ASSIGNMENT 01: Assignment on Practice of NumPy Library

Create a Numpy array containing the numbers from 1 to 10

In [1]:
```
import numpy as np
arr=np.arange(1,11)
print(arr)
```

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

Convert a given python list to numpy array

In [2]:
```
import numpy as np
my_list=[1,2,3,4,5]
my_array=np.array(my_list)
print(my_array)
```

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

Create 50 evenly spaced numbers between 1 and 10

In [3]:
```
import  numpy as np
my_array=np.linspace(1,10,50)
print(my_array)
```

```
[ 1.          1.18367347  1.36734694  1.55102041  1.73469388  1.91836735
  2.10204082  2.28571429  2.46938776  2.65306122  2.83673469  3.02040816
  3.20408163  3.3877551   3.57142857  3.75510204  3.93877551  4.12244898
  4.30612245  4.48979592  4.67346939  4.85714286  5.04081633  5.2244898
  5.40816327  5.59183673  5.7755102   5.95918367  6.14285714  6.32653061
  6.51020408  6.69387755  6.87755102  7.06122449  7.24489796  7.42857143
  7.6122449   7.79591837  7.97959184  8.16326531  8.34693878  8.53061224
  8.71428571  8.89795918  9.08163265  9.26530612  9.44897959  9.63265306
  9.81632653 10.        ]
```

create a 5X5 matrix which contains random samples from standard normal distribution

In [4]:
```
import numpy as np
matrix=np.random.normal(0,1,(5,5))
print(matrix)
```

```
[[-0.13090119 -0.52915085 -1.27446489 -0.56230264 -0.26462096]
 [ 1.21907352  1.18424062 -1.90818347 -0.96528377 -0.73085963]
 [-0.17711287  0.39457746  0.00804115  0.76743507 -1.13530414]
 [ 0.87886451 -1.10269832 -0.70209975 -0.58849543  0.07141402]
 [ 0.91361729  0.24076211  1.56225522 -0.73524895  1.36505296]]
```

Create 20 random integer numbers between 1 to 100 as a numpy array

In [5]:
```
import numpy as np
rndm_int=np.random.randint(1,101,20)
rndm_int
```

Out[5]:
```
array([10, 22, 49,  1, 93,  4, 84, 70, 95, 80, 90, 88, 25, 25,  8, 26, 45,
       81, 64, 69])
```

Given the numpy array 'arr' reverse its elements and find its size

In [6]:
```
import numpy as np
arr=np.array([1,2,3,4,5])
rev_arr=np.flip(arr,0)
size_rev_arr=rev_arr.size
print("Original ",arr)
```

```
print("Reversed ",rev_arr)
print("Original ",size_rev_arr)
```

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

Find the mean,median and standard deviation of the following numpy array

In [7]:
```
import numpy as np
arr=np.array([5,10,15,20,25])
mean=np.mean(arr)
median=np.median(arr)
std_deviation=np.std(arr)
print(arr)
print('Mean',mean)
print('Median',median)
print('Standard Deviation',std_deviation)
```

```
[ 5 10 15 20 25]
Mean 15.0
Median 15.0
Standard Deviation 7.0710678118654755
```

Create 3X3 matrix with all values set to 1

In [8]:
```
import numpy as np
matrix=np.ones((3,3))
print(matrix)
```

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

Create 3X3 matrix with all values set to 0

In [9]:
```
import numpy as np
matrix=np.zeros((3,3))
print(matrix)
```

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

Given two numpy arrays arr1 and arr2 concatenate them horizontally

In [10]:
```
import numpy as np
arr1=np.array([1,2,3,4,5])
arr2=np.array([6,7,8,9,10])
res=np.concatenate((arr1.reshape(1,-1),arr2.reshape(1,-1)),axis=1)
print(res)
```

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

Create a numpy array containing all even numbers from 0 to 20

In [11]:
```
import numpy as np
arr1=np.arange(0,21,2)
print(arr1)
```

```
[ 0  2  4  6  8 10 12 14 16 18 20]
```

Element-wise multiplication of two numpy arrays

In [12]:
```
import numpy as np
arr1=np.array([1,2,3,4,5])
arr2=np.array([6,7,8,9,10])
res=arr1*arr2
print(res)
```

```
[ 6 14 24 36 50]
```

Reshape a numpy array into a 2X3 matrix

In [13]:
```
import numpy as np
arr1=np.array([1,2,3,4,5,6])
reshape=arr1.reshape(2,3)
print(reshape)
```

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

Find the maximum and minimum values in a numpy array

In [14]:
```
import numpy as np
arr1=np.array([1,2,3,4,5,6])
print("maximum ",np.max(arr1))
print("maximum ",np.min(arr1))
```

```
maximum  6
maximum  1
```

Calculate the dot product of two numpy arrays

In [15]:
```
import numpy as np
arr1=np.array([1,2,3,4,5])
arr2=np.array([6,7,8,9,10])
dot_product=np.dot(arr1,arr2)
print(dot_product)
```

```
130
```

Create 2D numpy array with random floating point numbers between 0 to 1

In [16]:
```
import numpy as np
random_arr=np.random.rand(3,4)
print(random_arr)
```

```
[[0.83140146 0.58817119 0.34460686 0.69393307]
 [0.38158844 0.94324199 0.02596882 0.58110464]
 [0.9648661  0.02218178 0.88425532 0.55922529]]
```

Transpose a 2D numpy array

In [17]:
```
import numpy as np
arr1=np.array([[1,2,3,4,5],[6,7,8,9,10]])
print("Transpose matrix is ",np.transpose(arr1))
```

```
Transpose matrix is  [[ 1  6]
 [ 2  7]
 [ 3  8]
 [ 4  9]
 [ 5 10]]
```

Create slice and set value 100

In [18]:
```
import numpy as np
arr1=np.array([1,2,3,4,5,6,7,8,9])
slice_arr=arr1[0:6]
slice_arr[:]=100
print(slice_arr)
```

```
[100 100 100 100 100 100]
```

Slice the first 2 rows and the last 2 cols of 2D numpy array (right top corner)

In [19]:
```
import numpy as np
arr=np.array([[1,2,3],[4,5,6],[7,8,9]])
```

```
sliced_arr=arr[:2,1:]
print(sliced_arr)
```

```
[[2 3]
 [5 6]]
```

Given a numpy array of numbers how can you filter out all values greater than 5

In [20]:
```
import numpy as np
arr1=np.array([1,2,3,4,5,6,7,8,9])
print("Number greater than 5 ",arr1[arr1>=5])
```

```
Number greater than 5  [5 6 7 8 9]
```

Using boolean indexing, Extract all even numbers from a numpy array

In [21]:
```
import numpy as np
arr1=np.array([1,2,3,4,5,6,7,8,9,10])
print("Even numbers ",arr1[arr1%2==0])
```

```
Even numbers  [ 2  4  6  8 10]
```