

NumPy_Assignment1

Task 1&2&3:

Q#1: Create x array with elements equal to 1.

```
[4]: # write your code here ^_^
x = np.ones(5)
print(x)

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

Q#2: Create y array with elements equal to 0.

```
[448]: # write your code here ^_^
y = np.zeros(5)
print(y)

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

Q#3: Add x and y arrays.

```
[451]: # write your code here ^_^
x+y

[451]: array([1.,  1.,  1.,  1.,  1.])
```

Activate Windows
Go to Settings to activate Windows.

Task 4 & 5 :

Q#4: Print x array characteristics (e.g: dimension, shape, size, type).

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

Dimension : 1
Shape : (5,)
Size : 5
Data Type : float64
Array Type : <class 'numpy.ndarray'>
```

Q#5: Create a 2D array "called w" as the following:

11	12
13	14
15	16

```
[22]: # write your code here ^_^
w = np.array([[11,12],[13,14],[15,16]])
print(w)

[[11 12]
 [13 14]
 [15 16]]
```

Activate Windows
Go to Settings to activate Windows.

Task 6 & 7:

Q#6: Create z array contains the numbers from 1 to 3.

```
[460]: # write your code here ^_^
z = np.arange(1,4)
print(z)
[1 2 3]
```

Q#7: Combine the arrays z and w in horizontal way then save it in a new variable "newArray".

```
[463]: # write your code here ^_^
zzz.reshape(-1,1)
print("Array z :""\n",z)
print("Array w :""\n",w)
newArray = np.hstack((z, w))
print("New Array : ""\n", newArray)

Array z :
[[1]
 [2]
 [3]]
Array w :
[[11 12]
 [13 14]
 [15 16]]
New Array :
[[ 1 11 12]
 [ 2 13 14]
 [ 3 15 16]]
```

Activate Win
Go to Settings to

Task 8 & 9 :

Q#8: Print all elements of "newArray" using the loop.

```
[466]: # write your code here ^_^
for row in newArray :
    for element in row:
        print(element)
```

```
1
11
12
2
13
14
3
15
16
```

Q#9: Reverse the columns and rows of "newArray".

```
[469]: # write your code here ^_^
print("Befor Reverse : ""\n", newArray)
print("-----")
newArray=newArray[::-1,:-1]
print("After Reverse : ""\n", newArray)
```

```
Befor Reverse :
[[ 1 11 12]
 [ 2 13 14]
 [ 3 15 16]]
```

```
-----
After Reverse :
[[16 15 3]
 [14 13 2]
 [12 11 1]]
```

Task 10 & 11 :

Q#10: Decrement all elements of "newArray" with 1.

```
[472]: # write your code here ^_^
print("Befor Decrement : "\n", newArray)
print("-----")
newArray -= 1
print("After Decrement : "\n", newArray)
```

```
Befor Decrement :
[[16 15  3]
 [14 13  2]
 [12 11  1]]
-----
After Decrement :
[[15 14  2]
 [13 12  1]
 [11 10  0]]
```

Q#11: Find smallest and biggest values in "newArray".

```
[475]: # write your code here ^_^
print(newArray)
smallnum = np.min(newArray)
print (f"Smallest number : {smallnum}")
bignum = np.max(newArray)
print (f"Biggest number : {bignum}")
```

```
[[15 14  2]
 [13 12  1]
 [11 10  0]]
Smallest number : 0
Biggest number : 15
```

Task 12 & 13 & 14:

Q#12: Print the first row of "newArray" using indexing.

```
[478]: # write your code here ^_^
print(newArray)
print("First row : ",newArray[0])

[[15 14  2]
 [13 12  1]
 [11 10  0]]
First row :  [15 14  2]
```

Q#13: Print the number equals 12 of "newArray" using indexing.

```
[480]: # write your code here ^_^
number_12 = newArray[1][1]
print(number_12)

12
```

Q#14: Print the numbers equal 0 and 13 of "newArray" using indexing.

```
[483]: # write your code here ^_^
print("Numbers equal to 0:", newArray[newArray == 0])
print("Numbers equal to 13:", newArray[newArray == 13])
```

```
Numbers equal to 0: [0]
Numbers equal to 13: [13]
```

Task 15 :

Q#15: Change the shape of "newArray" to (9,1).

```
[501]: # write your code here ^_^
print("Befor change the shape : ""\n",newArray)
print("-----")

newArray=newArray.reshape(9,1)
print("After change the shape : ""\n",newArray)
```

Befor change the shape :

```
[[15 14  2]
 [13 12  1]
 [11 10  0]]
```

After change the shape :

```
[[15]
 [14]
 [ 2]
 [13]
 [12]
 [ 1]
 [11]
 [10]
 [ 0]]
```

Task 16 :

Note : can not change shape from (9,1) to (3,2) because first have 9 elements and second shape have 6 elements

Q#16: Change the shape of "newArray" to (3,2).

```
13]: # write your code here ^_^
#-- can not change shape from (9,1) to (3,2) becuse first have 9 elements and seconed shape have 6 elements--
print("Befor change the shape : ""\n",newArray)
print("-----")
newArray = newArray.reshape(3,3)
print("After change the shape : ""\n",newArray)
```

Befor change the shape :

```
[[15]
 [14]
 [ 2]
 [13]
 [12]
 [ 1]
 [11]
 [10]
 [ 0]]
```

After change the shape :

```
[[15 14  2]
 [13 12  1]
 [11 10  0]]
```