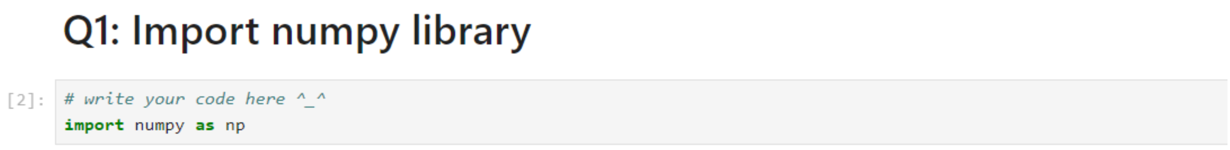
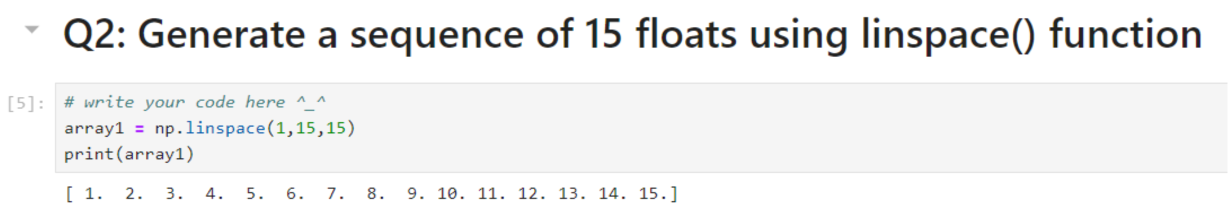
NumPy_Assignment2

Task 1&2:



Q1: Import numpy library

```
[2]: # write your code here ^_^
     import numpy as np
```

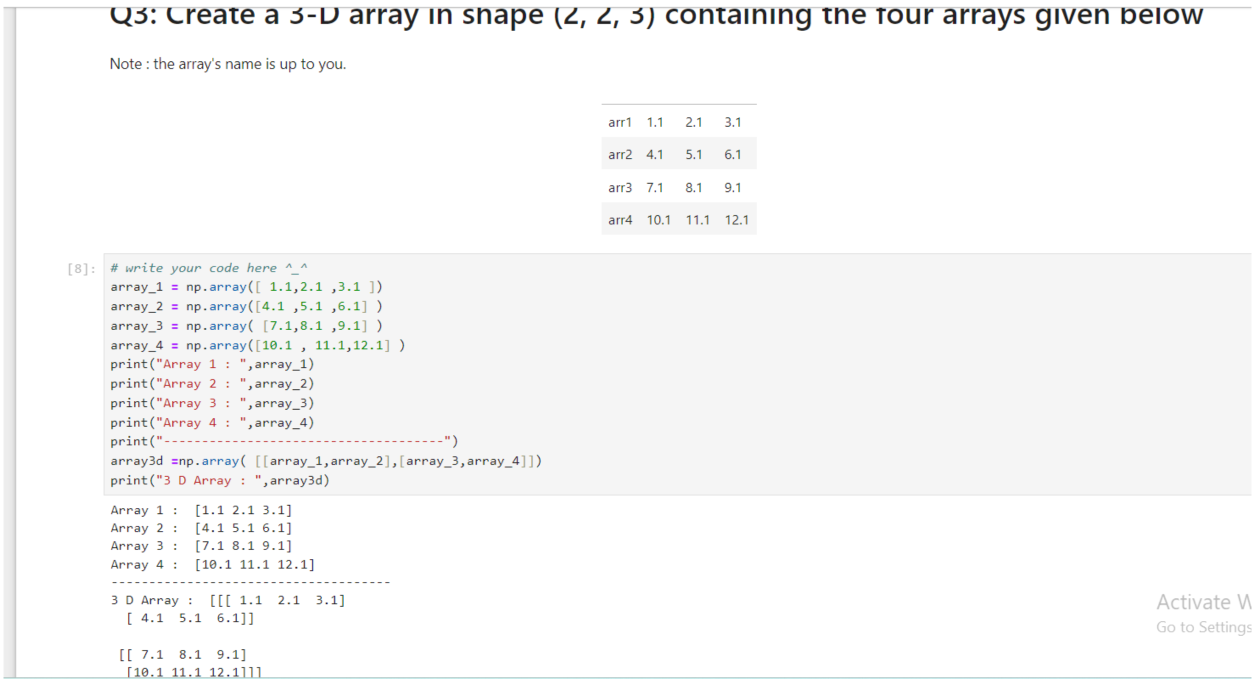


Q2: Generate a sequence of 15 floats using linspace() function

```
[5]: # write your code here ^_^
     array1 = np.linspace(1,15,15)
     print(array1)
```

```
[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14. 15.]
```

Task 3 :



Q3: Create a 3-D array in shape (2, 2, 3) containing the four arrays given below

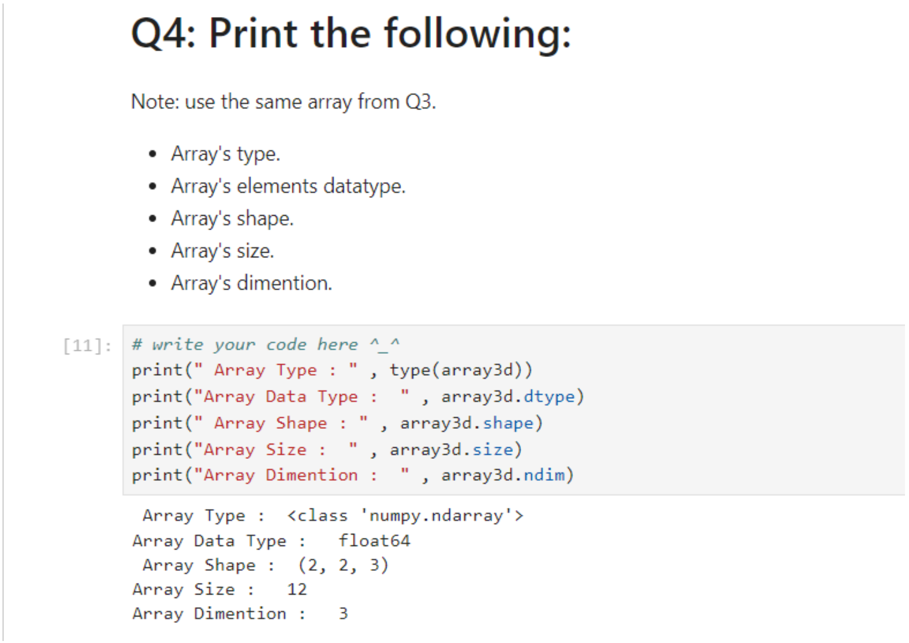
Note : the array's name is up to you.

arr1	1.1	2.1	3.1
arr2	4.1	5.1	6.1
arr3	7.1	8.1	9.1
arr4	10.1	11.1	12.1

```
[8]: # write your code here ^_^
     array_1 = np.array([ 1.1,2.1 ,3.1 ])
     array_2 = np.array([4.1 ,5.1 ,6.1] )
     array_3 = np.array( [7.1,8.1 ,9.1] )
     array_4 = np.array([10.1 , 11.1,12.1] )
     print("Array 1 : ",array_1)
     print("Array 2 : ",array_2)
     print("Array 3 : ",array_3)
     print("Array 4 : ",array_4)
     print("------------------------------------")
     array3d =np.array( [[array_1,array_2],[array_3,array_4]])
     print("3 D Array : ",array3d)
```

```
Array 1 :  [1.1 2.1 3.1]
Array 2 :  [4.1 5.1 6.1]
Array 3 :  [7.1 8.1 9.1]
Array 4 :  [10.1 11.1 12.1]
------------------------------------
3 D Array :  [[[ 1.1  2.1  3.1]
  [ 4.1  5.1  6.1]]

 [[ 7.1  8.1  9.1]
  [10.1 11.1 12.1]]]
```

Task 4 :



Q4: Print the following:

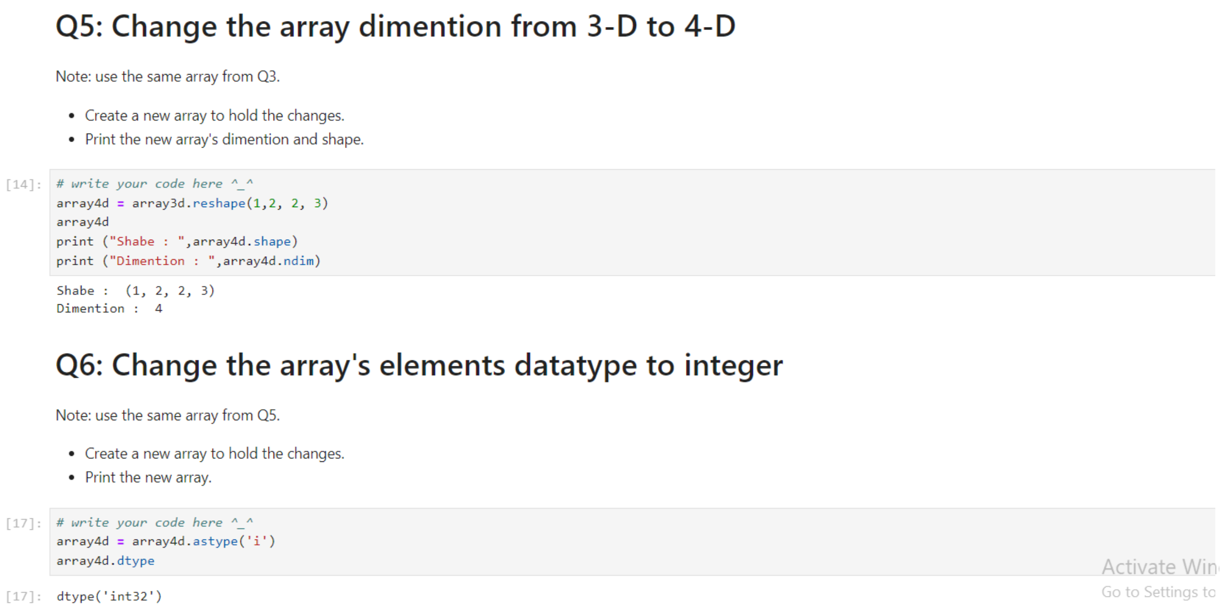
Note: use the same array from Q3.

- Array's type.
- Array's elements datatype.
- Array's shape.
- Array's size.
- Array's dimention.

```
[11]: # write your code here ^_^
      print(" Array Type : " , type(array3d))
      print("Array Data Type :  " , array3d.dtype)
      print(" Array Shape : " , array3d.shape)
      print("Array Size :  " , array3d.size)
      print("Array Dimention :  " , array3d.ndim)
```

```
 Array Type :  <class 'numpy.ndarray'>
Array Data Type :   float64
 Array Shape :  (2, 2, 3)
Array Size :   12
Array Dimention :   3
```

Task 5 & 6:



Q5: Change the array dimention from 3-D to 4-D

Note: use the same array from Q3.

- Create a new array to hold the changes.
- Print the new array's dimention and shape.

```
[14]: # write your code here ^_^
      array4d = array3d.reshape(1,2, 2, 3)
      array4d
      print ("Shabe : ",array4d.shape)
      print ("Dimention : ",array4d.ndim)
```

```
Shabe :  (1, 2, 2, 3)
Dimention :  4
```

Q6: Change the array's elements datatype to integer

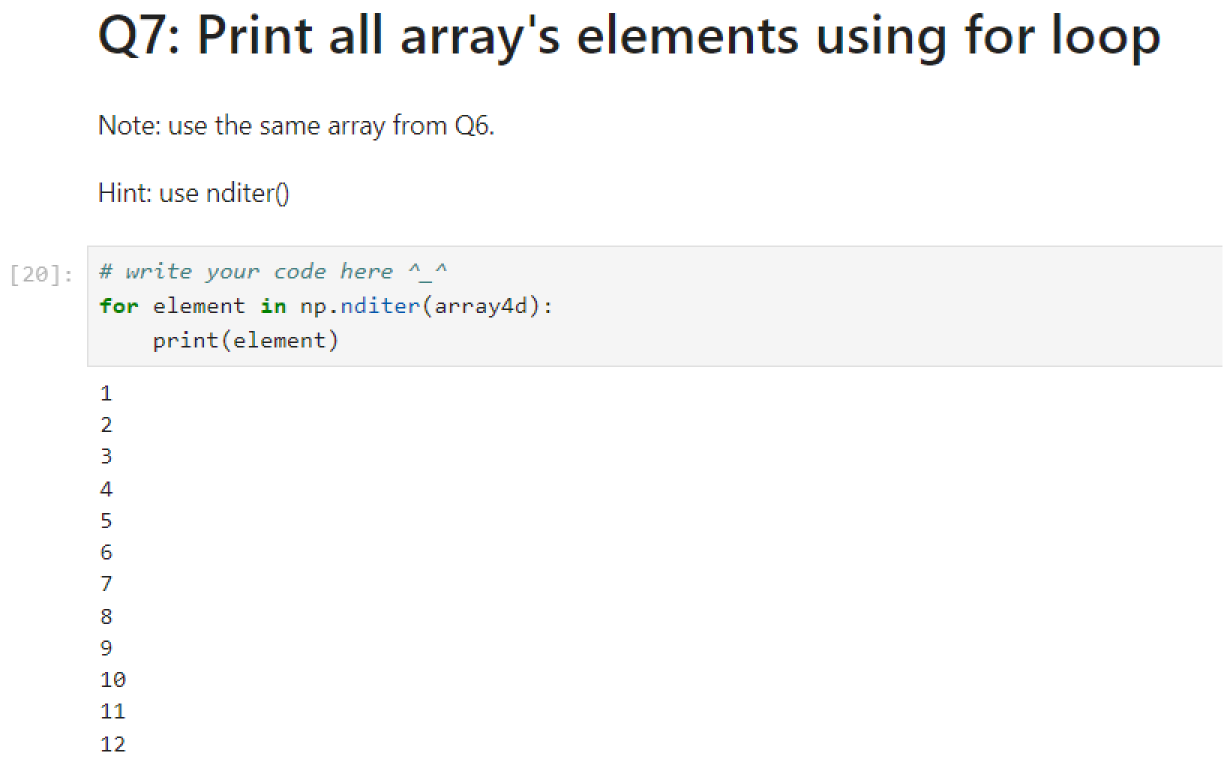
Note: use the same array from Q5.

- Create a new array to hold the changes.
- Print the new array.

```
[17]: # write your code here ^_^
      array4d = array4d.astype('i')
      array4d.dtype
```

```
[17]: dtype('int32')
```

Task 7 :



Q7: Print all array's elements using for loop

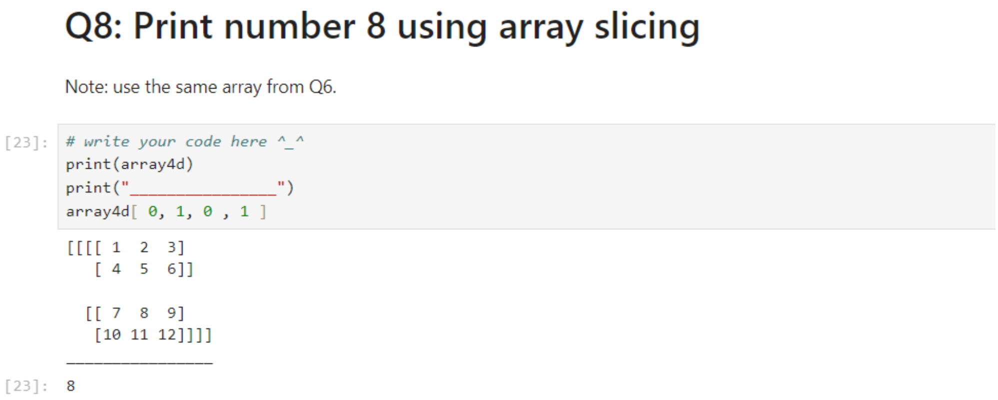
Note: use the same array from Q6.

Hint: use nditer()

```
[20]: # write your code here ^_^
      for element in np.nditer(array4d):
          print(element)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
```

Task 8 & 9:

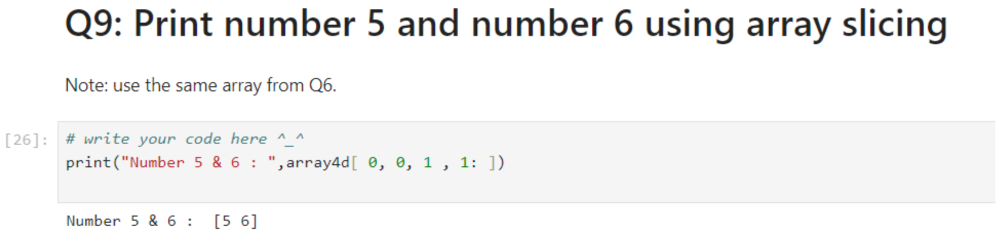


Q8: Print number 8 using array slicing

Note: use the same array from Q6.

```
[23]: # write your code here ^_^
      print(array4d)
      print("________________")
      array4d[ 0, 1, 0 , 1 ]
```

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

  [[ 7  8  9]
   [10 11 12]]]]
________________
[23]: 8
```



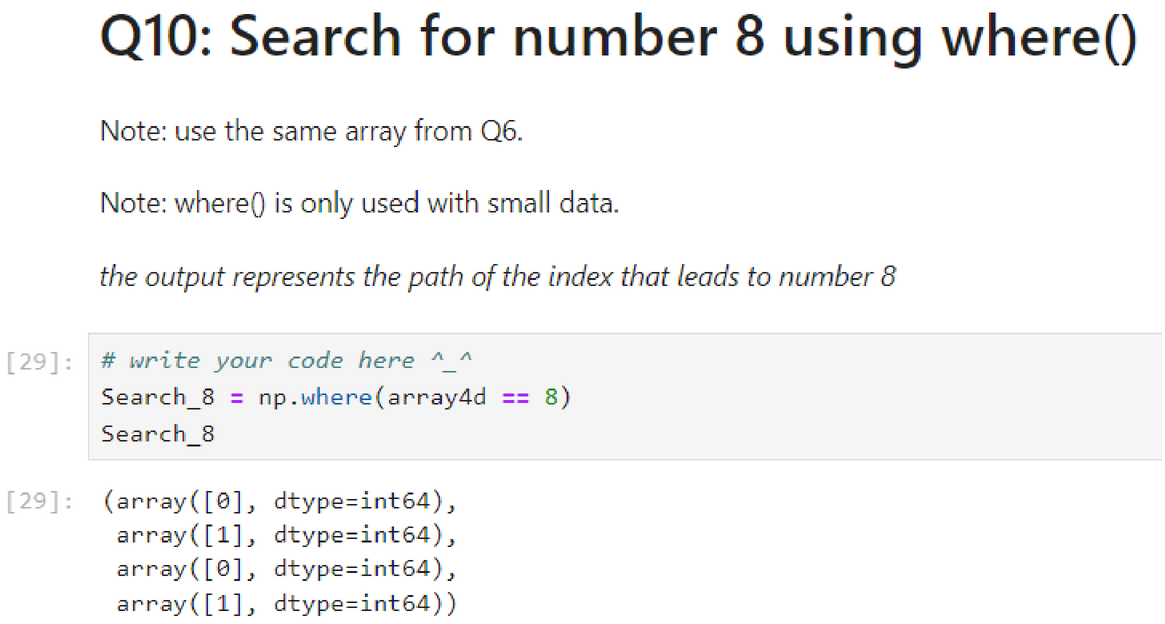
Q9: Print number 5 and number 6 using array slicing

Note: use the same array from Q6.

```
[26]: # write your code here ^_^
      print("Number 5 & 6 : ",array4d[ 0, 0, 1 , 1: ])
```

```
Number 5 & 6 :  [5 6]
```

Task 10 :



Q10: Search for number 8 using where()

Note: use the same array from Q6.

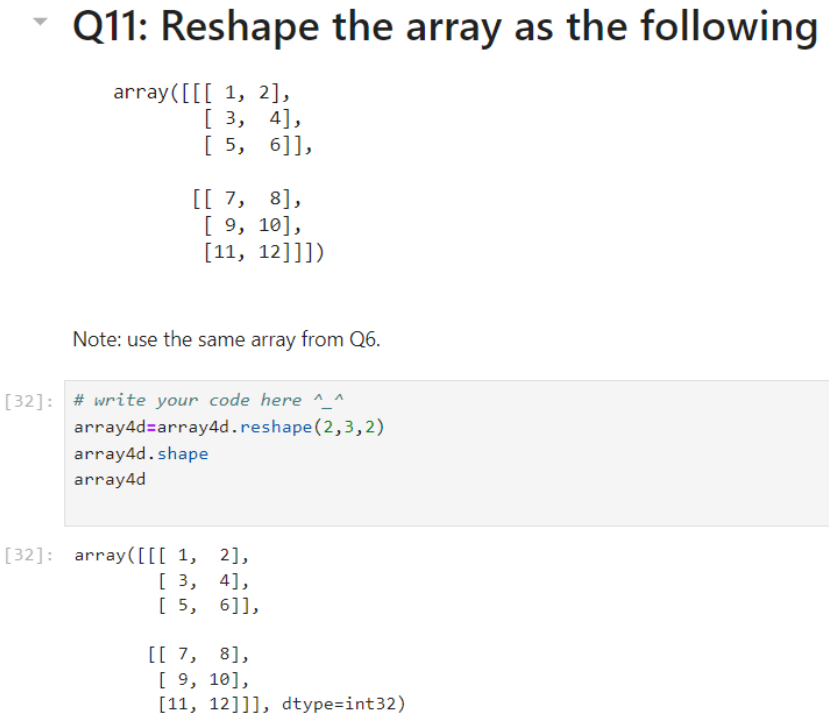
Note: where() is only used with small data.

the output represents the path of the index that leads to number 8

```
[29]: # write your code here ^_^
      Search_8 = np.where(array4d == 8)
      Search_8
```

```
[29]: (array([0], dtype=int64),
       array([1], dtype=int64),
       array([0], dtype=int64),
       array([1], dtype=int64))
```

Task 11 :



Q11: Reshape the array as the following

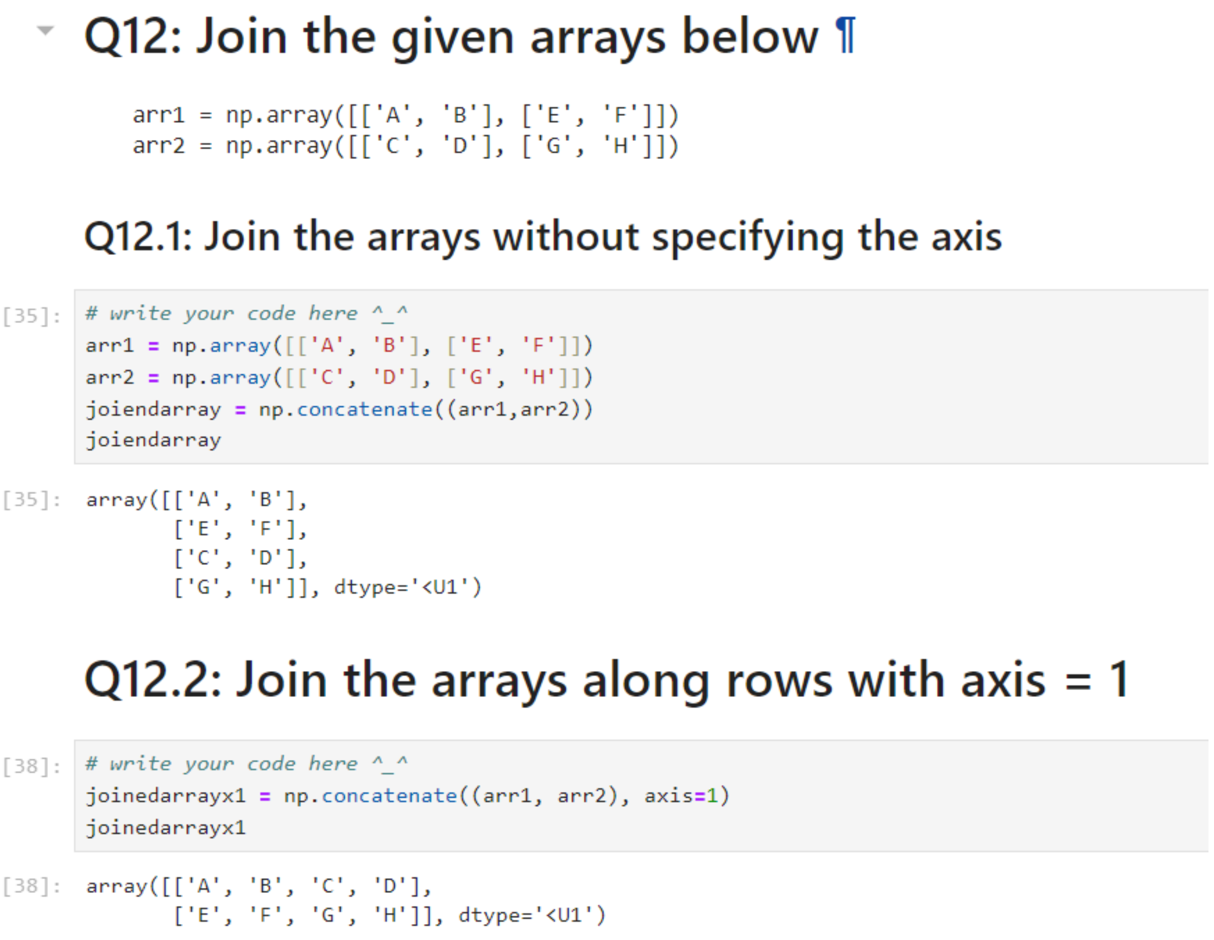
```
array([[[ 1, 2],
        [ 3,  4],
        [ 5,  6]],

       [[ 7,  8],
        [ 9, 10],
        [11, 12]]])
```

Note: use the same array from Q6.

```
[32]: # write your code here ^_^
      array4d=array4d.reshape(2,3,2)
      array4d.shape
      array4d
```

```
[32]: array([[[ 1,  2],
              [ 3,  4],
              [ 5,  6]],

             [[ 7,  8],
              [ 9, 10],
              [11, 12]]], dtype=int32)
```

Task 12 :



Q12: Join the given arrays below ¶

```
arr1 = np.array([['A', 'B'], ['E', 'F']])
arr2 = np.array([['C', 'D'], ['G', 'H']])
```

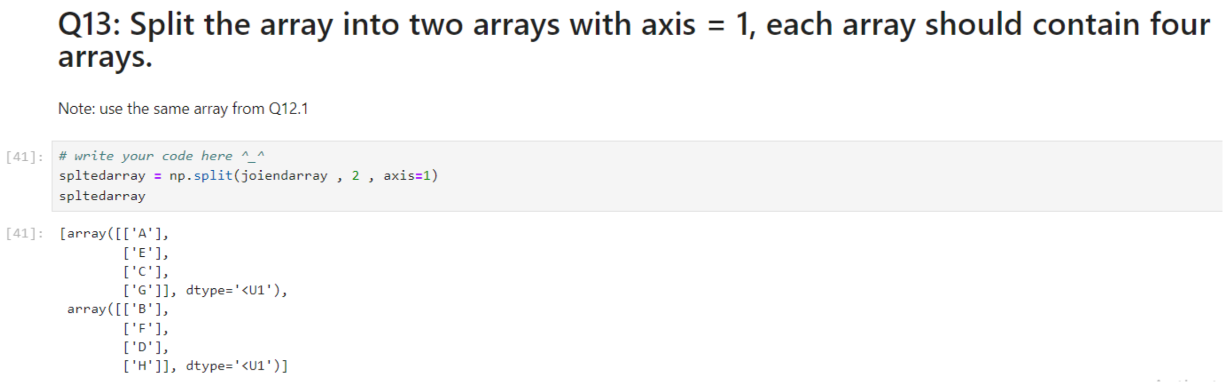
Q12.1: Join the arrays without specifying the axis

```
[35]: # write your code here ^_^
      arr1 = np.array([['A', 'B'], ['E', 'F']])
      arr2 = np.array([['C', 'D'], ['G', 'H']])
      joiendarray = np.concatenate((arr1,arr2))
      joiendarray

[35]: array([['A', 'B'],
             ['E', 'F'],
             ['C', 'D'],
             ['G', 'H']], dtype='<U1')
```

Q12.2: Join the arrays along rows with axis = 1

```
[38]: # write your code here ^_^
      joinedarrayx1 = np.concatenate((arr1, arr2), axis=1)
      joinedarrayx1

[38]: array([['A', 'B', 'C', 'D'],
             ['E', 'F', 'G', 'H']], dtype='<U1')
```

Task 13 :



Q13: Split the array into two arrays with axis = 1, each array should contain four arrays.

Note: use the same array from Q12.1

```
[41]: # write your code here ^_^
      spltedarray = np.split(joiendarray , 2 , axis=1)
      spltedarray

[41]: [array([['A'],
             ['E'],
             ['C'],
             ['G']], dtype='<U1'),
       array([['B'],
             ['F'],
             ['D'],
             ['H']], dtype='<U1')]
```