What are object arrays?

- Object arrays are basically array of any Python datatype.

Documentation: https://numpy.org/devdocs/reference/arrays.scalars.html#numpy.object

```
[ ] arr = np.array([1, 'm', [1,2,3]], dtype = 'object')
    arr
```

```
array([1, 'm', list([1, 2, 3])], dtype=object)
```

There is an exception to `.copy()`:

- **`.copy()` behaves as shallow copy when using `dtype='object'` array.**
- It will not copy object elements within arrays.

But arrays are supposed to be homogeous data. How is it storing data of various types?

Remember that everything is object in Python.

Just like Python list,

- The data actually **stored** in object arrays are **references to Python objects**, not the objects themselves.

Hence, their elements need not be of the same Python type.

As every element in array is an object, therefore the dtype=object.

Let's make a copy of object array and check whether it returns a shallow copy or deep copy.

```
[ ] copy_arr = arr.copy()
```

```
[ ] copy_arr
```

```
array([1, 'm', list([1, 2, 3])], dtype=object)
```

Now, let's try to modify the list elements in `copy_arr`.

```
[ ] copy_arr[2][0] = 999
```

```
[ ] copy_arr
```

```
array([1, 'm', list([999, 2, 3])], dtype=object)
```

Let's see if it changed the original array as well.

```
[ ] arr
```

```
array([1, 'm', list([999, 2, 3])], dtype=object)
```

It did change the original array.

Hence, **`.copy()` will return shallow copy when copying elements of array in object array.**

Any change in the 2nd level elements of array will be reflected in original array as well.

So, how do we create deep copy then?

We can do so using `copy.deepcopy()` method.

`copy.deepcopy()`

- Returns the deep copy of an array.

Documentation: https://docs.python.org/3/library/copy.html#copy.deepcopy

```
[ ] import copy
```

```
[ ] arr = np.array([1, 'm', [1,2,3]], dtype = 'object')
    arr
```

```
array([1, 'm', list([1, 2, 3])], dtype=object)
```

Let's make a copy using `deepcopy()`.

```
[ ] copy = copy.deepcopy(arr)
    copy
```

```
array([1, 'm', list([1, 2, 3])], dtype=object)
```

Let's modify the array inside copy array.

```
[ ] copy[2][0] = 999
    copy
```

```
array([1, 'm', list([999, 2, 3])], dtype=object)
```

```
[ ] arr
```

```
array([1, 'm', list([1, 2, 3])], dtype=object)
```

Notice that the changes in copy array didn't reflect back to original array.

`copy.deepcopy()` returns deep copy of an array.