NumPy Array Creation

May 15, 2022

[33]:
```
import numpy as np

a = np.array([10,20,30]) #1-D Array

print(a)
```

```
[10 20 30]
```

[34]:
```
#2-D Array

a = np.array([[20,30,40,8],[55,88,99,0],[29,77,9,0]])

print(a)
a.ndim
```

```
[[20 30 40  8]
 [55 88 99  0]
 [29 77  9  0]]
```

[34]:
```
2
```

[35]:
```
#3-D Array

a = np.array([[[10,20,30],[40,50,60]],[[70,80,90],[100,110,120]]])

print(a)
a.ndim
```

```
[[[ 10  20  30]
  [ 40  50  60]]

 [[ 70  80  90]
  [100 110 120]]]
```

[35]:
```
3
```

[36]:
```
#4-D Array
```

```
a = np.
 ↪array([[[[10,20,30],[40,50,60]],[[70,80,90],[100,110,120]],[[130,140,150],[160,170,180]]]])

print(a)
a.ndim
```

```
[[[[ 10  20  30]
   [ 40  50  60]]

  [[ 70  80  90]
   [100 110 120]]

  [[130 140 150]
   [160 170 180]]]]
```

[36]:
```
4
```

[37]:
```
#zeros

z = np.zeros((4,5)) #by default float
z
```

[37]:
```
array([[0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0.]])
```

[38]:
```
#zeros

z = np.zeros((3,5), dtype = int) #explicitly type casting
z
```

[38]:
```
array([[0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0]])
```

[39]:
```
#ones for 2 dimension array

o = np.ones((3,2), dtype ="int32") #creating array with required data type
o
```

[39]:
```
array([[1, 1],
       [1, 1],
       [1, 1]])
```

[40]:
```
#full
```

```
f = np.full((3,2),5)
f
```

[40]:
```
array([[5, 5],
       [5, 5],
       [5, 5]])
```

[41]:
```
#eye

e = np.eye(3)
e
```

[41]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

[42]:
```
e = np.eye(3,2)
e
```

[42]:
```
array([[1., 0.],
       [0., 1.],
       [0., 0.]])
```

[43]:
```
e = np.eye(N=3, M=5,k=0)
e
```

[43]:
```
array([[1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.]])
```

[45]:
```
e = np.eye(N=3, M=5,k=1)
e
```

[45]:
```
array([[0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.]])
```

[46]:
```
e = np.eye(N=3, M=5,k=2)
e
```

[46]:
```
array([[0., 0., 1., 0., 0.],
       [0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.]])
```

[47]:
```
e = np.eye(N=3, M=5,k=-1)
e
```

[47]:
```
array([[0., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0.]])
```

[49]:
```
#identity

e = np.identity(3)
e
```

[49]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

[55]:
```
#range function in numpy

r = np.arange(2,10)
r
```

[55]:
```
array([2, 3, 4, 5, 6, 7, 8, 9])
```

[59]:
```
#linspace (equaly spaced)

l = np.linspace(1,5) #by default it take n= 50 if you don't want to change␣
 ↪otherwise => (1,10,n=whatever)
l
```

[59]:
```
array([1.        , 1.08163265, 1.16326531, 1.24489796, 1.32653061,
       1.40816327, 1.48979592, 1.57142857, 1.65306122, 1.73469388,
       1.81632653, 1.89795918, 1.97959184, 2.06122449, 2.14285714,
       2.2244898 , 2.30612245, 2.3877551 , 2.46938776, 2.55102041,
       2.63265306, 2.71428571, 2.79591837, 2.87755102, 2.95918367,
       3.04081633, 3.12244898, 3.20408163, 3.28571429, 3.36734694,
       3.44897959, 3.53061224, 3.6122449 , 3.69387755, 3.7755102 ,
       3.85714286, 3.93877551, 4.02040816, 4.10204082, 4.18367347,
       4.26530612, 4.34693878, 4.42857143, 4.51020408, 4.59183673,
       4.67346939, 4.75510204, 4.83673469, 4.91836735, 5.        ])
```

[63]:
```
#empty array (uninitialised data/ to store future data)

e = np.empty(3) #result may be any number but actually it is empty
e
```

[63]:
```
array([1., 1., 1.])
```

[ ]:
```

```