NumPy Searching Arrays

Find the indexes where the value is 4:

```
import numpy as np
arr = np.array ([1,2,3,4,5,4,4])
x = np. where (arr==4)
print (x)
```

Output ~~Index 3, 5 and 6~~ ('array[3,5,6], dtype=int64),

Find the indexes where the values are even:

```
import numpy as np
arr = np. array ([1,2 13, 4, 5, 6, 7, 8])
x = np. where ( arr % 2 ==0)
print (x)
```

Output

(array([[1, 3, 5, 7]),} dtype = int 64),)

Find the indexes where the values are odd:

```
import numpy as np
arr = np.array (([1, 2, 3, 4, 5, 6, 7, 8])
x = np. where (arr % 2 ==1)
print (x)
```

Output

(array([0, 2, 4, 6]),} dtype = int 64),)

Search Sorted

```
import numpy as np
arr = np.array ([6, 7,8, 9])
x = np. searchsorted (arr, 7)
print (x)
```

output

1

Search from the Right Side

```
import numpy as np
arr = np.array([6,7,8,9])
x = np.searchsorted(arr,7, side='right')
print(x)        O/P → 2
```

Search Multiple Values

```
import numpy as np
arr = np.array([1,3,5,7])
x = np.searchsorted(arr,[2,4,6])
print(x)

array [1 2 3]
```

Numpy Sorting Arrays

⊙ Sort the array

```
import numpy as np
arr = np.array([3,2,0,1])
point (np.sort(arr))
```

Output

```
[0 1 2 3]
```

⊙ Sort the array alphabetically

```
import numpy as np
arr = np.array(['banana', 'cherry', 'apple'])
print (np.sort(arr))
```

Output

```
['apple' 'banana' 'cherry']
```

⑤ Sort a boolean array.

```
import numpy as np
arr= np.array([True, False, True])
print (np. sort (arr))
```

Output

```
[False True True]
```

⑥ Sorting a 2-D Array

```
import numpy as np
arr = np.array([[3,2,4],[5,0,1]])
print(np.sort(arr))
```

Output

```
[[2 3 4]
  [0 1 5]]
```

NumPy Filter Array

Getting some elements out of an existing array and creating a new array out of them is called filtering.

Create an array from the elements on index 0 and 2

```
import numpy as np
arr = np. array([41,42,43,44])
n= [True, False, True, false]
newarr = arr [x]
print ( newarr )
```

Output

```
[41, 43]
```

⊙ Creating the Filter Array

```
import numpy as np
arr = np.array([41, 42, 43, 44])
# create an empty list
 filter_arr = []
# go through each element in arr
 for element in arr:
# if the element is higher than 42, set
the value to True, otherwise False;
        if element > 42:
            filter_arr.append(True)
        else:
            filter_arr.append(False)
 newarr = arr[filter_arr]
 print(filter_arr)
 print(newarr)
```

Output

```
[False, False, True, True]
[~~43~~, [43  44]
```

Create a filter array that will return only even elements from the original array:

```
import numpy as np
arr = np.array([1,2,3,4,5,6,7])
# create an empty list
 filter_arr = []
# go through each element in arr
for element in arr:
   # if the element is completely
       divisible by 2, set the value to
         True, otherwise False
      if element % 2 == 0:
          filteredarr.append( True)
      else:
          filter_arr.append(False)
```

```
new arr = arr [filter_arr]
 print (filter-arr)
 print (newarr)
```

Output

```
[False, True, False, True, False, True, False]
 [2 4 6]
```

⊙ Creating Filter Directly From Array

```
import numpy as np
arr = np.array([41,42,43,44])
filter_arr = arr>42
newarr = arr[filter_arr]        O/P
print (filter-arr]          [F F T T]
print (newarr)               [43 44]
```

② Creating a filter array that will return only even elements from the original array:

```
import numpy as np
arr = np.array([1,2,3,4,5,6,7])
filter_arr = arr%2 == 0
newarr = arr[filter_arr]
 print (filter_arr)
  print (newarr)
```

Output

```
[False True F T F T F]
[2 4 6]
```