numpy practice2

December 1, 2021

[2]:

```
import numpy as np
import math
```

[5]:

```
a=np.zeros((1,3))
print(a)
b=np.ones((2,3))
print(b)
```

```
[[0. 0. 0.]]
[[1. 1. 1.]
 [1. 1. 1.]]
```

[7]:

```
print(np.random.rand(1,5))
```

```
[[0.30453947 0.09434557 0.57434114 0.93998154 0.51191337]]
```

[9]:

```
print(np.arange(10,100,5))
```

```
[10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95]
```

[12]:

```
print(np.linspace(10,100,5))
```

```
[ 10.   32.5  55.   77.5 100. ]
```

[14]:

```
print(np.linspace(10,100,5))
```

```
[ 10.   32.5  55.   77.5 100. ]
```

[19]:

```
a=np.array([[10,1],[9,11]])
b=np.array([[1,19],[5,10]])
print(a-b)
```

```
[[  9 -18]
 [  4   1]]
```

[20]:
```
print(a+b)
```

```
[[11 20]
 [14 21]]
```

[21]:
```
print(a@b)
```

```
[[ 15 200]
 [ 64 281]]
```

[22]:
```
print(a*b)
```

```
[[ 10  19]
 [ 45 110]]
```

[24]:
```
cel=10
cel>a
```

[24]:
```
array([[False,  True],
       [ True, False]])
```

[26]:
```
print(a.sum())
print(a.max())
print(a.min())
print(a.mean())
```

```
31
11
1
7.75
```

[29]:
```
print(np.arange(1,12,2).reshape(3,2))
```

```
[[ 1  3]
 [ 5  7]
 [ 9 11]]
```

[5]:
```
from PIL import Image
from IPython.display import display
im=Image.open('shock ayyava.jpg')
display(im)
```

[6]:

```
arr=np.array(im)
print(arr)
print(arr.shape)
```

```
[[[1 0 0]
  [1 0 0]
  [1 0 0]
  ...
  [1 1 1]
  [0 2 1]
  [0 2 1]]

 [[1 0 0]
  [1 0 0]
  [1 0 0]
  ...
  [1 1 1]
  [1 1 1]
  [0 2 1]]

 [[1 0 0]
  [1 0 0]
  [1 0 0]
  ...
  [1 1 1]
  [1 1 1]
  [1 1 1]]

 ...
 [[0 1 0]
  [0 1 0]
  [0 1 0]
  ...
  [1 0 0]
  [1 0 0]
  [1 0 0]]

 [[0 1 0]
  [0 1 0]
  [0 1 0]
  ...
  [3 0 0]
  [3 0 0]
  [3 0 0]]

 [[0 1 0]
  [0 1 0]
  [0 1 0]
  ...
  [3 0 0]
  [3 0 0]
  [3 0 0]]]
(640, 1351, 3)
```

[7]:
```
arr
```

[7]:
```
array([[[1, 0, 0],
        [1, 0, 0],
        [1, 0, 0],
        ...,
        [1, 1, 1],
        [0, 2, 1],
        [0, 2, 1]],

       [[1, 0, 0],
        [1, 0, 0],
        [1, 0, 0],
        ...,
        [1, 1, 1],
        [1, 1, 1],
        [0, 2, 1]],

       [[1, 0, 0],
        [1, 0, 0],
        [1, 0, 0],
        ...,
        [1, 1, 1],
        [1, 1, 1],
        [1, 1, 1]],

       ...,

       [[0, 1, 0],
        [0, 1, 0],
        [0, 1, 0],
        ...,
        [1, 0, 0],
        [1, 0, 0],
        [1, 0, 0]],

       [[0, 1, 0],
        [0, 1, 0],
        [0, 1, 0],
        ...,
        [3, 0, 0],
        [3, 0, 0],
        [3, 0, 0]],

       [[0, 1, 0],
        [0, 1, 0],
        [0, 1, 0],
        ...,
        [3, 0, 0],
        [3, 0, 0],
        [3, 0, 0]]], dtype=uint8)
```

[8]:

```
mask=np.full(arr.shape,255)
mask
```

[8]:

```
array([[[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],
       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       ...,

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]],

       [[255, 255, 255],
        [255, 255, 255],
        [255, 255, 255],
        ...,
        [255, 255, 255],
        [255, 255, 255],
        [255, 255, 255]]])
```

[10]:
```
mod=mask-arr
mod
```

[10]:
```
array([[[254, 255, 255],
        [254, 255, 255],
        [254, 255, 255],
        ...,
        [254, 254, 254],
        [255, 253, 254],
        [255, 253, 254]],

       [[254, 255, 255],
        [254, 255, 255],
        [254, 255, 255],
        ...,
        [254, 254, 254],
        [254, 254, 254],
        [255, 253, 254]],

       [[254, 255, 255],
        [254, 255, 255],
        [254, 255, 255],
        ...,
        [254, 254, 254],
        [254, 254, 254],
        [254, 254, 254]],

       ...,

       [[255, 254, 255],
        [255, 254, 255],
        [255, 254, 255],
        ...,
        [254, 255, 255],
        [254, 255, 255],
        [254, 255, 255]],

       [[255, 254, 255],
        [255, 254, 255],
        [255, 254, 255],
        ...,
        [252, 255, 255],
        [252, 255, 255],
        [252, 255, 255]],

       [[255, 254, 255],
        [255, 254, 255],
        [255, 254, 255],
        ...,
        [252, 255, 255],
        [252, 255, 255],
        [252, 255, 255]]])
```

[11]:

```
display(mod)
```

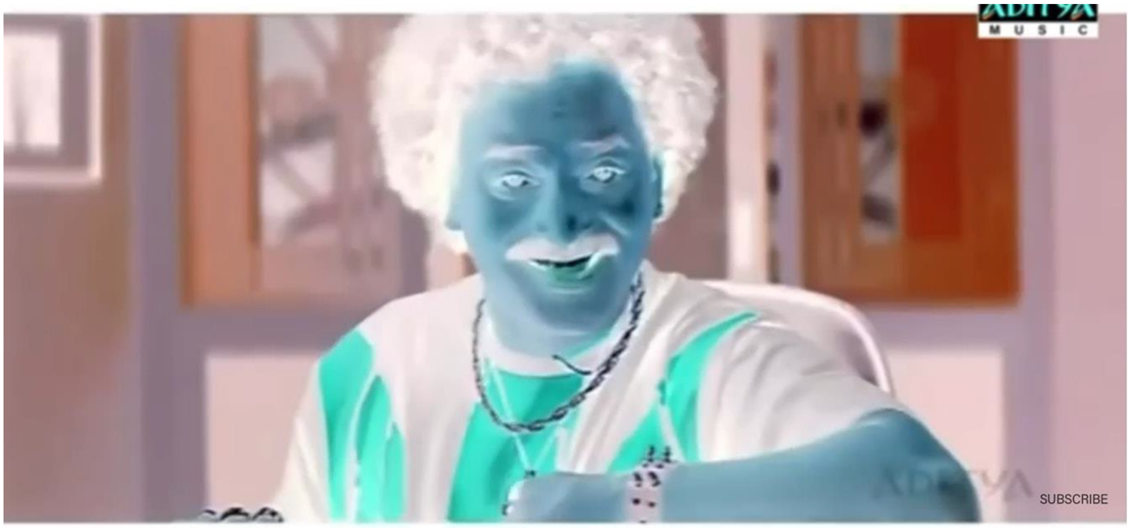
```
array([[[254, 255, 255],
         [254, 255, 255],
         [254, 255, 255],
         ...,
         [254, 254, 254],
         [255, 253, 254],
         [255, 253, 254]],
        [[254, 255, 255],
         [254, 255, 255],
         [254, 255, 255],
         ...,
         [254, 254, 254],
         [254, 254, 254],
         [255, 253, 254]],

        [[254, 255, 255],
         [254, 255, 255],
         [254, 255, 255],
         ...,
         [254, 254, 254],
         [254, 254, 254],
         [254, 254, 254]],

        ...,

        [[255, 254, 255],
         [255, 254, 255],
         [255, 254, 255],
         ...,
         [254, 255, 255],
         [254, 255, 255],
         [254, 255, 255]],

        [[255, 254, 255],
         [255, 254, 255],
         [255, 254, 255],
         ...,
         [252, 255, 255],
         [252, 255, 255],
         [252, 255, 255]],

        [[255, 254, 255],
         [255, 254, 255],
         [255, 254, 255],
         ...,
         [252, 255, 255],
         [252, 255, 255],
         [252, 255, 255]]])
```

[12]:
```
mod=mod.astype(np.uint8)
mod
```

[12]:

```
array([[[254, 255, 255],
        [254, 255, 255],
        [254, 255, 255],
        ...,
        [254, 254, 254],
        [255, 253, 254],
        [255, 253, 254]],

       [[254, 255, 255],
        [254, 255, 255],
        [254, 255, 255],
        ...,
        [254, 254, 254],
        [254, 254, 254],
        [255, 253, 254]],

       [[254, 255, 255],
        [254, 255, 255],
        [254, 255, 255],
        ...,
        [254, 254, 254],
        [254, 254, 254],
        [254, 254, 254]],

       ...,

       [[255, 254, 255],
        [255, 254, 255],
        [255, 254, 255],
        ...,
        [254, 255, 255],
        [254, 255, 255],
        [254, 255, 255]],

       [[255, 254, 255],
        [255, 254, 255],
        [255, 254, 255],
        ...,
        [252, 255, 255],
        [252, 255, 255],
        [252, 255, 255]],

       [[255, 254, 255],
        [255, 254, 255],
        [255, 254, 255],
        ...,
        [252, 255, 255],
        [252, 255, 255],
        [252, 255, 255]]], dtype=uint8)
```

[13]:
```
display(Image.fromarray(mod))
```



[ ]:
```

```

[49]:
```
abc=np.array([[1,11],[10,9],[15,5]])
abc
```

[49]:
```
array([[ 1, 11],
       [10,  9],
       [15,  5]])
```

[52]:
```
bcd=np.array([abc[1,1],abc[0,0],abc[1,0]])
bcd
```

[52]:
```
array([ 9,  1, 10])
```

[53]:
```
print(abc[abc>10])
```

```
[11 15]
```

[57]:
```
abc[:2]
abc[::-1]
bcd[:2]
bcd[::-1]
abc[:2,0:2]
```

[57]:
```
array([[ 1, 11],
       [10,  9]])
```

[59]:
```
abc[:2,0:1]
```

```
[59]: array([[ 1],
             [10]])
```

```
[1]: import numpy as np
     import math
```

```
[5]: bc=np.array([[1,12,11],[10,9,55]])
     print(bc[:2,:2])
     bc
```

```
[[ 1 12]
 [10  9]]
```

```
[5]: array([[ 1, 12, 11],
            [10,  9, 55]])
```

```
[6]: sub=bc[:2,:2]
     sub[0,0]=100
     print(bc)
     print(sub)
```

```
[[100  12  11]
 [ 10   9  55]]
[[100  12]
 [ 10   9]]
```

```
[2]: wines=np.genfromtxt("winequality-red.csv",delimiter=";",skip_header=1)
     wines
```

```
[2]: array([[ 7.4  ,  0.7  ,  0.   , ...,  0.56 ,  9.4  ,  5.   ],
            [ 7.8  ,  0.88 ,  0.   , ...,  0.68 ,  9.8  ,  5.   ],
            [ 7.8  ,  0.76 ,  0.04 , ...,  0.65 ,  9.8  ,  5.   ],
            ...,
            [ 6.3  ,  0.51 ,  0.13 , ...,  0.75 , 11.   ,  6.   ],
            [ 5.9  ,  0.645,  0.12 , ...,  0.71 , 10.2  ,  5.   ],
            [ 6.   ,  0.31 ,  0.47 , ...,  0.66 , 11.   ,  6.   ]])
```

```
[3]: print(wines[:1])
```

```
[[ 7.4     0.7     0.      1.9     0.076  11.     34.      0.9978  3.51
   0.56    9.4     5.    ]]
```

```
[2]: import numpy as np
     import math
```

```
[5]: wines=np.genfromtxt("winequality-red.csv",delimiter=";",skip_header=1)
     wines
```

```
[5]: array([[ 7.4  ,  0.7  ,  0.   , ...,  0.56 ,  9.4  ,  5.   ],
            [ 7.8  ,  0.88 ,  0.   , ...,  0.68 ,  9.8  ,  5.   ],
            [ 7.8  ,  0.76 ,  0.04 , ...,  0.65 ,  9.8  ,  5.   ],
            ...,
            [ 6.3  ,  0.51 ,  0.13 , ...,  0.75 , 11.   ,  6.   ],
            [ 5.9  ,  0.645,  0.12 , ...,  0.71 , 10.2  ,  5.   ],
            [ 6.   ,  0.31 ,  0.47 , ...,  0.66 , 11.   ,  6.   ]])
```

[8]:
```python
print(wines[:,:1])
print(wines[:,0])
print(wines[:,0:2])
```

```
[[7.4]
 [7.8]
 [7.8]
 ...
 [6.3]
 [5.9]
 [6. ]]
[7.4 7.8 7.8 ... 6.3 5.9 6. ]
[[7.4   0.7  ]
 [7.8   0.88 ]
 [7.8   0.76 ]
 ...
 [6.3   0.51 ]
 [5.9   0.645]
 [6.    0.31 ]]
```

[4]:
```python
print(wines[:,[0,2,4]])
```

```
[[7.4   0.    0.076]
 [7.8   0.    0.098]
 [7.8   0.04  0.092]
 ...
 [6.3   0.13  0.076]
 [5.9   0.12  0.075]
 [6.    0.47  0.067]]
```

[5]:
```python
print(wines[:,-1].mean())
```

```
5.6360225140712945
```

we can also add another attribute in genfromtxt. example is given below. grad=np.genfromtxt(“Admission_Predict.csv”, dtype=“None”, delimiter=“,”, skip_header=1, names=(‘serial number’, ‘GRE score’, ‘TOEFEL score’, ‘CGPA’))

[6]:
```python
print(wines[:,1].mean())
```

```
0.5278205128205128
```

[ ]:

```
```