

Exp 4

Array Manipulation (np.append, np.insert, np.resize, np.delete, np.concatenate, np.vstack, np.hstack)

append()

By using the insert() function, we can insert elements at our required index position. If we want to add elements always at end of the ndarray, then we have to go for the append() function.

Program 10: A program to append 100 element to the existing array of int.

```
import numpy as np9
a = np9.arange(10)
b = np9.append(a,100)
print("array a : ",a)
print("array b : ",b)
```

insert()

Insert values along the given axis before the given indices.

Syntax

insert(array, obj, values, axis=None)

where

obj-->Object that defines the index or indices before which 'values' are inserted.

values--->Values to insert into array.

axis ---->Axis along which to insert 'values'.

Program 1: A program to insert 7777 before index 2.

```
import numpy as np9
a = np9.arange(10)
b = np9.insert(a,2,7777)
print("array a : ",a)
```

```
print("array b : ",b)
```

Note:

- All the insertion points(indices) are identified at the beginning of the insert operation.
- The array should contain only homogeneous elements.
- By using the insert() function, if we are trying to insert any other type element, then that element will be converted to array type automatically before insertion.
- If the conversion is not possible then we will get an error.

resize()

The resize() function is used to create a new array with the specified shape. If the new array is larger than the original array, then the new array is filled with repeated copies of a.

Syntax:

```
numpy.resize(a, new_shape)
```

Example

```
import numpy as np
```

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

```
>>> np.resize(a, (3,2))
```

```
array([[1, 2],
```

```
       [3, 4],
```

```
       [1, 2]])
```

DELETE ELEMENTS FROM nd ARRAY

We can delete elements of ndarray by using delete() function.

Syntax

```
delete(arr, obj, axis=None)
```

where

- obj can be int, an array of ints, or slice

- for multi-dimensional arrays we must have to specify the axis, other-wise the default axis=None will be considered. In this case, first the array is flattened to the 1-D array and deletion will be performed

A program to delete a single element of 1-D array at a specified index.

```
import numpy as np9
a = np9.arange(10,101,10)
b = np9.delete(a,3) # to delete the element present at 3rd index
print("array a : ",a)
print("array b : ",b)
```

concatenate()

The concatenate() function returns an ndarray of the provided type that satisfies requirements.

Syntax:

```
numpy.concatenate((a1, a2, ...), axis=0, out=None)
```

Example

```
>>> import numpy as np
>>> x = np.array([[3, 4], [5, 6]])
>>> y = np.array([[7, 8]])
>>> np.concatenate((x,y), axis=0)
array([[3, 4],
       [5, 6],
       [7, 8]])
>>> import numpy as np
>>> x = np.array([[3, 4], [5, 6]])
>>> y = np.array([[7, 8]])
>>> np.concatenate((x, y.T), axis=1)
```

```
array([[3, 4, 7],  
       [5, 6, 8]])
```