20/07/23

DAY 2 PRACTICE QUESTIONS

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

1. Create an array with zeros and ones and print the output

In [6]:
```
a=np.zeros(5,dtype=np.int64)
b=np.ones(5,dtype=np.int64)
print(a)
print(b)
```

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

2. Create an array and print the output

In [7]:
```
arr=np.array([1,2,3,4])
print(arr)
```

```
[1 2 3 4]
```

3. Create an array whose initial content is random and print the output

In [13]:
```
a=np.random.rand(1,3)
print(a)
```

```
[[0.43687248 0.0919653  0.51125038]]
```

4. Create an array with the range of values with even intervals

In [17]:
```
a=np.linspace(1,10,4,dtype=np.int64)
print(a)
```

```
[ 1  4  7 10]
```

5. create an array with values that are spaced linearly in a specified interval

In [19]:
```
s=np.arange(1,10,3)
print(s)
```

```
[1 4 7]
```

6. Access and manipulate elements in the array

In [20]:
```
arr=np.array([1,2,4,6,8])
```

In [21]:
```
arr[0]
```

Out[21]: 1

In [22]:
```
arr[4]
```

Out[22]: 8

In [23]:
```
arr[2]=10
```

In [24]:
```
arr[2]
```

Out[24]: 10

7. Create a 2-dimensional array and check the shape of the array

In [26]:
```
arr=np.array([[1,2,3],[10,20,30]])
print(arr)
print(np.shape(arr))
```

```
[[ 1  2  3]
 [10 20 30]]
(2, 3)
```

8. Using the arange() and linspace() function to evenly space values in a specified interval

In [33]:
```
print(np.arange(1,10,3))
print(np.linspace(1,10,3,dtype=np.int64))
```

```
[1 4 7]
[ 1  5 10]
```

9. Create an array of random values between 0 and 1 in a given shape

In [37]:
```
arr=np.random.rand(3,4)
print(arr)
```

```
[[0.76554188 0.08001317 0.55512418 0.97681067]
 [0.10327572 0.27084852 0.67692192 0.91071292]
 [0.7633524  0.62247168 0.54565795 0.55111235]]
```

10. Repeat each element of an array by a specified number of times using repeat() and tile() functions

In [41]:
```
arr=np.array([[1,2,3],[10,20,30]])
print(np.tile(arr,3))
```

```
[[ 1  2  3  1  2  3  1  2  3]
 [10 20 30 10 20 30 10 20 30]]
```

In [42]:
```
print(np.repeat(arr,3))
```

```
[ 1  1  1  2  2  2  3  3  3 10 10 10 20 20 20 30 30 30]
```

11. How do you know the shape and size of an array?

In [44]:
```
#We can use the shape() and size() functions from numpy to find the shape and size of an array
a=np.array([[1,2,3],[20,40,60]])
print(np.shape(a))
```

```
(2, 3)
```

In [45]:
```
print(np.size(a))
```

```
6
```

12. Create an array that indicates the total number of elements in an array

In [46]:
```
arr=np.array([1,2,3,44,54,67,98,90])
print(np.size(arr))
```

```
8
```

13. To find the number of dimensions of the array

In [52]:
```
a=np.array([[[2,3,4],[10,20,30]],[[100,200,300],[600,700,800]]])
print(np.ndim(a))
```

```
3
```

14. Create an array and reshape into a new array

In [53]:
```
arr=np.array([[1,2,3],[10,20,30]])
print(arr.reshape(3,2))
```

```
[[ 1  2]
 [ 3 10]
 [20 30]]
```

15. Create a null array of size 10

In [54]:
```
z=np.array(np.zeros(10,dtype=np.int64))
print(z)
```

```
[0 0 0 0 0 0 0 0 0 0]
```

16. Create any array with values ranging from 10 to 49 and print the numbers whose remainders are zero when divided by 7

In [57]:
```
q=np.array(np.arange(10,50))
condition=q[(q%7==0)]
print(condition)
```

```
[14 21 28 35 42 49]
```

17. Create an array and check any two conditions and print the output

In [59]:
```
arr=np.array([1,2,3,4])
condition=arr[(arr>1)&(arr<4)]
print(condition)
```

```
[2 3]
```

18. Use Arithmetic operator and print the output using array

In [64]:
```
a=np.array([10,20,30])
b=np.array([1,2,3])
cal=a+b
print("Array addition=",cal)
```

```
Array addition= [11 22 33]
```

19. Use Relational operators and print the results using array

In [67]:
```
b=np.array([100,200,10,1,700])
cal=b[(b>100)]
print(cal)
```

```
[200 700]
```

20. Difference between python and ipython

```
@Python:
-Python is programming language
-It is used by programmers
@IPython:
-It is a enhanced interactive shell of python
-It is used by data scientists and researchers
```