

---

---

# NumPy

---

---

# Joining NumPy Arrays

```
import numpy as np
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]

## Joining NumPy 2D Arrays along rows (axis=1)

```
import numpy as np
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]]
```

# Joining NumPy Arrays using Stack Functions

```
import numpy as np
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]]
```

# Joining NumPy Arrays using Stacking Along Rows

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

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

# Joining NumPy Arrays using Stacking Along Columns

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

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

# Joining NumPy Arrays using Stacking Along Hight (Depth)

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

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

# Splitting NumPy Arrays

```
import numpy as np
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])]
```

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

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



# Searching in NumPy Arrays

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 4, 4])
x = np.where(arr == 4)
print(x)
```

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

- Find the indexes where the values are even
- Find the indexes where the values are odd

# Search Sorted in NumPy Arrays

```
import numpy as np
arr = np.array([6, 7, 8, 9])
x = np.searchsorted(arr, 7)
print(x)
```

1

Method called `searchsorted()` which performs a binary search in the array, and returns the index where the specified value would be inserted

# Search Sorted from Right in NumPy Arrays

```
import numpy as np
arr = np.array([6, 7, 8, 9])
x = np.searchsorted(arr, 7, side='right')
print(x)
```

2

# Sorting in NumPy Arrays

```
import numpy as np  
arr = np.array([3, 2, 0, 1])  
print(np.sort(arr))
```

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

# Sorting in NumPy Arrays

```
import numpy as np  
arr = np.array(['banana', 'cherry', 'apple'])  
print(np.sort(arr))
```

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

# Arithmetic Operations : Addition

```
import numpy as np
arr1 = np.array([10, 11, 12, 13, 14, 15])
arr2 = np.array([20, 21, 22, 23, 24, 25])
newarr = np.add(arr1, arr2)
print(newarr)
```

```
[30 32 34 36 38 40]
```

# Arithmetic Operations : Subtract

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50, 60])
arr2 = np.array([20, 21, 22, 23, 24, 25])
newarr = np.subtract(arr1, arr2)
print(newarr)
```

```
[-10  -1   8  17  26  35]
```

# Arithmetic Operations : Multiply

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50, 60])
arr2 = np.array([20, 21, 22, 23, 24, 25])
newarr = np.multiply(arr1, arr2)
print(newarr)
```

```
[ 200  420  660  920 1200 1500]
```



# Arithmetic Operations : Division

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50, 60])
arr2 = np.array([3, 5, 10, 8, 2, 33])
newarr = np.divide(arr1, arr2)
print(newarr)
```

```
[ 3.33333333  4.          3.          5.         25.         1.81818182]
```

# Arithmetic Operations : Power

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50, 60])
arr2 = np.array([3, 5, 6, 8, 2, 33])
newarr = np.power(arr1, arr2)
print(newarr)
```

```
[          1000          3200000          729000000 65536000000000          2500
           0]
```

# Arithmetic Operations : Mod

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50, 60])
arr2 = np.array([3, 7, 9, 8, 2, 33])
newarr = np.mod(arr1, arr2)
print(newarr)
```

```
[ 1  6  3  0  0 27]
```

# Arithmetic Operations : Remainder

```
import numpy as np
arr1 = np.array([10, 20, 30, 40, 50, 60])
arr2 = np.array([3, 7, 9, 8, 2, 33])
newarr = np.remainder(arr1, arr2)
print(newarr)
```

```
[ 1  6  3  0  0 27]
```

# Arithmetic Operations : Absolute

```
import numpy as np
arr = np.array([-1, -2, 1, 2, 3, -4])
newarr = np.absolute(arr)
print(newarr)
```

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

# Arithmetic Operations : Trunct

```
import numpy as np  
arr = np.trunc([-3.1666, 3.6667])  
print(arr)
```

```
[-3.  3.]
```

# Arithmetic Operations : Rounding

```
import numpy as np  
arr = np.around(3.1666, 2)  
print(arr)
```

3.17

# Arithmetic Operations : Sum

```
import numpy as np
arr1 = np.array([1, 2, 3])
arr2 = np.array([1, 2, 3])
newarr = np.sum([arr1, arr2])
print(newarr)
```



# Arithmetic Operations : Product

```
import numpy as np
arr = np.array([1, 2, 3, 4])
x = np.prod(arr)
print(x)
```

24

---