TAKE HOME LAB EXERCISE - 01

In [1]:
```
import numpy as np
import pandas as pd
```

(1).Create a 1D array with 10 values in it and then convert the 1D array to 2D array ?

In [3]:
```
np1 = np.arange(10)
print(np1)
np1_res = np1.reshape(5,2)
print(np1_res)
```

```
[0 1 2 3 4 5 6 7 8 9]
[[0 1]
 [2 3]
 [4 5]
 [6 7]
 [8 9]]
```

(2).Print or show only 3 decimal places of the numpy array `random_arr`

In [4]:
```
np2 = np.random.rand(5)
print('The array before round off: ',np2)

np2_roundoff =  np.around(np2, decimals = 3)
print('The array after round off to 3 decimal places is: ',np2_roundoff)
```

```
The array before round off:  [0.66064821 0.91316653 0.09611134 0.22702573 0.712
13718]
The array after round off to 3 decimal places is:  [0.661 0.913 0.096 0.227 0.7
12]
```

(3).The number of items printed to be limit in python numpy array a to a maximum of 6 elements.

```
 Intake : x = np.arange(15)
          array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 1
3, 14])

 Result : ([ 0,  1,  2, ..., 12, 13, 14])
```

In [ ]:

---

(4). Take two arrays `x` and `y` and Stack the `x` and `y` arrays Horizontal.

```
Intake:
            x = np.arange(10).reshape(2,-1)

            y = np.repeat(1, 10).reshape(2,-1)


Result:       ([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],
                [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])
```

In [4]:
```
x = np.arange(10).reshape(2,-1)
y = np.repeat(1, 10).reshape(2,-1)
nvs = np.hstack(tup=(x,y))
print(nvs)
```

```
[[0 1 2 3 4 1 1 1 1 1]
 [5 6 7 8 9 1 1 1 1 1]]
```

---

(5). Consider the given array's and print the values which are common in both the arrays.

```
Intake:

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

Results:

     ([2, 4])
```

In [5]:
```
x = np.array([1,2,3,2,3,4,3,4,5,6])
y = np.array([7,2,10,2,7,4,9,4,9,8])
print(np.intersect1d(x,y))
```

```
[2 4]
```

---

(6). There are two array x and y ,drop the items in `X` which already exist in `Y` .

INTAKE:

```
x = np.array([1,2,3,4,5])
y = np.array([5,6,7,8,9])
```

RESULT:

```
([1,2,3,4])
```

In [10]:
```
x = np.array([1,2,3,4,5])
y = np.array([5,6,7,8,9])
res = np.setdiff1d(x,y)
print(res)
```

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

---

(7). Reverse the rows of a 2D array.

Intake : `X = np.arange(9).reshape(3,3)`

Result :

```
([[6, 7, 8],
  [3, 4, 5],
  [0, 1, 2]])
```

In [26]:
```
X = np.arange(9).reshape(3,3)
np.flip(X,axis=0)
```

Out[26]:
```
array([[6, 7, 8],
       [3, 4, 5],
       [0, 1, 2]])
```

---

(8). Reverse the columns of a 2D array.

```
Intake : X = np.arange(9).reshape(3,3)

Result :     ([[2, 1, 0],
               [5, 4, 3],
               [8, 7, 6]])
```

In [25]:
```
X = np.arange(9).reshape(3,3)
np.flip(X,axis=1)
```

Out[25]:
```
array([[2, 1, 0],
       [5, 4, 3],
       [8, 7, 6]])
```

---

(9). Print the ranks for the given numeric array `x` .

```
Intake:
np.random.seed(10)
x = np.random.randint(20, size=10)
print(x)
[ 9  4 15  0 17 16 17  8  9  0]


Result:
[4 2 6 0 8 7 9 3 5 1]
```

In [18]:
```
#array = numpy.array([4,2,7,1])
#temp = array.argsort()
#ranks = numpy.empty_like(temp)
#ranks[temp] = numpy.arange(len(array))

np.random.seed(10)
x = np.random.randint(20, size=10)
print(x,'\n')
temp = x1d.argsort()
ranks = np.arange(len(x1d))[temp.argsort()]
print('The ranks of the given array is:')
print(ranks)
```

```
[ 9  4 15  0 17 16 17  8  9  0]

The ranks of the given array is:
[4 2 6 0 8 7 9 3 5 1]
```

---

(10). For the given numeric array print the rank array with same shape of numeric array.

```
Intake:
np.random.seed(10)
x = np.random.randint(20, size=[2,5])
print(a)
[[ 9  4 15  0 17]
[16 17  8  9  0]]

Result:
[[4 2 6 0 8]
[7 9 3 5 1]]
```

In [14]:
```
np.random.seed(10)
x = np.random.randint(20, size=[2,5])
x1d = x.flatten()
temp = x1d.argsort()
ranks = np.arange(len(x1d))[temp.argsort()]
res = ranks.reshape(2,5)
print(res)
```

```
[[4 2 6 0 8]
 [7 9 3 5 1]]
```

---

(11). Subtract the 1d array y_1d from the 2d array x_2d, such that each item of y_1d subtracts from respective row of x_2d.

```
Intake:
x_2d = np.array([[3,3,3],[4,4,4],[5,5,5]])
y_1d = np.array([1,2,3])

Result:
 [[2 2 2]
 [2 2 2]
 [2 2 2]]
```

In [31]:
```
x_2d = np.array([[3,3,3],[4,4,4],[5,5,5]])
y_1d = np.array([1,2,3])
x_2d - y_1d[:,None]
```

Out[31]:
```
array([[2, 2, 2],
       [2, 2, 2],
       [2, 2, 2]])
```

----------HAPPY LEARNING----------