Numpy Array Attributes

Attributes of Numpy arrays or ndarrays are properties or characteristics associated with these arrays that provide information about their structure and content. These attributes offer insights into the array's dimensions, data type, size, and memory usage etc.
Following are the main attributes of ndArrays:

1. ndarray.shape
2. ndarray.ndim
3. numpy.itemsize
4. numpy.flags

1. **ndarray.shape**: This array property provides a tuple containing the dimensions of the array and can also be utilized for resizing the array.

Example Code 1:

```
import numpy as np
a = np.array([[4,5,6],[7,8,9]])
print(a.shape)

Output
(2,3)
```

The code creates a 2x3 array named a, and prints its shape, which is (2, 3), indicating 2 rows and 3 columns.

Example Code 2:

```
# this resizes the ndarray
import numpy as np

a = np.array([[4,5,6],[7,8,9]])
a.shape = (3,2)
print(a)


Output
[[4 5]
 [6 7]
 [8 9]]
```

This code snippet creates a 2x3 array **a**, then resizes it to a 3x2 array using the **shape** property. Finally, it prints the resized array.

NumPy offers a reshape function for resizing arrays as well.

Example Code:

```
import numpy as np
a = np.array([[4,5,6],[7,8,9]])
b = a.reshape(3,2)
print(b)

Output
[[4 5]
 [6 7]
 [8 9]]
```

The code creates a NumPy array **a** with shape (2,3), then reshapes it to a new array **b** with shape (3,2), and finally prints the reshaped array b.

2. **ndarray.ndim**: This array attributes provides the number of dimensions (axes) in the array.

Example Code:

```
import numpy as np

# Create a 1-dimensional array
arr_1d = np.array([1, 2, 3])
print(arr_1d.ndim)

# Create a 3-dimensional array
arr_3d = np.array([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
print(arr_3d.ndim)

Output
1
3
```

In the code **arr_1d** is a 1-dimensional array, and **arr_3d** is a 3-dimensional array. The **ndim** attribute is used to query the number of dimensions of each array.

3. **numpy.itemsize:** This array attributes provides the size of each element in the array, measured in bytes**.**

```
# dtype of array is float32 (4 bytes)
import numpy as np
x = np.array([1,2,3,4,5], dtype = np.float32)
print(x.itemsize)

Output
4
```

Array x contains 32-bit floating-point numbers, so each element occupies 4 bytes.

4. **numpy.flags:** This function returns the current values of the attributes of the ndarray object.

Sr.No.	Attribute & Description
1	C_CONTIGUOUS (C) - The data is organized in a single, C-style contiguous segment.
2	F_CONTIGUOUS (F) - The data is organized in a single, Fortran-style contiguous segment.
3	OWNDATA (O) - The array either owns the memory it uses or borrows it from another object.
4	WRITEABLE (W) - The data area is writable; setting to False locks the data, making it read-only.
5	ALIGNED (A) - Both data and elements are appropriately aligned for the hardware.
6	UPDATEIFCOPY (U) - This array serves as a copy of another array; deallocation updates the base array with its contents.

Example Code:

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

Output
C_CONTIGUOUS : True
F_CONTIGUOUS : True
OWNDATA : True
WRITEABLE : True
ALIGNED : True
WRITEBACKIFCOPY : False
```

The code creates a NumPy array 'x' with elements [3,4,5,6,7] and prints its flags indicating memory layout and properties.