

Solution Review

This lesson gives a detailed solution review of the problem.

We'll cover the following



- Solution

Solution

Here's the merged solution to the problem that we discussed in the previous lesson. The solution also takes into account these two factors:

- Negative steps
- Multi-dimensional arrays

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4
5 def find_index(base, view):
6     """
7     Given an array that is a view of
8     `base[index]` is view
9     """
10
11     if not isinstance(view, np.ndarray):
12         return "..."
13
14     itemsize = view.itemsize
15
16     # Find the start and end positions
17     offset_start = (np.byte_bounds(base)[0] +
18                     np.byte_bounds(view)[0])
19     offset_stop = (np.byte_bounds(base)[1] +
20                   np.byte_bounds(view)[1])
21
22     # Calculate the start and stop indices
23     index_start = np.unravel_index(offset_start, base.shape)
24     index_stop = np.unravel_index(offset_stop, base.shape)
25
26     # Use the strides property to get the step
27     index_step = np.array(view.strides)
```



```
28     for i in range(len(index_s
29         start = index_start[i]
30         stop = index_stop[i]
31         step = index_step[i]
```



Now that we have learned about the anatomy of an array, let's move on to the next chapter “Code Vectorization”.