```python
In [ ]:  #creating 1D array of numbers from 0 to 9.

In [1]:  import numpy as np

In [2]:  array1 = np.array([0,1,2,3,4,5,6,7,8,9])
         print(array1)

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

In [ ]:  #create 3x3 NumPy array of all boolean true values.

In [3]:  a = np.ones((3,3), dtype=bool)
         print(a)

         [[ True  True  True]
          [ True  True  True]
          [ True  True  True]]

In [ ]:  #extract all odd numbers from array of 1-10.

In [4]:  array2 = np.array([1,2,3,4,5,6,7,8,9,10])
         print(array2[0::2])

         [1 3 5 7 9]

In [ ]:  #replace all odd numbers in an array of 1-10 with the value -1.

In [9]:  array3 = np.array([1,2,3,4,5,6,7,8,9,10])
         new = array3[0::2] = -1
         print(array3)

         [-1  2 -1  4 -1  6 -1  8 -1 10]

In [ ]:  #convert a 1D array to a 2D array with 2 rows.

In [11]: array4 = np.array([[1,2,3,4,5],[6,7,8,9,10]])
         print(array4)

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

In [ ]:  #create 2 arrays a and b, stack these two arrays vertically use the np.dot and np.sumr to calculate totals.

In [12]: a = np.arange(1,10).reshape(3,3)
         b = np.arange(11,20).reshape(3,3)
         c = np.dot(a,b)
         print(c)

         e= np.sum(c,axis = 1)
         print(e)

         [[ 90  96 102]
          [216 231 246]
          [342 366 390]]
         [ 288  693 1098]

In [ ]:  #create a pattern without hardcoding using only numpy numbers.

In [13]: a = np.array([1,1,1,2,2,2,3,3,3,1,2,3,1,2,3,1,2,3])
         a

Out[13]: array([1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3])

In [ ]:  #In two arrays a (1,2,3,4,5) and b (4,5,6,7,8,9), remove all repeating items present in array b.

In [20]: a = np.array([1,2,3,4,5])
         b = np.array([4,5,6,7,8,9])
         np.delete(b,[0,1])

Out[20]: array([6, 7, 8, 9])

In [ ]:  #Get all items between 5 and 10 from a and b and sume them together

In [21]: a = np.array([1,2,3,4,5])
         b = np.array([6,7,8,9])
         c = np.concatenate((a,b))
         d = c[4:9]
         print(d)

         [5 6 7 8 9]

In [22]: e = np.sum(d)
         print(e)

         35

In [ ]:
```