

NUMPY 20.07.2023

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

1. Create an array with zeros and ones

```
In [2]: a=np.zeros(4)  
print(a)
```

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

```
In [3]: b=np.ones(6)  
print(b)
```

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

```
In [4]: c=np.zeros(4,dtype=np.int64)  
print(c)
```

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

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

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

2. Create an array and print the output

```
In [6]: e = np.array([11,22,33])  
print(e)
```

```
[11 22 33]
```

```
In [7]: f=np.array([[1,2,3],[4,5,6]])  
print(f)
```

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

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

```
In [8]: print(np.empty(2))
```

```
[7.97878617e+227 4.41491748e+145]
```

```
In [9]: print(np.empty(3))
```

```
[4.45041255e-307 2.33645148e-307 1.33509389e-306]
```

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

```
In [15]: g=np.arange(2,11,2)  
print(g)
```

```
[ 2  4  6  8 10]
```

```
In [16]: h=np.arange(10,31,2)  
print(h)
```

```
[10 12 14 16 18 20 22 24 26 28 30]
```

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

```
In [17]: print(np.linspace(1,200,num=25))
```

```
[ 1.          9.29166667 17.58333333 25.875         34.16666667
 42.45833333 50.75         59.04166667 67.33333333 75.625
 83.91666667 92.20833333 100.5         108.79166667 117.08333333
125.375      133.66666667 141.95833333 150.25         158.54166667
166.83333333 175.125      183.41666667 191.70833333 200.         ]
```

```
In [18]: print(np.linspace(1,200,num=20,dtype= np.int64))
```

```
[ 1 11 21 32 42 53 63 74 84 95 105 116 126 137 147 158 168 179
189 200]
```

6. Access and manipulate elements in the array

```
In [26]: print(h[5])
```

```
20
```

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

```
In [45]: j=np.array([[10,20,30],[40,50,60],[70,80,90]])
print(j)
print(np.ndim(j))
```

```
[[10 20 30]
 [40 50 60]
 [70 80 90]]
2
```

```
In [21]: print(np.shape(j))
```

```
(3, 3)
```

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

```
In [52]: f=np.arange(0,51,+2)
         print(f)
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
 48 50]
```

```
In [53]: print(np.linspace(0,50,num=26,dtype= np.int64))
```

```
[ 0  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
 48 50]
```

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

```
In [55]: print(np.empty(10))
```

```
[ 2.          2.88888889  3.77777778  4.66666667  5.55555556  6.44444444
 7.33333333  8.22222222  9.11111111 10.          ]
```

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

```
In [54]: print(np.tile(e,3))
```

```
[11 22 33 11 22 33 11 22 33]
```

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

```
In [57]: print(np.shape(h))  
print(np.size(h))
```

```
(11,)  
11
```

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

```
In [59]: o=np.arange(1,31,2)  
print(o)  
print(np.size(o))
```

```
[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29]  
15
```

13. To find the number of dimensions of the array

```
In [60]: l=np.arange(1,39,2)  
print(l)  
print(np.ndim(l))
```

```
[ 1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37]  
1
```

14. Create an array and reshape into a new array

```
In [66]: x=np.arange(6)
print(x)
y=x.reshape(3,2)
print(y)
```

```
[0 1 2 3 4 5]
[[0 1]
 [2 3]
 [4 5]]
```

15.Create a null array of size 10

```
In [70]: z=np.zeros(11)
print(z)
```

```
[0. 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 [73]: di=np.arange(7,50,+7)
print(di)
```

```
[ 7 14 21 28 35 42 49]
```

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

```
In [95]: print(di[di<9])
print(di[di>9])
```

```
[7]
[14 21 28 35 42 49]
```

18. Use Arithmetic operator and print the output using array

```
In [111]: print(di[2]+di[3])
```

49

19. Use Relational operators and print the results using array

```
In [112]: print(di%7==0)
```

[True True True True True True True]

20. Difference between python and ipython

1.Programming language's standard interpreter,Enhanced interactive shell for Python. Basic shell with limited 2.interactive features,Rich command-line interface with enhanced features. 3.Basic command history navigation,Advanced history navigation and searching. 4.No built-in magic commands, Provides extensive magic commands for convenience 5.Basic help function available. Rich help system with detailed documentation. 6.Basic profiling using external modules.In-built line-by-line code profiler.

```
In [ ]:
```