

Sajeeb Datta 201-15-3119

Numpy Array

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

print(type(arr))

[1 2 3 4 5]
<class 'numpy.ndarray'>
```

```
arr = np.array((1, 2, 3, 4, 5))

print(arr)

[1 2 3 4 5]
```

0-D Array

```
arr = np.array(42)

print(arr)

42
```

1-D Array

```
arr = np.array([1, 2, 3, 4, 5])

print(arr)

[1 2 3 4 5]
```

2-D Array

```
arr = np.array([[1, 2, 3], [4, 5, 6]])

. . . ,
```

 0s completed at 12:59 AM

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

3-D Array

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
```

```
print(arr)
```

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

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

Number of Dimensions

```
a = np.array(42)
```

```
b = np.array([1, 2, 3, 4, 5])
```

```
c = np.array([[1, 2, 3], [4, 5, 6]])
```

```
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])
```

```
print(a.ndim)
```

```
print(b.ndim)
```

```
print(c.ndim)
```

```
print(d.ndim)
```

```
0
```

```
1
```

```
2
```

```
3
```

Access Array Elements

```
arr = np.array([1, 2, 3, 4])
```

```
print(arr[0])
```

```
1
```

```
arr = np.array([1, 2, 3, 4])
```

```
print(arr[1])
```

```
2
```

-

```
arr = np.array([1, 2, 3, 4])
```

```
print(arr[2] + arr[3])
```

7

Access 2-D Arrays

```
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
```

```
print('2nd element on 1st row: ', arr[0, 1])
```

2nd element on 1st row: 2

Fifth Element on the Second Row

```
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
```

```
print('5th element on 2nd row: ', arr[1, 4])
```

5th element on 2nd row: 10

Access 3-D Arrays

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
```

```
print(arr[0, 1, 2])
```

6

Negative Indexing

```
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
```

```
print('Last element from 2nd dim: ', arr[1, -1])
```

Last element from 2nd dim: 10

Slicing Arrays

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

```
print(arr[1:5])
```

```
[2 3 4 5]
```

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

```
print(arr[4:])
```

```
[5 6 7]
```

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

```
print(arr[:4])
```

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

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

```
print(arr[-3:-1])
```

```
[5 6]
```

Step

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

```
print(arr[1:5:2])
```

```
[2 4]
```

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

```
print(arr[::2])
```

```
[1 3 5 7]
```

Slicing 2-D Arrays

```
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
```

```
print(arr[1, 1:4])
```

```
[7 8 9]
```

```
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
```

```
print(arr[0:2, 2])
```

```
[3 8]
```

```
arr = np.array([[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]])
```

```
print(arr[0:2, 1:4])
```

```
[[2 3 4]
 [7 8 9]]
```

Data Types

```
arr = np.array([1, 2, 3, 4])
```

```
print(arr.dtype)
```

```
int64
```

```
arr = np.array(['apple', 'banana', 'cherry'])
```

```
print(arr.dtype)
```

```
<U6
```

```
arr = np.array([1, 2, 3, 4], dtype='S')
```

```
print(arr)
```

```
print(arr.dtype)
```

```
[b'1' b'2' b'3' b'4']
|S1
```

```
arr = np.array([1, 2, 3, 4], dtype='i4')
```

```
print(arr)
```

```
print(arr.dtype)
```

```
[1 2 3 4]
int32
```

Converting Data Types on Existing Arrays

```
arr = np.array([1.1, 2.1, 3.1])
```

```
newarr = arr.astype('i')

print(newarr)
print(newarr.dtype)

[1 2 3]
int32

arr = np.array([1.1, 2.1, 3.1])

newarr = arr.astype(int)

print(newarr)
print(newarr.dtype)

[1 2 3]
int64

arr = np.array([1, 0, 3])

newarr = arr.astype(bool)

print(newarr)
print(newarr.dtype)

[ True False  True]
bool
```

Array Copy VS View

```
arr = np.array([1, 2, 3, 4, 5])
x = arr.copy()
arr[0] = 42

print(arr)
print(x)

[42  2  3  4  5]
[1 2 3 4 5]

arr = np.array([1, 2, 3, 4, 5])
x = arr.view()
arr[0] = 42

print(arr)
print(x)
```

```
[42  2  3  4  5]
[42  2  3  4  5]
```

```
arr = np.array([1, 2, 3, 4, 5])
x = arr.view()
x[0] = 31
```

```
print(arr)
print(x)
```

```
[31  2  3  4  5]
[31  2  3  4  5]
```

```
arr = np.array([1, 2, 3, 4, 5])
```

```
x = arr.copy()
y = arr.view()
```

```
print(x.base)
print(y.base)
```

```
None
[1 2 3 4 5]
```

Shape of an Array

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

```
print(arr.shape)
```

```
(2, 4)
```

```
arr = np.array([1, 2, 3, 4], ndmin=5)
```

```
print(arr)
print('shape of array :', arr.shape)
```

```
[[[[[1 2 3 4]]]]]
shape of array : (1, 1, 1, 1, 4)
```

Iterating Arrays

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

```
for x in arr:
```

```
print(x)
```

```
1
2
3
```

Iterating 2-D Arrays

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
for x in arr:
    print(x)
```

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

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
```

```
for x in arr:
    for y in x:
        print(y)
```

```
1
2
3
4
5
6
```

Iterating 3-D Arrays

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
```

```
for x in arr:
    print(x)
```

```
[[1 2 3]
 [4 5 6]]
[[ 7  8  9]
 [10 11 12]]
```

```
arr = np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])
```

```
for x in arr:
    for y in x:
        for z in y:
            print(z)
```



```
1
2
3
4
5
6
7
8
9
10
11
12
```

Array Joining

```
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr = np.concatenate((arr1, arr2))
print(arr)

[1 2 3 4 5 6]

arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
arr = np.concatenate((arr1, arr2), axis=1)
print(arr)

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

arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
arr = np.stack((arr1, arr2), axis=1)
print(arr)

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

Array Splitting

```
arr = np.array([1, 2, 3, 4, 5, 6])

newarr = np.array_split(arr, 3)

print(newarr)

[array([1, 2]), array([3, 4]), array([5, 6])]

arr = np.array([1, 2, 3, 4, 5, 6])

newarr = np.array_split(arr, 4)

print(newarr)

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

Split into Arrays

```
arr = np.array([1, 2, 3, 4, 5, 6])

newarr = np.array_split(arr, 3)

print(newarr[0])
print(newarr[1])
print(newarr[2])

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

Splitting 2-D Arrays

```
arr = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10], [11, 12]])

newarr = np.array_split(arr, 3)

print(newarr)

[array([[1, 2],
        [3, 4]]), array([[5, 6],
        [7, 8]]), array([[ 9, 10],
        [11, 12]])]

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

```
newarr = np.array_split(arr, 3)

print(newarr)

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

Array Search

```
arr = np.array([1, 2, 3, 4, 5, 4, 4])

x = np.where(arr == 4)

print(x)

(array([3, 5, 6]),)
```

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

x = np.where(arr%2 == 0)

print(x)

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

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

x = np.where(arr%2 == 1)

print(x)

(array([0, 2, 4, 6]),)
```

Sorted Search

```
arr = np.array([6, 7, 8, 9])

x = np.searchsorted(arr, 7)

print(x)
```

```
arr = np.array([6, 7, 8, 9])

x = np.searchsorted(arr, 7, side='right')

print(x)

2
```

```
arr = np.array([1, 3, 5, 7])

x = np.searchsorted(arr, [2, 4, 6])

print(x)

[1 2 3]
```

```
arr = np.array([3, 2, 0, 1])

print(np.sort(arr))

[0 1 2 3]
```

```
arr = np.array(['banana', 'cherry', 'apple'])

print(np.sort(arr))

['apple' 'banana' 'cherry']
```

```
arr = np.array([True, False, True])

print(np.sort(arr))

[False  True  True]
```

```
arr = np.array([[3, 2, 4], [5, 0, 1]])

print(np.sort(arr))

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

Array Filtering

```
arr = np.array([41, 42, 43, 44])

x = [True, False, True, False]
```

```
newarr = arr[x]

print(newarr)

[41 43]
```

Colab paid products - Cancel contracts here