1. Write a NumPy program to convert a given array into a list and then convert it into an array again.

In [153]:

```
import numpy as np
```

In [154]:

```
A=np.arange(10)
print(str(type(A))+" : "+str(A))
A=A.tolist()
print(str(type(A))+" : "+str(A))
A=np.array(A)
print(str(type(A))+" : "+str(A))
```

```
<class 'numpy.ndarray'> : [0 1 2 3 4 5 6 7 8 9]
<class 'list'> : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
<class 'numpy.ndarray'> : [0 1 2 3 4 5 6 7 8 9]
```

2. Create a 5X2 integer array from a range between 100 to 200 such that the difference between each element is 10.

In [155]:

```
M=np.arange(100,200,10).reshape(5,2)
print(M)
```

```
[[100 110]
 [120 130]
 [140 150]
 [160 170]
 [180 190]]
```

3. Add the two 2D NumPy arrays and modify the resulting array by calculating the square root of each element

In [156]:

```
A=np.array([[1,2],[3,4]])
print("A=",A)
B=np.array([[5,6],[7,8]])
print("B=",B)
C=A+B
print("C=",C)
print("Calculating square root of each element")
C=np.sqrt(C)
print("C=",C)
```

```
A= [[1 2]
 [3 4]]
B= [[5 6]
 [7 8]]
C= [[ 6  8]
 [10 12]]
Calculating square root of each element
C= [[2.44948974 2.82842712]
 [3.16227766 3.46410162]]
```

4. Create an 8X3 integer array from a range between 10 to 34 such that the difference between each element is 1 and then Split the array into four equal-sized sub-arrays.

In [157]:

```
A=np.arange(10,34,1).reshape(8,3)
print(A)
A=np.split(A,4)
print(A)
```

```
[[10 11 12]
 [13 14 15]
 [16 17 18]
 [19 20 21]
 [22 23 24]
 [25 26 27]
 [28 29 30]
 [31 32 33]]
[array([[10, 11, 12],
       [13, 14, 15]]), array([[16, 17, 18],
       [19, 20, 21]]), array([[22, 23, 24],
       [25, 26, 27]]), array([[28, 29, 30],
       [31, 32, 33]])]
```

5. Consider the following array:

sampleArray = numpy.array([[34,43,73],[82,22,12],[53,94,66]])

i. Sort above array by second row
ii. Sort above array by second column
iii. Print max from axis 0 and min from axis 1
iv. Delete col 2 and insert new Column numpy.array([[10,10,10]]) in its place

In [158]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
print(sampleArray)
```

```
[[34 43 73]
 [82 22 12]
 [53 94 66]]
```

In [159]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
print(sampleArray)
sorted_array = sampleArray[np.argsort(sampleArray[:, 1])]
print(sorted_array)
```

```
[[34 43 73]
 [82 22 12]
 [53 94 66]]
[[82 22 12]
 [34 43 73]
 [53 94 66]]
```

In [160]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
print(sampleArray)
print(sampleArray.max(axis=0))
print(sampleArray.min(axis=1))
```

```
[[34 43 73]
 [82 22 12]
 [53 94 66]]
[82 94 73]
[34 12 53]
```

In [161]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray=np.delete(sampleArray,obj=1,axis=1)
sampleArray = np.insert(sampleArray, 1, [[10, 10, 10]],axis=1)
print(sampleArray)
```

```
[[34 10 73]
 [82 10 12]
 [53 10 66]]
```

6. Remove all the elements from an array that exist in another array

In [162]:

```
A=np.arange(10,20)
B=np.arange(1,15)
np.setdiff1d(A,B)
print(A)
```

```
[10 11 12 13 14 15 16 17 18 19]
```

7. Swap two columns in a 2d NumPy array

In [163]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray[:,[0, 1]] = sampleArray[:,[1, 0]]
print(sampleArray)
```

```
[[43 34 73]
 [22 82 12]
 [94 53 66]]
```

8. Swap two rows in a 2d NumPy array.

In [164]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray[[0, 1],:] = sampleArray[[1, 0],:]
print(sampleArray)
```

```
[[82 22 12]
 [34 43 73]
 [53 94 66]]
```

9. Reverse the order of rows of a 2D array

In [165]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray[:] = sampleArray[::-1]
print(sampleArray)
```

```
[[53 94 66]
 [82 22 12]
 [34 43 73]]
```

10. Reverse the order of columns of a 2D array

In [166]:

```
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray[:,:] = sampleArray[:,::-1]
print(sampleArray)
```

```
[[73 43 34]
 [12 22 82]
 [66 94 53]]
```

11. Retrieve common items between two python NumPy arrays?

In [167]:

```
A=np.arange(10,20)
B=np.arange(1,15)
print(np.intersect1d(A, B))
```

```
[10 11 12 13 14]
```

12. Retrive indices where elements of two arrays match

In [168]:

```
A=np.array([1,2,3,4,5,6,7,8])
B=np.array([1,1,2,3,5,6,9,8])
C=np.where(A == B)
print("Common indices: ",C[0])
```

```
Common indices:  [0 4 5 7]
```

13. Get all items between 5 and 10 from an array.

In [169]:

```
A=np.random.randint(1,11,20)
print("A= ",A)
print(A[np.logical_and(A>=5,A<=10)])
```

```
A=  [ 2  8  1  5 10  9 10 10  6  4  1  9  9  4 10  3  1  6 10  4]
[ 8  5 10  9 10 10  6  9  9 10  6 10]
```

14. For a 1D array with numeric values, find minimum, maximum, mean, median, standard deviation, 5th and 95th percentile, unique values, count of unique values, and the most frequent value.

In [170]:

```
A=np.random.randint(10,size=10)
print("A=",A)
print("Minimum: ",A.min())
print("Maximum: ",A.max())
print("Mean: ",A.mean())
print("Median: ",np.median(A))
print("Standard Deviation: ",A.std())
print("5th percentile: ",np.percentile(A,5))
print("9th percentile: ",np.percentile(A,95))
print("Unique values: ",np.unique(A))
print("Count of Unique values: ",np.unique(A).shape[0])
print("Most Frequent Value: ",np.bincount(A).argmax())
```

```
A= [5 6 8 0 1 6 2 2 0 7]
Minimum:  0
Maximum:  8
Mean:  3.7
Median:  3.5
Standard Deviation:  2.8653097563788803
5th percentile:  0.0
9th percentile:  7.549999999999999
Unique values:  [0 1 2 5 6 7 8]
Count of Unique values:  7
Most Frequent Value:  0
```

15. Write a NumPy program to create a 10x10 matrix, in which all the elements on the borders should be equal to 1, and rest others should be 0.

In [171]:

```
A=np.ones((10,10),dtype='int')
A[1:-1,1:-1]=0
print(A)
```

```
[[1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 0 0 0 0 0 0 0 0 1]
 [1 1 1 1 1 1 1 1 1 1]]
```

16. Write a NumPy program to create a 5x5 zero matrix with elements on the main diagonal equal to 1, 2, 3, 4, 5

In [174]:

```
A=np.diag(np.arange(1,6))
print(A)
```

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

17. Count the number of elements in a numpy array which are greater than 10. Further, multiply all such elements with value 10.

In [173]:

```
A=np.random.randint(1,21,10)
print(A)
for i in np.arange(A.shape[0]):
   if A[i]>10:
       A[i]*=10
print(A)
```

```
[19 16  2  3  6 14 10 17 17 10]
[190 160   2   3   6 140  10 170 170  10]
```

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
!cp  "/content/drive/MyDrive/Colab Notebooks/NumPy Exercises.ipynb" ./
!jupyter nbconvert --to html "NumPy Exercises"
```