Creating Array

```
import numpy

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

print(arr)
```

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

```
import numpy as np

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

print(arr)
```

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

```
import numpy as np

print(np.__version__)
```

```
     1.22.4
```

```
import numpy as np

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

print(arr)

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

```
     [1 2 3 4 5]
     <class 'numpy.ndarray'>
     int64
```

```
import numpy as np

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)
print(d)
```

```
     0
     1
     2
     3
     [[[1 2 3]
       [4 5 6]]

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

```
import numpy as np

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

print(arr)
print('number of dimensions :', arr.ndim)
```

```
     [[[[[1 2 3 4]]]]]
     number of dimensions : 5
     (1, 1, 1, 1, 4)
```

```
a=np.zeros((3,6))
print(a)
```

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

```
a=np.ones((3,6))
print(a)
print(a.dtype)
```

```
     [[1. 1. 1. 1. 1. 1.]
      [1. 1. 1. 1. 1. 1.]
      [1. 1. 1. 1. 1. 1.]]
     float64
```

Access Array Elements

```
import numpy as np

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

print(arr[0])
```

```
     1
```

```
import numpy as np

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

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

```
     7
```

```
import numpy as np

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
```

```
import numpy as np

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
```

```
import numpy as np

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

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

```
     6
```

```
import numpy as np

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

```
import numpy as np

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

print(arr[1:5])

print(arr[4:])

print(arr[:4])
print(arr[-3:-1])

print(arr[1:5:2])

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

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

```
import numpy as np

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

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

```
     [7 8 9]
```

```
import numpy as np

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

print(arr[0:2, 2])

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

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

Sorting

```
import numpy as np

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

print(np.sort(arr))
```

```
     [0 1 2 3]
```

```
import numpy as np

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

print(np.sort(arr))
```

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

```
import numpy as np

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

print(np.sort(arr))
```

```
     [False  True  True]
```

```
import numpy as np

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

print(np.sort(arr))
```

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

Aggregate Functions

```
import numpy as np
```

```
array1 = np.array([[10, 20, 30], [40, 50, 60]])

print("Mean: ", np.mean(array1))

print("Std: ", np.std(array1))

print("Var: ", np.var(array1))

print("Sum: ", np.sum(array1))

print("Prod: ", np.prod(array1))

print("Max: ",np.max(array1))

print("Min: ",np.min(array1))
```

```
Mean:  35.0
Std:  17.07825127659933
Var:  291.6666666666667
Sum:  210
Prod:  720000000
Max:  60
Min:  10
```

Shape of an Array

```
import numpy as np

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

print(arr.shape)
```

```
(2, 4)
```

```
import numpy as np

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)
```

Reshape of an Array

```
import numpy as np

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

newarr = arr.reshape(4, 3)

print(newarr)
```

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

```
import numpy as np

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

newarr = arr.reshape(2, 3, 2)

print(newarr)
```

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

 [[ 7  8]
  [ 9 10]
  [11 12]]]
```

Arithmetic Operations

```
import numpy as np

array1 = np.array([[1, 2, 3], [4, 5, 6]])
array2 = np.array([[7, 8, 9], [10, 11, 12]])

print(array1 + array2)
print("-" * 20)

print(array1 - array2)
print("-" * 20)

print(array1 * array2)
print("-" * 20)

print(array2 / array1)
print("-" * 40)

print(array1 ** array2)
print("-" * 40)
```

```
[[ 8 10 12]
 [14 16 18]]
--------------------
[[-6 -6 -6]
 [-6 -6 -6]]
--------------------
[[ 7 16 27]
 [40 55 72]]
--------------------
[[7.  4.  3. ]
 [2.5 2.2 2. ]]
----------------------------------------
[[         1        256      19683]
 [   1048576   48828125 2176782336]]
----------------------------------------
```

Others functions

```
arr = np.arange(32).reshape((8, 4))
arr
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23],
       [24, 25, 26, 27],
       [28, 29, 30, 31]])
```

```
arr = np.arange(15).reshape((3, 5))
arr
```

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

```
arr.T
```

```
array([[ 0,  5, 10],
       [ 1,  6, 11],
       [ 2,  7, 12],
       [ 3,  8, 13],
       [ 4,  9, 14]])
```

```
arr = np.random.rand(6, 3)
print(arr)
arr1 = np.random.randint(6, size = (3,4))
print(arr1)
```

```
[[0.6427884  0.98489737 0.91001298]
 [0.95103598 0.76732051 0.26073884]
 [0.33774995 0.05528029 0.7758937 ]
 [0.05403777 0.7532079  0.05476883]
 [0.22201759 0.89026304 0.33697266]
 [0.89731945 0.58064155 0.48202196]]
[[1 3 5 3]
 [4 4 5 3]
 [0 5 4 0]]
```