**NumPy – Advanced Indexing**

The slicing in NumPy array is only **used to present a view**whereas advanced indexing always returns a **copy of the data.** Advanced indexing is caused, when the selection object from ndarray is a non-tuple sequence object, [ndarray](https://numpy.org/doc/stable/reference/generated/numpy.ndarray.html#numpy.ndarray) object of integer or boolean data type, or a tuple with at least one sequence object.

Advanced indexing is used if the random numbers are selected from different rows in a ndarray and the items to choose are in no sequence. There are two types of advanced indexing: integer and Boolean.

**Integer indexing:**

Integer array indexing mechanism helps to selection of arbitrary items in the array based on its N-dimensional index. Each integer array represents a number of indexes into that dimension.

When the index consists of many integer arrays as the dimensions of target ndarray, it becomes too straightforward at that time.

**Example 1:**

One element of specified column from each row of ndarray object is selected. Thus, the row index contains all row numbers, and the column index specifies the element to be selected. The selection includes elements at (0,0), (1,1) and (2,0) from the first array.

In advanced indexing, first specify the row numbers of all the rows to be choosen and then specify the index of the actual element of that row to be choosen. Then creating a new ndarray y for the ndarray x.

Here [0,1,2,3] means the rows to be picked are 0th row, 1st row, 2nd row and 3rd row. Also, [0,1,0,0] means the 0th row pick the [0] index element, 1st row pick [1] index element, 2nd row picks the [0] index element and 3rd row pick the [0] index element.

**Example 2:**

For the below example, elements placed at corners of a 4X3 array are selected. The row indices of selection are [0, 0] and [3,3] whereas the column indices are [0,2] and [0,2].

**Boolean Indexing:**

Boolean Indexing is the type of advanced indexing that is used when elements are choosen from a ndarray based on some condition using comparison operators or some other operator.

**Example 3:**

**Example 4:**

Not a number (NaN) elements are omitted by using ~ (Complement operator).

**Example 5:**

To filter out the non-complex elements from an array.