Bhuvan Jammalamadaka

Professor Nitin Sanket

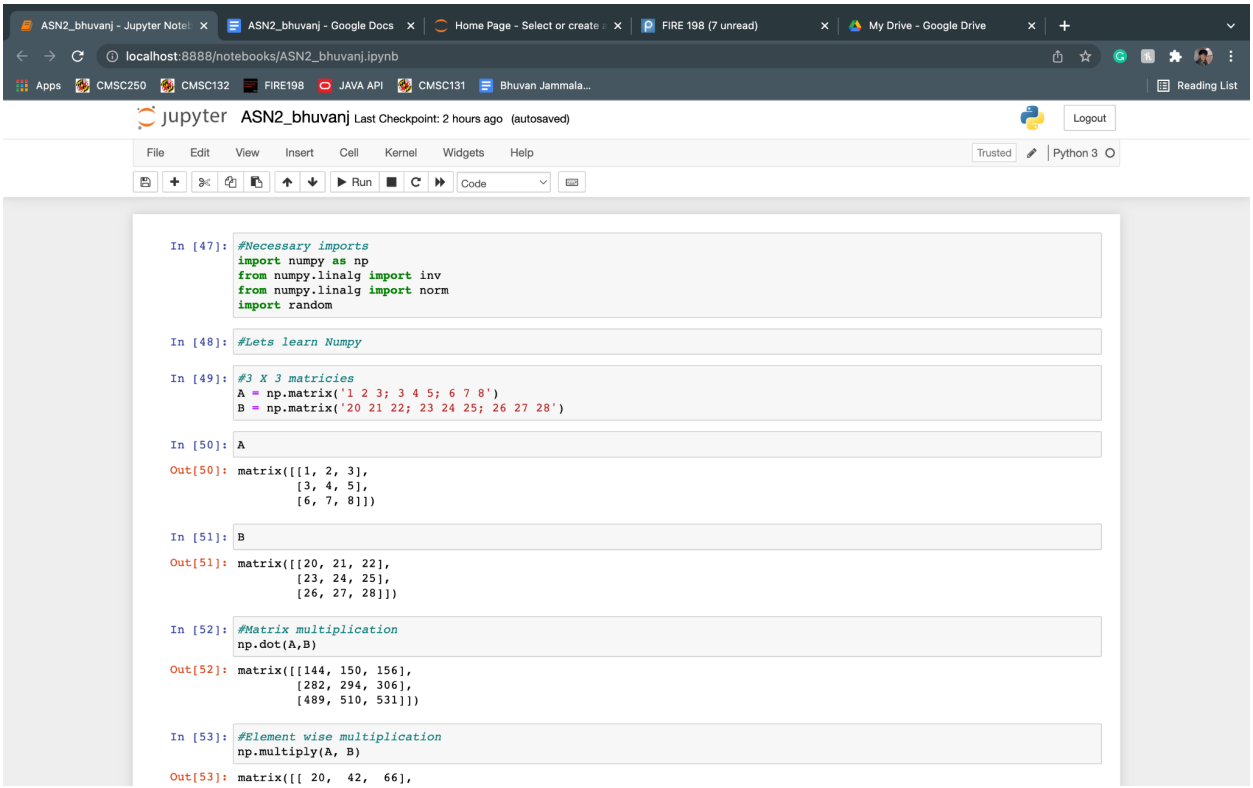
FIRE198 AUS

14 February 2022

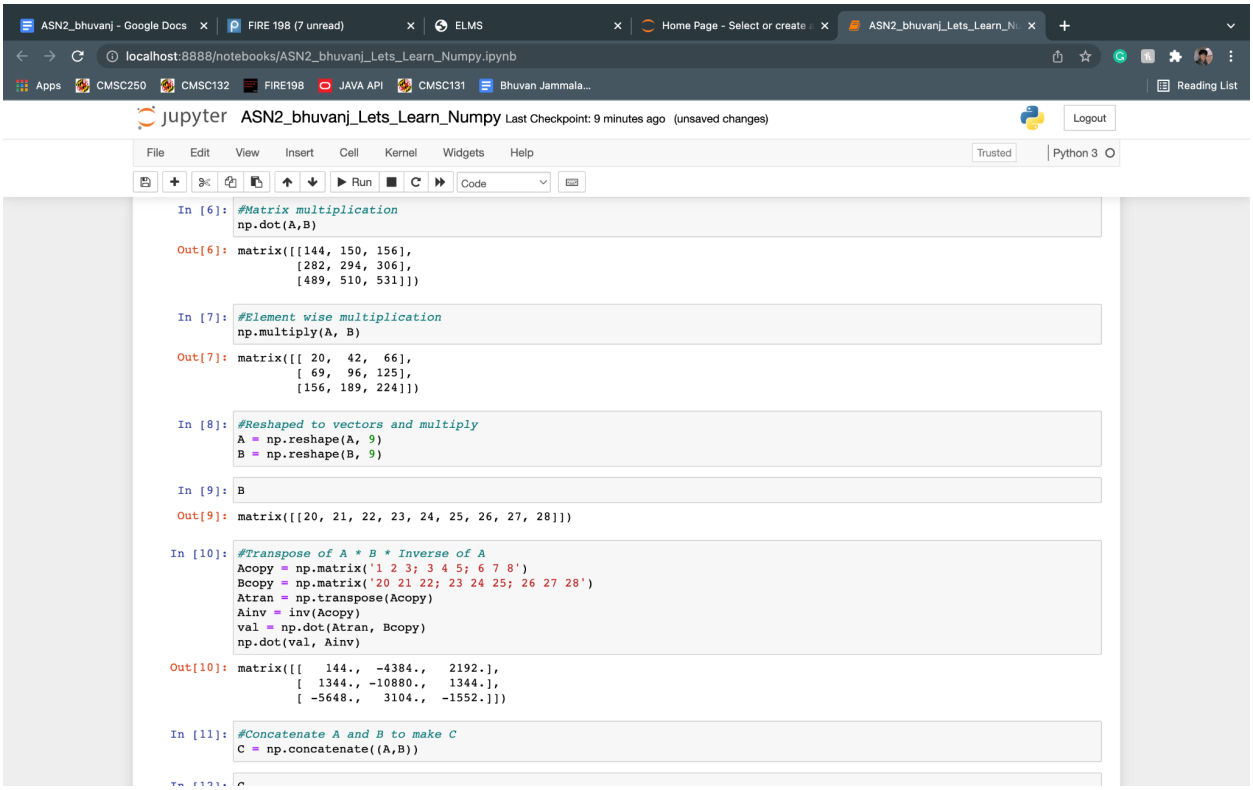
ASN2: Let's Become Pythonic

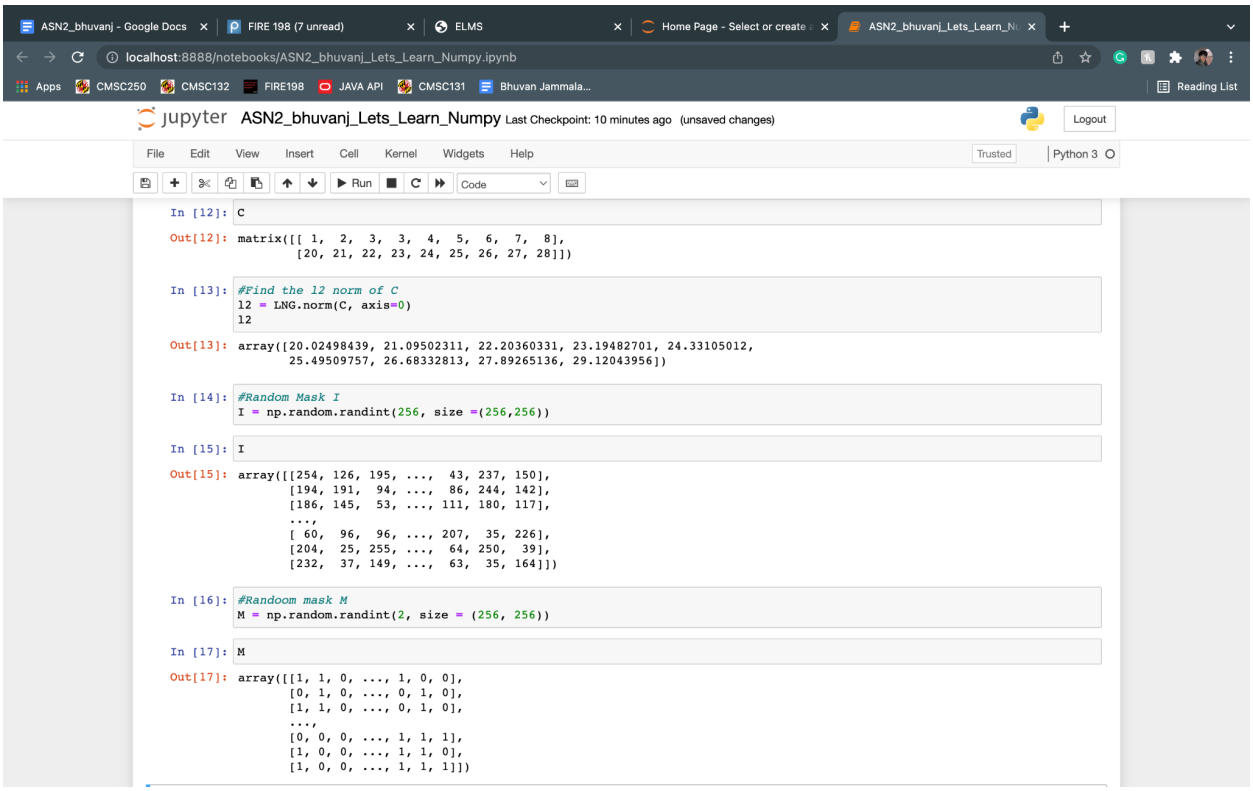
Why do you need __name__ == "__main__" in a python script?

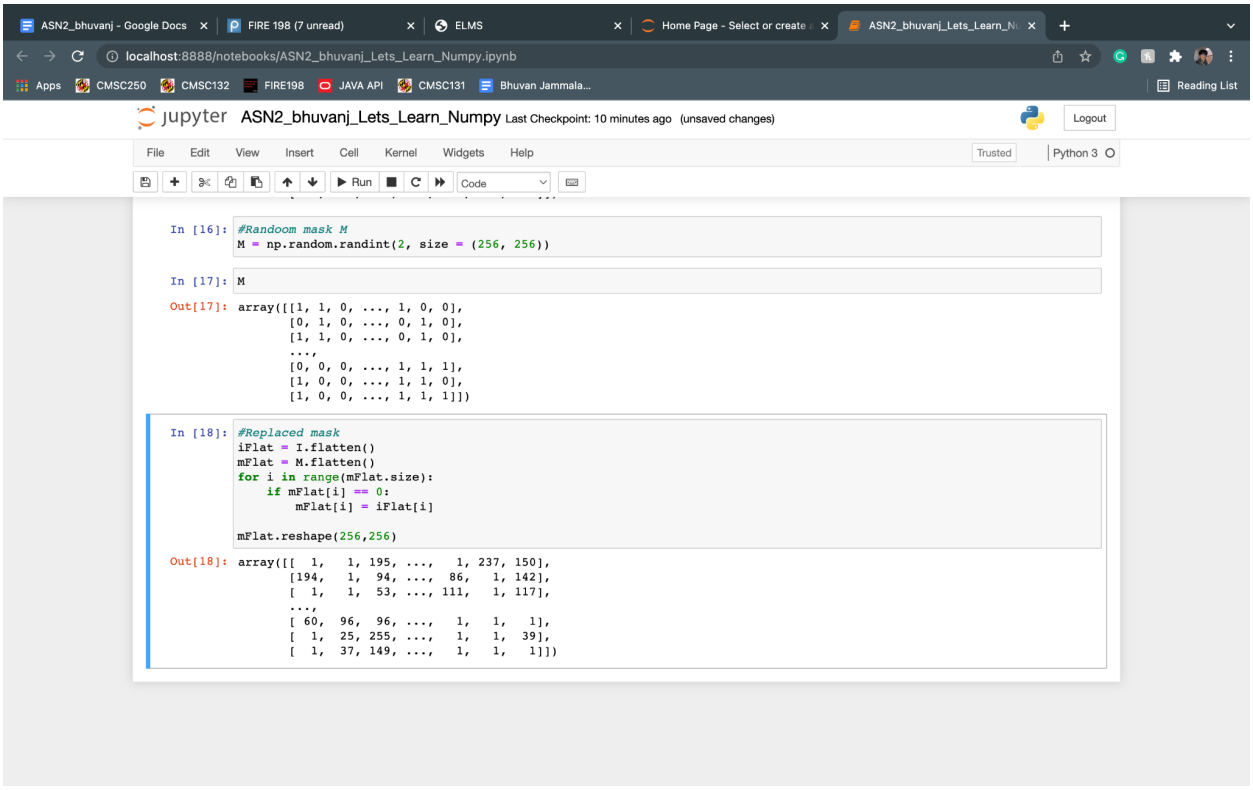
If the module is being run in the main program then the name will be set to "__main__". However, if code that is being imported from another module is being run, then the name of that module will be set to the "__name__" variable. A module in python can be viewed as a code library or a file that contains a set of functions, classes and variables.



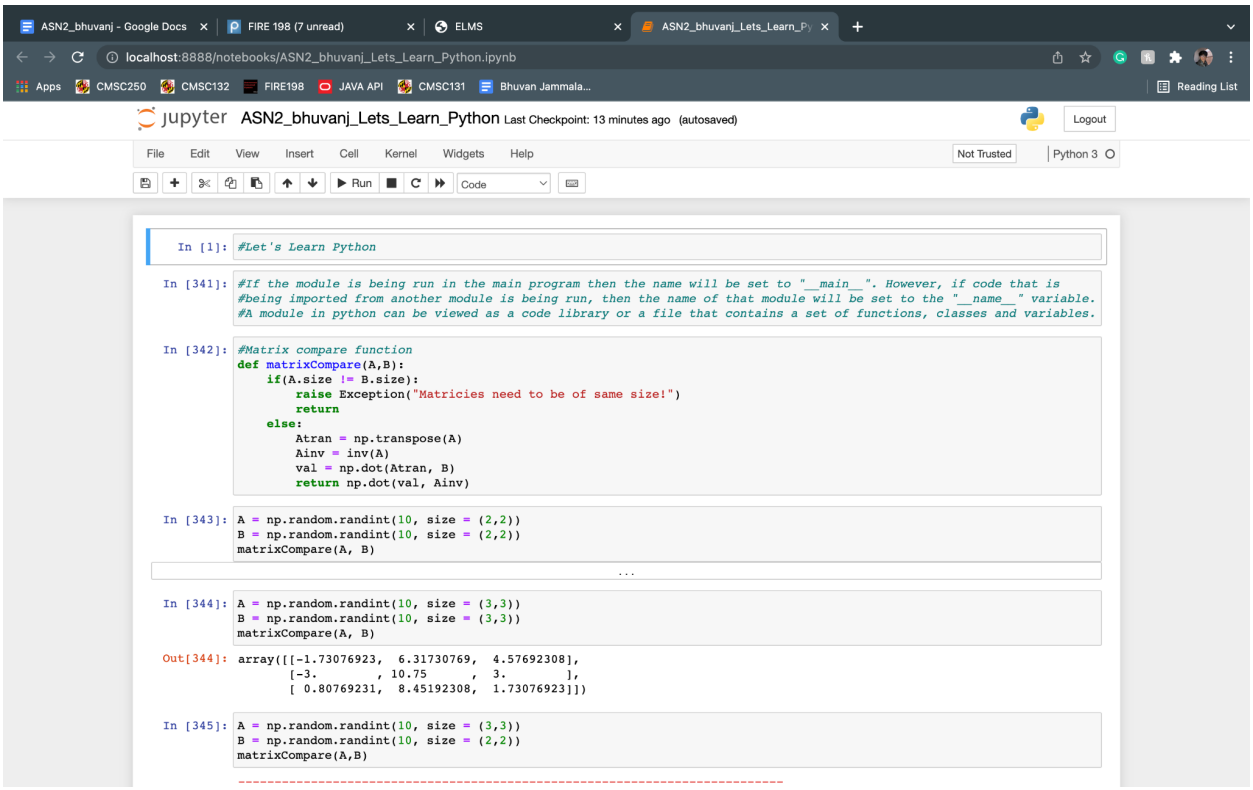
Outputs for part 1:

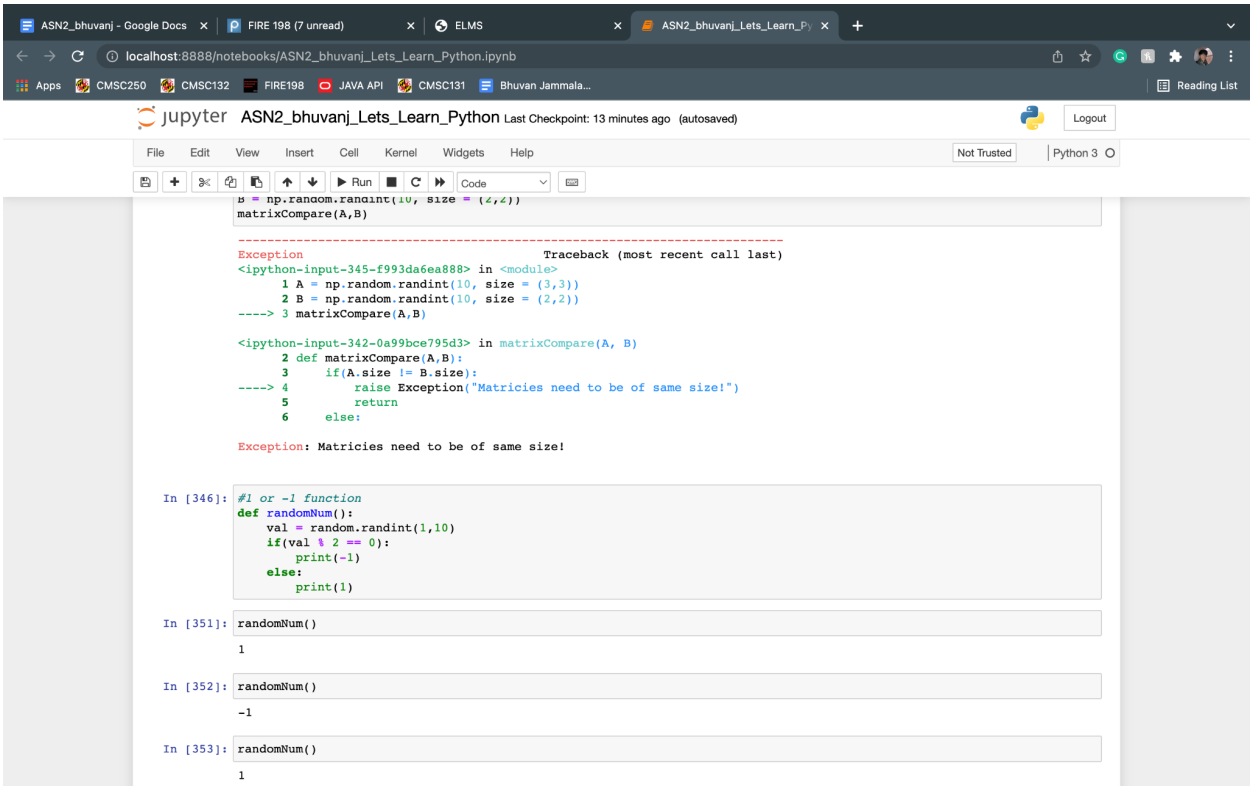


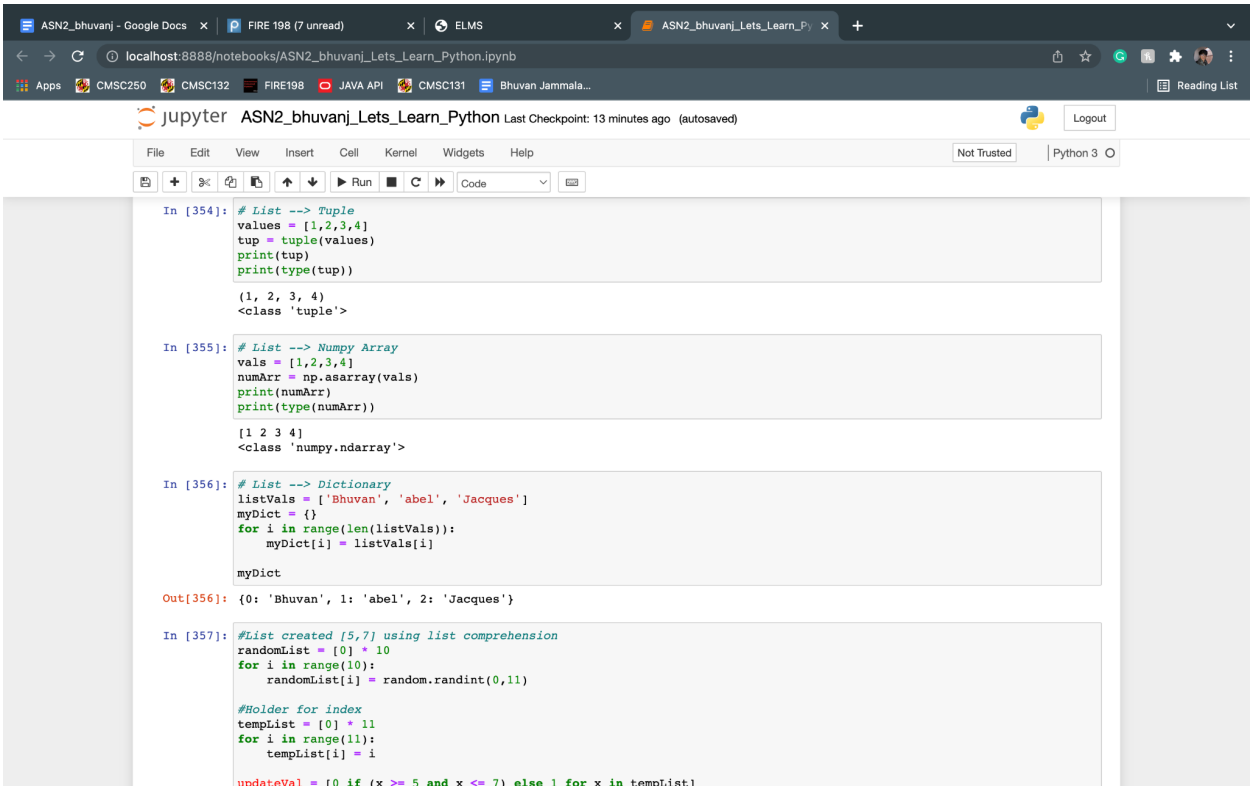


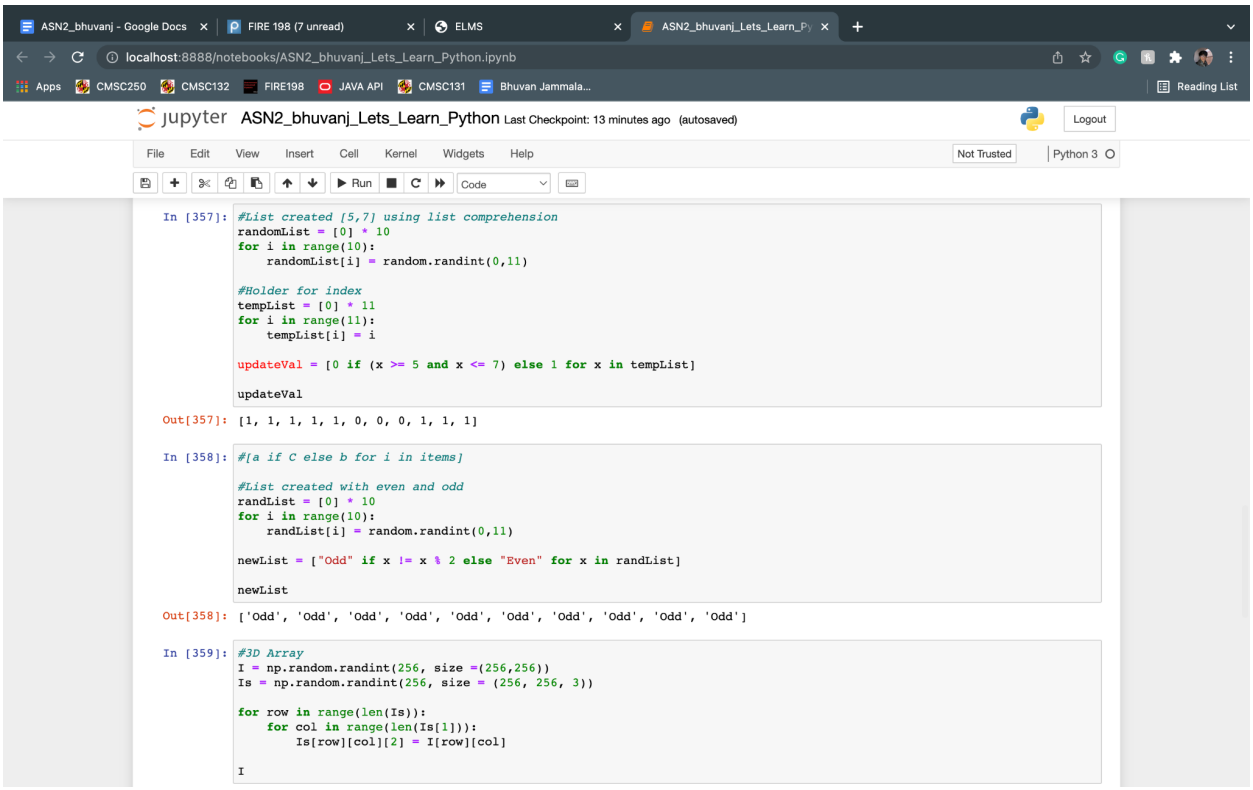


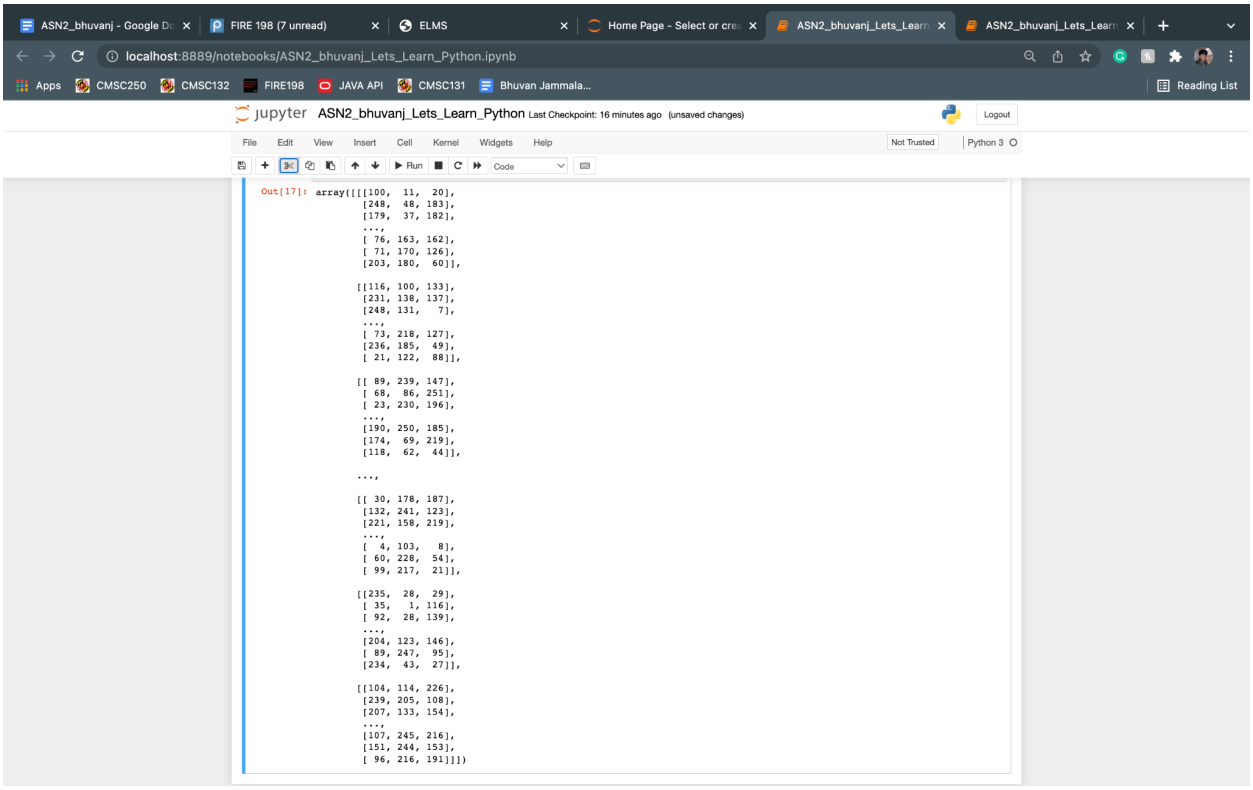
Outputs for part 2:


ASN2_bhuvanj - Google Docs
FIRE 198 (7 unread)
ELMS
ASN2_bhuvanj_Lets_Learn_Py
localhost:8888/notebooks/ASN2_bhuvanj_Lets_Learn_Python.ipynb
Apps
CMSC250
CMSC132
FIRE198
JAVA API
CMSC131
Bhuvan Jammala...
Reading List
jupyter ASN2_bhuvanj_Lets_Learn_Python Last Checkpoint: 13 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [1]: #Let's Learn Python
In [341]: #If the module is being run in the main program then the name will be set to "__main__". However, if code that is
#being imported from another module is being run, then the name of that module will be set to the "__name__" variable.
#A module in python can be viewed as a code library or a file that contains a set of functions, classes and variables.
In [342]: #Matrix compare function
def matrixCompare(A,B):
if(A.size != B.size):
raise Exception("Matricies need to be of same size!")
return
else:
Atran = np.transpose(A)
Ainv = inv(A)
val = np.dot(Atran, B)
return np.dot(val, Ainv)
In [343]: A = np.random.randint(10, size = (2,2))
B = np.random.randint(10, size = (2,2))
matrixCompare(A, B)
...
In [344]: A = np.random.randint(10, size = (3,3))
B = np.random.randint(10, size = (3,3))
matrixCompare(A, B)
Out[344]: array([[-1.73076923, 6.31730769, 4.57692308],
[-3. , 10.75 , 3.],
[0.80769231, 8.45192308, 1.73076923]])
In [345]: A = np.random.randint(10, size = (3,3))
B = np.random.randint(10, size = (2,2))
matrixCompare(A,B)


ASN2_bhuvanj - Google Docs
FIRE 198 (7 unread)
ELMS
ASN2_bhuvanj_Lets_Learn_Py
localhost:8888/notebooks/ASN2_bhuvanj_Lets_Learn_Python.ipynb
Apps
CMSC250
CMSC132
FIRE198
JAVA API
CMSC131
Bhuvan Jammala...
Reading List
jupyter ASN2_bhuvanj_Lets_Learn_Python Last Checkpoint: 13 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
matrixCompare(A,B)
Exception Traceback (most recent call last)
<ipython-input-345-f993da6ea888> in <module>
1 A = np.random.randint(10, size = (3,3))
2 B = np.random.randint(10, size = (2,2))
----> 3 matrixCompare(A,B)
<ipython-input-342-0a99bce795d3> in matrixCompare(A, B)
2 def matrixCompare(A,B):
3 if(A.size != B.size):
----> 4 raise Exception("Matricies need to be of same size!")
5 return
6 else:
Exception: Matricies need to be of same size!
In [346]: #1 or -1 function
def randomNum():
val = random.randint(1,10)
if(val % 2 == 0):
print(-1)
else:
print(1)
In [351]: randomNum()
1
In [352]: randomNum()
-1
In [353]: randomNum()
1


ASN2_bhuvanj - Google Docs
FIRE 198 (7 unread)
ELMS
ASN2_bhuvanj_Lets_Learn_Py
localhost:8888/notebooks/ASN2_bhuvanj_Lets_Learn_Python.ipynb
Apps
CMSC250
CMSC132
FIRE198
JAVA API
CMSC131
Bhuvan Jammala...
Reading List
jupyter
ASN2_bhuvanj_Lets_Learn_Python
Last Checkpoint: 13 minutes ago
(autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3
Run
Code
In [354]:
# List --> Tuple
values = [1,2,3,4]
tup = tuple(values)
print(tup)
print(type(tup))
(1, 2, 3, 4)
<class 'tuple'>
In [355]:
# List --> Numpy Array
vals = [1,2,3,4]
numArr = np.asarray(vals)
print(numArr)
print(type(numArr))
[1 2 3 4]
<class 'numpy.ndarray'>
In [356]:
# List --> Dictionary
listVals = ['Bhuvan', 'abel', 'Jacques']
myDict = {}
for i in range(len(listVals)):
myDict[i] = listVals[i]
myDict
Out[356]:
{0: 'Bhuvan', 1: 'abel', 2: 'Jacques'}
In [357]:
#List created [5,7] using list comprehension
randomList = [0] * 10
for i in range(10):
randomList[i] = random.randint(0,11)
#Holder for index
tempList = [0] * 11
for i in range(11):
tempList[i] = i


ASN2_bhuvanj - Google Docs
FIRE 198 (7 unread)
ELMS
ASN2_bhuvanj_Lets_Learn_Py
localhost:8888/notebooks/ASN2_bhuvanj_Lets_Learn_Python.ipynb
Apps
CMSC250
CMSC132
FIRE198
JAVA API
CMSC131
Bhuvan Jammala...
Reading List
jupyter ASN2_bhuvanj_Lets_Learn_Python Last Checkpoint: 13 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [357]:
#List created [5,7] using list comprehension
randomList = [0] * 10
for i in range(10):
randomList[i] = random.randint(0,11)
#Holder for index
tempList = [0] * 11
for i in range(11):
tempList[i] = i
updateVal = [0 if (x >= 5 and x <= 7) else 1 for x in tempList]
updateVal
Out[357]: [1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
In [358]:
#[a if C else b for i in items]
#List created with even and odd
randList = [0] * 10
for i in range(10):
randList[i] = random.randint(0,11)
newList = ["Odd" if x != x % 2 else "Even" for x in randList]
newList
Out[358]: ['Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd', 'Odd']
In [359]:
#3D Array
I = np.random.randint(256, size =(256,256))
Is = np.random.randint(256, size = (256, 256, 3))
for row in range(len(Is)):
for col in range(len(Is[1])):
Is[row][col][2] = I[row][col]
I


ASN2_bhuvanj - Google Do
FIRE 198 (7 unread)
ELMS
Home Page - Select or cre
ASN2_bhuvanj_Lets_Learn
ASN2_bhuvanj_Lets_Learn
localhost:8889/notebooks/ASN2_bhuvanj_Lets_Learn_Python.ipynb
Apps
CMSC250
CMSC132
FIRE198
JAVA API
CMSC131
Bhuvan Jammala...
Reading List
jupyter ASN2_bhuvanj_Lets_Learn_Python Last Checkpoint: 16 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Out[17]: array([[[100, 11, 20],
[248, 48, 183],
[179, 37, 182],
...,
[76, 163, 162],
[71, 170, 126],
[203, 180, 60]],
[[116, 100, 133],
[231, 138, 137],
[248, 131, 7],
...,
[73, 218, 127],
[236, 185, 49],
[21, 122, 88]],
[[89, 239, 147],
[68, 86, 251],
[23, 230, 196],
...,
[190, 250, 185],
[174, 69, 219],
[118, 62, 44]],
...,
[[30, 178, 187],
[132, 241, 123],
[221, 158, 219],
...,
[4, 103, 8],
[60, 228, 54],
[99, 217, 21]],
[[235, 28, 29],
[35, 1, 116],
[92, 28, 139],
...,
[204, 123, 146],
[89, 247, 95],
[234, 43, 27]],
[[104, 114, 226],
[239, 205, 108],
[207, 133, 154],
...,
[107, 245, 216],
[151, 244, 153],
[96, 216, 191]]])

1.) I decided to go with the Jupyter Notebook IDE because I have worked with pandas, numpy, matplotlib, and seaborn with jupyter notebooks. Because I have experience with the IDE I decided to go with something I am comfortable with.

4.) Some challenges that I faced with this assignment was relearning a lot of numpy commands that I had previously forgotten. Using stackoverflow and W3 schools helped a lot because they have detailed answers for almost every challenge that I ran into.

5.) I think this assignment was really cool because it was nice getting a refresher of topics that I had previously worked on. It was also very interesting to see the intersection between linear algebra and computer science. There were aspects of the lecture that were implemented in the assignment such as the RGB color scale is 0-256 in 3 dimensions. I am excited to see how we are going to be implementing the concepts we learnt in this assignment to the AUS.