INDIAN INSTITUTE OF TECHNOLOGY MADRAS

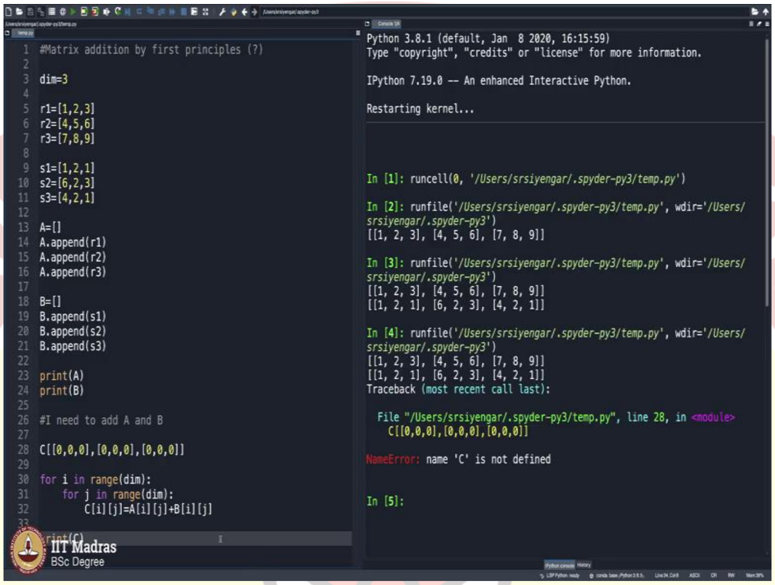
सिद्धिर्भवति कर्मजा

IIT Madras

ONLINE DEGREE

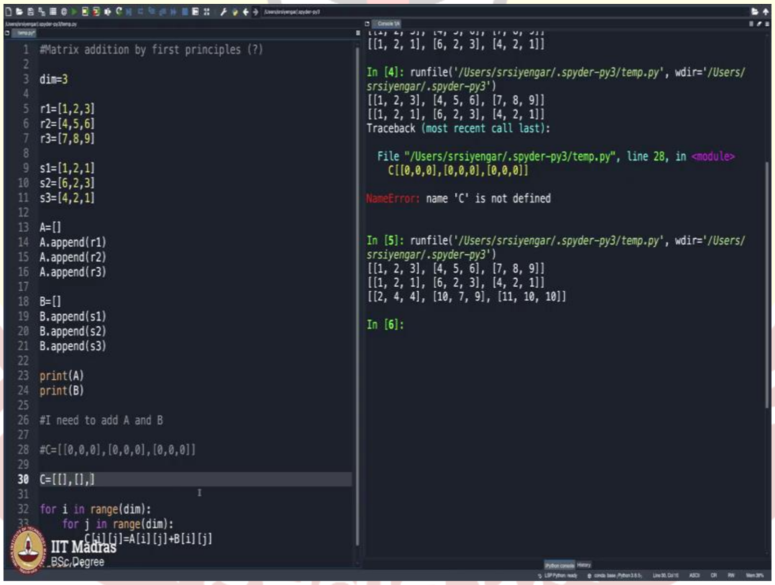
Programming in Python
Professor Sudarshan Iyengar
Department of Computer Science and Engineering
Indian Institute of Technology Ropar
Matrix Addition

(Refer Slide Time: 0:15)



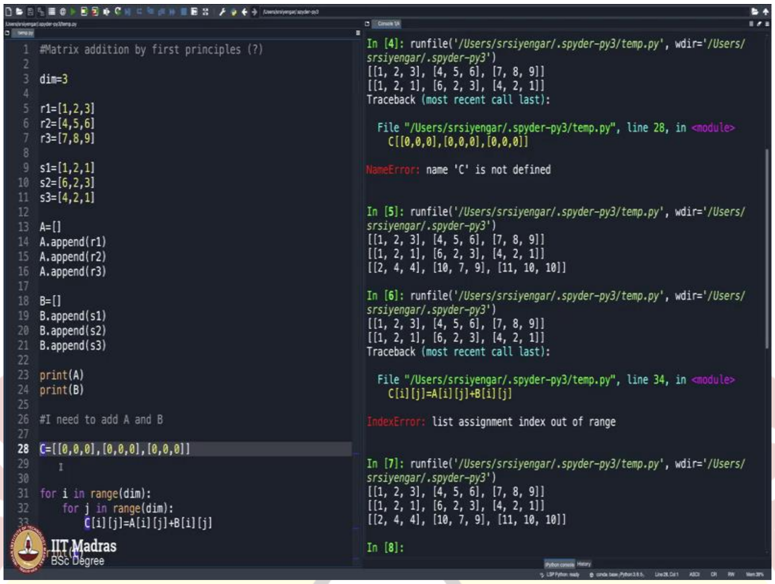
```python
#Matrix addition by first principles (?)

dim=3

r1=[1,2,3]
r2=[4,5,6]
r3=[7,8,9]

s1=[1,2,1]
s2=[6,2,3]
s3=[4,2,1]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)

print(A)
print(B)

#I need to add A and B

C[[0,0,0],[0,0,0],[0,0,0]]

for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]+B[i][j]

print(C)
```

```
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.

Restarting kernel...



In [1]: runcell(0, '/Users/srsiyengar/.spyder-py3/temp.py')

In [2]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]

In [3]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]

In [4]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, in <module>
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]:
```



```python
#Matrix addition by first principles (?)

dim=3

r1=[1,2,3]
r2=[4,5,6]
r3=[7,8,9]

s1=[1,2,1]
s2=[6,2,3]
s3=[4,2,1]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)

print(A)
print(B)

#I need to add A and B

#C=[[0,0,0],[0,0,0],[0,0,0]]

C=[[],[],]

for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]+B[i][j]
```

```
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]

In [4]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, in <module>
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]:
```



```python
#Matrix addition by first principles (?)

dim=3

r1=[1,2,3]
r2=[4,5,6]
r3=[7,8,9]

s1=[1,2,1]
s2=[6,2,3]
s3=[4,2,1]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)

print(A)
print(B)

#I need to add A and B

C=[[0,0,0],[0,0,0],[0,0,0]]


for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]+B[i][j]
```

```
In [4]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, in <module>
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 34, in <module>
    C[i][j]=A[i][j]+B[i][j]

IndexError: list assignment index out of range


In [7]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [8]:
```

Let us now look at the idea of matrix multiplication, let us write a Python program to do this. Please note that there are many library functions available where matrix related operations can be done very easily but then we are going to do it by first principles, let me write that down. Matrix addition by first principles. So, what do you mean by first principles?

By first principles you mean, you go to your kitchen and you start cooking and you see that your raw materials are not there, your ingredients are not there and what you do is you do not go to your shop and then buy them, you start growing them in your garden. That is what you mean by first principles, the point is to make your life hell by trying to do everything from your hand and you probably are wondering why should we do that, as I have been telling you people that is how you learn, you learn by doing things that are not ready-made but then creating things to whatever extent possible.

So, let us now go ahead and then try to write a piece of code for matrix addition, as and always we are going to use some bare minimum Python ideas but then not use a lot of library functions. So, what I will do is, please note as I proceed you will understand how I represent a matrix and how I do addition on two matrices.

Remember how we started this week. We started with lists, we started with lists of lists, I am going to make use of that idea, bear with me, things will become very easy and easy on your mind very soon as you note as I am typing. So, I especially like typing as I think, I do not generally come prepared with the ideas, that is because if I come prepared I stay a lot away from your level, you see I mean you cannot reach my level if I come very well prepared and try to write a very optimized code and things like that.

So, what I keep in my mind is I am you, trying to learn programming from the beginning, from scratch not knowing anything. So enough of my cacophony, I will start typing the code now. Let me say r1 is the first row of my matrix, let that be 1, 2, 3, r2 is let us say 4, 5, and 6, some matrix of your choice and then 7, 8, 9 this will be my matrix A, good.

So, and then another matrix with three rows let us say 1, 2, 1 and let us say second row will be six two three, in a minute you will understand what I am doing do not worry, 4, 2, 1 and A will be my matrix, it will be an array. I will append r1 to this, I will append r2 to this, I will append r3 to this. Some of you should be wondering why are you doing this circus of appending them, why cannot you just directly go and create a matrix? I want it to be easy on your mind, I am going very slowly line by line.

So, again create another array and then append s1, append s2 and append s3 and let me see how A and B looks like, let me first see A. I am executing this, I see A as this, a list of lists and then my B is as expected 1, 2, 1, 1, 2, 1, 6, 2, 3, 6, 2, 3 and so on. So, now I need to add these two matrices, I need to add A and B.

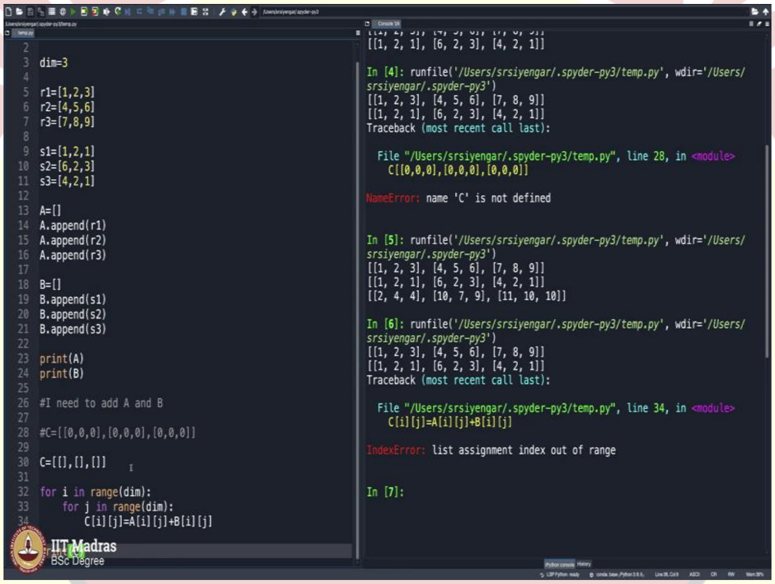
So, what I should now do is firstly I should understand the dimension of these matrices, the dimension of this matrices, I will call it as 3 for the time being I am only assuming I am taking square matrices where the number of rows is same as number of columns and they are just the size is 3.

And then now, what I will go ahead and do is create a new matrix C which contains all zeros. How do I do that? That is very easy, I will do it like this, 0 comma 0 comma 0 and then 0 comma 0 comma 0 and then 0 comma 0 comma 0. Now, what I will do is my C will be the answer, how? For i in range dim which is dimension which is 3 here, I call it, I could have put 3 itself here but then it is always useful to simply put dim here and then change it when your matrix is bigger than change it everywhere, you will understand in a minute.

For j in range dim I do C of i of j is equal to A of i of j plus B of i of j, A i j, B i j when you add that you get C i j as simple as that, pretty simple. And then I will go ahead and then print C now and then explain in detail there are a lot of things that I am going to explain now after executing this, I observe there is a problem it says there is no array called C, that is because I made a mistake here you see I should have put C equal to here.

And now I say execute, I get C here as you can see, I will scroll up, C is indeed 1, 2, 3 plus 1, 2, 1, 1 plus 1 is 2, 2 plus 2 is 4, 3 plus 1 is 4, 4 plus 6 is 10, 5 plus 2 is 7, 6 plus 3 is 9 and so on, very good. So, you see why should I do this, what if I do not do this and simply initialize C as a simple array of let us say 3 entries?

(Refer Slide Time: 6:26)



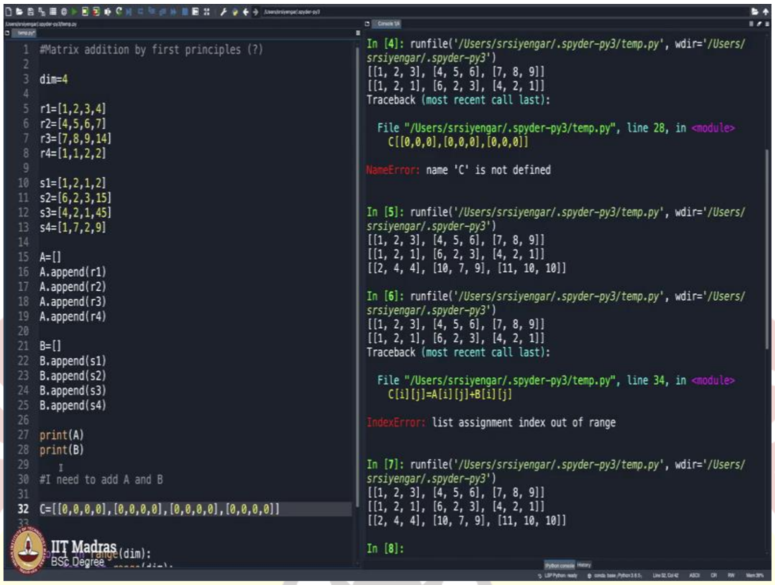
```

dim=3

r1=[1,2,3]
r2=[4,5,6]
r3=[7,8,9]

s1=[1,2,1]
s2=[6,2,3]
s3=[4,2,1]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)

print(A)
print(B)

#I need to add A and B

#C=[[0,0,0],[0,0,0],[0,0,0]]

C=[[],[],[]]

for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]+B[i][j]
```

```
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]

In [4]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, in <module>
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 34, in <module>
    C[i][j]=A[i][j]+B[i][j]

IndexError: list assignment index out of range


In [7]:
```

Python does not know how to handle it, there is a reason for it, the reason is that it does not know what you mean by C of 0 comma 0 there is nothing here, how are you able to assign something to it. So, of course it can be intelligent enough to initialize it to 0 but Python for various reasons does not do it and this is when you should understand programming languages independently has, they have their own sort of a startup brain, you should understand that this is the bare minimum that the Python does, on top of it you must use your intelligence logic and then code.

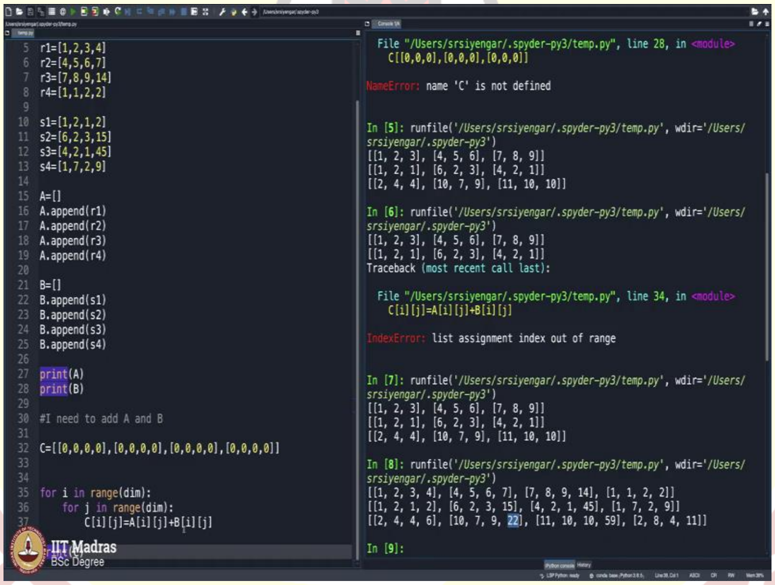
So, this is not acceptable, Python is not smart enough to figure out it should initialize it to 0, so use, put zeros there and then do the addition like this. In fact as you would have noted this entire thing can also become a part of a code just so that you can automate it, this very assignment of 0 can be done programmatically, try doing that I will not do it for you, you should figure out how to do it.

(Refer Slide Time: 7:27)



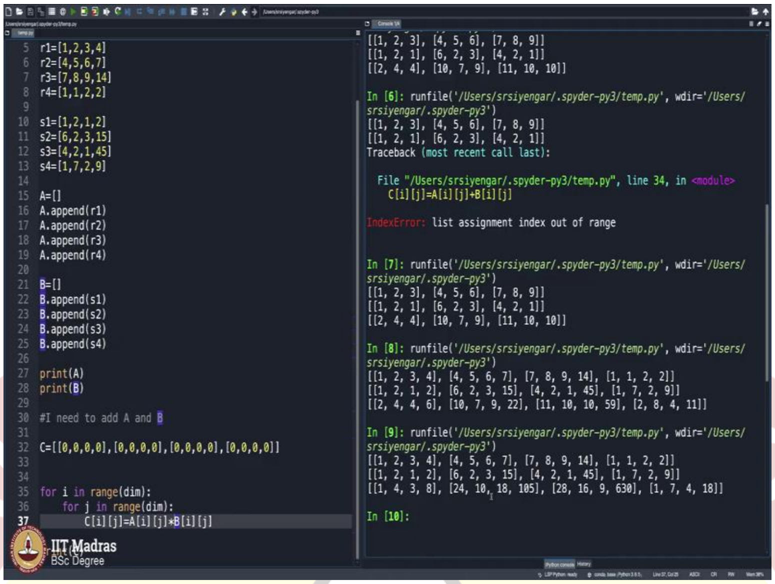
```
#Matrix addition by first principles (?)

dim=4

r1=[1,2,3,4]
r2=[4,5,6,7]
r3=[7,8,9,14]
r4=[1,1,2,2]

s1=[1,2,1,2]
s2=[6,2,3,15]
s3=[4,2,1,45]
s4=[1,7,2,9]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)
A.append(r4)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)
B.append(s4)

print(A)
print(B)

#I need to add A and B

C=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]

for i in range(dim):
```

```
In [4]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, in <module>
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 34, in <module>
    C[i][j]=A[i][j]+B[i][j]

IndexError: list assignment index out of range


In [7]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [8]:
```



```
r1=[1,2,3,4]
r2=[4,5,6,7]
r3=[7,8,9,14]
r4=[1,1,2,2]

s1=[1,2,1,2]
s2=[6,2,3,15]
s3=[4,2,1,45]
s4=[1,7,2,9]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)
A.append(r4)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)
B.append(s4)

print(A)
print(B)

#I need to add A and B

C=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]


for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]+B[i][j]

print(C)
```

```
  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, in <module>
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 34, in <module>
    C[i][j]=A[i][j]+B[i][j]

IndexError: list assignment index out of range


In [7]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [8]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3, 4], [4, 5, 6, 7], [7, 8, 9, 14], [1, 1, 2, 2]]
[[1, 2, 1, 2], [6, 2, 3, 15], [4, 2, 1, 45], [1, 7, 2, 9]]
[[2, 4, 4, 6], [10, 7, 9, 22], [11, 10, 10, 59], [2, 8, 4, 11]]

In [9]:
```



```
r1=[1,2,3,4]
r2=[4,5,6,7]
r3=[7,8,9,14]
r4=[1,1,2,2]

s1=[1,2,1,2]
s2=[6,2,3,15]
s3=[4,2,1,45]
s4=[1,7,2,9]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)
A.append(r4)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)
B.append(s4)

print(A)
print(B)

#I need to add A and B

C=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]


for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]*B[i][j]
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 34, in <module>
    C[i][j]=A[i][j]+B[i][j]

IndexError: list assignment index out of range


In [7]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [8]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3, 4], [4, 5, 6, 7], [7, 8, 9, 14], [1, 1, 2, 2]]
[[1, 2, 1, 2], [6, 2, 3, 15], [4, 2, 1, 45], [1, 7, 2, 9]]
[[2, 4, 4, 6], [10, 7, 9, 22], [11, 10, 10, 59], [2, 8, 4, 11]]

In [9]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[1, 2, 3, 4], [4, 5, 6, 7], [7, 8, 9, 14], [1, 1, 2, 2]]
[[1, 2, 1, 2], [6, 2, 3, 15], [4, 2, 1, 45], [1, 7, 2, 9]]
[[1, 4, 3, 8], [24, 10, 18, 105], [28, 16, 9, 630], [1, 7, 4, 18]]

In [10]:
```

Now, you did this for 3 dimension what if you want to do it for 4 dimension, change this to 4 and add more numbers here. Let us say I will add more numbers here, let us say 14 here and then maybe 2 here and then 16, 15 here, there may be 45 here, whatever, whatever and as you can see I am appending so this may require another row. So, I will say r4 is equal to some 1, 1, 2, 2, so I am trying to write a 4 cross 4 matrix you see and then 1, 7, 2, 9 and then I may have to append this as well here, you see, a lot of effort here in changing everything, you may want to think if this can be automated.

So, I print A and B and then C again should be one more row and 4 cross 4 with entries 0, so I leave it to you to tell me how you can automate all these things and not make it such a big story of manually doing everything. I leave it to you as homework there. And then pretty much we have done it, let me see whether this gives me the answer, this is indeed giving me the answer, 7 plus 15 is 22, so on and so forth.

So, now what I will do is I will comfortably come here and then put a into here and this results in matrix multiplication 1 into 1 is 1, 2 into 2 is 4, 3 into 1 is 3, 4 into 2 is 8 and so on and so forth, and I know all of you have your eyebrows up, thinking oh my goodness our instructor does not know what is matrix multiplication, yes, I really do not know why we cannot multiply matrices like this just the way we add, life would have been easier if matrix multiplication was simply component wise multiplication just the way we add two matrices.

But why on earth is matrix multiplication different? Why is it different from the usual multiplication? So, you may want to know that, that probably is the crux, if there is one nucleus for your data sciences program, BSc in data sciences, it is in fact the idea of matrices and in matrices a very elegant idea is indeed matrix multiplication and why do we multiply the row with column and then put one value there?

In fact, we take dot products you see to multiply matrices, we do not multiply like this, do we? We do not, that is not how it was taught in our high schools. Why do we multiply like that? Food for thought for all of you, think about it, there is a lot of online references for you, anyways the motive of the course is not to teach matrices but to tell you how to meddle with matrices.

So, what I will do right now is I will go ahead and write a piece of code to multiply two matrices, I hope you liked the code to add to matrices the naive way, the first principles way, the long way the sort of a very, very what I mean unintelligent way of writing a code because most of these things can be automated. I leave it to you to automate, in fact the forthcoming weeks we will be taking all this code and trying to write better code out of them by using better ideas in Python.