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

In [5]:
```
# 1. Create an empty and  a full Numpy array
A =np.empty((4,5))
print('Empty Numpy Array: ') ; print(A); print('\n')
B= np.arange(1,11).reshape(5,2)
print('Full Numpy Array: ') ; print(B); print('\n')
```

```
Empty Numpy Array:
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]


Full Numpy Array:
[[ 1  2]
 [ 3  4]
 [ 5  6]
 [ 7  8]
 [ 9 10]]
```

In [6]:
```
# create a numpy array filled with zeros

ZERO_ARRAY = np.zeros(20). reshape(4,5)
print('Numpy array filled with Zeros :');
print(ZERO_ARRAY)
```

```
Numpy array filled with Zeros :
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

In [8]:
```
# create a numpy array filled with all ones

ONE_ARRAY = np.ones(10)

print(ONE_ARRAY)

ONE_ARRAY_2d = np.ones(10).reshape(5,2)
print(ONE_ARRAY_2d)
```

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

In [39]:
```
# check where a NUMPY array contains a specific row

ARRAY = np.arange(1,51).reshape(5,10)
print('ARRAY:\n',ARRAY); print('\n')


row_checked = np.arange(2,13)
```

```
print('rowto be checked : ', row_checked) ; print('\n')

if len(row_checked)== len(ARRAY[0]):
    row_exists = np.any(np.all(ARRAY == row_checked, axis =1))

    if row_exists:
        print(f'{row_checked} is available in the numpy Array')
    else:
        print(f'{row_checked} is not available in NUMPY ARRAY')
elif len(row_checked) != len(ARRAY[0]):
    print('length of the row to be checked is ' , len(row_checked))
    print('length of the row of the ARRAY is ' , len(ARRAY[0]))
    print('The length of the row checked does not match the length of the ARRAY row')
```

```
ARRAY:
 [[ 1  2  3  4  5  6  7  8  9 10]
 [11 12 13 14 15 16 17 18 19 20]
 [21 22 23 24 25 26 27 28 29 30]
 [31 32 33 34 35 36 37 38 39 40]
 [41 42 43 44 45 46 47 48 49 50]]


rowto be checked :  [ 2  3  4  5  6  7  8  9 10 11 12]


length of the row to be checked is  11
length of the row of the ARRAY is  10
The length of the row checked does not match the length of the ARRAY row
```

In [6]:

```
# How to remove row s in Numpy arrays containing non numeric values

import numpy as np
ARRAY = np.array(
                [
                    [1,2,3,4],['python',5,6,7],
                    [8,9,'NUMPY',10],[11,12,13,14]
                ]
                )

print('Original Array : '); print(ARRAY); print('\n')
def remove_nonnumeric_values(ROW):
    try :
        ROW.astype(float)
        return True
    except ValueError :
        return False

MASK  = np.array( [remove_nonnumeric_values(ROW) for ROW in ARRAY]  )

NUMERIC_ARRAY  = ARRAY[MASK]
print('ARRAY containing only numeric values :')
print(NUMERIC_ARRAY)
```

```
Original Array :
[['1' '2' '3' '4']
 ['python' '5' '6' '7']
 ['8' '9' 'NUMPY' '10']
 ['11' '12' '13' '14']]


ARRAY containing only numeric values :
[['1' '2' '3' '4']
 ['11' '12' '13' '14']]
```

In [36]:
```
# Remove single dimention entries from the shape of the ARRAY

A = np.array([
                [
                    #[1, 2, 3],
                    [21,22,23]
                ],
                [
                    #[4, 5, 6],
                    [31,32,33]

                ],
                [
                    #[7, 8, 9],
                    [10,11,12]
                ]
            ])
print('Original Array') ;print(A) ; print('Shape of the  Array', A.shape); print('\n')

new = np.squeeze(A)
print("After squeezing ")
print(new); print(new.shape)
```

```
Original Array
[[[21 22 23]]

 [[31 32 33]]

 [[10 11 12]]]
Shape of the  Array (3, 1, 3)


[[21 22 23]
 [31 32 33]
 [10 11 12]]
(3, 3)
```

In [8]:
```
#  Find the number of occurances of a row in NUMPY ARRAY

import numpy as np
from collections import Counter as c

N = np.arange(1,11).reshape(5,2)

def row_counting_function(ARRAY) :
    LIST_ROWS  = [tuple(element) for element in ARRAY ] # rows must be converted into
    print(LIST_ROWS);
    count = c(LIST_ROWS) # from collections import Counter as c
    return count
```

```
RESULT = row_counting_function(N)
print('RESULT =', RESULT)
```

```
[(1, 2), (3, 4), (5, 6), (7, 8), (9, 10)]
RESULT = Counter({(1, 2): 1, (3, 4): 1, (5, 6): 1, (7, 8): 1, (9, 10): 1})
```

In [8]:
```
# Find the most frequent value in NUMPY
#METHOD -1

import numpy as np

N = np.array([
                [11,22,33,44],
                [22,33,22,55],
                [44,22,33,44]

            ])
print('Shape of the ARRAY is ', N.shape)

N_ravel = np.ravel(N)
print( 'The list of elements in the Array : ' ,  N_ravel)


D = {}
for element in N_ravel:
    if element in D :
        D[element] +=1
    else :
        D[element] = 1

print('the frequencies  Dictionary D:') ; print(D) ; print('\n')

for i in D.items():
    print(i[0] ,  i[1])

print('\n')
print('The element with the maximum occurances is ' , max(D , key = D.get) , 'which is
```

```
Shape of the ARRAY is  (3, 4)
The list of elements in the Array :  [11 22 33 44 22 33 22 55 44 22 33 44]
the frequencies  Dictionary D:
{11: 1, 22: 4, 33: 3, 44: 3, 55: 1}


11 1
22 4
33 3
44 3
55 1


The element with the maximum occurances is  22 which is  4
```

In [29]:
```
# Find the most frequent value in NUMPY
#METHOD -2

from collections import Counter  as C
import numpy as np
```

```
 # ------------------------------------------------
A = np.array([
                    [101,201,301],
                    [201,301,401],
                    [301,401,501]
            ])


LIST_A = np.ravel( A.tolist())
print('A array as List' ,LIST_A )  # the list must  be in 1 D only for Counter lib to

counter_list = C(LIST_A)

print('The List in Counter format :' , counter_list)
MAX_VALUE = 0
for element in  counter_list :
    if  MAX_VALUE < counter_list[element] :
        MAX_VALUE = counter_list[element]
print('max_value = ' , MAX_VALUE)
KEY = [k for k in counter_list if counter_list[k] == MAX_VALUE]
ELEMENT = KEY[0]
print(f'The element in the {ELEMENT} which occurs {MAX_VALUE} times' )
```

```
A array as List [101 201 301 201 301 401 301 401 501]
The List in Counter format : Counter({301: 3, 201: 2, 401: 2, 101: 1, 501: 1})
max_value =  3
The element in the 301 which occurs 3 times
```

In [1]:

```
# Combine a one Dimention and two dimention Array
import numpy as np

ONE_D = np.arange(1,6)

TWO_D = np.arange(6,16).reshape(5,2)

print('one d array :\n', ONE_D )
print('two d array : \n',TWO_D )

TWO_D_TO_ONE_D = TWO_D.ravel()

print(f" convereted 2 D to 1 D : , {TWO_D_TO_ONE_D}")

Result_Horizontal_Hstack = np.hstack([ONE_D , TWO_D_TO_ONE_D])
Result_Horizontal_Append =np.append( ONE_D , TWO_D_TO_ONE_D , axis = 0 )


print('Horizontally stacked Array =' , Result_Horizontal_Hstack)
print('Horizontally appended Array =' , Result_Horizontal_Append)
```

```
one d array :
 [1 2 3 4 5]
two d array :
 [[ 6  7]
 [ 8  9]
 [10 11]
 [12 13]
 [14 15]]
 convereted 2 D to 1 D : , [ 6  7  8  9 10 11 12 13 14 15]
Horizontally stacked Array = [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
Horizontally appended Array = [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

In [ ]:

In [ ]: