NUMPY INTERVIEW QUESTION PRACTICE

- Practrice for inte=erview purpose
- any doubt ask me

1. Multiplication of matrix using numpy

```
import numpy as np
```

```
a=np.arange(2,11)
print(a)
```

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

```
b=np.arange(1,10)
b
```

```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
c=a.reshape(3,3) # change the shape of matrx 1d to 2d
d=b.reshape(3,3)
print(c)
print(d)
```

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

```
c*d # element wise multiplcation
```

```
array([[ 2,  6, 12],
       [20, 30, 42],
       [56, 72, 90]])
```

```
# but want want to multiply marix use dot product
c.dot(d)
```

```
array([[ 42,  51,  60],
       [ 78,  96, 114],
       [114, 141, 168]])
```

```
d.dot(c)
```

```
array([[ 36,  42,  48],
       [ 81,  96, 111],
       [126, 150, 174]])
```

2. to change string elelment in specific numpy array to uppercase ,lowecase, captialise first letters ,title cae or swapcase

```
#uppercase
s="this is numpy Interview important Question SERIES"
print(s)
```

```
this is numpy Interview important Question SERIES
```

```
my_arr=np.array(s)
print(my_arr)
```

```
this is numpy Interview important Question SERIES
```

```
type(my_arr)
```

```
numpy.ndarray
```

```
u=np.char.upper(my_arr) # char.upper(array)
print(u)
```

```
THIS IS NUMPY INTERVIEW IMPORTANT QUESTION SERIES
```

```
#lowercase array
l=np.char.lower(my_arr)
print(l)
```

```
     this is numpy interview important question series
```

```
# capitalise first letter
c=np.char.capitalize(my_arr)
print(c)
print(my_arr)
```

```
     This is numpy interview important question series
     this is numpy Interview important Question SERIES
```

```
# tilte case
ti=np.char.title(my_arr)
print(ti)
```

```
     This Is Numpy Interview Important Question Series
```

```
#swapcase() change upper to lower or lower to upper
s=np.char.swapcase(my_arr)
print(s)
```

```
     THIS IS NUMPY iNTERVIEW IMPORTANT qUESTION series
```

3. filter a row of numpy array .HOW TO GET ROW NUMBER OF NUMPY ARRAY

```
ar=np.array([[1,2,3,4,5],[10,-3,40,32,9],[3,6,2,8,5]])
ar
```

```
     array([[ 1,  2,  3,  4,  5],
            [10, -3, 40, 32,  9],
            [ 3,  6,  2,  8,  5]])
```

```
#??????
```

4.How to remove rows which contains null values in numpy array?

```
arr=np.array([[10,5,2.5,4],
              [23,44,11,34],
              [78,34,np.nan,23],
              [23,22,45,np.nan]])
arr
```

```
     array([[10. ,  5. ,  2.5,  4. ],
            [23. , 44. , 11. , 34. ],
            [78. , 34. ,  nan, 23. ],
            [23. , 22. , 45. ,  nan]])
```

```
arr.shape
```

```
     (4, 4)
```

```
np.isnan(arr) # check retunr boolenan value if null vlaues nan
```

```
     array([[False, False, False, False],
            [False, False, False, False],
            [False, False,  True, False],
            [False, False, False,  True]])
```

```
np.isnan(arr).any(axis=1) # 3rd row and 4 th row having null values
```

```
     array([False, False,  True,  True])
```

```
arr[~np.isnan(arr).any(axis=1)] # select those rows which has no null values ,remove null vlues rows
```

```
     array([[10. ,  5. ,  2.5,  4. ],
            [23. , 44. , 11. , 34. ]])
```

```
arr[np.isnan(arr).any(axis=1)]
```

```
     array([[78., 34., nan, 23.],
            [23., 22., 45., nan]])
```

5. How to multiply string elemets within numpy array?

```
s=4*"mango"
s
```

```
'mangomangomangomango'
```

```
myar=np.array(["ash",'ghi',"kash",'dfg'])
myar
```

```
array(['ash', 'ghi', 'kash', 'dfg'], dtype='<U4')
```

```
mular=np.char.multiply(myar,2)
mular # multiply 2 times each string element
```

```
array(['ashash', 'ghighi', 'kashkash', 'dfgdfg'], dtype='<U8')
```

6.Find a minimum value iwth numpy array ?

```
a=np.arange(1,20)
a
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19])
```

```
np.min(a)
```

```
1
```

```
np.max(a)
```

```
19
```

```
b=np.arange(1,10).reshape(3,3)
b
```

```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
# minimum row elelmet row
```

```
np.apply_along_axis(lambda x:np.min(x),arr=b,axis=0)
```

```
array([1, 2, 3])
```

```
np.min(b)
```

```
1
```

7.Find diagonal value in numpy array?

8.Reshaping and crerate new array?

```
import numpy as np
```

```
ar=np.arange(16)
ar
```

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15])
```

```
len(ar)
```

```
16
```

```
# convert 1d ionto 4*4 2d array
ar.reshape(4,4)
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

```
ar.reshape(2,8)
```

```
array([[ 0,  1,  2,  3,  4,  5,  6,  7],
       [ 8,  9, 10, 11, 12, 13, 14, 15]])
```

```
a=ar.reshape(4,4)
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

```
# last rows first ,change the oreer of all rows second become3  change reverse
newb=a[::-1]
newb
```

```
array([[12, 13, 14, 15],
       [ 8,  9, 10, 11],
       [ 4,  5,  6,  7],
       [ 0,  1,  2,  3]])
```

9.How to padding in numpy array?

- how to border

```
# padding for 2d array/ matrix
ar=np.ones((2,2))
ar
```

```
array([[1., 1.],
       [1., 1.]])
```

```
# add padding
np.pad(ar,pad_width=1,mode="constant",constant_values=0)
```

```
array([[0., 0., 0., 0.],
       [0., 1., 1., 0.],
       [0., 1., 1., 0.],
       [0., 0., 0., 0.]])
```

```
np.pad(ar,pad_width=1,mode="constant",constant_values=2)
```

```
array([[2., 2., 2., 2.],
       [2., 1., 1., 2.],
       [2., 1., 1., 2.],
       [2., 2., 2., 2.]])
```

10. Calculate sin cosine values of numbers in numpy array?

```
ar1=np.array([0,30,60,90,180])
ar1
```

```
array([  0,  30,  60,  90, 180])
```

```
np.sin(ar1) # it converts values in radians
```

```
array([ 0.        , -0.98803162, -0.30481062,  0.89399666, -0.80115264])
```

```
# so convert array into radina form ar1*pi/180

newar1=ar1*np.pi/180
newar1
```

```
array([0.        , 0.52359878, 1.04719755, 1.57079633, 3.14159265])
```

```
np.sin(newar1)
```

```
array([0.00000000e+00, 5.00000000e-01, 8.66025404e-01, 1.00000000e+00,
       1.22464680e-16])
```

```
np.cos(newar1)
```

```
array([ 1.00000000e+00,  8.66025404e-01,  5.00000000e-01,  6.12323400e-17,
       -1.00000000e+00])
```

11. NUMPY SQUUEZE FUNCTIONS : REMOVE SINGLE DIMESION ENTRIES and check specific value present or not in numpy array ?

```
# change the dimensions of array using squeeze function
ar1=np.array([[[2,4,6],[6,7,8]]])
ar1
```

```
array([[[2, 4, 6],
        [6, 7, 8]]])
```

```
ar1.shape
```

```
(1, 2, 3)
```

```
# change shape or dimesion 3d to 2d
squeezear1=np.squeeze(ar1)
squeezear1
```

```
array([[2, 4, 6],
       [6, 7, 8]])
```

```
squeezear1.shape
```

```
(2, 3)
```

```
# check specific value present or not
3 in squeezear1
```

```
False
```

```
4 in squeezear1
```

```
True
```

12.Compare two numpy arrays?

```
a=np.array([121,23,22])
b=np.array([21,23,25])
print(a)
print(b)
```

```
[121  23  22]
[21 23 25]
```

```
a>b
```

```
array([ True, False, False])
```

```
a<b
```

```
array([False, False,  True])
```

```
a==b
```

```
array([False,  True, False])
```

```
a>=b
```

```
array([ True,  True, False])
```

```
a<=b
```

```
array([False,  True,  True])
```

```
np.greater_equal(a,b)
```

```
array([ True,  True, False])
```

```
np.less_equal(a,b)
```

```
array([False,  True,  True])
```

13.How to stack two numpy array?

- vewrtically and horizontally

```
a1=np.arange(10).reshape(2,5)
a1
```

```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

```
b1=np.arange(10,20).reshape(2,5)
b1
```

```
array([[10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

```
b1.shape
```

```
(2, 5)
```

```
np.concatenate([a1,b1],axis=0) # column wise merge
```

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

```
np.concatenate([a1,b1],axis=1) # row wise merge
```

```
array([[ 0,  1,  2,  3,  4, 10, 11, 12, 13, 14],
       [ 5,  6,  7,  8,  9, 15, 16, 17, 18, 19]])
```

```
# using hstack() #row wise
np.hstack([a1,b1])
```

```
array([[ 0,  1,  2,  3,  4, 10, 11, 12, 13, 14],
       [ 5,  6,  7,  8,  9, 15, 16, 17, 18, 19]])
```

```
np.vstack([a1,b1])#column wise verically
```

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14],
       [15, 16, 17, 18, 19]])
```

14.Put values in certain postion in numpy array ?

```
a=np.array([1,3,5,7,8,9])
a
```

```
array([1, 3, 5, 7, 8, 9])
```

```
b=np.array([2,8,4])
b
```

```
array([2, 8, 4])
```

```
a.put([1,4,5],b)
```

```
a
```

```
array([1, 2, 5, 7, 8, 4])
```

15.Delete the cerain column or second and insert new column in its place

```
a1=np.array([[1,3,4],[4,6,7],[6,7,8]])
b1=np.array([[2,4,5],[5,6,7]])
```

```
print(a1)
print(b1)
```

```
[[1 3 4]
 [4 6 7]
 [6 7 8]]
[[2 4 5]
 [5 6 7]]
```

```
#delete column
d=np.delete(a1,1,axis=1) # 2nd column deleted
d
```

```
array([[1, 4],
       [4, 7],
       [6, 8]])
```

```
column=np.array([10,20,30])
```

```
#insert new column
np.insert(d,1,column,axis=1)
```

```
array([[ 1, 10,  4],
       [ 4, 20,  7],
       [ 6, 30,  8]])
```

16.Create numpy arrays of 10 zeros,10 ones and 10 fives?

```
a=np.zeros(10)
a
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
b=np.ones(10)
b
```

```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

```
len(b)
```

```
10
```

17. Create random sequence in numpy array ?

```
a=np.array([1,2,3])
a
```

```
array([1, 2, 3])
```

```
b=np.repeat(a,2) # repeat two times each elelment in array
b
```

```
array([1, 1, 2, 2, 3, 3])
```

```
d=np.tile(a,2)# generate same squence two times
d
```

```
array([1, 2, 3, 1, 2, 3])
```

```
e=np.concatenate([b,d])
e
```

```
array([1, 1, 2, 2, 3, 3, 1, 2, 3, 1, 2, 3])
```

```
e.shape
```

```
(12,)
```

```
f=np.random.random(10)
f
```

```
array([0.90498186, 0.54846781, 0.42439939, 0.85138198, 0.68419838,
       0.42850532, 0.02947922, 0.98842686, 0.12805773, 0.92318913])
```

```
f=np.random.randint(1,30,5)
f
```

```
array([ 2,  2, 12,  6,  3])
```

18.Determine memory size of numpy array?

```
g=np.array([1,4,8,60])
g
```

```
array([ 1,  4,  8, 60])
```

```
g.itemsize
```

```
8
```

```
g.size
```

```
4
```

```
g.itemsize*g.size
```

```
32
```