

Bhuvan Jammalamadaka

Professor Nitin Sanket

FIRE198 AUS

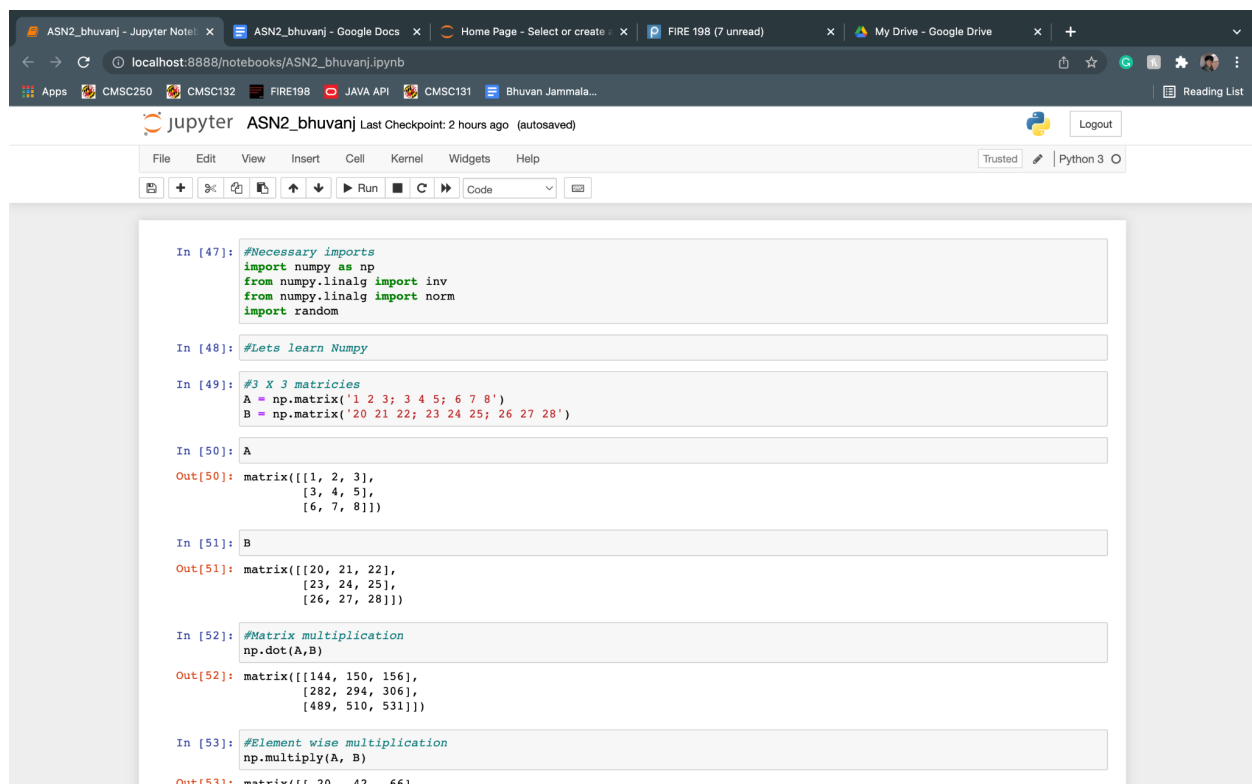
14 February 2022

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.



The screenshot shows a Jupyter Notebook interface with a browser window at the top. The notebook is titled "ASN2_bhuvan" and is running on a local host. The code in the notebook is as follows:

```
In [47]: #Necessary imports
import numpy as np
from numpy.linalg import inv
from numpy.linalg import norm
import random

In [48]: #Lets learn Numpy

In [49]: #3 X 3 matrices
A = np.matrix('1 2 3; 3 4 5; 6 7 8')
B = np.matrix('20 21 22; 23 24 25; 26 27 28')

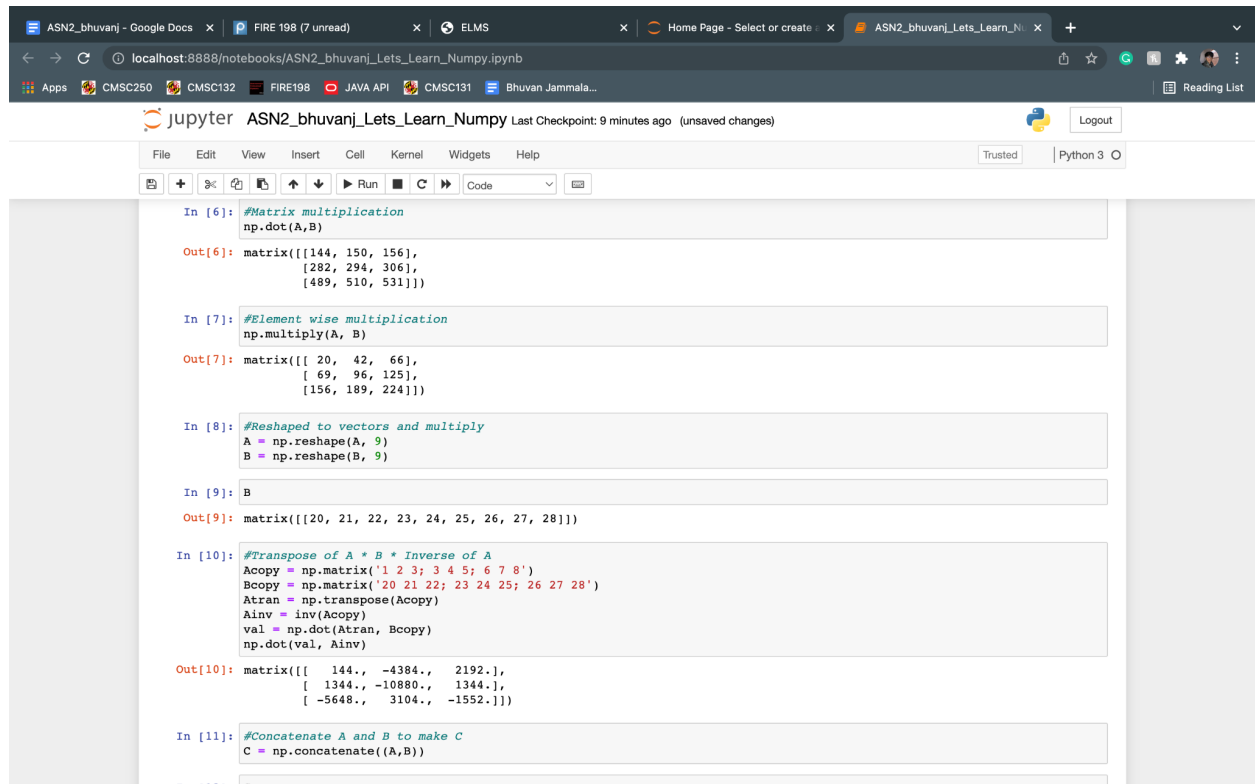
In [50]: A
Out[50]: matrix([[1, 2, 3],
                [3, 4, 5],
                [6, 7, 8]])

In [51]: B
Out[51]: matrix([[20, 21, 22],
                [23, 24, 25],
                [26, 27, 28]])

In [52]: #Matrix multiplication
np.dot(A,B)
Out[52]: matrix([[144, 150, 156],
                [282, 294, 306],
                [489, 510, 531]])

In [53]: #Element wise multiplication
np.multiply(A, B)
Out[53]: matrix([[ 20, 42, 66],
```

Outputs for part 1:



The screenshot displays a Jupyter Notebook interface with a browser window at the top. The browser tabs include 'ASN2_bhuvanaj - Google Docs', 'FIRE 198 (7 unread)', 'ELMS', 'Home Page - Select or create', and 'ASN2_bhuvanaj_Lets_Learn_Numpy'. The address bar shows 'localhost:8888/notebooks/ASN2_bhuvanaj_Lets_Learn_Numpy.ipynb'. The Jupyter Notebook interface has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, code execution, and output viewing. The notebook content shows several code cells and their outputs:

```
In [6]: #Matrix multiplication
np.dot(A,B)

Out[6]: matrix([[144, 150, 156],
               [282, 294, 306],
               [489, 510, 531]])

In [7]: #Element wise multiplication
np.multiply(A, B)

Out[7]: matrix([[ 20,  42,  66],
               [ 69,  96, 125],
               [156, 189, 224]])

In [8]: #Reshaped to vectors and multiply
A = np.reshape(A, 9)
B = np.reshape(B, 9)

In [9]: B

Out[9]: matrix([[20, 21, 22, 23, 24, 25, 26, 27, 28]])

In [10]: #Transpose of A * B * Inverse of A
Acopy = np.matrix('1 2 3; 3 4 5; 6 7 8')
Bcopy = np.matrix('20 21 22; 23 24 25; 26 27 28')
Atran = np.transpose(Acopy)
Ainv = inv(Acopy)
val = np.dot(Atran, Bcopy)
np.dot(val, Ainv)

Out[10]: matrix([[ 144., -4384., 2192.],
                 [ 1344., -10880., 1344.],
                 [-5648., 3104., -1552.]])

In [11]: #Concatenate A and B to make C
C = np.concatenate((A,B))

In [12]: C
```

```
ASN2_bhuvanji - Google Docs x P FIRE 198 (7 unread) x ELMS x Home Page - Select or create x ASN2_bhuvanji_Lets_Learn_Numpy
localhost:8888/notebooks/ASN2_bhuvanji_Lets_Learn_Numpy.ipynb
Apps CMSC250 CMSC132 FIRE198 JAVA API CMSC131 Bhuvan Jammala... Reading List

jupyter ASN2_bhuvanji_Lets_Learn_Numpy Last Checkpoint: 10 minutes ago (unsaved changes) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted

In [12]: C
Out[12]: matrix([[ 1,  2,  3,  3,  4,  5,  6,  7,  8],
                [20, 21, 22, 23, 24, 25, 26, 27, 28]])

In [13]: #Find the L2 norm of C
         L2 = L2G.norm(C, axis=0)
         L2
Out[13]: array([20.02498439, 21.09502311, 22.20360331, 23.19482701, 24.33105012,
                25.49509757, 26.68332813, 27.89265136, 29.12043956])

In [14]: #Random Mask I
         I = np.random.randint(256, size =(256,256))

In [15]: I
Out[15]: array([[254, 126, 195, ..., 43, 237, 150],
                [194, 191, 94, ..., 86, 244, 142],
                [186, 145, 53, ..., 111, 180, 117],
                ...,
                [ 60, 96, 96, ..., 207, 35, 226],
                [204, 25, 255, ..., 64, 250, 39],
                [232, 37, 149, ..., 63, 35, 164]])

In [16]: #Random mask M
         M = np.random.randint(2, size =(256, 256))

In [17]: M
Out[17]: array([[1, 1, 0, ..., 1, 0, 0],
                [0, 1, 0, ..., 0, 1, 0],
                [1, 1, 0, ..., 0, 1, 0],
                ...,
                [0, 0, 0, ..., 1, 1, 1],
                [1, 0, 0, ..., 1, 1, 0],
                [1, 0, 0, ..., 1, 1, 1]])
```

```
ASN2_bhuvanji - Google Docs x P FIRE 198 (7 unread) x ELMS x Home Page - Select or create x ASN2_bhuvanji_Lets_Learn_Numpy
localhost:8888/notebooks/ASN2_bhuvanji_Lets_Learn_Numpy.ipynb
Apps CMSC250 CMSC132 FIRE198 JAVA API CMSC131 Bhuvan Jammala... Reading List

jupyter ASN2_bhuvanji_Lets_Learn_Numpy Last Checkpoint: 10 minutes ago (unsaved changes) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted

In [16]: #Random mask M
         M = np.random.randint(2, size =(256, 256))

In [17]: M
Out[17]: array([[1, 1, 0, ..., 1, 0, 0],
                [0, 1, 0, ..., 0, 1, 0],
                [1, 1, 0, ..., 0, 1, 0],
                ...,
                [0, 0, 0, ..., 1, 1, 1],
                [1, 0, 0, ..., 1, 1, 0],
                [1, 0, 0, ..., 1, 1, 1]])

In [18]: #Replaced mask
         iFlat = I.flatten()
         mFlat = M.flatten()
         for i in range(mFlat.size):
             if mFlat[i] == 0:
                 mFlat[i] = iFlat[i]
         mFlat.reshape(256,256)
Out[18]: array([[ 1,  1, 195, ...,  1, 237, 150],
                [194,  1, 94, ..., 86,  1, 142],
                [ 1,  1, 53, ..., 111,  1, 117],
                ...,
                [ 60, 96, 96, ...,  1,  1,  1],
                [ 1, 25, 255, ...,  1,  1, 39],
                [ 1, 37, 149, ...,  1,  1,  1]])
```

Outputs for part 2:

AS2_bhuvanji Lets_Learn_Python Last Checkpoint: 13 minutes ago (autosaved)

```
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("Matrices need to be of same size!")
    return
    else:
        Atran = np.transpose(A)
        Ainvs = inv(A)
        val = np.dot(Atran, B)
        return np.dot(val, Ainvs)

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)
```

AS2_bhuvanji Lets_Learn_Python Last Checkpoint: 13 minutes ago (autosaved)

```
B = np.random.randint(10, size = (2,2))
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("Matrices need to be of same size!")
      5     return
      6     else:

Exception: Matrices 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
```

```
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

updateList = [0,15]
for i in range(5) and i < 7:
    else:
        for u in tempList:
```

ASN2_bhuvanaj - Google Docs x FIRE 198 (7 unread) x ELMS x ASN2_bhuvanaj_Lets_Learn_Py x +

localhost:8888/notebooks/ASN2_bhuvanaj_Lets_Learn_Python.ipynb

Apps CMSC250 CMSC132 FIRE198 JAVA API CMSC131 Bhuvan Jammala... Reading List

jupyter ASN2_bhuvanaj_Lets_Learn_Python Last Checkpoint: 13 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
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
templList = [0] * 11
for i in range(11):
    templList[i] = i

updateVal = [0 if (x >= 5 and x <= 7) else 1 for x in templList]

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_bhuvanaj - Google Docs x FIRE 198 (7 unread) x ELMS x Home Page - Select or create a new notebook x ASN2_bhuvanaj_Lets_Learn_Python x ASN2_bhuvanaj_Lets_Learn_Python x +

localhost:8889/notebooks/ASN2_bhuvanaj_Lets_Learn_Python.ipynb

Apps CMSC250 CMSC132 FIRE198 JAVA API CMSC131 Bhuvan Jammala... Reading List

jupyter ASN2_bhuvanaj_Lets_Learn_Python Last Checkpoint: 16 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
Out[17]: array([[1100, 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, 71],
 ...,
 [ 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.