>  
C=[[0,0,0],[0,0,0],[0,0,0]]

TypeError: list indices must be integers or slices, not tuple

In [6]: C=[[0,0,0],[0,0,0],[0,0,0]]

Now we will go ahead and we will try to see, if we can write the same piece of code the easy way. What do I mean by easy way? By easy way I mean, I am not going to think so much as much as I thought, the last week when I was trying to write a code for matrix multiplication. In fact, I was thinking about three, four loops. You see, I was I was trying to figure out how I could compute C of ij and I sort of messed up if you remember. Messed up as in, it was a long session, it was not clear on the mind.

So, I purposefully wanted to go slow and tell you people that your mind can get confused when the problem becomes complex. Functions, the idea of functions helps you stay organized; number 2. Number 2, is the idea of functions helps you write a small module some facility, and frequently call it whenever you want. So, it is like having a car and a car driver. Whenever you want to go somewhere, you just send a message and your car and your driver is ready. So, functions are like that.

When you keep using the same routine again and again and again, in your programming it is good to write that block of code and keep it separately. And that is what your def facility gives you in Python. So, let us go ahead and try to see if we can write a piece of code that uses functions and let us see how simplistic matrix multiplication becomes.

So, what should we do in matrix modification? I need to what, initialize C to 0. I need to consider two matrices A and B perfect. I need to find the dot product of two lists, correct, and then I am a little particular about putting foodstuffs at the end of the comments. So, and then what do I do let me think, I take the ith row of A and the jth column of B, and figure out what is C of ij. Ijth entry of C is ith row of A dot product with jth column of B. So, I need to pick

So, what I will do is I will individually write code for these things. So, for example, define initialize a matrix and I will say, dim, dim means the dimension. You will in a minute, see what I am trying to do. All I am trying to do is, I need a code here let us say on the terminal I am typing. If I said C equals initialize matrix 3, then, of course, it will throw error here because you do not have that function ready.

(Refer Slide Time: 07:32)

```
1 #initialise C to zero.
2 #I need to consider two matrices A and B.
3 #I need to find the dot product of two lists.
4 #I need to pick ith row and jth column in a matrix.
5
6 def initialize_mat(dim):
7     C=[]
8     for i in range(dim):
9         C.append([])
10    for i in range(dim):
11        for j in range(dim):
12            C[i].append(0)
13    return C
14
```

```
C=[(0,0,0),(0,0,0),(0,0,0)]
Traceback (most recent call last):
  File "<ipython-input-6-4153741c3e53>", line 1, in <module>
    C=[(0,0,0),(0,0,0),(0,0,0)]
TypeError: List indices must be integers or slices, not tuple

In [6]:
In [6]:
In [6]:
In [6]:
In [6]:
In [7]: C=[(0,0,0),(0,0,0),(0,0,0)]
In [8]: print(C)
[(0, 0, 0), (0, 0, 0), (0, 0, 0)]
In [9]: C=[[],[],[]]
In [10]: C[0].append([0,0,0])
In [11]: C
Out[11]: [[(0, 0, 0), [], []], []]
In [12]: runfile('/Users/srriyengar/Desktop/pod.py', wdir='/Users/srriyengar/Desktop')
In [13]: print(initialize_mat(3))
[(0, 0, 0), [0], [0]]
In [14]: runfile('/Users/srriyengar/Desktop/pod.py', wdir='/Users/srriyengar/Desktop')
In [15]:
```

So, I want, if it is two-dimensional matrix it should be 0, 0, 0, 0, three-dimension should be 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, so on so forth, for any dimension, this should work. Let me think. So, I will initialize C to this an empty list, and then I will say for i in range dim C append an empty



list well that is what I do. So, before going any further, I do this, I first want to initialize this to, boom, boom, boom, boom, boom.

So, and then I will say C of 0, append 0 comma 0 comma 0 do you get that? So, you have 0, 0, 0 here and see that is what I want to do. I have done this in this line here. So, I have switched on the video for this very session so that you can also see the expressions on the face of the programmer.

All sorts of moods, you can observe frustration, happiness, happiness only comes at the end, when the code runs without any errors. Until then it is lot of frustration. So, you see what is happening here. The best part is, I am able to think slowly. I only want to think about initializing a matrix. I do not want to worry about matrix multiplication here, because I have broken it into four pieces, which I have written here. As you can see, this these 4 pieces. So, I am only worrying about the first piece right now.

C append a small list and then for every entry in C again, for i in range dim. Why do I do that? You will understand in a minute. I will say C if I zeroth in entry, which is the first entry here as you can see this part, this part, I will append 0 here. How many times? As many times as the dimension of the matrix.

The matrix is three dimension. If they dim is 3, I will make a, make three entries of 0 here. I hope this works. The best part is, I have done something right now. Look at the psychology of a programmer, right now. Programmer has done something and I do not have the patience to even check whether it works or not the program way. I do not read every line of it. Whatever came to my mind, I saw on the terminal, and then I wrote a piece of code here. I can quickly check whether it works or not.

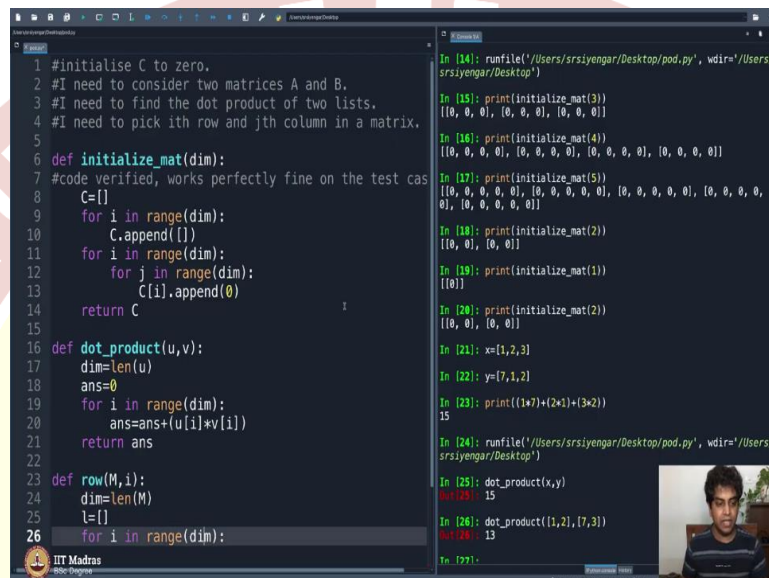
So, let me try doing that. I will come here, you know how to do this. So, I will say print initialize matrix, let us say three. Boom. There is a error. So, what is happening here? It is initializing to 0, 0, 0 but it should be less to 0, 0, 0. So, where am I going wrong? So, this code is not right, correct, let us fix this.

For i in range i include three brackets and then i in range of C of 1 append I should do it a few times, you see? Or maybe I should, I could do this. So, what will this do? This will for every time I append something C of i no, no, no, no, no, no, no, everything is perfect. Just that it should append a little more times. For j and range dim for every entry, it should append

00, that is it. So, this one only appends empty brackets and this one gets you into each one of those brackets, and appends zeros in it. The dimension dim number of times.

Probably, there is a easier way to do this in just one line. Some of you can try it, I clearly see that you can actually do it in one line, but let us stick to this code. So, return C I hope this works.

(Refer Slide Time: 12:02)



```
1 #initialize C to zero.
2 #I need to consider two matrices A and B.
3 #I need to find the dot product of two lists.
4 #I need to pick ith row and jth column in a matrix.
5
6 def initialize_mat(dim):
7     #code verified, works perfectly fine on the test cases
8     C=[]
9     for i in range(dim):
10         C.append([])
11         for j in range(dim):
12             C[i].append(0)
13     return C
14
15
16 def dot_product(u,v):
17     dim=len(u)
18     ans=0
19     for i in range(dim):
20         ans=ans+(u[i]*v[i])
21     return ans
22
23 def row(M,i):
24     dim=len(M)
25     l=[]
26     for i in range(dim):
```

```
In [14]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
In [15]: print(initialize_mat(3))
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
In [16]: print(initialize_mat(4))
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
In [17]: print(initialize_mat(5))
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]
In [18]: print(initialize_mat(2))
[[0, 0], [0, 0]]
In [19]: print(initialize_mat(1))
[[0]]
In [20]: print(initialize_mat(2))
[[0, 0], [0, 0]]
In [21]: x=[1,2,3]
In [22]: y=[7,1,2]
In [23]: print((1*7)+(2*1)+(3*2))
15
In [24]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
In [25]: dot_product(x,y)
Out[25]: 15
In [26]: dot_product([1,2],[7,3])
Out[26]: 13
In [27]:
```

Let me come here scroll up and then say, print initialize mat 3 boom, it works. So the best parties, it works for any dimension. Look at this 4, 5. Will it work for 2? Will it work for 1? It works for 1 that is a one-dimensional matrix one cross one matrix. So, 0, 0, 0, 0, perfect. So, this is the power of programming using functions.

I call it the functional approach to programming, but then there is something called functional programming, so you must be careful not to mess with both these terminologies. I like calling it functional approach to programming. So, use functions to code.

Alright, so now I look at this the psychology of the programmer, I now have peace of mind. I did not break my head on what initialize mat does inside this because the code is working fine. So, what I will do is I will come here. I will say code verified, verified. Works perfectly fine on the test cases. So, I like to say this that it gives me confidence that I have verified it.

So next, I will go to the next function, which is, I need to consider two matrices A and B, I need to find the dot product of two lists, that is important. When you take the ith row and the jth column you may want to find the dot product. How do you do that? How is dot product?

Done, so let me just think dot product of two vectors, let us say U and V is what, you have a list U have a list V, and you may want a temporary variable ans, where for i in range.

See here I assume that U and V are of the same dimension of the same length. So, that that again comes from your matrix dimension you see. Pause for a second and then think what I am trying to say. What I am trying to say here is, my dim will be len of U because length a U is same as length of V and U and V are simply entries in a matrix one is a row one is a column that is how you compute matrix multiplication.

So, I will say dim equals length of U. So, for i in range, what range dim, I will say answer is equal to answer plus U of i times V of i so we are used to writing i and j, if it is i and i mind gets confused. So, this is what I am doing. Component wise multiplication and then adding them this must return. Return I would say ans, again, psychology of the programmer, I quickly did the program, hoping that let us say, if I assume a list I would say x equals 1, 2, 3, then let us say Y equals 7, 1, 2. What is the dot product of this and this let me just write that down.

Let me scroll up. And then maybe the font size is a little smaller on the side. But I cannot help I cannot keep the font size big on both the sides. So, print 1 into 7, plus 2 into 1, plus 3 into 2. I know you are shouting aloud saying brackets needs to be put I am going to put that right now.

So, print so on and so on so and it is 15, that should be the answer. Let me just check. I will execute the program and use the dot product function here. Let us say even you need not use print, you can simply say dot product, X and Y or U and V or whatever. 15. Perfect. So, my dot product function is doing well.

I repeat, certain things are important that we repeat them as many times as possible, so that you are not confused. What is X, Y doing here, but what is U, V here. You might be wondering. You can put whatever you want here. In fact, you can directly put dot product of let us say two vectors, 1 comma 2, and 7 comma 3, so this is 7 plus 6, which is 13, it shows 13. You can put whatever you want here. The variables C here will be known to this part of the editor.

X is so much Y is so much you are simply passing X and Y, they are taken as U and V. And inside the function, the manipulation happens. All these sound like jargons but do not worry, so you will get used to it with time, but the point is just this you pass on the values of X, Y,



or alpha, beta, or me and you or let us say, elephant and a lion and a cat and a dog or whatever, whatever variables you are using you just pass in there and then it picks up from here. Then does the manipulation and returns the answer.

So far, so good, we found the dot product, we have initialized the matrix. I am sorry, it sounds like a rant, sounds like a philosophical discourse, but this is required we have to go slow. In a matter of a second in a matter of, let us say, I can challenge that anyone reasonably good with Python can finish the entire code in less than one or two minutes.

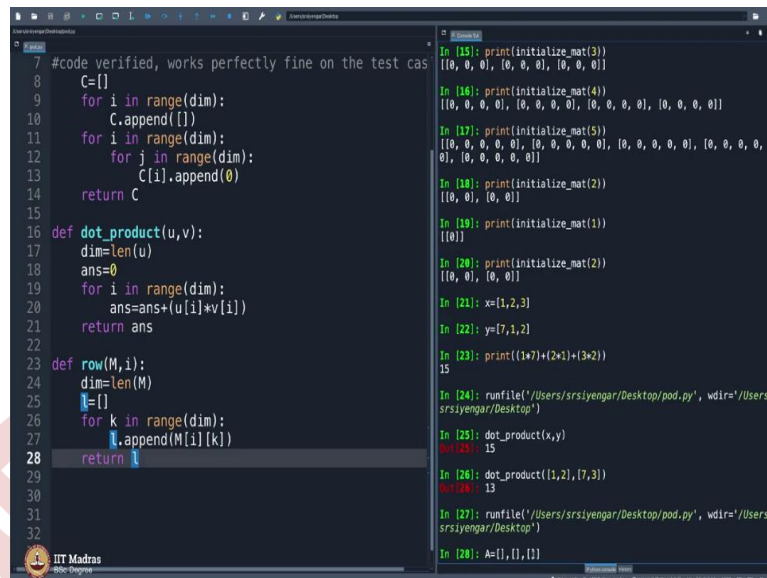
I can challenge, I can do it in two minutes. But on purpose I am trying to go very slowly, thinking every single step. Maybe I can do it in 2 minutes, and then debug it for the next 20 minutes. I do not know. I after closing this code, I will try to see if I can erase everything and do it in two minutes, and after that, I will brag, I will brag before I test myself that I can do it in two minutes. So, dot product is done. What do I do next? Psychology of the programmer. I wrote here that we need to pick the  $i$ th row and the  $j$ th column of the matrix.

Let me write a code a piece of function, which picks the  $i$ th row and the  $j$ th column. Define, I will call it  $i$ th row or how I do I put it, row  $i$  let me say row  $i$  I will write a function for this. For column I will write another function. How do I pick row  $i$ ? So, what is row  $i$ ?  $i$ th row is zeroth row or first row or second row, whatever it is, it should tell me so how do I do that? I will initialize a list  $l$ , I will return this list which is the  $i$ th row of the matrix.

So, what am I saying? I should put  $M$  comma  $i$  correct. The  $i$ th row of the matrix  $M$  correct. So, as you code only we will realize, you will start with a mistake and then you will converge towards error free code. So, in  $M$ , I would like to look at the  $i$ th row which means I will simply be for  $i$  in range,  $\dim$ . What is  $\dim$ ?  $\dim$  is the length of  $M$ .  $\dim$  is the length of  $M$ . You see I am stammering I am not very confident this happens while you are coding.

Sometimes I am blank. I do not know what to do next. You have to again stair at the monitor, look at where you are and then try to figure out what you are trying to do. Sometimes you may not be able to then you will want to go take rest or have a sip of coffee and then come back.

(Refer Slide Time: 19:10)



```
#code verified, works perfectly fine on the test cas
C=[]
for i in range(dim):
    C.append([])
for i in range(dim):
    for j in range(dim):
        C[i].append(0)
return C

def dot_product(u,v):
    dim=len(u)
    ans=0
    for i in range(dim):
        ans=ans+(u[i]*v[i])
    return ans

def row(M,i):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[i][k])
    return l

In [15]: print(initialize_mat(3))
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]

In [16]: print(initialize_mat(4))
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]

In [17]: print(initialize_mat(5))
[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

In [18]: print(initialize_mat(2))
[[0, 0], [0, 0]]

In [19]: print(initialize_mat(1))
[[0]]

In [20]: print(initialize_mat(2))
[[0, 0], [0, 0]]

In [21]: x=[1,2,3]

In [22]: y=[7,1,2]

In [23]: print((1*7)+(2*1)+(3*2))
15

In [24]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [25]: dot_product(x,y)
Out[25]: 15

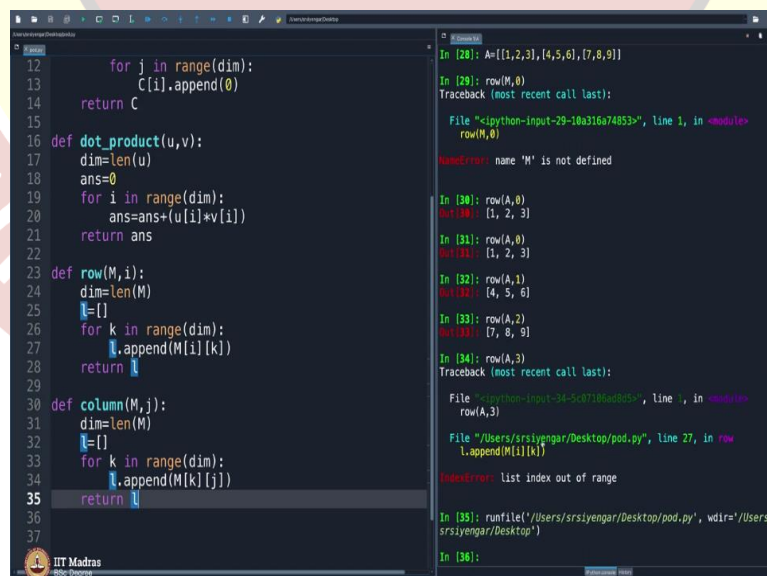
In [26]: dot_product([1,2],[7,3])
Out[26]: 13

In [27]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [28]: A=[[],[],[]]
```

So, for  $i$  in range dim, what do I do next, I need to append into  $l$  the matrix  $M$ 's  $i$ th row. So,  $i$  then I think I should not use  $i$  here I should use something else. Let us say  $k$  here. Because I need the  $i$ th row, and my column will vary between 0 to dim. I mean, zero to dim minus 1 rather. 0, 1, 2. My columns will vary, row will remain the same. It is the  $i$ th row, but the columns will vary. Good. So, however, clear it is even if the code is wrong, the output is right, I will not care. So, return  $l$ , let me see if this works or not. No error. So far, so good.

(Refer Slide Time: 20:35)



```
for j in range(dim):
    C[i].append(0)
return C

def dot_product(u,v):
    dim=len(u)
    ans=0
    for i in range(dim):
        ans=ans+(u[i]*v[i])
    return ans

def row(M,i):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[i][k])
    return l

def column(M,j):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[k][j])
    return l

In [28]: A=[[1,2,3],[4,5,6],[7,8,9]]

In [29]: row(M,0)
Traceback (most recent call last):
  File "<ipython-input-29-10a316a74853>", line 1, in <module>
    row(M,0)
NameError: name 'M' is not defined

In [30]: row(A,0)
Out[30]: [1, 2, 3]

In [31]: row(A,0)
Out[31]: [1, 2, 3]

In [32]: row(A,1)
Out[32]: [4, 5, 6]

In [33]: row(A,2)
Out[33]: [7, 8, 9]

In [34]: row(A,3)
Traceback (most recent call last):
  File "<ipython-input-34-5c87185ud9d5>", line 1, in <module>
    row(A,3)
  File "/Users/srsiyengar/Desktop/pod.py", line 27, in row
    l.append(M[i][k])
IndexError: list index out of range

In [35]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [36]:
```

So, I will take my matrix, let us say let me take a matrix  $A$ , which is three-dimensional, maybe, sorry, let me scroll up so that you can see it three dimensional with, let us say 1

comma 2 comma 3, 4 comma our favorite matrix 1, 2, 3, 4, 5, 6, 7, 8, 9. We have been using it since our 12 standard days. So, I will say row of M, first row zeroth row.

Name M is not defined, obviously, it should be A here. You see, I made a mistake. Here I use the variable M, but here I use A, so you should input A there, there is no matrix called M here. This part of I am showing my hands, I should probably use my cursor. So, this part of the computer or the – of my compiler knows only A it does not know M. So, this should give me the right answer.

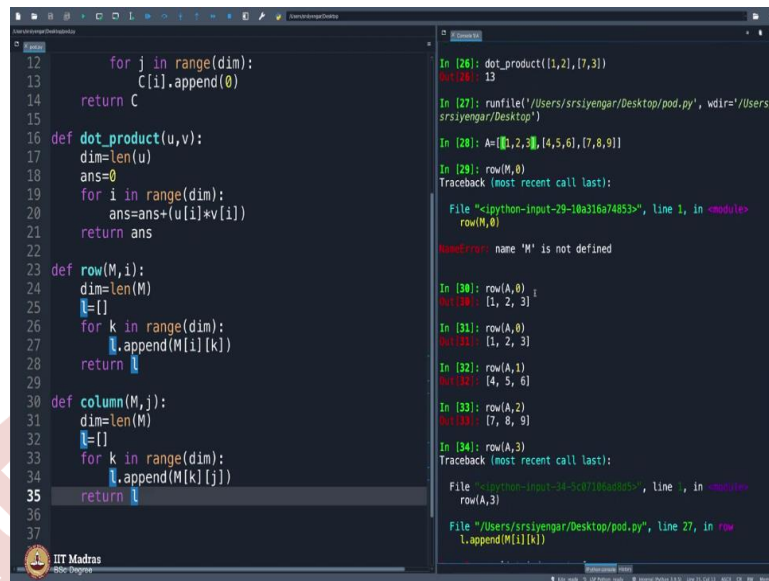
See, I told you right psychology of our programmer, there is a lot of frustrating moments and happy moments, moments, but there is also moments where you feel very happy. So, here is the moment where you feel very happy. You can see the first row, wow, second row, excellent. Third row. Yeah, compiler is not very happy because you went beyond the index it is only 0, 1, 2, that is why it is throwing up an error. Now, you see, in less than a few seconds, I will quickly write a code for column, which is jth column, let us say.

I can even put i here, but I will put j, it is easy in the mind. So it will be exactly the same dim equals length of M. Another advantage of, simply closing your eyes and coding, because if this code has something to do with the ith throw some minor manipulation will give you the jth column.

You see, I am confident I can do it, let us see. So, l equals again, it should be the same list for again, it should be k. For k in range, dim but then you append what, you append the kth row, and the jth column, the column remains the same, jth column and the rows change. If you are confused, at any part of you watching a programming session like this, you must pause and then think you should not go ahead with black boxes in your mind.

Or better even you should open a compiler and type the code alongside with me. Return l i suppose this works. Let us see column of the matrix A zeroth column 1, 4, 7. Wonderful. 1, 4, 7. Good.

(Refer Slide Time: 23:12)



```
12     for j in range(dim):
13         C[i].append(0)
14     return C
15
16 def dot_product(u,v):
17     dim=len(u)
18     ans=0
19     for i in range(dim):
20         ans=ans+(u[i]*v[i])
21     return ans
22
23 def row(M,i):
24     dim=len(M)
25     l=[]
26     for k in range(dim):
27         l.append(M[i][k])
28     return l
29
30 def column(M,j):
31     dim=len(M)
32     l=[]
33     for k in range(dim):
34         l.append(M[k][j])
35     return l
36
37
38
39
40
41
42
43
44
45
46
47
48
```

Execution Output:

```
In [26]: dot_product([1,2],[7,3])
Out[26]: 13

In [27]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [28]: A=[[1,2,3],[4,5,6],[7,8,9]]

In [29]: row(M,0)
Traceback (most recent call last):
  File "<ipython-input-29-10a316a74853>", line 1, in <module>
    row(M,0)
NameError: name 'M' is not defined

In [30]: row(A,0)
Out[30]: [1, 2, 3]

In [31]: row(A,0)
Out[31]: [1, 2, 3]

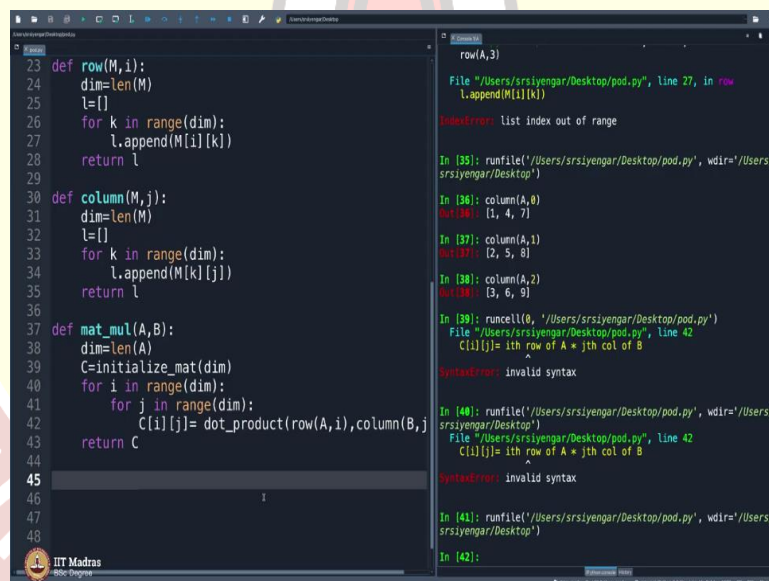
In [32]: row(A,1)
Out[32]: [4, 5, 6]

In [33]: row(A,2)
Out[33]: [7, 8, 9]

In [34]: row(A,3)
Traceback (most recent call last):
  File "<ipython-input-34-5c87106ad8b5>", line 1, in <module>
    row(A,3)
File "/Users/srsiyengar/Desktop/pod.py", line 27, in row
    l.append(M[i][k])
```

So right, 1, 4 and 7 is a first column.

(Refer Slide Time: 23:16)



```
23 def row(M,i):
24     dim=len(M)
25     l=[]
26     for k in range(dim):
27         l.append(M[i][k])
28     return l
29
30 def column(M,j):
31     dim=len(M)
32     l=[]
33     for k in range(dim):
34         l.append(M[k][j])
35     return l
36
37 def mat_mul(A,B):
38     dim=len(A)
39     C=initialize_mat(dim)
40     for i in range(dim):
41         for j in range(dim):
42             C[i][j]= dot_product(row(A,i),column(B,j))
43     return C
44
45
46
47
48
```

Execution Output:

```
row(A,3)
File "/Users/srsiyengar/Desktop/pod.py", line 27, in row
    l.append(M[i][k])
IndexError: list index out of range

In [35]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [36]: column(A,0)
Out[36]: [1, 4, 7]

In [37]: column(A,1)
Out[37]: [2, 5, 8]

In [38]: column(A,2)
Out[38]: [3, 6, 9]

In [39]: runcell(0, '/Users/srsiyengar/Desktop/pod.py')
File "/Users/srsiyengar/Desktop/pod.py", line 42
    C[i][j]= ith row of A * jth col of B
SyntaxError: invalid syntax

In [40]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')
File "/Users/srsiyengar/Desktop/pod.py", line 42
    C[i][j]= ith row of A * jth col of B
SyntaxError: invalid syntax

In [41]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/srsiyengar/Desktop')

In [42]:
```

What is the second column, let us check. Second column, the happiest of all moments is when you would write a piece of code, especially a function and then validate it this and it is all perfectly fine. 3, 6, 9. Is that right? 1, 2, 3, 4, 5, 6, 7, 8, 9, it must be right.

So, I am done with this. Now you see the marvelous magic that happens? Sorry, for the exaggeration. It is indeed, magical for me at least that in programming, you can break the otherwise complicated piece of code into smaller chunks, and then work on them. Dot product is done, row is done, ithh row is done, column jth column is done, initializing matrix is done.

What else do you have, I need to now multiply two matrices. Define mat multiplication, let us say of two matrices. Let us say A and B, whatever you give here, this you will be using within the function only.

The world outside this function knows nothing about those variables, A, B, or whatever you are using within the function. This is called scope, scope of a variable we will discuss that I had. It is not the right time to discuss a lot of jargons. This is the time for us to help you gain confidence in programming, and that is all I am going to do right now.

So, I am going to multiply matrix A with matrix B. How is that done? Firstly, I must initialize a matrix C, C equal to before that I must compute what is the dimension of a matrix? This matrix dimension would be let us say len of A why because A and B are the same dimension we are assuming throughout this program, throughout this course, that we are considering only square matrices.

If you consider what is called a rectangular matrices the number of rows is different from number of columns, blah, blah, blah lot of complications, let us not get there, you can do it. If you can do it, if you can do it do this you can do that too. Alright, so let us look at the simplistic case, where we are in the happy world where we only encompass square matrices.

So, length of A will be equal to length of B, otherwise, we cannot multiply them. We can in fact multiply square with a non-square matrix. We are getting it, we may hear it complicated, let us not get there. Dimension equals length of A. We know that A and B are of the same dimension assume it to be true. And then you see when you consider cases things get complicated that is why we should take the simplistic model.

And then what do I do? Why did I say l equals, dimension is len of A and then I need to initialize the matrix C. What is that? C equal to initialize matrix? What was the name of that function, initialize underscore mat, initialize underscore mat of dimension. Now you have C, which will have all leading zeros inside it, C is ready.

Now all that you should do is look at this magic now. For i in range dimension, for j in range dimension, you must compute C of ij that is all is your motive. You must come here, this new matrix C, which is the product of A and B. So, how do I do that? And this is what I mean magical, everything is ready for you right now. You initialize C, you can pick the ith row of a matrix and the jth column of a matrix as you are seeing.



Now I keep looking at the side because this is the side where I have the terminal. I do not know in the final video where the terminal or where the program part will be for you. So, I keep looking at this part, and then I am coding.  $C$  of  $ij$  is equal to  $i$ th row of  $A$  multiply dot product with  $j$ th column of  $B$ . Now what is that?

Let me see.  $C$  of, I will delete the first line there just so that I am clear, what is  $i$ th row of  $A$  row of a comma  $i$ , as simple as that. And then what is the  $j$ th column of  $B$ , column  $B$  of  $j$ , column  $B$ , comma  $j$  will give you the  $j$ th column of  $B$  row  $A$  comma  $i$  will give you the  $i$ th row of  $A$  and then you must now find the dot product of these two things that is very, very, very easy.

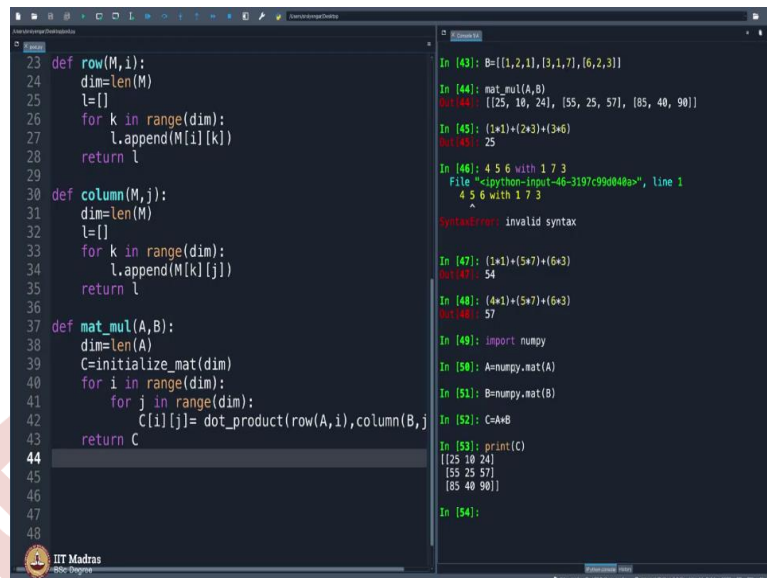
Is it really easy? It is easy because you have a function already kept ready. What was that function, I am just to be accurate dot underscore product is the name of the function. Dot underscore product of row of a comma  $A$  comma  $i$  column of  $B$  comma  $j$  will find the dot product of these two vectors, what two vectors, the vector row  $A$ ,  $i$  and the vector column  $B$ ,  $j$ , these two are lists you see, I mean, rather vectors, vectors lists.

I mean for programming we call it lists and in mathematics, we call it the vectors. I want to find the dot product of these two things that is already done before. So, the advantage here is that whatever you want to use bits and pieces, you keep them ready. So, that you can put all the pieces of the puzzle and you go hip, hip, hurray.

So, this should work. I suppose it does. If it does not, we will be disappointed. But I suppose God is great and this will work. So, return  $C$  and there you are, I will execute this invalid syntax, because I had to remove this. This was plain English for my understanding.  $C$  of  $ij$ , is  $i$ th row of  $A$  into  $j$ th column of  $B$ . You will not be doing it this hard if you practice well. I mean, one can go ahead and then quickly finish this in no time, but then I am doing it very slowly so that you can see how I am thinking.

So,  $C$   $ij$  is equal to dot product of row  $A$ ,  $i$  column  $B$ ,  $j$  written  $C$  everything is done. I will execute this, this should work. Thankfully, no errors, as you can see.

(Refer Slide Time: 29:15)



```
def row(M,i):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[i][k])
    return l

def column(M,j):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[k][j])
    return l

def mat_mul(A,B):
    dim=len(A)
    C=initialize_mat(dim)
    for i in range(dim):
        for j in range(dim):
            C[i][j]= dot_product(row(A,i),column(B,j))
    return C

In [43]: B=[[1,2,1],[3,1,7],[6,2,3]]
In [44]: mat_mul(A,B)
Out[44]: [[25, 10, 24], [55, 25, 57], [85, 40, 90]]
In [45]: (1*1)+(2*3)+(3*6)
Out[45]: 25
In [46]: 4 5 6 with 1 7 3
File "<ipython-input-46-3197c99d048a>", line 1
4 5 6 with 1 7 3
      ^
SyntaxError: invalid syntax
In [47]: (1*1)+(5*7)+(6*3)
Out[47]: 54
In [48]: (4*1)+(5*7)+(6*3)
Out[48]: 57
In [49]: import numpy
In [50]: A=numpy.mat(A)
In [51]: B=numpy.mat(B)
In [52]: C=A*B
In [53]: print(C)
[[25 10 24]
 [55 25 57]
 [85 40 90]]
In [54]:
```

Now what was a matrix A is already ready. I will take a matrix B, let us say, let me write a matrix, three columns. 1, 2, 3, 4, 5, 6, 7, 8, 9, and maybe 1, 2, 1, 3, 1, 7 and 6, 2, 3. This is my matrix B. And then I will say matrix multiply A comma B, let us see what it returns. It is returning a matrix, very good. Hopefully, this is the product of A and B, let us see. Let me take the first row of A and multiply with the first column of B. What is that? It is one times 1, 2, 3 is the first row, first column of B is 1, 3, 6, 1 into 1 plus 2 into 3, am I right? And then 3 into 6, 25, perfect.

So, the 00, C 00 is 25 makes sense? So, let me try to see if this is right, let us say what entry is this. Second row, third column of C, second row, third column of C meaning second row, third column, we should not say that. I mean oneth row and two-th column, second column because it starts from 0. Oneth row is 4, 5, 6. I must multiply 4,5,6 with what 2, 1 what is that third the second one is 1, 7, 3, with 1, 7, 3. Am I right? That is my 57. Correct.

Let me just be clear, this is C of 1 comma 2, 1, A of 1 first row and C 1, 7, 3 perfect. I think I am right. 4, 5, 6 with 1, 7, 3. I need to take its dot product that is all. 4, 5, 6 with 1, 7, 3 purposefully I made a note of it so that I can type it easily here. 1 into 1 plus 5 into 7, plus 6 into you would have done your automatic quickly 54. It says 57, what is going wrong. 1, 5, 6, 1, 7, 3 what am I doing? 4, 5, 6, 7 and 1, 7, 3? 4, 5, 6 and 1, 7, 3?

I am growing older I cannot 57. 57, perfect. It is 57 here and says 57 here. I think my, so if you are growing older there is a better way to find this out, I will show you that. Import numpy there is a built-in function to do this. As I show as I tried showing it to you the last week, A equals numpy of mat of A, it converts A to the type numpy B equals numpy mat B,

and then C equals A times B, print C, let us cross check this with this 25, 10, 24. 55,25,57. Perfect. It looks, look at this output 44th line. And this answer.

I have written the code that is built into the computer into the numpy library function. Anyways, I told you we are not going to use this many, many, many things that we are going to do will is there already, the point is to do it from first principles from the beginning.

So, what is the moral of the story? Peacefully, let us look at the piece of code that we wrote, we have broken that matrix multiplication thing that was so hard there. Although, English wise it was very easy, take the ith row and take the jth column and multiply them find the dot product that is the entry of C ij. That very same English we are converted here into program and the conversion was very easy simply because you took a modular approach using functions.

So, this is the power of programming using functions using modular approach. If you break them into small chunks, there are two advantages. One is easy in your mind. second advantage is that when you create a module, let us say like dot product, you can repeatedly use it, then use the same code in multiple places you can write a piece of code, keep it aside and then you whistle and it comes to you. I mean, to your rescue.

So, I hope you liked the video. I am sorry, probably in my video was a distraction when I was trying to code. I do not enjoy keeping my video on when I am coding, but I try doing it only once just for maybe fun sake. So, let us go ahead and see a couple more problems pertaining to functional approach to programming. Take care, bye