Using the numpy.array Function

numpy.array function:

The basic ndarray is created using array function in NumPy as follows –

```
numpy.array
```

The **numpy.array** function is a powerful constructor in NumPy, allowing the creation of arrays with various parameters. Below is an overview of its parameters:

```
numpy.array(object, dtype = None, copy = True, order =
None, subok = False, ndmin = 0)
```

S. No.	Parameter	Description
1.	Object	An array is returned by any object that exposes the array interface method, or alternatively, any (nested) sequence.
2.	Dtype	Desired data type of the array, optional.
3.	Copy	Optional specification of the desired data type for the array.
4.	Order	Specify either C (row major), F (column major), or A (any) to determine the array layout.
5.	Subok	By default, the array that is returned is coerced into being a base class array. If set to true, sub-classes will be allowed to pass through without coercion.
6.	ndmin	Defines the minimum dimensions of the resulting array.

Let us make Python programs to understand the ndarray better.

1. **Creating 1-D Array:** Python program to make a one dimensional array.

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

Output

```
[4 6 8]
```

This code creates a NumPy array named a, with the elements 4, 6, and 8, and then prints the array to the console.

i. **import numpy as np**: This line imports the NumPy library and gives it the alias **np**. This is a common convention to make the code more readable and concise. Now, you can use **np** instead of typing out **numpy** every time you want to use a function from the library.

ii. **a = np.array([4, 6, 8])**: This line creates a NumPy array named **a** with the values 4, 6, and 8. NumPy arrays are similar to lists in Python but have additional functionalities and optimizations for numerical operations.

iii. **print(a)**: This line prints the contents of the array **a** to the console. In this case, it will output something like:

2. **Creating 2-D Array**: Python program to create two dimensional array.

```
# multi- dimension arrays
import numpy as np
a = np.array([[3, 4], [9, 6]])
print(a)
```

Output

```
[[3 4]
 [9 6]]
```

This code creates a 2-dimensional NumPy array named a, with elements arranged in two rows and two columns and then prints the array to the console.

i. **import numpy as np**: This line imports the NumPy library and aliases it as **np** for convenience.

ii. **a = np.array([[3, 4], [9, 6]])**: This line creates a NumPy array named **a**. The array is a 2x2 matrix, meaning it has two rows and two columns. The values in the matrix are provided as a nested list **[[3, 4], [9, 6]]**.

3. **Creating Array with minimum dimension**: You can create an array with a minimum dimension using the ndmin parameter when creating the array. This parameter specifies the minimum number of dimensions that the resulting array should have.

```
# minimum dimensions
import numpy as np
a = np.array([1, 2, 3,4,5], ndmin = 2)
print(a)
```

Output

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

This code uses array function to create a 2-dimensional array with elements 1, 2, 3, 4, and 5. The ndmin=2 parameter ensures that the array has a minimum of two dimensions, even though the input is a 1-dimensional list. The output is a 2D array with a single row and five columns.

4. **Creating Array of Complex Numbers:** You can create an array of complex numbers by specifying the dtype parameter as complex when creating the array.

```
# dtype parameter
import numpy as np
a = np.array([1, 2, 3], dtype = complex)
print(a)
```

Output

```
[1.+0.j 2.+0.j 3.+0.j]
```

The dtype parameter specifies the data type of array elements. Here, dtype=complex is used to create an array of complex numbers from the integer values 1, 2, and 3.